Università degli studi di Napoli Parthenope



**Laundry Smart - SDD**

**Documento di Progettazione**

Progetto di Ingegneria del Software e Interazione

Uomo-Macchina

A.A. 2022/2023

*Data consegna:*

*Candidati:*

*Buonomo Alessio 0124002066*

*Martina Raimo 012400235 Professoressa Paola Barra*

*Giuseppe Candice0123002383 Professoressa Mariacarla Staffa*

Indice

[1. Introduzione 3](#_Toc9518)

[1.1 Scopo del sistema 3](#_Toc9519)

[1.2 Obiettivi di progettazione 3](#_Toc9520)

[1.3 Definizioni, acronimi e abbreviazioni 4](#_Toc9521)

[1.4 Riferimenti 4](#_Toc9522)

[1.5 Panoramica 5](#_Toc9523)

[2. Sistema corrente 14](#_Toc9524)

[3. Sistema proposto 15](#_Toc9525)

[3.1 Panoramica 15](#_Toc9526)

[3.2 Decomposizione del sistema 15](#_Toc9527)

[3.3 Hardware/Software mapping 17](#_Toc9528)

[3.4 Gestione dei dati persistenti 18](#_Toc9529)

[3.5 Controllo accessi e sicurezza 18](#_Toc9530)

[3.6 Decisioni sul flusso di controllo globale 19](#_Toc9531)

[3.7 Condizioni limite 19](#_Toc9532)

[4. Servizi del sottosistema 20](#_Toc9533)

[5. Glossario 21](#_Toc9534)

Elenco delle figure

*Figura 1.1: Diagramma Classi Progettazione*

*Figura 3.1: Decomposizione del Sistema*

*Figura 3.2: Diagramma Hw/Sw Mapping*

*Figura 4.1: Diagramma Sottosistemi software con annessi servizi offerti*

# 1.0 Introduzione

## 1.1 Scopo del sistema

SmartLaundry è il dispositivo che raccoglie lavaggio, asciugatura, deumidificazione e riscaldamento, tutto in unico sistema, permettendo all’utente di decidere se svolgere i processi in maniera manuale, o affidandosi agli automatismi del sistema.  
Il dispositivo tramite i dati inseriti dall’utente in fase di configurazione, calibra i sensori del dispositivo a cui è associato, per far sì che esso riesca a gestire anche delle situazioni critiche, modificando il suo comportamento.

## 1.2 Obiettivi di progettazione

* **Usabilità:** Apprendimento facile ed immediato delle modalità di utilizzo della piattaforma attraverso un’interfaccia semplice ed intuitiva, con funzionalità di chiara comprensibilità per l’utente.
* **Look and feel:** L’interfaccia di SmartLaundry dovrebbe essere responsiva e rispettare tutti i canoni di progettazione moderna.
* **Response time:** SmartLaundry dovrebbe fornire un risultato, a fronte della richiesta di un utente, con tempistiche accettabili.
* **Estensibilità**: SmartLaundry dovrebbe essere tale da permettere l’aggiunta di nuove funzionalità in modo semplice, senza dover modificare l’intero sistema.
* **Piattaforma di destinazione:** Il sistema dovrà essere sviluppato con Python per essere una WebApp ed accessibile tramite browser.
* **Portabilità:** SmartLaundry deve essere un sistema multi-piattaforma.
* **Gestione dei Dati Persistenti:** Gestione di un database attraverso DBMS MySQL, minuziosamente strutturato. Gestione nel dettaglio dei dati persistenti rispecchiando la complessità del dominio del sistema.

## 1.3 Definizioni, acronimi e abbreviazioni

Per agevolare la comprensione della documentazione, nella tabella sottostante sono fornite le definizioni univoche di alcuni termini utilizzati con frequenza. Inoltre, per semplificare la lettura, è stato incluso un elenco degli acronimi più comunemente utilizzati con il loro significato.

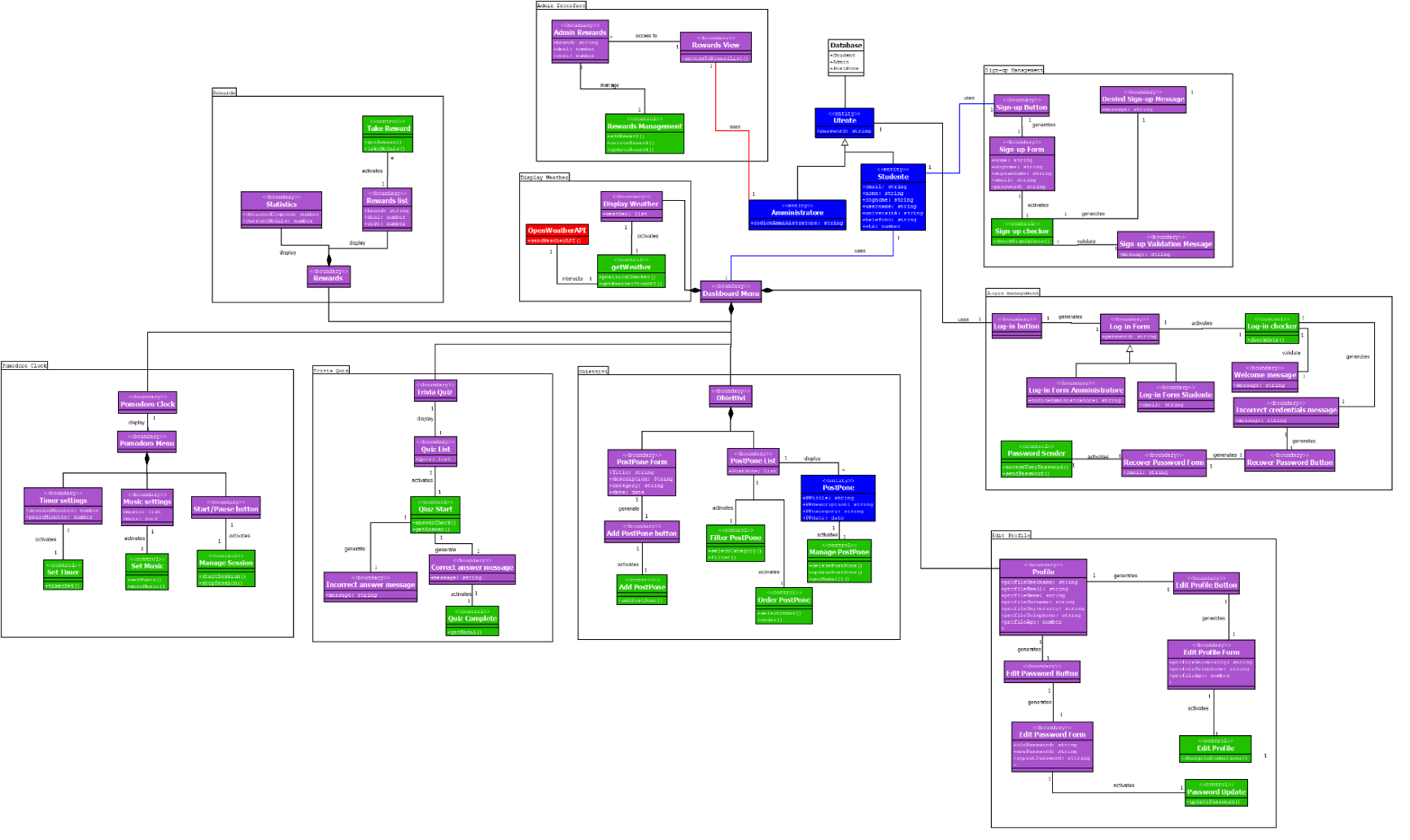
|  |  |  |
| --- | --- | --- |
|  | **Termini e Definizioni** | |
| Tecnico |  | Tipo di utente di Smart Laundry. Egli può accedere alle impostazioni del sistema. |
| Processo |  | Azione eseguita dal dispositivo tra le 4 disponibili. |
| Lavaggio |  | Sezione del sistema dove l’utente può gestire il lavaggio. |
| Asciugatura |  | Sezione del sistema dove l’utente può gestire l’asciugatura. |
| Deumidificatore |  | Sezione del sistema dove l’utente può gestire la deumidificazione. |
| Riscaldamento |  | Sezione del sistema dove l’utente può gestire il riscaldamento. |
| Smart Laundry (SL) |  | Sezione del sistema dove lo studente può riscuotere le ricompense che vuole. |
| Proprietario |  | Colui che usufruisce del sistema grazie ai permessi ricevuti. Nel sistema si prevedono utenti Studente e Amministratore. |
| Sensore |  | Sensore per il tracciamento dei dati metereologici intorno al dispositivo. |
| Modalità Automatica |  | Settaggio attraverso il quale, i processi vengono gestiti in modo autonomo |

## 1.4 Riferimenti

Per una migliore comprensione del lavoro presentato in questo documento, si prega di consultare il modello di Analisi dei Requisiti allegato, le immagini e la presentazione Power Point dell’interfaccia.

## 1.5 Panoramica

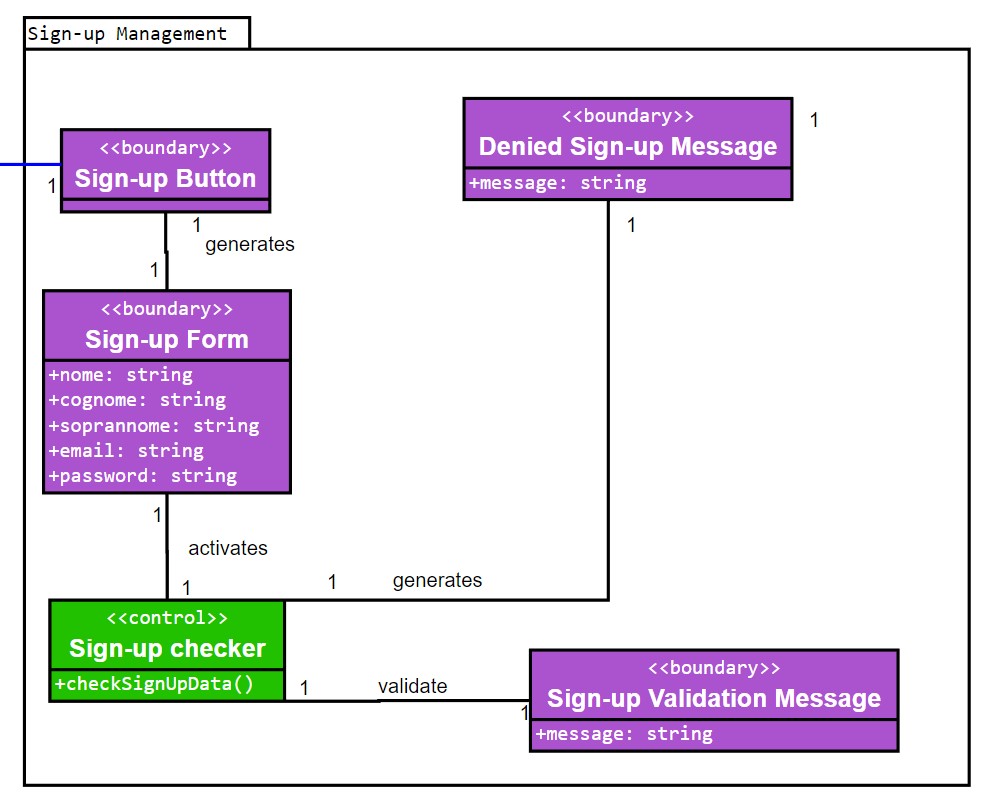
Un ulteriore raffinamento della fase di analisi ha portato alla seguente suddivisione in classi.



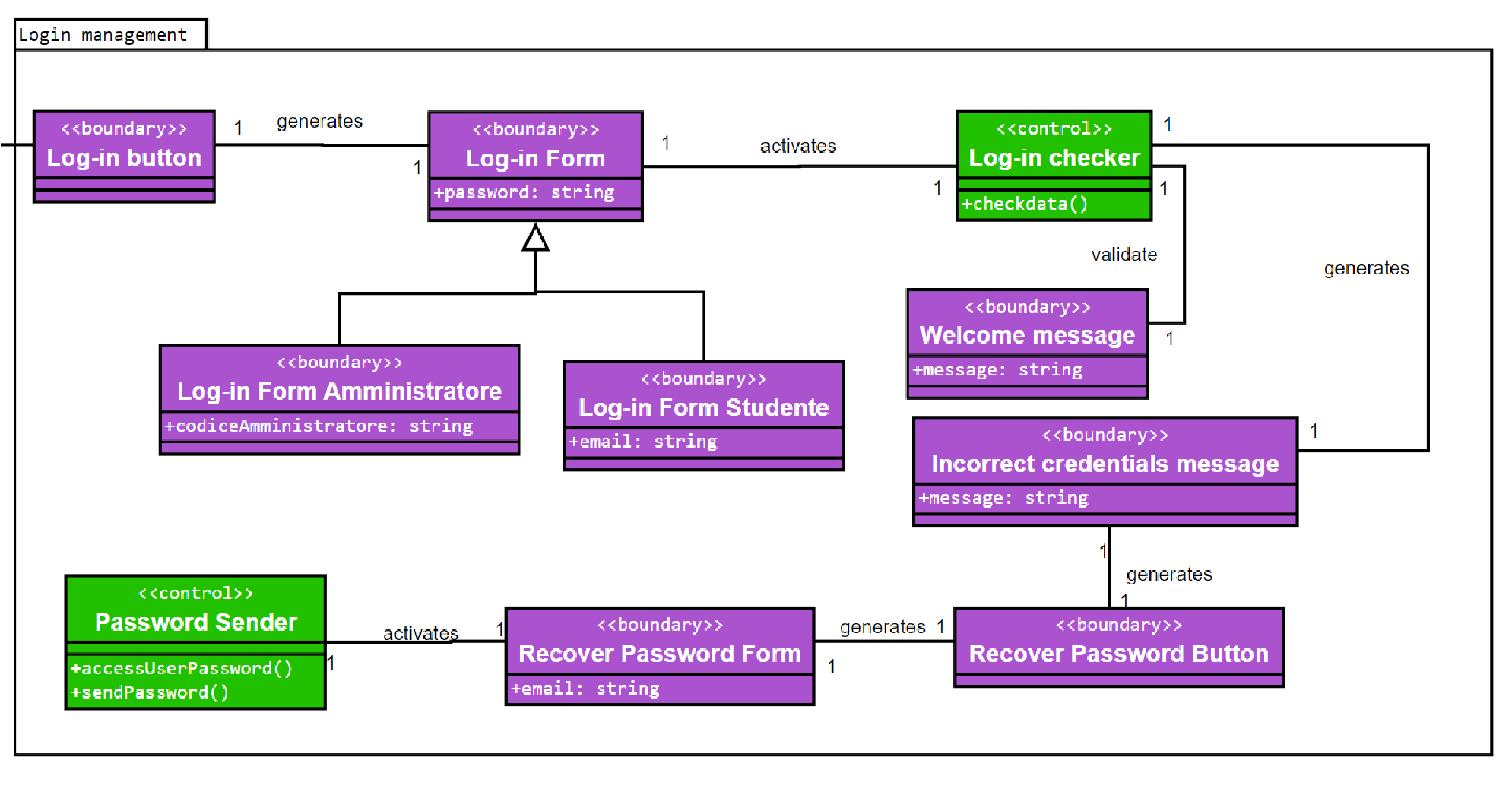
*Figura 1.1: Diagramma Classi Progettazione*

*Aprire il file “diagrammaprogettazione.svg” per una visione migliore*

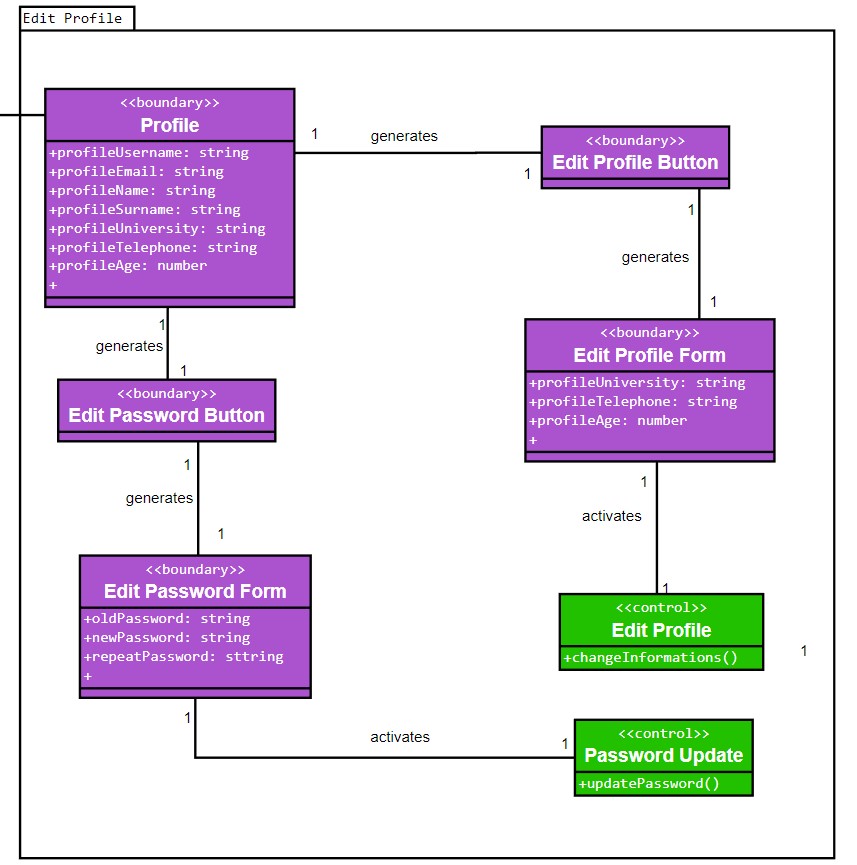
1- Sign-up Management



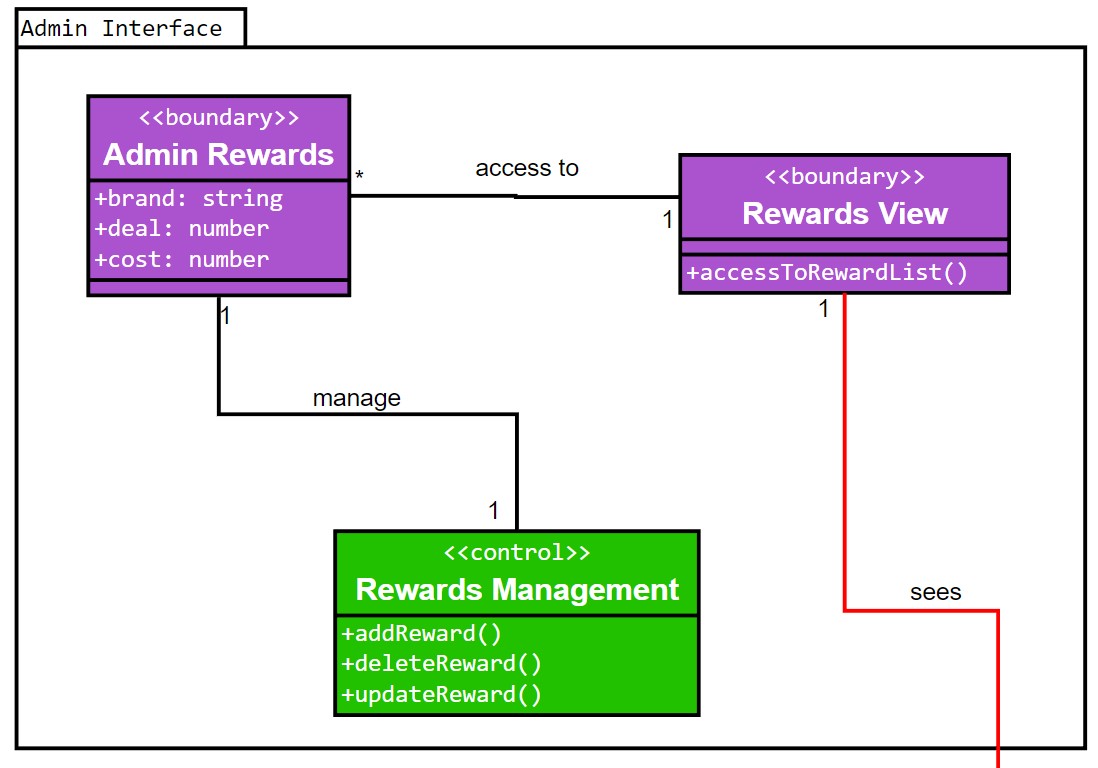
1. - Login Management



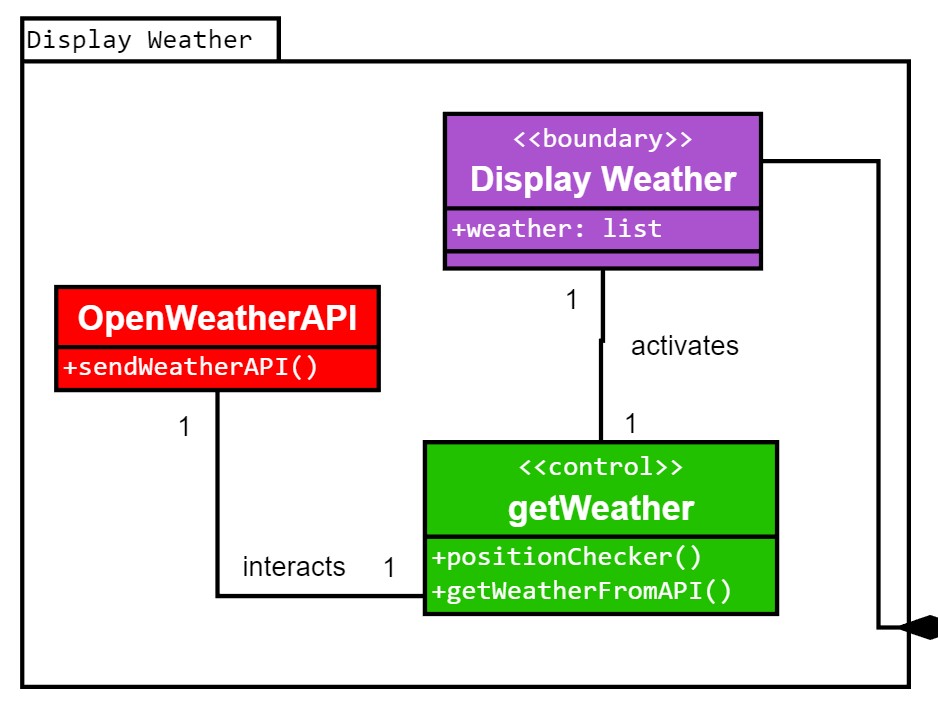
1. - Edit Profile



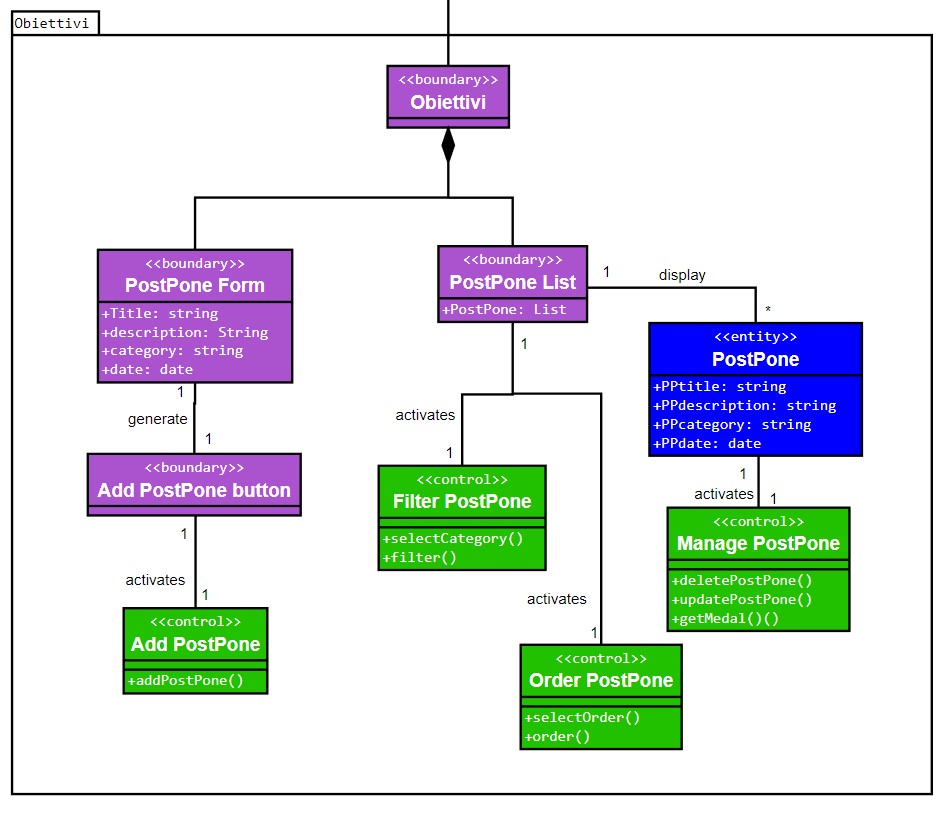
1. - Admin Interface



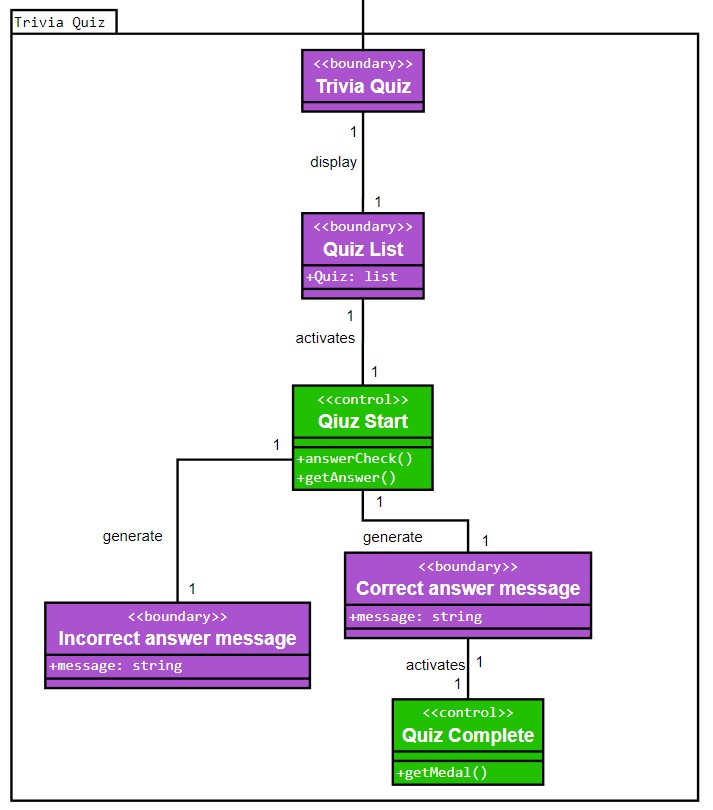
1. - Display Weather



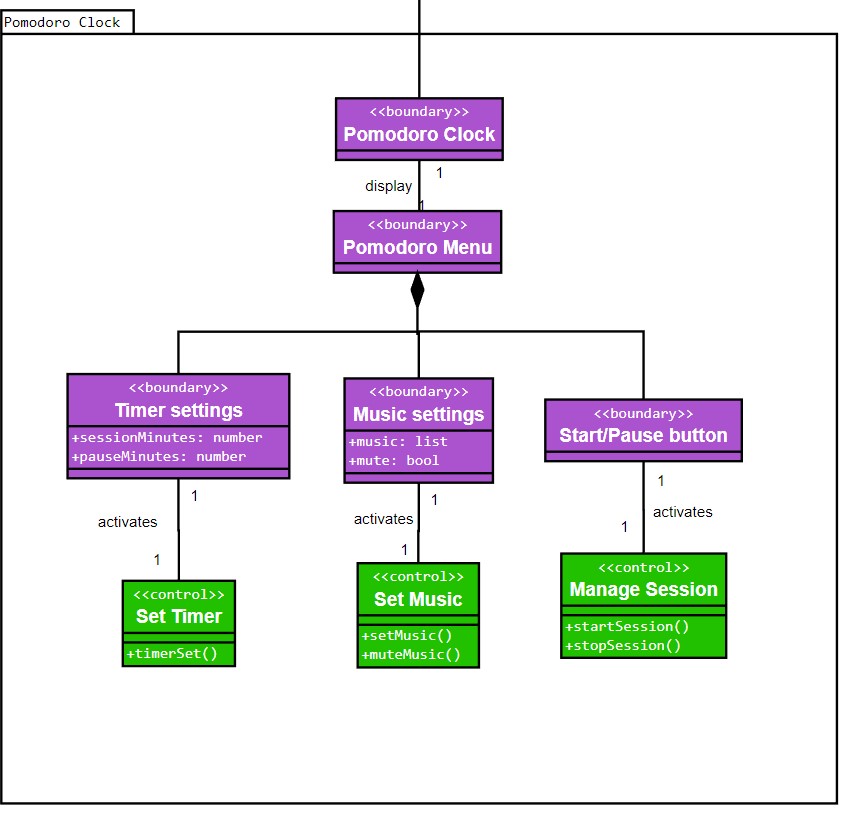
1. - Obiettivi



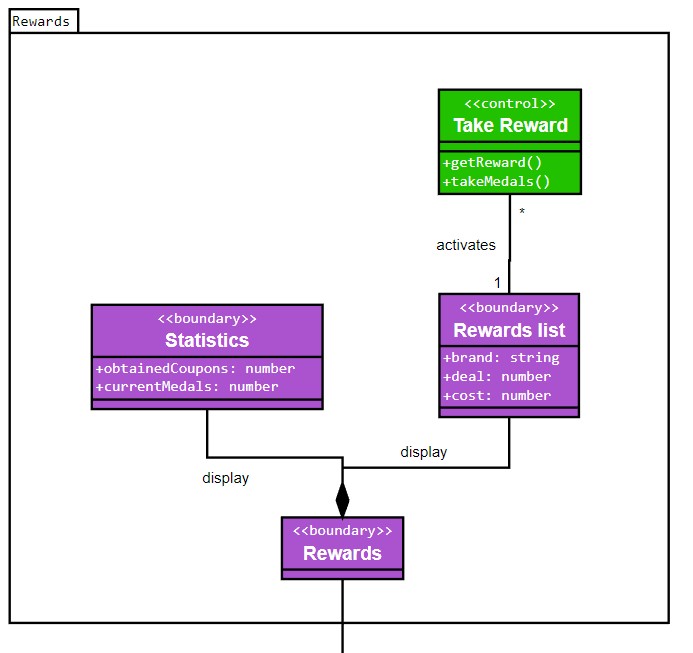
1. - Trivia Quiz



1. - Pomodoro Clock



1. - Rewards



# Sistema corrente

PostPone è un progetto greenfield, ovvero non si basa su una piattaforma già esistente e non è quindi vincolato da restrizioni imposte in precedenza. Tuttavia, per fornire riferimenti utili a chi utilizzerà la piattaforma, sono stati presi in considerazione elementi già esistenti nel campo delle piattaforme per studenti, come ad esempio "UNiDAYS", una web application che offre sconti agli studenti universitari, e "Focus To-Do", un'app per Android e iOS che permette di gestire impegni e utilizzare la tecnica del pomodoro.



# Sistema proposto

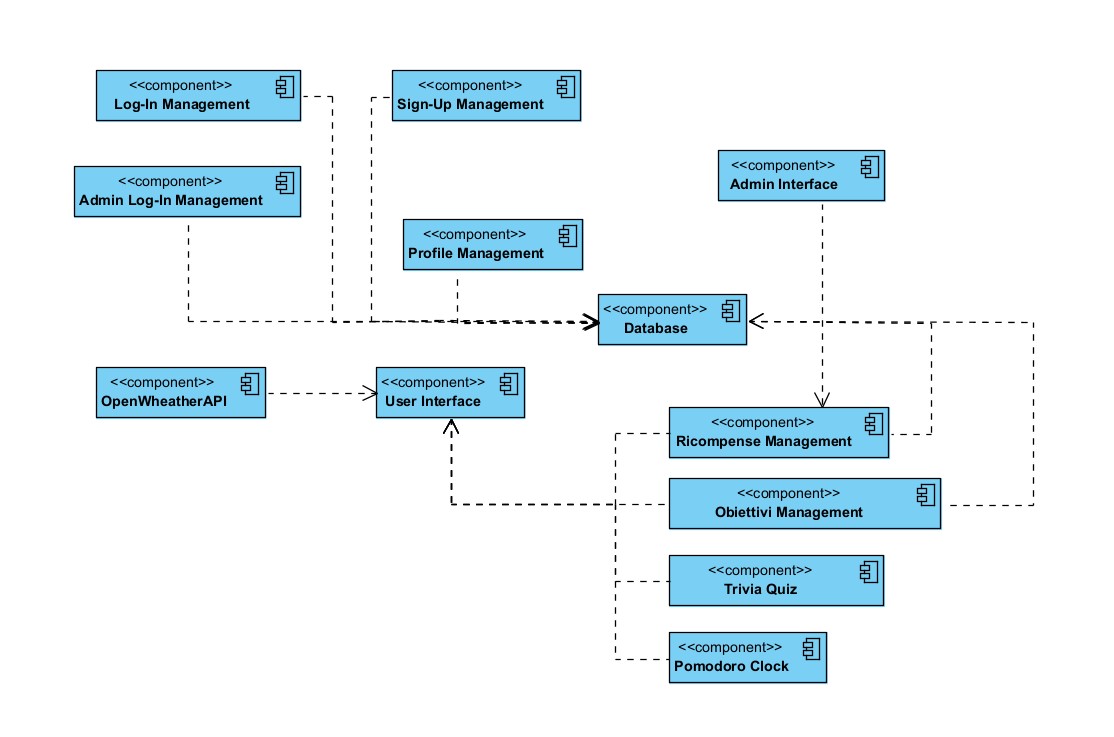
## Panoramica

Per la progettazione di questo sistema è stato scelto uno stile architetturale client/server. Esistono due tipi di client:

* L'utente, che utilizza il sistema per sfruttare le sue funzionalità;
* L'amministratore, che si connette al server per visualizzare e modificare la lista di codici sconto disponibili.

Il server offre un'API che fornisce diversi servizi ai client, in particolare l'utilizzo di OpenWeatherAPI per ottenere informazioni meteorologiche. Inoltre, gestisce l'accesso al database. Una delle ragioni che hanno portato alla scelta di una web application piuttosto che di un software stand-alone è la possibilità di effettuare l'accesso al sistema da qualsiasi dispositivo.

## Decomposizione del sistema



*Figura 3.1: Decomposizione del Sistema*

**Log-In Management:** questo sottosistema si occupa dell'interazione con l'utente durante la fase di accesso al sistema.

**Sign-Up Management:** questo sottosistema si occupa dell'interazione con l'utente durante la fase di registrazione al sistema.

**Admin Log-In Management:** questo sottosistema consente agli amministratori di accedere al sistema.

**Admin Interface:** questo sottosistema fornisce i servizi di interfaccia grafica che consentono agli amministratori di accedere alla propria area di competenza e svolgere le relative attività.

**Profile Management:** questo sottosistema si occupa di tutte le interazioni con l'utente relative alla modifica, cancellazione e visualizzazione del suo profilo.

**OpenWeatherAPI:** questo sottosistema fornisce all'interfaccia dati meteorologici relativi alla posizione dell'utente.

**User Interface:** questo sottosistema fornisce i servizi di interfaccia grafica che consentono agli utenti di accedere alla propria area di competenza.

**Ricompense Management:** questo sottosistema gestisce le interazioni con l'utente relative alla visualizzazione e ottenimento di codici sconto. Inoltre, si occupa delle interazioni con l'amministratore per quanto riguarda l'aggiunta, la visualizzazione, la modifica e la rimozione dei codici sconto disponibili.

**Obiettivi Management:** questo sottosistema gestisce le interazioni con l'utente per quanto riguarda l'aggiunta, il completamento, la visualizzazione e la rimozione dei propri PostPone.

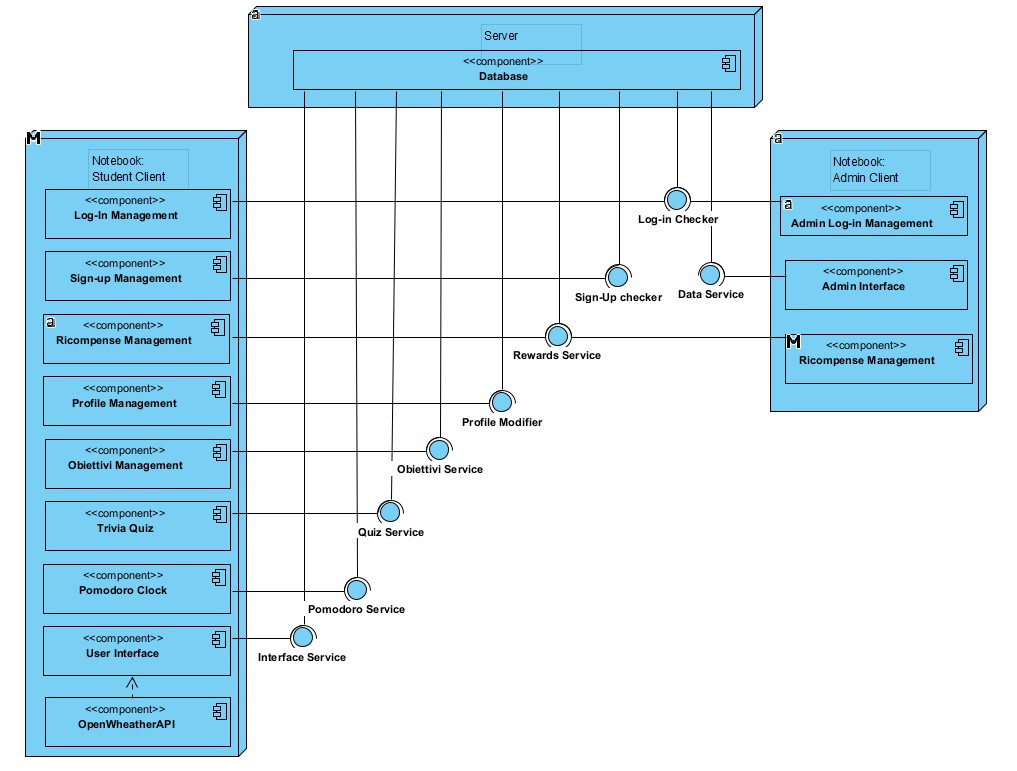
**Trivia Quiz:** questo sottosistema gestisce le interazioni con l'utente per quanto riguarda il completamento dei quiz presenti sulla piattaforma.

**Pomodoro Clock:** questo sottosistema gestisce le interazioni con l'utente relative alle sessioni di studio, durante le quali è possibile modificare la durata della sessione, la durata della pausa ed il sottofondo musicale.

**Database:** questo sottosistema DBMS (Database Management System) consente di salvare i dati del sistema in modo permanente.

## Hardware/Software mapping

Il sistema è progettato per garantire la massima disponibilità e può essere utilizzato da qualsiasi PC o altro dispositivo connesso a Internet, a condizione che l'utente si registri alla piattaforma per accedere ai servizi richiesti. Grazie alla sua accessibilità online, il sistema può essere utilizzato in qualsiasi momento e da qualunque luogo, fornendo una soluzione conveniente e flessibile per le esigenze dei suoi utenti.



*Figura 3.2: Diagramma Hw/Sw Mapping*

## Gestione dei dati persistenti

PostPone prevede la memorizzazione persistente di una grande quantità di dati mediante l'utilizzo di un database relazionale. Ciò permette di mantenere traccia di tutti i dati generati, richiesti, da elaborare e da trasmettere. Molte delle informazioni visualizzabili dagli utenti vengono ottenute eseguendo query sul database, il che comporta un alto tasso di utilizzo del database stesso. Per questo motivo, molte delle classi del modello a oggetti sono state progettate tenendo presente il mapping con le tabelle di un eventuale database.

## Controllo accessi e sicurezza

A questo punto della trattazione, è importante capire come identificare quale attore può utilizzare i servizi offerti da un determinato sottosistema. Come già accennato, ogni sottosistema può offrire diversi servizi, ma non tutti sono disponibili per tutti gli utenti. In base all'accesso di un utente specifico, è possibile stabilire quali interfacce e servizi del sottosistema o del sottosistema nel suo complesso sono accessibili o meno.

|  |  |  |
| --- | --- | --- |
| **Oggetti/Attori** | **Studente** | **Amministratore** |
| **Login management** | checkdata()  login()  accessUserPassword() sendPassword() | checkdata()  login()  accessUserPassword() sendPassword() |
| **Sign-up Management** | checkSignUpData() signUp() | — |
| **Edit Profile** | changeInformations() updatePassword() | — |
| **Admin Interface** | — | addRewards() deleteRewards() updateRewards() |
| **Display Weather** | positionChecker() getWeatherFromAPI() | — |
| **Obiettivi** | addPostPone() filterPostPone() orderPostPone() managePostPone() | — |
| **Trivia Quiz** | quizStart() quizComplete() | — |
| **Pomodoro Clock** | setTimer() setMusic() manageSession() | — |
| **rewards** | getRewards() takeMedal() | — |

## Decisioni sul flusso di controllo globale

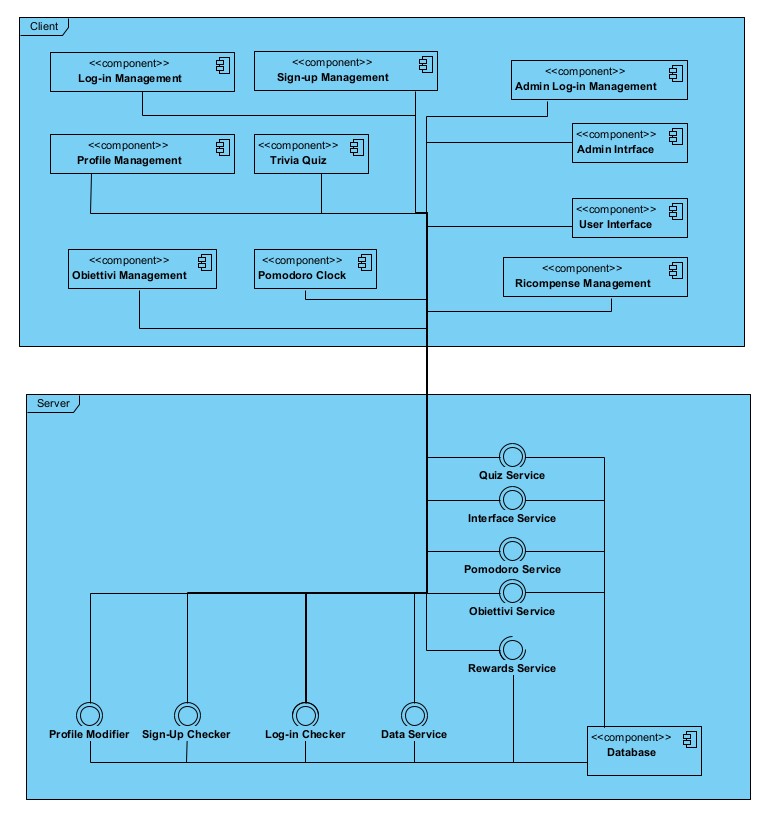
La grande quantità di richieste inviate al server rende necessario utilizzare un flusso di controllo basato sui thread per gestire l'accesso concorrente di queste richieste.E il pattern Facade identifica un controllo del thread che garantisce l'accesso concorrente di più utenti ai servizi del server, semplificando così l'interazione con il sistema complesso per l'utente finale.

## Condizioni limite

**Start e Shout down:** Il sistema PostPone è stato progettato per essere installato, avviato e, eventualmente, modificato e/o riavviato in caso di manutenzione da parte del team di sviluppo. Si prevede che una volta installato, il sistema sarà accompagnato da un'unità di backup persistente sufficientemente grande che effettuerà il backup sistematico di parte o di tutti i dati elaborati fino a quel momento.

**Initialization:** l’Admin può eseguire routine di configurazione del sistema una volta avviato.

# Servizi del sottosistema



*Figura 4.1: Diagramma Sottosistemi software con annessi servizi offerti*

# Glossario

La specifica della terminologia utilizzata è stata affrontata anche durante la stesura del documento RAD. Fare riferimento anche a quella per una visione d’insieme.

|  |  |
| --- | --- |
| **Termine** | **Descrizione** |
| **Display Weather** | Sezione della dashboard in cui è mostrato il meteo relativo alla posizione dell’utente per i prossimi 7 giorni grazie all’uso di un API |
| **Edit Password Form** | Form necessario al cambio password |
| **Edit Profile** | Procedura di cambio informazioni del profilo |
| **Edit Profile Form** | Form necessario alla modifica delle informazioni dello studente |
| **Log-in Form Amministratore** | Form necessario al login dell’amministratore nel sistema |
| **Log-in Form Studente** | Form necessario al login dello studente nel sistema |
| **OpenWeatherAPI** | API che comunica i dati relativi alle condizioni meteorologiche al sistema |
| **Password Sender** | Procedura che invia la password all’utente |
| **Password Update** | Procedura di cambio password |
| **PostPone Form** | Form di inserimento di un PostPone |
| **Quiz List** | Schermata dei quiz a disposizione dell’utente |
| **Recover Password Form** | Form necessario al recupero password |
| **Rewards List** | Schermata delle ricompense a disposizione dell’utente |
| **Rewards Management** | Insieme di procedure per il management delle ricompense affidato all’amministratore |
| **Sign-up checker** | Procedura di controllo del form di registrazione. |
| **Sign-up Form** | Form necessario alla registrazione al sistema |