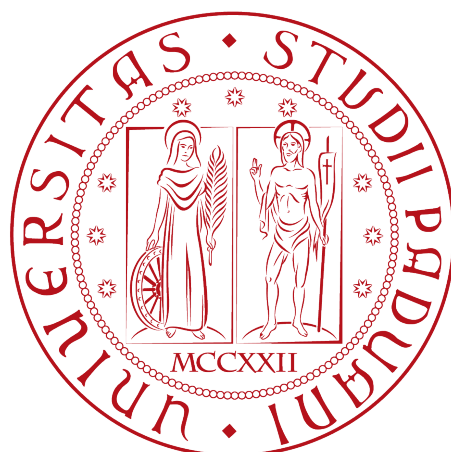


Università degli Studi di Padova

DIPARTIMENTO DI MATEMATICA "TULLIO LEVI-CIVITA"

CORSO DI LAUREA IN INFORMATICA



## Sviluppo di una web-app per la gestione degli ordini di ristoranti

*Tesi di laurea*

*Relatore*

Prof. Francesco Ranzato

*Laureando*

Luciano Wu

---

ANNO ACCADEMICO 2021-2022

Luciano Wu: *Sviluppo di una web-app per la gestione degli ordini di ristoranti*, Tesi di laurea, © Dicembre 2022.

# Sommario

Il presente documento descrive il lavoro svolto durante il periodo di stage, della durata di circa trecentoventi ore, dal laureando Luciano Wu presso l'azienda Azienda SyncLab S.p.A.

Gli obiettivi da raggiungere erano molteplici. In primo luogo era richiesto lo sviluppo di una web-app per la gestione delle ordinazioni dei piatti di un ristorante sushi all-you-can-eat. Tale applicazione permette di registrare le ordinazioni e tenere traccia dei piatti in arrivo. Le componenti fondamentali richiesti sono gestione del tavolo, gestione delle ordinazioni e infine gestione dell'accesso. Terzo ed ultimo obiettivo era l'integrazione delle varie maschere e produrre un documento tecnico che descrive le interfacce sviluppate.



*“Life is really simple, but we insist on making it complicated”*

— Confucius

# Ringraziamenti

*Innanzitutto, vorrei esprimere la mia gratitudine al Prof. Francesco Ranzato, relatore della mia tesi, per l'aiuto e il sostegno fornitomi durante la stesura del lavoro.*

*Ringrazio Fabio Pallaro e tutta SyncLab, per avermi dato la possibilità di svolgere lo stage e il progetto per misurare le mie capacità e imparare delle nuove.*

*Desidero ringraziare con affetto i miei genitori per il sostegno, il grande aiuto e per essermi stati vicini in ogni momento durante gli anni di studio.*

*Ho desiderio di ringraziare poi i miei amici, in modo particolare Diego e Samuele, per tutti i bellissimi anni passati insieme e le mille avventure vissute.*

*Padova, Dicembre 2022*

Luciano Wu



# Indice

|          |  |           |
|----------|--|-----------|
| <b>1</b> | <b>Introduzione</b>                    | <b>1</b>  |
| 1.1      | SyncLab . . . . .                      | 1         |
| 1.1.1    | Prodotti . . . . .                     | 2         |
| 1.2      | Introduzione al progetto . . . . .     | 3         |
| 1.3      | La soluzione individuata . . . . .     | 3         |
| 1.4      | Organizzazione del testo . . . . .     | 4         |
| <b>2</b> | <b>Il progetto di stage</b>            | <b>5</b>  |
| 2.1      | Pianificazione del lavoro . . . . .    | 5         |
| 2.2      | Obiettivi richiesti . . . . .          | 6         |
| 2.2.1    | Notazione . . . . .                    | 6         |
| 2.2.2    | Obiettivi fissati . . . . .            | 6         |
| 2.3      | Modalità di lavoro . . . . .           | 6         |
| 2.4      | Strumenti utilizzati . . . . .         | 7         |
| 2.4.1    | Comunicazione: . . . . .               | 7         |
| 2.4.2    | Sviluppo: . . . . .                    | 9         |
| <b>3</b> | <b>Analisi dei requisiti</b>           | <b>11</b> |
| 3.1      | Descrizione generale . . . . .         | 11        |
| 3.1.1    | Interfacce della web-app . . . . .     | 11        |
| 3.1.2    | Caratteristiche degli utenti . . . . . | 11        |
| 3.1.3    | Tecnologie utilizzate . . . . .        | 12        |
| 3.1.4    | Descrizione delle tecnologie . . . . . | 12        |
| 3.2      | Casi d'uso . . . . .                   | 14        |
| 3.2.1    | Introduzione . . . . .                 | 14        |
| 3.2.2    | Attori primari . . . . .               | 14        |
| 3.3      | Requisiti . . . . .                    | 15        |
| 3.3.1    | Panoramica . . . . .                   | 16        |
| <b>4</b> | <b>Progettazione e codifica</b>        | <b>17</b> |
| 4.1      | Progettazione . . . . .                | 17        |
| 4.1.1    | Architettura Angular . . . . .         | 17        |
| 4.1.2    | Architettura SushiLab . . . . .        | 18        |
| 4.1.3    | Progettazione delle viste . . . . .    | 19        |
| 4.1.4    | Progettazione API . . . . .            | 20        |
| 4.1.5    | Progettazione dei componenti . . . . . | 21        |
| 4.2      | Codifica . . . . .                     | 21        |
| 4.2.1    | Interfacce . . . . .                   | 21        |

|          |   |           |
|----------|---|-----------|
| 4.2.2    | Componenti service . . . . .                    | 22        |
| 4.2.3    | Componenti guard . . . . .                      | 22        |
| <b>5</b> | <b>Verifica e validazione</b>                   | <b>23</b> |
| 5.1      | Verifica . . . . .                              | 23        |
| 5.2      | Validazione e collaudo . . . . .                | 24        |
| <b>6</b> | <b>Conclusioni</b>                              | <b>25</b> |
| 6.1      | Raggiungimento degli obiettivi . . . . .        | 25        |
| 6.1.1    | Obiettivi fissati . . . . .                     | 25        |
| 6.1.2    | Svolgimento del lavoro . . . . .                | 26        |
| 6.2      | Conoscenze acquisite . . . . .                  | 28        |
| 6.3      | Valutazione . . . . .                           | 28        |
| 6.3.1    | L'azienda . . . . .                             | 28        |
| 6.3.2    | Personale . . . . .                             | 28        |
| <b>A</b> | <b>Descrizione dettagliata dei casi d'uso</b>   | <b>29</b> |
| <b>B</b> | <b>Elenco dei requisiti</b>                     | <b>43</b> |
| <b>C</b> | <b>Descrizione dettagliata delle interfacce</b> | <b>47</b> |
|          | <b>Glossario</b>                                | <b>61</b> |
|          | <b>Bibliografia</b>                             | <b>63</b> |



# Elenco delle figure

|      |   |    |
|------|---|----|
| 1.1  | Punti di forza di SyncLab . . . . .                                 | 1  |
| 1.2  | I vari prodotti di SyncLab . . . . .                                | 2  |
| 1.3  | Diagramma dei componenti della web-app . . . . .                    | 3  |
| 2.1  | Canale SyncLab su Discord . . . . .                                 | 7  |
| 2.2  | Board di Trello per il progetto sushi-lab . . . . .                 | 8  |
| 2.3  | Calendario personale per il progetto sushi-lab . . . . .            | 8  |
| 2.4  | Repo su GitHub per il progetto sushi-lab . . . . .                  | 9  |
| 2.5  | Interfaccia di Visual Studio Code . . . . .                         | 10 |
| 2.6  | Interfaccia con lo strumento ispezione di Google Chrome . . . . .   | 10 |
| 3.1  | Logo di Angular . . . . .   | 13 |
| 3.2  | Logo di HTML e CSS . . . . .  | 13 |
| 3.3  | Logo di Stoplight . . . . .   | 14 |
| 3.4  | Logo di Angular Material . . . . .                                  | 14 |
| 4.1  | Architettura Angular . . . . .                                      | 18 |
| 4.2  | Struttura file SushiLab . . . . .                                   | 19 |
| 4.3  | Figma SushiLab . . . . .  | 20 |
| 4.4  | Stoplight SushiLab . . . . .  | 21 |
| A.1  | Use Case - UC 1 . . . . .   | 29 |
| A.2  | Use Case - UC 1.1, UC 1.2, UC 1.3, UC 1.4, UC 1.5, UC 1.6 . . . . . | 30 |
| A.3  | Use Case - UC 2 . . . . .   | 32 |
| A.4  | Use Case - UC 2.1, UC 2.2, UC 2.3, UC 2.4 . . . . .                 | 32 |
| A.5  | Use Case - UC 3 . . . . .   | 34 |
| A.6  | Use Case - UC 3.1, UC 3.2, UC 3.3 . . . . .                         | 34 |
| A.7  | Use Case - UC 3.2.1 . . . . .                                       | 36 |
| A.8  | Use Case - UC 3.3.1 . . . . .                                       | 36 |
| A.9  | Use Case - UC 4 . . . . .   | 37 |
| A.10 | Use Case - UC 4.1, UC 4.2, UC 4.3 . . . . .                         | 38 |
| A.11 | Use Case - UC 4.4, UC4.5, UC4.6 . . . . .                           | 39 |
| A.12 | Use Case - UC 1.7, UC1.8, UC1.9 . . . . .                           | 40 |
| A.13 | Use Case - UC 5 . . . . .   | 41 |
| C.1  | Maschera menù di SushiLab . . . . .                                 | 47 |
| C.2  | Maschera gestione del tavolo di SushiLab . . . . .                  | 48 |
| C.3  | Maschera unisci sessione di SushiLab . . . . .                      | 48 |
| C.4  | Maschera QR-code tavolo di SushiLab . . . . .                       | 49 |

|      |   |    |
|------|---|----|
| C.5  | Maschera ordini del tavolo e QR-code degli ordini di SushiLab . . . . . | 49 |
| C.6  | Maschera ordini personali di SushiLab . . . . .                         | 50 |
| C.7  | Maschera ordini in arrivo di SushiLab . . . . .                         | 50 |
| C.8  | Maschera login di SushiLab . . . . .                                    | 51 |
| C.9  | Maschera registrazione di SushiLab . . . . .                            | 51 |
| C.10 | Maschera cambia password di SushiLab . . . . .                          | 52 |
| C.11 | Maschera area personale e blacklist di SushiLab . . . . .               | 53 |

## Elenco delle tabelle

|     |   |    |
|-----|---|----|
| 5.1 | Tabella dei test sui requisiti. . . . .                     | 24 |
| 6.1 | Tabella del raggiungimento degli obiettivi fissati. . . . . | 25 |
| 6.2 | Tabella del consuntivo di lavoro. . . . .                   | 27 |
| B.1 | Tabella dei requisiti. . . . .                              | 45 |

# Capitolo 1

## Introduzione

*In questo capitolo viene descritta l'azienda nella quale è stato svolto lo stage e viene spiegato il progetto di tirocinio.*

### 1.1 SyncLab

SyncLab è una Innovative Company collocata in tutta Italia, è nata nel 2002 con sede principale a Napoli ed è cresciuta velocemente. Attualmente, SyncLab ha 6 sedi in tutta Italia, più di 300 dipendenti e più di 150 clienti diretti e finali.

SyncLab propone servizi innovativi che aiutano i clienti nella realizzazione, progettazione e manutenzione di soluzioni IT. L'azienda ha collaborato con varie compagnie tra questi i più importanti sono: Tim, Trenitalia, HM, Grimaldi Lines, notartel, sky, eni, enel, vodafone, RayWay, Poste Italiane, Intesa Sanpaolo, Ministero dell'economia delle finanze, fastweb e UniCredit.



**Figura 1.1:** Punti di forza di SyncLab

### 1.1.1 Prodotti

Come accennato, SyncLab opera nel settore IT e i suoi prodotti nascono dalle competenze acquisite e maturate durante i loro 20 anni di collaborazioni. I prodotti coprono vari ambiti come quelli delle telecomunicazioni, utilities, finanza e salute.

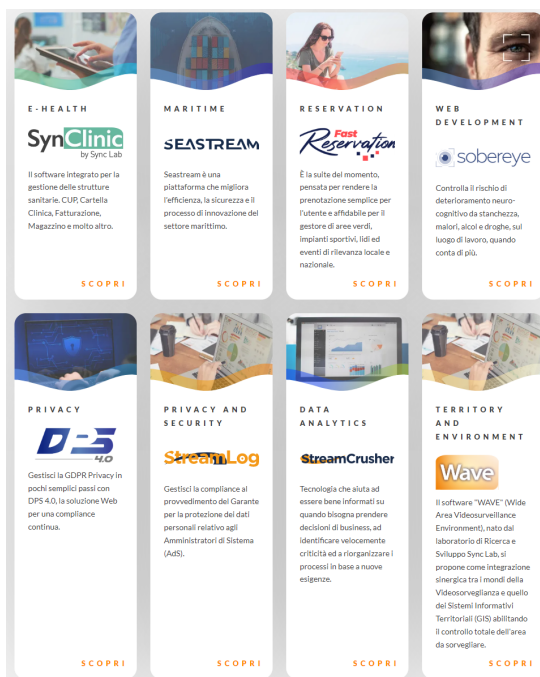


Figura 1.2: I vari prodotti di SyncLab

- **SynClinic:** software integrato per la gestione delle strutture sanitarie, il quale offre il sistema di cartella clinica digitale, il servizio di fatturazione, la gestione informatizzata dei farmaci, gli strumenti nativi di gestione amministrativa e tanto altro.
- **SEASTREAM:** una piattaforma nata per migliorare e potenziare le attività di business nel settore armatoriale e di altri operatori del mercato marittimo.
- **FastReservation:** applicazione realizzata per rendere il sistema di prenotazione più facile e affidabile per gli utenti.
- **Sobereye:** soluzione per la sicurezza proattiva per la prevenzione degli incidenti nei settori di trasporti, estrazione, costruzioni e industriale.
- **DPS 4.0:** applicativo che permette di gestire la GDPR Privacy Policy. L'applicativo, tramite una guida semplice sviluppata dagli ingegneri esperti nell'ambito dell'user experience, offre la possibilità di modificare ed aggiornare i documenti sulla privacy con il minimo sforzo.
- **StreamLog:** soluzione per la protezione dei dati personali relativo agli amministratori di sistema, offre la possibilità di effettuare il controllo degli accessi degli utenti ai sistemi in modo semplice ed efficace.
- **StreamCrusher:** tecnologia che serve per aiutare a prendere decisioni di business, identificando velocemente i punti critici e riformando i processi in base alle nuove esigenze.

- **Wave:** software nato per i sistemi di videosorveglianza, con l'obiettivo di avere una maggiore copertura territoriale col minor numero di telecamere installate e possibilmente utilizzare il minor numero possibile di risorse.

## 1.2 Introduzione al progetto

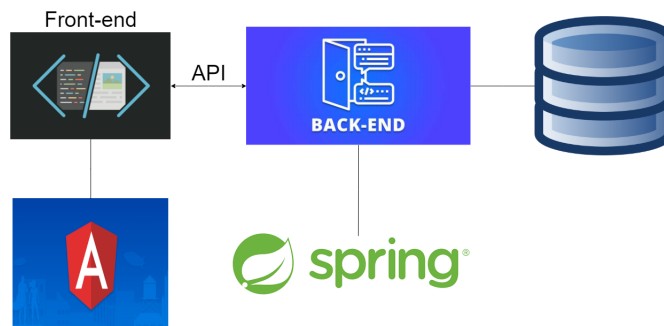
Lo scopo dello stage quello di realizzare una [web-app](#) per la gestione delle ordinazioni dei piatti di un ristorante sushi "all you can eat" con tale formula i ristoranti offrono la possibilità di ordinare senza limiti ad un prezzo fisso. Quindi i consumi dei clienti in questi locali è maggiore rispetto ai ristoranti tradizionali, perché con la formula "all-you-can-eat" spesso i clienti mangiano oltre il loro livello di sazietà, in aggiunta a questo il menù contiene centinaia di piatti diversi, il che comporta un elevato numero di ordinazioni da gestire causando così il problema di identificare chi ha ordinato un specifico piatto al momento della consegna.

## 1.3 La soluzione individuata

SyncLab ha deciso di risolvere questo problema tramite SushiLab, una web-app che offre la possibilità agli utenti di ordinare i piatti e tracciare tutte le ordinazioni. Per semplificare il riconoscimento del piatto nel momento dell'arrivo abbiamo deciso di inserire un'immagine a tutti i piatti.

Un'utente può inoltre registrarsi alla piattaforma per salvare dei piatti nella propria lista dei preferiti, aggiungere ingredienti alla propria blacklist e recensire un piatto.

SushiLab è composta da due parti, la parte [front-end](#) e la parte [back-end](#). Il back-end deve essere implementato tramite Java utilizzando il framework [Spring](#) e ha il ruolo principale di un web server, che deve comunicare con il data-base, in cui vengono salvate tutte le informazioni dei piatti e tutti gli ordini effettuati dai clienti. La parte di front-end è invece realizzata in Javascript, utilizzando il framework Angular. La comunicazione tra le due parti della web-app avviene tramite le chiamate [REST](#) fornite dal back-end. Ciascuna parte dell'applicazione è stata affidata a più persone. Discutendo con il tutor aziendale, Fabio Pallaro, abbiamo individuato i principali obiettivi per realizzare la web-app ed a me è stato assegnato il compito di sviluppare il front-end dell'applicazione. Si è deciso di realizzare una web-app in quanto essa non richiede di scaricare ed installare un'applicativo, così riducendo il tempo delle ordinazioni dei clienti.



**Figura 1.3:** Diagramma dei componenti della web-app

## 1.4 Organizzazione del testo

**Il secondo capitolo** descrive in dettaglio la pianificazione e gli obiettivi dello stage.

**Il terzo capitolo** descrive i principali casi d'uso e requisiti fondamentali individuati.

**Il quarto capitolo** approfondisce la progettazione e la codifica. La progettazione entra nel dettaglio dell'architettura, delle viste progettate e delle API, mentre la codifica descrive il processo di sviluppo dei principali componenti e service dell'applicazione.

**Il quinto capitolo** approfondisce la fase di verifica e validazione della web-app.

**Il sesto capitolo** descrive le conclusioni dell'intero progetto, parlando del prodotto finale e le competenze acquisite durante lo stage.

**A** contiene l'elenco completo dei casi d'uso individuati, fornendo una rappresentazione in formato UML e una descrizione formale di essi.

**B** contiene la tabella completa dei requisiti individuati, tracciando ogni requisito alla relativa fonte.

**C** contiene l'elenco completo delle interfacce e dei componenti sviluppati.

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*<sup>[gl]</sup>;
- i termini in lingua straniera o facenti parti del gergo tecnico sono evidenziati con il carattere *corsivo*.

# Capitolo 2

## Il progetto di stage

*In questo capitolo, viene descritta la pianificazione e gli obiettivi dello stage. Infine vengono elencati gli strumenti utilizzati durante lo svolgimento dello stage.*

### 2.1 Pianificazione del lavoro

#### Pianificazione settimanale

- Prima Settimana (40 ore)
  - Incontro con persone coinvolte nel progetto per discutere i requisiti e le richieste relativamente al sistema da sviluppare;
  - Presentazione strumenti di lavoro per la condivisione del materiale di studio e per la gestione dell'avanzamento;
  - Condivisione scaletta di argomenti;
  - Ripasso del linguaggio Java SE;
  - Ripasso concetti Web (Servlet, servizi REST, Json ecc.).
- Seconda Settimana (40 ore)
  - Studio principi generali di Spring Core (IOC, Dependency Injection);
  - Studio SpringBoot;
  - Studio Spring Data/DataRest.
- Terza Settimana (40 ore)
  - Ripasso linguaggio Javascript;
  - Studio del linguaggio TypeScript.
- Quarta Settimana (40 ore)
  - Studio piattaforma NodeJS e AngularCLI;
  - Studio framework Angular.
- Quinta Settimana (40 ore)
  - Analisi e studio del progetto SushiLab;
  - Progettazione ed implementazione della nuova maschera di accesso.
- Sesta Settimana (40 ore)
  - Progettazione ed implementazione nuova maschera "Inserimento Ordine e gestione del tavolo".
- Settima Settimana (40 ore)
  - Progettazione ed implementazione nuova maschera "Merge Ordini e Visualizzazione Ordine unico".

- Ottava Settimana - Conclusione (40 ore)
  - Verifica del funzionamento della web-app;
  - Validazione della web-app;
  - Termine integrazioni e collaudo finale.

## 2.2 Obiettivi richiesti

### 2.2.1 Notazione

Si farà riferimento ai requisiti secondo le seguenti notazioni:

- O per i requisiti obbligatori, vincolanti in quanto obiettivo primario richiesto dal committente;
- D per i requisiti desiderabili, non vincolanti o strettamente necessari, ma dal riconoscibile valore aggiunto;
- F per i requisiti facoltativi, rappresentanti valore aggiunto non strettamente competitivo.

Le sigle precedentemente indicate saranno seguite da una coppia sequenziale di numeri, identificativo del requisito.

### 2.2.2 Obiettivi fissati

Si prevede lo svolgimento dei seguenti obiettivi:

- Obbligatori:
  - O01: Acquisizione competenze sulle tematiche sopra descritte;
  - O02: Capacità di raggiungere gli obiettivi richiesti in autonomia seguendo il cronoprogramma;
  - O03: Portare a termine le implementazioni previste con una percentuale di superamento pari al 80%.
- Desiderabili:
  - D01: Portare a termine le implementazioni previste con una percentuale di superamento pari al 100%.
- Facoltativi:
  - F01: Apportare un valore aggiunto al gruppo di lavoro durante le fasi di progettazione delle interfacce.

## 2.3 Modalità di lavoro

L'azienda utilizza lo sviluppo agile del software. La modalità agile aiuta a ridurre il rischio di fallimento sviluppando il software in finestre di tempo limitate, chiamate iterazioni, che in genere durano qualche settimana. Ogni iterazione è un piccolo progetto a sé stante e deve contenere tutto ciò che è necessario per rilasciare un piccolo incremento nelle funzionalità del software: pianificazione, analisi dei requisiti, progettazione, implementazione, test e documentazione. Ogni settimana viene effettuato un incontro con il tutor aziendale per discutere dei problemi trovati durante lo sviluppo e fare il punto della situazione del progetto.



## 2.4 Strumenti utilizzati

### 2.4.1 Comunicazione:

Per avere una buona comunicazione con l'azienda si è deciso di utilizzare:

#### Discord:

Una piattaforma di **VoiP**, messaggistica istantanea e distribuzione digitale. Su discord gli utenti comunicano con chiamate vocali, video chiamate ed è anche possibile condividere lo schermo.

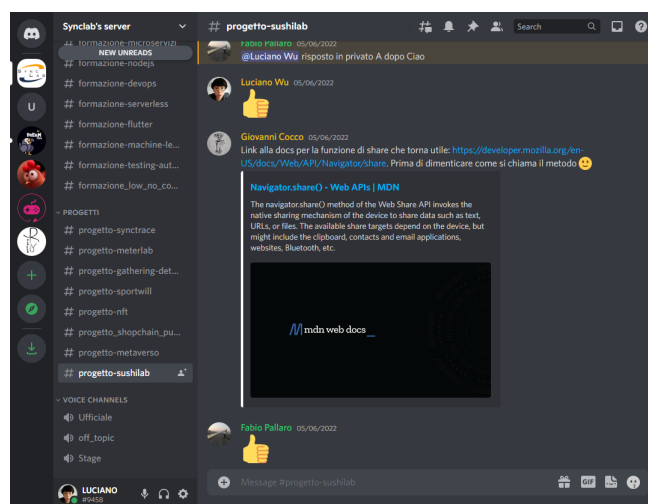


Figura 2.1: Canale SyncLab su Discord

#### Trello:

Un software gestionale in stile Kanban, in cui è possibile pianificare il progetto, condividere lo stato di svolgimento di una card con altri collaboratori, spostare varie card tra le liste e assegnarle ad un utente.

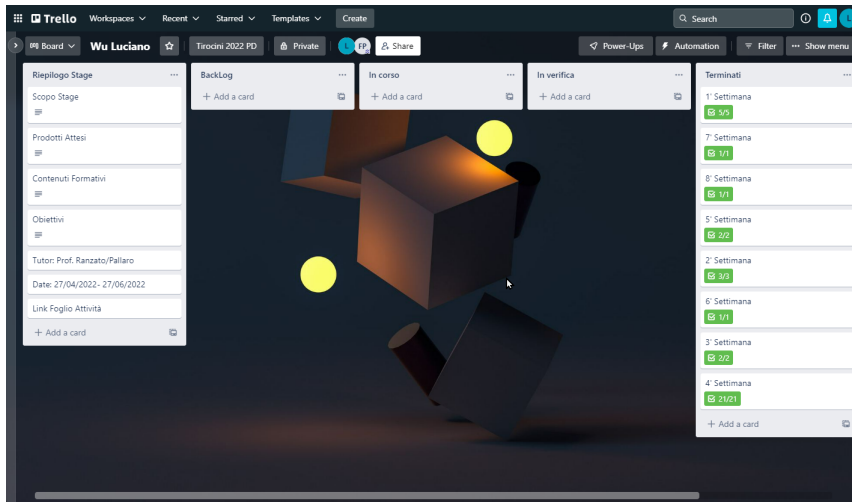


Figura 2.2: Board di Trello per il progetto sushi-lab

### Google Sheets:

Una web-app che fornisce tutte le funzionalità di un foglio elettronico, lo abbiamo utilizzato come un diario giornaliero, dove vengono descritti i compiti svolti durante una certa giornata.

| Date | Task   | Completed |
|------|--|-----------|
| 17   | 2022-05-18 Studio su lifecycle hooks e comunicazione tra i componenti di angular | ✓         |
| 18   | 2022-05-19 Studio su chiamato http, rest, build e deploy                         | ✓         |
| 19   | 2022-05-20 Studio su lazy loading e reactive forms                               | ✓         |
| 20   | 2022-05-23 Progettazione maschera accesso  | ✓         |
| 21   | 2022-05-24 Progettazione maschera tavolo   | ✓         |
| 22   | 2022-05-25 Progettazione maschera ordini   | ✓         |
| 23   | 2022-05-26 Progettazione maschera password dimenticata                           | ✓         |
| 24   | 2022-05-27 Progettazione guard   | ✓         |
| 25   | 2022-05-30 Progettazione maschera gestione tavolo(creazione sessione tavolo)     | ✓         |
| 26   | 2022-05-31 Progettazione maschera gestione tavolo(unione sessione tavolo)        | ✓         |
| 27   | 2022-06-01 Progettazione maschera gestione ordini del tavolo                     | ✓         |
| 28   | 2022-06-02 Progettazione maschera gestione ordini personali                      | ✓         |
| 29   | 2022-06-03 Progettazione maschera gestione ordini in arrivo                      | ✓         |
| 30   | 2022-06-06 Progettazione merge ordini tavolo                                     | ✓         |
| 31   | 2022-06-07 Progettazione visualizza ordine unico                                 | ✓         |
| 32   | 2022-06-08 Progettazione sposta ordini in arrivo                                 | ✓         |
| 33   | 2022-06-09 Abbellimento css maschera tavolo                                      | ✓         |
| 34   | 2022-06-10 Abbellimento css maschera ordini                                      | ✓         |
| 35   | 2022-06-14 Abbellimento css maschera password dimenticata                        | ✓         |
| 36   | 2022-06-15 Controllo completo su file di stile                                   | ✓         |
| 37   | 2022-06-16 Controllo completo su file di guard                                   | ✓         |
| 38   | 2022-06-17 Controllo completo su file di service                                 | ✓         |
| 39   | 2022-06-20 Controllo completo su file di type script                             | ✓         |
| 40   | 2022-06-21 Stesura documento analisi tecnica                                     | ✓         |
| 41   | 2022-06-22 Completamento documento analisi tecnica                               | ✓         |
| 42   | 2022-06-23 Preparazione collaudo finale  | ✓         |
| 43   | 2022-06-24 Collaudo finale   | ✓         |

Figura 2.3: Calendario personale per il progetto sushi-lab

### Google Meet:

Un software nato per le videochiamate sviluppato da Google, in cui è possibile mandare messaggi, condividere lo schermo e l'audio contemporaneamente, lo abbiamo utilizzato per alcuni incontri con alcuni collaboratori esterni per analisi dei requisiti.

### 2.4.2 Sviluppo:

Per avere una buona efficienza per lo sviluppo della web-app vengono utilizzati gli strumenti più popolari per lo sviluppo, che sono:

#### GitHub:

Servizio di hosting per lo sviluppo software, è implementato insieme con lo strumento di controllo di versione distribuito, nel mio caso è stato utilizzato per condividere e tracciare i file con gli altri collaboratori del progetto.

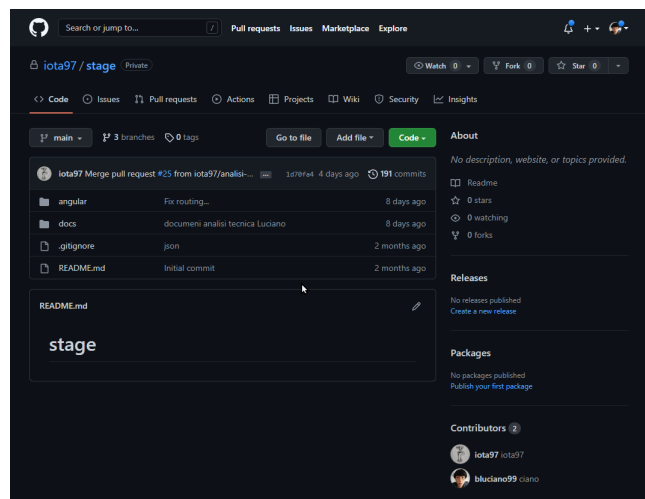


Figura 2.4: Repo su GitHub per il progetto sushi-lab

#### Visual Studio Code:

Un editor sviluppato da Microsoft. Contiene tutte le funzionalità di un editor ed è completamente gratuito. Le principali funzionalità utilizzate per il progetto sono:

- Le estensioni per il codice html, css e TypeScript;
- Il source control di Git integrato con Visual Studio Code;
- La funzione search con la quale è possibile ricercare un termine dentro tutti i file e farne la sostituzione.

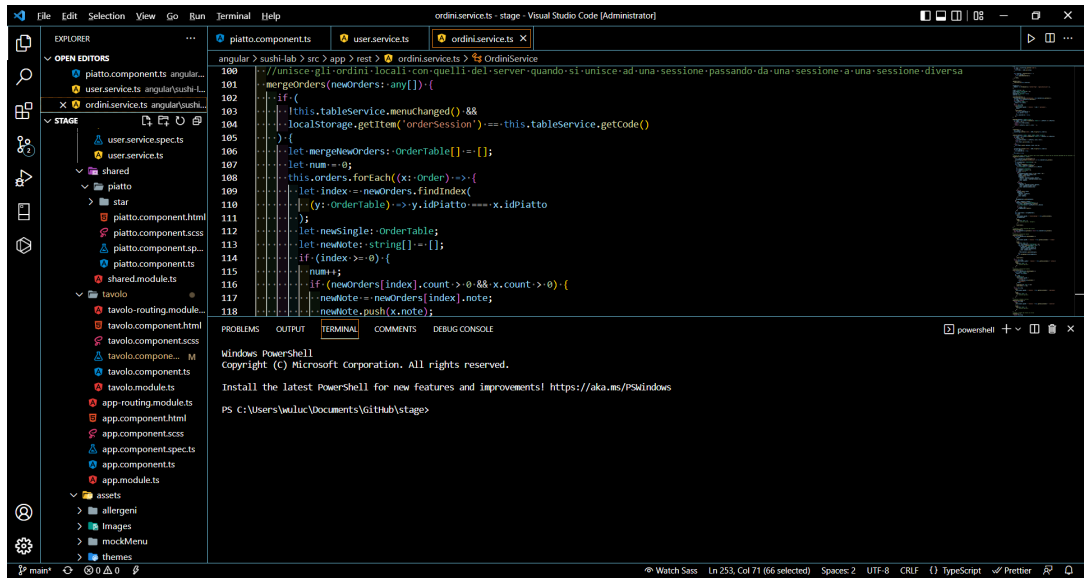


Figura 2.5: Interfaccia di Visual Studio Code

### Google Chrome:

Un browser web sviluppato da Google. È il browser più popolare al mondo grazie alla sua buona stabilità ed elevata velocità. La principale funzionalità utilizzata di Chrome durante lo sviluppo sono gli strumenti per sviluppatori, che permette di vedere tutti i dati di uno specifico elemento HTML presente nella pagina, in più è possibile vedere tutti i dati salvati in locale della pagina.

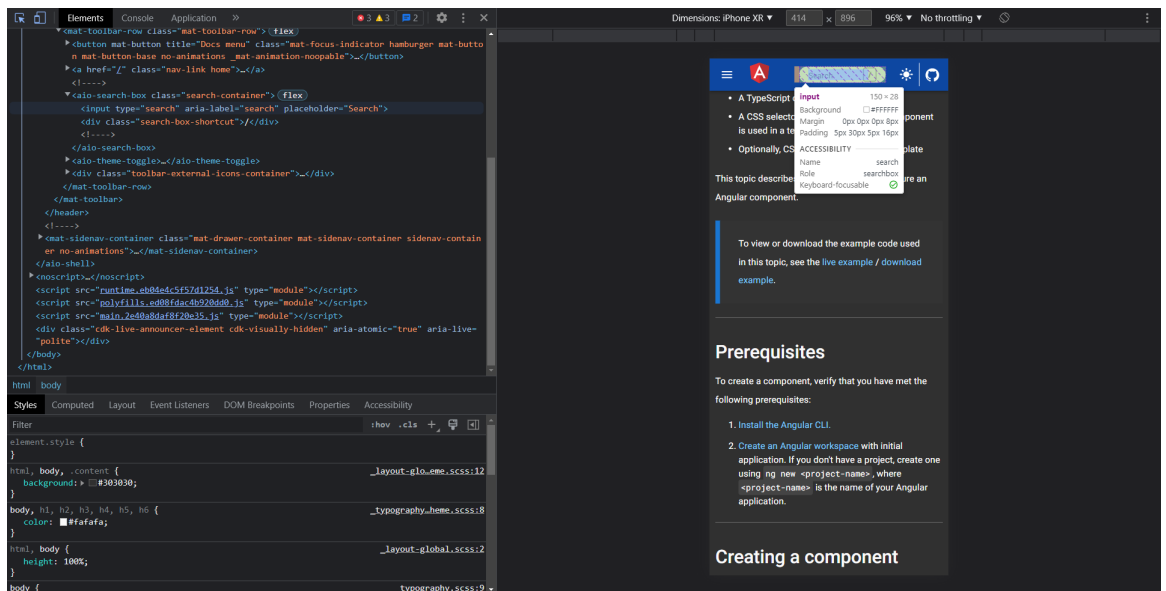


Figura 2.6: Interfaccia con lo strumento ispeziona di Google Chrome

## Capitolo 3

# Analisi dei requisiti

*In questo capitolo viene trattata l'analisi dei requisiti del modulo di front-end della web-app, con i casi d'uso e requisiti fondamentali individuati.*

### 3.1 Descrizione generale

#### 3.1.1 Interfacce della web-app

##### **Interfaccia menù**

L'utente può visualizzare il menù del ristorante dopo aver scansionato il QR-code presente sui tavoli. Il menù è composto da un insieme di categorie, in ognuna di esse sono presenti dei piatti. I piatti sono ordinati in base al loro id, che è un numero univoco dentro ogni menù.

##### **Interfaccia lista degli ordini**

L'utente può vedere i piatti ordinati del suo tavolo, dove ci sono anche i piatti ordinati dalle altre persone dello stesso tavolo, inoltre può vedere i suoi piatti personali e i piatti in arrivo.

##### **Interfaccia gestione del tavolo**

In questa [maschera](#) un utente può creare la [sessione](#) del tavolo se non è stata già creata, altrimenti può visualizzare il QR-code della sua sessione per permettere ad altri utenti di partecipare.

##### **Interfaccia area personale**

Qui l'utente può effettuare il login e successivamente se ha delle allergie potrà inserire degli ingredienti nella blacklist per nascondere i piatti dal menù che contengono quest'ultimi.

#### 3.1.2 Caratteristiche degli utenti

In questa sezione vengono descritti tutte le caratteristiche degli utenti che possono utilizzare la web-app.

### Utente non autenticato

Con il termine utente non autenticato ci si riferisce ad una qualsiasi persona non autenticata nel sistema, che può sfruttare le funzionalità di base offerte dalla piattaforma, ossia:

- Visualizzare il menù del ristorante;
- Visualizzare i singoli piatti in modalità dettaglio;
- Creare una sessione di un tavolo;
- Partecipare ad una sessione già esistente di un tavolo;
- Uscire dalla sessione di un tavolo;
- Aggiungere piatti negli ordini;
- Aggiungere note ai piatti ordinati;
- Spostare ordini in arrivo;
- Marcare un piatto in arrivo come arrivato;
- Effettuare la registrazione nella piattaforma;
- Effettuare il login.

### Utente autenticato

Invece, con il termine “utente autenticato” ci si riferisce ad una persona registrata nel database e che ha effettuato l’accesso nella piattaforma, la quale, oltre a sfruttare le funzionalità dell’utente non autenticato, può anche:

- Aggiungere piatti nei preferiti;
- Rimuovere piatti dai preferiti;
- Dare una recensione ad un piatto;
- Aggiungere ingredienti alla blacklist;
- Rimuovere gli ingredienti dalla blacklist;
- Visualizzare la lista dei preferiti;
- Effettuare il logout.

### 3.1.3 Tecnologie utilizzate

Per sviluppare la piattaforma vengono utilizzate le seguenti tecnologie:

- Angular: per la creazione dell’interfaccia utente;
- HTML5: per creare la struttura dell’interfaccia utente;
- CSS3: per lo stile dell’interfaccia, viene utilizzata la sintassi SCSS;
- Stoplight: per simulare le chiamate REST [API](#);
- Angular Material: per la creazione dei componenti.

### 3.1.4 Descrizione delle tecnologie

#### Angular

Angular è un framework open source per sviluppare applicazioni web, permette di dividere l’applicazione in più componenti, grazie a questo è possibile riutilizzare lo stesso modulo in più parti della web-app ed inoltre garantisce una maggiore manutenibilità e espandibilità.



**Figura 3.1:** Logo di Angular

## HTML5

L'HyperText Markup Language, noto come HTML, è il linguaggio di markup più popolare, utilizzato per progettare le strutture dei siti web. Viene utilizzato da Angular per creare la struttura iniziale delle varie interfacce che verranno a seguito modificate da Angular per generare la pagina dinamicamente.



**Figura 3.2:** Logo di HTML e CSS

## CSS3

Il CSS, sigla di Cascading Style Sheets, è il linguaggio utilizzato per modificare il layout delle pagine web, le regole vengono applicate nell'ordine in cui vengono scritte. Per il progetto è stata utilizzata una sua estensione SCSS, la quale è compatibile con tutte le versioni di CSS. Tramite SCSS è possibile dichiarare le regole CSS in blocchi il che aiuta a scrivere regole CSS in modo più veloce e comprensibile.

## TypeScript

TypeScript è un linguaggio di programmazione open source sviluppato da Microsoft ed utilizzato da Angular, il quale estende il classico JavaScript perciò è in grado di eseguire qualsiasi programma scritto in JavaScript senza nessuna modifica. TypeScript rende molto più flessibile JavaScript aggiungendo la firma dei metodi, classi, tipi di dato e tanto altro, questo permette di effettuare controlli automatici e rilevare i bug in fase di compilazione. Grazie a TypeScript il codice generato da Angular è eseguibile su tutti i principali web browser come Google Chrome, Firefox, Safari, Opera, Microsoft Edge e tanti altri.

## Stoplight

Stoplight è una piattaforma che offre la possibilità di realizzare le API, grazie alla sua semplice interfaccia utente ne velocizza la progettazione. Permette di collaborare con gli altri, condividendo tutte le API e mostrando per ognuna di esse l'endpoint, la descrizione e la risposta.



**Figura 3.3:** Logo di Stoplight

### Angular Material

Angular Material è un insieme di componenti grafici web già implementati, questi componenti possono essere direttamente utilizzati in Angular. Tutti i componenti offerti sono [responsive](#) e liberamente modificabili.



**Figura 3.4:** Logo di Angular Material

## 3.2 Casi d'uso

### 3.2.1 Introduzione

In questa sezione verranno presentati i casi d'uso individuati durante la fase di analisi dei requisiti, i quali fanno riferimento a tutte le funzionalità che la web-app SushiLab dovrà offrire ad ogni utente che vorrà interfacciarsi con essa. Verranno elencati i casi d'uso principali con la propria descrizione, mentre i casi d'uso secondari e le descrizioni schematiche potranno essere reperiti nell'appendice [A](#).

### 3.2.2 Attori primari

- Utente non Autenticato: utente che non ha ancora effettuato l'autenticazione sulla piattaforma. Può essere in possesso o meno delle credenziali per l'autenticazione. Ha funzionalità limitate rispetto ad un utente autenticato;
- Utente Autenticato: utente che ha effettuato l'autenticazione alla piattaforma tramite le proprie credenziali. Ha accesso ad ogni funzionalità messa a disposizione dalla piattaforma;
- Utente Generico: può essere sia un utente autenticato che un utente non autenticato.

#### UC1 - Visualizza menù

La schermata di visualizzazione menù consente di effettuare operazioni relative all'acquisto dei piatti in base alla tipologia di utente che interagisce con la piattaforma. Un utente generico può visualizzare un piatto in due modalità quella normale e quella in dettaglio.



In quella normale vengono visualizzati il nome, la descrizione, gli ingredienti e il prezzo, ed è possibile modificare la quantità dei piatti ordinati. Nella modalità in dettaglio l'utente può visualizzare la valutazione media e, opzionalmente, può aggiungere una nota al piatto.

L'autenticazione dell'utente nel sistema permette svolgere ulteriori azioni, quali:

- L'aggiunta di un piatto alla lista dei preferiti per tenere traccia dei migliori piatti ordinati;
- La visualizzazione del menù calibrato sulla blacklist dell'utente;
- Recensire pubblicamente un piatto del menù con una valutazione da una a cinque stelle.

### UC2 - Gestione del tavolo

La schermata di gestione del tavolo consente di effettuare operazioni relative alla sessione del tavolo. Scansionando il QR-code l'utente visualizza la pagina per la gestione del tavolo che permette di creare la sessione all'interno della web-app o di partecipare a quella già esistente. Dopo aver creato la sessione l'utente può generare il relativo QR-code o un codice alfanumerico per permettere ad altre persone di unirsi. In ogni momento un utente può abbandonare la sessione in cui partecipa.

### UC3 - Lista degli ordini

La maschera della lista degli ordini si compone di tre sezioni, quali: [ordini del tavolo](#), [ordini in arrivo](#) e [ordini personali](#). La sezione degli ordini del tavolo permette di visualizzare gli ordini di tutti gli utenti partecipanti alla stessa sessione, questi rappresentano gli ordini non ancora inviati al ristorante e possono quindi essere modificati. L'utente può generare tramite l'app il QR-code degli ordini da scannerizzare per inviare l'ordine alla cucina. La lista degli ordini in arrivo permette di visualizzare i piatti in preparazione. Infine la sezione degli ordini personali contiene lo storico dei piatti ordinati durante la sessione in cui si trova l'utente, permettendogli di visualizzare tutte le caratteristiche di un piatto, la nota e la quantità ordinata.

### UC4 - Area personale

La pagina dell'area personale permette all'utente non autenticato di effettuare il login nella piattaforma tramite l'email e la password per accedere alle funzionalità complete dell'applicazione, di registrarsi su sushiLab e di effettuare il recupero delle credenziali. In seguito all'autenticazione l'utente può accedere e modificare la propria blacklist degli ingredienti.

## 3.3 Requisiti

In base a quanto definito nell'analisi dei requisiti è stata individuata una lista di requisiti che deve essere soddisfatta al termine del progetto di stage. Per ogni requisito individuato è stato assegnato un codice univoco, la descrizione del requisito e il caso d'uso riferito. Tutti i requisiti sono mostrati dentro una tabella e può essere reperita nell'appendice [B](#).

### 3.3.1 Panoramica

Un utente che accede al sistema deve potersi autenticare e registrare fornendo un'email e una password. L'utente deve poter accedere alla sezione del menù visualizzando tutti i piatti oppure quelli filtrati. I piatti del menù si possono visualizzare in modalità normale e dettagliata ed essere aggiunti agli ordini. L'utente deve essere in grado di creare, partecipare o uscire da una sessione di un tavolo. All'interno di una sessione l'utente deve poter visualizzare le liste degli ordini del tavolo, in arrivo e quelli personali mostrando per ogni ordine il nome, la descrizione e l'immagine dei piatti contenuti. Per ogni ordine del tavolo l'utente deve poter generare un codice-QR per l'invio di quest'ultimo. I piatti inviati vengono poi spostati nella lista degli ordini in arrivo. Un utente autenticato deve essere in grado di recensire un piatto, aggiungerlo alla propria lista dei preferiti e modificare la propria blacklist degli ingredienti. Infine in caso di smarrimento della password l'utente deve poter effettuare la richiesta di recupero fornendo l'email.

# Capitolo 4

## Progettazione e codifica

*In questo capitolo vengono approfondite la progettazione e la codifica. La progettazione entra nel dettaglio dell'architettura, delle viste progettate e delle API, mentre la codifica descrive il processo di sviluppo dei principali componenti e service dell'applicazione.*

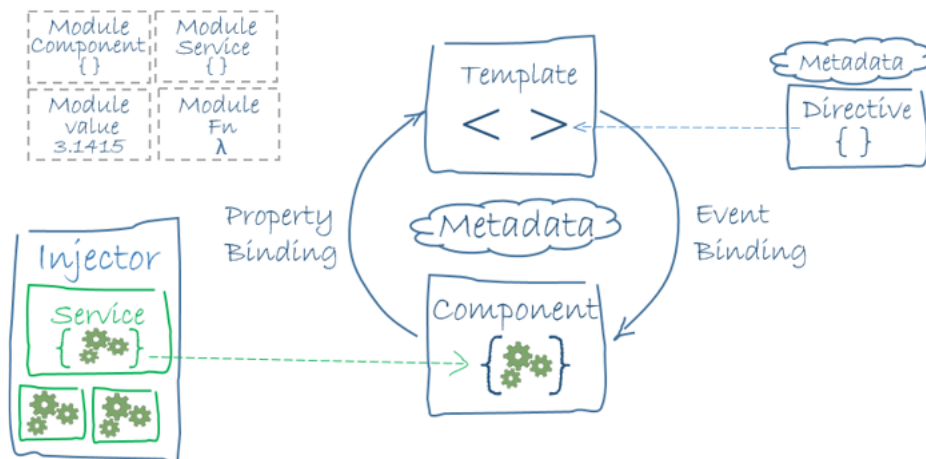
### 4.1 Progettazione

#### 4.1.1 Architettura Angular

Un'applicazione Angular è formata da un insieme di moduli, un modulo è un contenitore di funzionalità che permette di organizzare il codice di un'applicazione, può essere composto a sua volta da altri moduli oppure da uno o più componenti, i quali definiscono un elemento dell'interfaccia utente. Il modulo principale è chiamato root e per convenzione viene nominato "AppModule".

Ogni componente possiede un template HTML, che definisce la struttura della vista e i vari eventi che può emettere. Ogni evento emesso, se gestito, può modificare lo stato interno del componente, che si riflette nella struttura HTML e in fine scatena l'aggiornamento della vista.

I componenti utilizzano dei servizi che forniscono funzionalità specifiche, ad esempio il metodo login della classe Auth.Service, ma non sono correlate direttamente alla vista, essi sono inseriti come delle dipendenze, potenzialmente in più componenti e questo rende tali servizi modulari. Non solo i servizi sono riutilizzabili ma anche i componenti, ciò rende l'applicazione Angular più semplice da comprendere e manutenibile in futuro.



**Figura 4.1:** Architettura Angular

### 4.1.2 Architettura SushiLab

La web-app segue l'architettura spiegata precedentemente ed è anche quella consigliata dal sito ufficiale di Angular.

La cartella principale del progetto è "app/" che contiene tutti i componenti, i servizi e il root.

Per ogni componente è stata creata una cartella apposita, che contiene i file .ts per la logica, .html per la struttura, .scss per lo stile e infine i suoi componenti figli. Per i componenti condivisi si è creato una cartella "shared/" dove vengono salvati i componenti che sono utilizzati in più parti dell'applicazione.

Nella cartella REST vengono salvati tutti i file service, in cui ci sono metodi che vengono chiamati in più componenti dell'applicazione al fine di massimizzare il riuso del codice.

Nella cartella assets vengono salvate le immagini e le icone usate da tutti i componenti.

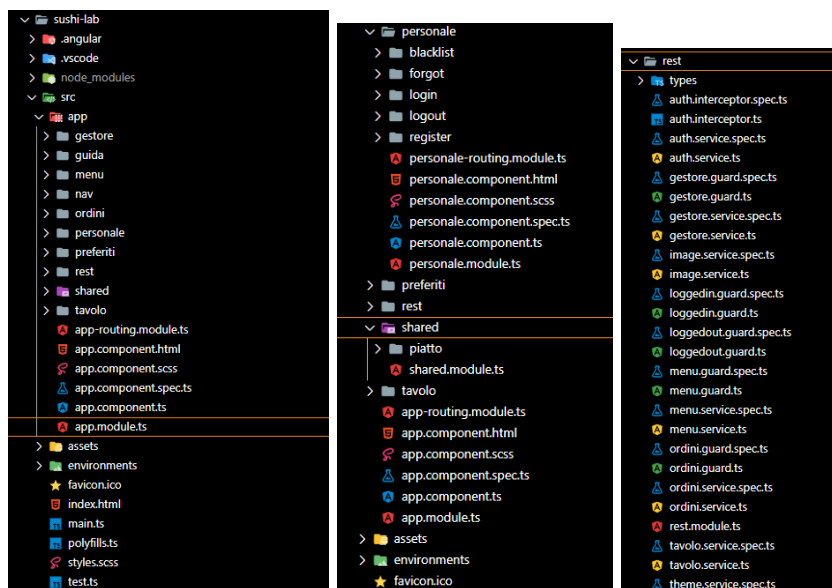


Figura 4.2: Struttura file SushiLab

### 4.1.3 Progettazione delle viste

All'inizio è stato fatto un meeting con il tutor aziendale per chiarire le funzionalità e i requisiti che la web-app deve avere, dopo di che è iniziata la progettazione dei [mock](#) delle viste tramite l'editor grafico online Figma.

Tramite il sistema di progettazione di Figma sono state create le bozze delle viste per posizionare gli elementi all'interno delle pagine e per stabilire la sequenza delle viste. Il posizionamento dei componenti è scelto in base alla frequenza dei click su di essi e seguendo lo stile delle web-app più popolari della stessa categoria. Sono stati decisi i seguenti aspetti:

- I colori principali e lo sfondo dell'applicazione;
- La posizione del bottone della navbar che è in alto a destra;
- La posizione del logo della piattaforma che si trova in alto a sinistra;
- I bottoni, il testo e i form devono avere lo stesso stile e colore in base alla loro funzionalità;
- Lo stile del piatto in modalità dettaglio in menù e nella lista degli ordini personali è la stessa.



Figura 4.3: Figma SushiLab

#### 4.1.4 Progettazione API

Durante il periodo di stage non era ancora presente il back-end per l'applicativo, quindi si è deciso con l'azienda di progettare dei mock API per il testing utilizzando la piattaforma Stoplight.

La progettazione delle API è stata facilitata dall'interfaccia semplice e user-friendly di Stoplight, che permette di definire rapidamente i path, il body e i rispettivi metodi, e grazie anche ai mock-up delle viste prima definite.

Per ogni chiamata REST bisogna definire:

- **Nome:** identifica l'API;
- **Descrizione:** spiega in dettaglio la funzione della API;
- **Metodo:** definisce il tipo di chiamata, è stato usato:
  - **get:** per richiedere dei dati al server, come la chiamata per ottenere il menù;
  - **post:** per inviare dei dati sintetici al server, come i dati per il login;
  - **delete:** per eliminare dei dati dal server, come l'eliminazione della sessione del tavolo.
- **Path:** definisce il percorso per chiamare l'API;
- **Risposta:** configura la risposta che ritorna l'API, è stato usato:
  - **200:** richiesta andata a buon fine;
  - **201:** creazione andata a buon fine;
  - **204:** richiesta andata a buon fine ma il contenuto è vuoto;
  - **401:** non autorizzato;
  - **404:** non trovato;
  - **406:** non accettato;
  - **500:** errore interno.

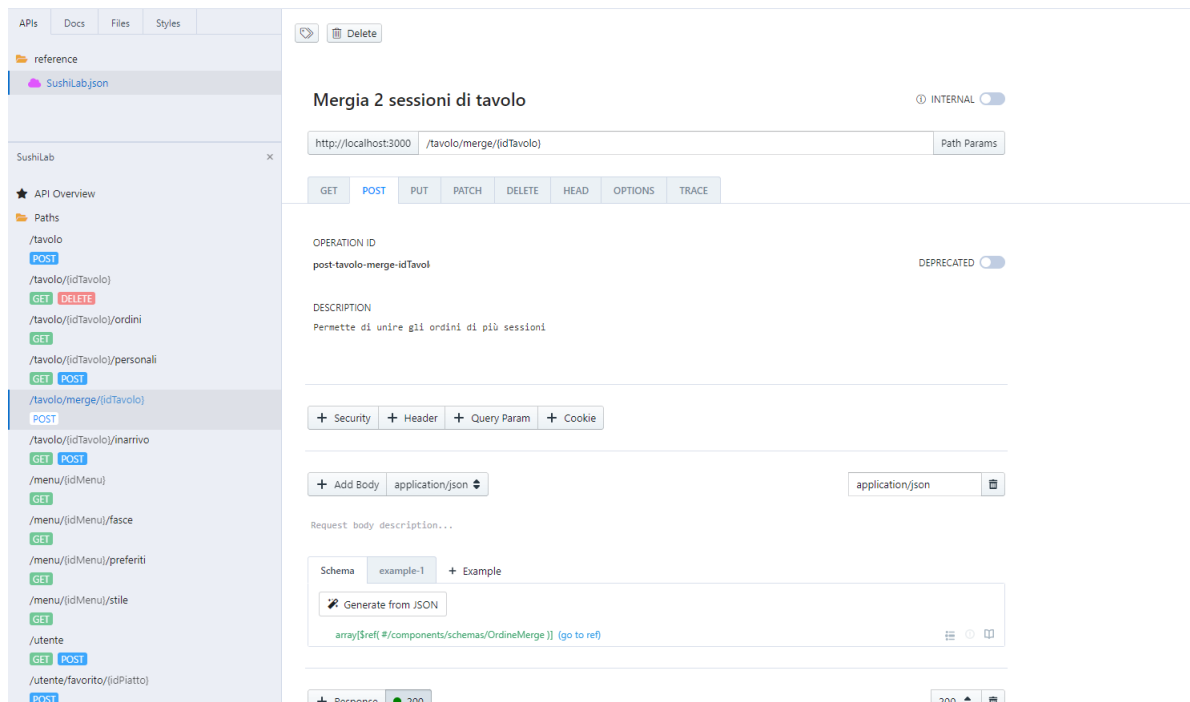


Figura 4.4: Stoplight Sushilab

### 4.1.5 Progettazione dei componenti

I componenti sono stati individuati tramite l'analisi dei requisiti e la progettazione delle viste. Ogni componente ha delle funzionalità specifiche e tutti assieme costituiscono la web-app, coprendo tutti i requisiti richiesti.

## 4.2 Codifica

### 4.2.1 Interfacce

Dopo aver progettato i mock della visuale tramite figma e le API tramite Stoplight è iniziata la fase di codifica delle interfacce. La creazione dei template di default dei componenti è stata molto rapida grazie al command-line interface di Angular, utilizzando il comando 'ng generate component "nome del componente"' che crea una cartella contenente tutti i file necessari per il rendering del componente. Dopo la creazione dei file principali si è continuato a modificare il component.ts, che rappresenta la logica del componente, inserendo tutte le funzionalità che quest'ultimo deve avere. Al termine della modifica si è passati alla struttura e allo stile, quindi a modificare il component.html e il component.css. Tutte le interfacce, con le loro funzionalità e una descrizione dettagliata, possono essere reperite nell'appendice C.

### 4.2.2 Componenti service

Per la comunicazione e gestione dei vari componenti vengono adoperati i file service. Per ogni servizio viene creato un file.service utilizzabile da uno o più component.

I servizi utilizzati sono:

- auth.service: per gestire le autenticazioni;
- menu.service: per gestire tutte le funzionalità del menù;
- ordini.service: per gestire le ordinazioni;
- tavolo.service: per gestire le sessioni di tavolo;
- user.service: per gestire le funzionalità dell'area personale.

### 4.2.3 Componenti guard

Arrivato alla fine della codifica si definiscono i componenti guard, che permettono di imporre delle regole di accesso per la navigazione in un determinato componente. La verifica delle regole di guard avviene all'interno del metodo canActivate, appoggiandosi ai diversi service e ai dati locali di sessione.



# Capitolo 5

## Verifica e validazione

*In questa sezione vengono elencati i test di validazione effettuati e una descrizione della validazione.*

### 5.1 Verifica

Viene riportata una tabella dei test di unità sui vari componenti e sulle loro funzionalità, per ogni test viene assegnato un codice, il requisito associato, una descrizione e lo stato.

| Test | Requisito | Descrizione  | Stato   |
|------|-----------|--|---------|
| T1   | ROF1      | Il componente menù viene creato  | Passato |
| T2   | ROF3      | Il componente piatto viene creato  | Passato |
| T3   | ROF4      | La quantità viene aumentata  | Passato |
| T4   | ROF5      | La quantità viene diminuita  | Passato |
| T5   | ROF6      | Il componente piatto viene visualizzato in modalità dettaglio            | Passato |
| T6   | ROF8      | Il componente gestione del tavolo viene creato                           | Passato |
| T7   | ROF9      | La sessione viene generata   | Passato |
| T8   | ROF10     | L'utente viene unito ad una sessione                                     | Passato |
| T9   | ROF11     | L'utente esce dalla sessione   | Passato |
| T10  | ROF12     | Il QR-code del tavolo viene generato                                     | Passato |
| T11  | ROF13     | Il componente gestione lista degli ordini viene creato                   | Passato |
| T12  | ROF17     | La lista degli ordini del tavolo viene spostata in arrivo                | Passato |
| T13  | ROF18     | Il QR-code della lista degli ordini viene generato                       | Passato |
| T14  | ROF20     | Il piatto della lista degli ordini in arrivo viene marcato come arrivato | Passato |

| Test | Requisito | Descrizione   | Stato   |
|------|-----------|---|---------|
| T15  | ROF21     | Il componente area personale viene creato                                     | Passato |
| T16  | ROF22     | Il componente register viene creato   | Passato |
| T17  | ROF22     | I campi sbagliati del form di registrazione vengono evidenziati               | Passato |
| T18  | ROF22     | Il submit del form deve registrare l'utente                                   | Passato |
| T19  | ROF23     | Il componente login viene creato  | Passato |
| T20  | ROF23     | I campi sbagliati del form di login vengono evidenziati                       | Passato |
| T21  | ROF23     | Il submit del form deve effettuare il login dell'utente                       | Passato |
| T22  | ROF24     | Il componente forgot viene creato   | Passato |
| T23  | ROF24     | I campi sbagliati del form per il recupero della password vengono evidenziati | Passato |
| T24  | ROF24     | Il submit del form deve reimpostare la password                               | Passato |
| T25  | ROF25     | Il componente logout viene creato   | Passato |
| T26  | ROF25     | Il click sul componente logout deve effettuare il logout dell'utente          | Passato |
| T27  | ROF26     | Il componente blacklist viene creato  | Passato |
| T28  | ROF26     | L'ingrediente viene inserito nella blacklist                                  | Passato |
| T29  | ROF27     | L'ingrediente viene rimosso dalla blacklist                                   | Passato |
| T30  | ROF28     | Il piatto viene aggiunto alla lista dei preferiti                             | Passato |
| T31  | ROF29     | Il piatto viene rimosso dalla lista dei preferiti                             | Passato |
| T32  | ROF30     | Il componente star viene creato   | Passato |
| T34  | ROF30     | La recensione viene generata correttamente                                    | Passato |
| T35  | ROF31     | Il componente lista preferiti viene creato                                    | Passato |

**Tabella 5.1:** Tabella dei test sui requisiti.

## 5.2 Validazione e collaudo

Nell'ultima settimana dello stage ho controllato tutte le funzionalità implementate durante l'intero progetto, dove ho testato tutte le attività che sono svolgibili da un utente generale. Infine è stato mostrato il prodotto finale del progetto sushiLab al tutor aziendale con una demo completa di tutte le funzionalità della web-app.

# Capitolo 6

## Conclusioni

### 6.1 Raggiungimento degli obiettivi

#### 6.1.1 Obiettivi fissati

Tutti gli obiettivi fissati all'inizio dello stage con il tutor aziendale sono stati raggiunti, oltre ai requisiti obbligatori sono stati soddisfatti anche i requisiti desiderabili e i facoltativi. Infatti la web-app fornisce tutte le funzionalità richieste e offre inoltre la possibilità di salvare i piatti nella lista dei preferiti, filtrare il menù rimuovendo i piatti contenenti gli ingredienti presenti nella blacklist e fornire una recensione ad un piatto.

| Obiettivi fissati |  |          |
|-------------------|--|----------|
| Codice            | Descrizione  | Esito    |
| O01               | Acquisizione competenze sulle tematiche sopra descritte  | Superato |
| O02               | Capacità di raggiungere gli obiettivi richiesti in autonomia seguendo il cronoprogramma            | Superato |
| O03               | Portare a termine le implementazioni previste con una percentuale di superamento pari al 80%       | Superato |
| D01               | Portare a termine le implementazioni previste con una percentuale di superamento pari al 100%      | Superato |
| F01               | Apportare un valore aggiunto al gruppo di lavoro durante le fasi di progettazione delle interfacce | Superato |

**Tabella 6.1:** Tabella del raggiungimento degli obiettivi fissati.

### 6.1.2 Svolgimento del lavoro

I lavori pianificati per le prime settimane di stage sono stati svolti senza particolari difficoltà, infatti le attività di studio e formazione sono terminate con un piccolo anticipo rispetto alla pianificazione e il tempo risparmiato è stato utilizzato alla fine per implementare le funzionalità aggiuntive. Questo guadagno di tempo è dovuto ad una buona familiarità con i linguaggi di programmazione acquisita durante gli anni di studio nell'ambito dell'informatica.

La fase di progettazione ed implementazione delle maschere è risultata più rapida del previsto anche grazie alla mia esperienza pregressa come cameriere in un ristorante sushi.

La parte che ha impiegato più ore di quelle previste è stata la fase di integrazione tra le varie maschere dove bisognava mantenere i dati salvati di ogni pagina, questo problema è stato risolto grazie ai componenti service forniti da Angular e al local storage del browser. Un altro strumento che ha dato un grande aiuto è stato discord perché ci ha permesso di comunicare facilmente con altri collaboratori del progetto.

| Settimana         | Attività   |
|-------------------|--|
| Prima settimana   | <ul style="list-style-type: none"> <li>• Incontro con il tutor aziendale e i collaboratori</li> <li>• Identificazione dei requisiti</li> <li>• Formazione su strumenti di lavoro</li> <li>• Ripasso HTML5, JavaScript e css</li> </ul>                 |
| Seconda settimana | <ul style="list-style-type: none"> <li>• Formazione su Spring Core</li> <li>• Formazione su Spring Boot</li> <li>• Formazione su linguaggio TypeScript</li> <li>• Formazione su Angular components</li> </ul>  |
| Terza settimana   | <ul style="list-style-type: none"> <li>• Formazione su Angular data binding e pipes</li> <li>• Formazione su Angular routing e forms</li> <li>• Formazione su Angular lifecycle hooks</li> <li>• Analisi dei requisiti</li> </ul>                      |
| Quarta settimana  | <ul style="list-style-type: none"> <li>• Progettazione delle interfacce tramite Figma</li> <li>• Identificazione dei colori della web-app</li> <li>• Formazione sulla libreria Angular material</li> <li>• Progettazione maschera d'accesso</li> </ul> |

| Settimana         | Attività   |
|-------------------|--|
| Quinta settimana  | <ul style="list-style-type: none"><li>• Progettazione navbar</li><li>• Progettazione maschera menù</li><li>• Progettazione maschera gestione del tavolo</li><li>• Progettazione maschera gestione degli ordini</li><li>• Progettazione API</li></ul> |
| Sesta settimana   | <ul style="list-style-type: none"><li>• Progettazione componenti service</li><li>• Realizzazione funzionamento merge ordini</li><li>• Realizzazione funzionamento sposta ordini</li><li>• Progettazione componenti guard</li></ul>                   |
| Settima settimana | <ul style="list-style-type: none"><li>• Integrazione dei componenti</li><li>• Decorazione finale della web-app</li></ul>   |
| Ottava settimana  | <ul style="list-style-type: none"><li>• Verifica e validazione della web-app</li><li>• Documentazione analisi tecnica</li><li>• Collaudo finale</li></ul>  |

**Tabella 6.2:** Tabella del consuntivo di lavoro.

## 6.2 Conoscenze acquisite

Durante lo stage nell'azienda SyncLab, ho acquisito nuove conoscenze tecniche. Ho imparato ad usare Angular che è uno dei più famosi framework per lo sviluppo front-end nelle applicazioni. L'apprendimento di Angular è stato svolto principalmente in autonomia, consultando vari siti web e video a tema, ottenendo così una maggiore comprensione del framework.

Durante la fase di testing ho approfondito il framework Spring di Java al fine di progettare dei mock delle [API](#) che rispecchiano il più possibile l'implementazione del back-end futuro.

Lavorare in Angular mi ha dato la possibilità di imparare TypeScript, ciò mi ha permesso di avere una visione più ampia dei problemi che riguardano JavaScript, come il sistema dei tipi e il processo di debugging.

Ogni settimana è stato fatto un incontro in sede con il tutor aziendale, dove si è discusso l'avanzamento della web-app, le difficoltà e la ripianificazione della settimana seguente in caso di cambiamenti, ciò mi ha permesso di familiarizzare con il metodo Agile.

## 6.3 Valutazione

### 6.3.1 L'azienda

Durante il processo di sviluppo tutte le persone dell'azienda sono state molto disponibili, inizialmente mi hanno illustrato il back-end di un grande progetto fornendomi consigli su come scrivere codice al fine di facilitare il debugging. Il tutor aziendale è sempre stato molto gentile e mi ha dato la possibilità di lavorare sul front-end, che è sempre stato il mio ambito di sviluppo preferito.

Infine tutto ciò mi ha permesso di imparare a lavorare con scadenze rigorose come accade dentro ad un team vero e proprio.

### 6.3.2 Personale

Lo stage mi ha ampiamente soddisfatto, mi ha fatto conoscere meglio l'ambiente di lavoro di un programmatore permettendomi di usare tutte le competenze acquisite durante il corso di studio universitario.

Durante il tirocinio ho visto l'intero processo di sviluppo di un applicativo, partendo da un problema reale arrivando all'implementazione della web-app, questo mi è stato particolarmente importante per il mio avanzamento professionale da programmatore. Arrivando alla fine, l'intera esperienza dello stage è più che positiva, ritengo che il tirocinio sia molto importante per uno studente, poiché permette di sfruttare e valutare le proprie conoscenze acquisite e in particolare per capire e riflettere sul percorso di studio scelto, infatti lo stage mi ha confermato di voler continuare la carriera lavorativa nell'ambito dello sviluppo.

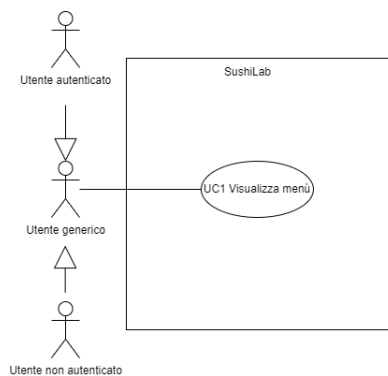
## Appendice A

# Descrizione dettagliata dei casi d'uso

In questa sezione verranno descritti tutti i casi d'uso dettagliamente. La descrizione generale dei casi d'uso è reperibile alla sezione 3.2.

### Utente generico

#### UC1 - Visualizza menù



**Figura A.1:** Use Case - UC 1

- **Descrizione:** L'utente visualizza il menù del ristorante.
- **Attore Primario:** L'utente generico.
- **Precondizione:** L'utente si trova dentro la web-app.
- **Postcondizione:** Viene visualizzato il menù del ristorante.
- **Scenario principale:**
  - L'utente si trova dentro il sistema;
  - L'utente clicca sul bottone menù;
  - Viene mostrato il menù del ristorante.

## UC1.1 - Visualizza categorie

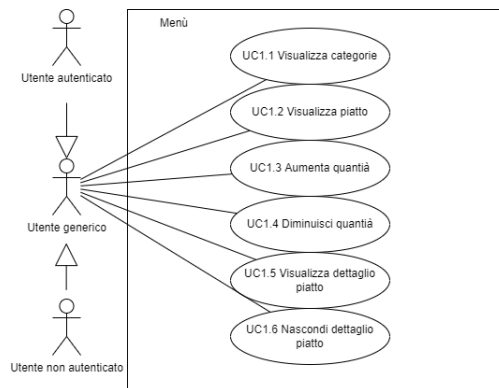


Figura A.2: Use Case - UC 1.1, UC 1.2, UC 1.3, UC 1.4, UC 1.5, UC 1.6

- **Descrizione:** L'utente visualizza le categorie del menù.
- **Attore Primario:** L'utente generico.
- **Precondizione:** L'utente si trova dentro la sezione menù.
- **Postcondizione:** Vengono visualizzati i nomi delle categorie.
- **Scenario principale:**
  - L'utente si trova nella sezione menù;
  - Vengono mostrate le categorie del menù.

## UC1.2 - Visualizza piatto

- **Descrizione:** L'utente visualizza i piatti del menù mostrando il numero, nome, prezzo, ingredienti, allergeni, limitazioni e la quantità. La quantità di default è 0 ed indica che il piatto non è ancora stato ordinato.
- **Attore Primario:** L'utente generico.
- **Precondizione:** L'utente si trova dentro la sezione menù.
- **Postcondizione:** Vengono visualizzati i piatti del menù.
- **Scenario principale:**
  - L'utente si trova nella sezione menù;
  - Vengono mostrati i piatti del menù.

## UC1.3 - Aumenta quantità

- **Descrizione:** L'utente aumenta la quantità di un piatto.
- **Attore Primario:** L'utente generico.
- **Precondizione:** L'utente si trova nella sezione menù o lista degli ordini personali.
- **Postcondizione:** Viene aggiunto il piatto specifico con la quantità aggiornata negli ordini.
- **Scenario principale:**
  - L'utente si trova nella sezione menù;
  - L'utente clicca sul bottone + di un piatto;
  - Viene aggiunto il piatto negli ordini.
- **Scenario alternativo:**
  - L'utente si trova nella sezione menù o nella lista degli ordini personali;



- L'utente clicca sul bottone + di un piatto che è già presente negli ordini;
- Viene aumentata la quantità del piatto negli ordini.

#### UC1.4 - Diminuisce quantità

- **Descrizione:** L'utente diminuisce la quantità di un piatto.
- **Attore Primario:**L'utente generico.
- **Precondizione:** L'utente si trova dentro la sezione menù o nella lista degli ordini personali.
- **Postcondizione:** Viene diminuita la quantità del piatto specifico negli ordini.
- **Scenario principale:**
  - L'utente si trova nella sezione menù o nella lista degli ordini personali;
  - L'utente clicca sul bottone - di un piatto con quantità maggiore di 1;
  - Viene diminuita la quantità del piatto negli ordini.
- **Scenario alternativo:**
  - L'utente si trova nella sezione menù o nella lista degli ordini personali;
  - L'utente clicca sul bottone - di un piatto con quantità uguale a 1;
  - Viene rimosso il piatto dagli ordini.

#### UC1.5 - Visualizza dettaglio piatto

- **Descrizione:** L'utente visualizza i dettagli di un piatto nel menù, mostrando la recensione del piatto e il text-box per inserire una nota.
- **Attore Primario:**L'utente generico.
- **Precondizione:** L'utente si trova dentro la sezione menù.
- **Postcondizione:** Vengono visualizzati i dettagli di un piatto specifico.
- **Scenario principale:**
  - L'utente si trova nella sezione menù;
  - L'utente clicca sul bottom mostra dettagli;
  - Vengono mostrati i dettagli di un piatto.

#### UC1.6 - Nascondi dettaglio piatto

- **Descrizione:** L'utente nasconde i dettagli di un piatto specifico.
- **Attore Primario:**L'utente generico.
- **Precondizione:** L'utente si trova dentro la sezione menù con un piatto in modalità dettaglio.
- **Postcondizione:** Vengono nascosti i dettagli del piatto specifico.
- **Scenario principale:**
  - L'utente si trova nella sezione menù;
  - L'utente clicca sul bottom nascondi dettagli;
  - Vengono nascosti i dettagli del piatto.

## UC2 - Gestione del tavolo

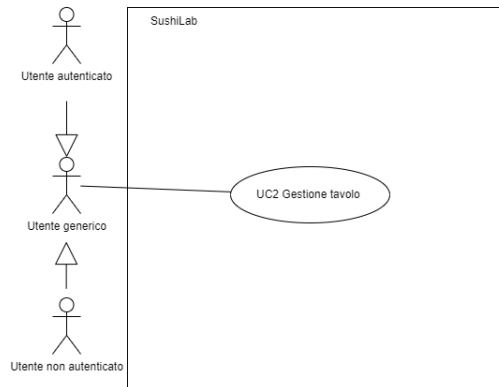


Figura A.3: Use Case - UC 2

- **Descrizione:** L'utente visualizza la maschera di gestione del tavolo.
- **Attore Primario:** L'utente generico.
- **Precondizione:** L'utente si trova dentro la web-app.
- **Postcondizione:** Viene visualizzata la maschera di gestione del tavolo.
- **Scenario principale:**
  - L'utente si trova dentro il sistema;
  - L'utente clicca sul bottone gestione del tavolo nella navbar;
  - Viene mostrata la maschera di gestione del tavolo.

## UC2.1 - Generazione sessione tavolo

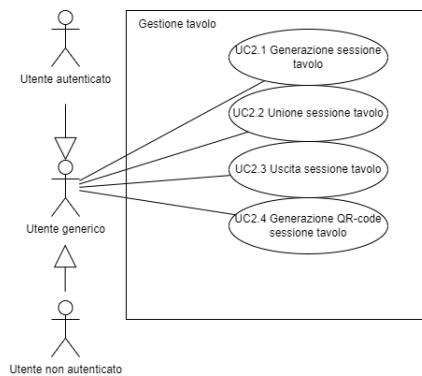


Figura A.4: Use Case - UC 2.1, UC 2.2, UC 2.3, UC 2.4

- **Descrizione:** L'utente genera la sessione del tavolo.
- **Attore Primario:** L'utente generico.
- **Precondizione:** L'utente si trova dentro la sezione gestione del tavolo.
- **Postcondizione:** L'utente entra nella sessione generata del tavolo.
- **Scenario principale:**
  - L'utente si trova dentro la sezione gestione del tavolo;
  - L'utente clicca sul bottone crea sessione;

- Viene creata la sessione;
- L'utente viene inserito nella sessione creata.

#### UC2.2 - Unione sessione tavolo

- **Descrizione:** L'utente si unisce alla sessione del tavolo.
- **Attore Primario:**L'utente generico.
- **Precondizione:** L'utente si trova dentro la sezione gestione del tavolo.
- **Postcondizione:** L'utente entra nella sessione che è stata inserita.
- **Scenario principale:**
  - L'utente si trova dentro la sezione gestione del tavolo;
  - L'utente clicca sul bottone unisciti a una sessione;
  - L'utente inserisce il numero della sessione;
  - L'utente clicca sul bottone unisciti;
  - L'utente viene inserito nella sessione.

#### UC2.3 - Uscita sessione tavolo

- **Descrizione:** L'utente esce dalla sessione del tavolo.
- **Attore Primario:**L'utente generico.
- **Precondizione:** L'utente si trova dentro la sezione gestione del tavolo ed è dentro ad una sessione.
- **Postcondizione:** L'utente esce dalla sessione del tavolo.
- **Scenario principale:**
  - L'utente si trova dentro la sezione gestione del tavolo;
  - L'utente clicca sul bottone esci dalla sessione;
  - L'utente viene rimosso dalla sessione.

#### UC2.4 - Generazione QR-code sessione tavolo

- **Descrizione:** L'utente genera il QR-code dalla sessione del tavolo per mostrarlo agli altri al fine di farli unire scansionando il codice-QR.
- **Attore Primario:**L'utente generico.
- **Precondizione:** L'utente si trova dentro la sezione gestione del tavolo ed è dentro ad una sessione.
- **Postcondizione:** L'utente genera il QR-code dalla sessione del tavolo.
- **Scenario principale:**
  - L'utente si trova dentro la sezione gestione del tavolo;
  - L'utente genera il QR-code della sessione;
  - Viene mostrato il QR-code;

## UC3 - Lista degli ordini

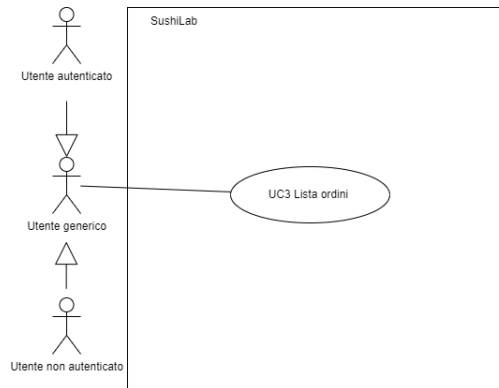


Figura A.5: Use Case - UC 3

- **Descrizione:** L'utente visualizza la maschera di gestione degli ordini.
- **Attore Primario:** L'utente generico.
- **Precondizione:** L'utente si trova dentro ad una sessione del tavolo.
- **Postcondizione:** Viene visualizzata la maschera di gestione degli ordini.
- **Scenario principale:**
  - L'utente si trova dentro il sistema con una sessione del tavolo attiva;
  - L'utente clicca sul bottone lista degli ordini nella navbar;
  - Viene mostrata la maschera di gestione degli ordini.

## UC3.1 - Visualizza lista degli ordini del tavolo

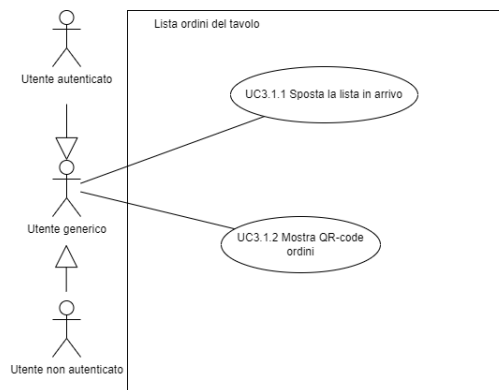


Figura A.6: Use Case - UC 3.1, UC 3.2, UC 3.3

- **Descrizione:** L'utente visualizza la lista degli ordini della sessione del tavolo in cui si trova.
- **Attore Primario:** L'utente generico.
- **Precondizione:** L'utente si trova dentro la sezione lista degli ordini.
- **Postcondizione:** Viene visualizzata la lista degli ordini del tavolo.
- **Scenario principale:**

- L'utente si trova dentro la sezione gestione degli ordini;
- L'utente clicca sul bottone "tavolo";
- Viene mostrata la lista dei piatti ordinati del tavolo.

### UC3.2 - Visualizza lista degli ordini personali

- **Descrizione:** L'utente visualizza la lista degli ordini personali.
- **Attore Primario:**L'utente generico.
- **Precondizione:** L'utente si trova dentro la sezione lista degli ordini.
- **Postcondizione:** Viene visualizzata la lista degli ordini personali.
- **Scenario principale:**
  - L'utente si trova dentro la sezione gestione degli ordini;
  - L'utente clicca sul bottone "personali";
  - Viene mostrata la lista dei piatti ordinati dall'utente stesso.

### UC3.3 - Visualizza lista degli ordini in arrivo

- **Descrizione:** L'utente visualizza la lista degli ordini in arrivo.
- **Attore Primario:**L'utente generico.
- **Precondizione:** L'utente si trova dentro la sezione lista degli ordini.
- **Postcondizione:** Viene visualizzata la lista degli ordini in arrivo.
- **Scenario principale:**
  - L'utente si trova dentro la sezione gestione degli ordini;
  - L'utente clicca sul bottone "in arrivo";
  - Viene mostrata la lista dei piatti in arrivo.

#### UC3.1.1 - Sposta la lista in arrivo

- **Descrizione:** L'utente sposta la lista degli ordini in arrivo.
- **Attore Primario:**L'utente generico.
- **Precondizione:** L'utente si trova dentro la sezione lista degli ordini del tavolo.
- **Postcondizione:** Viene spostata la lista degli ordini del tavolo in arrivo.
- **Scenario principale:**
  - L'utente si trova dentro la sezione gestione degli ordini del tavolo;
  - L'utente clicca sul bottone sposta la lista in arrivo;
  - Viene spostata la lista dei piatti ordinati personali in modalità dettaglio;
  - Viene mostrato all'utente il messaggio "ordini spostati correttamente".

#### UC3.1.2 - Mostra QR-code ordini

- **Descrizione:** L'utente genera il QR-code della lista degli ordini per dopo mostrarlo al cameriere.
- **Attore Primario:**L'utente generico.
- **Precondizione:** L'utente si trova dentro la sezione lista degli ordini del tavolo.
- **Postcondizione:** Viene mostrato il QR-code della lista degli ordini.
- **Scenario principale:**
  - L'utente si trova dentro la sezione gestione degli ordini del tavolo;
  - L'utente clicca sul bottone QR-code;
  - Viene generato il QR-code degli ordini;
  - Viene mostrato il QR-code.

## UC3.2.1 - Visualizza in dettaglio lista degli ordini personali

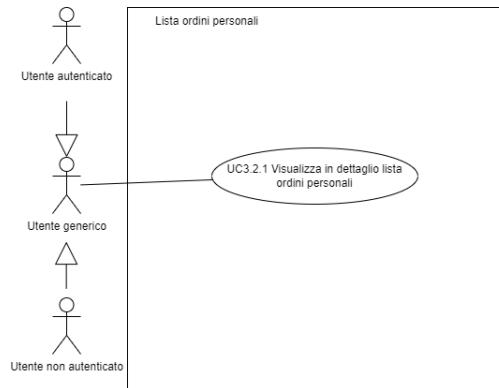


Figura A.7: Use Case - UC 3.2.1

- **Descrizione:** L'utente visualizza la lista degli ordini personali in modalità dettaglio.
- **Attore Primario:**L'utente generico.
- **Precondizione:** L'utente si trova dentro la sezione lista degli ordini personali.
- **Postcondizione:** Viene visualizzata la lista degli ordini personali con i piatti in modalità dettaglio.
- **Scenario principale:**
  - L'utente si trova dentro la sezione gestione degli ordini personali;
  - L'utente clicca sul bottone "lente" con il +;
  - Viene mostrata la lista dei piatti ordinati personali in modalità dettaglio.

## UC3.3.1 - Ricezione piatto

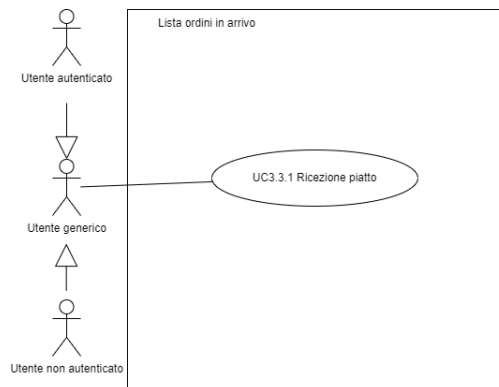


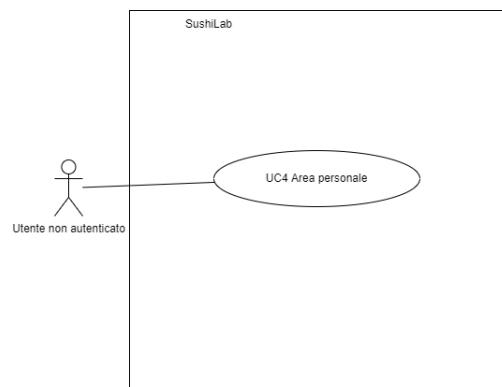
Figura A.8: Use Case - UC 3.3.1

- **Descrizione:** L'utente marca un piatto in arrivo come ricevuto.
- **Attore Primario:**L'utente generico.
- **Precondizione:** L'utente si trova dentro la sezione gestione degli ordini in arrivo e ha almeno un piatto nella lista in arrivo.

- **Postcondizione:** L'utente marca il piatto come arrivato diminuendo di 1 la sua quantità.
- **Scenario principale:**
  - L'utente si trova dentro la sezione gestione lista degli ordini in arrivo;
  - L'utente clicca sul bottone "v" di un piatto;
  - Viene diminuita di 1 la sua quantità.
- **Scenario alternativo:**
  - L'utente si trova dentro la sezione gestione lista degli ordini in arrivo;
  - L'utente clicca sul bottone "v" di un piatto con quantità uguale a 1;
  - Viene diminuita di 1 la quantità del piatto e viene disabilitato il bottone.

## Utente non autenticato

### UC4 - Area personale



**Figura A.9:** Use Case - UC 4

- **Descrizione:** L'utente visualizza la maschera dell'area personale.
- **Attore Primario:** L'utente non autenticato.
- **Precondizione:** L'utente si trova dentro la web-app.
- **Postcondizione:** Viene visualizzata la maschera dell'area personale.
- **Scenario principale:**
  - L'utente si trova dentro il sistema;
  - Viene mostrata la maschera dell'area personale.

## UC4.1 - Registrazione

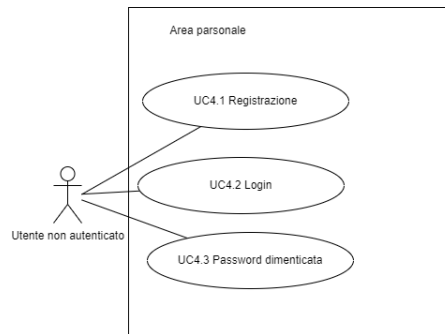


Figura A.10: Use Case - UC 4.1, UC 4.2, UC 4.3

- **Descrizione:** L'utente viene registrato nella piattaforma.
- **Attore Primario:** L'utente non autenticato.
- **Precondizione:** L'utente si trova dentro la web-app.
- **Postcondizione:** Vengono salvati i dati dell'utente inseriti durante la fase di registrazione nel data-base.
- **Scenario principale:**
  - L'utente si trova dentro l'area personale;
  - L'utente clicca sul bottone registrati;
  - Viene mostrato il form di registrazione;
  - L'utente inserisce l'email;
  - L'utente inserisce la password;
  - L'utente ripete la password;
  - L'utente clicca sul bottone registrati;
  - Viene registrato correttamente l'account.

## UC4.2 - Login

- **Descrizione:** L'utente effettua il login nella piattaforma.
- **Attore Primario:** L'utente non autenticato.
- **Precondizione:** L'utente si trova dentro la web-app.
- **Postcondizione:** Viene effettuato il login.
- **Scenario principale:**
  - L'utente si trova dentro l'area personale;
  - L'utente inserisce l'email;
  - L'utente inserisce la password;
  - L'utente clicca sul bottone login;
  - Viene effettuato il login correttamente.

## UC4.3 - Password dimenticata

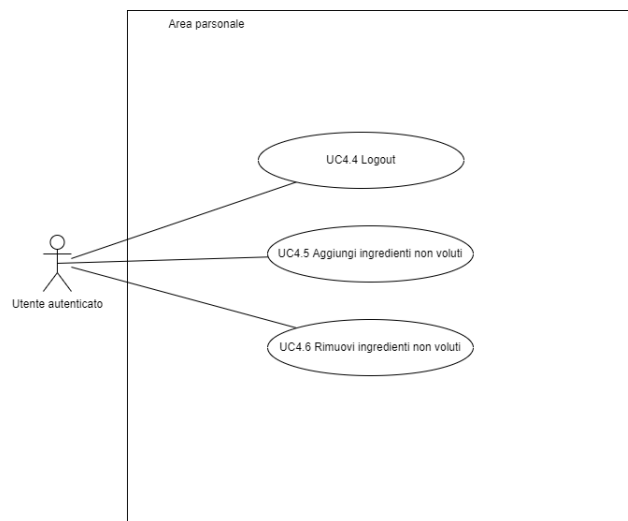
- **Descrizione:** L'utente reimposta la password del proprio account.
- **Attore Primario:** L'utente non autenticato.
- **Precondizione:** L'utente si trova dentro la web-app.
- **Postcondizione:** Viene aggiornata la nuova password nel data-base.
- **Scenario principale:**



- L'utente si trova dentro l'area personale;
- L'utente clicca sul bottone password dimenticata;
- Viene mostrato il form di recupero password;
- L'utente inserisce l'email;
- L'utente clicca sul bottone ottieni codice;
- L'utente arriva nel secondo form tramite il link mandato tramite l'email;
- L'utente inserisce la password;
- L'utente ripete la password;
- L'utente clicca sul bottone cambia password;
- Viene cambiata correttamente la password.

## Utente autenticato

### UC4.4 - Logout



**Figura A.11:** Use Case - UC 4.4, UC4.5, UC4.6

- **Descrizione:** L'utente effettua il logout.
- **Attore Primario:** L'utente autenticato.
- **Precondizione:** L'utente si trova dentro la web-app ed ha effettuato il login.
- **Postcondizione:** Viene effettuato il logout.
- **Scenario principale:**
  - L'utente si trova dentro l'area personale;
  - L'utente clicca sul bottone logout;
  - Viene effettuato il logout dell'utente.

### UC4.5 - Aggiungi ingredienti non voluti

- **Descrizione:** L'utente inserisce un ingrediente nella blacklist.
- **Attore Primario:** L'utente autenticato.
- **Precondizione:** L'utente si trova dentro la web-app e ha effettuato il login.
- **Postcondizione:** Viene inserito l'ingrediente nella blacklist.
- **Scenario principale:**

- L'utente si trova dentro l'area personale;
- L'utente clicca sul bottone blacklist degli ingredienti;
- L'utente inserisce il nome dell'ingrediente;
- L'utente clicca sul bottone +;
- Viene inserito l'ingrediente nella blacklist.

#### UC4.6 - Rimuovi ingredienti non voluti

- **Descrizione:** L'utente rimuove un ingrediente dalla blacklist.
- **Attore Primario:** L'utente autenticato.
- **Precondizione:** L'utente si trova dentro la web-app ed ha effettuato il login.
- **Postcondizione:** Viene rimosso l'ingrediente dalla blacklist.
- **Scenario principale:**
  - L'utente si trova dentro l'area personale;
  - L'utente clicca sul bottone blacklist degli ingredienti;
  - L'utente clicca sul bottone - di un ingrediente già esistente;
  - Viene rimosso l'ingrediente dalla blacklist.

#### UC1.7 - Aggiungi preferiti

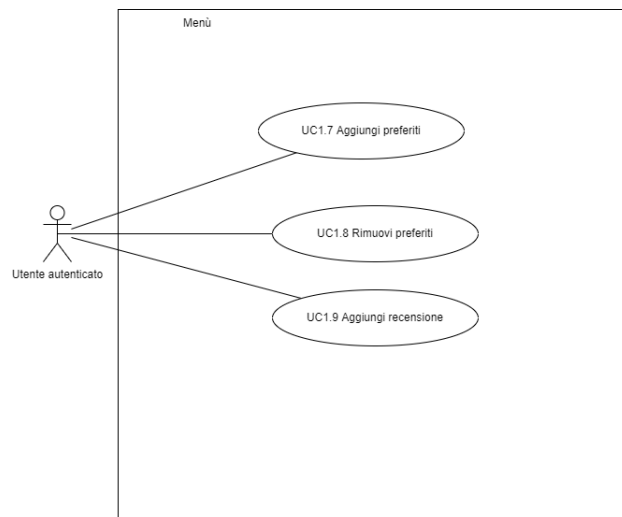


Figura A.12: Use Case - UC 1.7, UC1.8, UC1.9

- **Descrizione:** L'utente aggiunge un piatto alla lista dei preferiti.
- **Attore Primario:** L'utente autenticato.
- **Precondizione:** L'utente si trova dentro nella sezione menù, lista degli ordini personali o lista degli ordini in arrivo.
- **Postcondizione:** Viene inserito il piatto nei preferiti.
- **Scenario principale:**
  - L'utente si trova nella sezione menù, lista degli ordini personali o lista degli ordini in arrivo;
  - L'utente clicca sul bottone "cuore grigio";
  - Viene inserito il piatto nella lista dei preferiti.

### UC1.8 - Rimuovi preferiti

- **Descrizione:** L'utente rimuove un piatto dalla lista dei preferiti.
- **Attore Primario:**L'utente autenticato.
- **Precondizione:** L'utente si trova dentro nella sezione menù, lista degli ordini personali, lista degli ordini in arrivo o lista preferiti.
- **Postcondizione:** Viene rimosso il piatto dalla lista dei preferiti.
- **Scenario principale:**
  - L'utente si trova nella sezione menù;
  - L'utente clicca sul bottone "cuore rosa";
  - Viene rimosso il piatto dalla lista dei preferiti.

### UC1.9 - Aggiungi recensione

- **Descrizione:** L'utente aggiunge una recensione per un piatto.
- **Attore Primario:**L'utente autenticato.
- **Precondizione:** L'utente ha selezionato un piatto in modalità dettaglio.
- **Postcondizione:** Viene aggiornata la recensione del piatto.
- **Scenario principale:**
  - L'utente ha selezionato un piatto in modalità dettaglio;
  - L'utente clicca su una delle 5 stelle;
  - Viene inserita la recensione del piatto.

### UC5 - Visualizza lista preferiti

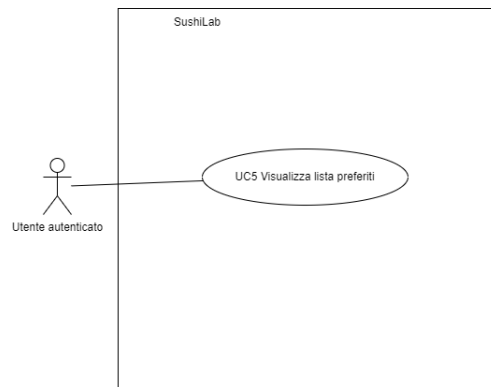


Figura A.13: Use Case - UC 5

- **Descrizione:** L'utente visualizza la lista dei preferiti.
- **Attore Primario:**L'utente autenticato.
- **Precondizione:** L'utente si trova dentro la web-app ed ha effettuato il login.
- **Postcondizione:** Viene mostrata la lista dei preferiti.
- **Scenario principale:**
  - L'utente si trova dentro la web-app;
  - L'utente entra nella sezione lista preferiti;
  - Viene mostrata la lista dei preferiti.



## Appendice B

# Elenco dei requisiti

In questa sezione vengono elencati tutti i requisiti. Ogni riga della tabella, che rappresenta un requisito, ha un codice identificativo, una descrizione del requisito e la fonte. Il codice dei requisiti è così strutturato R(F/Q/V)(N/D/O) dove:

R = requisito

F = funzionale

Q = qualitativo

V = di vincolo

N = obbligatorio (necessario)

D = desiderabile

Z = opzionale

### Lista dei requisiti

| Tabella dei requisiti |  |       |
|-----------------------|--|-------|
| Codice                | Requisito  | Fonte |
| ROF1                  | L'utente può accedere al menù del ristorante   | UC1   |
| ROF2                  | All'utente viene mostrato il menù del ristorante con tutte le categorie  | UC1.1 |
| ROF3                  | All'utente viene mostrato il menù del ristorante con tutti i piatti delle rispettive categorie di appartenenza | UC1.2 |
| ROF4                  | L'utente può aumentare la quantità di un piatto nel menù   | UC1.3 |
| ROF5                  | L'utente può diminuire la quantità di un piatto nel menù   | UC1.4 |
| ROF6                  | L'utente può impostare la visualizzazione dei piatti del menù nella modalità dettaglio                         | UC1.5 |
| ROF7                  | L'utente può impostare la visualizzazione dei piatti del menù nella modalità normale                           | UC1.6 |
| ROF8                  | L'utente può accedere alla maschera per la gestione del tavolo   | UC2   |
| ROF9                  | L'utente può generare la sessione del tavolo in cui si trova   | UC2.1 |
| ROF10                 | L'utente può unirsi alla sessione di un tavolo   | UC2.2 |

| Tabella dei requisiti |  |         |
|-----------------------|--|---------|
| Codice                | Requisito  | Fonte   |
| ROF11                 | L'utente può uscire dalla sessione di un tavolo  | UC2.3   |
| ROF12                 | L'utente può generare il QR-code della sessione del tavolo in cui si trova   | UC2.4   |
| ROF13                 | L'utente può accedere alla maschera per la gestione della lista degli ordini   | UC3     |
| ROF14                 | All'utente viene mostrata la lista degli ordini del tavolo   | UC3.1   |
| ROF15                 | All'utente viene mostrata la lista degli ordini personali  | UC3.2   |
| ROF16                 | All'utente viene mostrata la lista degli ordini in arrivo  | UC3.3   |
| ROF17                 | L'utente può spostare la lista degli ordini del tavolo in arrivo   | UC3.3.1 |
| ROF18                 | L'utente può generare il QR-code della lista degli ordini del tavolo in cui si trova   | UC3.1.2 |
| ROF19                 | L'utente può impostare la visualizzazione dei piatti della lista degli ordini personali in modalità dettaglio                                    | UC3.2.1 |
| ROF20                 | L'utente può marcare un piatto della lista degli ordini in arrivo come ricevuto  | UC3.3.1 |
| ROF21                 | L'utente non autenticato può accedere all'area personale   | UC4     |
| ROF22                 | L'utente non autenticato deve riuscire ad inserire email, password e conferma password nel form di registrazione per effettuare la registrazione | UC4.1   |
| ROF23                 | L'utente non autenticato deve riuscire ad inserire l'email e la password nel form di login per effettuare il login                               | UC4.2   |
| ROF24                 | L'utente non autenticato deve riuscire ad inserire l'email e la nuova password nel form della password dimenticata                               | UC4.3   |
| ROF25                 | L'utente autenticato deve riuscire ad effettuare il logout nell'area personale   | UC4.4   |
| ROF26                 | L'utente autenticato può aggiungere ingredienti non voluti nella blacklist   | UC4.4   |
| ROF27                 | L'utente autenticato può rimuovere ingredienti non voluti nella blacklist  | UC4.4   |
| ROF28                 | L'utente autenticato può aggiungere un piatto nella lista dei preferiti  | UC4.4   |
| ROF29                 | L'utente autenticato può rimuovere un piatto dalla lista dei preferiti   | UC4.4   |
| ROF30                 | L'utente autenticato può aggiungere una recensione per un piatto   | UC1.7   |

| Tabella dei requisiti |  |         |
|-----------------------|--|---------|
| Codice                | Requisito  | Fonte   |
| ROF31                 | L'utente autenticato può visualizzare la propria lista dei preferiti                     | UC5     |
| ROV1                  | L'interfaccia utente del sistema dovrà essere sviluppata sfruttando il framework Angular | SyncLab |
| ROV2                  | Lo stile dell'interfaccia utente del sistema dovrà essere sviluppato sfruttando CSS-3    | SyncLab |
| ROV3                  | Le chiamate API devono essere implementate tramite Stoptlight                            | SyncLab |
| ROV4                  | È necessario dividere le varie maschere in componenti diversi di Angular                 | SyncLab |
| ROV5                  | La web-app dovrà funzionare sul browser Microsoft Edge dalla versione più recente        | SyncLab |
| ROV6                  | La web-app dovrà funzionare sul browser Google Chrome dalla versione più recente         | SyncLab |
| ROV7                  | La web-app dovrà funzionare sul browser Firefox dalla versione più recente               | SyncLab |
| ROV8                  | La web-app dovrà funzionare sul browser Safari dalla versione più recente                | SyncLab |
| RDV9                  | Il codice sorgente dovrà essere commentato   | SyncLab |
| ROQ1                  | Il codice sorgente della piattaforma sarà reperibile su GitHub                           | SyncLab |
| ROQ2                  | Fornire una sezione tutorial che spieghi come si utilizza la web-app                     | SyncLab |

**Tabella B.1:** Tabella dei requisiti.





## Appendice C

# Descrizione dettagliata delle interfacce

In questa sezione vengono descritte tutte le interfacce della web-app e i servizi utilizzati. La descrizione delle principali interfacce sono reperibili alle sezione 4.2.

### Menù

Viene mostrato il menù del ristorante che comprende tutte le categorie e i loro piatti.

#### Funzionalità:

- L'utente può aumentare la quantità di un piatto per aggiungerlo o aumentare la sua quantità nell'ordine del tavolo;
- L'utente può diminuire la quantità di un piatto per rimuoverlo o diminuire la sua quantità nell'ordine del tavolo;
- L'utente può visualizzare il piatto in modalità dettaglio per dare una recensione al piatto o inserire una nota per il piatto;
- L'utente può aggiungere o rimuovere il piatto dalla lista dei preferiti.

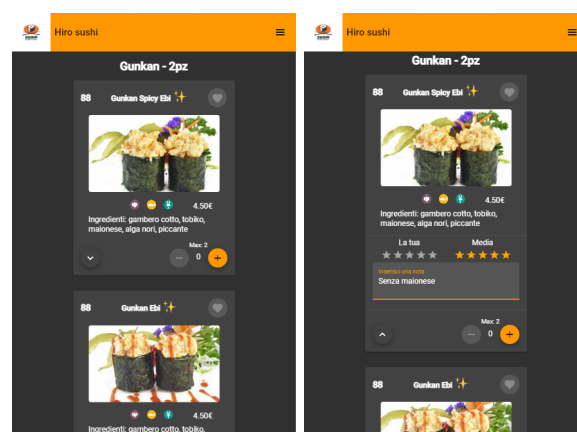


Figura C.1: Maschera menù di SushiLab

### Gestione del tavolo

Viene mostrata la maschera di gestione del tavolo.

#### Funzionalità:

- L'utente può creare una sessione del tavolo;
- L'utente può andare al form per unirsi ad una sessione.

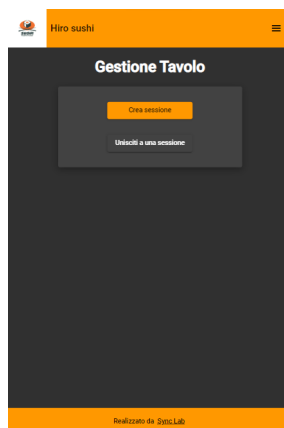


Figura C.2: Maschera gestione del tavolo di SushiLab

### Partecipazione sessione

Interfaccia tramite la quale è possibile unirsi ad una sessione del tavolo.

#### Funzionalità:

- L'utente può unirsi ad una sessione inserendo il codice di sessione, se quest'ultimo non rispetta la validazione del form non si abilita il bottone uniscisi;
- L'utente può tornare nella pagina gestione del tavolo.

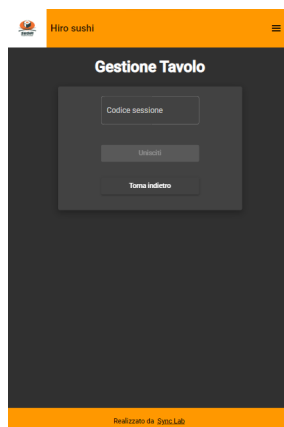


Figura C.3: Maschera uniscisi sessione di SushiLab

### QR-code tavolo

Viene mostrato il QR-code del tavolo che permette agli altri utenti che si trovano sullo stesso tavolo dell'utente di unirsi.

**Funzionalità:**

- L'utente può uscire dalla sessione premendo il bottone esci dalla sessione.

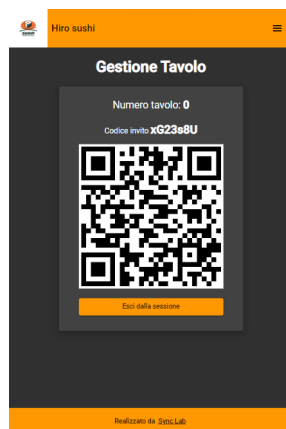


Figura C.4: Maschera QR-code tavolo di SushiLab

**Lista degli ordini del tavolo**

Viene mostrata la lista degli ordini della sessione del tavolo.

**Funzionalità:**

- L'utente può spostare la lista in arrivo;
- L'utente può generare il QR-code premendo sul bottone QR-code;
- L'utente può andare in altre sezioni della lista degli ordini utilizzando la navbar interna.

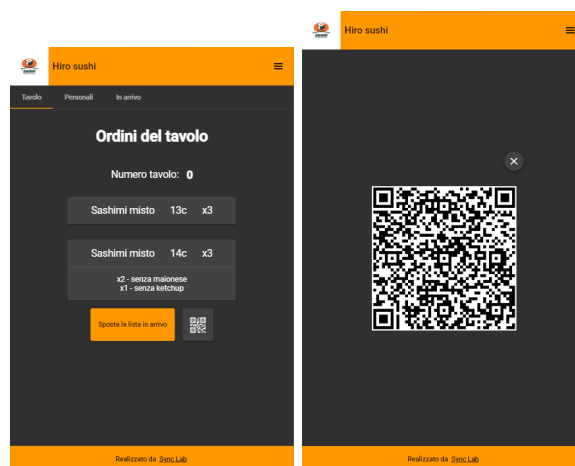


Figura C.5: Maschera ordini del tavolo e QR-code degli ordini di SushiLab

**Lista degli ordini personali**

Viene mostrata la lista degli ordini personali.

**Funzionalità:**

- L'utente può visualizzare i piatti in modalità dettaglio;

- L'utente può visualizzare i piatti in modalità normale;
- L'utente può aggiungere un piatto nella lista dei preferiti o rimuoverlo dalla lista;
- L'utente può dare una recensione ad un piatto.

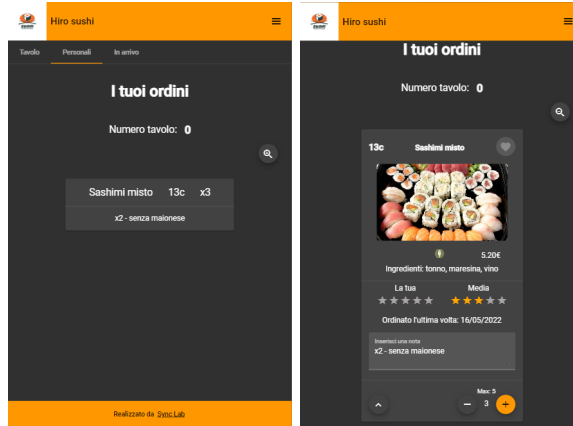


Figura C.6: Maschera ordini personali di SushiLab

### Lista degli ordini in arrivo

Viene mostrata la lista degli ordini in arrivo.

#### Funzionalità:

- L'utente può visualizzare i piatti in modalità normale o dettagliata;
- L'utente può marcare un piatto come arrivato;
- L'utente può aggiungere o rimuovere un piatto dalla lista dei preferiti;
- L'utente può dare una recensione ad un piatto.

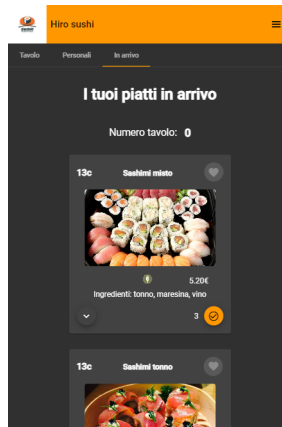


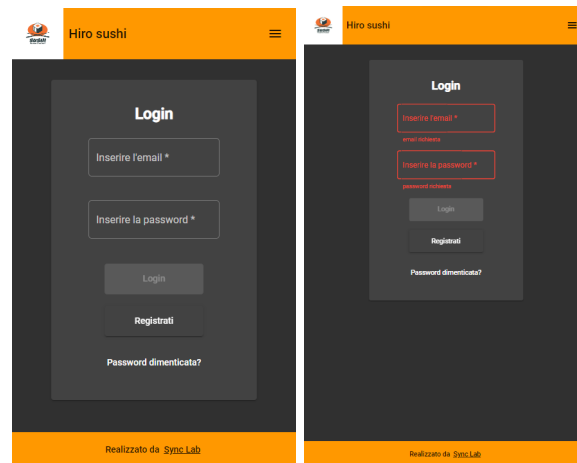
Figura C.7: Maschera ordini in arrivo di SushiLab

### Login

Viene mostrato il form per effettuare il login dove è possibile effettuare l'autenticazione.

#### Funzionalità:

- L'utente può effettuare il login inserendo i campi correttamente altrimenti vengono evidenziati in rosso i campi sbagliati;
- L'utente può andare al form di registrazione;
- L'utente può andare al form per il recupero della password.



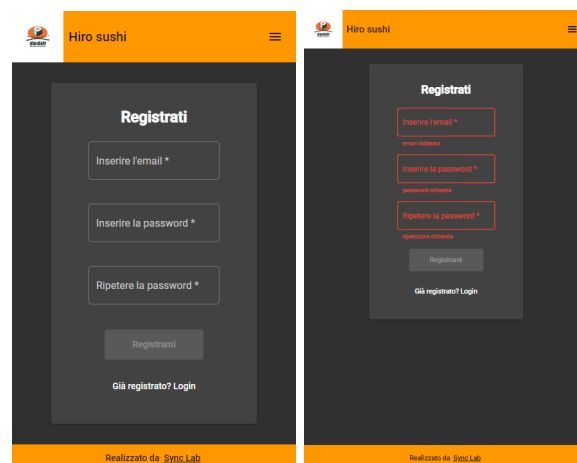
**Figura C.8:** Maschera login di SushiLab

## Registrazione

Viene mostrato il form di registrazione dove è possibile registrarsi.

### Funzionalità:

- L'utente può effettuare la registrazione inserendo i campi correttamente altrimenti vengono evidenziati in rosso i campi sbagliati;
- L'utente può andare al form di login.



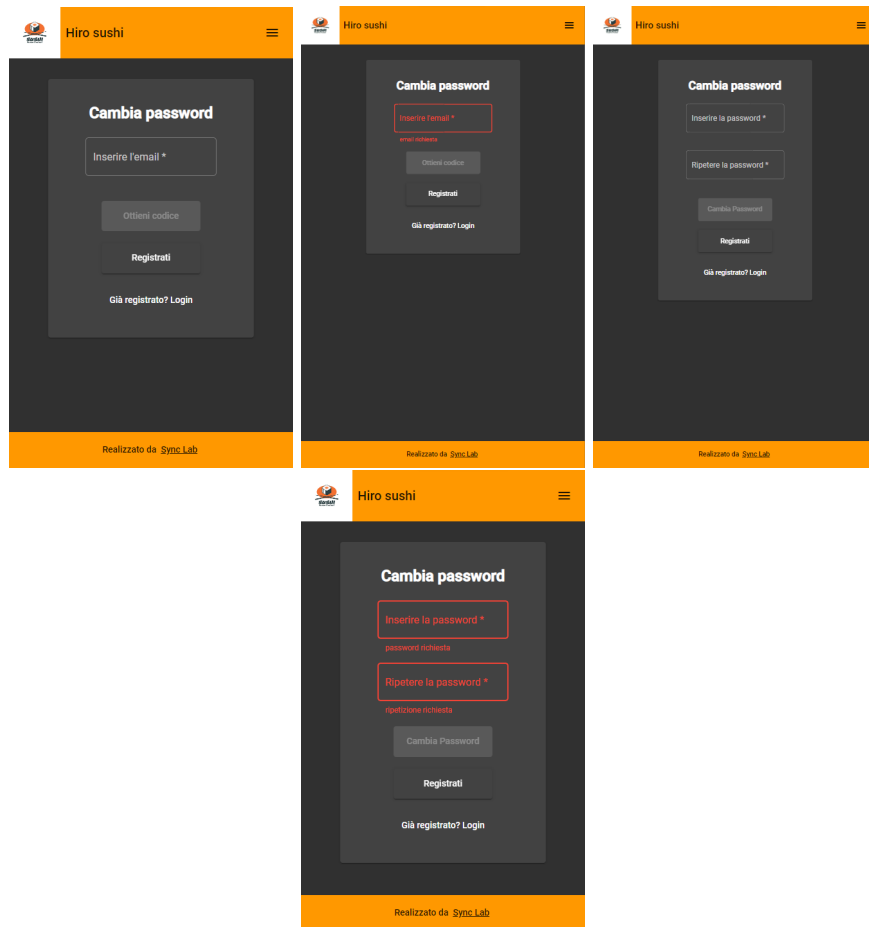
**Figura C.9:** Maschera registrazione di SushiLab

## Password Dimenticata

Viene mostrato il form per il recupero password.

### Funzionalità:

- L'utente può cambiare la password inserendo i campi correttamente altrimenti vengono evidenziati in rosso i campi sbagliati;
- L'utente può andare al form di registrazione;
- L'utente può andare al form di login.



**Figura C.10:** Maschera cambia password di SushiLab

## Area Personale

Viene mostrata l'area personale dell'utente dove può visualizzare i dati personali.

### Funzionalità:

- L'utente può effettuare il logout;
- L'utente può andare nella sezione blacklist degli ingredienti per inserire o rimuovere degli allergeni o delle preferenze.

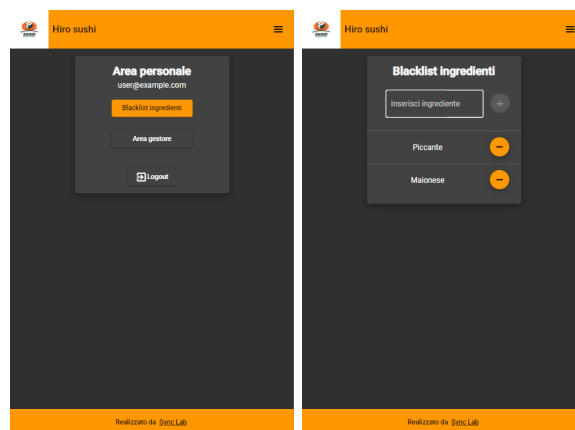


Figura C.11: Maschera area personale e blacklist di SushiLab

### Menù component

Componente usato per creare la maschera del menù.

#### Componenti e servizi usati:

- piatto.component;
- menu.service;
- ordini.service.

### Nav component

Componente usato per creare la maschera della navbar. Composto da un insieme di link che porta l'utente nelle varie sezioni della web-app.

#### Metodi:

- isLoggedIn(): ritorna true se l'utente è autenticato;
- haveMenu(): ritorna true se è presente il menù nel local storage;
- haveTable(): ritorna true se è presente la sessione del tavolo nel local storage.

#### Componenti e servizi usati:

- auth.service;
- menu.service;
- table.service.

### Ordini component

Componente usato per creare la maschera della lista degli ordini.

#### Metodi:

- moveOrders(): sposta la lista degli ordini del tavolo in arrivo;
- changeZoom(): cambia la modalità di presentazione dei piatti;
- showQrCode(): mostra il QR-code della lista degli ordini;
- closeQrCode(): chiude la maschera di QR-code;
- getAllOrders(): prende tutti gli ordini dal server.

#### Componenti e servizi usati:

- piatto-arrivo.component;
- piatto.component;
- piatto-ordine.component;

- auth.service;
- menu.service;
- table.service.

### **Piatto-arrivo component**

Componente usato per creare i piatti della sezione lista degli ordini in arrivo.

#### **Metodi:**

- toggleFav(): cambia lo stato di preferito di un determinato piatto;
- updateStar(): aggiorna la recensione di un determinato piatto;
- decreaseCount(): chiamato quando utente marca un piatto come arrivato e diminuisce la quantità di un piatto in arrivo.

#### **Componenti e servizi usati:**

- auth.service;
- menu.service.

### **Piatto-ordine component**

Componente usato per creare il piatto in modalità normale per la sezione degli ordini.

### **Personale component**

Componente usato per creare la maschera dell'area personale, dove si può vedere i credenziali dell'utente e la sezione blacklist.

#### **Metodi:**

- isLoggedIn(): ritorna true se l'utente non è autenticato.

#### **Componenti e servizi usati:**

- blacklist.component;
- forgot.component;
- login.component;
- logout.component;
- register.component;
- auth.service;
- user.service.

### **Blacklist component**

Componente usato per creare la sezione blacklist degli ingredienti, viene mostrato la blacklist ed è possibile rimuovere o aggiungere un ingrediente.

#### **Metodi:**

- onAdd(): metodo chiamato per aggiungere un ingrediente nella blacklist;
- onRemove(): metodo chiamato per rimuovere un ingrediente dalla blacklist;
- isValid(): ritorna true se l'ingrediente inserito è valido.

#### **Componenti e servizi usati:**

- user.service.

### **Login component**

Componente usato per creare il form login.

#### **Metodi:**



- `login()`: metodo chiamato quando l'utente conferma il form, vengono controllati i campi e infine viene chiamato API per effettuare l'autenticazione.

**Componenti e servizi usati:**

- `auth.service`.

**Register component**

Componente usato per creare il form per la registrazione.

**Metodi:**

- `register()`: metodo chiamato quando l'utente conferma il form, vengono controllati i campi e infine viene chiamato API per la registrazione.

**Componenti e servizi usati:**

- `user.service`.

**Preferiti component**

Componente usato per creare la maschera dei preferiti.

**Metodi:**

- `toggleExpand()`: cambia la modalità di visualizzazione dei piatti;
- `toggleFav()`: cambia lo stato di preferito di un determinato piatto;
- `updateStar()`: aggiorna la recensione di un determinato piatto;
- `increaseCount()`: aumenta la quantità di un piatto negli ordini;
- `decreaseCount()`: diminuisce la quantità di un piatto negli ordini;
- `noteChanged()`: cambia la nota di un piatto negli ordini.

**Componenti e servizi usati:**

- `star.component`;
- `ordini.service`;
- `auth.service`;
- `menu.service`.

**Tavolo component**

Componente usato per creare la maschera gestione del tavolo.

**Metodi:**

- `changeState()`: cambia lo stato di visualizzazione della sezione gestione del tavolo in base alla sessione;
- `removeSession()`: rimuove la sessione del tavolo;
- `setSession()`: modifica la sessione del tavolo;
- `updateURL()`: aggiorna il percorso del link per generare il QR-code del tavolo;
- `createSession()`: crea la sessione del tavolo.

**Componenti e servizi usati:**

- `table.service`;
- `menu.service`.

**auth.service**

Servizio utilizzato per gestire le autenticazioni. **Metodi:**

- `login(email: string, password: string)`: metodo chiamato per effettuare il login dell'utente;
- `setSession()`: salva i dati dell'utente dopo il login nel local storage per mantenere la sessione;

- `logout()`: metodo chiamato per effettuare il logout dell'utente;
- `isLoggedIn()`: ritorna true se è presente nel local storage la sessione dell'utente;
- `getExpirations()`: ritorna la scadenza della sessione dell'autenticazione.

#### **menu.service**

Servizio utilizzato per gestire tutte le funzionalità del menù.

##### **Metodi:**

- `title()`: ritorna il nome del menù;
- `haveMenu()`: ritorna true se è presente un menù nel local storage;
- `get menuId()`: ritorna l'id del menù;
- `set menuId(id: number)`: salva l'id del menù nel local storage;
- `applyBlacklist(menu: Menu)`: filtra il menù in base alla blacklist;
- `setFavorite(idPiatto:number, favorite:boolean)`: aggiunge o rimuove un piatto dalla lista dei preferiti;
- `setRating(idPiatto: number, rate: number)`: salva la recensione del piatto;
- `getPreferiti()`: richiede al server la lista dei preferiti dell'utente.

#### **ordini.service**

Servizio utilizzato per gestire le ordinazioni.

##### **Metodi:**

- `updateRemoteOrder()`: aggiorna gli ordini del server mandandogli la lista degli ordini ogni lasso di tempo;
- `getOrded(idPiatto:number)`: ritorna indice nel array degli ordini del piatto passato;
- `clearOrder()`: svuota array degli ordini presente nel local storage;
- `getSessionCode()`: ritorna il codice sessione del tavolo;
- `getOrdersTable()`: ritorna l'ordine del tavolo;
- `getOrdersUser()`: ritorna l'ordine dell'utente;
- `getOrdersArriving()`: ritorna l'ordine in arrivo;
- `moveOrders()`: sposta l'ordine del tavolo in arrivo;
- `toQrCode()`: ritorna il QR-code degli ordini.

#### **tavolo.service**

Servizio utilizzato per gestire le sessioni del tavolo.

##### **Metodi:**

- `createSession()`: creare una sessione del tavolo;
- `getSession()`: ritorna il numero del tavolo;
- `getCode()`: ritorna il codice di sessione;
- `haveSession()`: ritorna true se l'utente si trova in una sessione;
- `setSession(code: string)`: imposta il codice di sessione con il codice passato;
- `removeSession()`: rimuove i dati della sessione del tavolo.

#### **user.service**

Servizio utilizzato per gestire le funzionalità dell'area personale.

##### **Metodi:**

- `pushBlacklist()`: chiama il server per aggiornare la blacklist degli ingredienti;
- `getUtente()`: richiede al server per ottenere le informazione dell'utente;

- `setUtente(user: User)`: manda al server i dati dell'utente per completare la registrazione;
- `changePassword(password: string)`: manda al server la password nuova per aggiornare quella vecchia.







# Glossario

**API** in informatica con il termine *Application Programming Interface API* (ing. interfaccia di programmazione di un'applicazione) si 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. [12](#), [28](#), [61](#)

**Back-end** parte del software che elabora i dati generati dal front-end e comunica con il data-base per scrivere e leggere dati. [3](#), [61](#)

**Front-end** è la parte di un sistema software che gestisce l'interazione con l'utente o con sistemi esterni che producono dati di ingresso per il back-end.. [3](#), [61](#)

**Maschera** indica l'interfaccia che visualizza l'utente. [11](#), [61](#)

**Mock** sono degli oggetti simulati che riproducono il comportamento di oggetti reali in modo controllato. [19](#), [61](#)

**Ordini del tavolo** è l'insieme dei piatti aggiunti negli ordini di tutti gli utenti della stessa sessione di un tavolo. [15](#), [61](#)

**Ordini in arrivo** è l'insieme dei piatti che l'utente ha ordinato e sono in preparazione dall'ristorante. [15](#), [61](#)

**Ordini personali** è l'insieme dei piatti che l'utente ha ordinato in una determinata sessione. [15](#), [61](#)

**Responsive** indica una tecnica di web design per la realizzazione di siti in grado di adattarsi automaticamente al dispositivo coi quali vengono visualizzati.. [14](#), [61](#)

**REST** è uno stile architetturale per sistemi distribuiti. L'espressione "representational state transfer" e il suo acronimo, REST, fu introdotto nel 2000 nella tesi di dottorato di Roy Fielding e vennero rapidamente adottati dalla comunità di sviluppatori Internet. I metodi più utilizzati sono: GET, POST, PUT, PATCH e DELETE. [3](#), [61](#)

**Sessione** gruppo di clienti appartenenti ad una stesso tavolo. [11](#), [61](#)

**Spring** un framework open source per lo sviluppo delle applicazioni su piattaforma Java, nato nel 2002. Le sue principali punti di forza sono: flessibilità ,modularità, elevata testabilità e una grande community.. [3](#), [61](#)

**VoiP** in telecomunicazioni e informatica, indica una tecnologia che rende possibile effettuare una conversazione, analoga a quella che si potrebbe ottenere con una

rete telefonica, sfruttando una connessione Internet o una qualsiasi altra rete di telecomunicazioni che utilizzi il protocollo IP.. 7, 61

**Web-app** applicazione memorizzato su un server remoto distribuito su internet e può essere utilizzato tramite un browser. 3, 62



# Bibliografia

## Siti web consultati

*Documentazione Angular.* URL: <https://angular.io/>.

*Documentazione Spring.* URL: <https://spring.io/>.

*Documentazione TypeScript.* URL: <https://www.typescriptlang.org/>.

*Sito ufficiale di SyncLab.* URL: <https://www.synclab.it/>.