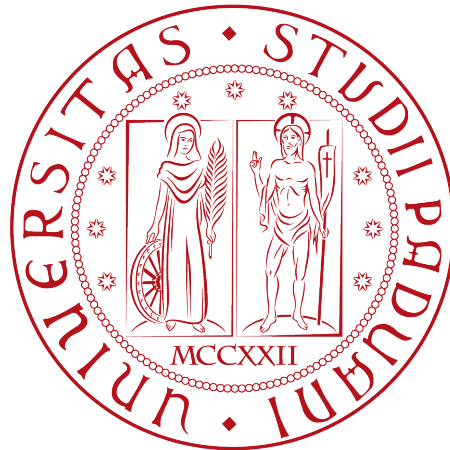


Università degli Studi di Padova

DIPARTIMENTO DI MATEMATICA “TULLIO LEVI-CIVITA”

CORSO DI LAUREA IN INFORMATICA



Analisi, progettazione e sviluppo di
un'applicazione web per la gestione di
distributori automatici

Tesi di laurea

Relatore

Prof.ssa Ombretta Gaggi

Laureando

Luca Polese

1225425

ANNO ACCADEMICO 2022-2023

Luca Polese: *Analisi, progettazione e sviluppo di un'applicazione web per la gestione di distributori automatici*, Tesi di laurea, © Luglio 2023.

“ Anyone who has lost track of time when using a computer knows the propensity to dream, the urge to make dreams come true and the tendency to miss lunch. ”

Tim Berners Lee



Dedicato ai miei genitori e a Valentina

Dedicato a chi è sempre stato al mio fianco, fisicamente o con il pensiero.
Dedicato a chi non ha potuto esserci, ma che in qualche modo mi ha sempre sostenuto.
I miei traguardi sono anche merito vostro.

Dedicato a me per non aver mai mollato e per aver continuato a credere in me stesso.

Sommario

Il presente documento descrive il lavoro svolto durante il periodo di stage, della durata di circa trecento ore, dal laureando Luca Polese presso l'azienda *Moku S.r.l* nel periodo che va dal 08/05/2023 al 30/06/2023.

L'obiettivo principale da raggiungere era quello di sviluppare delle interfacce di una *web app* che permettesse agli amministratori del progetto **Da Mario** di accedere alla *dashboard* principale e visualizzare le statistiche relative all'andamento delle vendite suddivise per *location*, macchinette ed eventualmente prodotto. Successivamente è stata introdotta la possibilità di configurare i prodotti legando i prodotti macchinetta con quelli piattaforma.

Il mio lavoro si è concentrato in particolar modo sull'analisi dei requisiti, sulla progettazione del sistema da realizzare e sullo sviluppo dell'applicazione attraverso il *framework* *Angular*. Tutti i dati vengono ricavati dal *backend* tramite delle *query* *GraphQL*.

Alla conclusione dello stage, ho realizzato tutte le statistiche richieste dal cliente, e ho iniziato lo sviluppo della pagina per la configurazione dei prodotti. Il sito è stato pubblicato in ambiente *staging* nella sua prima versione.

“This is only a foretaste of what is to come,
and only the shadow of what is going to be.”

— Alan Mathison Turing

‘The Mechanical Brain. Answer Found to 300 Year Old Problem’
The Times newspaper
11 Giugno 1949

Ringraziamenti

Al termine del mio percorso accademico triennale e della stesura di questa tesi, desidero dedicare questa sezione per ringraziare tutte le persone che mi sono state vicine.

Desidero innanzitutto ringraziare i miei genitori che mi sono sempre rimasti accanto durante tutti gli alti e bassi che ci sono stati in questi anni. Grazie per aver condiviso con me tutte le gioie e le soddisfazioni, ma anche per avermi dato la forza di guardare oltre nei momenti più complicati. Il vostro incoraggiamento è ciò che mi ha permesso di non fermarmi di fronte alle difficoltà e di puntare sempre più in alto. Il vostro amore e il vostro sostegno non sono mai mancati e per questo non finirò mai di esservi grato.

Un pensiero speciale va soprattutto a Valentina, che ha saputo supportarmi, ma soprattutto sopportarmi, durante questo percorso. I suoi preziosi consigli rimangono sempre custoditi nella mia mente. Grazie per avermi ascoltato parlare per ore di informatica, per aver fatto finta di capire e per avermi sempre dato ragione. In questo modo ha alimentato la mia passione per questa materia.

Un ringraziamento va anche alle mie nonne, ai miei cugini e ai miei zii: il loro appoggio mi ha dato la carica per poter affrontare le sfide con la mentalità giusta, proiettato verso nuovi obiettivi. Le vostre parole, la vostra presenza e la vostra esperienza hanno contribuito a rendermi ciò che sono.

In particolare, vorrei esprimere la mia gratitudine alla Professoressa Ombretta Gaggi, relatrice della mia tesi, per avermi pazientemente seguito durante i mesi di stage e per il sostegno fornito durante la stesura del mio lavoro. A lei va il mio pensiero di stima profonda, in particolar modo per aver sempre prontamente risposto alle mie domande. La ringrazio anche per avermi prestato degli introvabili volumi relativi agli argomenti del suo corso.

Per ultimi, ma non per importanza, ringrazio tutti i ragazzi della compagnia dei RVRs con i quali ho trascorso dei momenti indimenticabili. Grazie per avermi aiutato a "staccare" la mente dalle preoccupazioni e dagli impegni universitari. Una particolare riconoscenza va anche ai miei amici e colleghi del gruppo UniPD - Terzo Banco. In-

sieme ne abbiamo passate veramente tante in questi anni! Grazie per i chiarimenti, gli appunti, i ripassi, le chiacchierate prima e dopo le lezioni e gli studi in vista degli esami: la vostra presenza ha dato colore anche ai giorni più grigi.

Padova, Luglio 2023

Luca Polese

Indice

1	Introduzione	1
1.1	L'azienda	1
1.2	Ruoli aziendali	2
1.3	L'idea di progetto	4
1.4	La soluzione individuata	5
1.5	Organizzazione del testo	7
2	Processi e metodologia di sviluppo	8
2.1	Situazione attuale nel panorama informatico	8
2.2	Metodologia agile: nascita e principi	9
2.3	Metodo di sviluppo adottato in <i>Moku S.r.l</i>	9
2.3.1	User stories	11
2.4	Strumenti a supporto dei processi	11
2.4.1	Jira	11
2.4.2	Bitbucket	13
2.4.3	Slack	14
2.4.4	Google Calendar	14
2.4.5	Passbolt	14
3	Descrizione dello stage	15
3.1	Il contesto del progetto	15
3.1.1	Struttura del <i>backend</i>	17
3.2	Analisi preventiva dei rischi	18
3.3	Obiettivi dello stage	27
3.4	Vincoli tecnologici	28
3.4.1	Angular	28
3.4.2	TypeScript	29
3.4.3	GraphQL	29
3.4.4	Apollo Client	29
3.4.5	WebStorm	30
3.5	Il valore accademico dello stage	30
4	Analisi dei requisiti	31
4.1	Personas	31
4.2	Entità	31
4.2.1	Piattaforma web	31
4.2.2	Sezione	32
4.2.3	Amministratore	32

4.2.4	<i>Location</i>	32
4.2.5	Device (Macchinetta)	33
4.2.6	Prodotto	33
4.3	Casi d'uso	34
4.3.1	Casi d'Uso - Riconoscimento dell'utente amministratore	34
4.3.2	Casi d'Uso - Statistiche "Vendite totali"	36
4.3.3	Casi d'Uso - Statistiche "Prodotti più venduti"	38
4.3.4	Casi d'Uso - Statistiche "Numero di utenti"	41
4.3.5	Casi d'Uso - Statistiche "Spesa media utenti"	43
4.3.6	Casi d'Uso - Statistiche "Spesa per fascia oraria"	45
4.3.7	Casi d'Uso - Macchine	48
4.3.8	Casi d'Uso - Prodotti	55
4.3.9	Casi d'Uso - Utenti	58
4.3.10	Casi d'Uso - Promozioni	58
4.4	Tracciamento dei requisiti	61
4.4.1	Requisiti funzionali	62
4.4.2	Requisiti di vincolo	65
4.4.3	Requisiti qualitativi	65
5	Progettazione e codifica	67
5.1	Progettazione architetturale	67
5.1.1	Architettura di Angular	67
5.1.2	Architettura del prodotto	69
5.2	Progettazione delle interfacce grafiche	69
5.2.1	Interfaccia di login	70
5.2.2	Pagina principale	71
5.3	Codifica	86
5.3.1	<i>Design Pattern</i> utilizzati	86
5.3.2	Struttura <i>software</i>	88
5.3.3	Interfacce realizzate	89
5.3.4	Pubblicazione del sito	94
6	Verifica e validazione	96
6.1	Verifica	96
6.1.1	Lighthouse	96
6.1.2	Browser testati	97
6.1.3	Dispositivi testati	98
6.1.4	Analisi statica	98
6.1.5	Analisi dinamica	99
6.2	Validazione	99
7	Conclusioni	100
7.1	Consuntivo finale	101
7.2	Raggiungimento degli obiettivi	101
7.3	Conoscenze acquisite	102
7.4	Valutazione personale	103
	Acronimi e abbreviazioni	104
	Glossario	105

INDICE

viii

Bibliografia

111

Elenco delle figure

1.1	Logo dell'azienda Moku S.r.l	1
1.2	Logo dell'applicazione Da Mario	4
1.3	Schermata principale del <i>plugin Active Admin</i> contenente dati di <i>staging</i>	5
2.1	<i>Scrum</i> - rappresentazione grafica del ciclo di sviluppo	10
2.2	<i>Jira</i> - esempio di <i>board</i> raffigurante il primo <i>Sprint</i>	12
2.3	Tempo - esempio di <i>timesheet</i> settimanale	13
2.4	<i>Bitbucket</i> - schermata principale del <i>repository</i>	13
2.5	<i>Slack</i> - schermata principale di esempio	14
3.1	Esempi di dispositivi <i>Cimbali</i> e <i>MageX</i>	16
3.2	Raffigurazione dello schema del <i>backend</i>	18
5.1	Architettura del <i>framework Angular</i>	68
5.2	Tavola raffigurante la pagina di <i>login</i> alla piattaforma	70
5.3	Tavola raffigurante la <i>dashboard</i>	72
5.4	Tavola raffigurante le statistiche relative alle vendite totali	73
5.5	Tavola raffigurante le statistiche relative ai prodotti più venduti	74
5.6	Tavola raffigurante le statistiche relative al numero di utenti che ha effettuato almeno un acquisto	74
5.7	Tavola raffigurante le statistiche relative alla spesa media per utente	75
5.8	Tavola raffigurante le statistiche relative alla spesa per fascia oraria	76
5.9	Tavola raffigurante la pagina contenente le <i>location</i> con le relative macchinette	77
5.10	Tavola raffigurante la pagina contenente le macchinette con i relativi prodotti	78
5.11	Tavola raffigurante i dettagli di un prodotto da configurare a cui non è stato assegnato il prodotto piattaforma	79
5.12	Tavola raffigurante i dettagli di un prodotto da configurare a cui è stato assegnato il prodotto piattaforma	80
5.13	Tavola raffigurante i dettagli di un prodotto configurato con una promozione attiva	81
5.14	Tavola raffigurante la pagina contenente i prodotti piattaforma	82
5.15	Tavola raffigurante la pagina contenente i dettagli dei prodotti piattaforma	83
5.16	Tavola raffigurante la pagina contenente gli utenti del sistema	84
5.17	Tavola raffigurante la pagina contenente le promozioni del sistema	85
5.18	Tavola raffigurante la pagina contenente la configurazione delle promozioni del sistema	86

5.19	Screenshot che rappresenta l'interfaccia di <i>login</i>	90
5.20	Screenshot che rappresenta l'interfaccia di navigazione	90
5.21	Screenshot che rappresenta le statistiche relative alle vendite totali	91
5.22	Screenshot che rappresenta le statistiche relative ai prodotti più venduti	92
5.23	Screenshot che rappresenta le statistiche relative al numero di utenti	92
5.24	Screenshot che rappresenta le statistiche relative alla spesa media degli utenti	93
5.25	Screenshot che rappresenta le statistiche relative spesa per fascia oraria	93
5.26	Screenshot che rappresenta l'interfaccia relativa alle location	94
5.27	Screenshot che rappresenta l'interfaccia relativa alla specifica macchinetta	94
6.1	Risultati del report di Lighthouse	97

Elenco delle tabelle

1.1	Tabella delle figure aziendali	3
3.1	Analisi sull'inesperienza con le tecnologie da utilizzare	20
3.2	Analisi sulla perdita di file	21
3.3	Analisi dei problemi legati all'uso di <i>software</i> di terze parti	22
3.4	Analisi dei problemi legati agli impegni personali e/o universitari	24
3.5	Analisi dei problemi legati agli elevati costi delle attività	25
3.6	Analisi dei problemi derivanti da conflitti/contrasti fra membri	26
3.7	Analisi dei problemi legati alla scarsa comunicazione fra membri del team	27
4.1	Tabella che raccoglie la tracciatura dei requisiti funzionali del progetto	65
4.2	Tabella che raccoglie la tracciatura dei requisiti di vincolo del progetto	65
4.3	Tabella che raccoglie la tracciatura dei requisiti qualitativi del progetto	66
7.1	Tabella con il consuntivo delle ore utilizzate durante il progetto	101
7.2	Tabella riassuntiva degli obiettivi soddisfatti	101
7.3	Tabella riassuntiva degli obiettivi parzialmente soddisfatti	102
7.4	Tabella contenente l'obiettivo obbligatorio non soddisfatto	102

Capitolo 1

Introduzione

Questo capitolo introduttivo fornisce una breve presentazione dell'azienda ospitante, di com'è composta e dell'ambito in cui opera. Una parte del capitolo è stata dedicata inoltre alla descrizione del progetto di stage, alla richiesta del cliente e alla soluzione trovata.

1.1 L'azienda

L'azienda ospitante è *Moku S.r.l.*¹ (Figura 1.1), una *software house*^[g] situata in via Schiavonia 3 a Treviso e che ha sede legale a Roncade (TV). Attualmente la società conta più di venti dipendenti che si suddividono fra sviluppatori *frontend*^[g], *backend*^[g], *mobile*^[g] e designer di prodotto.



Figura 1.1: Logo dell'azienda Moku S.r.l

L'azienda nasce nel 2013 come una *startup*^[g] incubata da *H-Farm*^[g] e per i primi anni di vita sviluppa ed evolve l'omonimo progetto², che ha l'obiettivo di aiutare gli studenti universitari a digitalizzare i propri appunti e a prendere annotazioni.

Con l'evolversi dell'idea imprenditoriale, *Moku S.r.l* ha indirizzato il proprio *core business* verso la fornitura di servizi a terzi. In questa ottica il team segue tutte le fasi del progetto a partire dalla definizione dei requisiti e dalla realizzazione della *user experience (UX)*^[g] fino alla consegna dei prodotti che vengono loro assegnati.

L'obiettivo principale dell'azienda è quello di instaurare una stretta collaborazione con il cliente, accompagnandolo lungo l'intero processo di sviluppo del prodotto. Successivamente, viene fornito il supporto post vendita al cliente.

¹*Moku S.r.l.* URL: <https://www.moku.io/it/>.

²una piattaforma per semplificare lo studio, con cui prendere appunti in modo collaborativo e condividere facilmente le proprie dispense eliminando la necessità di fare molteplici copie del proprio materiale

Attualmente, *Moku S.r.l* assume il ruolo di guida dell'innovazione. Dopo un iniziale incontro con il cliente, il team realizza un piano operativo e strategico mirato a potenziare l'efficacia e, l'efficienza dei processi aziendali.

Fra i vari servizi proposti³ troviamo:

- Sviluppo *software*: sviluppo delle idee tramite un processo guidato che mette in relazione l'esperienza e il metodo di *Moku* con le necessità e la *vision* dell'azienda richiedente;
- Consulenza: aiutare le aziende a definire piani e strategie mirate al raggiungimento del target prestabilito. Il fornitore diventa anche *partner* tecnologico e prestando supporto al cliente nel suo *business* tramite il rispetto del *budget* e degli obiettivi fissati;
- *Visual design*: l'utente interagisce con le interfacce grafiche dei prodotti. Per ottenere le interfacce più semplici e comprensibili, l'azienda fornisce l'analisi del flusso delle informazioni e realizza delle tavole che rappresentino concretamente una soluzione ai problemi sottoposti.

1.2 Ruoli aziendali

L'organico aziendale è rappresentato da molteplici figure professionali. Ognuna di queste può essere identificata e raggruppata in base alle competenze specifiche, all'area in cui opera e al livello di responsabilità all'interno della società, come segue:

Ruolo	Competenze
CEO - <i>Chief Executive Officer</i> Simone Pozzobon	In italiano "Amministratore Delegato", è la figura che all'interno di un'organizzazione aziendale è il massimo responsabile: ha il compito di guidare la visione strategica dell'azienda, ha potere decisionale sulle azioni e sulla struttura dell'attività aziendale e coordina le operazioni assicurando gli obiettivi. Rappresenta l'azienda e cerca di attrarre investitori e/o talenti da inserire.
CTO - <i>Chief Technology Officer</i> Michele Redolfi	In italiano "Direttore tecnico" è un membro del consiglio di amministrazione aziendale ed ha il compito di coordinare la scelta e l'adozione di tecnologie nei progetti. Ha il compito di guidare lo sviluppo e l'implementazione delle soluzioni tecnologiche per supportare la visione e gli obiettivi dell'azienda supervisionando il reparto tecnico e collaborando con i team di sviluppo. Lavora a stretto contatto con il CEO e gli altri membri del team dirigenziale.

³*Moku S.r.l* - Servizi. URL: <https://www.moku.io/it/servizi/>.

Ruolo	Competenze
CFO - <i>Chief Financial Officer</i> Piera Nathalie Scalco	In italiano "Direttore finanziario", è un ruolo chiave per la gestione delle attività finanziarie e contabili dell'azienda. Il CFO gestisce il capitale sociale e l'amministrazione oltre ad altre funzioni finanziarie critiche. Il CFO deve far in modo di gestire le risorse economiche in modo tale da distribuirle fra i vari progetti e di utilizzarle in maniera efficiente .
CCO - <i>Chief Creative Officer</i> Marco Mezzavilla	In italiano "Direttore creativo", si occupa di sviluppare e implementare una strategia di comunicazione al fine di promuovere l'immagine dell'azienda. Il suo obiettivo è quello di creare dei messaggi chiave, di definire e sviluppare l'identità dell'azienda e di implementare iniziative di comunicazione orientate al messaggio del cliente.
Analisti	Ossia coloro che effettuano l'analisi dei requisiti di un prodotto. Analizzano pertanto il problema al fine di definire le necessità del cliente. Per fare ciò valutano i processi aziendali e il contesto in cui inserire il prodotto richiesto esaminandolo in termini economici e tecnologici. Definiscono i modelli e l'architettura del sistema ed i diagrammi di flusso.
Sviluppatori	Si occupano di realizzare la soluzione tramite l'attività di codifica. Partecipano alla fase di implementazione e manutenzione del prodotto. Qui verrà implementata l'architettura, assicurandosi che risponda alle specifiche. Le loro attività principali consistono in: scrivere codice mantenibile, creare test ad hoc per la verifica e la validazione del codice e redigere il manuale utente. Ogni programmatore ha competenze tecniche e forma la categoria più popolosa all'interno del progetto.
Designer	Il designer è un professionista specializzato nella progettazione della UX dei prodotti <i>software</i> . Il compito principale del designer è quello di lavorare a stretto contatto con gli sviluppatori e con il cliente finale del prodotto. Questo gli permette di comprendere quali sono le sue esigenze e che cosa devono poter fare gli utenti finali. Una volta derivati i requisiti, li traduce in soluzioni di design intuitivi e funzionali. Realizza inoltre interfacce utente coinvolgenti e user-friendly, che offrano un'esperienza piacevole e intuitiva agli utenti. Questo include la definizione dell'architettura dell'informazione, la progettazione dei layout, la scelta dei colori, dei tipi di carattere e degli elementi visivi e la creazione di prototipi.

Tabella 1.1: Tabella delle figure aziendali

1.3 L'idea di progetto

Il mercato dei distributori automatici è in continua espansione e la loro diffusione è in particolar modo favorita dalla facilità di automatizzazione del processo di erogazione dei prodotti e dal sostegno che apportano le macchinette che diventano sempre più *smart*.

CONFIDA⁴, l'associazione di categoria che rappresenta la filiera della distribuzione automatica di cibi e bevande a livello nazionale, realizza annualmente uno studio che monitora il mercato delle *vending machines*.

Secondo uno degli ultimi rapporti infatti:

il 2021 è stato l'anno della svolta per il settore della distribuzione automatica italiana. [...] il *vending*, infatti, ha chiuso l'anno con un fatturato di 1,435 miliardi di euro segnando una crescita del 12,11%^a

^aCONFIDA - Dati di settore. URL: <https://www.confida.com/dati-di-settore/>.

Nello specifico, il guadagno delle macchinette automatiche si è concentrato sulla vendita di bevande calde.

Moku S.r.l. sta sviluppando un'ecosistema per la gestione di *vending machine smart* per conto dell'azienda *Infinite Area*⁵. *Infinite Area* è un'importante realtà del territorio che si occupa di cultura d'impresa e progetti di innovazione. Offre spazi di *coworking*^[g] ed eventi atti a stimolare il cambiamento e a favorire la crescita delle aziende.

Il progetto gestito da *Moku S.r.l.* prende il nome di **Da Mario**⁶ (Figura 1.2) e vuole consentire agli utenti delle *location* gestite da *Infinite Area* di accedere ai servizi di *vending* attraverso un'applicazione mobile intuitiva, che permetta di gestire un proprio borsellino, di definire le proprie preferenze, di visualizzare gli acquisti effettuati.

Ogni utente può acquistare bevande calde o fredde e snack presso le *vending machine*. L'app offre un'esperienza di acquisto personalizzata, offrendo agli utenti di scegliere tra una vasta gamma di opzioni, personalizzare gli ordini e utilizzare funzionalità come il pagamento *mobile*.



Figura 1.2: Logo dell'applicazione Da Mario

I destinatari di questo prodotto sono pertanto gli utenti di *coworking*, uffici ed hotel, ma punta ad essere eventualmente esteso in futuro anche ad altre realtà.

Allo stato attuale l'ecosistema del prodotto presenta:

- un *backend* (sviluppato in *Ruby on Rails*^[g]);

⁴CONFIDA. URL: <https://www.confida.com/chi-siamo/>.

⁵*Infinite Area*. URL: <https://infinitearea.com/>.

⁶*Portfolio - Da Mario*. URL: <https://infinitearea.com/startup-studio/>.

- un app *mobile* (sia per sistemi iOS⁷ che per Android⁸);
- una piattaforma per l'amministrazione generata a partire dal *backend*.

L'interfaccia di amministrazione è stata precedentemente realizzata per mezzo del *plugin Active Admin*⁹ (Figura 1.3). Questo *plugin* consente di generare delle interfacce di tipo amministrativo in modo semplice, seguendo un linguaggio familiare allo sviluppatore *backend*. L'idea all'origine di questo *tool*, infatti, è quella di astrarre degli schemi comuni su cui si fondano le moderne applicazioni aziendali così da facilitare l'implementazione agli sviluppatori.

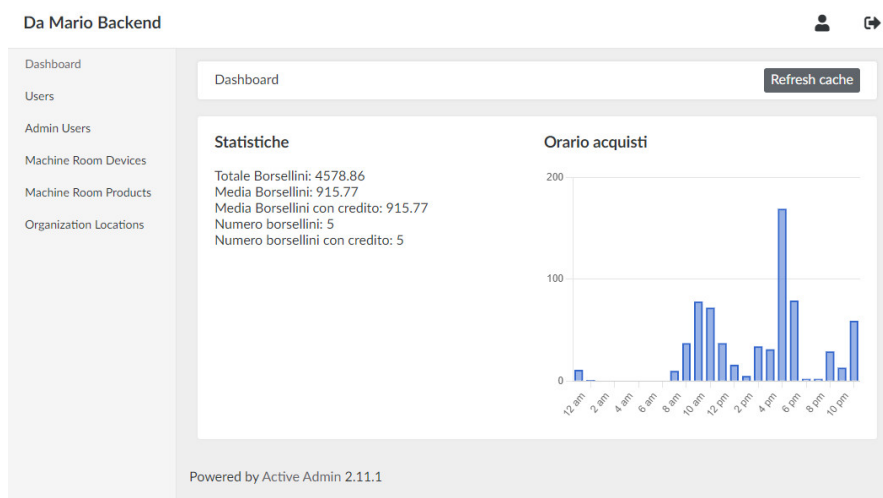


Figura 1.3: Schermata principale del *plugin Active Admin* contenente dati di *staging*

L'uso di *Active Admin* è funzionale alla visualizzazione dei dati lato amministrazione. Nell'applicazione attuale però, i dati esposti sotto forma di viste, risultano poco intuitivi e poco espressivi. Nell'ottica di avere una moltitudine di utenti amministratori con differenti conoscenze informatiche, l'attuale implementazione risulta limitata: l'utente amministratore ha uno spettro ridotto delle informazioni ricavabili dal *backend* e interagisce con una grafica statica e molto spesso in forma tabellare.

Il progetto di stage che mi è stato affidato, prevede di sviluppare delle maschere che riducano al minimo tali problematiche, rendendo appetibile e accattivante la lettura di statistiche e facilitando la gestione della personalizzazione dei prodotti presenti nei distributori.

1.4 La soluzione individuata

Il sistema è basato su servizi, il che ne favorisce la scalabilità.

Il servizio principale è quello denominato **Sala Macchine**, che interagisce con le macchinette e che effettua uno scambio di messaggi tramite *REST* in due direzioni:

⁷ *Da Mario - iOS*. URL: <https://apps.apple.com/us/app/da-mario/id1580905176/>.

⁸ *Da Mario - Android*. URL: https://play.google.com/store/apps/details?id=com.ia_shop/.

⁹ *Active Admin*. URL: <https://activeadmin.info/>.

- **Da Mario** inoltra le richieste per ottenere le informazioni di interesse a **Sala Macchine**;
- **Sala Macchine** risponde alle richieste di **Da Mario** oppure invia degli aggiornamenti tramite *webhooks* (come verrà discusso in seguito).

L'utente amministratore interagisce direttamente con l'interfaccia grafica che ho realizzato. Lo sviluppo di questa parte rappresenta interamente l'ambito del *frontend*.

Invece l'utente base del sistema utilizza l'app *mobile* per acquistare, personalizzare e salvare prodotti.

La *web app* sviluppata si occupa di fornire all'utente delle maschere che permettano di:

- effettuare il *login* alla piattaforma;
- visualizzare le principali statistiche relative ai dati di vendita filtrate per data/*location*/dispositivo/prodotto;
- personalizzare un prodotto.

Il *frontend* comunica direttamente con il *backend* attraverso le *Application Program Interface (API)*^[g] *GraphQL*^[g]. L'uso delle *API* consente di separare nettamente le due parti del sistema, riducendo sensibilmente le dipendenze. Nello specifico la dipendenza principale è rappresentata dall'interfaccia.

La scelta architetturale relativa alla piattaforma di amministrazione è stata discussa direttamente con team di sviluppo *frontend (FE)* dell'azienda. È così emerso che la soluzione ideale per questo progetto fosse quella di optare per una *Single Page Application*.

Una *Single Page Application* è un'app *web* in cui tutte le informazioni sono contenute in un'unica pagina *Hyper Text Markup Language (HTML)*^[g]. Nello specifico, tutti i dati del sito vengono scaricati all'apertura della pagina, pertanto non sarà necessario ricaricarla ad ogni richiesta dell'utente.

Tale scelta è favorita da due punti chiave:

- Il *framework* adottato per il progetto è *Angular*, il quale facilita lo sviluppo delle *Single Page Applications* e delle *Progressive Applications*;
- La *Single Page Application* permette di aggiornare dinamicamente i contenuti della pagina in base all'evento che è stato generato dall'utente.

Questa soluzione risulta essere adatta al progetto descritto in questa tesi, in quando, una volta effettuato il login, l'amministratore potrà aver accesso a tutte le schermate presenti nella piattaforma a partire dalla pagina principale. In ogni momento, l'utente potrà visualizzare i contenuti della pagina che ha selezionato nella barra di navigazione. Non sarà pertanto necessario un continuo *refresh* grazie all'applicazione delle rotte (argomento che verrà trattato in seguito).

L'uso di *Angular* invece è determinato principalmente dal fatto che la maggior parte dei progetti *FE* sviluppati dall'azienda adottano questa libreria. *Angular* ha una struttura complessa e rigida che si basa su moduli contenenti componenti. Il *framework* risulta più complesso dei diretti concorrenti, ma allo stesso tempo, avendo una struttura architetturale solida, fare scelte ingegneristiche sbagliate risulta meno probabile.

1.5 Organizzazione del testo

Il secondo capitolo introduce la situazione attuale nel panorama informatico per quanto riguarda le metodologie di sviluppo. Ho pertanto confrontato la questione della pianificazione del progetto con la necessità di velocizzare sempre più la produzione di *software*. Di seguito ho spiegato la metodologia agile, il *framework Scrum* e indicato come gestisce i propri processi l'azienda *Moku S.r.l.*

Il terzo capitolo tratta il contesto del progetto, in particolare approfondisce le tematiche su cui si basa. Ho descritto le varie parti che compongono il sistema nel dettaglio, in particolar modo ho approfondito il funzionamento del *backend*. In seguito ho elencato i rischi che ho analizzato prima dell'inizio dello stage e ho definito gli obiettivi del progetto che mi è stato affidato. Infine ho elencato le tecnologie che ho utilizzato ed illustra i punti relativi all'utilità dei tirocini durante il percorso accademico.

Il quarto capitolo raccoglie l'analisi dei requisiti che ho effettuato durante il primo approccio con il progetto che mi è stato assegnato. In particolare sono stati individuati gli utenti finali della *web-app*, le entità del progetto, i casi d'uso ed i requisiti.

Il quinto capitolo approfondisce lo studio effettuato in termini di progettazione dell'applicazione. Ho discusso com'è sviluppata l'architettura del sistema, basandomi su quella del *framework*. In seguito ho rappresentato tutte le funzionalità descritte nell'analisi dei requisiti sotto forma di bisogni lato software. Il capitolo continua con una descrizione dei *pattern* che ho implementato e della struttura del *software* che ho utilizzato. In conclusione ho elencato le interfacce che ho realizzato e spiegato il processo di pubblicazione del sito in ambiente di *staging*.

Il sesto capitolo descrive come sono state effettuate la verifica e la validazione, distinguendo nello specifico i *software* utilizzati.

Nel settimo capitolo illustra il consuntivo finale, gli obiettivi raggiunti, le conoscenze acquisite ed una valutazione personale dello stage.

Riguardo la stesura del testo, relativamente al documento sono state adottate le seguenti convenzioni tipografiche:

- gli acronimi, le abbreviazioni e i termini ambigui o di uso non comune menzionati vengono definiti nel glossario, situato alla fine del presente documento;
- per la prima occorrenza dei termini riportati nel glossario viene utilizzata la seguente nomenclatura: *parola*^[g] ;
- i termini in lingua straniera o facenti parti del gergo tecnico, i nomi di azienda e i nomi di prodotti sono evidenziati con il carattere *corsivo*;
- in **grassetto** verranno inseriti alcuni termini rilevanti all'interno del paragrafo, oppure il nome del progetto (e.g. **Da Mario**);
- in **monospace** verranno inseriti i nomi delle classi, i pezzi di codice estratti dal prodotto, i nomi delle funzioni e le variabili (e.g. **NgModule**);
- ogni riferimento ipertestuale (termini di glossario e note a piè di pagina) è identificabile tramite un *link* cliccabile.

Capitolo 2

Processi e metodologia di sviluppo

Questo capitolo introduce la situazione attuale legata alle scelte dei metodi di sviluppo, descrive il metodo agile ed il metodo di lavoro adottato dall'azienda, valutando specificamente il ciclo di vita del prodotto e gli strumenti utilizzati a sostegno del processo di sviluppo

2.1 Situazione attuale nel panorama informatico

A livello globale, la tendenza odierna delle aziende è quella di doversi adattare sempre di più ad un cambiamento rapido e continuo.

Le imprese si trovano spesso a dover affrontare un mercato in costante evoluzione che richiede un'adeguata risposta da parte delle stesse.

Al giorno d'oggi il *software* risulta essere un elemento fondamentale in qualsiasi dominio applicativo e settore poiché la maggior parte dei processi aziendali adottano un'automazione (come ad esempio la gestione degli ingressi, delle uscite, dei salari, delle ferie, dei permessi, etc.). Le aziende cercano pertanto di rimanere sempre più al passo con i tempi seguendo il progresso.

In particolar modo le aziende informatiche subiscono quotidianamente la pressione della concorrenza e devono essere sempre pronte a rispondere in maniera rapida ed efficace alle richieste del mercato, con condizioni economiche variabili.

Come afferma Ian Sommerville nel suo volume *Ingegneria del Software*:

Oggi la rapidità dello sviluppo e della consegna è quindi il requisito più critico per la maggior parte dei sistemi *software* aziendali. In effetti, molte aziende sono disposte ad accettare compromessi sulla qualità e sui requisiti pur di avere una rapida consegna del *software* [25].

Da qui nasce l'esigenza di adottare una nuova metodologia di sviluppo che permetta di adattare in modo **efficace** ed **efficiente** la continua variazione dei requisiti.

2.2 Metodologia agile: nascita e principi

Quanto espresso nel paragrafo precedente entra nettamente in contrasto con l'idea condivisa fra gli ingegneri del *software* negli anni '80-'90.

In quel periodo, infatti, si riteneva che il successo di un progetto fosse legato alla capacità di pianificare e controllare il processo di sviluppo. Nello specifico, Sommerville afferma che si diffuse il principio secondo cui:

si potesse avere *software* migliore attraverso l'attenta pianificazione dei progetti, la garanzia di qualità formale, l'uso di metodi di analisi e progettazione supportati da strumenti *software* e processi di sviluppo *software* controllati e rigorosi [25].

Questo tipo di approccio risultava ideale per team di **grandi dimensioni**, ma determinava un *overhead* tale che il costo della pianificazione superava quello del processo di sviluppo stesso. Il tempo che veniva impiegato nel processo decisionale sul funzionamento del sistema, riduceva drasticamente le risorse necessarie per lo sviluppo del prodotto.

Agile nasce come metodo di sviluppo *software* che si basa sui processi **iterativi** e **incrementali**. I requisiti e le soluzioni adottate si evolvono costantemente nel tempo grazie alla comprensione sempre più approfondita del problema tramite il dialogo continuo con il cliente. Questo approccio promuove lo sviluppo adattivo per mezzo della pianificazione evolutiva.

Il principio alla base di questo metodo è di dare priorità allo sviluppo del *software* (concentrando le risorse umane nella realizzazione del prodotto) piuttosto che sulla documentazione e sulla pianificazione.

Per quanto detto in precedenza, l'obiettivo di agile è di consegnare in modo **incrementale** il prodotto *software*, aggiungendo man mano nuove funzionalità e soddisfacendo sempre più requisiti.

La metodologia agile è stata originariamente definita nel 2001 nel *Manifesto per lo sviluppo agile del software* (Agile Manifesto). Il manifesto agile stabilisce il seguente ordine di rilevanza delle attività di un progetto informatico:

Gli individui e le interazioni più che i processi e gli strumenti
Il software funzionante più che la documentazione esaustiva
La collaborazione col cliente più che la negoziazione dei contratti
Rispondere al cambiamento più che seguire un piano^a

^aManifesto Agile. URL: <https://agilemanifesto.org/iso/it/manifesto.html>.

2.3 Metodo di sviluppo adottato in *Moku S.r.l*

Moku S.r.l. per la realizzazione dei propri prodotti adotta la metodologia agile descritta in precedenza. Nello specifico l'azienda utilizza il *framework Scrum*.

La gestione delle risorse viene affidata al *project manager* di ogni progetto.

2.3.0.1 Scrum

Il *framework* Scrum si basa sulla metodologia agile, di cui ne segue i principi. La scelta legata alla sua definizione è nata dalla necessità di determinare un metodo uniforme per gestire e organizzare i progetti che seguono la filosofia agile. Scrum nasce nel 2001 dall'intuizione di Ken Schwaber e Jeff Sutherland. Il termine Scrum fa riferimento alla "mischia" tipica nella disciplina del rugby (ossia una formazione di giocatori) in cui il team è spinto ad apprendere e ad adeguare lo sviluppo del *software* tramite l'esperienza. Attualmente, l'adozione del metodo agile nella gestione dei progetti è ampiamente diffusa, pertanto Scrum emerge come il *framework* prevalente.

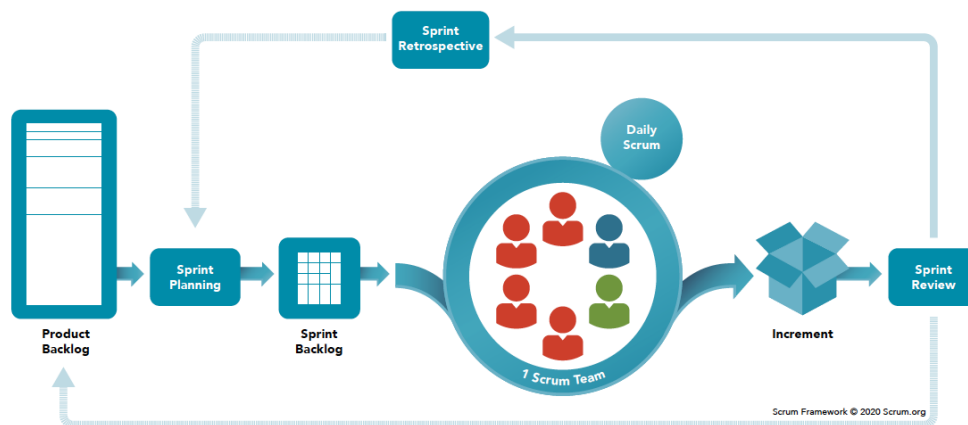


Figura 2.1: Scrum - rappresentazione grafica del ciclo di sviluppo

Fonte: Scrum - What is Scrum? URL: <https://www.scrum.org/resources/what-scrum-module>

Il ciclo di vita del *software* si basa su un ciclo di *Sprint*, come illustrato nella (Figura 2.1). Ogni ciclo di *Sprint* corrisponde ad un incremento del prodotto da consegnare al cliente. Prima dell'inizio di ogni *Sprint*, ha luogo una riunione che determina un sottoinsieme dei task presenti nel *product backlog* che dovranno essere realizzate (e che solitamente sono quelle con priorità maggiore).

Il *product backlog* raccoglie:

la lista degli elementi, come le caratteristiche del prodotto, i requisiti e il miglioramento dell'ingegnerizzazione, di cui si dovrà occupare il team di Scrum [25].

Inizialmente tutte le attività deriveranno dalla fare di analisi dei requisiti e dalle *user stories* (descritte nella sezione a seguire); queste si evolveranno nel tempo con l'aggiornamento dei requisiti e con l'avanzamento del prodotto.

Moku S.r.l adotta questo modello di sviluppo in quanto:

- favorisce il lavoro ai team, essendo un'azienda dalle risorse ridotte;

- rimane al passo con i tempi;
- è proattivo e reattivo al cambiamento, in quanto i requisiti vengono aggiornati rapidamente e spesso frequentemente;
- favorisce la comprensione dettagliata del prodotto;
- concentra i membri dei progetti sullo sviluppo, riducendo i tempi dedicati alle attività meno importanti, fra i quali la documentazione.

Il lavoro viene così suddiviso in *Sprint* che in *Moku* sono generalmente della durata di 2 settimane.

Uno sprint deve rappresentare un periodo rapido di lavoro, al termine del quale si svolge una *Sprint review*. La *Sprint review* è una riunione che avviene tra i membri del team di progetto e il cliente, con l'obiettivo di dimostrare quanto realizzato e valutare eventuali aggiunte e/o modifiche. Durante questo incontro si determina anche lo stato di avanzamento del prodotto e si pianifica lo *Sprint* successivo.

2.3.1 User stories

L'aggiornamento costante dei requisiti di un progetto devono essere gestiti correttamente. In agile solitamente non troviamo delle specifiche attività di analisi dei requisiti, ma questi vengono dedotti tramite il dialogo continuo con l'utente, il test del prodotto da parte degli utenti finali e grazie all'esperienza maturata in fase di sviluppo.

Moku sfrutta il contatto diretto con il cliente per poter raccogliergli le esigenze. Vengono poi create delle *story card* che raccolgono task della stessa *user story*, con le stime in termini di risorse necessarie e tempo richiesto per portarle a compimento.

2.4 Strumenti a supporto dei processi

Con l'avvento dell'*Open Source* e la diffusione massiva dei servizi *online*, lo sviluppo *software* si è trasformato sempre più in un lavoro fortemente collaborativo. Sono nati pertanto molti sistemi a pagamento che facilitano il lavoro di gruppo raccogliendo la *codebase* di un'azienda e favorendo il monitoraggio del processo di sviluppo [12].

Per poter svolgere in modo *efficace* ed *efficiente* il lavoro in team, sono stati selezionati degli strumenti a supporto dei vari processi aziendali. Ne segue un elenco di quelli che ho utilizzato durante il mio stage con relativa descrizione e utilità nel contesto imprenditoriale di *Moku S.r.l.*

2.4.1 Jira

Jira (Figura 2.2) è un *Issue Tracking System* di *Atlassian*, ossia un *software* che viene utilizzato per gestire e tracciare le segnalazioni di problemi, i *bug* e le richieste di funzionalità in un progetto *software*. *Jira* fornisce una piattaforma centralizzata per la gestione del ciclo di vita dei problemi, dalla segnalazione iniziale alla risoluzione finale, consentendo al team di sviluppo di collaborare e monitorare lo stato di ogni problema.¹ Fra i vantaggi del suo utilizzo troviamo:

- una maggiore trasparenza;

¹ *Jira*. URL: <https://www.atlassian.com/it/software/jira>.

- una maggiore flessibilità e possibilità di personalizzazione, rispetto ai diretti concorrenti;
- una migliore organizzazione del *work-flow*;
- una maggiore efficienza nella risoluzione dei problemi;
- sostiene lo sviluppo con il *framework Scrum* adottato dall'azienda;
- migliora la gestione dei *bug*, attraverso il tracciamento di tutta la storia;
- facilita il lavoro di verifica dello stato del prodotto al *project manager*.

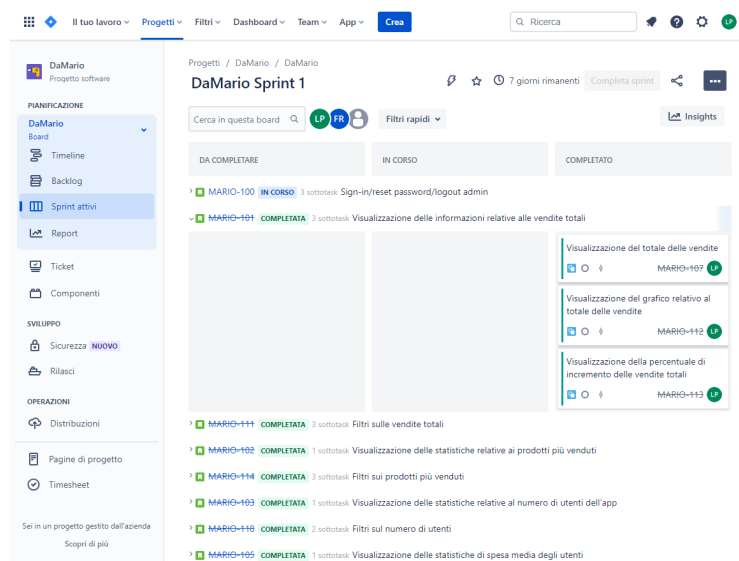


Figura 2.2: Jira - esempio di *board* raffigurante il primo *Sprint*

2.4.1.1 Tempo

Tempo (Figura 2.3) è un *plugin* integrabile su Jira utilizzato per tracciare le ore di lavoro a partire dai task.

Ha come scopo principale quello di aumentare la produttività e l'efficienza all'interno dell'azienda, creando *report* periodici.

Favorisce la stesura di piani di lavoro per ogni progetto e facilita il miglioramento delle stime effettuate dall'azienda nei progetti più recenti in base all'andamento di quelli conclusi o in corso d'opera.²

²Tempo - *plugin*. URL: <https://www.tempio.io/>.

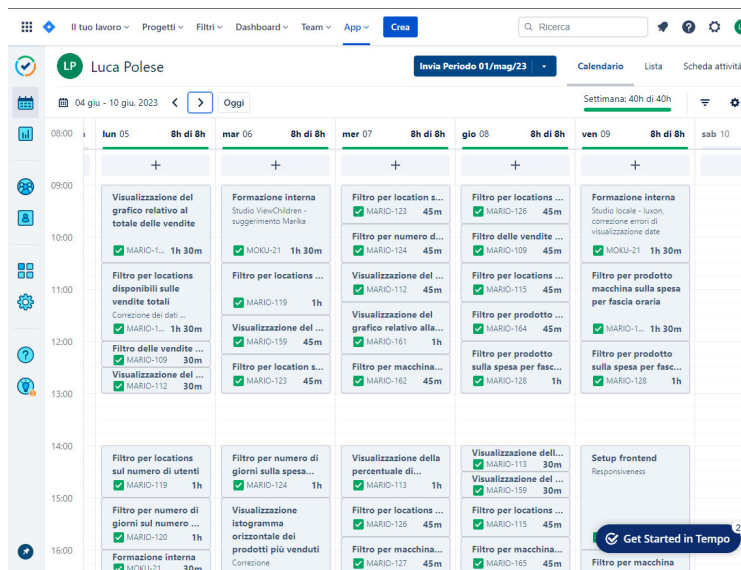


Figura 2.3: Tempo - esempio di *timesheet* settimanale

2.4.2 Bitbucket

Bitbucket (Figura 2.4) è uno strumento per il controllo della versione dell'azienda *Atlassian*. È un sistema *software* che consente di gestire le versioni dei file di un progetto, permettendo ai diversi membri del team di lavorare in modo collaborativo, sincronizzato e controllato sul codice sorgente, sulla documentazione o su qualsiasi altro tipo di file. *Bitbucket* registra tutte le modifiche effettuate sui file (mediante l'uso di *git*) e permette di ripristinare qualsiasi versione precedente, in modo da avere uno storico completo delle modifiche apportate nel tempo. Questo *software* facilita inoltre la gestione dei conflitti tra versioni diverse di uno stesso file. Ogni membro del team può lavorare su *branch* separati per sviluppare funzionalità in parallelo senza interferire tra di loro.

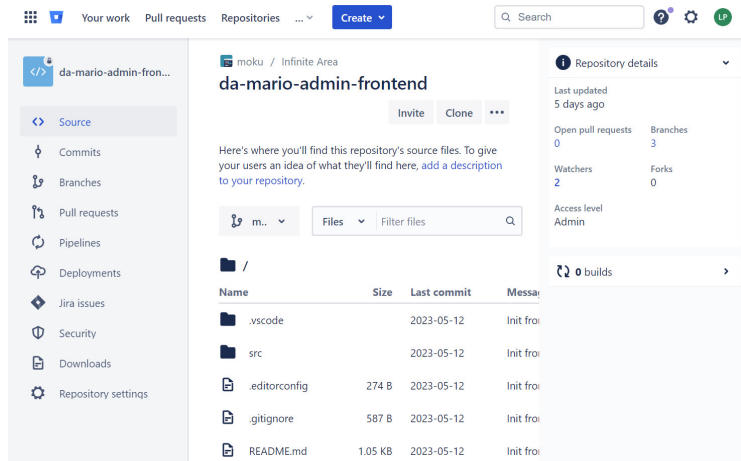


Figura 2.4: *Bitbucket* - schermata principale del *repository*

2.4.3 Slack

Slack (Figura 2.5) è un piattaforma per la messaggistica che viene utilizzata all'interno di team aziendali. La sua caratteristica principale è quella di suddividere la comunicazione intra-aziendale in più canali in un singolo servizio.

L'adozione di più canali con differenti *topic*, favorisce la collaborazione e la suddivisione per progetto. È possibile anche attivare una comunicazione fra singoli membri in forma di chat privata.

Ogni utente del servizio potrà anche aggiungere dei *plugin* per migliorare la propria produttività: un esempio può essere dato proprio dal *plugin Jira* che consente di rimanere costantemente aggiornati sullo stato dei task assegnati.

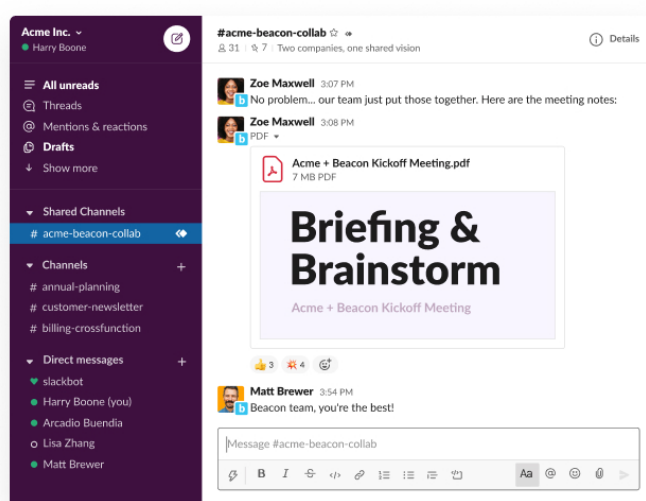


Figura 2.5: Slack - schermata principale di esempio

2.4.4 Google Calendar

Google Calendar è un sistema che consente di creare, gestire e condividere più calendari. È anche possibile importare calendari da altri servizi *online* o dal *computer* e integrare *Google Calendar* con gli altri servizi *Google* grazie all'account aziendale.

Ogni membro del team segna le proprie ferie, i propri permessi e gli eventi nel calendario aziendale condiviso così. Permette inoltre di gestire i turni di *smartworking*.

2.4.5 Passbolt

Passbolt è un *password manager Open Source* utilizzato molto da aziende, professionisti e dai privati. Ha come obiettivo quello di aumentare il livello di sicurezza delle proprie credenziali, salvandole in un contesto centralizzato.

Adotta una crittografia *end-to-end* e tutte le *password* vengono crittografate tramite la *master password* che conosce solo il proprietario delle *password* stesse o da soggetti autorizzati.³

³*Passbolt*. URL: <https://www.passbolt.com/>.

Capitolo 3

Descrizione dello stage

*Il seguente capitolo tratta le tematiche su cui si è basato il progetto definendo una separazione fra le varie parti che compongono il sistema, descrive nel dettaglio il funzionamento del *backend*, analizza i rischi e gli obiettivi del progetto affidatomi, elenca le tecnologie utilizzate ed illustra i punti relativi all'utilità dei tirocini durante il percorso accademico.*

3.1 Il contesto del progetto

Il progetto di stage prevedeva lo sviluppo di un'applicazione *web* (*frontend*) da integrare al già esistente ecosistema di un gestore di distributori automatici collocati nel territorio. Il mio inserimento nel progetto è stato supervisionato dal mio tutor aziendale, nonché CEO di *Moku S.r.l.*, Simone Pozzobon. In tutto il processo di analisi, progettazione e sviluppo del prodotto sono stato affiancato da esperti dell'ambiente di *frontend*.

Il team di **Da Mario** è composto da più figure professionali, in quanto rappresenta un progetto che gestisce in toto la distribuzione dei prodotti forniti dalle macchinette. In particolar modo l'ecosistema è suddiviso in:

- *backend* che ha le seguenti responsabilità:
 - fornisce alle applicazioni (*mobile*, *web-app* amministratore) le informazioni di cui hanno bisogno;
 - gestisce l'identificazione dei prodotti macchinetta, dei prodotti piattaforma e consente all'amministratore di associarli in un unico prodotto (ibrido);
 - mantiene le informazioni relative agli utenti quali: l'anagrafica, il borsellino, gli acquisti effettuati;
 - consente di tracciare le vendite dei prodotti finali, e permette agli utenti di effettuare ordini salvandone le informazioni necessarie (previa valutazione di errori di invalidità dei dati);
 - permette all'amministratore/agli amministratori di modificare, aggiungere o rimuovere le informazioni dei prodotti, degli utenti e delle promozioni.
- *mobile* che ha le seguenti responsabilità:
 - consentire all'utente di poter visualizzare i prodotti acquistabili;

- permettere all’utente di selezionare i suoi prodotti preferiti e di definirne una lista;
 - consentire all’utente di personalizzare le caratteristiche dei prodotti;
 - ricaricare il proprio borsellino digitale;
 - visualizzare lo storico degli acquisti.
- *frontend* che ha le seguenti responsabilità:
 - permettere la visualizzazione delle statistiche: vendite totali, prodotti più venduti, numero di utenti che hanno acquistato dalle macchinette, spesa media degli utenti, spesa media per fascia oraria;
 - filtrare i dati delle statistiche in base a parametri definiti come *range* di date, *location*, *devices*, prodotti;
 - consentire la gestione delle *location*, delle macchinette e dei prodotti in esse contenute. Nello specifico l’amministratore dovrà essere in grado di aggiungere e modificare tutti gli elementi descritti in precedenza;
 - gestire l’associazione dei prodotti macchinetta con i prodotti piattaforma e la modifica dei parametri per ogni singolo prodotto;
 - visualizzare, aggiungere e modificare i dati anagrafici degli utenti;
 - aggiungere promozioni.

Attualmente il prodotto è alla prima versione realizzata. In essa sono state integrate la gestione di macchinette del caffè dell’azienda *Cimbali* ed i distributori automatici della marca *Magex* (Figura 3.1).

Le macchinette *Magex* comunicano con il *backend* tramite *callback* su *API* di tipo *REST*.

La macchinette *Cimbali* utilizzano un protocollo proprietario su *socket*.

La comunicazione con la macchinetta viene astratta per mezzo di un dispositivo esterno che consente di tradurre il protocollo proprietario implementato dalla seconda azienda in messaggi *Message Queuing Telemetry Transport (MQTT)*.



Figura 3.1: Esempi di dispositivi *Cimbali* e *Magex*

Tutti i dati che vengono prodotti dalle macchinette delle due sopracitate aziende vengono raccolti e gestiti da *backend* e successivamente analizzati lato *frontend*. Il *backend* mantiene le informazioni dell'utente, ed effettua eventualmente dei collegamenti con *software* di gestione degli utenti (*Nexodus* per il *coworking*).

Moku S.r.l realizza ogni parte del servizio.

3.1.1 Struttura del *backend*

Il sistema ha una struttura a servizi e sfrutta la comunicazione fra due differenti *backend*.

Il *backend* che conserva le principali informazioni per scambiare i messaggi con tutti i dispositivi è nominato **Sala Macchine**. A Sala Macchine fanno capo le seguenti responsabilità:

- conosce i protocolli adottati dalle due aziende di distributori;
- gestisce dello scambio di informazioni con altri due *backend*, uno dei quali è quello di **Da Mario**.

Sala Macchine espone un *API REST* che utilizza per comunicare con tutti i dispositivi. Nello specifico il *backend* fa da server HTTP per le macchinette della casa *Magex* e comunica con i dispositivi *Cimbali* attraverso uno scambio di messaggi sfruttando il protocollo *MQTT* in una comunicazione client-client.

Questo *backend* permette di gestire ordini, *location* e *device*. La gestione degli ordini rappresenta quanto l'amministratore vuole far fare alla macchinetta.

Il *backend* **Da Mario** comunica direttamente con il *backend* **Sala Macchine** e con le applicazioni (*mobile* e *web* per l'amministrazione).

Le applicazioni comunicano con il *backend* **Da Mario** attraverso le *API GraphQL*. Le interazioni sono unidirezionali, in quanto è l'applicazione a fare le domande al server, il quale risponde con le informazioni richieste.

La comunicazione fra **Sala Macchine** e **Da Mario** invece è bidirezionale:

- Quando **Da Mario** richiede delle informazioni, **Sala Macchine** risponde;
- Se avviene un evento di interesse anche per **Da Mario**, viene scatenato un *webhook*.

Il *backend* **Da Mario** consente anche la gestione degli ordini e delle ricariche di un cliente: dall'app *mobile* arriva la richiesta dell'utente di poter comprare qualcosa. **Da Mario** invia questa informazione a *Stripe*, una piattaforma *online* per l'elaborazione dei pagamenti, che genera un *payment intent* che ha un identificativo che viene ritornato all'app. Dall'app l'utente effettua il pagamento effettivo.

Sala Macchine opera seguendo lo stesso principio: l'utente fa un ordine, crea l'ordine via *GraphQL* in **Da Mario** che lo concettualizza come qualcosa che l'utente vuole. Quest'ultimo trasforma l'ordine nel formato richiesto da **Sala Macchine**, ossia in un formato comprensibile anche dalla macchinetta, e determina la serie di operazioni che ogni macchinetta deve svolgere. Si ottiene così che l'utente riesce ad effettuare l'ordine. Segue la Figura 3.2 che rappresenta graficamente quanto espresso in precedenza.

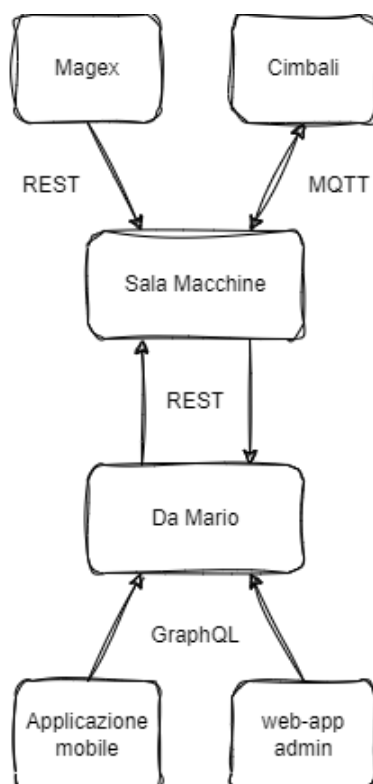


Figura 3.2: Rappresentazione dello schema del *backend*

3.2 Analisi preventiva dei rischi

Con l'avvio dello stage, la probabilità di incontrare delle problematiche può risultare maggiore in quanto, generalmente, lo studente ha un'esperienza ridotta nel campo del progetto in cui opera e per la necessità di ambientarsi ed adeguarsi agli standard aziendali.

In questi casi è necessario mitigare gli effetti di questi eventi per mezzo della stesura analisi dei rischi, durante la fase iniziale di analisi.

Per gestire al meglio le problematiche derivanti da differenti condizioni verificatesi in corso d'opera, ho redatto questa sezione di tesi in cui ho listato i possibili rischi e individuati una serie di passi da mettere in atto qualora qualcuno di essi si dovesse concretizzare.

Secondo lo standard [ISO/IEC 31000:2009](https://project.lsst.org/riskmanagement/31000-2009.pdf/)¹, il processo di gestione dei rischi è composto da 5 passi:

1. **Identificazione dei rischi:** la prima consiste nell'identificare le fonti di rischio, le aree di impatto, gli eventi (compresi i cambiamenti di circostanze), le loro cause e le loro potenziali conseguenze.

Ciò deve avvenire attraverso un'analisi delle [attività](#)^[g] con l'obiettivo di generare un elenco completo dei rischi, basato su quegli eventi che potrebbero creare,

¹ISO/IEC 31000-2009. URL: <https://project.lsst.org/riskmanagement/31000-2009.pdf/>.

migliorare, impedire, degradare, accelerare o ritardare il raggiungimento degli obiettivi.

2. **Analisi dei rischi:** questo passo comporta lo sviluppo di una comprensione del rischio. L'analisi dei rischi fornisce un contributo alla valutazione del rischio e alle decisioni sulla necessità di trattare i rischi, sulle strategie e i metodi di trattamento del rischio più appropriati.
L'analisi dei rischi può anche fornire un contributo alle decisioni da prendere quando si devono fare delle scelte e le opzioni riguardano diversi tipi e livelli di rischio.
3. **Valutazione dei rischi:** lo scopo della valutazione dei rischi è di aiutare a prendere decisioni, sulla base dei risultati dell'analisi dei rischi, in merito a quali rischi debbano essere trattati e quale sia la priorità per l'attuazione del trattamento.
La valutazione del rischio consiste nel confrontare il livello di rischio rilevato durante il processo di analisi con i criteri di rischio stabiliti quando il contesto è stato considerato.
Sulla base di questo confronto, si può valutare la necessità di un trattamento.
4. **Trattamento dei rischi:** dopo aver valutato i rischi, è importante decidere come affrontarli. Ciò può includere l'adozione di misure preventive per evitare che si verifichino, il trasferimento del rischio attraverso l'utilizzo di assicurazioni o il trattamento del rischio attraverso l'adozione di misure di mitigazione per ridurre l'impatto.
5. **Monitoraggio e revisione dei rischi:** entrambe queste attività devono far parte della pianificazione del processo di gestione del rischio e richiedono un controllo regolare (periodici o ad hoc).

È fondamentale che quanto descritto sopra sia costantemente attuato durante lo svolgimento del progetto, in quanto con l'avanzamento delle attività potrebbero:

- manifestarsi nuovi problemi che richiedono di trovare una soluzione;
- risultare inefficaci le soluzioni dei problemi già elencati.

Per determinare i rischi, è stata introdotta la seguente convenzione:

R[Tipologia][Indice]: Nome associato al rischio

Tipologia: valore alfabetico che rappresenta la categoria di rischio

- **T:** Rischi legati alle tecnologie;
- **O:** Rischi legati all'organizzazione;
- **G:** Rischi legati ai membri del gruppo di progetto.

Indice: valore numerico incrementale che determina univocamente il rischio per ogni Tipologia

RT1: Scarsa conoscenza delle tecnologie da utilizzare
--

RT1: Scarsa conoscenza delle tecnologie da utilizzare	
Descrizione	Si tratta di un rischio legato all'utilizzo di tecnologie o sistemi di cui ho poca o nessuna esperienza. Le modalità e i tempi di apprendimento possono variare a seconda della tecnologia e la documentazione a disposizione.
Conseguenze	In generale, l'inesperienza con la tecnologia può comportare errori di utilizzo, difficile astrazione dei requisiti sotto forma di codice e conseguenti rallentamenti nello sviluppo del progetto. Sottovalutarlo potrebbe rendere più difficoltoso determinare quanto tempo dev'essere dedicato all'apprendimento, generando ritardi.
Probabilità di occorrenza	Elevata.
Pericolosità	Elevata.
Precauzioni	Per garantire il successo del progetto e ridurre al minimo il rischio di problematiche di comprensione, è importante adottare alcune precauzioni. A tal fine, sarà necessario dedicare le prime settimane di stage allo studio delle tecnologie adottate e considerare di dedicare una parte delle ore delle settimane all'autoformazione per acquisire le competenze necessarie alla realizzazione delle attività assegnate. Inoltre, nel caso in cui si dovessero verificare delle difficoltà generali di comprensione, sarà necessario richiedere il supporto da parte del tutor o del team di progetto.
Piano di contingenza	La mia formazione si baserà prevalentemente sulla documentazione ufficiale di ogni tecnologia. qualora questa non fosse sufficiente verranno richiesti ulteriori materiali di approfondimento al tutor o ai membri del team FE.

Tabella 3.1: Analisi sull'inesperienza con le tecnologie da utilizzare

RT2: Perdita di file	
Descrizione	È il rischio che il codice sviluppato possa andare perso o risultare danneggiato, impossibilitando la continuazione del lavoro in atto o causando ingenti ritardi.
Conseguenze	La perdita di file può avere un impatto significativo sull' <i>economicità</i> ^[g] dello sviluppo del prodotto in quanto grava sulle sue tempistiche. Se dei file vengono persi, il lavoro svolto fino a quel momento dev'essere rifatto, causando ritardi e aumentando i costi.
Probabilità di occorrenza	Bassa.

RT2: Perdita di file	
Pericolosità	Media.
Precauzioni	È necessario gestire il progetto con un sistema di versionamento (nel caso specifico, <i>git</i> ^[gl]) ed effettuare regolarmente <i>commit</i> ^[gl] dei file modificati. Versionare il codice permetterà di tenere traccia delle modifiche apportate ad ogni documento, facilitando il lavoro di recupero in caso di perdita.
Rilevamento	È importante essere in grado di rilevare tempestivamente la perdita di file in modo da poter ripristinare quanto mancante adottando le misure di contingenza necessarie. Un modo per rilevare la perdita di file è monitorare regolarmente il progetto tramite i tool forniti da git e dall' <i>Integrated Development Environment (IDE)</i> ^[gl] . Tutti i file nuovi/aggiornati verranno caricati nei repository remoti specifici (disponibili nella pagina <i>Bitbucket</i> ^[gl] dell'organizzazione). Dovrà essere verificata periodicamente la presenza e la versione dei file, limitando così gli errori di sincronizzazione.
Piano di contingenza	In caso di perdita di file è necessario avere una fonte da cui poter ripristinare i file (copia di <i>backup/Bitbucket</i>). Questo permetterà di fare il <i>rollback</i> ^[gl] del progetto a una versione precedente o di ricreare manualmente i file persi.

Tabella 3.2: Analisi sulla perdita di file

RT3: Problemi legati all'uso di <i>software</i> di terze parti	
Descrizione	Per lo sviluppo del prodotto, l'azienda <i>Moku S.r.l.</i> ha deciso di utilizzare degli strumenti <i>software</i> di terze parti. La presenza di problemi o malfunzionamenti non dipende dal team.
Conseguenze	I problemi legati al <i>software</i> di terze parti possono essere ritardi e costi aggiuntivi per la loro risoluzione o per la sostituzione del <i>software</i> stesso. Inoltre, possono compromettere la qualità e la funzionalità del prodotto finale.
Probabilità di occorrenza	Bassa.
Pericolosità	Elevata.

RT3: Problemi legati all'uso di <i>software</i> di terze parti	
Precauzioni	Per prevenire questo rischio, è essenziale svolgere una valutazione attenta dei <i>software</i> di terze parti prima di utilizzarli nel progetto. È fondamentale assicurarsi che siano adatti e compatibili con le specifiche esigenze del progetto, evitando di scegliere <i>software</i> che potrebbero causare incompatibilità o problemi di sicurezza. Inoltre, è opportuno monitorare costantemente l'utilizzo dei <i>software</i> per identificare eventuali malfunzionamenti o vulnerabilità che potrebbero compromettere l'integrità del progetto.
Rilevamento	Per anticipare eventuali problemi legati all'utilizzo di <i>software</i> di terze parti, è importante adottare una strategia di gestione del rischio che includa una valutazione attenta di tali <i>software</i> prima di utilizzarli nel progetto. In particolare, è necessario verificare la loro affidabilità e compatibilità con le esigenze del progetto. Successivamente, per rilevare eventuali problemi, è opportuno adottare tecniche di testing del <i>software</i> durante lo sviluppo del progetto o di monitoraggio delle prestazioni durante il suo utilizzo. In questo modo, sarà possibile individuare tempestivamente eventuali malfunzionamenti o incompatibilità e adottare le misure necessarie per correggerli.
Piano di contingenza	In caso di problemi legati al <i>software</i> /servizi di terze parti, è fondamentale che i membri del team segnalino immediatamente il problema al Responsabile di progetto. In questo modo, sarà possibile valutare il grado di criticità del problema e pianificare le azioni correttive necessarie. Nel peggiore dei casi, se non è possibile risolvere il problema, sarà necessario procedere con la sostituzione del <i>software</i> /servizio con una soluzione alternativa il più possibile simile a quello utilizzato inizialmente. È importante prevedere un piano di contingenza già durante la fase di pianificazione del progetto, in modo da essere pronti ad affrontare eventuali problemi.

Tabella 3.3: Analisi dei problemi legati all'uso di *software* di terze parti

RO1: Impegni personali e/o universitari
--

RO1: Impegni personali e/o universitari	
Descrizione	Durante lo sviluppo del progetto di stage potrebbe verificarsi un evento imprevisto o una malattia che impedisce allo studente di dedicarsi completamente alle attività di stage. Questo può causare ritardi nel raggiungimento degli obiettivi e potenziali difficoltà nel monitorare il progresso del progetto.
Conseguenze	Gli eventi imprevisti o le malattie possono influire negativamente sullo svolgimento dello stage, riducendo la disponibilità oraria e causando possibili ritardi. Ciò potrebbe compromettere il raggiungimento degli obiettivi stabiliti e richiedere la revisione delle scadenze prefissate.
Probabilità di occorrenza	Media.
Pericolosità	Media.
Precauzioni	È importante considerare la possibilità di eventi imprevisti o malattie nel piano di stage. Sarà necessario pianificare un margine di tempo sufficiente per affrontare eventuali ritardi causati da tali situazioni impreviste. Inoltre, è consigliabile mantenere una comunicazione aperta e tempestiva con il responsabile dello stage per segnalare eventuali difficoltà o imprevisti che potrebbero influire sullo svolgimento del progetto.
Rilevamento	Per rilevare tempestivamente eventuali eventi imprevisti o malattie che potrebbero influire sullo svolgimento dello stage, è necessario stabilire un sistema di comunicazione regolare con il responsabile dello stage. Lo studente dovrà segnalare immediatamente qualsiasi situazione che possa interferire con la sua disponibilità o che richieda un periodo di assenza. Inoltre, è possibile utilizzare strumenti di monitoraggio del tempo impiegato per le attività al fine di rilevare eventuali ritardi o anomalie nello svolgimento delle attività.

RO1: Impegni personali e/o universitari	
Piano di contingenza	Nel caso in cui si verifichi un evento imprevisto o una malattia che impedisce allo studente di continuare le attività di stage, sarà necessario agire tempestivamente per mitigare gli effetti negativi. Sarà necessario rivalutare le scadenze e gli obiettivi del progetto, e se necessario, apportare modifiche alla pianificazione delle attività. Sarà inoltre importante coinvolgere il responsabile dello stage nella revisione della situazione e valutare le possibili soluzioni, come l'assegnazione temporanea delle attività a un altro membro del team o il riassegnamento delle risorse. L'obiettivo principale sarà quello di minimizzare l'impatto dell'evento imprevisto o della malattia sull'avanzamento del progetto.

Tabella 3.4: Analisi dei problemi legati agli impegni personali e/o universitari

RO2: Elevati costi delle attività	
Descrizione	In un progetto è importante pianificare accuratamente ogni attività. A tale scopo, si dovranno valutare il costo in termini di tempo di esecuzione e di quantità di risorse necessarie (e.g. il numero di persone coinvolte e il loro tempo di lavoro). Tuttavia, può accadere che il valore associato a ciascuna attività venga sottostimato/sovrastimato.
Conseguenze	Una sottostima dei costi delle attività può portare a ritardi nella pianificazione del progetto, poiché lo studente potrebbe non avere a disposizione il tempo necessario per completare le attività come previsto. D'altra parte, una sovrastima dei costi delle attività potrebbe portare a uno spreco di tempo, poiché lo studente potrebbe dedicare più tempo del necessario alle attività, utilizzando risorse in modo inefficiente.
Probabilità di occorrenza	Media-Alta.
Pericolosità	Alta.

RO2: Elevati costi delle attività	
Precauzioni	Per limitare il rischio legato ad aumento costi non previsti nel corso del progetto, è fondamentale che il team di sviluppo abbia fatto una pianificazione accurata delle attività e dei costi associati ad esse. Ciò implica la necessità di utilizzare strumenti di pianificazione e di controllo che permettano di monitorare il progresso del progetto e individuare eventuali problemi in anticipo. Inoltre, è importante che lo studente abbia una cognizione dettagliata delle esigenze e dei requisiti del progetto, in modo da poter pianificare le attività in modo realistico, evitando di incorrere in ostacoli che potrebbero portare ad aumento spese non previste. Infine, è fondamentale che il gruppo collabori attivamente con il Responsabile di Progetto per individuare e risolvere eventuali problemi o incongruenze nella programmazione delle attività.
Rilevamento	Il cruscotto rappresenta uno strumento fondamentale per il monitoraggio dello stato di avanzamento del progetto da parte del Responsabile di Progetto e dei membri del gruppo. Attraverso di esso è possibile verificare il progresso delle attività e individuare eventuali problematiche in tempo utile. Tuttavia, per garantire l'efficacia del cruscotto, è necessario che ogni componente del team rispetti le scadenze fissate per le attività e i tempi pianificati per ogni incremento. In caso contrario, è fondamentale comunicare tempestivamente eventuali impedimenti al Responsabile di Progetto, il quale provvederà a ridistribuire le attività.
Piano di contingenza	Per gestire eventuali cambiamenti nel progetto è necessario avere un processo di gestione del cambiamento ben strutturato. Se i cambiamenti sono di piccola entità e non influenzano significativamente lo sviluppo del progetto, è possibile che il team possa gestirli internamente, ridistribuendo le attività in modo efficiente. Tuttavia, se i cambiamenti sono di una certa entità, è necessario coinvolgere il committente per trovare un accordo sulla gestione degli stessi. In questo modo, si cerca di mantenere il progetto sulla buona strada, evitando impatti negativi sui budget e sui tempi previsti.

Tabella 3.5: Analisi dei problemi legati agli elevati costi delle attività

RG1: Rischio di conflitti interni/contrast

RG1: Rischio di conflitti interni/contrast	
Descrizione	I conflitti interni possono essere causati da diverse opinioni sulla direzione da prendere per il progetto, sulla soluzione da adottare per un problema specifico o sull'allocazione delle risorse. Possono anche essere causati da problemi di comunicazione all'interno del team (si veda Tabella 3.7).
Conseguenze	I conflitti interni possono causare ritardi nel progetto, aumentare i costi e diminuire la qualità del lavoro svolto. Possono anche creare un clima lavorativo teso e portare a una diminuzione del morale del team.
Probabilità di occorrenza	Media.
Pericolosità	Alta.
Precauzioni	Per prevenire i conflitti interni, è importante promuovere una comunicazione aperta e trasparente all'interno del team e incoraggiare il coinvolgimento di tutti i membri nella prendere decisioni importanti. Inoltre, è importante discutere tali problematiche con il Responsabile di Progetto, il quale dovrà gestire efficacemente i conflitti quando si presentano.
Rilevamento	I conflitti interni possono essere rilevati attraverso il <i>feedback</i> dei membri del team o attraverso l'osservazione delle dinamiche del gruppo.
Piano di contingenza	In caso di conflitti interni, è necessario ristabilire un ambiente di lavoro collaborativo in cui tutti i membri del team possano esprimere le loro opinioni in modo rispettoso e costruttivo. Se necessario il Responsabile di Progetto effettuerà una riassegnazione dei compiti ai vari componenti ed eventualmente cercherà di sanare gli attriti insieme al resto del team.

Tabella 3.6: Analisi dei problemi derivanti da conflitti/contrast fra membri

RG2: Problemi di comunicazione	
Descrizione	Una scarsa attitudine al lavoro di squadra, mancanza di contatto o la scarsa conoscenza degli strumenti di comunicazione, potrebbero comportare incomprensioni tra i componenti del team. Una comunicazione efficace e costante all'interno del gruppo risulta fondamentale per permettere al progetto di progredire in linea con le scadenze e con gli obiettivi fissati.

RG2: Problemi di comunicazione	
Conseguenze	Adottare un metodo di comunicazione inefficace può causare ritardi o problematiche nella realizzazione del progetto, con conseguenze negative come lo stress e la tensione all'interno del team.
Probabilità di occorrenza	Medio-alta.
Pericolosità	Alta.
Precauzioni	Per prevenire o minimizzare questo rischio, è importante che il Responsabile di Progetto monitori e promuova un adeguato livello di comunicazione attiva tra tutti i membri del team. Sarà lui a doversi assicurare che tutti i membri del team abbiano accesso agli strumenti di comunicazione e sappiano come utilizzarli. Le responsabilità e le attività dovranno essere espresse in modo chiaro e assegnate equamente ad ogni membro del team per garantire una maggiore collaborazione e coordinamento.
Rilevamento	È fondamentale evitare situazioni in cui la mancata o scarsa comunicazione possa causare ritardi nelle decisioni progettuali importanti, pertanto periodicamente il Responsabile di Progetto effettuerà dei sondaggi per raccogliere feedback e opinioni dei membri del team sulla comunicazione all'interno del gruppo. Risulterà indispensabile l'osservazione diretta del comportamento dei membri del gruppo per identificare eventuali problemi di comunicazione.
Piano di contingenza	In caso di scarsa comunicazione o di comportamenti non adeguati, sarà necessaria un'adeguata segnalazione al Responsabile del Progetto, il quale dovrà indagare sul motivo e, se necessario, organizzare una riunione nel minor tempo possibile per discutere la situazione e trovare una soluzione.

Tabella 3.7: Analisi dei problemi legati alla scarsa comunicazione fra membri del team

3.3 Obiettivi dello stage

Per specificare gli obiettivi relativi a questo progetto di stage, verrà utilizzata la seguente notazione:

- *O* indica gli obiettivi obbligatori da raggiungere, che vincolano la riuscita del progetto e che pertanto sono definiti come primari dal tutor;
- *D* indica gli obiettivi desiderabili da raggiungere al termine del periodo di stage. Non sono strettamente vincolanti o necessari, ma la loro realizzazione aggiunge valore al prodotto realizzato;
- *F* indica gli obiettivi facoltativi che migliorano il prodotto ottenuto ma che non rappresentano una necessità per renderlo competitivo.

Le sigle precedentemente indicate saranno seguite da una coppia sequenziale di numeri che indicano lo specifico requisito.

- Obbligatori:
 - O01: Acquisire competenze nello sviluppo *web* con *TypeScript*;
 - O02: Acquisire competenze nello sviluppo *web* con *Angular*;
 - O03: Sviluppare la pagina di **login** alla piattaforma di amministrazione;
 - O04: Sviluppare il **logout** dalla piattaforma di amministrazione;
 - O05: Sviluppare le rotte che aggiornano la pagina principale in cui l'utente viene reindirizzato dopo l'autenticazione;
 - O06: Sviluppare la pagina **Statistiche** della piattaforma;
 - O07: Sviluppare il cambio *password*.
- Desiderabili:
 - D01: Sviluppare la pagina **Prodotti** della piattaforma;
- Facoltativi:
 - F01: Sviluppare la pagina **Macchine** della piattaforma;
 - F02: Sviluppare la pagina **Utenti** della piattaforma;
 - F03: Sviluppare la pagina **Promozioni** della piattaforma;

3.4 Vincoli tecnologici

3.4.1 Angular

Angular è un *framework* utilizzato per lo sviluppo *frontend Open Source* basato su *TypeScript*, sviluppato dall'azienda *Google*.

L'idea alla base di questo *framework* è quella di consentire di creare applicazioni *web* scalabili e complesse, offrendo un'architettura solida e una vasta gamma di funzionalità integrate.

Angular consente di creare applicazioni *web* basate su componenti e segue il *pattern* architetturale *MVVM (Model-View-ViewModel)*.

L'architettura di base di *Angular* comprende componenti, moduli e servizi, il che lo rende adattabile ad ogni tipo di contesto, con qualsiasi livello di complessità.

Angular supporta inoltre il *binding* dei dati bidirezionale, ossia consente agli elementi dell'interfaccia utente di essere "legati" ai dati del modello: quando i dati del modello cambiano, l'interfaccia utente viene aggiornata, e viceversa. Questo semplifica la sincronizzazione dei dati tra la vista e il modello, riducendo la necessità di gestire manualmente gli aggiornamenti dell'interfaccia utente.

Nel contesto del progetto di stage, si è scelto di adottare *Angular* come principale *framework* anche perché è il più utilizzato all'interno dei progetti aziendali, ed è semplice comprenderne il funzionamento generale.

3.4.2 TypeScript

TypeScript è un linguaggio di programmazione ad oggetti particolarmente indicato per lo sviluppo *web*.

Nasce come progetto *Open Source* sviluppato dall'azienda *Microsoft*.

L'idea alla base di questo linguaggio è quella di essere un *superset* di *JavaScript*. *TypeScript* infatti aggiunge a *JavaScript* un sistema per la tipizzazione statica, che gli consente di dichiarare il tipo dei dati da rappresentare come variabili, funzioni, valori di ritorno.

L'uso di *TypeScript* favorisce un sviluppo più sicuro, semplificando la manutenzione. *TypeScript* permette infatti di identificare gli errori a tempo di compilazione anziché a tempo di esecuzione, riducendo così sensibilmente la possibilità di introduzione di *bug* all'interno del *software*.

La scelta di adottare questo linguaggio è stata determinata in seguito al vincolo di utilizzare *Angular* come *framework* principale del progetto.

Il linguaggio con cui è stato sviluppato *angular* è per l'appunto *TypeScript* di cui ne offre un supporto.

Il codice del progetto è stato pertanto scritto interamente in *TypeScript*.

3.4.3 GraphQL

La comunicazione con il *backend* avviene tramite *GraphQL*.

GraphQL è un linguaggio che viene utilizzato per l'interrogazione di *API* che è stato sviluppato da *Facebook* nel 2012 e rilasciato come *Open Source* nel 2015.

Rispetto al suo diretto concorrente *SQL*, consente di ricevere i dati dal *backend* in formato più ideale per il *frontend*, ma soprattutto consente di definire in modo dichiarativo le richieste di dati e ottenere solo le informazioni necessarie. Questo fatto consente di ridurre il sovraccarico delle richieste e migliorare le prestazioni dell'applicazione.

Il suo utilizzo è motivato in particolar modo dal fatto che il *backend* di **Da Mario** comunica con le il *backend* **Sala Macchine** utilizzando questo linguaggio.

Ho utilizzato questo linguaggio per scrivere tutte le *query* e le *mutation* necessarie al progetto per ricevere o modificare dati presenti nel *backend*.

Non ho gestito personalmente la definizione della struttura dei dati di backend in quanto al mio inserimento nel progetto, esisteva ed era ben consolidato il *backend*. Il mio compito è stato pertanto quello di comprendere come utilizzare questa tecnologia e come inviare e ricevere i dati dal server.

3.4.4 Apollo Client

Apollo Client è una libreria *JavaScript Open Source* che fornisce uno strumento per lo sviluppatore *frontend* che consente di gestire le richieste e le risposte *GraphQL* lato *client*.

Questo client è stato originariamente progettato per essere utilizzato con il *framework* *React*, ma è sono state sviluppate in seguito delle librerie che ne adattano l'utilizzo anche con il *framework* *Angular*. L'obiettivo di questo *client* è quello di semplificare l'integrazione di *GraphQL* nell'applicazione.

3.4.5 WebStorm

WebStorm è un ambiente di sviluppo integrato IDE sviluppato da *JetBrains*, specificamente progettato per lo sviluppo di applicazioni *web*. È ampiamente utilizzato dagli sviluppatori *frontend* per la scrittura di codice *HTML*, *JavaScript* e *TypeScript*.

Questo IDE è stato scelto infatti in quanto adottato dalla maggioranza degli sviluppatori *frontend* di *Moku S.r.l.* *WebStorm* infatti fornisce allo sviluppatore innumerevoli plugin che migliorano l'esperienza di sviluppo *web* supporta una vasta gamma di linguaggi orientati al *web*.

3.5 Il valore accademico dello stage

Come riporta il documento scientifico “*The Academic Value of Internships: Benefits Across Disciplines and Student Backgrounds*”, l'adozione degli stage nei processi aziendali sono visti come un'opportunità vantaggiosa sia per i datori di lavoro che per gli studenti:

- per le **aziende** gli studenti rappresentano una risorsa che snellisce il lavoro di alcuni progetti. Le aziende non sono così costrette ad assumere del personale extra da dedicare a determinati processi aziendali;
- per lo **studente** risulta un strumento fondamentale per poter dedicare una certa quantità di tempo allo studio/approfondimento delle tecnologie e per iniziare così a costruire una carriera lavorativa.

Nella letteratura scientifica relativa all'istruzione e alla formazione aziendale è stato dimostrato che un candidato che ha avuto un'esperienza di questo tipo, viene selezionato più facilmente dai reclutatori.

Ad oggi il *focus* principale delle aziende risulta essersi spostato verso tutti quegli aspetti che vanno oltre l'ambito puramente teorico ossia gli attributi non legati allo studio universitario. Studi empirici relativi a contesti educativi hanno riportato che l'esperienza di stage ha avuto un impatto positivo sui voti finali e sulla preparazione dello studente [7].

Per quanto riguarda l'azienda *Moku S.r.l.*, lo stage curricolare ha doppiamente un valore aggiunto, in quanto sfrutta la formazione dei due mesi di tirocinio per fondare le basi per l'assunzione di nuovi dipendenti da aggiungere al proprio organico aziendale.

Capitolo 4

Analisi dei requisiti

Il seguente capitolo raccoglie quanto ho analizzato durante il primo approccio con il progetto che mi è stato assegnato. In particolare sono stati individuati gli utenti finali della web-app, le entità del progetto, i casi d'uso ed i requisiti ricavati

4.1 Personas

Per poter analizzare più attentamente il problema sottoposto, si è reso necessario individuare le *personas* del sistema.

Con il termine *personas* indichiamo in modo astratto tutti gli utenti che interagiranno con la piattaforma di amministrazione.

L'identificazione dei soggetti tipo consentono al team di Da Mario di effettuare un'analisi dei requisiti più dettagliata e orientata ai bisogni dell'utente finale.

Dallo studio del profilo dell'utente, sono risultati le seguenti caratteristiche:

- Persona di età media che accede abitualmente alla piattaforma;
- Persona di età media che accede saltuariamente alla piattaforma;
- Persona di età media che vuole visualizzare solo le statistiche;
- Persona di età media che si occupa di aggiornare le informazioni dei prodotti;
- Persona di età media che vuole accedere ai dati dell'utente per gestire le sue informazioni personali;
- Persona di età media che vuole verificare lo stato dell'arte delle macchinette che gestisce;
- Persona di età media che vuole aggiornare i prodotti in vendita in base agli acquisti effettuati dagli utenti.

4.2 Entità

4.2.1 Piattaforma web

La piattaforma *web* rappresenta il servizio offerto all'utente amministratore che ne gestisce i contenuti.

4.2.2 Sezione

Ogni pagina del sito presenta una o più sezioni. Ogni sezione presenta differenti tipi di contenuti.

Una sezione può contenere:

- dati che l'utente vuole visualizzare;
- *form* che l'utente compila per aggiornare dei contenuti;
- *card* che l'utente può selezionare per richiedere la visualizzazione di contenuti personalizzati in base alle proprie esigenze.

Una sezione può inoltre rappresentare:

- l'intero contenuto della pagina;
- un componente figlio del *wrapper*.
Nello specifico si tratta del componente padre, ossia un componente padre crea un binding su una o più variabili che immagazzinano l'informazione necessaria per presentare i componenti figli con dati personalizzati in base agli *input*.

4.2.3 Amministratore

L'amministratore è l'unico utente che può avere accesso alla piattaforma.

L'utente può aver effettuato almeno un accesso o può non aver mai inserito le proprie credenziali.

E richiede l'autenticazione per consentire la gestione dei dati del sistema.

4.2.4 Location

Ogni *location* rappresenta il luogo in cui sono posizionate le macchinette gestite da *Infinite Area*. Le *location* sono caratterizzate dalle seguenti informazioni:

```
query getLocation{
  locations{
    address
    devices{
      category
      disabled
      id
      image
      kind
      location
      name
      products
      spot
    }
    id
    latitude
    longitude
    name
  }
}
```

4.2.5 Device (Macchinetta)

Il *device* (che in seguito chiameremo come macchinetta) rappresenta univocamente un dispositivo presente in una specifica *location* che viene gestita da *Infinite Area*. Ogni *device* è caratterizzato dalle seguenti informazioni:

```
query getDevices{
  devices{
    category
    disabled
    id
    image {
      url
    }
    kind
    location {
      address
      devices
      id
      latitude
      longitude
      name
    }
    name
    products{
      available
      description
      device
      disabled
      id
      image
      name
      parameters
      platformProduct
      price
      quantity
    }
    spot
  }
}
```

4.2.6 Prodotto

L'entità Prodotto definisce, un oggetto che può essere venduto all'interno nelle macchinette gestite dal sistema.

In *Da Mario* esistono due tipologie differenti di Prodotto:

- Prodotto piattaforma: ossia il prodotto generico che aggrega alcuni prodotti specifici presenti all'interno di una o più macchinette. Definisce le informazioni di base di ogni tipologia di prodotto.

Per comprendere meglio il concetto, prendiamo come esempio un **caffè espresso**. Il caffè espresso può essere presente in ogni macchinetta, ed ognuna di esse lo identifica in un modo differente. Il prodotto piattaforma è quel prodotto che

definisce le caratteristiche di base che tutti i caffè espresso di ogni macchinetta (come ad esempio il prezzo impostato a 0.5€);

- **Prodotti macchinetta:** ossia il prodotto specifico presente all'interno di una singola macchinetta. Riprendendo l'esempio precedente, è possibile che ogni macchinetta gestita abbia il proprio prodotto "caffè espresso" che può differenziarsi rispetto agli altri per vari parametri, come ad esempio il costo, l'aroma, la quantità di acqua erogata per singolo caffè, etc.

4.3 Casi d'uso

Una volta individuate le entità che devono comporre il prodotto *software* da realizzare ho proseguito con la stesura dei casi d'uso legati all'attore **utente amministratore**. I vari casi d'uso sono stati raggruppati per ambito, così da semplificarne la lettura. La numerazione non verrà influenzata e ne verrà mantenuta la sequenzialità.

Per esprimere con chiarezza quanto presente in questa sezione, ogni caso d'uso è definito con una serie di informazioni aggiuntive per garantirne una comprensione completa e precisa. Tra queste, troviamo:

- **Attori Principali:** le entità esterne al sistema che interagiscono con esso per raggiungere un obiettivo specifico;
- **Precondizioni:** le condizioni del sistema che devono essere soddisfatte prima che il caso d'uso possa essere eseguito. Gli *input* sono opzionali, ma utili per specificare con precisione le informazioni che l'attore porta all'interno del sistema;
- **Postcondizioni:** le condizioni del sistema che si verificano al termine dell'esecuzione del caso d'uso. Gli *output* sono opzionali, ma utili per descrivere i valori o gli oggetti che il sistema restituisce come conseguenza dell'esecuzione del caso d'uso.
- **Descrizione:** una breve descrizione che delinea il caso d'uso in modo chiaro e completo e fornisce un elenco numerato che rappresenta il flusso degli eventi principali che si verificano durante l'esecuzione del caso d'uso;
- **Estensioni:** (opzionale) impiegate per modellare scenari alternativi. Al verificarsi di una determinata condizione, il caso d'uso ad essa collegata viene interrotto.

4.3.1 Casi d'Uso - Riconoscimento dell'utente amministratore

UC1: Autenticazione dell'utente

Attori Principali: Utente non autenticato.

Precondizioni: L'utente amministratore è registrato nel sistema.

Postcondizioni: L'utente viene riconosciuto e viene riportato alla schermata principale della piattaforma di amministrazione.

Descrizione:

- L'utente visualizza la piattaforma di *login* in cui inserisce:

- l'*email* nell'apposito *form*;
 - la *password* nell'apposito *form*.
- L'utente clicca il bottone di *login* e si accredita con successo.

Scenario Alternativo: L'utente non viene riconosciuto in quanto si è verificato un errore in fase di autenticazione. Viene così visualizzato un messaggio di errore (UC2)

UC2: Visualizzazione dell'errore di autenticazione

Attori Principali: Utente non autenticato

Precondizioni: L'utente ha inserito le credenziali sulla vista di login, ma si è verificata una condizione che ha dato origine ad un errore

Postcondizioni: L'utente viene avvisato dell'avvenuto errore tramite un messaggio che ne descrive la situazione

Descrizione: Può essersi verificata una delle seguenti condizioni:

- Le credenziali inserite non sono corrette;
- Non esiste alcun utente amministratore con quelle credenziali;
- Esiste un problema lato *backend*;
- L'utente non è connesso ad Internet;

UC3: Logout

Attori Principali: Utente amministratore autenticato

Precondizioni: L'utente amministratore ha effettuato il *login* con successo e si trova sulla pagina principale della piattaforma

Postcondizioni: L'utente viene disconnesso dalla piattaforma di amministrazione e riportato alla pagina di *login*

Descrizione: L'utente viene disconnesso con successo

UC4: Recupero *password*

Attori Principali: Utente non autenticato

Precondizioni:

- L'utente che richiede di recuperare la *password* ha inserito il proprio indirizzo *email*;
- L'utente amministratore è presente all'interno degli utenti del *database* degli amministratori.

Postcondizioni: L'utente amministratore riceve una mail contenente il *link* per il recupero della *password*

Descrizione:

- L'utente amministratore ha dimenticato la *password*;
- L'utente amministratore riceve una mail per il recupero della *password*;
- L'utente amministratore imposta la nuova *password*.

4.3.2 Casi d'Uso - Statistiche "Vendite totali"

UC5: Visualizzazione statistiche sulle vendite totali

Attori Principali: Utente amministratore autenticato

Precondizioni: L'utente amministratore ha effettuato il *login* con successo al pannello di amministrazione.

Postcondizioni: L'utente amministratore visualizza le informazioni relative alle vendite totali.

Descrizione: L'utente amministratore desidera visualizzare le informazioni relative alle vendite totali nel sistema:

- L'utente amministratore accede al pannello di amministrazione;
- Il sistema calcola e mostra le informazioni relative alle vendite totali;
- L'utente amministratore naviga alla sezione delle vendite;
- L'utente amministratore visualizza le informazioni relative alle vendite totali.

UC6: Visualizzazione statistiche sulle vendite totali filtrate per *location*

Attori Principali: Utente amministratore autenticato

Precondizioni:

- L'utente amministratore ha effettuato il *login* con successo al pannello di amministrazione;
- L'utente amministratore ha visualizzato le statistiche sulle vendite totali;
- L'utente amministratore ha selezionato una *location*.

Postcondizioni: L'utente amministratore visualizza le informazioni relative alle vendite totali filtrate in base alla *location* selezionata.

Descrizione: L'utente amministratore desidera visualizzare solo le vendite totali di una determinata *location* del sistema:

- L'utente amministratore accede al pannello di amministrazione;
- Il sistema calcola e mostra le informazioni relative alle vendite totali;
- L'utente amministratore naviga alla sezione delle vendite;

- L'utente amministratore visualizza le informazioni relative alle vendite totali;
- L'utente amministratore seleziona l'opzione per filtrare le vendite per *location*;
- Il sistema applica il filtro e mostra solo le vendite appartenenti alla *location* scelta;
- L'utente amministratore visualizza la lista delle vendite filtrate.

UC7: Visualizzazione statistiche sulle vendite totali filtrate per macchinetta

Attori Principali: Utente amministratore autenticato

Precondizioni:

- L'utente amministratore ha effettuato il *login* con successo al pannello di amministrazione;
- L'utente amministratore ha visualizzato le statistiche sulle vendite totali;
- L'utente amministratore ha selezionato una *location*;
- L'utente amministratore ha selezionato una macchinetta.

Postcondizioni: L'utente amministratore visualizza le informazioni relative alle vendite totali filtrate in base alla macchinetta selezionata.

Descrizione: L'utente amministratore desidera visualizzare solo le vendite totali di una determinata macchinetta di una determinata *location* del sistema:

- L'utente amministratore accede al pannello di amministrazione;
- Il sistema calcola e mostra le informazioni relative alle vendite totali;
- L'utente amministratore naviga alla sezione delle vendite;
- L'utente amministratore visualizza le informazioni relative alle vendite totali;
- L'utente amministratore seleziona l'opzione per filtrare le vendite per *location*;
- L'utente amministratore seleziona l'opzione per filtrare le vendite per macchinetta;
- Il sistema applica il filtro e mostra solo le vendite appartenenti alla macchinetta scelta;
- L'utente amministratore visualizza la lista delle vendite filtrate.

UC8: Visualizzazione statistiche sulle vendite totali filtrate per numero di giorni

Attori Principali: Utente amministratore autenticato

Precondizioni:

- L'utente amministratore ha effettuato il *login* con successo al pannello di amministrazione;
- L'utente amministratore ha visualizzato le statistiche sulle vendite totali.

Postcondizioni: L'utente amministratore visualizza le informazioni relative alle vendite totali filtrate in base al *range* di giorni selezionato.

Descrizione: L'utente amministratore desidera visualizzare solo le vendite totali di una determinato *range* di giorni:

- L'utente amministratore accede al pannello di amministrazione;
- Il sistema calcola e mostra le informazioni relative alle vendite totali;
- L'utente amministratore naviga alla sezione delle vendite;
- L'utente amministratore visualizza le informazioni relative alle vendite totali;
- L'utente amministratore seleziona l'opzione per filtrare le vendite per *range* di giorni;
- Il sistema applica il filtro e mostra solo le vendite appartenenti al *range* scelto;
- L'utente amministratore visualizza la lista delle vendite filtrate.

4.3.3 Casi d'Uso - Statistiche "Prodotti più venduti"

UC9: Visualizzazione statistiche sui prodotti più venduti

Attori Principali: Utente amministratore autenticato

Precondizioni: L'utente amministratore ha effettuato il *login* con successo al pannello di amministrazione.

Postcondizioni: L'utente amministratore visualizza le informazioni relative ai prodotti più venduti.

Descrizione: L'utente amministratore desidera visualizzare le informazioni relative ai prodotti più venduti nel sistema:

- L'utente amministratore accede al pannello di amministrazione;
- Il sistema calcola e mostra le informazioni relative ai prodotti più venduti;
- L'utente amministratore naviga alla sezione dei prodotti più venduti;
- L'utente amministratore visualizza le informazioni relative ai prodotti più venduti.

UC10: Visualizzazione statistiche sui prodotti più venduti filtrate per *location*

Attori Principali: Utente amministratore autenticato

Precondizioni:

- L'utente amministratore ha effettuato il *login* con successo al pannello di amministrazione;
- L'utente amministratore ha visualizzato le statistiche sui prodotti più venduti;
- L'utente amministratore ha selezionato una *location*.

Postcondizioni: L'utente amministratore visualizza le informazioni relative ai prodotti più venduti filtrate in base alla *location* selezionata.

Descrizione: L'utente amministratore desidera visualizzare solo i prodotti più venduti di una determinata *location* del sistema:

- L'utente amministratore accede al pannello di amministrazione;
- Il sistema calcola e mostra le informazioni relative ai prodotti più venduti;
- L'utente amministratore naviga alla sezione dei prodotti più venduti;
- L'utente amministratore visualizza le informazioni relative ai prodotti più venduti;
- L'utente amministratore seleziona l'opzione per filtrare i prodotti più venduti per *location*;
- Il sistema applica il filtro e mostra solo i prodotti appartenenti alla *location* scelta;
- L'utente amministratore visualizza la lista dei prodotti più venduti filtrati.

UC11: Visualizzazione statistiche sui prodotti più venduti filtrate per macchinetta

Attori Principali: Utente amministratore autenticato

Precondizioni:

- L'utente amministratore ha effettuato il *login* con successo al pannello di amministrazione;
- L'utente amministratore ha visualizzato le statistiche sulle prodotti più venduti;
- L'utente amministratore ha selezionato una *location*;
- L'utente amministratore ha selezionato una macchinetta.

Postcondizioni: L'utente amministratore visualizza le informazioni relative ai prodotti più venduti filtrate in base alla macchinetta selezionata.

Descrizione: L'utente amministratore desidera visualizzare solo i prodotti più venduti di una determinata macchinetta di una determinata *location* del sistema:

- L'utente amministratore accede al pannello di amministrazione;
- Il sistema calcola e mostra le informazioni relative ai prodotti più venduti;

- L'utente amministratore naviga alla sezione dei prodotti più venduti;
- L'utente amministratore visualizza le informazioni relative ai prodotti più venduti;
- L'utente amministratore seleziona l'opzione per filtrare i prodotti più venduti per *location*;
- L'utente amministratore seleziona l'opzione per filtrare i prodotti per macchinetta;
- Il sistema applica il filtro e mostra solo i prodotti più venduti appartenenti alla macchinetta scelta;
- L'utente amministratore visualizza la lista dei prodotti più venduti filtrati.

UC12: Visualizzazione statistiche sui prodotti più venduti filtrate per numero di giorni

Attori Principali: Utente amministratore autenticato

Precondizioni:

- L'utente amministratore ha effettuato il *login* con successo al pannello di amministrazione;
- L'utente amministratore ha visualizzato le statistiche sulle prodotti più venduti.

Postcondizioni: L'utente amministratore visualizza le informazioni relative ai prodotti più venduti filtrate in base al *range* di giorni selezionato.

Descrizione: L'utente amministratore desidera visualizzare solo i prodotti più venduti di un determinato *range* di giorni:

- L'utente amministratore accede al pannello di amministrazione;
- Il sistema calcola e mostra le informazioni relative ai prodotti più venduti;
- L'utente amministratore naviga alla sezione delle vendite;
- L'utente amministratore visualizza la lista delle vendite disponibili e le informazioni relative alle prodotti più venduti;
- L'utente amministratore seleziona l'opzione per filtrare le vendite per *range* di giorni;
- Il sistema applica il filtro e mostra solo le vendite appartenenti al *range* scelto;
- L'utente amministratore visualizza la lista delle vendite filtrate.

4.3.4 Casi d'Uso - Statistiche "Numero di utenti"

UC13: Visualizzazione statistiche sul numero di utenti che hanno effettuato acquisti

Attori Principali: Utente amministratore autenticato

Precondizioni: L'utente amministratore ha effettuato il *login* con successo al pannello di amministrazione.

Postcondizioni: L'utente amministratore visualizza le informazioni relative numero di utenti che hanno effettuato acquisti.

Descrizione: L'utente amministratore desidera visualizzare le informazioni relative al numero di utenti che hanno effettuato acquisti nel sistema:

- L'utente amministratore accede al pannello di amministrazione;
- Il sistema calcola e mostra le informazioni relative al numero di utenti che hanno effettuato acquisti;
- L'utente amministratore naviga alla sezione del numero di utenti che hanno effettuato acquisti;
- L'utente amministratore visualizza le informazioni relative al numero di utenti che hanno effettuato acquisti.

UC14: Visualizzazione statistiche sul numero di utenti che hanno effettuato acquisti filtrate per *location*

Attori Principali: Utente amministratore autenticato

Precondizioni:

- L'utente amministratore ha effettuato il *login* con successo al pannello di amministrazione;
- L'utente amministratore ha visualizzato le statistiche sul numero di utenti che hanno effettuato acquisti;
- L'utente amministratore ha selezionato una *location*.

Postcondizioni: L'utente amministratore visualizza le informazioni relative al numero di utenti che hanno effettuato acquisti filtrate in base alla *location* selezionata.

Descrizione: L'utente amministratore desidera visualizzare solo il numero di utenti che hanno effettuato acquisti di una determinata *location* del sistema:

- L'utente amministratore accede al pannello di amministrazione;
- Il sistema calcola e mostra le informazioni relative al numero di utenti che hanno effettuato acquisti;
- L'utente amministratore naviga alla sezione del numero di utenti che hanno effettuato acquisti;

- L'utente amministratore visualizza le informazioni relative ai prodotti più venduti;
- L'utente amministratore seleziona l'opzione per filtrare il numero di utenti che hanno effettuato acquisti per *location*;
- Il sistema applica il filtro e mostra solo il numero di utenti che hanno effettuato acquisti appartenenti alla *location* scelta;
- L'utente amministratore visualizza la lista del numero di utenti che hanno effettuato acquisti filtrati.

UC15: Visualizzazione statistiche sul numero di utenti che hanno effettuato acquisti filtrate per numero di giorni

Attori Principali: Utente amministratore autenticato

Precondizioni:

- L'utente amministratore ha effettuato il *login* con successo al pannello di amministrazione;
- L'utente amministratore ha visualizzato le statistiche sul numero di utenti.

Postcondizioni: L'utente amministratore visualizza le informazioni relative al numero di utenti che hanno effettuato acquisti filtrate in base al *range* di giorni selezionato.

Descrizione: L'utente amministratore desidera visualizzare solo il numero di utenti che hanno effettuato acquisti in un determinato *range* di giorni:

- L'utente amministratore accede al pannello di amministrazione;
- Il sistema calcola e mostra le informazioni relative al numero di utenti che hanno effettuato acquisti;
- L'utente amministratore naviga alla sezione del numero di utenti che hanno effettuato acquisti;
- L'utente amministratore visualizza la lista delle informazioni relative al numero di utenti che hanno effettuato acquisti;
- L'utente amministratore seleziona l'opzione per filtrare il numero di utenti che hanno effettuato acquisti *range* di giorni;
- Il sistema applica il filtro e mostra solo numero di utenti che hanno effettuato acquisti appartenenti al *range* scelto;
- L'utente amministratore visualizza la lista filtrata del numero di utenti che hanno effettuato acquisti.

4.3.5 Casi d'Uso - Statistiche "Spesa media utenti"

UC16: Visualizzazione statistiche sulla spesa media degli utenti

Attori Principali: Utente amministratore autenticato

Precondizioni: L'utente amministratore ha effettuato il *login* con successo al pannello di amministrazione.

Postcondizioni: L'utente amministratore visualizza le informazioni relative alla spesa media degli utenti.

Descrizione: L'utente amministratore desidera visualizzare le informazioni relative alla spesa media degli utenti nel sistema:

- L'utente amministratore accede al pannello di amministrazione;
- Il sistema calcola e mostra le informazioni relative alla spesa media degli utenti;
- L'utente amministratore naviga alla sezione della spesa media degli utenti;
- L'utente amministratore visualizza le informazioni relative alla spesa media degli utenti.

UC17: Visualizzazione statistiche sulla spesa media degli utenti filtrate per *location*

Attori Principali: Utente amministratore autenticato

Precondizioni:

- L'utente amministratore ha effettuato il *login* con successo al pannello di amministrazione;
- L'utente amministratore ha visualizzato le statistiche sulla spesa media degli utenti;
- L'utente amministratore ha selezionato una *location*.

Postcondizioni: L'utente amministratore visualizza le informazioni relative alla spesa media degli utenti filtrate in base alla *location* selezionata.

Descrizione: L'utente amministratore desidera visualizzare solo la spesa media degli utenti di una determinata *location* del sistema:

- L'utente amministratore accede al pannello di amministrazione;
- Il sistema calcola e mostra le informazioni relative alla spesa media degli utenti;
- L'utente amministratore naviga alla sezione della spesa media degli utenti;
- L'utente amministratore visualizza le informazioni relative alla spesa media degli utenti;
- L'utente amministratore seleziona l'opzione per filtrare la spesa media degli utenti per *location*;

- Il sistema applica il filtro e mostra solo i prodotti appartenenti alla *location* scelta;
- L'utente amministratore visualizza la lista della spesa media degli utenti filtrati.

UC18: Visualizzazione statistiche sulla spesa media degli utenti filtrate per macchinetta

Attori Principali: Utente amministratore autenticato

Precondizioni:

- L'utente amministratore ha effettuato il *login* con successo al pannello di amministrazione;
- L'utente amministratore ha visualizzato le statistiche sulla spesa media degli utenti;
- L'utente amministratore ha selezionato una *location*;
- L'utente amministratore ha selezionato una macchinetta.

Postcondizioni: L'utente amministratore visualizza le informazioni relative alla spesa media degli utenti filtrate in base alla macchinetta selezionata.

Descrizione: L'utente amministratore desidera visualizzare solo la spesa media degli utenti di una determinata macchinetta di una determinata *location* del sistema:

- L'utente amministratore accede al pannello di amministrazione;
- Il sistema calcola e mostra le informazioni relative alla spesa media degli utenti;
- L'utente amministratore naviga alla sezione della spesa media degli utenti;
- L'utente amministratore visualizza le informazioni relative alla spesa media degli utenti;
- L'utente amministratore seleziona l'opzione per filtrare la spesa media degli utenti per *location*;
- L'utente amministratore seleziona l'opzione per filtrare la prodotti per macchinetta;
- Il sistema applica il filtro e mostra solo la spesa media degli utenti appartenenti alla macchinetta scelta;
- L'utente amministratore visualizza la lista della spesa media degli utenti filtrati.

UC19: Visualizzazione statistiche sulla spesa media degli utenti filtrate per numero di giorni

Attori Principali: Utente amministratore autenticato

Precondizioni:

- L'utente amministratore ha effettuato il *login* con successo al pannello di amministrazione;
- L'utente amministratore ha visualizzato le statistiche sulla spesa media degli utenti.

Postcondizioni: L'utente amministratore visualizza le informazioni relative alla spesa media degli utenti filtrate in base al *range* di giorni selezionato.

Descrizione: L'utente amministratore desidera visualizzare solo la spesa media degli utenti di un determinato *range* di giorni:

- L'utente amministratore accede al pannello di amministrazione;
- Il sistema calcola e mostra le informazioni relative alla spesa media degli utenti;
- L'utente amministratore naviga alla sezione delle vendite;
- L'utente amministratore visualizza la lista delle vendite disponibili e le informazioni relative alla spesa media degli utenti;
- L'utente amministratore seleziona l'opzione per filtrare le vendite per *range* di giorni;
- Il sistema applica il filtro e mostra solo le vendite appartenenti al *range* scelto;
- L'utente amministratore visualizza la lista delle vendite filtrate.

4.3.6 Casi d'Uso - Statistiche "Spesa per fascia oraria"

UC20: Visualizzazione statistiche sulla spesa per fascia oraria

Attori Principali: Utente amministratore autenticato

Precondizioni: L'utente amministratore ha effettuato il *login* con successo al pannello di amministrazione.

Postcondizioni: L'utente amministratore visualizza le informazioni relative alla spesa per fascia oraria.

Descrizione: L'utente amministratore desidera visualizzare le informazioni relative alla spesa per fascia oraria nel sistema:

- L'utente amministratore accede al pannello di amministrazione;
- Il sistema calcola e mostra le informazioni relative alla spesa per fascia oraria;
- L'utente amministratore naviga alla sezione della spesa per fascia oraria;
- L'utente amministratore visualizza le informazioni relative alla spesa per fascia oraria.

UC21: Visualizzazione statistiche sulla spesa per fascia oraria filtrate per *location*

Attori Principali: Utente amministratore autenticato

Precondizioni:

- L'utente amministratore ha effettuato il *login* con successo al pannello di amministrazione;
- L'utente amministratore ha visualizzato le statistiche sulla spesa per fascia oraria;
- L'utente amministratore ha selezionato una *location*.

Postcondizioni: L'utente amministratore visualizza le informazioni relative alla spesa per fascia oraria filtrate in base alla *location* selezionata.

Descrizione: L'utente amministratore desidera visualizzare solo la spesa per fascia oraria di una determinata *location* del sistema:

- L'utente amministratore accede al pannello di amministrazione;
- Il sistema calcola e mostra le informazioni relative alla spesa per fascia oraria;
- L'utente amministratore naviga alla sezione della spesa per fascia oraria;
- L'utente amministratore visualizza le informazioni relative alla spesa per fascia oraria;
- L'utente amministratore seleziona l'opzione per filtrare la spesa per fascia oraria per *location*;
- Il sistema applica il filtro e mostra solo i prodotti appartenenti alla *location* scelta;
- L'utente amministratore visualizza la lista filtrata della spesa per fascia oraria.

UC22: Visualizzazione statistiche sulla spesa per fascia oraria filtrate per macchinetta

Attori Principali: Utente amministratore autenticato

Precondizioni:

- L'utente amministratore ha effettuato il *login* con successo al pannello di amministrazione;
- L'utente amministratore ha visualizzato le statistiche sulla spesa per fascia oraria;
- L'utente amministratore ha selezionato una *location*;
- L'utente amministratore ha selezionato una macchinetta.

Postcondizioni: L'utente amministratore visualizza le informazioni relative alla spesa per fascia oraria filtrate in base alla macchinetta selezionata.

Descrizione: L'utente amministratore desidera visualizzare solo la spesa per fascia oraria di una determinata macchinetta di una determinata *location* del sistema:

- L'utente amministratore accede al pannello di amministrazione;
- Il sistema calcola e mostra le informazioni relative alla spesa per fascia oraria;
- L'utente amministratore naviga alla sezione della spesa per fascia oraria;
- L'utente amministratore visualizza le informazioni relative alla spesa per fascia oraria;
- L'utente amministratore seleziona l'opzione per filtrare la spesa per fascia oraria per *location*;
- L'utente amministratore seleziona l'opzione per filtrare la prodotti per macchinetta;
- Il sistema applica il filtro e mostra solo la spesa per fascia oraria appartenenti alla macchinetta scelta;
- L'utente amministratore visualizza la lista della spesa per fascia oraria filtrata.

UC23: Visualizzazione statistiche sulla spesa per fascia oraria filtrate per prodotto

Attori Principali: Utente amministratore autenticato

Precondizioni:

- L'utente amministratore ha effettuato il *login* con successo al pannello di amministrazione;
- L'utente amministratore ha visualizzato le statistiche sulla spesa per fascia oraria;
- L'utente amministratore ha selezionato una *location*;
- L'utente amministratore ha selezionato una macchinetta;
- L'utente amministratore ha selezionato un prodotto.

Postcondizioni: L'utente amministratore visualizza le informazioni relative alla spesa per fascia oraria filtrate in base al prodotto selezionata.

Descrizione: L'utente amministratore desidera visualizzare solo la spesa per fascia oraria di un determinato prodotto di una determinata macchinetta di una determinata *location* del sistema:

- L'utente amministratore accede al pannello di amministrazione;
- Il sistema calcola e mostra le informazioni relative alla spesa per fascia oraria;
- L'utente amministratore naviga alla sezione della spesa per fascia oraria;
- L'utente amministratore visualizza le informazioni relative alla spesa per fascia oraria;
- L'utente amministratore seleziona l'opzione per filtrare la spesa per fascia oraria per *location*;

- L'utente amministratore seleziona l'opzione per filtrare la spesa per fascia oraria per macchinetta;
- L'utente amministratore seleziona l'opzione per filtrare la prodotti per prodotto;
- Il sistema applica il filtro e mostra solo la spesa per fascia oraria appartenenti al prodotto scelto;
- L'utente amministratore visualizza la lista della spesa per fascia oraria filtrata.

UC24: Visualizzazione statistiche sulla spesa per fascia oraria filtrate per numero di giorni

Attori Principali: Utente amministratore autenticato

Precondizioni:

- L'utente amministratore ha effettuato il *login* con successo al pannello di amministrazione;
- L'utente amministratore ha visualizzato le statistiche sulla spesa per fascia oraria.

Postcondizioni: L'utente amministratore visualizza le informazioni relative alla spesa per fascia oraria filtrate in base al *range* di giorni selezionato.

Descrizione: L'utente amministratore desidera visualizzare solo la spesa per fascia oraria di un determinato *range* di giorni:

- L'utente amministratore accede al pannello di amministrazione;
- Il sistema calcola e mostra le informazioni relative alla spesa per fascia oraria;
- L'utente amministratore naviga alla sezione delle vendite;
- L'utente amministratore visualizza la lista delle vendite disponibili e le informazioni relative alla spesa per fascia oraria;
- L'utente amministratore seleziona l'opzione per filtrare le vendite per *range* di giorni;
- Il sistema applica il filtro e mostra solo le vendite appartenenti al *range* scelto;
- L'utente amministratore visualizza la lista delle vendite filtrate.

4.3.7 Casi d'Uso - Macchine

UC25: Visualizzazione della lista delle *location*

Attori Principali: Utente amministratore autenticato

Precondizioni: L'utente amministratore ha effettuato il *login* con successo al pannello di amministrazione.

Postcondizioni: L'utente amministratore visualizza la lista completa delle *location*

disponibili.

Descrizione: L'utente amministratore desidera visualizzare l'intera lista delle *location* disponibili nel sistema:

- L'utente amministratore accede al pannello di amministrazione;
- Il sistema calcola e mostra le informazioni relative alle *location* disponibili;
- L'utente amministratore visualizza le informazioni relative alle *location*.

UC26: Creazione di una nuova *location*

Attori Principali: Utente amministratore autenticato

Precondizioni: L'utente amministratore ha effettuato il *login* con successo al pannello di amministrazione.

Postcondizioni: L'utente amministratore ha creato una nuova *location* che è stata aggiunta al sistema.

Descrizione: L'utente amministratore vuole aggiungere una nuova *location* nel sistema:

- L'utente amministratore accede al pannello di amministrazione.
- L'utente amministratore clicca per l'aggiunta della nuova *location*.
- Il sistema mostra un *form* da compilare per la creazione della *location*.
- L'utente amministratore inserisce il nome, l'indirizzo, la longitudine e la latitudine della nuova *location*.
- L'utente amministratore conferma la creazione della *location*;
- Il sistema valida i dati inseriti e crea una nuova *location* nel sistema;
- Il sistema aggiorna la pagina delle *location*;
- L'utente amministratore visualizza i dettagli della nuova *location* creata.

UC27: Modifica di una *location*

Attori Principali: Utente amministratore autenticato

Precondizioni:

- L'utente amministratore ha effettuato il *login* con successo al pannello di amministrazione;
- Esiste almeno una *location*.

Postcondizioni: Le modifiche alla *location* sono state salvate correttamente nel sistema

Descrizione: L'utente amministratore vuole mod una nuova *location* nel sistema:

- L'utente amministratore accede al pannello di amministrazione.
- L'utente amministratore visualizza la lista delle *location* disponibili.
- L'utente amministratore seleziona la *location* da modificare.
- Il sistema mostra un *form* con i dati da modificare della *location*.
- L'utente amministratore apporta le modifiche desiderate ai campi della *location*.
- L'utente amministratore salva le modifiche;
- Il sistema valida i dati inseriti e aggiorna la specifica *location* nel sistema;
- Il sistema aggiorna la pagina delle *location*;
- L'utente amministratore visualizza i dettagli aggiornati della *location*.

UC28: Visualizzazione delle macchinette di una *location*

Attori Principali: Utente amministratore autenticato

Precondizioni: L'utente amministratore ha effettuato il *login* con successo al pannello di amministrazione.

Postcondizioni: L'utente amministratore visualizza la lista completa delle macchinette associate ad una *location*.

Descrizione: L'utente amministratore desidera visualizzare la lista delle macchinette presenti nel sistema e suddivise per *location*:

- L'utente amministratore accede al pannello di amministrazione;
- Il sistema calcola e mostra i dettagli relativi alle *location* disponibili;
- L'utente amministratore scorre fino a raggiungere la *location* di suo interesse;
- L'utente amministratore visualizza la lista delle macchinette associate alla *location*.

UC29: Creazione di una nuova macchinetta in una *location*

Attori Principali: Utente amministratore autenticato

Precondizioni: L'utente amministratore ha effettuato il *login* con successo al pannello di amministrazione.

Postcondizioni: L'utente amministratore ha creato una nuova macchinetta che è stata aggiunta al sistema.

Descrizione: L'utente amministratore vuole aggiungere una nuova macchinetta nel sistema:

- L'utente amministratore accede al pannello di amministrazione.
- L'utente amministratore scorre fino alla *location* di interesse in cui aggiungere la nuova macchinetta;

- L'utente amministratore clicca per l'aggiunta della nuova macchinetta.
- Il sistema mostra un *form* da compilare per la creazione della macchinetta.
- L'utente amministratore inserisce della nuova macchinetta.
- L'utente amministratore conferma la creazione della macchinetta;
- Il sistema valida i dati inseriti e crea una nuova macchinetta nel sistema;
- Il sistema aggiorna la pagina delle *location*;
- L'utente amministratore visualizza i dettagli della nuova macchinetta creata.

UC30: Modifica di una macchinetta in una *location*

Attori Principali: Utente amministratore autenticato

Precondizioni:

- L'utente amministratore ha effettuato il *login* con successo al pannello di amministrazione;
- Esiste almeno una *location*;
- Esiste almeno una macchinetta associata alla *location*;

Postcondizioni: Le modifiche della macchinetta selezionata sono state salvate correttamente nel sistema

Descrizione: L'utente amministratore desidera apportare modifiche alle informazioni di una macchinetta specifica nella *location* in cui è posizionata nel sistema:

- L'utente amministratore accede al pannello di amministrazione.
- L'utente amministratore visualizza la lista delle *location* disponibili.
- L'utente amministratore scorre fino alla macchinetta da visualizzare;
- L'utente seleziona la macchinetta;
- Il sistema mostra le informazioni relative alla macchinetta selezionata;
- L'utente amministratore visualizza la macchinetta da modificare.
- L'utente amministratore clicca un pulsante per la modifica dei dettagli della macchinetta;
- Il sistema mostra un *form* con i dati da modificare della *location*.
- L'utente amministratore apporta le modifiche desiderate ai campi del prodotto.
- L'utente amministratore salva le modifiche;
- Il sistema valida i dati inseriti e aggiorna la specifica *location* nel sistema;
- Il sistema aggiorna la pagina delle macchinette;

- L'utente amministratore visualizza i dettagli aggiornati della macchinetta.

UC31: Visualizzazione della lista di prodotti disponibili in una macchinetta

Attori Principali: Utente amministratore autenticato

Precondizioni:

- L'utente amministratore ha effettuato il *login* con successo al pannello di amministrazione;
- Esiste almeno una *location*;
- Esiste almeno una macchinetta.

Postcondizioni: L'utente amministratore visualizza la lista completa dei prodotti associati alla macchinetta selezionata

Descrizione: L'utente amministratore desidera visualizzare la lista delle macchinette presenti nel sistema e suddivise per *location*:

- L'utente amministratore accede al pannello di amministrazione.
- L'utente amministratore visualizza la lista delle *location* disponibili.
- L'utente amministratore scorre fino alla macchinetta da visualizzare;
- L'utente seleziona la macchinetta;
- Il sistema mostra le informazioni relative alla macchinetta selezionata;
- L'utente amministratore visualizza la lista dei prodotti disponibili divisi per macchinetta.

UC32: Visualizzazione delle informazioni di dettaglio di un prodotto di una macchinetta

Attori Principali: Utente amministratore autenticato

Precondizioni:

- L'utente amministratore ha effettuato il *login* con successo al pannello di amministrazione;
- Esiste almeno una *location*;
- Esiste almeno una macchinetta;
- Esiste almeno un prodotto.

Postcondizioni: L'utente amministratore visualizza le informazioni di dettaglio del prodotto di una macchinetta selezionato.

Descrizione: L'utente amministratore desidera visualizzare le informazioni di dettaglio di un prodotto specifico di una macchinetta nel sistema:

- L'utente amministratore accede al pannello di amministrazione.
- L'utente amministratore visualizza la lista delle *location* disponibili.
- L'utente amministratore scorre fino alla macchinetta da visualizzare;
- L'utente seleziona la macchinetta;
- Il sistema mostra i dettagli relativi alla macchinetta selezionata;
- L'utente amministratore seleziona il prodotto di cui desidera visualizzare le informazioni di dettaglio;
- L'utente amministratore visualizza i dettagli del prodotto selezionato.

UC33: Modifica delle informazioni di dettaglio di un prodotto di una macchinetta

Attori Principali: Utente amministratore autenticato

Precondizioni:

- L'utente amministratore ha effettuato il *login* con successo al pannello di amministrazione;
- Esiste almeno una *location*;
- Esiste almeno una macchinetta;
- Esiste almeno un prodotto.

Postcondizioni: Le modifiche alle informazioni di dettaglio del prodotto di una macchinetta sono salvate correttamente nel sistema.

Descrizione: L'utente amministratore desidera apportare modifiche alle informazioni di dettaglio di un prodotto specifico di una macchinetta nel sistema:

- L'utente amministratore accede al pannello di amministrazione.
- L'utente amministratore visualizza la lista delle *location* disponibili.
- L'utente amministratore scorre fino alla macchinetta da visualizzare;
- L'utente seleziona la macchinetta;
- Il sistema mostra le informazioni relative alla macchinetta selezionata;
- L'utente amministratore visualizza i prodotti contenuti nella macchinetta.
- L'utente amministratore seleziona il prodotto da modificare;
- L'utente amministratore visualizza i dettagli dell'item selezionato.
- L'utente amministratore clicca un pulsante per la modifica dei dettagli della macchinetta;
- Il sistema mostra un *form* con i dati da modificare della *location*.

- L'utente amministratore apporta le modifiche desiderate ai campi del prodotto.
- L'utente amministratore salva le modifiche;
- Il sistema valida i dati inseriti e aggiorna lo specifico prodotto nel sistema;
- Il sistema aggiorna la pagina dei prodotti;
- L'utente amministratore visualizza i dettagli aggiornati dei prodotti.

UC34: Associazione di un prodotto piattaforma ad un prodotto macchinetta

Attori Principali: Utente amministratore autenticato

Precondizioni:

- L'utente amministratore ha effettuato il *login* con successo al pannello di amministrazione;
- Esiste almeno una *location*;
- Esiste almeno una macchinetta;
- Esiste almeno un prodotto;
- Non esiste un'associazione fra un prodotto macchinetta e un prodotto piattaforma.

Postcondizioni: Il prodotto macchinetta è associato con successo ad un prodotto piattaforma.

Descrizione: L'utente amministratore desidera associare un prodotto macchinetta con un prodotto piattaforma nel sistema:

- L'utente amministratore accede al pannello di amministrazione.
- L'utente amministratore visualizza la lista delle *location* disponibili.
- L'utente amministratore scorre fino alla macchinetta da visualizzare;
- L'utente seleziona la macchinetta;
- Il sistema mostra i dettagli relativi alla macchinetta selezionata;
- L'utente amministratore seleziona il prodotto di cui desidera visualizzare le informazioni di dettaglio;
- L'utente amministratore visualizza i dettagli del prodotto selezionato;
- L'utente amministratore seleziona il prodotto piattaforma con cui vuole associare il prodotto fra le opzioni fornite dal sistema per la macchinetta selezionata;
- Il sistema conferma l'associazione del prodotto macchinetta con il prodotto piattaforma;
- L'utente amministratore visualizza i dettagli aggiornati del prodotto macchinetta.

UC35: Dissociazione di un prodotto piattaforma da un prodotto macchinetta**Attori Principali:** Utente amministratore autenticato**Precondizioni:**

- L'utente amministratore ha effettuato il *login* con successo al pannello di amministrazione;
- Esiste almeno una *location*;
- Esiste almeno una macchinetta;
- Esiste almeno un prodotto;
- Esiste un'associazione fra un prodotto macchinetta e un prodotto piattaforma.

Postcondizioni: Il prodotto macchinetta è dissociato con successo dal un prodotto piattaforma.**Descrizione:** L'utente amministratore desidera dissociare un prodotto piattaforma da un prodotto macchinetta nel sistema:

- L'utente amministratore accede al pannello di amministrazione.
- L'utente amministratore visualizza la lista delle *location* disponibili.
- L'utente amministratore scorre fino alla macchinetta da visualizzare;
- L'utente seleziona la macchinetta;
- Il sistema mostra i dettagli relativi alla macchinetta selezionata;
- L'utente amministratore seleziona il prodotto di cui desidera visualizzare le informazioni di dettaglio;
- L'utente amministratore visualizza i dettagli del prodotto selezionato;
- L'utente amministratore clicca il pulsante per dissociare il prodotto macchinetta dal prodotto piattaforma;
- Il sistema conferma di aver dissociato il prodotto macchinetta dal prodotto piattaforma;
- L'utente amministratore visualizza i dettagli aggiornati del prodotto macchinetta.

4.3.8 Casi d'Uso - Prodotti

UC36: Visualizzazione della lista dei prodotti piattaforma**Attori Principali:** Utente amministratore autenticato**Precondizioni:** L'utente amministratore ha effettuato il *login* con successo al pannello di amministrazione.

Postcondizioni: L'utente amministratore visualizza la lista completa dei prodotti piattaforma disponibili.

Descrizione: L'utente amministratore desidera visualizzare l'intera lista dei prodotti piattaforma disponibili nel sistema:

- L'utente amministratore accede al pannello di amministrazione;
- Il sistema calcola e mostra le informazioni relative ai prodotti piattaforma disponibili;
- L'utente amministratore visualizza le informazioni relative alle prodotti piattaforma.

UC37: Visualizzazione della lista dei prodotti piattaforma filtrati per categoria

Attori Principali: Utente amministratore autenticato

Precondizioni: L'utente amministratore ha effettuato il *login* con successo al pannello di amministrazione.

Postcondizioni: L'utente amministratore visualizza la lista filtrata dei prodotti piattaforma disponibili.

Descrizione: L'utente amministratore desidera visualizzare solo i prodotti piattaforma della categoria di interesse che sono disponibili nel sistema:

- L'utente amministratore accede al pannello di amministrazione;
- Il sistema calcola e mostra le informazioni relative ai prodotti piattaforma disponibili;
- L'utente amministratore visualizza le informazioni relative alle prodotti piattaforma;
- L'utente amministratore seleziona l'opzione per filtrare i prodotti in base alla categoria di suo interesse;
- Il sistema applica il filtro e mostra solo i prodotti filtrati in base alla categoria scelta;
- L'utente amministratore visualizza la lista dei prodotti filtrati.

UC38: Creazione di un nuovo prodotto piattaforma

Attori Principali: Utente amministratore autenticato

Precondizioni: L'utente amministratore ha effettuato il *login* con successo al pannello di amministrazione.

Postcondizioni: Il nuovo prodotto piattaforma è stato creato e aggiunto al sistema.

Descrizione: L'utente amministratore vuole aggiungere un nuovo prodotto piattaforma nel sistema:

- L'utente amministratore accede al pannello di amministrazione.

- L'utente amministratore clicca un pulsante per aggiungere il nuovo prodotto piattaforma.
- Il sistema mostra un *form* da compilare per la creazione del prodotto piattaforma.
- L'utente amministratore inserisce le informazioni specifiche del prodotto nell'apposito *form*;
- L'utente amministratore conferma la creazione del prodotto piattaforma;
- Il sistema valida i dati inseriti e crea una nuova prodotto piattaforma nel sistema;
- Il sistema aggiorna la pagina dei prodotti;
- L'utente amministratore visualizza i dettagli del nuovo prodotto piattaforma creato.

UC39: Configurazione dei dosaggi di un prodotto piattaforma

Attori Principali: Utente amministratore autenticato

Precondizioni:

- L'utente amministratore ha effettuato il *login* con successo al pannello di amministrazione.
- Esiste almeno un prodotto nel sistema.

Postcondizioni: Il nuovo prodotto piattaforma è stato creato e aggiunto al sistema.

Descrizione: L'utente amministratore vuole aggiungere un nuovo prodotto piattaforma nel sistema:

- L'utente amministratore accede al pannello di amministrazione;
- L'utente amministratore visualizza i prodotti piattaforma disponibili;
- L'utente seleziona un prodotto piattaforma di cui modificare i dosaggi;
- Il sistema mostra i dettagli del prodotto selezionato.
- L'amministratore apporta le modifiche ai dosaggi del prodotto, specificando le quantità corrette degli ingredienti o componenti;
- Il sistema valida i dati inseriti e aggiorna i dosaggi del prodotto piattaforma nel sistema;
- Il sistema aggiorna la pagina dei prodotti;
- L'utente amministratore visualizza i dettagli aggiornati del prodotto piattaforma.

4.3.9 Casi d'Uso - Utenti

UC40: Visualizzazione della della lista di utenti

Attori Principali: Utente amministratore autenticato

Precondizioni: L'utente amministratore ha effettuato il *login* con successo al pannello di amministrazione.

Postcondizioni: L'utente amministratore visualizza la lista completa degli utenti iscritti nel sistema.

Descrizione: L'utente amministratore desidera visualizzare l'intera lista degli utenti disponibili nel sistema:

- L'utente amministratore accede al pannello di amministrazione;
- Il sistema calcola e mostra le informazioni relative agli utenti disponibili;
- L'utente amministratore visualizza le informazioni relative agli utenti.

UC41: Visualizzazione della lista degli utenti filtrati per dipendenti

Attori Principali: Utente amministratore autenticato

Precondizioni: L'utente amministratore ha effettuato il *login* con successo al pannello di amministrazione.

Postcondizioni: L'utente amministratore visualizza la lista filtrata degli utenti che sono anche dipendenti dell'azienda.

Descrizione: L'utente amministratore desidera visualizzare solo gli utenti che sono anche dipendenti dell'azienda:

- L'utente amministratore accede al pannello di amministrazione;
- Il sistema calcola e mostra le informazioni relative agli utenti del sistema;
- L'utente amministratore visualizza le informazioni relative agli utenti del sistema;
- L'utente amministratore seleziona l'opzione per filtrare gli utenti in base al loro status;
- Il sistema applica il filtro e mostra solo gli utenti che sono dipendenti dell'azienda;
- L'utente amministratore visualizza la lista dei dipendenti.

4.3.10 Casi d'Uso - Promozioni

UC42: Visualizzazione della lista di promozioni

Attori Principali: Utente amministratore autenticato

Precondizioni: L'utente amministratore ha effettuato il *login* con successo al pannello di amministrazione.

Postcondizioni: L'utente amministratore visualizza la lista completa delle promozioni disponibili.

Descrizione: L'utente amministratore desidera visualizzare l'intera lista delle promozioni disponibili nel sistema:

- L'utente amministratore accede al pannello di amministrazione;
- Il sistema calcola e mostra le informazioni relative alle promozioni disponibili;
- L'utente amministratore visualizza le informazioni relative alle promozioni.

UC43: Visualizzazione del dettaglio di una promozione

Attori Principali: Utente amministratore autenticato

Precondizioni:

- L'utente amministratore ha effettuato il *login* con successo al pannello di amministrazione;
- Esiste almeno una promozione.

Postcondizioni: L'utente amministratore visualizza i dettagli della promozione selezionata.

Descrizione: L'utente amministratore desidera visualizzare i dettagli completi di una promozione specifica del sistema:

- L'utente amministratore accede al pannello di amministrazione;
- Il sistema calcola e mostra le informazioni relative alle promozioni disponibili;
- L'utente amministratore visualizza le informazioni relative alle promozioni;
- L'utente amministratore seleziona la promozione di cui desidera visualizzare i dettagli;
- Il sistema mostra i dettagli completi della promozione selezionata;
- L'utente amministratore visualizza i dettagli della promozione.

UC44: Visualizzazione della lista delle promozioni piattaforma filtrati per categoria

Attori Principali: Utente amministratore autenticato

Precondizioni: L'utente amministratore ha effettuato il *login* con successo al pannello di amministrazione.

Postcondizioni: L'utente amministratore visualizza la lista filtrata delle promozioni attive.

Descrizione: L'utente amministratore desidera visualizzare solo le promozioni che sono attive:

- L'utente amministratore accede al pannello di amministrazione;
- Il sistema calcola e mostra le informazioni relative alle promozioni disponibili;
- L'utente amministratore visualizza le informazioni relative alle promozioni;
- L'utente amministratore seleziona l'opzione per filtrare le promozioni base al loro status;
- Il sistema applica il filtro e mostra solo le promozioni attive;
- L'utente amministratore visualizza la lista delle promozioni attive.

UC45: Creazione di una nuova promozione

Attori Principali: Utente amministratore autenticato

Precondizioni: L'utente amministratore ha effettuato il *login* con successo al pannello di amministrazione.

Postcondizioni: Il nuovo prodotto piattaforma è stato creato e aggiunto al sistema.

Descrizione: L'utente amministratore vuole aggiungere un nuovo prodotto piattaforma nel sistema:

- L'utente amministratore accede al pannello di amministrazione.
- L'utente amministratore clicca un pulsante per aggiungere la nuova promozione.
- Il sistema mostra un *form* da compilare per la creazione la promozione.
- L'utente amministratore inserisce le informazioni specifiche della promozione nell'apposito *form*;
- L'utente amministratore conferma la creazione della promozione;
- Il sistema valida i dati inseriti e crea una nuova promozione nel sistema;
- Il sistema aggiorna la pagina dei promozioni;
- L'utente amministratore visualizza i dettagli della nuova promozione creata.

UC46: Modifica di una promozione

Attori Principali: Utente amministratore autenticato

Precondizioni:

- L'utente amministratore ha effettuato il *login* con successo al pannello di amministrazione;
- Esiste almeno una promozione.

Postcondizioni: Le modifiche alla promozione sono state salvate correttamente nel sistema

Descrizione: L'utente amministratore vuole effettuare delle modifiche ad una promozione nel sistema:

- L'utente amministratore accede al pannello di amministrazione.
- L'utente amministratore visualizza la lista delle promozioni disponibili.
- L'utente amministratore seleziona la promozione da modificare.
- Il sistema mostra un *form* con i dati da modificare della promozione.
- L'utente amministratore apporta le modifiche desiderate ai campi della promozione.
- L'utente amministratore salva le modifiche;
- Il sistema valida i dati inseriti e aggiorna la specifica promozione nel sistema;
- Il sistema aggiorna la pagina delle promozioni;
- L'utente amministratore visualizza i dettagli aggiornati delle promozioni.

4.4 Tracciamento dei requisiti

In seguito al completamento della stesura dei casi d'uso, mi sono concentrato sulla tracciatura dei requisiti del progetto. A questo scopo in seguito sono state definite delle tabelle che ne tracciano i contenuti.

La categorizzazione dei requisiti è definita in base agli obiettivi dello stage.

Dal lato dell'azienda ospitante, tutti i seguenti requisiti risultano obbligatori o, in alcuni casi, desiderabili.

I requisiti sono identificati da un codice univoco che segue la convenzione:

R[Tipologia][Importanza][Codice]

- **Tipologia:** rappresenta il tipo di requisito che può assumere uno dei seguenti valori letterali:
 - **V:** requisito di **Vincolo** con cui si descrivono i vincoli relativi ai servizi che il sistema offre;
 - **F:** requisito **Funzionale** con cui si descrivono i servizi o le funzioni che il sistema offre;
 - **Q:** requisito di **Qualità** con cui si descrivono i vincoli di qualità da realizzare.
- **Importanza:** ossia definisce un indice di importanza che viene associata ad ogni requisito e può assumere uno dei seguenti valori numerici:
 - **O:** requisito **Obbligatorio** che deve essere necessariamente soddisfatto per garantire la presenza delle funzionalità di base all'interno del sistema;
 - **D:** requisito **Desiderabile** che non vincola il sistema nel suo funzionamento. La sua implementazione fornirà al prodotto una maggiore completezza;
 - **F:** requisito **Facoltativo** che determina ulteriore completezza all'interno del sistema. Rispetto ai precedenti, ha maggiore probabilità di comportare un dispendio di risorse che favoriscono l'aumento dei costi del progetto.
- **Codice:** identificatore univoco, tipicamente a due cifre, che insieme alla tipologia ha la funzione di identificare il requisito in esame.

4.4.1 Requisiti funzionali

Requisito	Descrizione	Fonte
RFO-01	L'utente amministratore deve poter accedere alla piattaforma	UC1
RFO-02	L'utente amministratore deve poter visualizzare un'errore di autenticazione in fase di accesso alla piattaforma	UC2
RFO-03	L'utente amministratore deve poter effettuare il <i>logout</i> dalla piattaforma	UC3
RDO-04	L'utente amministratore deve poter recuperare la <i>password</i> in caso di smarrimento	UC4
RFO-05	L'utente amministratore deve poter visualizzare il grafico che rappresenta le vendite totali effettuate nel sistema.	UC5
RFO-06	L'utente amministratore deve poter visualizzare la variazione delle vendite totali effettuate nel sistema rispetto al periodo precedente.	UC5
RFO-07	L'utente amministratore deve poter filtrare per <i>location</i> la lista delle vendite per visualizzare le vendite totali.	UC6
RFO-08	L'utente amministratore deve poter filtrare per macchinetta la lista delle vendite per visualizzare le vendite totali.	UC7
RFO-09	L'utente amministratore deve poter filtrare la lista delle vendite per visualizzare le vendite totali degli ultimi 7 giorni.	UC8
RFO-10	L'utente amministratore deve poter filtrare la lista delle vendite per visualizzare le vendite totali degli ultimi 30 giorni.	UC8
RFO-11	L'utente amministratore deve poter filtrare la lista delle vendite per visualizzare le vendite totali degli ultimi 60 giorni.	UC8
RFO-12	L'utente amministratore deve poter filtrare la lista delle vendite per visualizzare le vendite totali in un range di date scelto dal calendario.	UC8
RFO-13	L'utente amministratore deve poter visualizzare le statistiche sui prodotti più venduti nel sistema	UC9
RFO-14	L'utente amministratore deve poter filtrare per <i>location</i> la lista delle vendite per visualizzare i prodotti più venduti.	UC10
RFO-15	L'utente amministratore deve poter filtrare per macchinetta la lista delle vendite per visualizzare i prodotti più venduti.	UC11
RFO-16	L'utente amministratore deve poter filtrare la lista delle vendite per visualizzare i prodotti più venduti negli ultimi 7 giorni.	UC12
RFO-17	L'utente amministratore deve poter filtrare la lista delle vendite per visualizzare i prodotti più venduti negli ultimi 30 giorni.	UC12
RFO-18	L'utente amministratore deve poter filtrare la lista delle vendite per visualizzare i prodotti più venduti negli ultimi 60 giorni.	UC12

Requisito	Descrizione	Fonte
RFO-19	L'utente amministratore deve poter filtrare la lista delle vendite per visualizzare i prodotti più venduti in un range di date scelto dal calendario.	UC12
RFO-20	L'utente amministratore deve poter visualizzare le statistiche sul numero di utenti registrati nell'app	UC13
RFO-21	L'utente amministratore deve poter filtrare per <i>location</i> la lista delle vendite per visualizzare il numero di utenti che hanno acquistato qualcosa.	UC14
RFO-22	L'utente amministratore deve poter filtrare la lista delle vendite per visualizzare il numero di utenti che ha acquistato negli ultimi 7 giorni.	UC15
RFO-23	L'utente amministratore deve poter filtrare la lista delle vendite per visualizzare il numero di utenti che ha acquistato negli ultimi 30 giorni.	UC15
RFO-24	L'utente amministratore deve poter filtrare la lista delle vendite per visualizzare il numero di utenti che ha acquistato negli ultimi 60 giorni.	UC15
RFO-25	L'utente amministratore deve poter filtrare la lista delle vendite per visualizzare il numero di utenti che ha acquistato in un range di date scelto dal calendario.	UC15
RFO-26	L'utente amministratore deve poter visualizzare le statistiche sulla spesa media complessiva degli utenti	UC16
RFO-27	L'utente amministratore deve poter filtrare per <i>location</i> la lista delle vendite per visualizzare la spesa media degli utenti.	UC17
RFO-28	L'utente amministratore deve poter filtrare per macchinetta la lista delle vendite per visualizzare la spesa media degli utenti.	UC18
RFO-29	L'utente amministratore deve poter filtrare la lista delle vendite per visualizzare la spesa media degli utenti negli ultimi 7 giorni.	UC19
RFO-30	L'utente amministratore deve poter filtrare la lista delle vendite per visualizzare la spesa media degli utenti negli ultimi 30 giorni.	UC19
RFO-31	L'utente amministratore deve poter filtrare la lista delle vendite per visualizzare la spesa media degli utenti negli ultimi 60 giorni.	UC19
RFO-32	L'utente amministratore deve poter filtrare la lista delle vendite per visualizzare la spesa media degli utenti in un range di date scelto dal calendario.	UC19
RFO-33	L'utente amministratore deve poter visualizzare le statistiche di spesa media degli utenti per fascia oraria.	UC20
RFO-34	L'utente amministratore deve poter filtrare per <i>location</i> la lista delle vendite per visualizzare la spesa media degli utenti per fascia oraria.	UC21
RFO-35	L'utente amministratore deve poter filtrare per macchinetta la lista delle vendite per visualizzare la spesa media degli utenti per fascia oraria.	UC22

Requisito	Descrizione	Fonte
RFO-36	L'utente amministratore deve poter filtrare per <i>location</i> la lista delle vendite per visualizzare la spesa media degli utenti per fascia oraria.	UC23
RFO-37	L'utente amministratore deve poter filtrare la lista delle vendite per visualizzare la spesa media degli utenti per fascia oraria negli ultimi 7 giorni.	UC24
RFO-38	L'utente amministratore deve poter filtrare la lista delle vendite per visualizzare la spesa media degli utenti per fascia oraria negli ultimi 30 giorni.	UC24
RFO-39	L'utente amministratore deve poter filtrare la lista delle vendite per visualizzare la spesa media degli utenti per fascia oraria negli ultimi 60 giorni.	UC24
RFO-40	L'utente amministratore deve poter filtrare la lista delle vendite per visualizzare la spesa media degli utenti in un range di date scelto dal calendario.	UC24
RFF-41	L'utente amministratore deve poter visualizzare una lista completa delle <i>location</i> presenti nel sistema.	UC25
RFF-42	L'utente amministratore deve poter creare una nuova <i>location</i> , specificando il nome e l'indirizzo.	UC26
RFF-43	L'utente amministratore deve poter modificare le informazioni di una <i>location</i> , come il nome o l'indirizzo.	UC27
RFF-44	L'utente amministratore deve poter visualizzare una lista delle macchinette presenti in una specifica <i>location</i> .	UC28
RFF-45	L'utente amministratore deve poter creare una nuova macchinetta all'interno di una specifica <i>location</i> .	UC29
RFF-46	L'utente amministratore deve poter modificare le informazioni di una macchinetta all'interno di una specifica <i>location</i> .	UC30
RFF-47	L'utente amministratore deve poter visualizzare una lista dei prodotti disponibili in una specifica macchinetta di un distributore automatico.	UC31
RFF-48	L'utente amministratore deve poter visualizzare le informazioni di dettaglio di un prodotto associato a una macchinetta, come il nome, il prezzo e la disponibilità.	UC32
RFF-50	L'utente amministratore deve poter modificare le informazioni di dettaglio di un prodotto associato a una macchinetta.	UC33
RFD-51	L'utente amministratore deve poter associare un prodotto piattaforma ad un prodotto macchinetta di un distributore automatico.	UC34
RFD-52	L'utente amministratore deve poter rimuovere l'associazione tra un prodotto piattaforma da un prodotto macchinetta di un distributore automatico.	UC35
RFD-53	L'utente amministratore deve poter visualizzare una lista completa dei prodotti disponibili.	UC36
RFF-54	L'utente amministratore deve poter filtrare la lista dei prodotti per visualizzare solo quelli di una specifica categoria.	UC37

Requisito	Descrizione	Fonte
RFD-55	L'utente amministratore deve poter creare nuovi prodotti piattaforma.	UC38
RFD-56	L'utente amministratore deve poter configurare i dosaggi di un prodotto, ad esempio per specificare la quantità di zucchero o latte in una bevanda.	UC39
RFF-57	L'utente amministratore deve poter visualizzare una lista completa degli utenti registrati, inclusi dipendenti e utenti base.	UC40
RFF-58	L'utente amministratore deve poter filtrare la lista degli utenti per visualizzare solo quelli che sono dipendenti.	UC41
RFF-59	L'utente amministratore deve poter visualizzare una lista completa delle promozioni disponibili.	UC42
RFF-60	L'utente amministratore deve poter visualizzare i dettagli completi di una promozione.	UC43
RFF-61	L'utente amministratore deve poter filtrare la lista delle promozioni visualizzare solo quelle attualmente attive.	UC44
RFF-62	L'utente amministratore deve poter creare nuove promozioni.	UC45
RFF-63	L'utente amministratore deve poter modificare le promozioni.	UC46

Tabella 4.1: Tabella che raccoglie la tracciatura dei requisiti funzionali del progetto

4.4.2 Requisiti di vincolo

Requisito	Descrizione	Fonte
RVO-01	Il progetto dovrà essere sviluppato utilizzando il <i>framework Angular 15</i>	Scelta interna
RVO-02	Il progetto dovrà essere sviluppato utilizzando il linguaggio <i>TypeScript</i>	Scelta interna
RVO-03	Il progetto dovrà essere sviluppato utilizzando la libreria <i>Chart.js</i>	Scelta interna
RVO-04	Il progetto dovrà essere sviluppato utilizzando la libreria <i>Angular Material</i>	Scelta interna
RVO-05	Il progetto dovrà essere sviluppato utilizzando <i>Apollo Client</i>	Scelta interna
RVO-06	L'applicazione <i>web</i> dovrà comunicare con il <i>backend</i> tramite <i>GraphQL</i>	Scelta interna
RVO-07	L'applicazione dovrà comunicare con il <i>backend Da Mario</i>	Scelta interna
RVO-08	I dati relativi all'utente amministratore devono rimanere nel <i>local storage</i> del <i>browser</i>	Scelta interna

Tabella 4.2: Tabella che raccoglie la tracciatura dei requisiti di vincolo del progetto

4.4.3 Requisiti qualitativi

Requisito	Descrizione	Fonte
RQO-01	Ogni scelta di aggiornamento della <i>UX</i> dev'essere discussa con il designer e documentata	Scelta interna
RQO-02	Il sito dev'essere consultabile da <i>computer</i>	Scelta interna
RQO-03	Il sito dev'essere consultabile da <i>tablet</i>	Scelta interna
RQF-04	Il sito dev'essere consultabile da <i>mobile</i>	Scelta interna
RQO-05	Il sito si deve adattare in base alla dimensione dello schermo	Scelta interna

Tabella 4.3: Tabella che raccoglie la tracciatura dei requisiti qualitativi del progetto

Capitolo 5

Progettazione e codifica

In questo capitolo viene rappresentato lo studio effettuato in termini di progettazione dell'applicazione. Viene discusso il sistema a partire dall'architettura, che riprende quella della [framework](#), fino a rappresentare in modo dettagliato tutte le funzionalità descritte nell'analisi dei requisiti sotto forma di bisogni lato software. Il capitolo continua con una descrizione dei pattern che ho implementato e della struttura del software che ho utilizzato. In conclusione ho elencato le interfacce che ho realizzato e spiegato il processo di pubblicazione del sito in ambiente di staging

5.1 Progettazione architetturale

5.1.1 Architettura di Angular

Come riporta la documentazione di [Angular](#):

L'architettura di un'applicazione [Angular](#) si basa su alcuni concetti fondamentali. Gli elementi di base del [framework Angular](#) sono i **componenti Angular**, organizzati in `NgModules`. Gli `NgModules` raccolgono il codice correlato in insiemi funzionali. [2]

Un'applicazione [Angular](#) è sempre composta da almeno un modulo, quello principale, che consente di avviare il prodotto e che per convenzione è denominato `AppModule`. Essendo l'applicazione basata su componenti, ognuno di essi ha una struttura ricorrente, infatti con il comando `ng generate component <nome-componente>` di default vengono creati:

- Una cartella con il nome del componente;
- Un file che contiene le funzioni ed i metadati del componente, `<nome-componente>.component.ts`,
- Un file che contiene il *template HTML*, `<nome-componente>.component.html`,
- Un file che contiene il foglio di stile del componente, `<nome-componente>.component.scss`,
- Un file che contiene le specifiche di test, `<nome-componente>.component.spec.ts`.¹

¹*Angular - Component*. URL: <https://v15.angular.io/guide/component-overview>.

Angular consente di creare applicazioni in *HTML* e *JavaScript/TypeScript*. È possibile sviluppare un'applicazione componendo più *template* che adottano *HTML* in unione al linguaggio di marcatura specifico del *framework* stesso.

Angular adotta *Model View ViewModel* come *pattern* architetturale, consentendo così di dividere la logica di business da quella di presentazione.

Per poter visualizzare la struttura architetturale del sistema, la documentazione fornisce lo schema che è rappresentato in Figura 5.1.

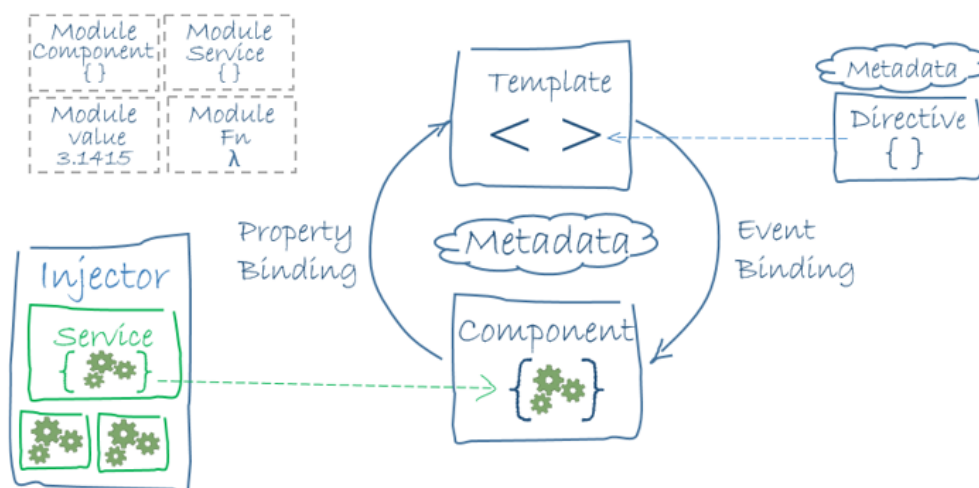


Figura 5.1: Architettura del *framework Angular*

Segue un approfondimento delle varie parti dell'architettura.

5.1.1.1 Componenti

I **componenti** *Angular* rappresentano tipicamente le viste del sistema *software*. Definiscono internamente delle classi che contengono i dati e la logica di business. Il *template HTML* fornisce la vista che verrà visualizzata dall'utente.

Per poter definire più componenti diverse, viene adottato il *pattern Decorator* (di cui seguirà una descrizione nella sezione 5.3.1). Questo *pattern* consente di identificare la classe sottostante come un componente di cui fornisce il modello e i metadati ad esso collegati. Le direttive del *template* forniscono la logica del programma e il *markup* di *binding* collega i dati dell'applicazione con il *Document Object Model* (DOM).

5.1.1.2 Moduli

Tutte le applicazioni *Angular* sono modulari. *NgModule* rappresenta il sistema che viene utilizzato dal *framework* per gestire la modularità della *web app*.

La documentazione definisce *NgModule* come:

contenitori di un blocco di codice coeso dedicato a un dominio applicativo, a un flusso di lavoro o a un insieme di funzionalità strettamente correlate.^a

^aAngular - Architettura. URL: <https://v15.angular.io/guide/architecture>.

La gestione modulare dell'applicazione favorisce lo sviluppo delle applicazioni. Grazie ai moduli è possibile sfruttare il *lazy-loading*, cioè la possibilità di caricare un blocco di codice solamente su richiesta, evitando di riempire la memoria con tutte le parti che compongono il prodotto.

5.1.1.3 Servizi

Come definito nella documentazione di *Angular*:

Il servizio è una categoria ampia che comprende qualsiasi valore, funzione o caratteristica di cui un'applicazione ha bisogno.^a

^a*Angular - Service*. URL: <https://v15.angular.io/guide/creating-injectable-service>.

Si distinguono da componenti e moduli per aumentare la modularità e la riusabilità del codice e rendere il codice più efficiente. L'adozione di servizi consente di separare maggiormente tutto ciò che è di dominio del componente (quindi aspetti per lo più legati alla vista) da aspetti di elaborazione. La documentazione cita infatti:

Idealmente, il compito di un componente è quello di consentire l'esperienza dell'utente e nient'altro. Un componente dovrebbe presentare proprietà e metodi per il binding dei dati, per mediare tra la vista (resa dal modello) e la logica dell'applicazione (che spesso include una nozione di modello).^a

^a*Angular - Service*. URL: <https://v15.angular.io/guide/creating-injectable-service>.

L'idea che ha portato all'aggiunta dei servizi in *Angular* è quella di delegare ad esso alcune funzioni che normalmente verrebbero assegnate al componente.

Facendo così il servizio può essere iniettato come dipendenza all'interno di un qualsiasi componente. *Angular* fornisce quindi uno strumento che migliora la gestione della logica dell'applicazione, spostandone la complessità nei servizi e facendo in modo che i componenti sfruttino la *Dependency Injection*.

5.1.2 Architettura del prodotto

La struttura alla base del mio progetto rispecchia fedelmente l'architettura predefinita dal *framework*. L'applicativo è contenuto all'interno della cartella `app`. Per ogni componente è stata creata l'omonima cartella contenente i documenti discussi alla sezione 5.1.1.1. Sono state aggiunte inoltre una cartella `core` che contiene i componenti principali del progetto ed una cartella `shared` che raccoglie i *fragments* per le chiamate *GraphQL*, il componente *loading* che mostra a schermo l'attesa dovuta al caricamento dei dati e tutte quelle funzioni ausiliarie di utilità per tutti i componenti.

5.2 Progettazione delle interfacce grafiche

Questa sezione del capitolo focalizza l'attenzione sulla progettazione delle interfacce grafiche della *web app* che mi è stata assegnata.

Prima di intraprendere la fase di codifica è stato necessario identificare tutte le parti che compongono il sistema.

Essendo stato inserito in un contesto già esistente, il *wireframe* del prodotto era già stato definito, quindi il processo di progettazione si è concentrato sulla valutazione della fattibilità della realizzazione di quanto studiato in precedenza dal designer.

Di conseguenza, durante il processo di progettazione il concetto originale è stato modificato anche in funzione delle funzionalità fornite da *Angular*.

La prototipazione del prodotto è stata realizzata utilizzando il *software Figma*, di cui verranno inseriti in seguito degli *screenshot* rappresentanti le tavole.

Durante questa fase, l'obiettivo principale che ho dovuto tenere in considerazione è stato quello di ottenere una navigazione ottimale per l'utente amministratore. L'ottica che ho seguito è stata quella di consentire un accesso intuitivo a tutte le funzionalità disposte dalla piattaforma, semplificando le operazioni di gestione dei distributori automatici.

5.2.1 Interfaccia di login

L'utente non autenticato o l'utente non amministratore autenticato, viene riportato nella pagina di *login*.

La pagina di *login* è composta da due campi di testo, uno per l'inserimento dell'indirizzo *email* e l'altro per l'inserimento della *password*.

Inoltre, è presente un pulsante che consente di effettuare il *login* e di conseguenza di accedere alla piattaforma.

Qualora l'utente non ricordasse la *password*, viene fornita anche un'ancora per poter raggiungere una pagina alternativa contenente un *form* per il recupero della *password*.

La Figura 5.2 mostra la tavola relativa alla pagina di *login*.

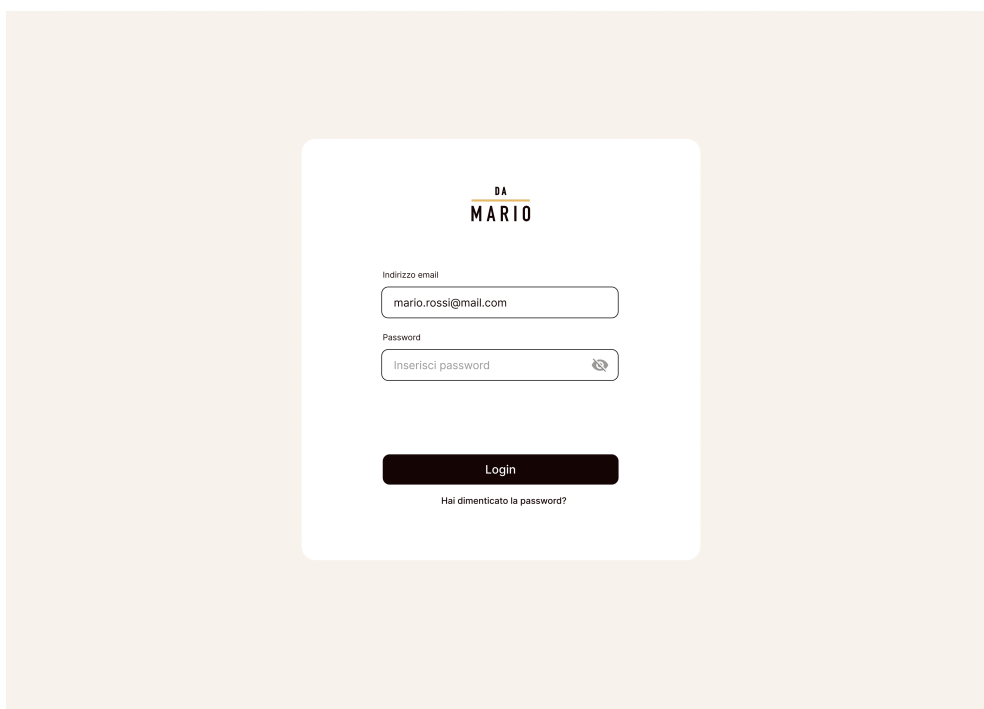


Figura 5.2: Tavola raffigurante la pagina di *login* alla piattaforma

5.2.1.1 Flusso delle informazioni

L'utente per potersi accreditare alla piattaforma dovrà seguire i passi riportati nell'elenco descritto in seguito:

- La pagina di *login* viene visualizzata all'avvio dell'applicazione. L'utente si trova di fronte ad un modulo di accesso che richiede all'utente di inserire le proprie credenziali per autenticarsi come amministratore;
- L'utente clicca sul pulsante che ne conferma le credenziali. A questo punto le credenziali vengono validate. La risposta del *backend* determinerà il passo successivo;
- Nel caso in cui l'utente inserisca credenziali non valide o non esista un account corrispondente, l'interfaccia di *login* visualizza un messaggio di errore appropriato per informare l'utente del problema;
- Se le credenziali dell'utente sono valide, l'interfaccia di *login* indirizza l'utente alla pagina principale della piattaforma di amministrazione;
- In fase di autenticazione viene creata una sessione utente che tiene traccia dell'utente autenticato attraverso un *token*. La gestione delle sessioni utente permette di mantenere l'autenticazione durante la navigazione all'interno dell'applicazione e di accedere alle informazioni dell'utente autenticato quando necessario.

5.2.2 Pagina principale

In seguito alla creazione del *token* di autenticazione, l'utente viene portato nella pagina principale della piattaforma, che d'ora in poi chiameremo *dashboard*. La *dashboard* si compone di due elementi:

- la barra di navigazione con cui l'utente può cambiare videata a seconda delle informazioni che vuole ricavare dal *backend*;
- la pagina contenente i dati del *link* selezionato nella barra di navigazione.

La Figura 5.3 che rappresenta un esempio di visualizzazione della pagina non appena l'utente amministratore effettua l'accesso.

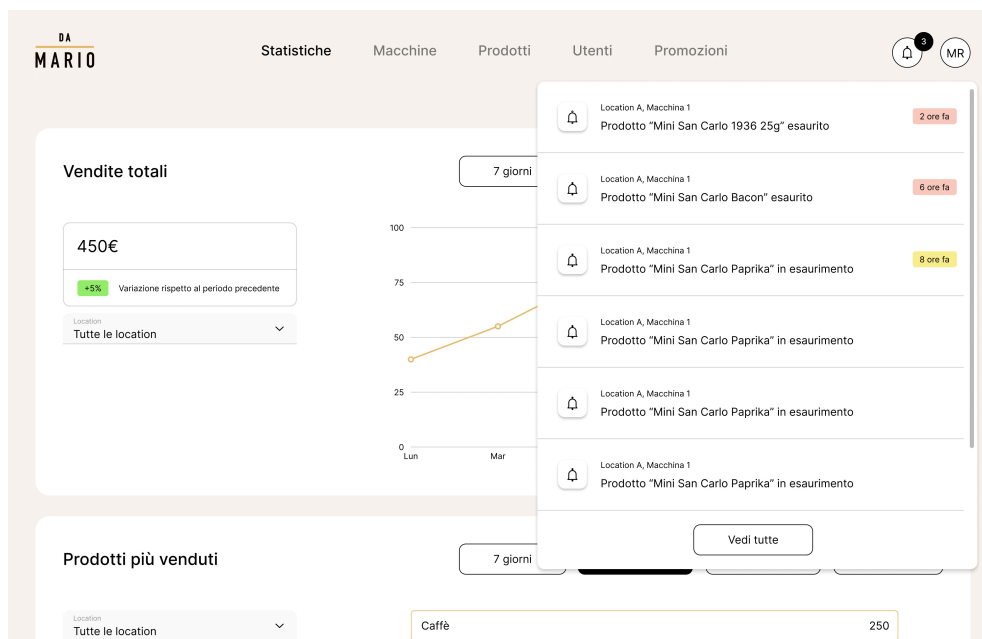


Figura 5.3: Tavola raffigurante la *dashboard*

Si può notare che nella barra in alto l'utente ha la possibilità di visualizzare il proprio profilo utente e sfogliare le notifiche relative alle macchinette.

5.2.2.1 Statistiche

Nella pagina **Statistiche** l'utente amministratore può visualizzare tutte le statistiche relative all'andamento delle vendite dei prodotti, all'orario principale di acquisto degli utenti e al numero degli utenti che effettuano acquisti.

In ognuna delle sezioni presenti in questa schermata l'utente può visualizzare:

- I dati raccolti dal *backend* sotto forma di grafico;
- I giorni a cui fanno riferimento i dati;
- I filtri che possono essere applicati sui dati.

Viene pertanto lasciata massima libertà all'utente amministratore che può restringere il campo dati a seconda delle proprie necessità.

In seguito, per ogni sezione della pagina, vengono definite le operazioni che l'utente può eseguire.

Tutti i dati che vengono presentati all'utente sono di default relativi agli ultimi 30 giorni.

Vendite totali Nella sezione *Vendite totali*, l'utente amministratore visualizza i dati relativi al guadagno ottenuto nell'ultimo periodo.

La sezione permette di:

- Visualizzare il guadagno dell'ultimo periodo;

- Confrontare il guadagno con il periodo precedente (ed eventualmente individuare il guadagno o la perdita in percentuale);
- Filtrare i dati per *location*;
- Filtrare i dati per *location* e per macchinetta;
- Filtrare i dati per periodo di 7/30/60 giorni;
- Filtrare i dati scegliendo un periodo *custom* dal calendario a comparsa.

La Figura 5.4 che rappresenta graficamente questa sezione.

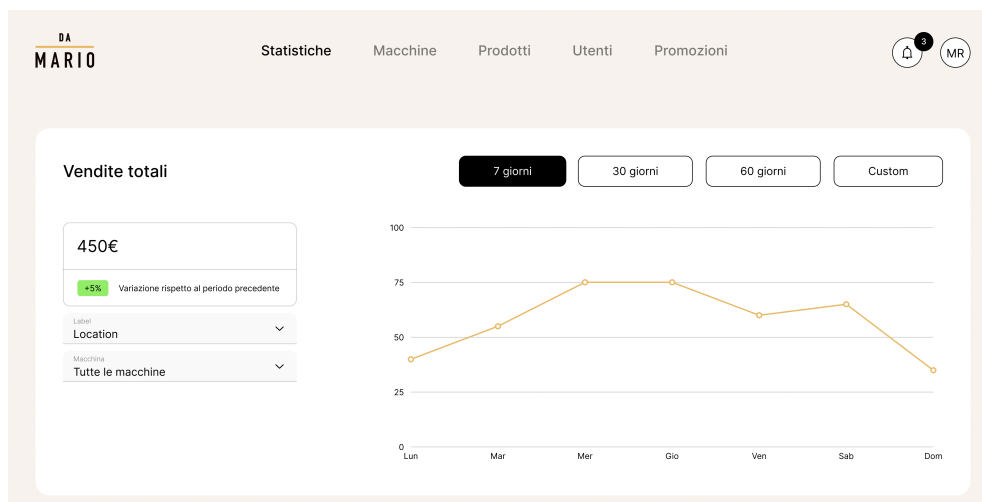


Figura 5.4: Tavola raffigurante le statistiche relative alle vendite totali

Prodotti più venduti Nella sezione *Prodotti più venduti*, l'utente amministratore visualizza i dati relativi alla top 10 dei prodotti che hanno venduto maggiormente nell'ultimo periodo, disposti in ordine decrescente.

La sezione permette di:

- Visualizzare la lista dei prodotti più venduti con nome del prodotto e quantità erogata;
- Filtrare i dati per *location*;
- Filtrare i dati per *location* e per macchinetta;
- Filtrare i dati per periodo di 7/30/60 giorni;
- Filtrare i dati scegliendo un periodo *custom* dal calendario a comparsa.

La Figura 5.5 che rappresenta graficamente questa sezione.

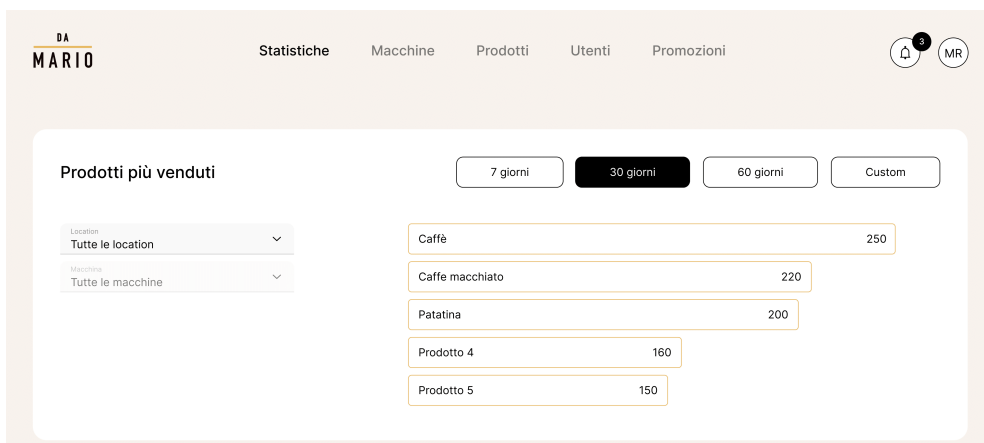


Figura 5.5: Tavola raffigurante le statistiche relative ai prodotti più venduti

Numero di utenti Nella sezione *Numero di utenti*, l'utente amministratore visualizza i dati relativi al numero di utenti che hanno effettuato acquisti nei distributori automatici nell'ultimo periodo.

La sezione permette di:

- Visualizzare il grafico contenente il numero di utenti che hanno effettuato acquisti;
- Filtrare i dati per *location*;
- Filtrare i dati per periodo di 7/30/60 giorni;
- Filtrare i dati scegliendo un periodo *custom* dal calendario a comparsa.

La Figura 5.16 che rappresenta graficamente questa sezione.

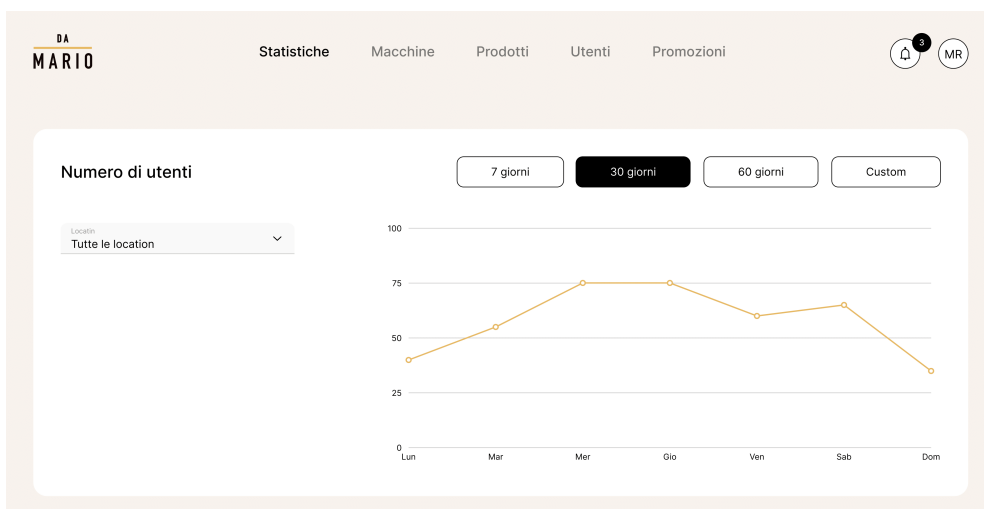


Figura 5.6: Tavola raffigurante le statistiche relative al numero di utenti che ha effettuato almeno un acquisto

Spesa media per utente Nella sezione *Spesa media per utente*, l'utente amministratore visualizza i dati relativi alla media della spesa effettuata dagli utenti nell'ultimo periodo. La spesa media viene calcolata come la somma totale dei guadagni ottenuti, suddivisi per il numero di utenti che hanno acquistato in quella determinata giornata/settimana.

La sezione permette di:

- Visualizzare la spesa media per utente dell'ultimo periodo;
- Filtrare i dati per *location*;
- Filtrare i dati per *location* e per macchinetta;
- Filtrare i dati per periodo di 7/30/60 giorni;
- Filtrare i dati scegliendo un periodo *custom* dal calendario a comparsa.

La Figura 5.7 che rappresenta graficamente questa sezione.

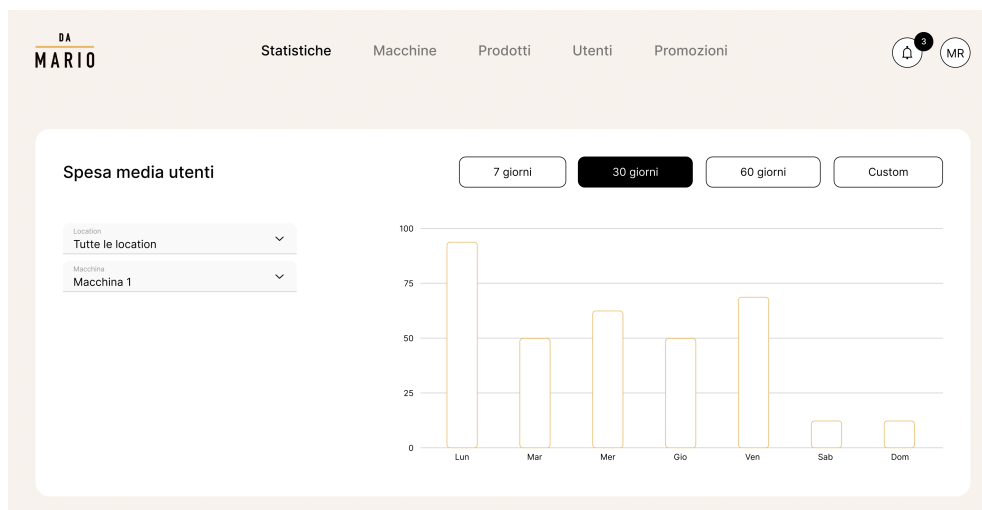


Figura 5.7: Tavola raffigurante le statistiche relative alla spesa media per utente

Spesa per fascia oraria Nella sezione *Spesa media per utente*, l'utente amministratore visualizza i dati relativi alla media della spesa effettuata dagli utenti nell'ultimo periodo. La spesa media viene calcolata come la somma totale dei guadagni ottenuti per ogni ora, suddivisi per il numero di utenti che hanno acquistato suddivisi per ora. La sezione permette di:

- Visualizzare la spesa media per utente dell'ultimo periodo;
- Filtrare i dati per *location*;
- Filtrare i dati per *location* e per macchinetta;
- Filtrare i dati per *location*, per macchinetta e per prodotto macchinetta;
- Filtrare i dati per periodo di 7/30/60 giorni;
- Filtrare i dati scegliendo un periodo *custom* dal calendario a comparsa.

La Figura 5.8 che rappresenta graficamente questa sezione.

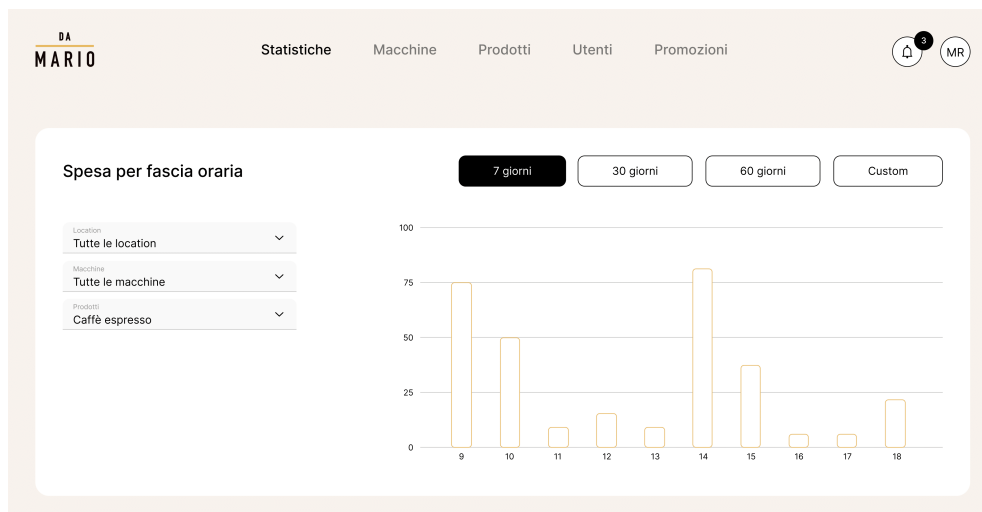


Figura 5.8: Tavola raffigurante le statistiche relative alla spesa per fascia oraria

5.2.2.2 Macchine

La pagina nominata **Macchine**, permette all'utente amministratore di visualizzare tutte le macchinette che sono gestite dall'azienda.

L'obiettivo è quello di consentire all'amministratore di poter aggiungere e/o modificare delle macchinette. Ogni *device* in questa pagina è aggregato per *location*, all'interno di una sezione che ne contenente i dati.

I bottoni presenti in questa pagina consentono di:

- Aggiungere una nuova *location*;
- Modificare una *location* esistente;
- Aggiungere una nuova macchinetta in una specifica *location*.

L'utente può visualizzare:

- La lista delle *location* con i relativi indirizzi;
- Una *card* per ogni macchinetta presente nella *location*;
- I dati specifici di ogni macchinetta, quali l'azienda della macchinetta e la sua posizione all'interno della *location*;
- Un etichetta nella *card* della macchinetta che indica il numero di prodotti che devono ancora essere configurati.

La Figura 5.9 che rappresenta graficamente quanto espresso in precedenza.

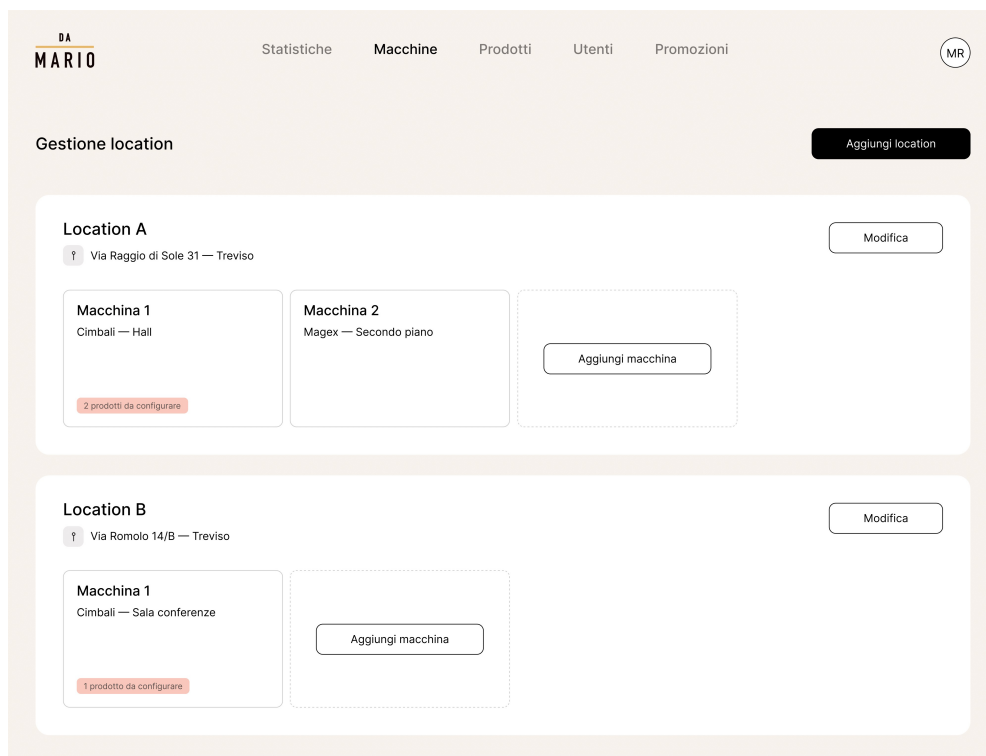


Figura 5.9: Tavola raffigurante la pagina contenente le *location* con le relative macchinetta

Quando un utente clicca una *card*, viene riportato ad una pagina contenente i dettagli della macchinetta (Figura 5.10).

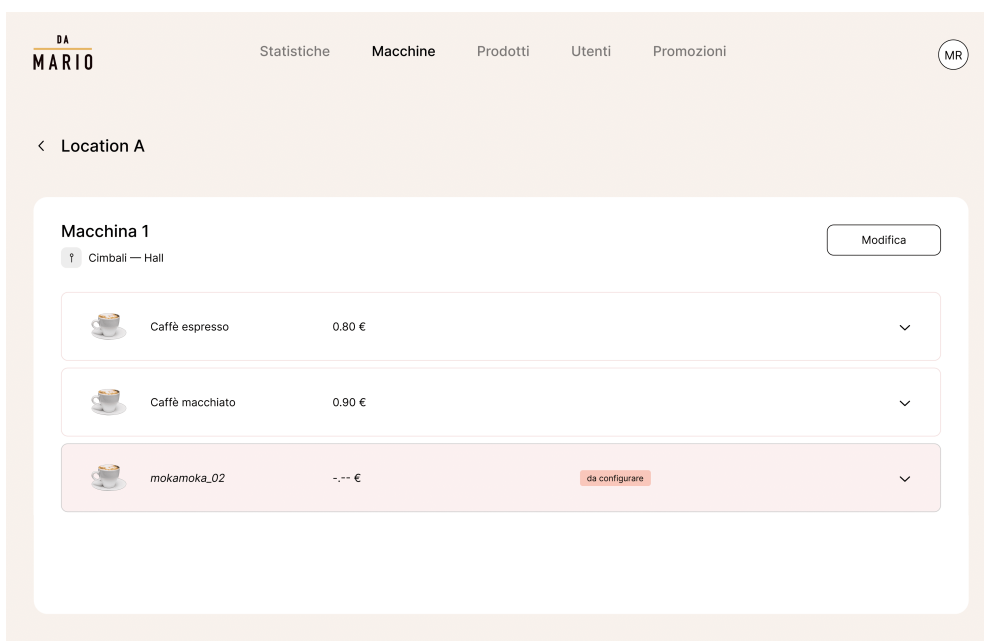


Figura 5.10: Tavola raffigurante la pagina contenente le macchinette con i relativi prodotti

L'obiettivo di questa pagina è quello di consentire all'amministratore di modificare i dati della macchinetta e di configurare, aggiungere e/o modificare i prodotti della macchinetta selezionata.

I bottoni presenti in questa pagina consentono di:

- Modificare i dati relativi alla macchinetta selezionata;
- Visualizzare o nascondere i dati relativi ai prodotti presenti nella macchinetta;
- Tornare alla pagina precedente.

L'utente può visualizzare:

- Le *card* dei prodotti con le informazioni principali (immagine prodotto, nome, prezzo);
- Se un prodotto macchinetta è configurato:
 - Il prodotto piattaforma a cui è associato il prodotto macchinetta;
 - I parametri che sono stati configurati per quel prodotto macchinetta;
 - Eventualmente anche le promozioni attive per quel prodotto (Figura 5.13).
- Se un prodotto macchinetta è da configurare:
 - La lista dei prodotti piattaforma che può associare e un pulsante per poter creare un nuovo prodotto piattaforma (Figura 5.11);
 - La lista dei parametri configurabili quando ha associato il prodotto piattaforma (Figura 5.12).

DA
MARIO

Statistiche **Macchine** Prodotti Utenti Promozioni MR

< Location A

Macchina 1 Modifica

📍 Cimbali — Hall

	Caffè espresso	0.80 €	▼
	Caffè macchiato	0.90 €	▼
	mokamoka_02	--- €	da configurare ▲

1 Associa prodotto

Devi associare questo item ad uno dei prodotti configurati nella piattaforma. Se non è ancora stato aggiunto un prodotto adatto, puoi crearlo.

Associa prodotto
Associa prodotto ▼

oppure

Crea nuovo prodotto

L'item ha 3 proprietà configurabili:

1. coffeeDoseMM1
2. coffeeWaterDose
3. hotMilkDose

Figura 5.11: Tavola raffigurante i dettagli di un prodotto da configurare a cui non è stato assegnato il prodotto piattaforma

The screenshot displays the 'DA MARIO' management interface. At the top, there are navigation tabs: 'Statistiche', 'Macchine', 'Prodotti', 'Utenti', and 'Promozioni'. The current view is for 'Location A' and 'Macchina 1'. A list of coffee items is shown, including 'Caffè espresso' (0.80 €), 'Caffè macchiato' (0.90 €), and 'mokamoka_02' (--- €) which is marked as 'da configurare'. A configuration panel for 'Mocaccino' is open, featuring a 'Salva' button and several adjustable parameters: 'Prezzo' (1.10), 'Coffee Dose (coffeeDoseMM1)' (115 mg), 'Coffee Water dose (coffeeWaterDose)' (40 cc), and 'Hot Milk Dose (hotMilkDose)' (80 mg). A warning message states: 'Assicurati che i parametri dell'item corrispondano a quelli del prodotto "Mocaccino" e salva.' An image of a coffee cup is also present in the configuration panel.

Figura 5.12: Tavola raffigurante i dettagli di un prodotto da configurare a cui è stato assegnato il prodotto piattaforma

The screenshot displays the 'Macchine' management interface. At the top, there's a navigation bar with 'MARIO' logo, 'Statistiche', 'Macchine', 'Prodotti', 'Utenti', 'Promozioni', and a user profile icon 'MR'. Below the navigation, the page title is 'Location A'. The main content area is titled 'Macchina 1' and shows a list of coffee products. The first two are 'Caffè espresso' (0.80 €) and 'Caffè macchiato' (0.90 €). The third is 'Prodotto 3' (1.10 €) with an active promotion tag. The 'Prodotto 3' card is expanded to show configuration options: 'Associa prodotto' (set to 'Moccacino'), 'Prezzo e parametri del prodotto "Moccacino"' (Price: 1.10, Coffee Dose: 115 mg, Coffee Water Dose: 40%, Hot Milk Dose: 80 mg), and 'Promozioni' (active from 01/12/22 to 31/12/22 for 'Coworking' users). There are 'Modifica' buttons for each product and the promotion, and a 'Dissocia' button for the product association.

Figura 5.13: Tavola raffigurante i dettagli di un prodotto configurato con una promozione attiva

Ogni *card* contiene un *form*. Questo è compilabile solo in due casi:

- Se è stato cliccato il pulsante per la modifica dei parametri del prodotto configurato;
- Se il prodotto macchinetta non è stato configurato.

5.2.2.3 Prodotti

La pagina nominata **Prodotti**, permette all'utente amministratore di visualizzare tutti i prodotti piattaforma che sono venduti nelle macchinette gestite dall'azienda.

L'obiettivo è quello di consentire all'amministratore di poter aggiungere e/o modificare i prodotti piattaforma. Ogni prodotto piattaforma in questa pagina è rappresentato sotto forma di *card*.

I bottoni presenti in questa pagina consentono di:

- Aggiungere un nuovo prodotto piattaforma;
- Modificare un prodotto piattaforma;
- Filtrare i prodotti piattaforma in base alla loro categoria di appartenenza.

L'utente può visualizzare:

- Una *card* per ogni piattaforma presente nel sistema;
- I dati specifici di ogni piattaforma, quali l'immagine del prodotto, il nome del prodotto ed il prezzo.

La Figura 5.14 che rappresenta graficamente quanto espresso in precedenza.

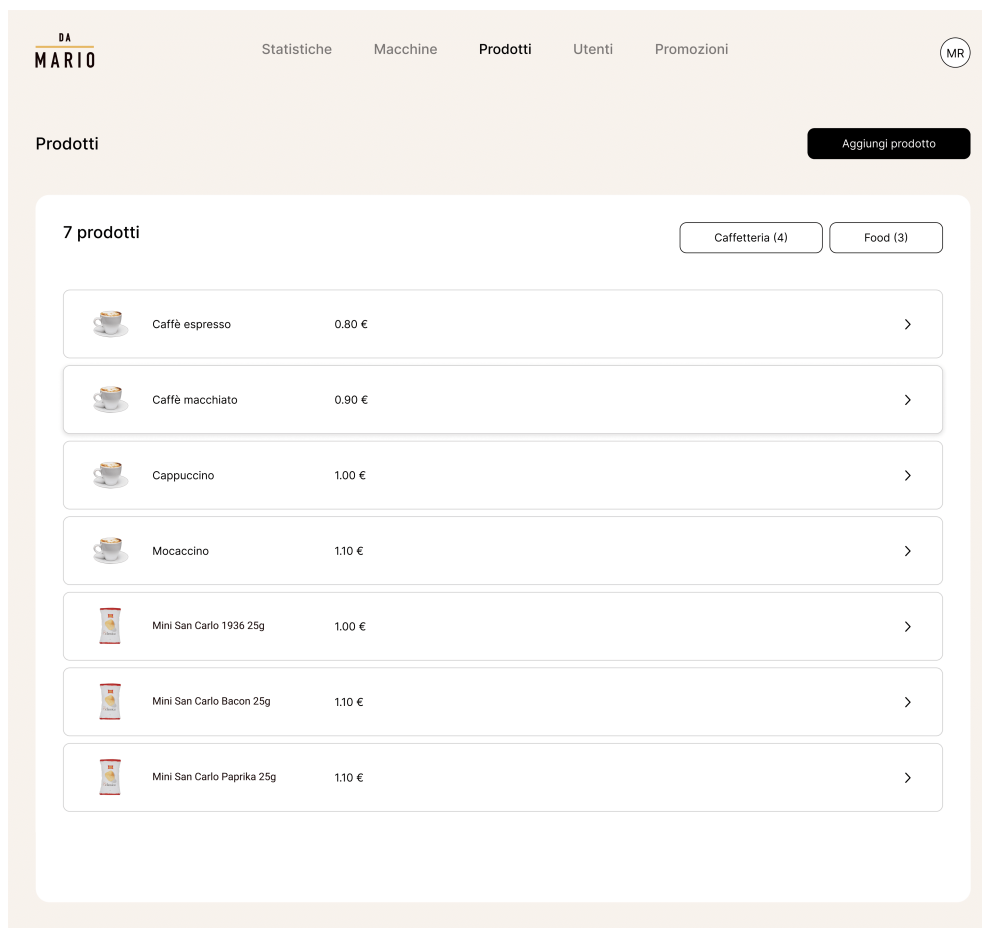


Figura 5.14: Tavola raffigurante la pagina contenente i prodotti piattaforma

Cliccando su una *card*, l'amministratore viene portato in una nuova pagina in cui gli verranno presentati tutti i parametri che potrà modificare. Questi rappresentano la configurazione di base che poi verrà riflessa su tutti i prodotti piattaforma che sono ad esso associati.

Per ogni prodotto piattaforma è possibile:

- Aggiornare il nome;
- Aggiornare il prezzo;
- Cambiare l'immagine;
- Impostare l'immagine di default;
- Creare dei nuovi parametri;
- Configurare i parametri già esistenti;
- Salvare o annullare le modifiche.

La Figura 5.15 che rappresenta graficamente quanto espresso in precedenza.

The screenshot shows the 'Prodotti' page of the MARIO platform. The header includes the MARIO logo, navigation links for 'Statistiche', 'Macchine', 'Prodotti', 'Utenti', and 'Promozioni', and a user profile icon 'MR'. Below the header, there is a 'Prodotti' section with an 'Aggiungi prodotto' button. The main content area displays '7 prodotti' with two filter buttons: 'Caffetteria (4)' and 'Food (3)'. A list of seven products is shown, each with a small icon, the product name, the price in Euros, and a right-pointing arrow.

Icona	Nome Prodotto	Prezzo	Azione
Caffè	Caffè espresso	0.80 €	>
Caffè	Caffè macchiato	0.90 €	>
Cappuccino	Cappuccino	1.00 €	>
Mocaccino	Mocaccino	1.10 €	>
Mini San Carlo	Mini San Carlo 1936 25g	1.00 €	>
Mini San Carlo	Mini San Carlo Bacon 25g	1.10 €	>
Mini San Carlo	Mini San Carlo Paprika 25g	1.10 €	>

Figura 5.15: Tavola raffigurante la pagina contenente i dettagli dei prodotti piattaforma

5.2.2.4 Utenti

La pagina nominata **Utenti**, permette all'utente amministratore di visualizzare tutti gli utenti presenti nel sistema.

L'obiettivo è quello di consentire all'amministratore di poter aggiungere e/o modificare gli utenti presenti nella piattaforma. Ogni utente appartiene ad una riga di una tabella. Ogni riga è rappresentata sotto forma di *card* che porta ad una nuova pagina in cui è possibile aggiornare i dati dell'utente.

I bottoni presenti in questa pagina consentono di:

- Aggiungere un nuovo utente;
- Filtrare gli utenti in base alla loro categoria di appartenenza (utenti base/dipendenti).

La Figura 5.16 che rappresenta graficamente quanto espresso in precedenza.

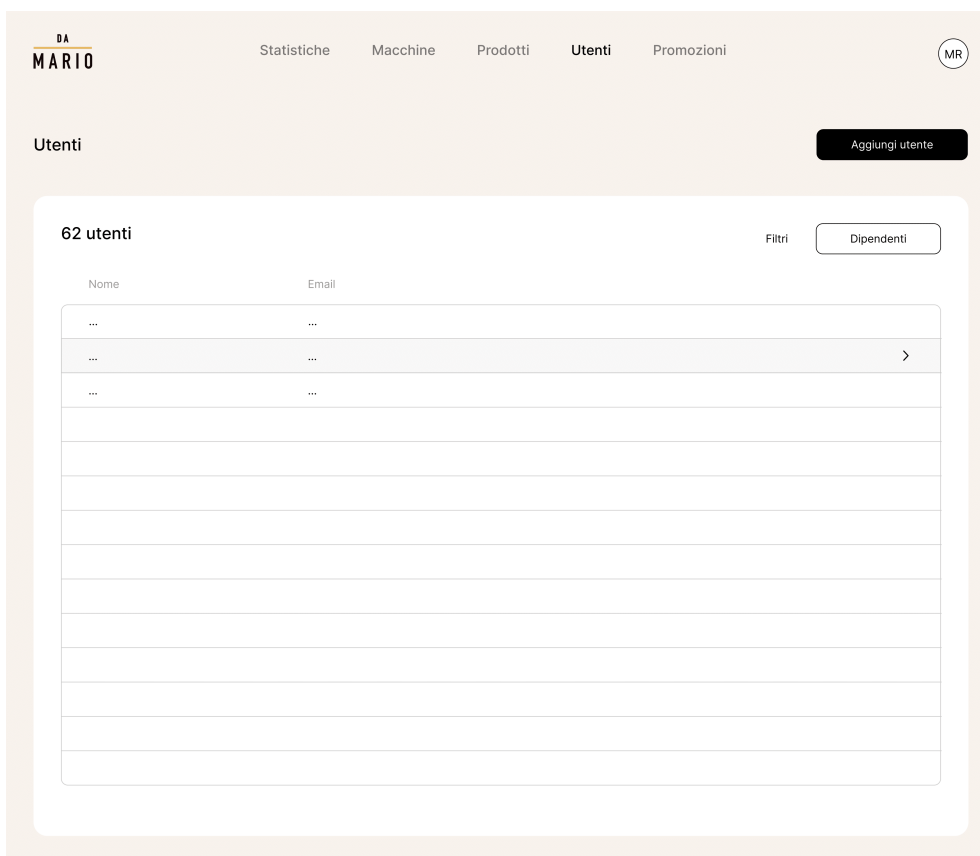


Figura 5.16: Tavola raffigurante la pagina contenente gli utenti del sistema

5.2.2.5 Promozioni

La pagina nominata **Promozioni**, permette all'utente amministratore di visualizzare tutte le promozioni presenti nel sistema.

L'obiettivo è quello di consentire all'amministratore di poter aggiungere e/o modificare le promozioni presenti nella piattaforma. Le promozioni hanno un'etichetta che rappresenta le categorie di utenti a cui sono destinate, il prodotto, la data di validità, la *location* ed il prezzo scontato.

I bottoni presenti in questa pagina consentono di:

- Aggiungere una nuova promozione;
- Filtrare le promozioni in base al loro status (attiva/disattivata).

La Figura 5.17 che rappresenta graficamente quanto espresso in precedenza.

The screenshot shows the 'Promozioni' page in the DA MARIO system. The page has a navigation bar with 'Statistiche', 'Macchine', 'Prodotti', 'Utenti', and 'Promozioni'. There is a 'DA MARIO' logo and a user profile icon 'MR'. A button 'Aggiungi promozione' is visible. The main content area shows '2 promozioni' and a 'Filtri' button with 'Attive' selected. The table below lists the promotions:

Utenti	Prodotto	Validità	Location	Prezzo
Dipendenti	Caffè espresso	—	Tutte	0.72€ 0.80€
Dipendenti Ospiti	Caffè macchiato	01/09/22 — 31/11/22	Tutte	0.80€ 0.90€
Dipendenti	Caffè espresso	01/11/22 — 15/11/22	1 location	0.72€ 0.80€

Figura 5.17: Tavola raffigurante la pagina contenente le promozioni del sistema

Ogni riga è rappresentata sotto forma di *card* al click si espande e presenta tutti i parametri configurabili .

The screenshot displays the 'Promozioni' management interface. At the top, there's a navigation bar with 'Statistiche', 'Macchine', 'Prodotti', 'Utenti', and 'Promozioni'. A user profile icon 'MR' is in the top right. Below the navigation, the page title 'Promozioni' is shown next to an 'Aggiungi promozione' button. The main content area shows '2 promozioni' with an 'Attive' status indicator. A table lists the promotions:

Utenti	Prodotto	Validità	Location	Prezzo
Dipendenti	Caffè espresso	—	Tutte	0.72€ (0.80€)
Dipendenti	Ospiti	01/09/22 — 31/11/22	Tutte	0.80€ (0.90€)

The configuration form for the first promotion includes:

- Options: Dipendenti, Coworking
- Prodotto: Caffè espresso
- Scegli una data: 01/12/2022 — 31/12/2022
- Options: Location A, Location B
- Tipologia di sconto: Fisso
- Prezzo base: 1.10, Prezzo scontato: 1.00

Figura 5.18: Tavola raffigurante la pagina contenente la configurazione delle promozioni del sistema

5.3 Codifica

5.3.1 *Design Pattern* utilizzati

Per lo sviluppo del progetto di stage, sono stati utilizzati vari *Design Pattern*. Alcuni di questi *pattern* vengono utilizzati intrinsecamente dal *framework*, mentre altri sono stati appositamente adottati per risolvere alcuni tipi di problematiche.

5.3.1.1 *Dependency Injection*

Con lo sviluppo di parti piccole del sistema come possono essere dei moduli o dei componenti, è possibile che sia necessario chiamare delle variabili o delle funzioni

provenienti da altri moduli o componenti.

Per risolvere questo tipo di problematica, è stato utilizzato il *pattern Dependency Injection*. Questo *pattern* permette di iniettare le dipendenze di un componente, in modo da poterle utilizzare all'interno di esso.

Solitamente in *Angular* viene utilizzato all'interno del costruttore di un componente così da aumentare la modularità e la flessibilità dei componenti stessi. In *Angular* vengono per lo più iniettati servizi, ma possono anche essere iniettate variabili. Tali dipendenze vengono istanziate solo quando necessario.²

5.3.1.2 Decorator

Il principio alla base del *pattern Decorator* in *Angular* è quello di memorizzare tutti i metadati relativi ad una classe o ad un metodo.

Come definiscono gli autori di “*Design Patterns - Elementi per il riuso di software a oggetti*”, le idee alla base del *pattern Decorator* sono:

- Si vuole poter aggiungere responsabilità a singoli oggetti dinamicamente ed in modo trasparente, cioè senza coinvolgere altri oggetti;
- Si vuole poter togliere responsabilità agli oggetti;
- L'estensione attraverso la definizione di sottoclassi non è praticabile. [13]

Il *pattern Decorator* è comunemente utilizzato all'interno del progetto: un esempio pratico è rappresentato dalle direttive `@Input()` e `@Output()` che vengono utilizzate per stabilire la comunicazione tra componenti genitori e componenti figli.

La direttiva `@Input()` viene utilizzata nel progetto per passare dati dal componente genitore a un componente figlio. Questo consente al componente genitore di controllare il comportamento o l'aspetto del componente figlio.

La direttiva `@Output()` invece viene utilizzata nel progetto per creare un'interfaccia di comunicazione dal componente figlio al componente genitore. Questo permette al componente figlio di notificare il componente genitore di determinate azioni o cambiamenti.

5.3.1.3 Observer

Quando parliamo di *Observer*, indichiamo un *Design Pattern* comportamentale che viene ampiamente utilizzato nel contesto di *Angular*.

Questo *pattern* infatti favorisce una comunicazione efficiente fra componenti, in quanto definisce un modo flessibile per poter gestire gli aggiornamenti e le notifiche.

In un progetto di *Angular*, il *pattern* Observer può essere applicato per gestire l'interazione tra i componenti, consentendo loro di osservare e rispondere ai cambiamenti di stato o di dati di altri componenti.

È lo stesso *framework* che fornisce la classe `EventEmitter` che viene solitamente utilizzato per pubblicare dei valori fra componenti.

²*Angular - Dependency Injection*. URL: <https://v15.angular.io/guide/dependency-injection-overview>.

5.3.1.4 *Singleton*

Il *Singleton* è un *Design pattern* che consente di avere una sola istanza di un componente all'interno dell'applicazione. Come riporta la documentazione di [Angular](#), esistono due modi per poter rendere un servizio un *Singleton*:

- il primo consiste nell'impostare la proprietà `providedIn` al valore `root` di `@Injectable()`;
- il secondo consiste nell'includere il servizio all'interno dell'`AppModule` oppure di includerlo in un modulo e importare il modulo nell'`AppModule`.³

Un esempio di Singleton lo fornisce la documentazione:

```
import { Injectable } from '@angular/core';

@Injectable({
  providedIn: 'root',
})
export class UserService {
}
```

5.3.1.5 *Strategy*

Nel caso specifico di questo progetto, l'uso del *pattern Strategy*, mi ha consentito di evitare la distruzione di un componente, non appena veniva cambiata la rotta. La scelta è stata determinata principalmente dalla necessità di conservare i dati del componente quando si fa *click* su una *card*, oltre a migliorare le prestazioni del sito.

[Angular](#) infatti mette a disposizione la classe `BaseRouteReuseStrategy` che fornisce le funzioni di base per poter riutilizzare un componente. Nel mio progetto ho creato un `CustomRouteReuseStrategy` che gestisse il riutilizzo del componente secondo le mie necessità, modificandone il comportamento. Il *pattern strategy* permette infatti di definire molteplici algoritmi con diverso comportamento, evitando le scelte condizionali multiple.

5.3.2 *Struttura software*

5.3.2.1 *Componenti sviluppate*

Le componenti sviluppate sono:

- **Dashboard:** Componente per la visualizzazione delle statistiche;
- **Devices:** Componente per la gestione delle *location* e delle macchinette;
- **Header:** Componente che consente all'utente amministratore di selezionare la pagina da visualizzare dalla barra di navigazione, oppure di visualizzare le notifiche;
- **Loading:** Componente utilizzata per far visualizzare all'utente un caricamento in corso;
- **Login:** Componente per permettere all'utente amministratore di inserire le proprie credenziali per accreditarsi alla piattaforma;

³*Angular - Singleton*. URL: <https://v15.angular.io/guide/singleton-services>.

- **Logout:** Componente che consente all'utente amministratore di uscire dalla piattaforma e dal proprio profilo;
- **Platform-Product:** Componente che permette all'utente amministratore di gestire la configurazione dei prodotti piattaforma;
- **Product:** Componente che permette all'utente amministratore di gestire la configurazione dei prodotti macchinetta.

5.3.2.2 Moduli sviluppati

I moduli sviluppati sono:

- **Admin:** Gestisce il *routing* dell'utente, ne verifica l'accreditamento e riporta alla pagina di *login*;
- **Dashboard:** Gestisce le statistiche;
- **Devices:** Gestisce le *location* e le macchinette;
- **Header:** Gestisce il *routing* della barra di navigazione;
- **Loading:** Gestisce la componente che rende visibile il caricamento dei dati;
- **Login:** Gestisce il *token* di sessione dell'utente amministratore;
- **Logout:** Gestisce la distruzione del *token* di sessione dell'utente amministratore;
- **Platform-Product:** Gestisce i prodotti piattaforma;
- **Product:** Gestisce la configurazione dei prodotti macchinetta.

5.3.2.3 Servizi sviluppati

I servizi sviluppati sono:

- **Admin:** Gestisce l'*header* dell'utente amministratore;
- **Dashboard:** Gestisce la comunicazione con il *backend* e richiede le informazioni relative alle statistiche;
- **Devices:** Gestisce la comunicazione con il *backend* e richiede le informazioni relative alle *location* e alle macchinette;
- **Platform-Product:** Gestisce la comunicazione con il *backend* e richiede le informazioni relative ai prodotti piattaforma;

5.3.3 Interfacce realizzate

5.3.3.1 Interfaccia di login

La prima pagina che ho realizzato durante il mio percorso di stage, è stata l'interfaccia di *login*. Essendo uno dei requisiti obbligatori, mi ha consentito di familiarizzare con il *framework* e con la gestione dei *form* in *Angular* (Figura 5.19).

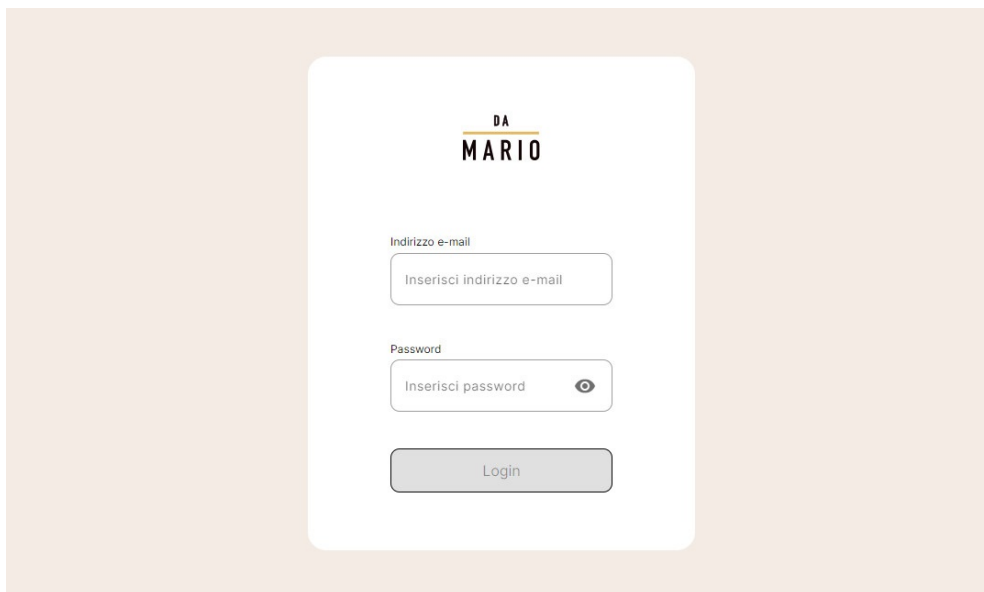


Figura 5.19: Screenshot che rappresenta l'interfaccia di *login*

Essendo l'interfaccia grafica composta da due campi necessari per effettuare il login, ho verificato che:

- i dati inseriti nei campi fossero validi (verificati tramite un *custom ErrorStateMatcher*);
- i dati siano inseriti nei campi.

La verifica dei dati viene fatta solamente lato *backend* in quanto per ragioni di sicurezza è consigliato evitare di far gestire le verifiche di autenticazione lato *frontend*.

Quando i dati risultano validi, è possibile cliccare il bottone per il tentativo di autenticazione alla piattaforma. Se il *login* risulta andato a buon fine, verrà istanziato un *token*.

Questo *token* verrà gestito direttamente da un servizio presente all'interno del progetto che manterrà viva la sessione, finché l'utente non effettua il *logout* dalla piattaforma.

5.3.3.2 Interfaccia di navigazione

L'interfaccia di navigazione è stata realizzata per permettere all'utente amministratore di poter navigare all'interno della piattaforma (Figura 5.20).

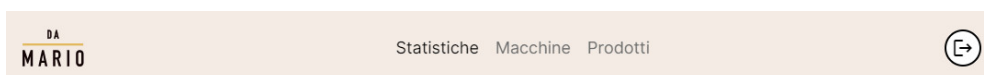


Figura 5.20: Screenshot che rappresenta l'interfaccia di navigazione

Il modulo relativo a questo componente viene caricato solo all'occorrenza. All'avvio dell'applicazione infatti, l'utente amministratore visualizzerà la schermata dedicata al login, senza alcun menù.

5.3.3.3 Interfaccia "Statistiche"

La realizzazione della pagina contenente le statistiche ha richiesto la maggior parte del tempo in quanto rappresentava l'obiettivo primario da completare durante questo stage. In aggiunta, la necessità del team era quella di poter mostrare al cliente una prima versione del prodotto.

La caratteristica principale di questa schermata è determinata dalla complessità della gestione dei dati presenti in essa.

In questa schermata è possibile visualizzare tutte le statistiche che sono state pensate dal team e discusse in precedenza nei casi d'uso e nella sezione relativa alla progettazione.

Un primo esempio di come si visualizza la schermata principale della pagina è rappresentato nella Figura 5.21 in cui è presente la visualizzazione delle statistiche relative alle vendite totali.

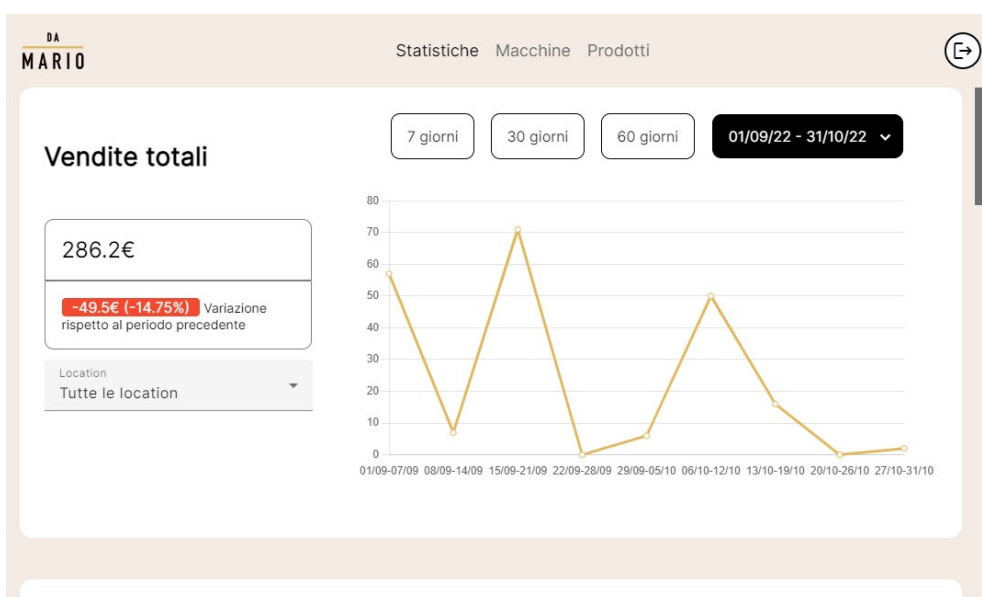


Figura 5.21: Screenshot che rappresenta le statistiche relative alle vendite totali

Seguono gli screenshot relativi alle statistiche di:

- Prodotti più venduti (Figura 5.22);
- Numero di utenti (Figura 5.23);
- Spesa media utenti (Figura 5.24);
- Spesa per fascia oraria (Figura 5.25).

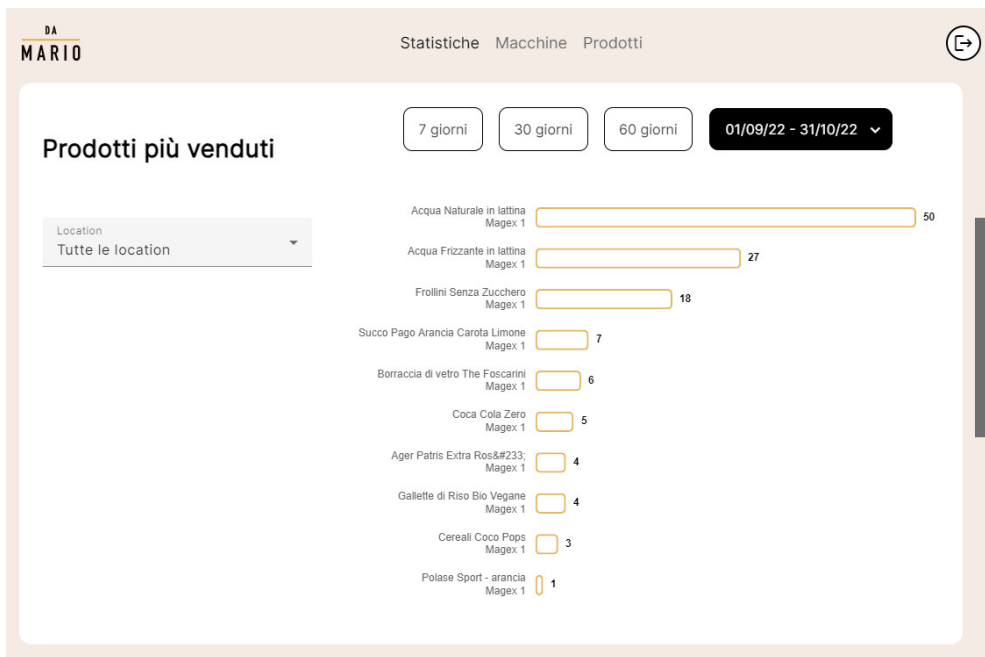


Figura 5.22: Screenshot che rappresenta le statistiche relative ai prodotti più venduti

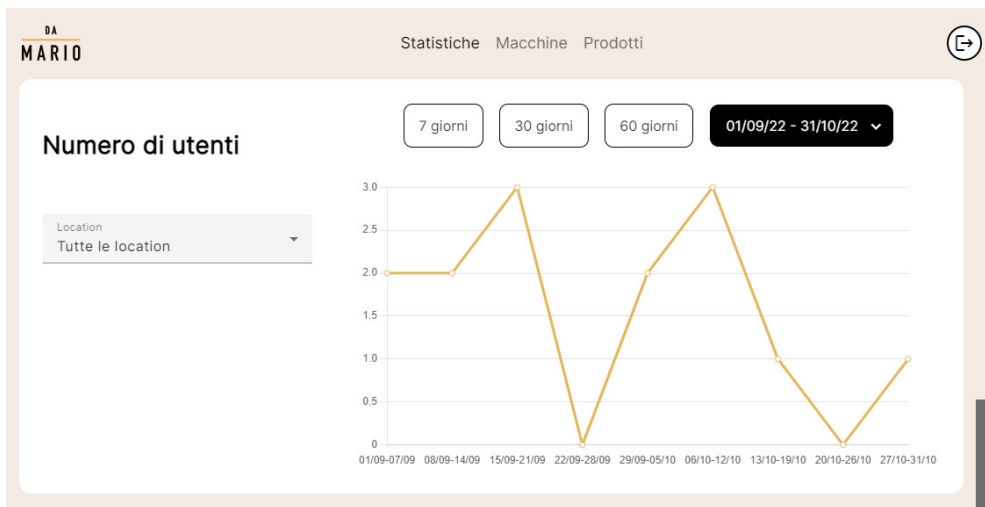


Figura 5.23: Screenshot che rappresenta le statistiche relative al numero di utenti

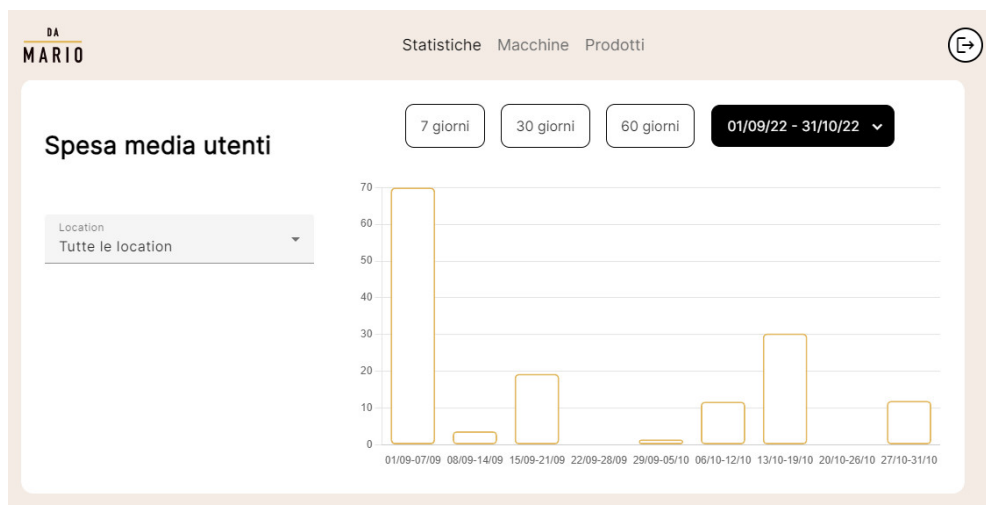


Figura 5.24: Screenshot che rappresenta le statistiche relative alla spesa media degli utenti

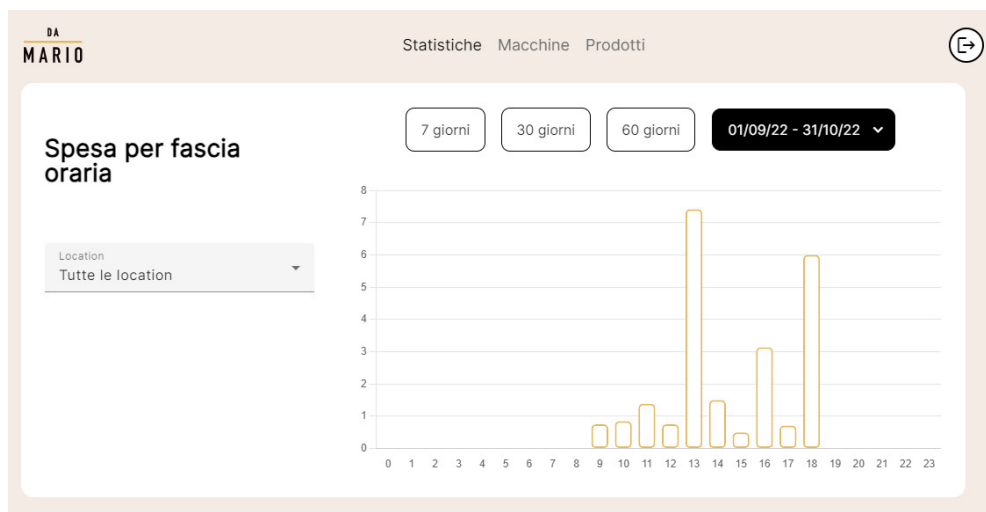


Figura 5.25: Screenshot che rappresenta le statistiche relative spesa per fascia oraria

5.3.3.4 Interfaccia "Macchinette"

La pagina legata alla visualizzazione, alla modifica e all'aggiunta di *location* e macchinette risulta incompleta a causa del tempo residuo contenuto. Dei requisiti che sono stati definiti per questa pagina, ho completato le seguenti funzionalità:

- Visualizzazione delle *location* con le relative macchinette (Figura 5.26);
- Visualizzazione delle macchinette con la lista dei prodotti (Figura 5.27);
- *Routing* verso la pagina precedente (da macchinetta a *location*).

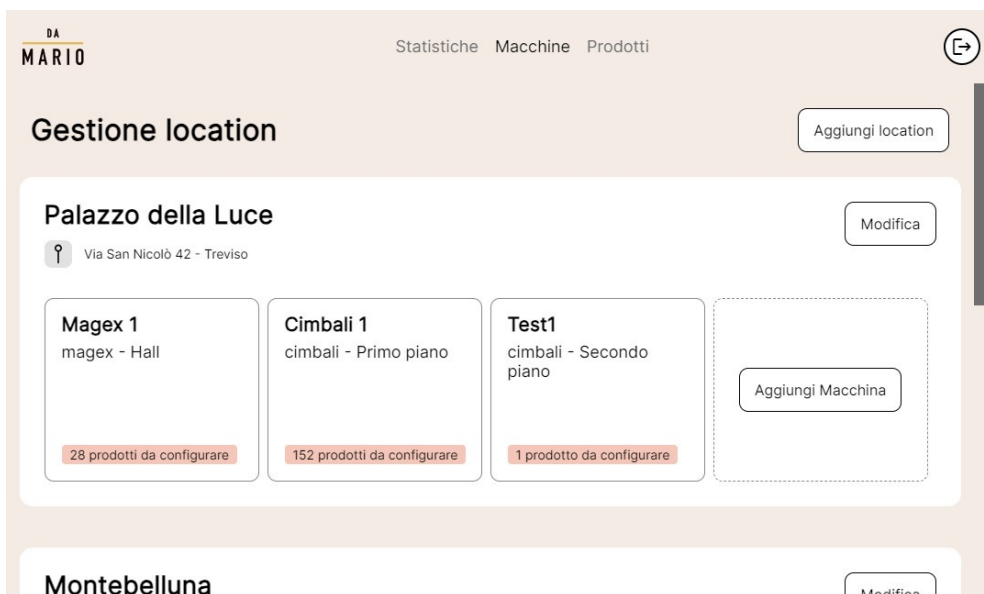


Figura 5.26: Screenshot che rappresenta l'interfaccia relativa alle location

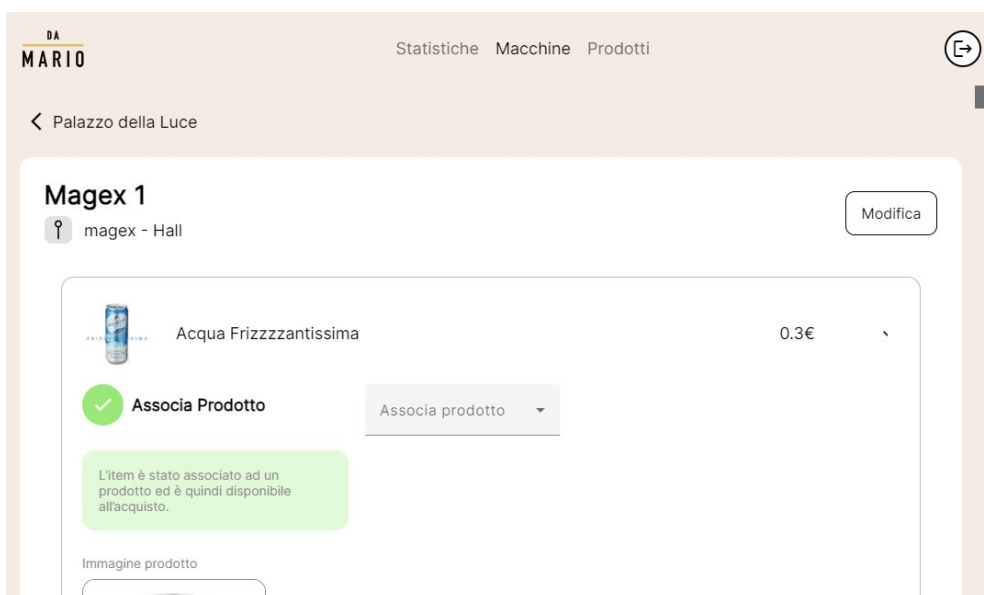


Figura 5.27: Screenshot che rappresenta l'interfaccia relativa alla specifica macchinetta

Pertanto in questa sezione non sono presenti le funzionalità di aggiunta e modifica.

5.3.4 Pubblicazione del sito

Verso il termine dello stage, la *web app* è stata pubblicata.

In particolare nella prima versione del prodotto sono state incluse la pagina di *login* e la pagina delle statistiche.

La piattaforma su cui è stato pubblicato il sito è *Netlify*: *Netlify* è una piattaforma di *hosting* e *deployment* che offre un'infrastruttura agile e performante per applicazioni *web*. Questa applicazione però non è ancora ufficialmente a disposizione dell'azienda, in quanto è in *staging*. Il prodotto è rimasto in *staging* fino al completamento del mio stage, in quanto le modifiche che sono state attuate al *backend* non sono ancora definitive, pertanto è stato impossibile passare allo status di produzione.

Inoltre, nell'ambito del processo di *deployment*, è stato utilizzato anche *Airbrake*. *Airbrake* è un servizio di monitoraggio degli errori e di notifica delle anomalie che consente di identificare e risolvere prontamente eventuali problemi o malfunzionamenti dell'applicazione. L'adozione di *Airbrake* all'interno dell'applicazione ha permesso di rilevare errori durante la compilazione del progetto, ma anche durante la fase di *deploy*. *Airbrake* infatti consente di tracciare tutti gli errori runtime, inviando delle notifiche qualora si presentassero delle anomalie nel sistema, con conseguente miglioramento dell'affidabilità del prodotto e garantendo una maggiore sicurezza del prodotto.

Capitolo 6

Verifica e validazione

6.1 Verifica

Il processo di verifica è un processo che ha nota importanza nell'ambiente dello sviluppo *software*. Per motivi legati principalmente al tempo residuo al termine delle settimane di stage, ho dovuto concentrare la mia attenzione in particolar modo sull'analisi di tipo statico del prodotto che ho realizzato e sui test legati al *browser*.

6.1.1 Lighthouse

Lighthouse è uno strumento di analisi e di miglioramento della qualità delle pagine *web*. Questo *tool* esegue una serie di test sulle pagine *web*, e genera un *report* sulle prestazioni, l'accessibilità, le *best practice* e l'indicizzazione da parte dei motori di ricerca (SEO). *Lighthouse* è stato integrato nei sistemi *Chromium* e può essere eseguito da *DevTools*. Questo strumento è stato utilizzato per valutare la qualità del prodotto realizzato e per migliorare le prestazioni del sito *web*.

Non avendo avuto a disposizione molto tempo da dedicare al miglioramento del sistema, possiamo dire che i risultati del *report* sono soddisfacenti, come si deduce dalla Figura 6.1.

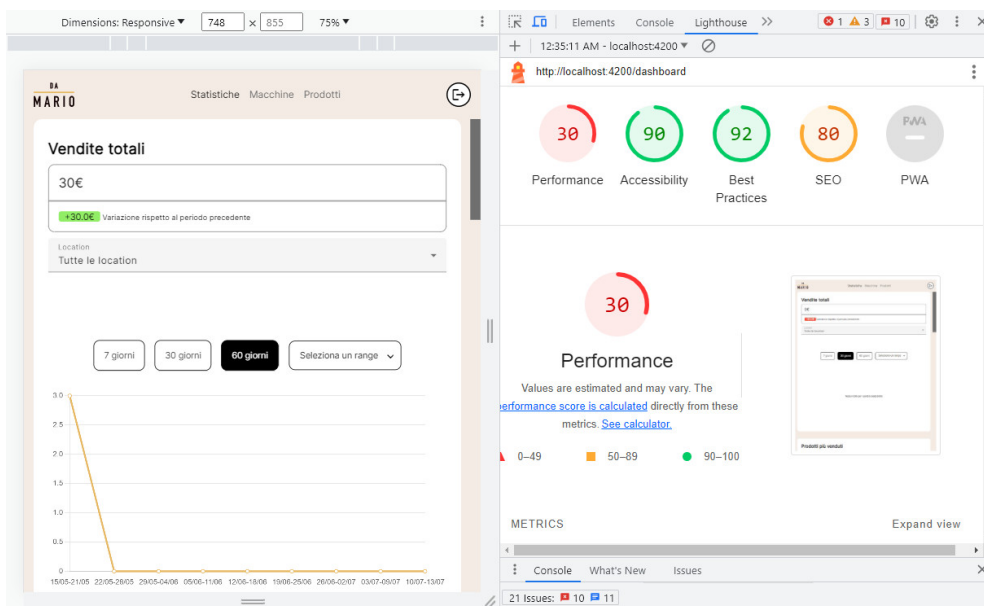


Figura 6.1: Risultati del report di Lighthouse

I test automatici però evidenziano che esiste una criticità per quanto concerne le *performance* della piattaforma.

La causa principale è dovuta alla presenza di *media* particolarmente pesanti e all'utilizzo massivo del linguaggio *TypeScript*. Il calcolo dovuto alle funzioni eseguite dal *framework Angular* e la necessità di attendere i dati richiesti a *backend* determinano un notevole rallentamento del *rendering* delle pagine del sistema.

Nel contesto del presente progetto, l'ottimizzazione per i motori di ricerca (SEO) non rappresenta un parametro particolarmente rilevante. Questa affermazione è giustificata dal fatto che la piattaforma sarà utilizzata **esclusivamente** dagli utenti amministratori del sistema. Non è richiesta pertanto una presenza prominente tra i primi risultati delle ricerche sui principali motori di ricerca.

L'obiettivo principale del progetto, infatti, è di fornire un'interfaccia *user-friendly* e intuitiva per utenti amministratori con differenti conoscenze informatiche che ponga particolare attenzione all'usabilità e alle funzionalità specifiche richieste dal committente.

Gli sviluppi futuri della piattaforma potrebbero per esempio integrare delle chiamate a *backend* più semplici con una raccolta minore di dati, una semplificazione dell'interfaccia e un *rendering* anticipato delle pagine che l'utente visualizza.

Si potrebbe inoltre diminuire il numero delle chiamate a *backend* favorendo il riuso dei componenti, come spiegato nel capitolo precedente.

6.1.2 Browser testati

Per quanto riguarda i test legati al browser, ho testato il prodotto su *Google Chrome*, *Microsoft Edge*, *Apple Safari* e *Mozilla Firefox*.

Il prodotto è stato testato anche su *tablet*, in particolare su *iPad*.

Non sono state riscontrate particolari criticità di presentazione dei dati. Segue una lista dei browser testati:

- *Chrome*, versione 110;
- *Firefox*, versione 105;
- *Safari*, versione 16.4;
- *Edge*, versione 114.

6.1.3 Dispositivi testati

Il sito nasce per essere utilizzato con dispositivi dallo schermo oltre i 1024px. Inizialmente pertanto le *media query* implementate coprivano lo spettro dei dispositivi dai *tablet* fino ai *computer* di fascia elevata (schermi in 4K).

Pur non essendo la *responsiveness* per *smartphone* un requisito obbligatorio, ho implementato una prima versione base di visualizzazione dei contenuti. Risulta però complesso gestire una mole di dati così elevata e rendere facilmente consultabili i grafici per uno schermo di dimensioni ridotte.

Si consiglia agli utenti che volessero testare il prodotto con il proprio cellulare di tenerlo in modalità *landscape* per migliorare la comprensione del contenuto della pagina delle statistiche. Segue una lista dei dispositivi testati:

- *PC* (Windows, Mac) 16", Risoluzione: 3K pixel;
- *Laptop* (Windows, Mac) 13/14", Risoluzione: 1920 x 1080/4K pixel;
- *iPad Air*, Risoluzione 1536 x 2048 pixel;
- *iPhone XS*, Risoluzione 2436 x 1125 pixel;
- *iPhone 6*, Risoluzione 1334 x 750 pixel.

6.1.4 Analisi statica

L'analisi statica è una tecnica che consiste principalmente nella lettura del codice stesso. Non è pertanto necessario eseguire il codice per verificarlo. Durante tutto il periodo di codifica, ho svolto l'analisi statica del codice adottando diverse modalità:

- Adottando walkthrough;
- Effettuando inspection;
- Rileggendo il codice scritto;
- Verificando gli errori segnalati dall'IDE e dal compilatore.

Per favorire una più attenta verifica che il codice non presentasse dei *bug*, con il team di sviluppo abbiamo adottato uno strumento che consente di effettuare l'analisi statica del codice chiamato *ESLint*. Lo scopo principale di *ESLint* è quello di fornire agli sviluppatori uno strumento automatizzato per il controllo della qualità del codice, contribuendo a migliorare l'efficienza e l'affidabilità del processo di sviluppo. Attraverso una serie di regole configurabili, *ESLint* analizza il codice sorgente, individua le possibili violazioni delle *best practice* di programmazione e genera avvisi o errori, consentendo così agli sviluppatori di correggere e ottimizzare il codice.

6.1.5 Analisi dinamica

L'analisi dinamica richiede l'esecuzione del *software*. Tuttavia, a causa di limiti temporali, non ho avuto la possibilità di dedicare il tempo necessario per la scrittura completa dei test. Il prodotto che ho realizzato infatti non è mai stato eseguito nell'ambiente di produzione. La verifica che ho effettuato è stata pertanto incentrata sul controllo del flusso degli eventi scatenati durante la navigazione nella piattaforma. Alcuni colleghi che non avevano mai visto il progetto, si sono immedesimati in utenti tipo del prodotto e hanno effettuato degli *stress test* per verificare che non ci fossero delle anomalie nel prodotto.

6.2 Validazione

Il processo di validazione in un progetto di un sito *web* mira a garantire che il prodotto finale soddisfi i requisiti stabiliti e che sia privo di errori o difetti. La validazione consiste nel verificare che il sito *web* si comporti come previsto, rispettando le specifiche funzionali e di design.

Per quanto detto in precedenza il tempo residuo da poter dedicare a questi due processi è risultato insufficiente.

Ho però portato a termine dei test unitari e funzionali inerenti l'interfaccia di *login*. Ho così verificato che le funzioni implementate fossero corrette e che rispettassero i requisiti.

Siccome il prodotto non è ancora ultimato, la validazione è stata principalmente condotta internamente.

Durante il mio periodo di stage non ho avuto modo di mostrare al cliente quanto realizzato, pertanto non mi è stato possibile effettuare alcuna validazione esterna (ossia con il cliente stesso e/o gli utenti finali della piattaforma).

Capitolo 7

Conclusioni

L'obiettivo principale da raggiungere era quello di sviluppare delle interfacce di una *web app* che permettesse agli amministratori del progetto **Da Mario** di accedere alla *dashboard* principale e visualizzare le statistiche relative all'andamento delle vendite suddivise per *location*, macchinette ed eventualmente prodotto. Successivamente è stata introdotta la possibilità di configurare i prodotti legando i prodotti macchinetta con quelli piattaforma.

Al termine delle mie otto settimane di stage, il prodotto che ho realizzato è stato ritenuto soddisfacente dall'azienda ospitante. Allo stato attuale la *web app* è funzionante, ed è pronta per essere utilizzata in un contesto reale, con i dati del *backend* di produzione. Il *software* ottenuto è stato messo a disposizione del cliente, momentaneamente in fase di *staging*, ma in una prima versione contenente tutte le statistiche richieste.

Le prime settimane di stage sono state dedicate alla comprensione del sistema, degli obiettivi e all'analisi dei requisiti.

L'assenza totale del cliente, ha reso più complesso il processo di analisi, in quanto i requisiti e i casi d'uso del progetto sono stati estratti esclusivamente dal *wireframe* che mi è stato messo a disposizione.

In questo caso la colpa è da imputare al committente che ha posticipato ripetutamente gli appuntamenti relativi alla discussione del prodotto da realizzare. Non ho potuto pertanto ricevere un vero e proprio *feedback* sulla correttezza dell'analisi, oltre a quella che mi ha fornito il tutor (responsabile del prodotto) che ha dato parere positivo.

In queste prime settimane non si sono verificate particolari problematiche, in quanto gli obiettivi settimanali stilati dal mio tutor sono sempre stati raggiunti in anticipo e con profitto.

La fase di progettazione ha richiesto un impegno minore, sia in termini di lavoro che di fatica. Come detto in precedenza, le tavole del prodotto erano già state progettate prima del mio inserimento nel progetto, pertanto il mio lavoro è stato per lo più legato alla progettazione dell'architettura del prodotto e alla valutazione di fattibilità delle proposte definite nelle tavole su *Figma*.

Durante la fase di sviluppo del prodotto, si sono verificati dei rallentamenti, dovuti in parte all'aggiornamento dei requisiti. Questo è stato in parte legato al fatto di aver compreso le necessità del sistema man mano che il prodotto veniva sviluppato. Di conseguenza sono dovuto intervenire direttamente sul codice, modificando parti già consolidate del progetto.

Di conseguenza il poco tempo residuo è stato suddiviso e impiegato nelle le attività rimanenti di testing e documentazione del prodotto.

7.1 Consuntivo finale

Viene riportata di seguito la tabella riassuntiva delle ore impiegato nelle varie attività dello stage:

Attività	Ore di lavoro
Comprensione sistema e obiettivi	20
Analisi dei requisiti	30
Progettazione	40
Studio e setup ambiente di sviluppo	40
Implementazione	150
Test e validazione	10
Documentazione	10
<i>Totale</i>	<i>300</i>

Tabella 7.1: Tabella con il consuntivo delle ore utilizzate durante il progetto

La prima settimana si è concentrata sull'apprendimento e sulla comprensione del progetto mentre la settimana successiva ha avuto come obiettivo di iniziare l'analisi dei requisiti. Rispetto al piano di massima sono servite più ore per l'analisi e meno per la progettazione in quanto le tavole del prodotto erano già state anticipatamente realizzate.

Sono risultate necessarie un maggior numero di ore per lo studio ed il *setup* dell'ambiente a causa di alcune problematiche verificatesi in corso d'opera.

Di conseguenza le ore ridotte residue al termine la fase di sviluppo, sono state suddivise fra *testing* e documentazione.

7.2 Raggiungimento degli obiettivi

Conclusa la fase di sviluppo, gli che sono stati soddisfatti sono i seguenti:

Codice	Descrizione	Stato
O01	Acquisire competenze nello sviluppo <i>web</i> con <i>TypeScript</i>	Soddisfatto
O02	Acquisire competenze nello sviluppo <i>web</i> con <i>Angular</i>	Soddisfatto
O03	Sviluppare la pagina di <i>login</i> alla piattaforma di amministrazione	Soddisfatto
O04	Sviluppare il <i>logout</i> dalla piattaforma di amministrazione	Soddisfatto
O05	Sviluppare le rotte che aggiornano la pagina principale in cui l'utente viene reindirizzato dopo l'autenticazione	Soddisfatto
O06	Sviluppare la pagina Statistiche della piattaforma	Soddisfatto

Tabella 7.2: Tabella riassuntiva degli obiettivi soddisfatti

Dato il tempo residuo limitato, ho iniziato a sviluppare alcuni dei requisiti desiderabili e alcuni dei requisiti facoltativi, nello specifico:

Codice	Descrizione	Stato
D01	Sviluppare la pagina Prodotti della piattaforma	Soddisfatto (parzialmente)
F01	Sviluppare la pagina Macchine della piattaforma	Soddisfatto (parzialmente)

Tabella 7.3: Tabella riassuntiva degli obiettivi parzialmente soddisfatti

Non è stato possibile invece soddisfare l'obiettivo che segue, in quanto non erano disponibili le *mutation* a *backend* per poter aggiornare i dati dell'utente amministratore:

Codice	Descrizione	Stato
O07	Sviluppare il cambio <i>password</i>	Non soddisfatto

Tabella 7.4: Tabella contenente l'obiettivo obbligatorio non soddisfatto

7.3 Conoscenze acquisite

Durante il periodo di stage ho appreso molteplici tecnologie che mi torneranno utili anche in futuro, visto il mio interesse per l'ambiente del *frontend*.

L'analisi dei requisiti e la fase di progettazione sono molto dettagliate in quanto ho astratto tutte le necessità espresse dal cliente, svolgendo in parte anche una funzione di analista. Queste attività mi hanno consentito di migliorare la mia capacità di astrarre dei requisiti lato sviluppatore a partire da requisiti lato utente.

Ho avuto modo di essere seguito da esperti nel settore del *frontend*, pertanto sono stato aiutato ad affinare il mio ragionamento nello sviluppo *software*: l'esperienza e le conoscenze dei miei colleghi, mi hanno consentito di risolvere determinati problemi che ho affrontato durante il mio percorso di stage e di vedere delle soluzioni alternative più avanzate.

Ho maturato esperienza nello sviluppo *web* grazie al *framework Angular* e tramite il linguaggio *TypeScript*.

Ho migliorato la mia attitudine al lavoro in team, in quanto il contesto aziendale in cui sono stato inserito, è altamente collaborativo e basato su processi di miglioramento continuo e di scambio reciproco di conoscenze. Mediante il confronto tra le esperienze accademiche precedentemente affrontate e quella derivante dall'esperienza di stage, ho avuto l'opportunità di immergermi in una realtà progettuale caratterizzata da un considerevole aumento del numero di partecipanti, i quali presentavano conoscenze e competenze altamente diversificate tra di loro.

Il mio progetto ha consentito all'azienda di esplorare *Angular 15* e fungere così da pioniere per le novità introdotte da questa versione.

L'esperienza lavorativa ha anche influenzato in positivo le mie capacità di raggiungere obiettivi entro le scadenze fissate.

7.4 Valutazione personale

La motivazione alla base della mia decisione di intraprendere lo stage presso *Moku S.r.l* è derivata da testimonianze positive fornite da alcuni miei conoscenti, dalla presenza di un ambiente lavorativo aperto e sereno e soprattutto dall'interesse suscitato dal progetto che mi è stato presentato durante l'evento *Stage-IT*. Questa prima impressione ha trovato conferma anche durante il mio percorso di stage.

Questa esperienza si è rivelata infatti altamente formativa, in quanto mi ha permesso di migliorare le mie capacità di programmazione lato *web*, ma anche perché mi ha consentito di dare un primo sguardo verso il mondo lavorativo.

Sono soddisfatto del lavoro svolto, in quanto ho raggiunto gli obiettivi fissati e ho avuto modo di lavorare con tecnologie che non avevo mai utilizzato prima. La sperimentazione sul campo ha aumentato il mio interesse per l'ambiente di *frontend* e sicuramente continuerò a documentarmi a riguardo.

Indubbiamente rispetto al mondo accademico, l'ambiente lavorativo risulta più complesso, ma anche più dinamico, in quanto è necessario essere reattivi e flessibili di fronte alle problematiche di sviluppo. Di conseguenza è necessario essere anche proattivi al cambiamento. Le competenze acquisite durante il percorso accademico hanno svolto un ruolo cruciale nel tradurre i requisiti in un sistema effettivamente fruibile e funzionale. Infine, mi ritengo totalmente soddisfatto di questo percorso di stage, in quanto ho avuto modo di lavorare in un ambiente collaborativo e stimolante, che mi ha permesso di crescere dal punto di vista professionale. È stata una soddisfazione personale, poter tradurre il teorico in un prodotto realmente utilizzabile.

Acronimi e abbreviazioni

API Application Program Interface. 6, 16, 17, 29, 105–107, 110

DOM Document Object Model. 68, 106

FE Frontend. 6, 20, 107

HTML Hyper Text Markup Language. 6, 30, 67, 68, 107

IDE Integrated Development Environment. 21, 30, 107

MQTT Message Queuing Telemetry Transport. 16, 17, 108

UX user experience. 1, 3, 66, 110

Glossario

Angular Angular è una piattaforma di sviluppo scritta in linguaggio *TypeScript*. È composta da piccoli sottosistemi, tra cui un framework *JavaScript*, un'interfaccia a riga di comando, un servizio linguistico e una ricca collezione di librerie di terze parti. [iii](#), [ix](#), [6](#), [28](#), [29](#), [65](#), [67–70](#), [87–89](#), [97](#), [101](#), [102](#)

Application Program Interface In italiano "interfaccia di programmazione di un'applicazione" indica ogni insieme di procedure disponibili al programmatore, di solito raggruppate a formare un *set* di strumenti specifici per l'espletamento di un determinato compito all'interno di un certo programma. La finalità è ottenere un'astrazione, di solito tra l'*hardware* e il programmatore o tra *software* a basso e quello ad alto livello semplificando così il lavoro di programmazione. [104](#)

attività Secondo lo standard [ISO/IEC 12207:1995](#), un'attività di ingegneria del *software* è definita come un'operazione, un'insieme di operazioni o una fase che è necessaria per soddisfare uno o più obiettivi. Tali obiettivi possono includere la corretta comprensione delle richieste dell'utente, la progettazione, la codifica, la verifica, la documentazione, il rilascio, la manutenzione, ecc. [18](#)

backend Il backend si riferisce a qualsiasi parte di un programma *software* o di un sito *web* che gli utenti non vedono e si contrappone al *frontend*. Il backend ha lo scopo di elaborare e memorizzare dati. [iii](#), [ix](#), [1](#), [4–7](#), [15–18](#), [29](#), [35](#), [65](#), [71](#), [72](#), [89](#), [90](#), [95](#), [97](#), [100](#), [102](#), [109](#)

Bitbucket *Bitbucket* è una piattaforma di *hosting* per la gestione del codice sorgente basata su *git*. Offre funzionalità di controllo di versione distribuito, permettendo ai team di sviluppatori di collaborare nella gestione dei progetti *software*, nel monitoraggio dei cambiamenti, nel controllo degli accessi e nella revisione del codice. *Bitbucket* fornisce anche strumenti per l'integrazione con altre applicazioni di sviluppo, come l'integrazione continua e lo strumento di pianificazione dei progetti. [21](#)

branch Un *branch* rappresenta una linea di sviluppo separata e indipendente all'interno di un *repository* e consente a più componenti del gruppo di lavorare contemporaneamente sullo stesso progetto evitando le interferenze prodotte dal lavoro in parallelo. [13](#), [107](#)

callback In programmazione, una *callback* è una funzione che viene passata come argomento a un'altra funzione e viene chiamata in seguito a un evento o a un'azione specifica. La *callback* consente di gestire il flusso di esecuzione del programma, consentendo alla funzione chiamante di eseguire determinate operazioni o di proseguire con altre azioni una volta completata l'operazione asincrona. [16](#), [110](#)

codebase Con *codebase* si intende, in informatica e in particolar modo nell'ambito dello *sviluppo software*, una collezione di codice sorgente. Rappresenta tutto ciò che è necessario per poter realizzare un prodotto software o parte di esso. 11

commit In un sistema di controllo di versione, un *commit* rappresenta l'azione di salvataggio delle modifiche apportate a un *repository* di codice. Quando un programmatore effettua un *commit*, i cambiamenti vengono registrati e memorizzati nel *repository*, consentendo di tenere traccia delle diverse versioni del codice nel tempo. I *commit* vengono generalmente accompagnati da un messaggio descrittivo che spiega le modifiche apportate. 21

coworking Attività di lavoro adottato in particolar modo dai liberi professionisti. È caratterizzato dalla compresenza in uno spazio fisico condiviso in cui vengono disposte delle postazioni autonome: favorisce la collaborazione fra i dipendenti in presenza e quelli in remoto. 4, 17

Document Object Model Il *Document Object Model* è la rappresentazione strutturata e gerarchica di un documento HTML, XHTML o XML. Il **DOM** consente di manipolare il contenuto, la struttura e lo stile di un documento utilizzando metodi e proprietà forniti dalle **API** dei *browser web*. Organizza il documento come un albero di nodi, in cui ogni nodo rappresenta un elemento, un attributo, un testo o qualsiasi altra entità presente nel documento. Ogni nodo nel **DOM** può avere figli, genitori e fratelli, formando una struttura gerarchica che riflette la struttura del documento stesso. Attraverso di esso, è possibile accedere e modificare dinamicamente gli elementi del documento, aggiungere o rimuovere elementi, modificare i loro attributi, gestire gli eventi e molto altro ancora. 104

economicità Si riferisce alla capacità di produrre risultati desiderati utilizzando il minor numero possibile di risorse o costi. In altre parole, l'economicità si riferisce all'**efficienza** con cui vengono utilizzate le risorse per raggiungere un obiettivo specifico. 20

efficacia Grado di conformità del prodotto rispetto alle norme vigenti e agli obiettivi prefissati. Per perseguire una buona efficacia è necessario impiegare sufficienti risorse. 2, 8, 11

efficienza Raggiungere gli obiettivi prefissati perseguendo una strategia di risparmio delle risorse (tempo e persone) a disposizione. È inversamente proporzionale alla quantità di risorse impiegate per le attività richieste e quindi è un indicatore quantitativo del loro consumo. 2, 3, 8, 11, 106

framework *Framework* è un termine largamente utilizzato nel contesto informatico. Fa riferimento ad un'infrastruttura di supporto che fornisce un insieme di strumenti, librerie, componenti e convenzioni di programmazione per facilitare lo sviluppo di *software*. Un *framework* offre un'architettura predefinita che consente ai programmatori di concentrarsi sulla logica di *business* dell'applicazione anziché sulle questioni tecniche di basso livello. Fornisce un ambiente strutturato e riutilizzabile che aiuta a velocizzare lo sviluppo e a migliorare l'**efficienza**, consentendo ai team di sviluppo di concentrarsi sulle funzionalità specifiche dell'applicazione. iii, ix, 6, 7, 9, 10, 12, 28, 29, 65, 67–69, 86, 87, 89, 97, 102, 109

frontend Il frontend di un programma *software* o di un sito *web* è tutto ciò con cui l'utente interagisce. Dal punto di vista dell'utente, il frontend è sinonimo di interfaccia utente. Dal punto di vista dello sviluppatore, è la progettazione dell'interfaccia e lo sviluppo dei metodi che la fanno funzionare. 1, 6, 15–17, 28–30, 90, 102–105

git È un sistema di controllo versione distribuito, utilizzato principalmente per gestire il codice sorgente durante lo sviluppo del *software*. Permette ai team di sviluppo di lavorare contemporaneamente sullo stesso progetto, tenere traccia delle modifiche, gestire i conflitti e riunire le modifiche in un'unica versione del progetto. *Git* è e supporta molte funzionalità come *e merge*, *stashing*, *cherry-pick*, e la possibilità di lavorare con *remoto* su *server*. 13, 21, 105

GraphQL Linguaggio di interrogazione per *API* sviluppato da *Facebook* nel 2012 e rilasciato come open source nel 2015. iii, 6, 17, 29, 65, 69

H-Farm H-Farm è un realtà fondata da Riccardo Donadon nel 2005 che si occupa di favorire la nascita e la crescita di *startup* di cui è il più grande incubatore in Italia. Interviene sia sul piano finanziario, con un ruolo da *venture capitalist*, sia su quello amministrativo e di *marketing*, permettendo così ai giovani imprenditori di focalizzarsi sul *core business*. 1

Hyper Text Markup Language Descrive il linguaggio principale di marcatura testi utilizzato in ambito *web* per poter realizzare la formattazione e l'impaginazione dei documenti ipertestuali. Viene sviluppato agli inizi degli anni 90 da Tim Berners-Lee quando lavorava per il CERN di Ginevra. Ad oggi lo standard più attuale è HTML5: uno standard in continua evoluzione che permette di integrare standard funzionali e *API*. 104

incrementale Modello di sviluppo del *software* in cui il prodotto finale viene sviluppato attraverso una serie di incrementi funzionali. In ogni incremento, una nuova funzionalità viene aggiunta al *software*, creando una versione migliorata del prodotto. Questo processo continua fino a quando il *software* raggiunge la sua forma finale. Il modello incrementale è caratterizzato da un approccio iterativo e si concentra sulla consegna continua di funzionalità utili per il cliente, consentendo di rispondere in modo flessibile ai cambiamenti nei requisiti durante il processo di sviluppo. Le prime funzionalità implementate solitamente sono quelle più importanti, in questo modo, con i successivi incrementi queste funzionalità tendono a stabilizzarsi per prime. 9

Integrated Development Environment In italiano "Ambiente di Sviluppo Integrato", è un *software* che fornisce un ambiente completo per lo sviluppo, il *debug* e la gestione di programmi o applicazioni *software*. Offre strumenti e funzionalità per facilitare il processo di sviluppo, come un editor di codice con evidenziazione della sintassi, un compilatore o interprete integrato, strumenti di *debugging* e un sistema di gestione dei progetti. Viene utilizzato dagli sviluppatori per scrivere, testare il codice in un'unica interfaccia, semplificando il flusso di lavoro e aumentando l'efficienza. 104

ISO/IEC 12207:1995 E'uno standard internazionale per il ciclo di vita del *software* che descrive un insieme di processi comuni al *software*. Il modello di ciclo di vita

del *software* descritto in ISO/IEC 12207:1995 comprende un insieme di processi che definiscono attività come l'analisi dei requisiti, la progettazione, la codifica, la verifica e la manutenzione del *software*. Lo standard definisce anche una serie di attività di supporto, come la gestione della configurazione, la gestione della qualità e la gestione del progetto, che devono essere eseguite durante tutto il ciclo di vita del *software*. È uno standard internazionale per il ciclo di vita del *software* che descrive un insieme di processi comuni al *software*. Il modello di ciclo di vita del *software* descritto in ISO/IEC 12207:1995 comprende un insieme di processi che definiscono attività come l'analisi dei requisiti, la progettazione, la codifica, la verifica e la manutenzione del *software*. Lo standard definisce anche una serie di attività di supporto, come la gestione della configurazione, la gestione della qualità e la gestione del progetto, che devono essere eseguite durante tutto il ciclo di vita del *software*. 105

ISO/IEC 31000:2009 E'uno standard internazionale che definisce i principi e le linee guida per la gestione del rischio. Si applica a qualsiasi tipo di organizzazione e a qualsiasi tipo di rischio, sia esso finanziario, operativo, tecnologico o di altro tipo. Lo standard fornisce un quadro di riferimento per la gestione del rischio che consente alle organizzazioni di identificare, valutare e gestire i rischi in modo sistematico ed efficace. 18

iterativo Iterativo è un termine che si riferisce a un processo o un'attività che viene eseguita in modo ripetitivo o ciclico. Nell'ambito dello sviluppo del *software*, l'approccio iterativo è un modello di sviluppo che prevede la divisione del lavoro in una serie di cicli chiamati iterazioni. In un contesto iterativo, l'obiettivo è ottenere progressivamente risultati parziali e miglioramenti incrementali nel corso del tempo. Ogni iterazione coinvolge la pianificazione, lo sviluppo, il test e la valutazione del lavoro svolto, al fine di apportare modifiche, apportare miglioramenti o affrontare requisiti aggiuntivi. 9

Message Queuing Telemetry Transport MQTT è un protocollo di messaggistica basato su standard, o un insieme di regole, utilizzato per la comunicazione tra macchine. I sensori intelligenti, i dispositivi indossabili e altri dispositivi di Internet delle cose (IoT) devono in genere trasmettere e ricevere dati su una rete con risorse limitate e larghezza di banda limitata. Questi dispositivi IoT utilizzano MQTT per la trasmissione dei dati, in quanto è facile da implementare e può comunicare i dati IoT in modo efficiente. MQTT supporta la messaggistica tra dispositivi e cloud e tra cloud e dispositivo¹. 104

mobile Solitamente riferito come "applicazione *mobile*" è un *software* sviluppato specificamente per essere utilizzato su dispositivi informatici *wireless* di piccole dimensioni, come *smartphone* e *tablet*, piuttosto che su *computer desktop* o portatili. 1, 4-6, 15, 17, 66

open source Open source significa non solo avere accesso al codice sorgente, ma anche rispettare determinati criteri di distribuzione del *software*. Questi criteri includono: la libera redistribuzione senza richiedere *royalty* o compensi; l'inclusione del codice sorgente e la possibilità di distribuirlo in forma compilata; il permesso di apportare modifiche e creare opere derivate; l'integrità del codice sorgente

¹MQTT - Definizione. URL: <https://aws.amazon.com/it/what-is/mqtt/#:~:text=MQTT%20C3%A8%20un%20protocollo%20di,per%20la%20comunicazione%20tra%20macchine..>

dell'autore, consentendo la distribuzione di "*file patch*" per le modifiche; l'assenza di discriminazione verso persone, gruppi o campi di attività; l'applicazione dei diritti connessi al programma a tutti coloro che lo ridistribuiscono senza licenze aggiuntive; l'indipendenza della licenza rispetto a una particolare distribuzione di *software*; l'assenza di restrizioni su altri *software* distribuiti insieme al *software open source*; e la neutralità tecnologica, senza clausole subordinate a una singola tecnologia o interfaccia². [11](#), [14](#), [28](#), [29](#), [107](#), [109](#), [110](#)

repository In informatica, è una raccolta di *file* e *directory*(cartelle) che costituiscono la versione corrente di un progetto software o di un insieme di dati. Un *repository* consente di gestire e controllare la cronologia delle modifiche apportate ai file e di mantenere traccia dei cambiamenti effettuati da ogni membro del *team*. Permette inoltre di collaborare con altri utenti, in modo che possano contribuire al progetto e apportare le loro modifiche in modo controllato e coordinato. [ix](#), [13](#), [105–107](#)

REST REST, acronimo di *Representational State Transfer*, è uno stile architetturale utilizzato per progettare servizi web che si basano sul protocollo HTTP. È un'approccio leggero e scalabile che promuove l'interoperabilità tra sistemi distribuiti. Nell'architettura REST, le risorse (come dati, funzionalità o servizi) sono identificate da URL (*Uniform Resource Locator*) e possono essere manipolate attraverso quattro operazioni fondamentali. . [5](#), [16](#), [17](#), [110](#)

rollback Il *rollback* è un'azione che comporta il ripristino di un sistema o di un'applicazione a uno stato precedente, solitamente per annullare le modifiche che sono state apportate. Nel contesto dello sviluppo del *software*, un *rollback* viene eseguito quando si verifica un errore o un problema nel codice o nel *database* e si decide di tornare a una versione precedente che funzionava correttamente. Il *rollback* è un meccanismo di sicurezza che consente di ripristinare un sistema allo stato precedente e prevenire il deterioramento o la perdita di dati. . [21](#)

Ruby on Rails È un *framework Open Source* per applicazioni *web* scritto in Ruby da David Heinemeier Hansson. Nell'ambito del progetto descritto in questa tesi, rappresenta quanto è stato adottato per realizzare il *backend*. [4](#)

smartwork Flessibilità prevista dalla legge all'interno di un rapporto di lavoro subordinato, finalizzata a incrementare la produttività e a facilitare il lavoratore nelle sue esigenze personali³. [14](#)

software house azienda specializzata nell'ingegneria e nello sviluppo del *software*; può lavorare su commissione per risolvere esigenze specifiche o produrre *software* di uso più generale da immettere sul mercato. [1](#)

startup Una *startup* è un'azienda o un progetto intrapreso da un imprenditore per cercare, sviluppare e convalidare un modello di business scalabile. [1](#), [107](#)

JavaScript Linguaggio di programmazione multiparadigma e orientato agli eventi che gli sviluppatori utilizzano per realizzare pagine *web* interattive con effetti

² Open source - Definizione. URL: <https://opensource.org/osd/>.

³ Treccani - smartwork. URL: [https://www.treccani.it/enciclopedia/smartwork_%28altro%29/#:~:text=\(smart%20work\)%2C%20s.%20m.%20inv,lavoratore%20nelle%20sue%20esigenze%20personali..](https://www.treccani.it/enciclopedia/smartwork_%28altro%29/#:~:text=(smart%20work)%2C%20s.%20m.%20inv,lavoratore%20nelle%20sue%20esigenze%20personali..)

dinamici, attraverso delle funzioni di *scripting*. Viene tipicamente utilizzato sia lato client che lato server, in quanto è possibile adottare *Node.js* per la creazione di *API REST*. Gli script vengono invocati all'avvento di alcuni eventi generati tipicamente dall'utente. 29, 30, 68, 105, 110

TypeScript Linguaggio di programmazione *Open Source* sviluppato da *Microsoft*. Rappresenta un superset di *JavaScript*, in quanto estende la sintassi. A differenza di *JavaScript*, che è un linguaggio di *scripting* interpretato, *TypeScript* introduce un processo di compilazione in cui il codice *TypeScript* viene tradotto in *JavaScript*. Durante la compilazione, vengono verificate le corrispondenze di tipo e vengono individuati gli errori potenziali prima dell'esecuzione del programma.. 28, 65, 68, 97, 101, 102, 105

user experience Approccio secondo il quale la qualità dell'interazione con uno strumento tecnologico non è riducibile unicamente all'assenza di problemi e al successo nel raggiungimento di uno scopo pragmatico, ma deve considerare anche aspetti quali il divertimento, il coinvolgimento emotivo, la motivazione, la piacevolezza estetica, la gratificazione⁴. 104

webhook Rappresenta una funzione di *callback* che permette a due *API* di comunicare fra di loro attraverso l'uso degli eventi e in modo semplice. Vengono tipicamente adottati in contesti di applicazioni *web* per lo scambio di informazioni di sistemi separati. 6, 17

wireframe Un *wireframe* è un tipo di rappresentazione visiva di base di un'interfaccia utente, spesso utilizzato nel processo di progettazione dell'usabilità e dell'interfaccia di un'applicazione o di un sito *web*. Il *wireframe* fornisce una rappresentazione scheletrica e schematica dei *layout*, delle posizioni degli elementi e della navigazione dell'interfaccia, senza includere dettagli estetici come colori o grafica. Viene utilizzato per definire la struttura e l'organizzazione di una pagina o di un'applicazione, permettendo ai progettisti e agli sviluppatori di visualizzare e condividere le idee in modo rapido ed efficiente. 70, 100

⁴Treccani - *user experience*. URL: https://www.treccani.it/enciclopedia/user-experience_%28Lessico-del-XXI-Secolo%29/#:~:text=E2%80%93Approccio%20secondo%20il%20quale%20la,emotivo%2C%20la%20motivazione%2C%20la%20piacevolezza/.

Bibliografia

Riferimenti bibliografici

- [13] E. Gamma. *Design Patterns: Elementi Per Il Riutilizzo Di Software a Oggetti Professionale* / Pearson education Italia. Pearson Education Italia, 2002. ISBN: 9788871921501. URL: <https://books.google.it/books?id=IkzmAAACAAJ> (cit. a p. 87).
- [25] Ian Sommerville. *Ingegneria del software*. Pearson, 2017. ISBN: 9788891902245. URL: https://www.pearson.it/opera/pearson/0-6424-ingegneria_del_software (cit. alle pp. 8–10).

Pubblicazioni scientifiche

- [7] Jens Binder et al. «The academic value of internships: Benefits across disciplines and student backgrounds». In: *Contemporary Educational Psychology* 41 (dic. 2014). DOI: [10.1016/j.cedpsych.2014.12.001](https://doi.org/10.1016/j.cedpsych.2014.12.001) (cit. a p. 30).
- [12] Themistoklis Diamantopoulos, Nikolaos Saoulidis e Andreas Symeonidis. «Automated issue assignment using topic modelling on Jira issue tracking data». In: *IET Software* 17 (mag. 2023). DOI: [10.1049/sfw2.12129](https://doi.org/10.1049/sfw2.12129) (cit. a p. 11).

Siti web consultati

- [1] *Active Admin*. URL: <https://activeadmin.info/> (cit. a p. 5).
- [2] *Angular - Architettura*. URL: <https://v15.angular.io/guide/architecture> (cit. alle pp. 67, 68).
- [3] *Angular - Component*. URL: <https://v15.angular.io/guide/component-overview> (cit. a p. 67).
- [4] *Angular - Dependency Injection*. URL: <https://v15.angular.io/guide/dependency-injection-overview> (cit. a p. 87).
- [5] *Angular - Service*. URL: <https://v15.angular.io/guide/creating-injectable-service> (cit. a p. 69).
- [6] *Angular - Singleton*. URL: <https://v15.angular.io/guide/singleton-services> (cit. a p. 88).
- [8] *CONFIDA*. URL: <https://www.confida.com/chi-siamo/> (cit. a p. 4).

- [9] *CONFIDA - Dati di settore*. URL: <https://www.confida.com/dati-di-settore/> (cit. a p. 4).
- [10] *Da Mario - Android*. URL: https://play.google.com/store/apps/details?id=com.ia_shop/ (cit. a p. 5).
- [11] *Da Mario - iOS*. URL: <https://apps.apple.com/us/app/da-mario/id1580905176/> (cit. a p. 5).
- [14] *Infinite Area*. URL: <https://infinitearea.com/> (cit. a p. 4).
- [15] *ISO/IEC 31000-2009*. URL: <https://project.lsst.org/riskmanagement/31000-2009.pdf/> (cit. a p. 18).
- [16] *Jira*. URL: <https://www.atlassian.com/it/software/jira> (cit. a p. 11).
- [17] *Manifesto Agile*. URL: <https://agilemanifesto.org/iso/it/manifesto.html> (cit. a p. 9).
- [18] *Moku S.r.l*. URL: <https://www.moku.io/it/> (cit. a p. 1).
- [19] *Moku S.r.l - Servizi*. URL: <https://www.moku.io/it/servizi/> (cit. a p. 2).
- [20] *MQTT - Definizione*. URL: <https://aws.amazon.com/it/what-is/mqtt/#:~:text=MQTT%20%C3%A8%20un%20protocollo%20di,per%20la%20comunicazione%20tra%20macchine.> (cit. a p. 108).
- [21] *Open source - Definizione*. URL: <https://opensource.org/osd/> (cit. a p. 109).
- [22] *Passbolt*. URL: <https://www.passbolt.com/> (cit. a p. 14).
- [23] *Portfolio - Da Mario*. URL: <https://infinitearea.com/startup-studio/> (cit. a p. 4).
- [24] *Scrum - What is Scrum?* URL: <https://www.scrum.org/resources/what-scrum-module> (cit. a p. 10).
- [26] *Tempo - plugin*. URL: <https://www.tempo.io/> (cit. a p. 12).
- [27] *Treccani - smartwork*. URL: [https://www.treccani.it/enciclopedia/smartwork_%28altro%29/#:~:text=\(smart%20work\)%2C%20s.%20m.%20inv,lavoratore%20nelle%20sue%20esigenze%20personali.](https://www.treccani.it/enciclopedia/smartwork_%28altro%29/#:~:text=(smart%20work)%2C%20s.%20m.%20inv,lavoratore%20nelle%20sue%20esigenze%20personali.) (cit. a p. 109).
- [28] *Treccani - user experience*. URL: https://www.treccani.it/enciclopedia/user-experience_%28Lessico-del-XXI-Secolo%29/#:~:text=%E2%80%93Approccio%20secondo%20il%20quale%20la,emotivo%2C%20la%20motivazione%2C%20la%20piacevolezza/ (cit. a p. 110).