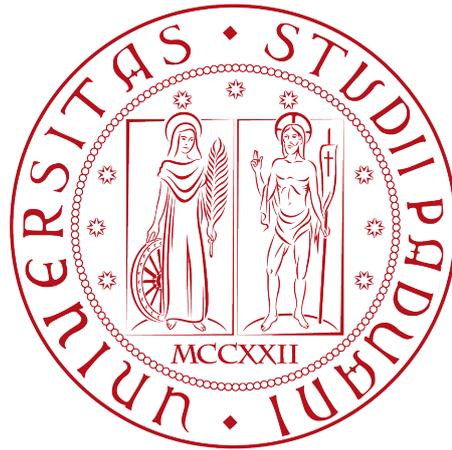


Università degli Studi di Padova

DIPARTIMENTO DI MATEMATICA "TULLIO LEVI-CIVITA"

CORSO DI LAUREA IN INFORMATICA



Sviluppo di un'applicazione *mobile* per
riscaldatori per ambienti

Tesi di laurea

Relatore

Prof. Francesco Ranzato

Laureando

Samuele Rizzato

ANNO ACCADEMICO 2022-2023

Samuele Rizzato: *Sviluppo di un'applicazione mobile per riscaldatori per ambienti*, Tesi di laurea, © Febbraio 2023.

Sommario

Il presente documento descrive il lavoro svolto durante il periodo di stage, della durata di circa trecentoventi ore, dal laureando Samuele Rizzato presso l'azienda Moku S.r.l. Lo scopo era quello di sviluppare un'applicazione Android per interfacciarsi con dei riscaldatori ad infrarossi. Gli obiettivi da raggiungere erano molteplici.

In primo luogo era richiesto lo sviluppo dell'interfaccia grafica che avrebbe permesso di visualizzare e configurare i vari riscaldatori. In secondo luogo era richiesta l'implementazione della comunicazione effettiva tra l'app e i dispositivi attraverso la tecnologia Bluetooth Low Energy.

A termine dello stage sono state implementate le interfacce per visualizzare i riscaldatori e i loro dettagli, lo scan dei dispositivi e alcune richieste e risposte per la comunicazione con le periferiche.

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.

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

Ringrazio tutto il team di Moku per la disponibilità e la gentilezza mostrata nei miei confronti.

Ho desiderio di ringraziare poi i miei amici, in particolar modo Diego e Luciano per tutti i fantastici anni passati insieme e le mille avventure vissute ed infine tutte le persone che ho conosciuto durante il percorso universitario.

Padova, Febbraio 2023

Samuele Rizzato

Indice

1	Introduzione	1
1.1	L'azienda	1
1.2	Introduzione al progetto	2
1.3	Organizzazione del testo	2
2	Processi e metodologie	3
2.1	Processi di sviluppo	3
2.1.1	Metodologia Agile	3
2.2	Strumenti utilizzati	4
2.2.1	Comunicazione e organizzazione	4
2.2.2	Sviluppo	4
3	Descrizione dello stage	6
3.1	Descrizione del progetto	6
3.2	Obiettivi dello stage	7
3.3	Pianificazione	7
3.3.1	Variazioni nella pianificazione	8
4	Analisi dei requisiti	9
4.1	Attori primari e secondari	9
4.1.1	Attore primario	9
4.1.2	Attori secondari	9
4.2	Scenari	10
4.2.1	Scenario schermata principale	10
4.2.2	Scenario schermata dettagli	10
4.3	Requisiti	10
5	Progettazione e codifica	13
5.1	Tecnologie	13
5.1.1	Descrizione del Bluetooth Low Energy	14
5.2	Progettazione	14
5.2.1	Architettura	14
5.2.2	Descrizione dei <i>package</i>	15
5.3	Codifica	16
5.3.1	Schermata principale	16
5.3.2	<i>Appliance Detail</i>	18
5.3.3	Comunicazione bluetooth	20
5.4	Verifica e Validazione	21

6 Conclusioni	23
6.1 Consuntivo dello stage	23
6.2 Raggiungimento degli obiettivi	24
6.3 Conoscenze acquisite	24
6.4 Valutazione personale	24
A Appendice A	27
A.1 Casi d'uso	27
A.2 Tracciamento dei requisiti	35
A.2.1 Elenco dei requisiti	35
A.2.2 Tracciamento Fonte - Requisiti	38
B Appendice B	41
Acronimi e abbreviazioni	45
Glossario	46
Bibliografia	49

Elenco delle figure

1.1	Logo dell'azienda	1
4.1	Attori primari nei diagrammi dei casi d'uso	9
4.2	Attori secondari nei diagrammi dei casi d'uso	10
4.3	Comunicazione bluetooth tra smartphone e <i>appliance</i>	11
5.1	<i>Package</i> dell'applicazione	14
5.2	Struttura <i>package</i> ble	15
5.3	Schermata appliance list con riscaldatori di categoria 1, 2, 3 e 4	16
5.4	Schermata appliance list vuota	17
5.5	Schermata appliance detail, modifica nome, impostazione timer e visualizzazione foto	19
5.6	Diagramma di sequenza della scannerizzazione e connessione di un riscaldatore	20
A.1	Scenario schermata principale	27
A.2	Scenario visualizzazione dei dettagli di un <i>appliance</i>	29
A.3	UC7 - impostazione controlli	31
A.4	UC8 - impostazione timer	34

Elenco delle tabelle

3.1	Tabella del codice di un obiettivo	7
3.2	Tabella degli obiettivi del tirocinio	7
3.3	Tabella delle attività dello stage	8
6.1	Tabella del consuntivo finale di stage	23
6.2	Tabella del raggiungimento degli obiettivi di tirocinio	24
A.1	Tabella del codice di un requisito	36
A.2	Tabella del tracciamento dei requisiti funzionali	37
A.3	Tabella del tracciamento dei requisiti qualitativi	38
A.4	Tabella del tracciamento dei requisiti di vincolo	38
A.5	Tabella del tracciamento dei requisiti prestazionali	38
A.6	Tabella del tracciamento fonte - requisiti	39
B.1	Tabella del tracciamento dei requisiti funzionali soddisfatti relativi alla schermata principale	41
B.2	Tabella del tracciamento dei requisiti funzionali soddisfatti relativi alla schermata <i>Appliance Detail</i>	43
B.3	Tabella del tracciamento dei requisiti qualitativi, di vincolo e prestazionali soddisfatti	43

Capitolo 1

Introduzione

In questo capitolo viene presentata l'azienda presso cui è stato svolto lo stage e l'organizzazione dei capitoli del testo.

1.1 L'azienda

Moku S.r.l. è un'azienda nata nel 2013 in provincia di Treviso. Moku fornisce soluzioni software che curano la parte di sviluppo e design di un prodotto. Durante gli anni ha toccato diversi campi come la biomedicina, la domotica e la logistica. L'azienda utilizza una metodologia di lavoro [Agile^{\[8\]}](#) per gestire i flussi dei progetti commissionati.



Figura 1.1: Logo dell'azienda

Il team dell'azienda oggi ha più di venti persone suddivise nei seguenti ruoli:

- * **CEO:** l'amministratore delegato è il componente che gestisce i clienti, gli investitori e i partner;
- * **CTO:** è il componente incaricato di scegliere e coordinare le tecnologie impiegate dall'azienda;
- * **CFO:** è il componente che si occupa della pianificazione delle attività finanziarie;
- * **project manager:** è la figura responsabile di sovrintendere e pianificare le attività del progetto;
- * **analista:** si occupa di raccogliere e definire i requisiti del cliente;
- * **sviluppatore:** colui che implementa la soluzione definita in fase di progettazione;
- * **designer:** realizza l'interfaccia utente e la user experience del software da sviluppare.

1.2 Introduzione al progetto

Il progetto riguarda la domotica, un campo che negli ultimi anni si sta diffondendo sempre di più al fine di migliorare la qualità di vita quotidiana. L'applicazione da sviluppare è, appunto, pensata per semplificare l'interazione e la configurazione di riscaldatori per ambienti, grazie alla sua interfaccia, e rendere più intelligente, rispetto agli elettrodomestici tradizionali, l'accensione e l'autoregolazione di alcuni parametri.

1.3 Organizzazione del testo

- * Il [secondo capitolo](#) descrive la metodologia di lavoro a cui ricorre il team di Moku e gli strumenti utilizzati per cooperare con i membri del gruppo.
- * Il [terzo capitolo](#) approfondisce il progetto di stage andando ad illustrare gli obiettivi fissati con l'azienda e la pianificazione delle attività.
- * Il [quarto capitolo](#) approfondisce la fase di analisi del tirocinio mostrando i diversi scenari e i requisiti raccolti.
- * Il [quinto capitolo](#) illustra le tecnologie utilizzate per lo sviluppo, la progettazione dell'architettura e le interfacce dell'applicazione.
- * Nel [sesto capitolo](#) viene descritto come si è concluso lo stage e le mie impressioni sull'esperienza di tirocinio presso l'azienda.

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

- * gli acronimi, le abbreviazioni e i termini ambigui o di uso non comune menzionati vengono definiti nel glossario, situato alla fine del presente documento;
- * per la prima occorrenza dei termini riportati nel glossario viene utilizzata la seguente nomenclatura: *parola*^[g];
- * i termini in lingua straniera o facenti parti del gergo tecnico sono evidenziati con il carattere *corsivo*.

Capitolo 2

Processi e metodologie

Il capitolo illustra i processi di sviluppo, le metodologie e gli strumenti adottati dal tirocinante e dai membri dall'azienda per lavorare in gruppo.

2.1 Processi di sviluppo

2.1.1 Metodologia Agile

Moku utilizza un approccio Agile per realizzare *software* funzionante e che soddisfi le aspettative del cliente. Viene usato il *framework* SCRUM, il quale permette di produrre *software* funzionante e ripetutamente, aiuta il team a riflettere sui successi e gli insuccessi conseguiti e facilita la comunicazione con il cliente. Tre sono i principi su cui si fonda:

- * **Trasparenza:** gli aspetti significativi del processo devono essere visibili ai responsabili del lavoro e devono seguire degli standard comuni, ad esempio una definizione condivisa della parola *done* per chi realizza il lavoro e chi lo controlla;

- * **Ispezione:** gli artefatti prodotti devono subire ispezione frequente in modo tale da accertare che i risultati prodotti rispettino gli obiettivi e individuare eventuali differenze rispetto a ciò che è da realizzare.

- * **Adattamento:** chi ispeziona se trova uno o più aspetti del processo di produzione al di fuori dei limiti accettabili deve rapidamente apportare aggiustamenti al fine di minimizzare ulteriori deviazioni.

I progetti Scrum progrediscono attraverso degli sprint, ossia carichi di lavoro da completare in un periodo che può andare da 1 a 4 settimane. Prima di ogni sprint c'è una fase di pianificazione in cui si stabiliscono quali obiettivi raggiungere e le tempistiche.

2.2 Strumenti utilizzati

2.2.1 Comunicazione e organizzazione

Google Meet

Google Meet è un'applicazione per le videoconferenze utilizzata durante lo stage per le comunicazioni interne quando un membro del team era in *smart working* e con il cliente.

Jira

Jira è un programma per il tracciamento delle segnalazioni sviluppato da Atlassian. Nel corso dello stage è stata utilizzata particolarmente la funzione delle *board* al fine di pianificare le attività del progetto.

Slack

Slack è un'applicazione per la collaborazione che consente di scambiare messaggi istantaneamente con i membri del team e che offre la possibilità di effettuare delle chiamate. Il programma, oltretutto, permette di creare specifici canali in modo tale da organizzare il lavoro. Durante lo stage è stato utilizzato per scambiare rapidi messaggi con il tutor e per collaborare con il resto del team attraverso appositi canali.

Passbolt

Passbolt è un *password manager* [open source](#)^[g] progettato per essere utilizzato da gruppi di persone. Nel progetto è stato utile per accedere al profilo condiviso di Figma.

2.2.2 Sviluppo

Android studio

Android studio è un ambiente di sviluppo integrato gratuito nato per la creazione di applicazioni Android native. Il *software* mette a disposizione, oltre alle più comuni funzionalità di un [Integrated Development Environment \(IDE\)](#)^[g], degli emulatori per testare l'applicazione che si sta sviluppando o di installare quest'ultima direttamente in uno *smartphone*.

Bitbucket

Bitbucket è un servizio di *hosting web* adatto a progetti basati su [Version Control System \(VCS\)](#)^[g] Mercurial o Git.

Figma

Figma è un *editor* di grafica vettoriale e uno strumento di prototipazione accessibile attraverso l'omonimo sito *web*. Durante lo stage è stato utilizzato per consultare i *mockup* dell'interfaccia grafica.

Git

Version control system gratuito e *open source* che permette di mantenere una cronologia di tutte le modifiche dei *file* e facilita diverse operazioni come il ripristino di file a versioni precedenti oppure l'individuazione e la risoluzione di conflitti.

Sourcetree

Una interfaccia grafica la quale offre una rappresentazione visuale dei repository così da semplificare le operazioni da compiere tramite Git. Nel progetto si è preferito utilizzare questo programma piuttosto che quello fornito da Android studio.

Capitolo 3

Descrizione dello stage

In questo capitolo viene descritto il progetto, gli obiettivi fissati ed infine la pianificazione delle attività di stage.

3.1 Descrizione del progetto

Moku S.r.l. sta sviluppando per conto di un'importante azienda del territorio un'app per l'installazione ed il controllo di riscaldatori per ambienti. L'applicazione principale è nel mondo della ristorazione. L'app dovrà rilevare e configurare i controller installati tramite bluetooth, caricare i parametri di configurazione del controller e dei riscaldatori ad esso associati (fino a 5) ed abilitare, se richiesto, la comunicazione tramite IP (WiFi o Ethernet). I riscaldatori non sono tutti uguali e possono essere raggruppati principalmente in quattro categorie:

- * **Categoria 1:** riscaldatori che possono essere solamente accesi o spenti;
- * **Categoria 2:** riscaldatori che possono essere accesi, spenti e di cui si può regolare l'intensità di riscaldamento;
- * **Categoria 3:** riscaldatori che possono essere accesi, spenti, di cui si può impostare il livello di potenza di riscaldamento solamente su determinati valori e che hanno una lampada collegata di cui si può regolare l'intensità della luce;
- * **Categoria 4:** riscaldatori come **categoria 3**, ma il cui livello di potenza di riscaldamento può essere regolato liberamente.

Una volta configurati i riscaldatori questi possono essere inseriti in zone diverse e si potrà controllare il loro livello di potenza e (se previsto) il livello di luminosità della lampada collegata. Altre funzionalità aggiuntive prevedono la possibilità di impostare dei timer e degli scenari, da attivare al bisogno o in momenti predeterminati. Il controllo può avvenire direttamente tramite [Bluetooth Low Energy \(BLE\)](#)^[g] oppure passando per un [backend](#)^[g] se tramite wifi. Moku Srl si occuperà del design, della progettazione e della realizzazione delle app mobile native iOS e Android. Per il progetto di stage era previsto di realizzare alcune parti dell'app Android, quali la lista dei riscaldatori, la schermata di dettaglio di quest'ultimi, il flusso di configurazione e di login. Il backend viene realizzato da un altro fornitore.

3.2 Obiettivi dello stage

Di seguito vengono riportati gli obiettivi stabiliti dall'azienda per l'attività di stage. Il codice identificativo dell'obiettivo è composto da:

O [TIPO] [NUMERO]

	Descrizione
TIPO	Indica la categoria di obiettivo: O : obbligatorio D : desiderabile Z : opzionale
NUMERO	numero identificativo

Tabella 3.1: Tabella del codice di un obiettivo

Codice	Descrizione
OO1	Gestione e pianificazione del progetto attraverso kanban board condivisa.
OO2	Analisi e progettazione applicazione Mobile, a partire dai requisiti raccolti.
OO3	Analisi ed implementazione flussi login/logout/signup.
OO4	Analisi ed integrazione flusso installazione BLE.
OO5	Analisi ed integrazione flusso configurazione IP.
OO6	Integrazione gestione comandi BLE.
OO7	Implementazione interfacce gestione Appliance, Zone, Scene.
OD1	Integrazione comandi tramite Cloud.
OD2	Suite di testing del software prodotto.
OD3	Documentazione completa.
OZ1	Ulteriori modifiche all'applicazione che esulano da quando riportato in questo documento.

Tabella 3.2: Tabella degli obiettivi del tirocinio

3.3 Pianificazione

Prima dell'inizio del tirocinio sono state pianificate le attività in questo modo:

Settimana	Attività	Ore
1	Comprensione sistema e obiettivi.	20
1	Analisi dei requisiti.	20
2-3	Progettazione.	60
3	Studio e setup ambiente di sviluppo.	20
4-7	Implementazione.	150
7-8	Test e validazione.	30
8	Documentazione.	20
	Totale	320

Tabella 3.3: Tabella delle attività dello stage

3.3.1 Variazioni nella pianificazione

Diversamente da quanto stabilito lo stage è iniziato con due giorni di ritardo, tuttavia sono stati recuperati durante l'ultima settimana. Ci sono stati poi dei ritardi durante la seconda settimana a causa di problemi di salute miei e di altri membri del team che hanno rallentato il lavoro.

Capitolo 4

Analisi dei requisiti

Nel seguente capitolo viene descritta l'analisi dei requisiti svolta per il progetto andando ad illustrare gli scenari e i requisiti raccolti.

Nelle sezioni successive verrà fatta una descrizione ad alto livello dei casi d'uso e dei requisiti, una visione più dettagliata la si trova nell'appendice [A](#).

4.1 Attori primari e secondari

Nei casi d'uso l'attore è un soggetto, solitamente esterno al sistema, che interagisce con l'applicazione ed in particolare si dice attore primario colui che avvia il particolare caso.

4.1.1 Attore primario

utente generico che interagisce con l'applicazione per visualizzare e configurare i vari riscaldatori.



Figura 4.1: Attori primari nei diagrammi dei casi d'uso

4.1.2 Attori secondari

Gli attori secondari nei casi individuati sono i seguenti:

- * *appliance*, ovvero il riscaldatore con cui il software interagisce per effettuare le varie operazioni;
- * fotocamera, l'applicazione predefinita nello *smartphone* per scattare foto;

* galleria, l'applicazione predefinita per esplorare la galleria del telefono.



Figura 4.2: Attori secondari nei diagrammi dei casi d'uso

4.2 Scenari

4.2.1 Scenario schermata principale

Nella schermata principale l'utente visualizza tre sezioni chiamate *appliances*, *zones*, *scenes*. La scheda *appliances* deve poter permettere la visualizzazione dei riscaldatori installati e scannerizzati tramite bluetooth, di conseguenza se quest'ultimo è disattivato la schermata dovrà mostrare un errore all'utente con un messaggio che lo invita ad attivare la funzionalità. Nella scheda *zones* dovrà essere mostrata una lista di zone ossia un insieme di riscaldatori i quali potranno essere controllati contemporaneamente. *Scenes*, infine, permetterà di vedere una lista di scene, ovvero un un insieme di comandi, che possono essere anche programmati, da dare agli *appliances*.

4.2.2 Scenario schermata dettagli

Ogni *appliance* possiede diversi parametri:

- * nome;
- * zona, gruppo di *appliance* a cui appartiene;
- * foto, un'immagine inserita tramite la galleria del telefono oppure tramite la fotocamera;
- * controlli, questi permettono di accendere, spegnere e regolare la potenza del riscaldamento o delle luci;
- * *countdown timer*, determina quanto tempo manca allo spegnimento automatico, può essere attivato solo quando anche i controlli sono accesi.

Nella schermata di dettaglio l'utente potrà modificare questi campi.

4.3 Requisiti

Dagli scenari precedenti si deduce che l'utente dovrà poter comandare dei riscaldatori attraverso tre modalità principali e la comunicazione con questi avverrà attraverso la tecnologia bluetooth.



Figura 4.3: Comunicazione bluetooth tra smartphone e *appliance*

Avviata l'applicazione nella schermata principale sarà possibile impostare i parametri dei riscaldatori scannerizzati; dal momento che il software è pensato anche per il mondo della ristorazione risulterà più comodo e rapido ad utenti come i camerieri impostare direttamente il riscaldamento dal telefono.

Il riscaldamento e l'intensità delle luci potranno essere regolati per ogni singolo *appliance* o a gruppi utilizzando la funzionalità delle *zones*. Usando la *feature* delle *scenes* si potrà, inoltre, programmare l'accensione e lo spegnimento a determinate ore di gruppi di riscaldatori, ad esempio dall'apertura alla chiusura del ristorante.

Riepilogando nelle tre schede principali si visualizzeranno rispettivamente delle liste di *appliances*, *zones* e *scenes* inoltre nel caso di lista vuota dovrà essere possibile inserire un nuovo elemento.

Tramite l'app dovrà essere possibile anche personalizzare la configurazione dei parametri degli *appliance*. Il nome, i controlli e il timer sono obbligatori mentre la zona assegnata e la foto sono opzionali. Per quanto riguarda la foto dovrà essere possibile rimuoverla e inserirla usando la fotocamera o la galleria del dispositivo, previo controllo dei permessi. Al termine dell'inserimento se si verificherà un problema dovrà essere notificato all'utente con un messaggio d'errore.

Infine dovrà essere presente il meccanismo per la connessione tramite bluetooth, il quale dovrà rilevare i dispositivi vicini entro un certo periodo di tempo, in quanto l'operazione di scannerizzazione consuma molta batteria, e comunicare con essi.

Capitolo 5

Progettazione e codifica

Il capitolo descrive le tecnologie e l'architettura utilizzate nell'applicazione, inoltre vengono descritte le interfacce implementate.

5.1 Tecnologie

Di seguito viene data una panoramica delle tecnologie utilizzate.

Android

Android è un sistema operativo per dispositivi mobili sviluppato da Google, progettato principalmente per sistemi *embedded* quali *smartphone* e *tablet*, inoltre è il sistema operativo per telefoni più diffuso al mondo.

Kotlin

Kotlin è un linguaggio di programmazione *general purpose*, multi-paradigma, *open source* sviluppato dall'azienda JetBrains, si basa sulla Java Virtual Machine ed è ispirato ad altri linguaggi di programmazione come Java. Kotlin è un linguaggio a [tipizzazione statica](#)^[g] con [inferenza di tipo](#)^[g], ed è particolarmente orientato alla programmazione a oggetti permettendo anche un approccio funzionale. Kotlin è il linguaggio da preferire per lo sviluppo di app Android come dichiarato da Google il 7 maggio 2019.

Gradle

Gradle è un sistema *open source* per l'automazione dello sviluppo fondato sulle idee di Apache Ant e Apache Maven, usa un [Domain Specific Language \(DSL\)](#)^[g] basato su Groovy per dichiarare la configurazione del progetto invece di [eXtensible Markup Language \(XML\)](#)^[g] come accade per Apache Maven.

Material Design

Il Material Design è un design sviluppato da Google, le cui regole di progettazione si concentrano su un maggiore uso di layout basati su una griglia, animazioni e transizioni

ed effetti di profondità come l'illuminazione e le ombre. Material Design è supportato nativamente a partire da Android 5.0.

5.1.1 Descrizione del Bluetooth Low Energy

Bluetooth Low Energy (abbreviato BLE) è una tecnologia *wireless Personal Area Network (PAN)*^[8] progettata e commercializzata dal Bluetooth *Special Interest Group* per nuove applicazioni nel settore dell'assistenza sanitaria, per i beacon e per l'industria dell'intrattenimento domestico. Rispetto al Bluetooth "classico", il Bluetooth Low Energy ha lo scopo di fornire un consumo energetico e un costo notevolmente ridotto, mantenendo un intervallo di comunicazione simile, questo consente alle applicazioni di comunicare con dispositivi che hanno rigidi requisiti di consumo dell'energia, come sensori di prossimità, dispositivi per il fitness o cardiofrequenzimetri. L'interazione tra le periferiche avviene tramite *service*, ovvero una raccolta di caratteristiche correlate per una funzione particolare. Una caratteristica è un insieme di dati trasferiti, ad esempio la temperatura di un apparecchio. La lettura delle caratteristiche avviene secondo determinati profili, vale a dire un insieme di specifiche, le quali determinano come funziona un dispositivo e che quindi stabiliscono le modalità di lettura in una particolare applicazione.

5.2 Progettazione

5.2.1 Architettura

Nello sviluppo dell'applicazione purtroppo non ho fatto in tempo a implementare l'architettura, seguendo i principi *Representational state transfer (REST)*^[8], che era stata pianificata per lo stage. Al termine del tirocinio l'applicazione era suddivisa in diversi package come mostrato in figura 5.1.

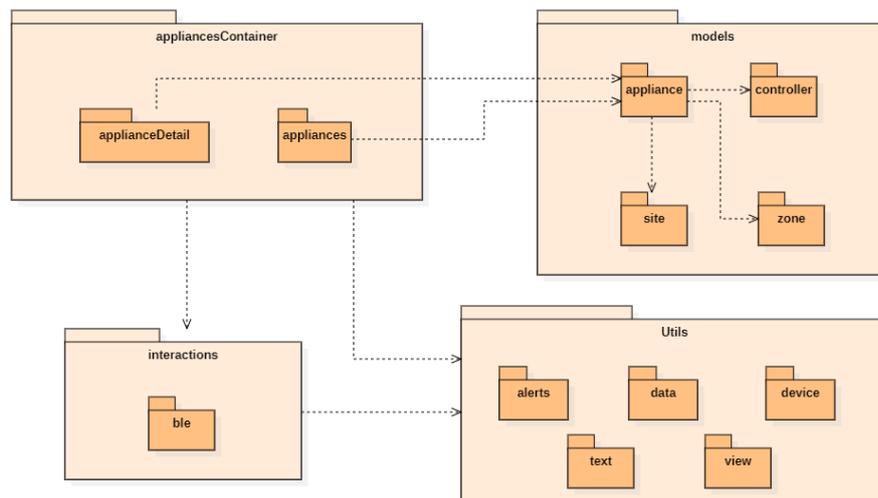


Figura 5.1: Package dell'applicazione

5.2.2 Descrizione dei *package*

appliancesContainer

Il package si occupa della visualizzazione delle interfacce utente in particolare:

- * **appliances**: mostra la lista di riscaldatori rilevati nelle vicinanze e permette l'interazione con quest'ultimi;
- * **applianceDetail**: mostra i dettagli di uno specifico *appliance* e ne consente la modifica.

models

Contiene tutti i modelli e fornisce una rappresentazione dei dati dell'applicazione. Viene suddiviso in:

- * **appliance**: fornisce tutte le informazioni di un riscaldatore;
- * **controller**: si occupa dell'interazione con un *appliance* e del suo stato di connessione;
- * **site**: si occupa di gestire i dati riguardanti la posizione del riscaldatore;
- * **zone**: fornisce una rappresentazione di una zona a cui possono essere assegnati i dispositivi.

interactions

Gestisce la parte di comunicazione con i riscaldatori ed, in particolare, **ble** si occupa dell'interazione tramite tecnologia Bluetooth Low Energy, il package è a sua volta suddiviso in:

- * **requests**: contiene le tipologie di richieste che possono essere inviate al riscaldatore;
- * **responses**: contiene le tipologie di risposte che si possono ricevere.

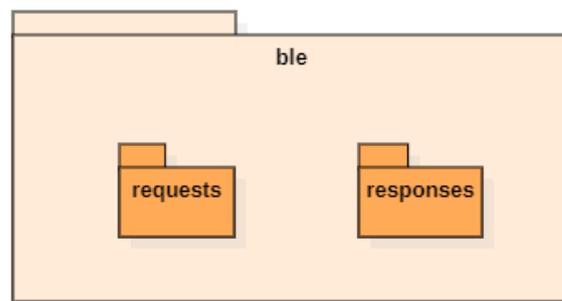


Figura 5.2: Struttura *package* ble

utils

Contiene classi le quali offrono funzionalità ausiliari come la formattazione di stringhe o conversioni di valori.

5.3 Codifica

Per l'applicazione sono state sviluppate principalmente due schermate, ossia la schermata principale che mostra la lista di riscaldatori e la schermata di dettaglio di quest'ultimi. Di seguito viene fornita una descrizione delle possibili operazioni che possono essere eseguite dall'utente. I requisiti soddisfatti per ogni interfaccia si trovano nell'appendice B.

5.3.1 Schermata principale

Descrizione interfaccia utente

In questa schermata l'utente dopo aver acceso il bluetooth può visualizzare la lista di riscaldatori scannerizzati e impostare i loro controlli come lo stato e la potenza del riscaldamento o delle luci (se presenti).

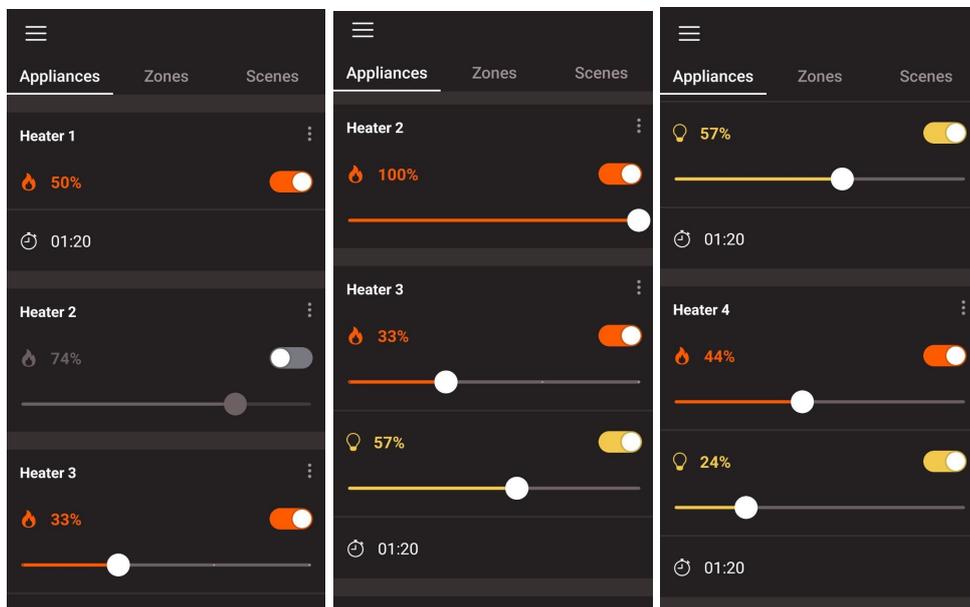


Figura 5.3: Schermata appliance list con riscaldatori di categoria 1, 2, 3 e 4

Nella schermata, inoltre, se non vengono rilevati *appliance* installati verrà visualizzato un bottone per aggiungere dei riscaldatori come mostrato in figura 5.4.

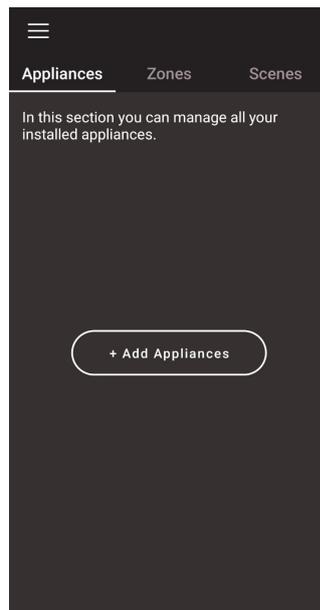


Figura 5.4: Schermata appliance list vuota

Descrizione classi

Nella visualizzazione dell'interfaccia vengono usate diverse classi:

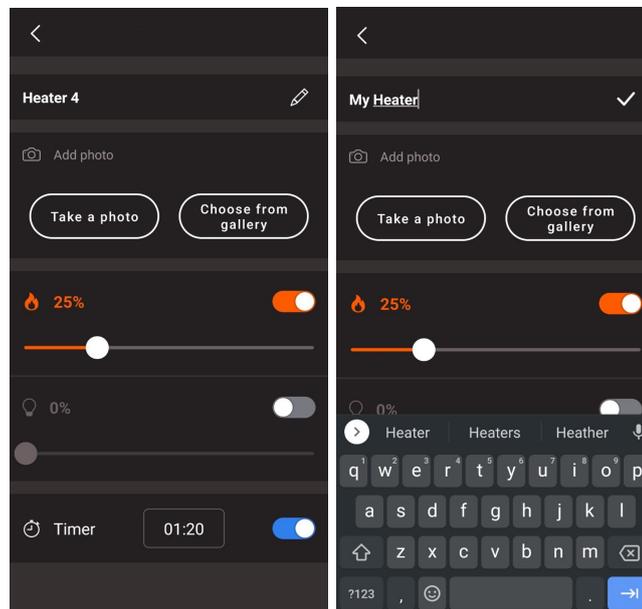
- * **MainActivity**: permette di visualizzare la lista di opzioni della schermata principale ovvero *appliances*, *zones*, *scenes* attraverso il widget *tablayout*;
- * **AppliancesListFragment**: visualizza la lista dei riscaldatori rilevati attraverso il widget *recyclerview* e mantiene lo stato della [User Interface \(UI\)](#)^[g];
- * **AppliancesListAdapter**: si occupa di costruire i vari *viewholder* della lista e di effettuarne il *binding*;
- * **ApplianceCategory1ViewHolder**: visualizza il layout dei riscaldatori di categoria 1, il quale è composto da una *textview* per il nome e uno *switch* per l'accensione e lo spegnimento;
- * **ApplianceCategory2ViewHolder**: visualizza il layout dei riscaldatori di categoria 2, il quale è uguale alla prima categoria ma viene aggiunto uno *slider* per regolare la potenza di riscaldamento;
- * **ApplianceCategory3ViewHolder**: visualizza il layout dei riscaldatori di categoria 3, questo caso è simile alla categoria 2, ma vengono aggiunti dei widget per l'attivazione o la regolazione dell'illuminazione, inoltre il riscaldamento può essere impostato solo su determinati valori;
- * **ApplianceCategory4ViewHolder**: visualizza il layout dei riscaldatori di categoria 4, simile al terzo ma il riscaldamento può essere regolato liberamente.

Descrizione degli obiettivi raggiunti

Per la schermata principale si doveva sviluppare le tre sezioni *appliance*, *zones* e *scenes* visibili in cima all'interfaccia (5.3 o 5.4), mentre il menù accessibile attraverso l'icona in alto a sinistra non era previsto nel progetto. Al termine dello stage *zones* e *scenes* non sono state implementate per mancanza di tempo. La schermata doveva, inoltre, permettere di controllare se il bluetooth era abilitato sul dispositivo e, in caso negativo, visualizzare un messaggio d'errore chiedendo all'utente di attivarlo, per quanto riguarda questi requisiti sono riuscito a realizzare la parte di controllo dei permessi e di attivazione ma non la parte grafica.

5.3.2 *Appliance Detail*

In questa schermata l'utente può modificare i dettagli di un particolare riscaldatore. L'interfaccia consente di cambiare il nome, aggiungere o rimuovere la foto, modificare i controlli come accadeva nella lista degli *appliance* ed infine impostare il *countdown timer*.



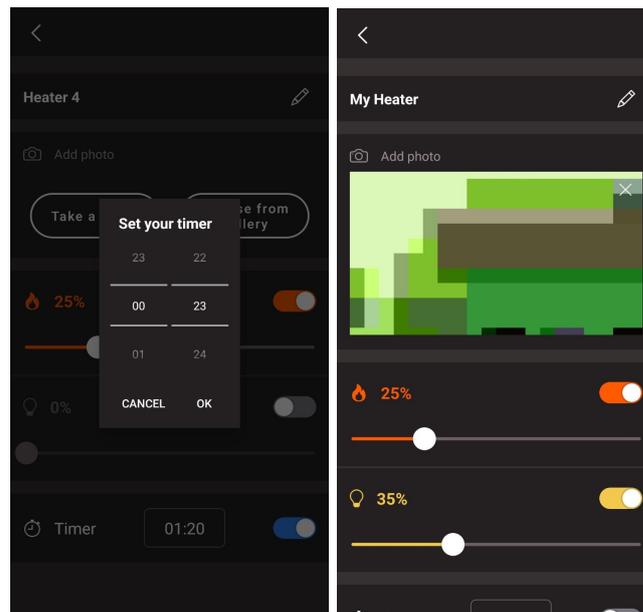


Figura 5.5: Schermata appliance detail, modifica nome, impostazione timer e visualizzazione foto

Descrizione classi

Nella visualizzazione dell'interfaccia vengono usate diverse classi:

- * **ApplianceDetailActivity:** visualizza l'intera schermata che è composta dalla barra di navigazione con il bottone per tornare indietro e i dettagli;
- * **ApplianceDetailAdapter:** costruisce, mantiene la posizione e effettua il *binding* dei diversi *viewholder* all'interno del *recyclerview*;
- * **InfoCellViewHolder:** visualizza la sezione per la modifica del nome dell'*appliance*, la quale è composta da un bottone che consente di visualizzare un *edittext* per cambiare il nome oppure una *textview* se non si è nello stato di modifica;
- * **PhotoCellViewHolder:** visualizza la foto del riscaldatore oppure due bottoni per inserire un'immagine tramite fotocamera o galleria;
- * **ControlsCellViewHolder:** visualizza i controlli di un riscaldatore;
- * **TimerCellViewHolder:** visualizza uno *switch* per attivare o disattivare il timer e una *textview* per visualizzare il tempo o far apparire una finestra per modificare l'ora;
- * **ButtonsCellViewHolder:** contiene il bottone per eliminare il riscaldatore.

Descrizione degli obiettivi raggiunti

Per questa interfaccia di dettaglio sono stati raggiunti i requisiti individuati, l'utente può visualizzare il nome di un riscaldatore e premere un bottone per passare alla modalità di modifica e ripremerlo per confermare il nuovo nome. Nella schermata

se l'*appliance* presenta già una foto questa verrà visualizzata e potrà essere rimossa tramite un bottone in alto a destra, se invece non possiede nessuna immagine vengono visualizzati due pulsanti che permettono rispettivamente di aggiungere l'immagine tramite fotocamera o galleria del telefono. Quando si tenta di aprire la fotocamera o la galleria vengono preventivamente controllati i permessi concessi in modo tale che se non sono presenti viene visualizzata una finestra di sistema che chiederà all'utente di fornire quest'ultimi all'applicazione. All'interno della schermata è presente una sezione per abilitare e regolare i controlli (riscaldamento e luci) del riscaldatore e una per impostare il *timer*.

5.3.3 Comunicazione bluetooth

In entrambe le interfacce mostrate in precedenza si fa utilizzo della classe **BLEManager** per effettuare le operazioni di scan, connessione, trasmissione e ricezione. Nella comunicazione, inoltre, le richieste e le risposte sono state rappresentate attraverso delle classi per leggere e scrivere i dati in forma di byte. Nel diagramma di sequenza in figura 5.6 vi è un esempio di come avviene la scansione e connessione ad un dispositivo utilizzando la classe BLEManager e le [callback](#)^[8] messe a disposizione da Android.

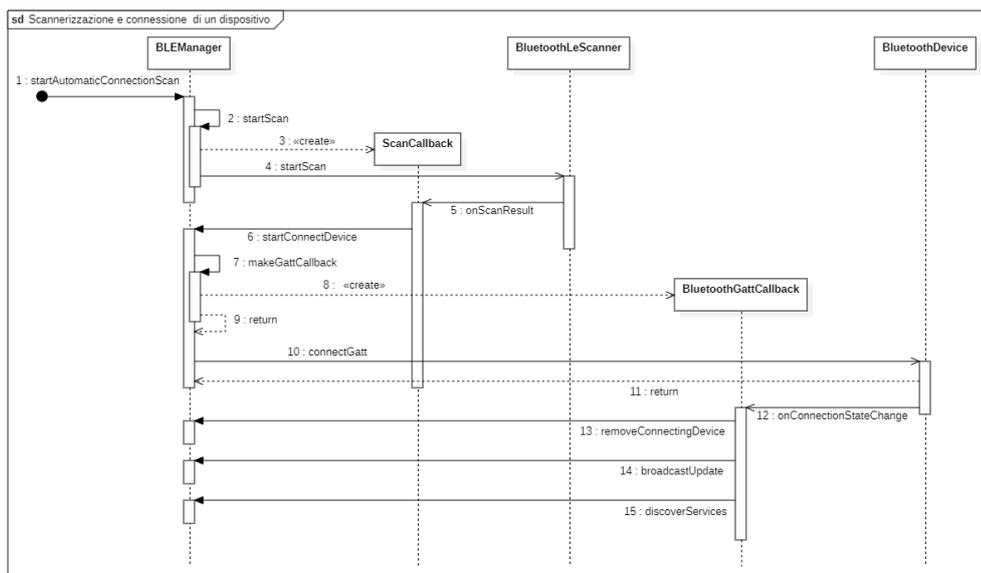


Figura 5.6: Diagramma di sequenza della scannerizzazione e connessione di un riscaldatore

Dopo aver chiamato *startAutomaticConnectionScan* si fa partire la scannerizzazione dei dispositivi vicini, questi vengono poi filtrati tramite nome poiché lo scan potrebbe rilevare altre periferiche diverse dai riscaldatori. Quando viene trovato un dispositivo valido si tenta la connessione attraverso *startConnectDevice* che crea la *BluetoothGattCallback* per gestire lo stato della connessione al riscaldatore. Terminato il tentativo, se è andato a buon fine, viene chiamato *onConnectionStateChange* per informare il BLEManager e procedere con l'inserimento nella lista dei dispositivi bluetooth connessi e il *discover* dei servizi dell'*appliance*.

5.4 Verifica e Validazione

Per la parte di verifica e validazione non sono stati scritti dei test in quanto il tempo rimasto a mia disposizione dopo aver sviluppato le schermate non era sufficiente. Le uniche verifiche effettivamente compiute sono state il controllo del codice sorgente grazie al compilatore di kotlin e qualche mio test manuale usando il logcat di Android Studio.

Capitolo 6

Conclusioni

In quest'ultimo capitolo vengono riportate le riflessioni personali sull'esperienza di tirocinio e gli obiettivi raggiunti.

6.1 Consuntivo dello stage

I compiti della prima settimana sono stati svolti senza troppe difficoltà, nonostante abbia iniziato in ritardo. La seconda settimana, come scritto nella sezione 3.3.1, per problemi di malattia c'è stato un piccolo rallentamento, anche se ho comunque provato a continuare da casa. Ho passato più tempo del previsto nella fase di studio, di conseguenza ho iniziato più tardi l'implementazione del progetto e ciò non mi ha permesso di completare le attività di testing, validazione e documentazione. Dalla fine della quarta settimana mi sono dedicato alla schermata principale, mentre avrei dovuto cominciare la schermata dei dettagli, dunque ho cercato di accelerare lo sviluppo in modo tale da realizzare almeno la comunicazione bluetooth entro le ultime due settimane.

Settimana	Attività	Ore
1	Comprensione sistema e obiettivi.	15
1	Analisi dei requisiti.	25
2-3	Progettazione.	60
3-4	Studio e setup ambiente di sviluppo.	40
4-8	Implementazione.	180
-	Test e validazione.	0
-	Documentazione.	0
	Totale	320

Tabella 6.1: Tabella del consuntivo finale di stage

6.2 Raggiungimento degli obiettivi

Al termine dello stage risultano diversi obiettivi non completati, in particolare 2 obiettivi obbligatori, mentre altri risultano parzialmente completati poiché non sono riuscito a realizzare tutte le interfacce.

Codice	Descrizione	Stato
OO1	Gestione e pianificazione del progetto attraverso kanban board condivisa.	completato
OO2	Analisi e progettazione applicazione Mobile, a partire dai requisiti raccolti.	completato
OO3	Analisi ed implementazione flussi login/logout/-signup.	non completato
OO4	Analisi ed integrazione flusso installazione BLE.	parzialmente completato
OO5	Analisi ed integrazione flusso configurazione IP.	non completato
OO6	Integrazione gestione comandi BLE.	parzialmente completato
OO7	Implementazione interfacce gestione Appliance, Zone, Scene.	parzialmente completato
OD1	Integrazione comandi tramite Cloud.	non completato
OD2	Suite di testing del software prodotto.	non completato
OD3	Documentazione completa.	non completato
OZ1	Ulteriori modifiche all'applicazione che esulano da quando riportato in questo documento.	non completato

Tabella 6.2: Tabella del raggiungimento degli obiettivi di tirocinio

6.3 Conoscenze acquisite

Lo stage mi ha permesso di vedere come funzionano determinate tecnologie come il bluetooth e di imparare un linguaggio di programmazione più recente come Kotlin. Il tirocinio mi ha introdotto per la prima volta allo sviluppo mobile Android, al lavoro di gruppo in un contesto aziendale e, di conseguenza, alle tecnologie e sistemi che vengono impiegati.

6.4 Valutazione personale

Lo stage mi ha permesso per la prima volta di affrontare un ambiente di lavoro reale e di conoscere nuove persone. L'esperienza mi ha introdotto allo sviluppo mobile, un ambito che non avevo mai esplorato e che, dopo aver fatto il tirocinio, trovo molto interessante. Per quanto riguarda gli obiettivi raggiunti speravo di completarne di più,

purtroppo ho speso tanto tempo sulla parte di apprendimento e avrei dovuto rivolgermi più spesso al tutor aziendale per risolvere alcuni dubbi, inoltre la settimana di malattia ha leggermente allungato i tempi. Il team di Moku si è dimostrato molto disponibile a chiarire ogni mio problema sia dal punto di vista tecnico che organizzativo, inoltre mi hanno fornito diversi consigli riguardo i programmi da utilizzare e l'impostazione dell'ambiente di lavoro. L'esperienza mi ha permesso di applicare quanto visto durante il corso di laurea e di vedere le soluzioni che vengono adottate in un progetto reale. Nel complesso mi ritengo soddisfatto del progetto assegnato, di quanto svolto e di aver scelto Moku come azienda.

Appendice A

Appendice A

In questa sezione del documento vengono descritti i casi individuati durante la fase di analisi del prodotto attraverso dei diagrammi [Unified Modeling Language \(UML\)](#)^[g] e i rispettivi requisiti. Non vengono mostrate ulteriori funzionalità individuate in quanto non facenti parte del progetto di stage.

A.1 Casi d'uso

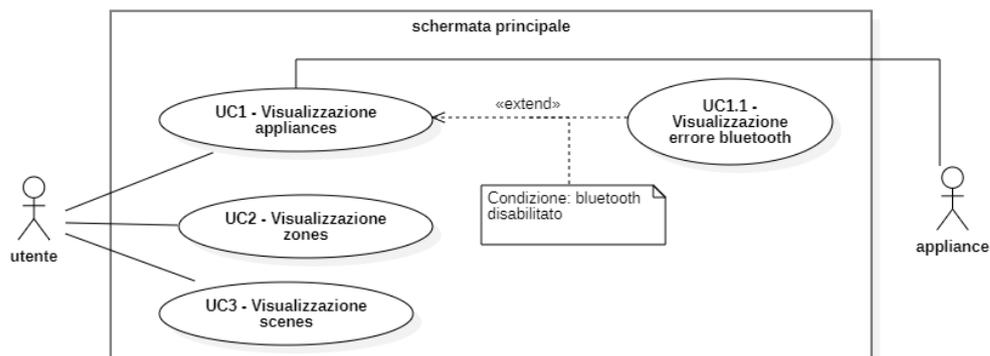


Figura A.1: Scenario schermata principale

UC1: Visualizzazione *appliances*

- * **Attori Primari:** utente.
- * **Attori Secondari:** *appliance*.
- * **Precondizioni:** L'utente ha aperto la scheda di visualizzazione dei riscaldatori.
- * **Descrizione:** La scheda *appliances* deve permettere all'utente di visualizzare una lista dei dispositivi rilevati nelle vicinanze.
- * **Postcondizioni:** L'utente visualizza una lista di *appliances*.

- * **Scenario Alternativo:** L'utente accede alla schermata con il bluetooth disattivato e visualizza un messaggio d'errore [UC1.1](#).

UC1.1: Visualizzazione errore bluetooth

- * **Attori Primari:** utente.
- * **Precondizioni:** L'utente ha aperto la scheda di visualizzazione dei riscaldatori ed ha il bluetooth disabilitato.
- * **Descrizione:** L'utente visualizza un messaggio di errore che lo informa di abilitare il bluetooth per rilevare i dispositivi nelle vicinanze.
- * **Postcondizioni:** L'utente visualizza il messaggio d'errore.

UC2: Visualizzazione *zones*

- * **Attori Primari:** utente.
- * **Precondizioni:** L'utente ha aperto la scheda di visualizzazione delle zone.
- * **Descrizione:** La scheda *zones* deve permettere all'utente di visualizzare una lista di zone a cui sono associati gli appliance.
- * **Postcondizioni:** L'utente visualizza una lista di zone.

UC3: Visualizzazione *scenes*

- * **Attori Primari:** utente.
- * **Precondizioni:** L'utente ha aperto la scheda di visualizzazione delle scene.
- * **Descrizione:** La scheda *scenes* deve permettere all'utente di visualizzare una lista di scene, ovvero un insieme di comandi che possono essere pianificati in base a diversi parametri.
- * **Postcondizioni:** L'utente visualizza una lista di scene.

UC4: Modifica nome *appliance*

- * **Attori Primari:** utente.
- * **Precondizioni:** L'utente ha aperto la scheda dei dettagli dell'*appliance* e vuole modificarne il nome.

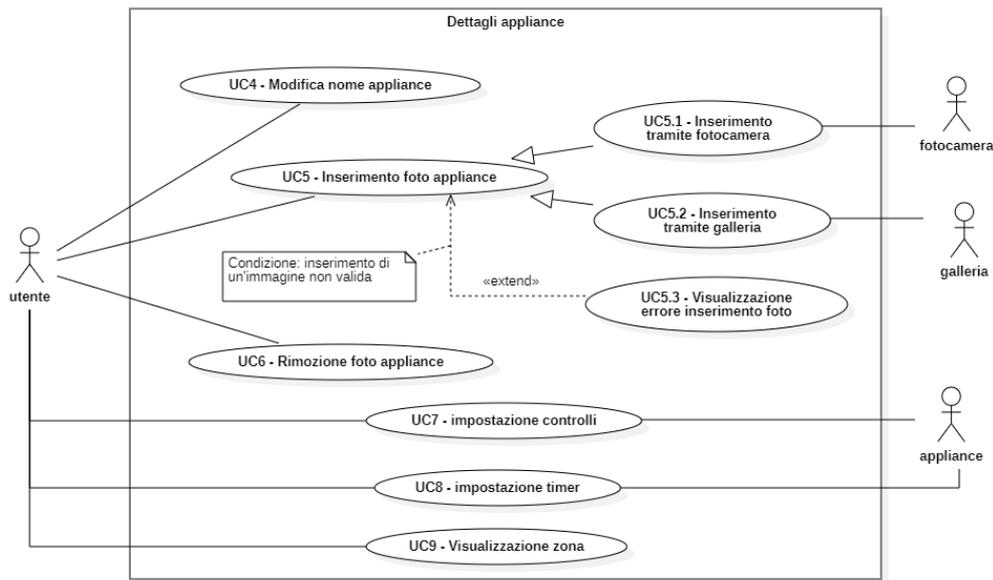


Figura A.2: Scenario visualizzazione dei dettagli di un *appliance*

- * **Descrizione:** L'utente deve poter modificare il nome del riscaldatore e confermare l'operazione.
- * **Postcondizioni:** Il nome dell'*appliance* è stato cambiato.

UC5: Inserimento foto *appliance*

- * **Attori Primari:** utente.
- * **Precondizioni:** L'utente ha aperto la scheda dei dettagli dell'*appliance* e non è presente nessuna foto.
- * **Descrizione:** L'utente deve poter inserire una foto.
- * **Postcondizioni:** Viene aggiunta la foto al riscaldatore.
- * **Scenario Alternativo:** L'immagine scelta dall'utente non è valida e perciò deve visualizzare un messaggio d'errore [UC5.3](#).

UC5.1: Inserimento tramite fotocamera

- * **Attori Primari:** utente.
- * **Attori Secondari:** fotocamera.

- * **Precondizioni:** L'utente ha aperto la scheda dei dettagli dell'*appliance* e seleziona l'inserimento dell'immagine tramite fotocamera.
- * **Descrizione:** L'utente deve poter scattare una foto tramite l'applicazione fotocamera e confermare l'operazione.
- * **Postcondizioni:** Viene aggiunta la foto scattata al riscaldatore.

UC5.2: Inserimento tramite galleria

- * **Attori Primari:** utente.
- * **Attori Secondari:** galleria.
- * **Precondizioni:** L'utente ha aperto la scheda dei dettagli dell'*appliance* e seleziona l'inserimento dell'immagine tramite galleria.
- * **Descrizione:** L'utente deve poter selezionare una foto dalla galleria del telefono e confermare l'operazione.
- * **Postcondizioni:** Viene aggiunta una foto al riscaldatore.

UC5.3: Visualizzazione errore inserimento foto

- * **Attori Primari:** utente.
- * **Precondizioni:** L'utente seleziona un'immagine non valida.
- * **Descrizione:** L'utente visualizza un messaggio d'errore a causa dell'inserimento di un'immagine non valida.
- * **Postcondizioni:** L'utente riceve il messaggio d'errore nell'inserimento della foto.

UC6: Rimozione foto *appliance*

- * **Attori Primari:** utente.
- * **Precondizioni:** L'utente ha aperto la scheda dei dettagli dell'*appliance* ed è presente una foto.
- * **Descrizione:** L'utente deve poter rimuovere la foto visualizzata.
- * **Postcondizioni:** Viene rimossa la foto del riscaldatore.

UC7: Impostazione controlli

- * **Attori Primari:** utente.
- * **Attori Secondari:** *appliance*.
- * **Precondizioni:** L'utente ha aperto la scheda dei dettagli dell'*appliance* e vuole modificare i controlli.
- * **Descrizione:** L'utente deve poter modificare i controlli dell'*appliance* come l'attivazione e disattivazione del riscaldamento.
- * **Postcondizioni:** Vengono impostati i controlli.

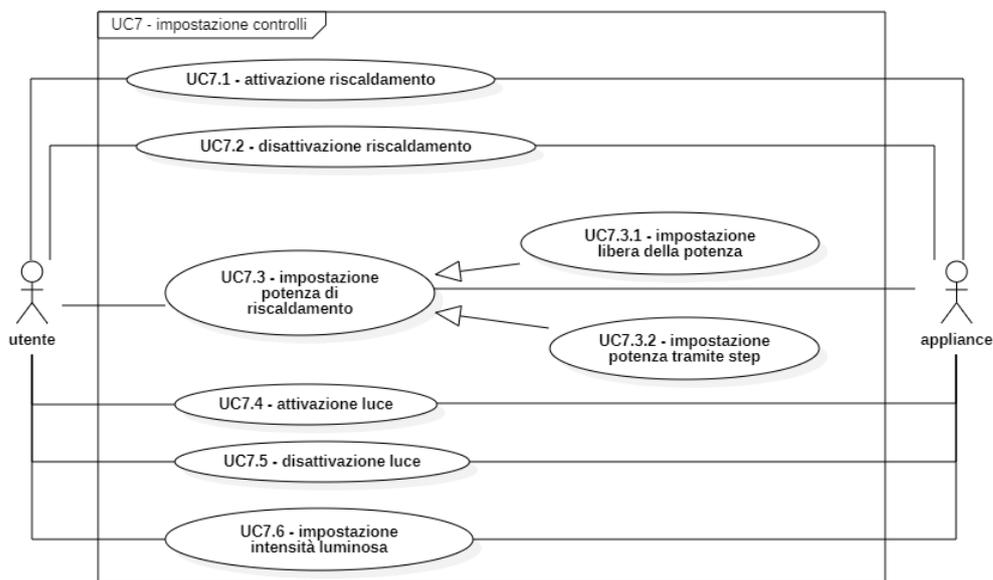


Figura A.3: UC7 - impostazione controlli

UC7.1: Attivazione riscaldamento

- * **Attori Primari:** utente.
- * **Attori Secondari:** *appliance*.
- * **Precondizioni:** L'utente visualizza i controlli e vuole accendere il riscaldamento.
- * **Descrizione:** L'utente deve poter attivare il riscaldamento.
- * **Postcondizioni:** Viene notificato l'*appliance* dell'accensione del riscaldamento.

UC7.2: Disattivazione riscaldamento

- * **Attori Primari:** utente.
- * **Attori Secondari:** *appliance*.
- * **Precondizioni:** L'utente visualizza i controlli e vuole spegnere il riscaldamento.
- * **Descrizione:** L'utente deve poter disattivare il riscaldamento.
- * **Postcondizioni:** Viene notificato l'*appliance* dello spegnimento del riscaldamento.

UC7.3: Impostazione potenza di riscaldamento

- * **Attori Primari:** utente.
- * **Attori Secondari:** *appliance*.
- * **Precondizioni:** L'utente visualizza i controlli e il riscaldamento è acceso.
- * **Descrizione:** L'utente deve poter regolare la potenza di riscaldamento dell'*appliance*.
- * **Postcondizioni:** Viene notificato l'*appliance* con il nuovo valore della potenza.

UC7.3.1: Impostazione libera della potenza

- * **Attori Primari:** utente.
- * **Attori Secondari:** *appliance*.
- * **Precondizioni:** L'utente visualizza i controlli e il riscaldamento è acceso.
- * **Descrizione:** L'utente seleziona un valore a scelta tra quello minimo e massimo di potenza.
- * **Postcondizioni:** Viene notificato l'*appliance* con il nuovo valore della potenza.

UC7.3.2: Impostazione potenza tramite step

- * **Attori Primari:** utente.
- * **Attori Secondari:** *appliance*.

- * **Precondizioni:** L'utente visualizza i controlli e il riscaldamento è acceso.
- * **Descrizione:** L'utente seleziona un valore di potenza tra gli step disponibili.
- * **Postcondizioni:** Viene notificato l'*appliance* con il nuovo valore della potenza.

UC7.4: Attivazione luce

- * **Attori Primari:** utente.
- * **Attori Secondari:** *appliance*.
- * **Precondizioni:** L'utente visualizza i controlli e vuole accendere la luce del riscaldatore.
- * **Descrizione:** L'utente deve poter attivare la luce.
- * **Postcondizioni:** Viene notificato l'*appliance* dell'accensione della luce.

UC7.5: Disattivazione luce

- * **Attori Primari:** utente.
- * **Attori Secondari:** *appliance*.
- * **Precondizioni:** L'utente visualizza i controlli e vuole spegnere la luce.
- * **Descrizione:** L'utente deve poter disattivare la luce.
- * **Postcondizioni:** Viene notificato l'*appliance* dello spegnimento della luce.

UC7.6: Impostazione intensità luminosa

- * **Attori Primari:** utente.
- * **Attori Secondari:** *appliance*.
- * **Precondizioni:** L'utente visualizza i controlli e la luce è accesa.
- * **Descrizione:** L'utente deve poter regolare l'intensità luminosa dell'*appliance* selezionando un valore a scelta in un intervallo.
- * **Postcondizioni:** Viene notificato l'*appliance* con il nuovo valore dell'intensità luminosa.

UC8: Impostazione *timer*

- * **Attori Primari:** utente.
- * **Attori Secondari:** *appliance*.
- * **Precondizioni:** L'utente ha aperto la scheda dei dettagli dell'*appliance* e vuole impostare il *timer*.
- * **Descrizione:** L'utente deve poter impostare i vari parametri del *timer* come l'accensione o lo spegnimento.
- * **Postcondizioni:** Viene impostato il timer del riscaldatore.

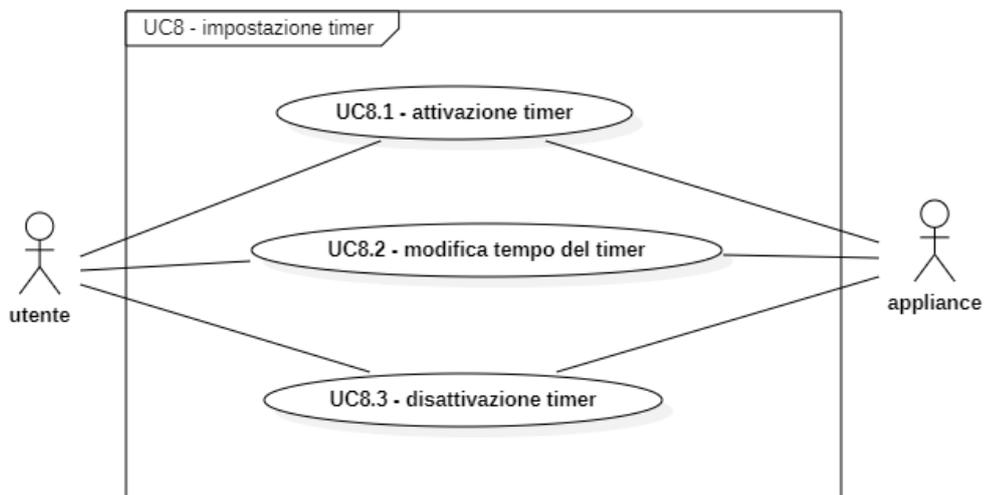


Figura A.4: UC8 - impostazione timer

UC8.1: Attivazione *timer*

- * **Attori Primari:** utente.
- * **Attori Secondari:** *appliance*.
- * **Precondizioni:** L'utente visualizza le impostazioni per il *timer*.
- * **Descrizione:** L'utente deve poter accendere il *timer*.
- * **Postcondizioni:** Viene attivato il *timer* e notificato al riscaldatore.

UC8.2: Modifica tempo del *timer*

- * **Attori Primari:** utente.
- * **Attori Secondari:** *appliance*.
- * **Precondizioni:** L'utente visualizza le impostazioni ed il *timer* è attivo.
- * **Descrizione:** L'utente deve poter impostare le ore e i minuti e confermare l'operazione.
- * **Postcondizioni:** Viene aggiornato il tempo del *timer* e notificato al riscaldatore.

UC8.3: Disattivazione *timer*

- * **Attori Primari:** utente.
- * **Attori Secondari:** *appliance*.
- * **Precondizioni:** L'utente visualizza le impostazioni per il *timer*.
- * **Descrizione:** L'utente deve poter spegnere il *timer*.
- * **Postcondizioni:** Viene disattivato il *timer* e notificato al riscaldatore.

UC9: Visualizzazione zona

- * **Attori Primari:** utente.
- * **Precondizioni:** L'*appliance* appartiene ad una zona e l'utente ha aperto la schermata dei dettagli.
- * **Descrizione:** L'utente deve poter visualizzare il nome della zona o uno spazio vuoto se non è presente.
- * **Postcondizioni:** L'utente visualizza la zona del riscaldatore.

A.2 Tracciamento dei requisiti

A.2.1 Elenco dei requisiti

Da un'attenta analisi dei requisiti e degli use case effettuata sul progetto è stata stilata la tabella che traccia i requisiti in rapporto agli use case.

Sono stati individuati diversi tipi di requisiti e si è quindi fatto utilizzo di un codice identificativo per distinguerli.

Il codice dei requisiti è così strutturato:

R[TIPO] [PADRE] . [FIGLIO]

	Descrizione
TIPO	Indica la categoria di requisito: F : funzionale Q : qualità V : vincolo P : prestazionale
PADRE	numero identificativo del requisito padre
FIGLIO	numero identificativo del requisito figlio

Tabella A.1: Tabella del codice di un requisito

Nelle tabelle [A.2](#), [A.3](#), [A.4](#) e [A.5](#) sono riassunti i requisiti e il loro tracciamento con gli use case delineati in fase di analisi.

Requisito	Descrizione	Fonte
RF1	L'interfaccia permette di visualizzare i riscaldatori scannerizzati	UC1
RF1.1	L'applicazione deve controllare se il bluetooth è abilitato per visualizzare i riscaldatori scannerizzati	UC1.1
RF1.2	L'applicazione visualizza un messaggio d'errore nel caso in cui il bluetooth sia disabilitato per visualizzare i riscaldatori scannerizzati	UC1.1
RF2	L'interfaccia permette di visualizzare una lista di zone	UC2
RF3	L'interfaccia permette di visualizzare una lista di scene	UC3
RF4	L'interfaccia dei dettagli permette di visualizzare il nome di un <i>appliance</i>	UC4
RF4.1	L'interfaccia dei dettagli permette di modificare il nome di un <i>appliance</i>	UC4
RF5	L'interfaccia dei dettagli permette di inserire la foto di un <i>appliance</i>	UC5
RF5.1	L'interfaccia dei dettagli permette di inserire una foto tramite la fotocamera del telefono	UC5.1
RF5.2	L'interfaccia dei dettagli permette di inserire una foto tramite la galleria del telefono	UC5.2

RF5.3	L'applicazione visualizza un messaggio d'errore nel caso in cui venga inserita un'immagine non valida	UC5.3
RF6	L'interfaccia dei dettagli permette di rimuovere la foto dell' <i>appliance</i>	UC6
RF7	L'interfaccia dei dettagli permette di impostare i controlli del riscaldatore	UC7
RF7.1	L'applicazione deve consentire all'utente di accendere il riscaldamento	UC7.1
RF7.2	L'applicazione deve consentire all'utente di spegnere il riscaldamento	UC7.2
RF7.3	L'applicazione deve consentire all'utente di impostare la potenza di riscaldamento	UC7.3
RF7.4	L'applicazione deve consentire all'utente di scegliere la potenza di riscaldamento liberamente all'interno di un intervallo	UC7.3.1
RF7.5	L'applicazione deve consentire all'utente di scegliere la potenza di riscaldamento in un insieme ristretto di valori	UC7.3.2
RF7.6	L'applicazione deve consentire all'utente di accendere la luce collegata al riscaldatore	UC7.4
RF7.7	L'applicazione deve consentire all'utente di spegnere la luce collegata al riscaldatore	UC7.5
RF7.8	L'applicazione deve consentire all'utente di impostare l'intensità della luce dell' <i>appliance</i>	UC7.6
RF8	L'interfaccia dei dettagli permette di impostare i parametri del <i>timer</i>	UC8
RF8.1	L'applicazione deve consentire all'utente di abilitare il <i>timer</i>	UC8.1
RF8.2	L'applicazione deve consentire all'utente di scegliere un nuovo tempo di <i>countdown</i>	UC8.2
RF8.3	L'applicazione deve consentire all'utente di disabilitare i parametri del <i>timer</i>	UC8.3
RF9	L'applicazione deve consentire all'utente di visualizzare la zona dell' <i>appliance</i>	UC9

Tabella A.2: Tabella del tracciamento dei requisiti funzionali

Requisito	Descrizione	Fonte
-----------	-------------	-------

RQ1	L'applicazione deve essere sviluppata in Kotlin.	Azienda
RQ2	I file dell'applicazione devono essere versionati tramite Git con il paradigma Git-flow.	Azienda
RQ3	I file dell'applicazione devono essere salvati nel repository del progetto utilizzando Bitbucket e Sourcetree.	Azienda

Tabella A.3: Tabella del tracciamento dei requisiti qualitativi

Requisito	Descrizione	Fonte
RV1	L'applicazione deve usare la tecnologia BLE per comunicare con i riscaldatori.	Azienda

Tabella A.4: Tabella del tracciamento dei requisiti di vincolo

Requisito	Descrizione	Fonte
RP1	L'applicazione deve scannerizzare i dispositivi vicini per 15 secondi.	Azienda

Tabella A.5: Tabella del tracciamento dei requisiti prestazionali

A.2.2 Tracciamento Fonte - Requisiti

Fonte	Requisiti
UC1	RF1
UC1.1	RF1.1
UC1.1	RF1.2
UC2	RF2
UC3	RF3
UC4	RF4
UC4.1	RF4.1
UC5	RF5
UC5.1	RF5.1
UC5.2	RF5.2

UC5.3	RF5.3
UC6	RF6
UC7	RF7
UC7.1	RF7.1
UC7.2	RF7.2
UC7.3	RF7.3
UC7.3.1	RF7.4
UC7.3.2	RF7.5
UC7.4	RF7.6
UC7.5	RF7.7
UC7.6	RF7.8
UC8	RF8
UC8.1	RF8.1
UC8.2	RF8.2
UC8.3	RF8.3
UC9	RF9
Azienda	RQ1
Azienda	RQ2
Azienda	RQ3
Azienda	RV1
Azienda	RP1

Tabella A.6: Tabella del tracciamento fonte - requisiti

Appendice B

Appendice B

In questo capitolo vengono riportati i requisiti soddisfatti durante il periodo di tirocinio. I requisiti funzionali sono divisi per schermata.

I requisiti riguardanti le zone e le scene in tabella B.1 risultano non soddisfatti in quanto le schermate non sono state implementate per tempo.

Requisito	Descrizione	Stato
RF1	L'interfaccia permette di visualizzare i riscaldatori scannerizzati	soddisfatto
RF1.1	L'applicazione deve controllare se il bluetooth è abilitato per visualizzare i riscaldatori scannerizzati	soddisfatto
RF1.2	L'applicazione visualizza un messaggio d'errore nel caso in cui il bluetooth sia disabilitato per visualizzare i riscaldatori scannerizzati	non soddisfatto
RF2	L'interfaccia permette di visualizzare una lista di zone	non soddisfatto
RF3	L'interfaccia permette di visualizzare una lista di scene	non soddisfatto

Tabella B.1: Tabella del tracciamento dei requisiti funzionali soddisfatti relativi alla schermata principale

Requisito	Descrizione	Stato
RF4	L'interfaccia dei dettagli permette di visualizzare il nome di un <i>appliance</i>	soddisfatto
RF4.1	L'interfaccia dei dettagli permette di modificare il nome di un <i>appliance</i>	soddisfatto

RF5	L'interfaccia dei dettagli permette di inserire la foto di un <i>appliance</i>	soddisfatto
RF5.1	L'interfaccia dei dettagli permette di inserire una foto tramite la fotocamera del telefono	soddisfatto
RF5.2	L'interfaccia dei dettagli permette di inserire una foto tramite la galleria del telefono	soddisfatto
RF5.3	L'applicazione visualizza un messaggio d'errore nel caso in cui venga inserita un'immagine non valida	soddisfatto
RF6	L'interfaccia dei dettagli permette di rimuovere la foto dell' <i>appliance</i>	soddisfatto
RF7	L'interfaccia dei dettagli permette di impostare i controlli del riscaldatore	soddisfatto
RF7.1	L'applicazione deve consentire all'utente di accendere il riscaldamento	soddisfatto
RF7.2	L'applicazione deve consentire all'utente di spegnere il riscaldamento	soddisfatto
RF7.3	L'applicazione deve consentire all'utente di impostare la potenza di riscaldamento	soddisfatto
RF7.4	L'applicazione deve consentire all'utente di scegliere la potenza di riscaldamento liberamente all'interno di un intervallo	soddisfatto
RF7.5	L'applicazione deve consentire all'utente di scegliere la potenza di riscaldamento in un insieme ristretto di valori	soddisfatto
RF7.6	L'applicazione deve consentire all'utente di accendere la luce collegata al riscaldatore	soddisfatto
RF7.7	L'applicazione deve consentire all'utente di spegnere la luce collegata al riscaldatore	soddisfatto
RF7.8	L'applicazione deve consentire all'utente di impostare l'intensità della luce dell' <i>appliance</i>	soddisfatto
RF8	L'interfaccia dei dettagli permette di impostare i parametri del <i>timer</i>	soddisfatto
RF8.1	L'applicazione deve consentire all'utente di abilitare il <i>timer</i>	soddisfatto
RF8.2	L'applicazione deve consentire all'utente di scegliere un nuovo tempo di <i>countdown</i>	soddisfatto
RF8.3	L'applicazione deve consentire all'utente di disabilitare i parametri del <i>timer</i>	soddisfatto

RF9	L'applicazione deve consentire all'utente di visualizzare la zona dell' <i>appliance</i>	soddisfatto
-----	--	-------------

Tabella B.2: Tabella del tracciamento dei requisiti funzionali soddisfatti relativi alla schermata *Appliance Detail*

Requisito	Descrizione	Fonte
RQ1	L'applicazione deve essere sviluppata in Kotlin.	soddisfatto
RQ2	I file dell'applicazione devono essere versionati tramite Git con il paradigma Git-flow.	soddisfatto
RQ3	I file dell'applicazione devono essere salvati nel repository del progetto utilizzando Bitbucket e Sourcetree.	soddisfatto
RV1	L'applicazione deve usare la tecnologia BLE per comunicare con i riscaldatori.	soddisfatto
RP1	L'applicazione deve scannerizzare i dispositivi vicini per 15 secondi.	soddisfatto

Tabella B.3: Tabella del tracciamento dei requisiti qualitativi, di vincolo e prestazionali soddisfatti

Acronimi e abbreviazioni

- BLE** Bluetooth Low Energy. 6
- DSL** Domain Specific Language. 13
- IDE** Integrated Development Environment. 4
- PAN** Personal Area Network. 14
- REST** Representational state transfer. 14
- UI** User Interface. 17
- UML** Unified Modeling Language. 27
- VCS** Version Control System. 4
- XML** eXtensible Markup Language. 13

Glossario

agile La "metodologia agile" è un approccio allo sviluppo del software basato sulla distribuzione continua di software efficienti creati in modo rapido e iterativo. In pratica, le metodologie di sviluppo software agile consistono nel rilasciare rapidamente modifiche al software in piccole porzioni con l'obiettivo di migliorare la soddisfazione dei clienti.. [1](#)

backend Il termine inglese *backend* in informatica denota la parte che permette l'effettivo funzionamento delle interazioni con il frontend, ossia la parte visibile all'utente e con cui può interagire, come un'interfaccia grafica. Nell'ambito dello sviluppo software il *backend* può essere visto come la parte che elabora i dati ricevuti dal *frontend*. [6](#)

BLE *Bluetooth Low Energy* (abbreviato BLE) è una tecnologia *wireless* progettata e commercializzata dal Bluetooth *Special Interest Group* per nuove applicazioni nel settore dell'assistenza sanitaria, fitness, per i beacon, per la sicurezza, per l'industria dell'intrattenimento domestico e per le industrie automobilistiche e dell'automazione. Il Bluetooth Low Energy ha lo scopo di fornire un consumo energetico e un costo notevolmente ridotto, mantenendo un intervallo di comunicazione simile. Infine è indipendente ed incompatibile con il Bluetooth classico. [45](#)

callback In programmazione, una callback è, in genere, una funzione, o un "blocco di codice" che viene passata come parametro ad un'altra funzione. In particolare, quando ci si riferisce alla callback richiamata da una funzione, la callback viene passata come argomento ad un parametro della funzione chiamante. In questo modo la chiamante può realizzare un compito specifico (quello svolto dalla callback) che non è, molto spesso, noto al momento della scrittura del codice. Se ci si riferisce alla callback come funzione richiamata dal sistema operativo, di norma ciò si utilizza allo scopo di gestire particolari eventi: dal premere un bottone con il mouse, allo scrivere caratteri in un campo di testo. Ciò consente, quindi, a un programma di livello più basso, di richiamare una funzione (o servizio) definita a un livello più alto. [20](#)

DSL Un domain specific language nello sviluppo software e nell'ingegneria di dominio è un linguaggio di programmazione o un linguaggio di specifica dedicato a particolari problemi di un dominio, a una particolare tecnica di rappresentazione e/o a una particolare soluzione tecnica. [45](#)

IDE Un ambiente di sviluppo integrato, in inglese *integrated development environment* o IDE, è un software che, in fase di programmazione, supporta i programmatori

nello sviluppo e debugging del codice sorgente di un programma. Spesso l'IDE aiuta lo sviluppatore segnalando errori di sintassi del codice direttamente in fase di scrittura, oltre a tutta una serie di strumenti e funzionalità di supporto alla fase stessa di sviluppo e debugging. 45

inferenza di tipo L'inferenza di tipo si riferisce al rilevamento automatico del tipo di un'espressione in un linguaggio di programmazione senza, ad esempio, dover specificarlo esplicitamente attraverso delle parole chiave. 13

open source è un termine utilizzato per indicare software per computer rilasciato con una licenza in cui il titolare del copyright concede agli utenti i diritti di utilizzare, studiare, modificare e distribuire il software e il relativo codice sorgente a chiunque e per qualsiasi scopo.. 4

PAN In telecomunicazioni una PAN, sigla che sta per *personal area network*, è una rete informatica utilizzata per permettere la comunicazione tra diversi dispositivi (telefono, PC tascabile, ecc.) vicini a un singolo utente. I singoli dispositivi possono anche non appartenere all'utente in questione. Il raggio di azione di una PAN è tipicamente di alcuni metri. 45

REST *Representational state transfer* (REST) è uno stile architetturale per sistemi distribuiti. Il termine REST rappresenta un sistema di trasmissione di dati su HTTP senza ulteriori livelli. REST prevede che la scalabilità del Web e la crescita siano risultati di pochi principi chiave di progettazione come, ad esempio, lo stato dell'applicazione e le funzionalità sono divisi in risorse web oppure ogni risorsa è unica e indirizzabile usando sintassi universale per uso nei link ipertestuali. 45

tipizzazione statica In programmazione, la tipizzazione statica è una particolare politica di tipizzazione, ovvero di assegnazione di tipi alle variabili. Nei linguaggi a tipizzazione statica, il tipo di ogni variabile, viene stabilito direttamente nel codice sorgente dove viene assegnato esplicitamente per mezzo di parole chiave apposite, come ad esempio `int`, `long`, `float`, `char`, ecc... 13

UI l'interfaccia utente anche conosciuta come *UI*, *User Interface* è un'interfaccia uomo-macchina, ovvero ciò che si frappone tra una macchina e un utente, consentendone l'interazione reciproca. L'interfaccia utente viene generalmente realizzata come strato software esterno alla parte del sistema operativo o del programma che contiene invece la logica dell'esecuzione o elaborazione. Questo significa che i dati immessi dall'utente, o che a quest'ultimo sono resi disponibili, non sono generati all'interno del codice che gestisce l'interfaccia stessa, ma in uno strato soprastante a diretto contatto con l'utente. 45

UML in ingegneria del software *UML*, *Unified Modeling Language* (ing. linguaggio di modellazione unificato) è un linguaggio di modellazione e specifica basato sul paradigma object-oriented. L'*UML* svolge un'importantissima funzione di "lingua franca" nella comunità della progettazione e programmazione a oggetti. Gran parte della letteratura di settore usa tale linguaggio per descrivere soluzioni analitiche e progettuali in modo sintetico e comprensibile a un vasto pubblico. 45

VCS Il controllo versione (version control), in informatica, è la gestione e il tracciamento di cambiamenti multipli al codice di un software. Un Version Control System è, quindi, un programma che aiuta i team di sviluppo a gestire le modifiche che vengono apportate al codice sorgente. [45](#)

XML l'XML, ovvero *eXtensible Markup Language*, è un metalinguaggio per la definizione di linguaggi di markup, ovvero un linguaggio basato su un meccanismo sintattico che consente di definire e controllare il significato degli elementi contenuti in un documento o in un testo. [45](#)

Bibliografia

Siti web consultati

Android basics in kotlin. URL: <https://developer.android.com/courses/android-basics-kotlin/course>.

Android BLE overview. URL: <https://developer.android.com/guide/topics/connectivity/bluetooth/ble-overview>.

Android reference. URL: <https://developer.android.com/reference>.

Kotlin reference. URL: <https://kotlinlang.org/>.

Stackoverflow. URL: <https://stackoverflow.com/>.

Wikipedia. URL: <https://www.wikipedia.org/>.