

# **Design Document**

Students&Companies

Elia Falzoni - 10763955

Andrea Toffoli - 10725294

Andrea Torti - 10730470

Release Date: 09/02/2025

Version 1.1

# Contents

<b>1</b>	<b>Introduction</b>	<b>4</b>
1.1	Purpose . . . . .	4
1.2	Scope . . . . .	4
1.3	Definition, Acronyms, Abbreviations . . . . .	5
1.4	Revision History . . . . .	5
1.5	Reference Documents . . . . .	5
1.6	Document Structure . . . . .	5
<b>2</b>	<b>Architectural Design</b>	<b>7</b>
2.1	Overview . . . . .	7
2.2	Component view . . . . .	8
2.2.1	High Level Diagram . . . . .	8
2.2.2	Low Level Diagram . . . . .	9
2.3	Deployment view . . . . .	10
2.4	Runtime view . . . . .	12
2.5	Component interfaces . . . . .	29
2.6	Selected architectural styles and patterns . . . . .	30
2.7	Other design decisions . . . . .	31
2.7.1	Availability . . . . .	31
2.7.2	Scalability . . . . .	31
2.7.3	Ease of Deployment . . . . .	31
2.7.4	Data Storage . . . . .	31
<b>3</b>	<b>User Interface Design</b>	<b>32</b>
3.1	Overview . . . . .	32
3.2	UI Mockups . . . . .	33
<b>4</b>	<b>Requirements Traceability</b>	<b>40</b>
<b>5</b>	<b>Implementation, Integration and Test Plan</b>	<b>42</b>
5.1	Overview and Implementation Plan . . . . .	42
5.2	Features Identification . . . . .	42
5.3	Integration Strategy . . . . .	43
5.4	System Testing Strategy . . . . .	45

<b>6 Effort Spent</b>	<b>46</b>
<b>7 References</b>	<b>46</b>
7.1 References . . . . .	46
7.2 Used Tools . . . . .	46

# 1 Introduction

## 1.1 Purpose

The purpose of this document is to define the architectural and design decisions for the development of the Students&Companies platform. Building on the requirements outlined in the RASD, this Design Document provides a detailed blueprint for the system's implementation. This document aims to:

- Describe the high-level architecture of the system, including its main components and their interactions,
- Provide detailed design views, such as component diagrams, deployment diagrams, and runtime behavior,
- Ensure traceability between the functional requirements and the design elements. Define key design decisions, including selected patterns, technologies, and any constraints or assumptions influencing the implementation.

## 1.2 Scope

Student&Companies (S&C) is a new platform designed to streamline the internship matching process. It aims to help students find internships aligned with their skills and education while enabling companies to easily identify suitable candidates for their projects. Universities play a key role by facilitating complaint resolution for students and companies involved in internships and mediating communication between all stakeholders.

S&C will be delivered as a web application with a modern, thick-client architecture. Upon each page load, all necessary data will be fetched from the server, allowing users to interact with the application through JavaScript-based functions.

The software architecture will follow a classic, monolithic, 3-tier model:

- Web Server: Responsible for directly handling user requests and serving web pages or assets.
- Application Server: Processes business logic, handles user authentication, manages workflows, and coordinates communication with the database and other services.
- Database Management System (DBMS): Manages persistent data storage and retrieval, including user data, internship details, and university information.

Additionally, a mail server will handle sending and receiving emails between S&C users, such as notifications, confirmations, and communication between students, companies, and universities.

To enhance the platform's value, S&C will leverage AI-powered recommendation system. This system will analyze data collected from students, companies, and universities to provide personalized suggestions. For students, this includes tips on improving their CVs and identifying the best-fit internships. For companies, it involves guidance on promoting their internship opportunities effectively. The AI processing will be performed on rented infrastructure, such as an AWS GPU cluster, ensuring the computational power needed for efficient, real-time data analysis and machine learning tasks.

Renting AI infrastructure was chosen for its flexibility and cost-effectiveness. This approach allows S&C to scale resources as demand fluctuates, avoiding the upfront investment and maintenance costs of owning specialized hardware. It also ensures access to cutting-edge technology, which is vital for delivering accurate and timely recommendations, while allowing the development team to focus on improving the platform rather than managing infrastructure.

### 1.3 Definition, Acronyms, Abbreviations

- S&C: Students&Companies
- DD: Design Document
- RASD: Requirements Analysis and Specification Document
- UI: User Interface

### 1.4 Revision History

- Version 1.0 (07/01/2025)
- Version 1.1 (09/02/2025): fixed architecture-related paragraphs to ensure consistency.

### 1.5 Reference Documents

- Specification Document assignment

### 1.6 Document Structure

The Design Document is composed of 7 sections, detailed below:

**Introduction** The first section of the chapter highlights the importance of the Design Document, providing comprehensive definitions and detailed explanations of the architecture that hold the system's functioning, as well as the acronyms and abbreviations that are employed in the description of it. It also revisits the scope of the Students&Companies system.

**Architectural Design** The second section outlines the main components of the system and their relationships. It places a strong emphasis on design decisions, architectural styles, patterns, and paradigms.

**User Interface Design** This third section describes the system's user interface, presenting mockups and detailed explanations of the main pages.

**Requirements Traceability** The fourth section outlines the system's requirements, illustrating how they are fulfilled by the design decisions.

**Implementation, Integration, and Test Plan** The fifth section provides an overview of the implementation of the system's various components. It also describes their integration process and presents a comprehensive plan for testing them.

**Effort Spent** This section includes details about the number of hours each group member dedicated to the creation of this document.

**References** The final section lists all the documents referenced during the preparation of this Design Document.

## 2 Architectural Design

### 2.1 Overview

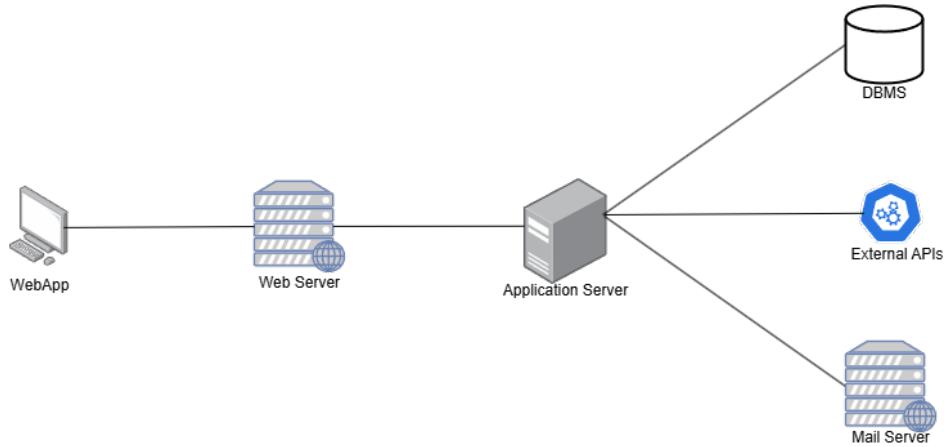


Figure 1: S&C architectural design overview

#### Client side:

- **WebApp:** the client-side of the Students&Companies platform is implemented as a web application, accessible through modern web browsers on desktop and mobile devices. The web app provides an interface for accessing various parts of the application.

#### Server side:

The server-side of the Students&Companies platform acts as the backbone of the system, handling business logic, data management, and communication between the client-side and the database. It processes user requests, retrieves or updates data, and ensures secure and efficient interactions.

- **Web Server:** the web server's primary role is to handle requests from the client-side and deliver the necessary files to run the front-end application.
- **Application Server:** the application server is where the real business logic and application-specific processing happens. This server handles complex tasks and serves as the middle layer between the client-side and the database.
- **DBMS:** it stores all the personal information about the users who are subscribed to S&C as well as the recommendation data and interviews' and complaints' related chats.
- **Mail Server:** it sends emails related to login OTP, password reset and recommendation-feedback questionnaires.
- **External APIs:** they handle requests to external services, such as the AI recommendations, CVs tips and internship's description enhancing ones.

## 2.2 Component view

### 2.2.1 High Level Diagram

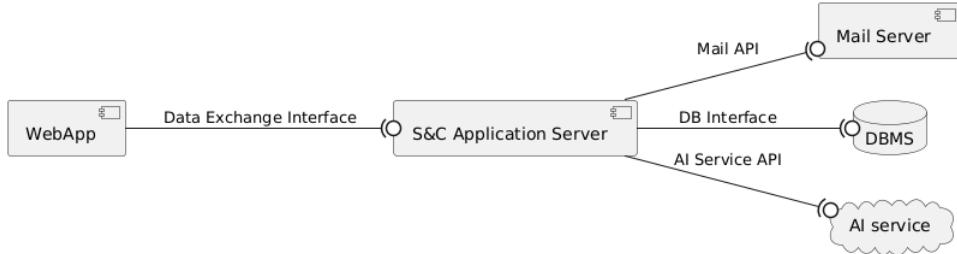


Figure 2: High-level S&C architectural design overview

In the figure above, a high-level component diagram of S&C is presented, illustrating the external components and their communication with the server, specifically:

- **WebApp:** S&C access point. It fetches data from server through the Data Exchange Interface in order to provide S&C to all users.
- **S&C Application Server:** it interacts with the WebApp through the Data Exchange Interface and with services external from S&C in order to provide various features.
- **Mail Server:** responsible for sending emails about recommendation feedback, login OTPs and password-reset links.
- **DBMS:** stores users' personal information, recommendations data and chats related to interviews and complaints. It's connected to the Model Manager.
- **AI Service:** external service in order to provide recommendations and tips for CVs' and internships' description enhancing.

### 2.2.2 Low Level Diagram

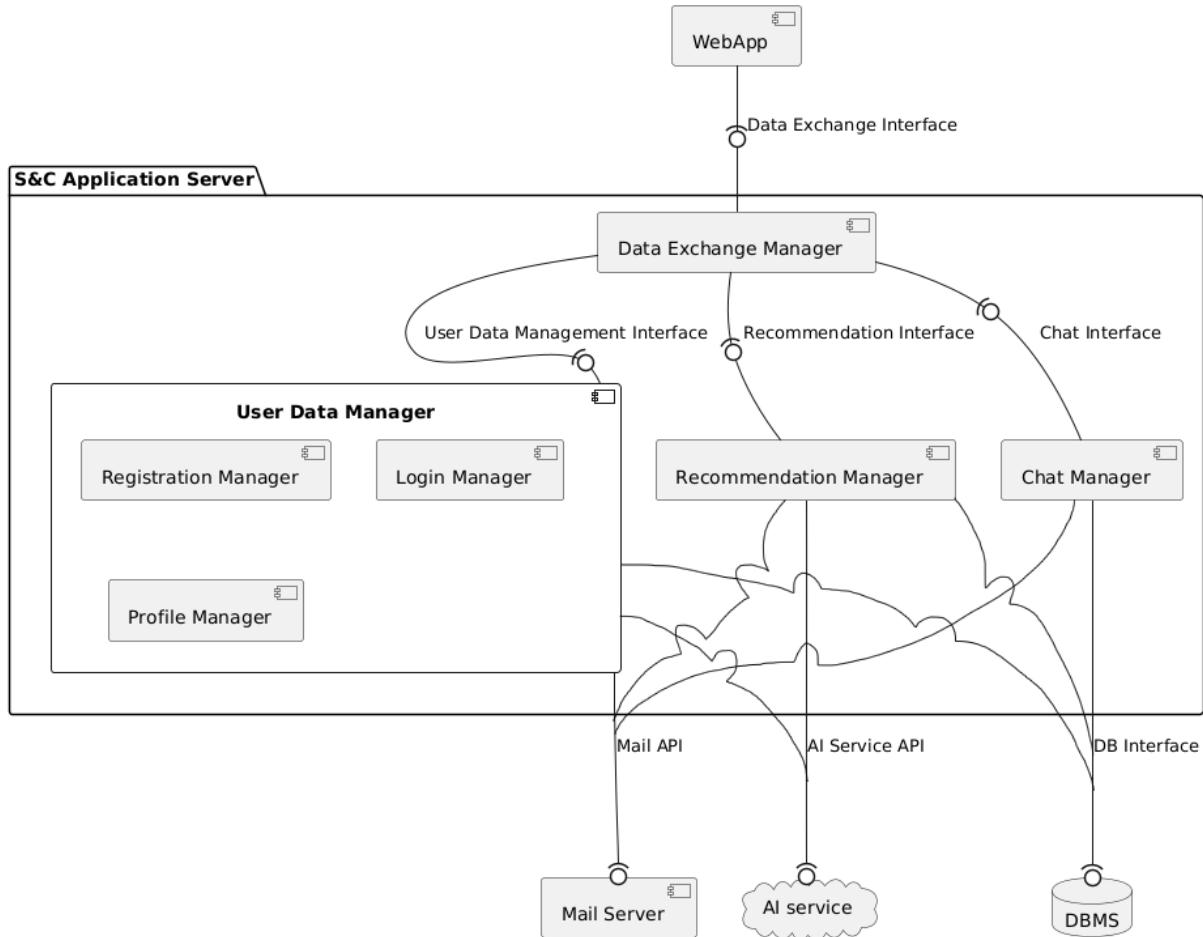


Figure 3: Low-level S&C component view overview

In the figure above, a low-level component diagram of S&C is presented, illustrating the internal and external components and their interconnections, specifically:

- **Data exchange Manager:** pivotal component which manages the communication between users and S&C server. It handles the data exchange between the parties, ensuring the users' devices receive necessary information for working as thick clients.
- **Recommendation Manager:** it manages the creation and management of recommendations through the interaction with the externalized AI service, by sending to the latter the data fetched from the DBMS. Also, when requested, it handles the recommendations to the users through the Data Exchange Manager. It interacts with the mail server when it needs to send the feedback questionnaires to the users.
- **User Data Manager:** it handles the sign-up, log-in and password reset processes. Moreover, it enables users to manage their profiles. It interacts with the DBMS when users want to retrieve or modify their data. It also interacts with the mail server for email verification and with the AI service for personalized AI suggestions regarding CVs and Internship offers. It will handle all these functions separately through its

three contained components: the Registration manager, the Login manager and the Profile manager.

- **Chat Manager:** it manages the communications between all the actors of S&C, chats and video calls, interacting with the DBMS and the Data exchange manager directly. It also interacts with the mail server in case it needs to send the questionnaires to the students.

## 2.3 Deployment view

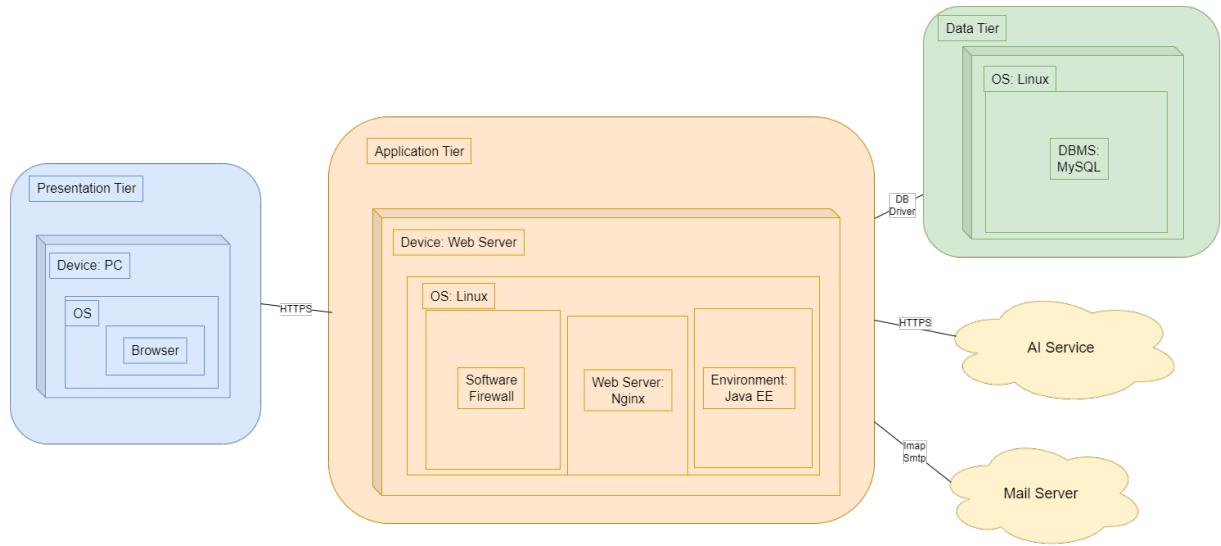


Figure 4: Deployment Diagram

- **Presentation Tier:**

- Device: End-users access the system through a personal computer (PC).
- OS: Runs on the user's PC to support web browsing.
- Browser: Used to interact with the system via HTTPS.

- **Application Tier:**

- Device: Web Server powered by Linux OS.
- Software Firewall: Secures incoming and outgoing traffic.
- Web Server: Nginx manages HTTP/HTTPS requests and routes them to the application.
- Environment: Java EE hosts the business logic and interacts with the Data Tier and external services.

- **Data Tier:**

- Device: Database Server powered by Linux OS.
- DBMS: MySQL is used for data storage and management.

- Communication: Database drivers connect the Application Tier to the Data Tier.

- **External Services:**

- AI Service: Integrated via HTTPS for advanced functionalities.
- Mail Server: Handles email communication via IMAP/SMTP protocols.

- **Communication Protocols:**

- HTTPS: Ensures secure communication between tiers and external services.
- Database Driver: Facilitates data retrieval and manipulation.
- IMAP/SMTP: Used for email interactions between the Application Tier and Mail Server.

- **Key Features:**

- Modular and tiered architecture ensures separation of concerns.
- Secure communication through firewalls and encrypted protocols.
- External scalable components for handling high user traffic

## 2.4 Runtime view

The Data Exchange Manager component will never be modeled in the following sequence diagrams since all requests coming from/going to the WebApp will pass through it.

This sequence diagram represents the Student or Company Operator Login process. Once the user gets on the S&C landing page, he clicks on the "Login" button, then, inserts his credentials and clicks on the new "Login" button. Credentials validation is performed by the User Data Manager, which interacts with the DBMS. If the inserted credentials are valid, the user is redirected to S&C homepage, if not, an error message is shown.

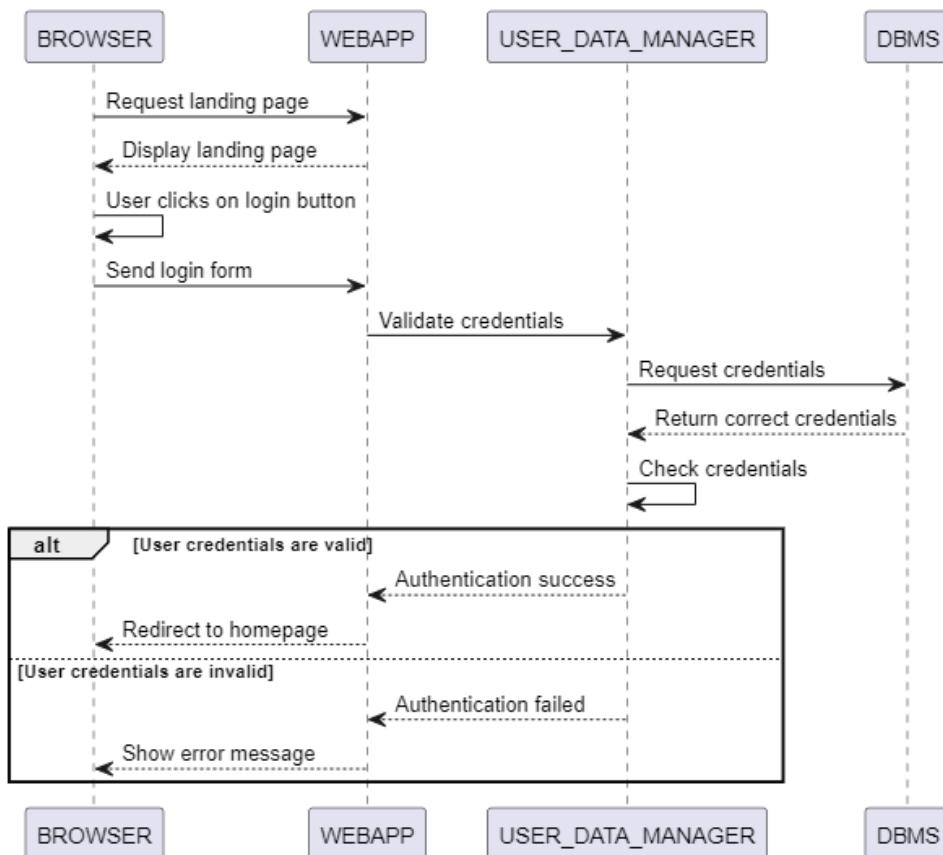


Figure 5: Student or Company Operator Login

This sequence diagram delineates the procedure for student or company registration. Upon accessing the S&C landing page, the user initiates the process by selecting the 'Sign Up' button, subsequently completing the appropriate form based on their identity and selecting 'Register'. Form validation is executed by the User Data Manager, which interfaces directly with the DBMS to verify the entered data. Upon successful verification, a confirmation email is dispatched to the user, prompting them to review it. The hyperlink contained therein directs the user to the confirmation S&C page, enabling the system to incorporate the user into the database. Consequently, the user is redirected to the login page.

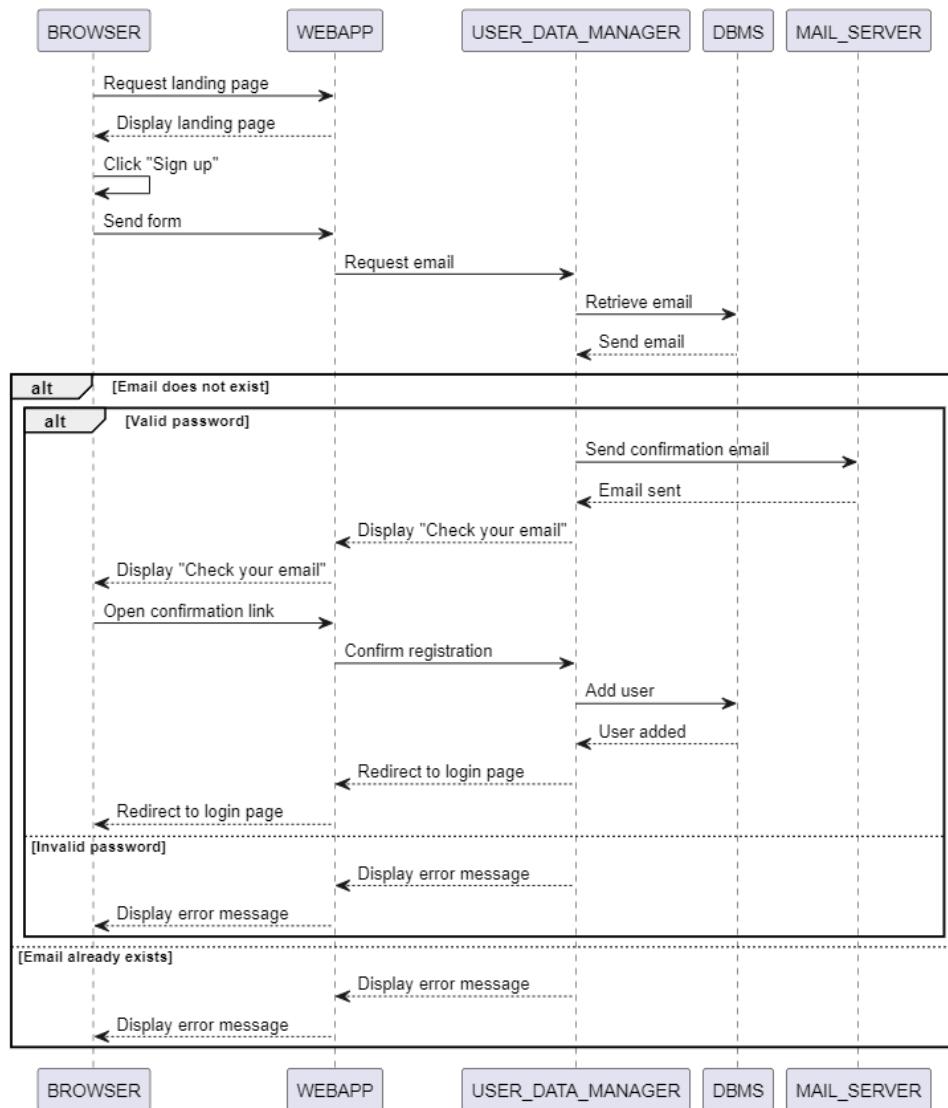


Figure 6: Student or Company sign-up

This sequence diagram illustrates the process of managing recommendations for a company operator. The user in question can see the list of system recommended candidates in his homepage (once fetched from the DBMS by the Recommendation Manager), and for each one, he may click on the student card to view more details about him (thanks to the interaction between the User Data Manager and the database), then, accept or decline the recommendation, or click "back" to close the details popup.

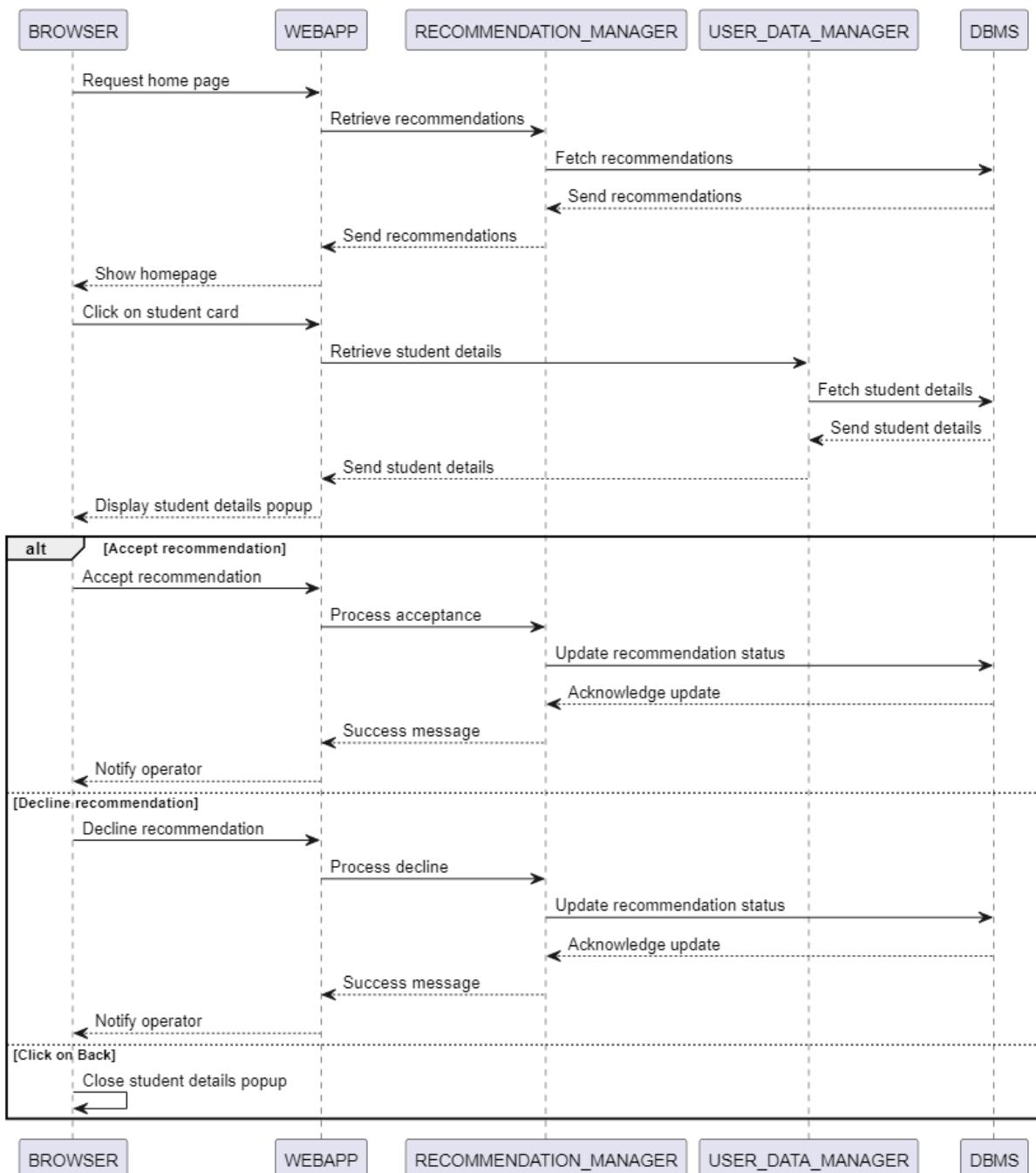


Figure 7: Company accepts/declines system's recommendations

This sequence diagram represents the process of a student updating their CV. Initially, the student requests their profile page via the browser, and the web application responds by displaying it. The student then uploads their CV file, which is sent to the web application for processing. S&C validates the file through the User Data Manager. If the file is empty, the system triggers an error response, and the browser displays an error message to the student. Otherwise, the CV data is saved in the database, and a success confirmation is sent back, allowing the browser to display a success message.

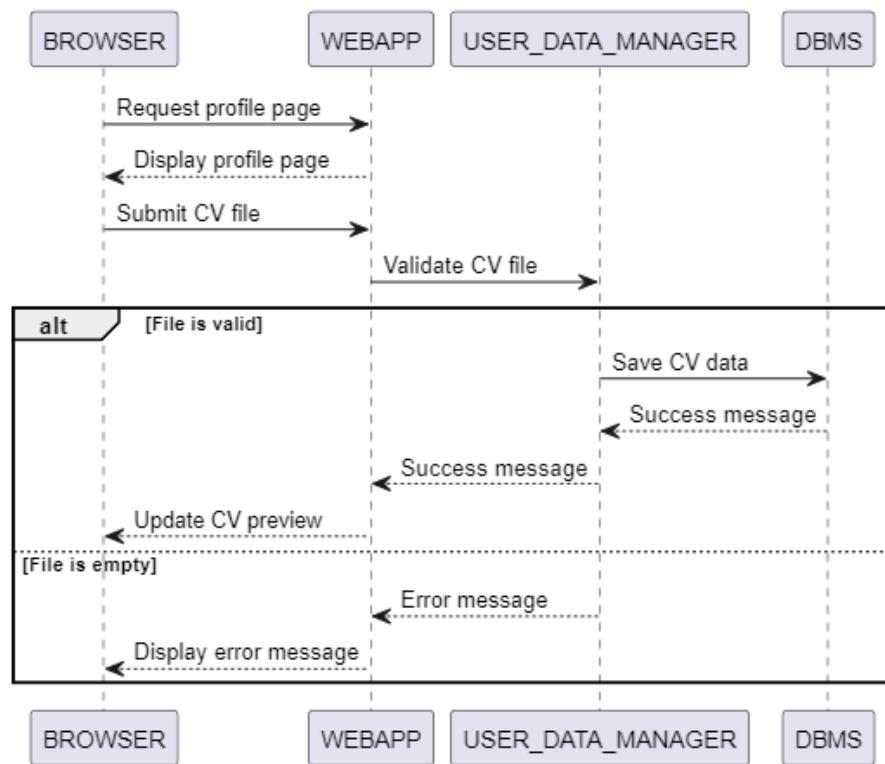


Figure 8: Student updates CV

This sequence diagram illustrates how a student searches for internships in the S&C system. The student requests the "Search for Internships" page, applies filters, and views a refined list of internships retrieved from the database. Upon selecting an internship, detailed information is displayed. All these actions are performed by the User Data Manager interacting with the DBMS. If the student clicks "like," their preference is saved in the database, and the WebApp confirms the action. Returning to the previous list is possible by clicking the "back" button.

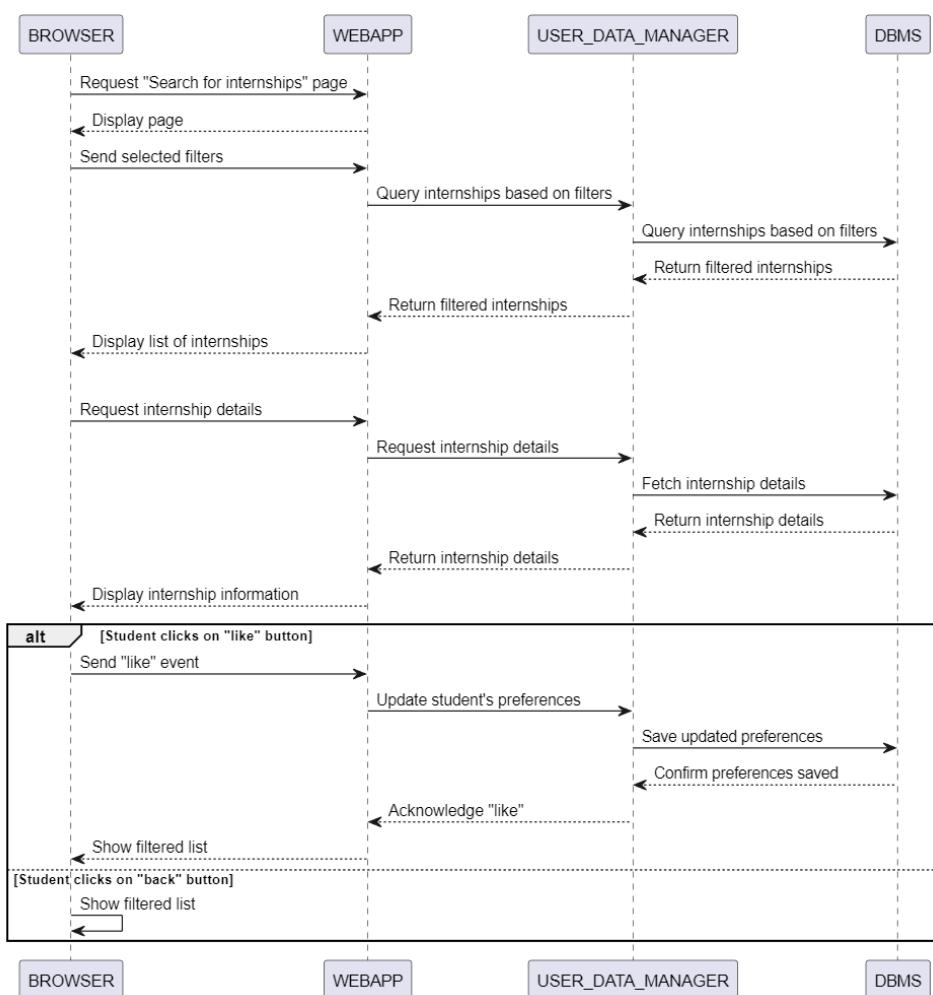


Figure 9: Student Proactively Looks For Internships

This sequence diagram represents the process of a student or company operator viewing analytics data. The interaction begins when the user requests their profile page via the browser, and the WebApp responds by displaying the profile page. The user clicks the "Analytics" button, which triggers a request from the WebApp to the Recommendation Manager for analytics data. The Recommendation Manager fetches the required analytics from the database and returns the data back to the WebApp. Finally, the WebApp sends the analytics data to the browser, where it is displayed for the user.

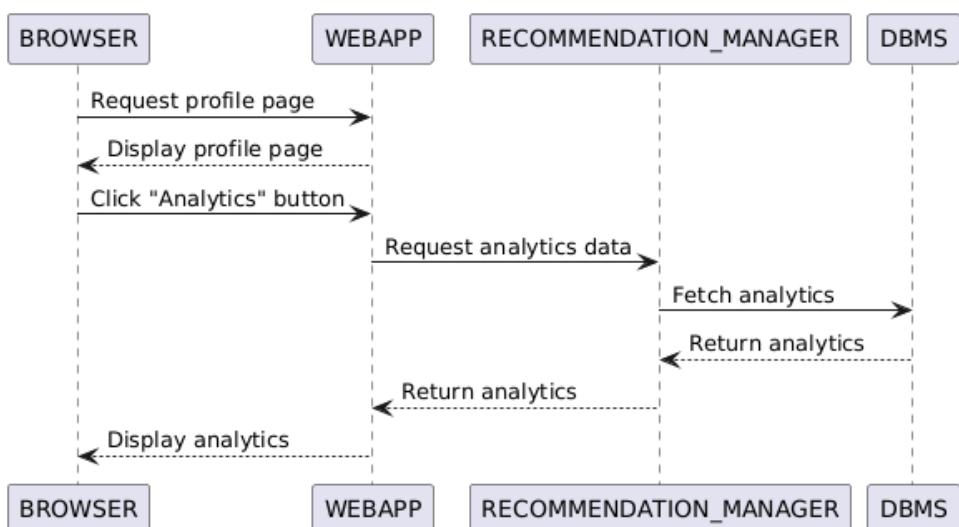


Figure 10: Student or Company looks at analytics

This sequence diagram illustrates the process of submitting a complaint about an internship. The user begins by accessing the home page and navigating to the "Manage Internships" section, where he views a list of internships, fetched from the DBMS by the User Data Manager. After selecting an internship, the user fills out a complaint form. If they choose to end the internship, an additional text field is required. The WebApp validates the form, displaying an error message if mandatory fields are missing. If the form is complete, the complaint is forwarded to the User Data Manager and saved in the database, and a confirmation message is shown to the user.

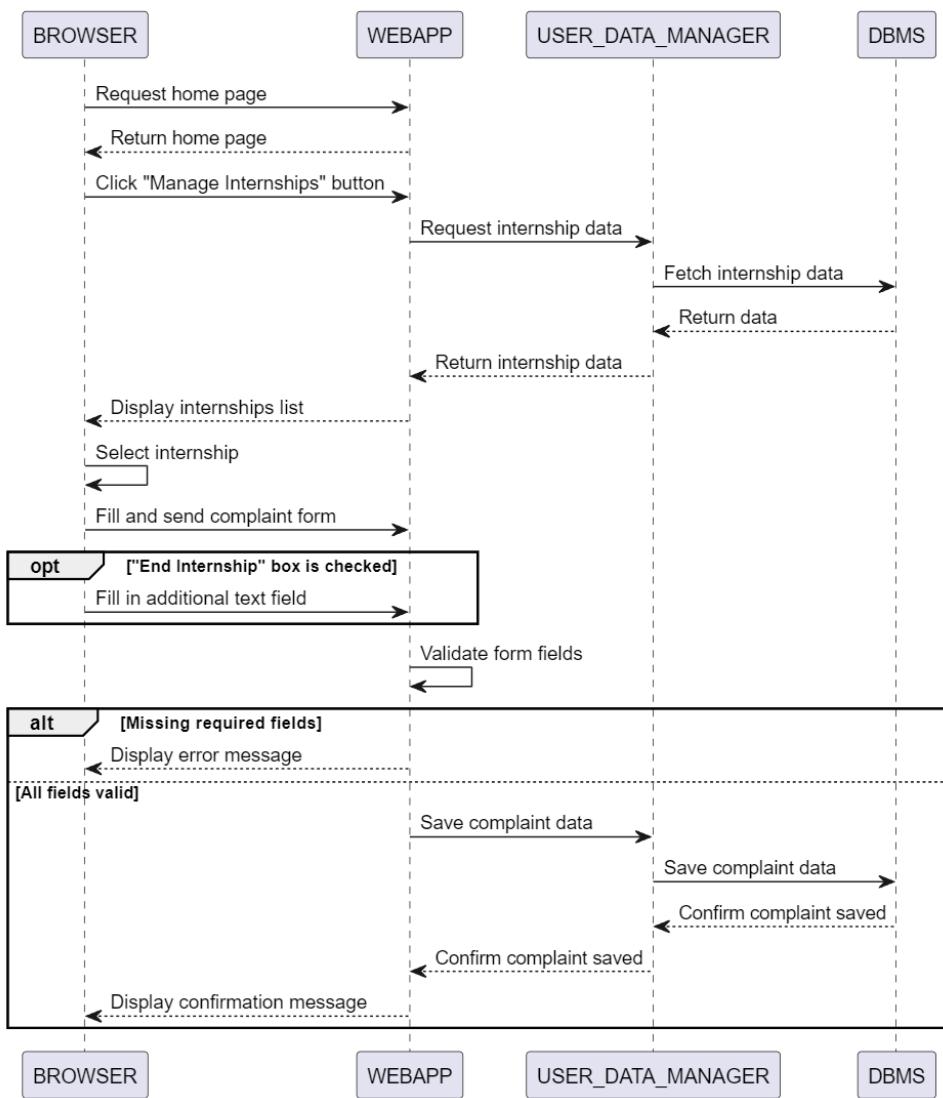


Figure 11: Student or Company files a complaint

This sequence diagram illustrates the process of handling a questionnaire for a student or company operator. The Recommendation Manager initiates the process by prompting the Mail Server to send an email containing the questionnaire link. When the recipient accesses the link, the browser requests the questionnaire page, which the WebApp provides. After filling out the questionnaire, the user submits it through the browser. If required fields are incomplete, the WebApp displays an error message. Otherwise, the WebApp forwards the completed questionnaire to the Recommendation Manager, which stores it in the database. Once stored, a confirmation is sent back to the WebApp, which then displays a confirmation message to the user.

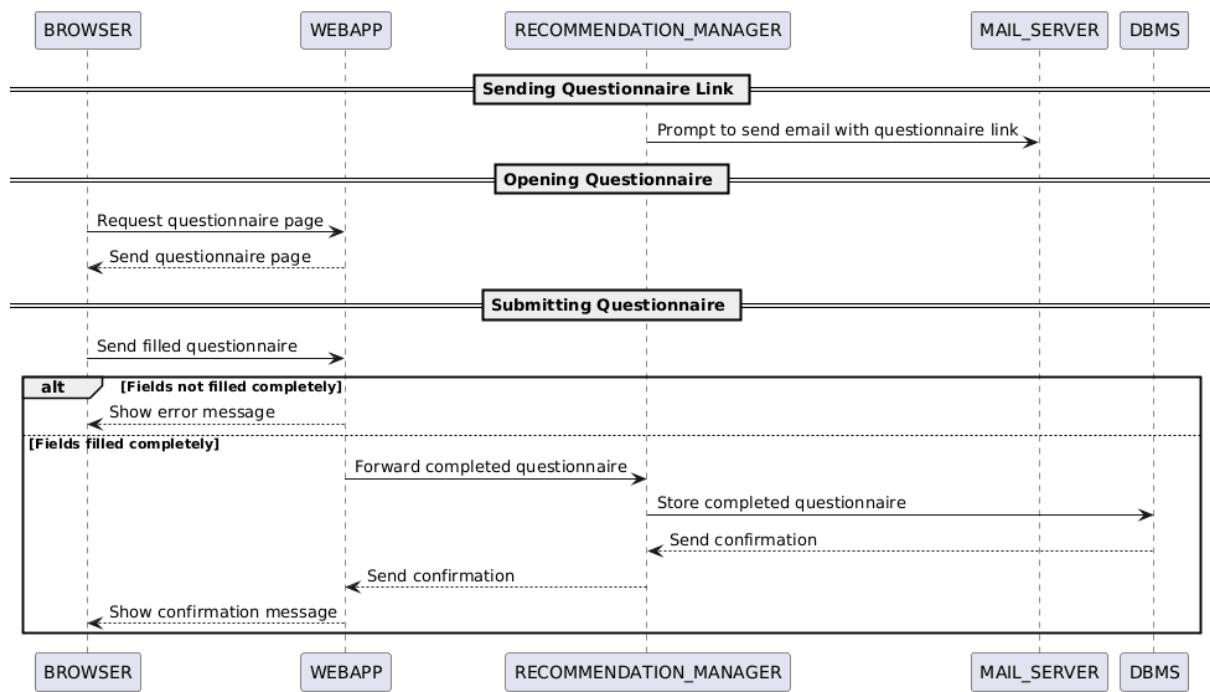


Figure 12: Student or Company Operator fills feedback questionnaire

This diagram represents the login process of an authorized university operator: once accessed the landing page and, then, the login page, the user inserts his email and, if the provided email address exists in the S&C system, an email containing the OTP is sent to the user (if not, an error message is shown). These actions are performed thanks to the interaction of the User Data Manager with the DBMS (email existence checking) and the Mail Server (email sending). The university operator inserts the received OTP: if it's valid, a "success" message is shown and the user it's redirected to the homepage, otherwise, an error message is displayed. The OTP correctness check is performed by the User Data Manager.

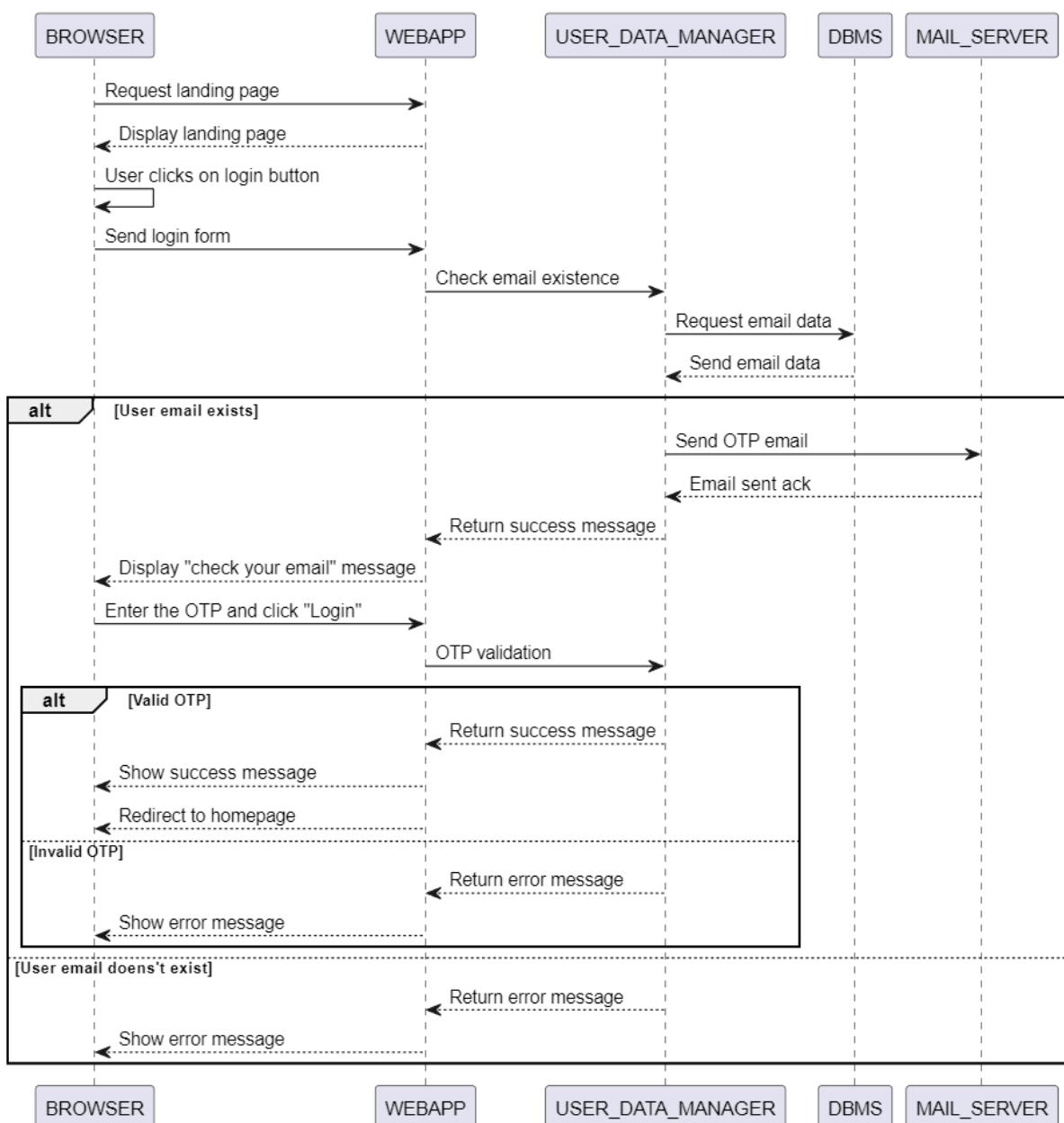


Figure 13: University Operator Login

This diagram represents the chat feature. Once the user has accessed the chat list, he is able to select an existing chat or starting an new one by the dedicated button, which permits to search for a contact (which if it doesn't exist shows an error message). Once all necessary data has been fetch from the DBMS by the Chat Manager, the user is able to type in a message and click "Send": if the message is empty, an error message is displayed. Also, S&C gives the opportunity to start a video call with the dedicated button and a dedicated request is made to the Chat Manager.

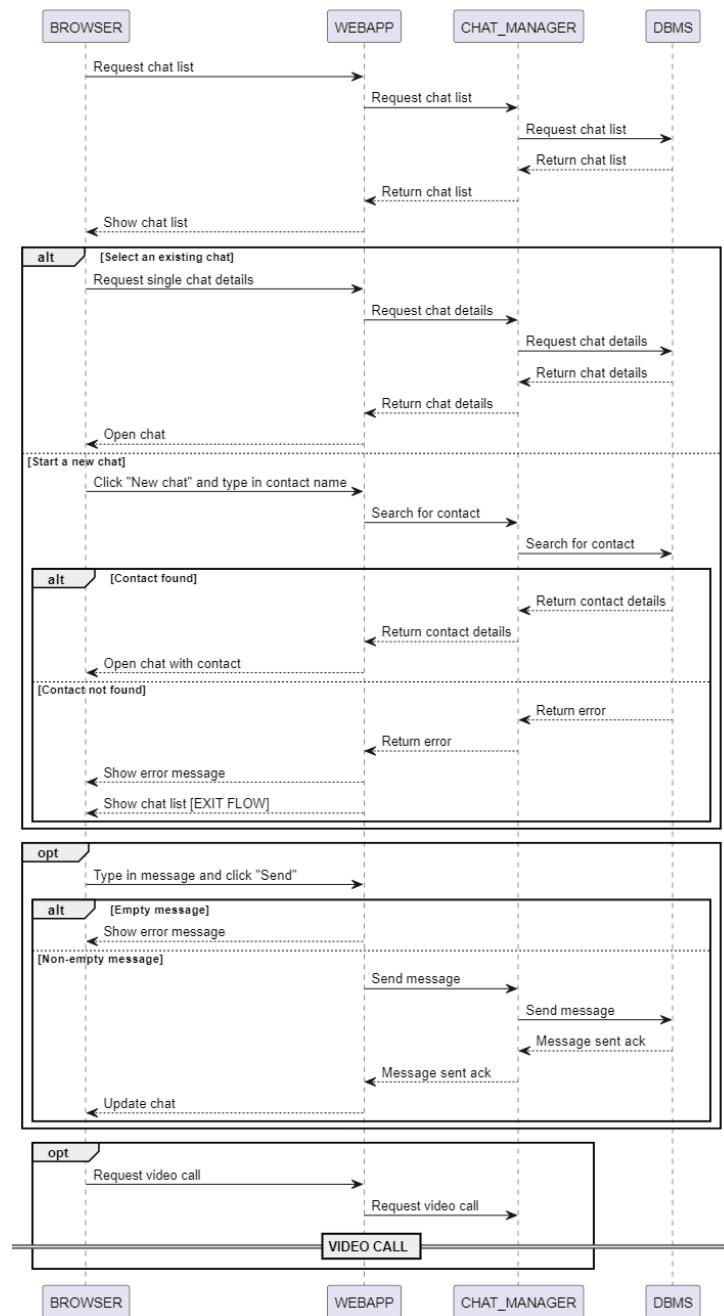


Figure 14: Student or Company or University Operator chats

This diagram represents the account's password reset process, initiated by a student or a company operator. Once the user has accessed the landing page, clicked "Login" and inserted his account's email address, he clicks on the "Reset password" button. If the inserted email isn't associated to an S&C account, an error message is shown (the check is performed by the User Data Manager, interacting with the DBMS). So, a reset-password link is sent to the account's associated email by the external Mail Server, prompted by the User Data Manager. Then, when the user clicks on the link, the password-reset page is displayed and the user is able to insert the new chosen password: if it meets the minimum password requirements (which is checked by the User Data Manager), the process ends successfully by the User Data Manager updating the information on the DBMS, otherwise, an error message is shown.

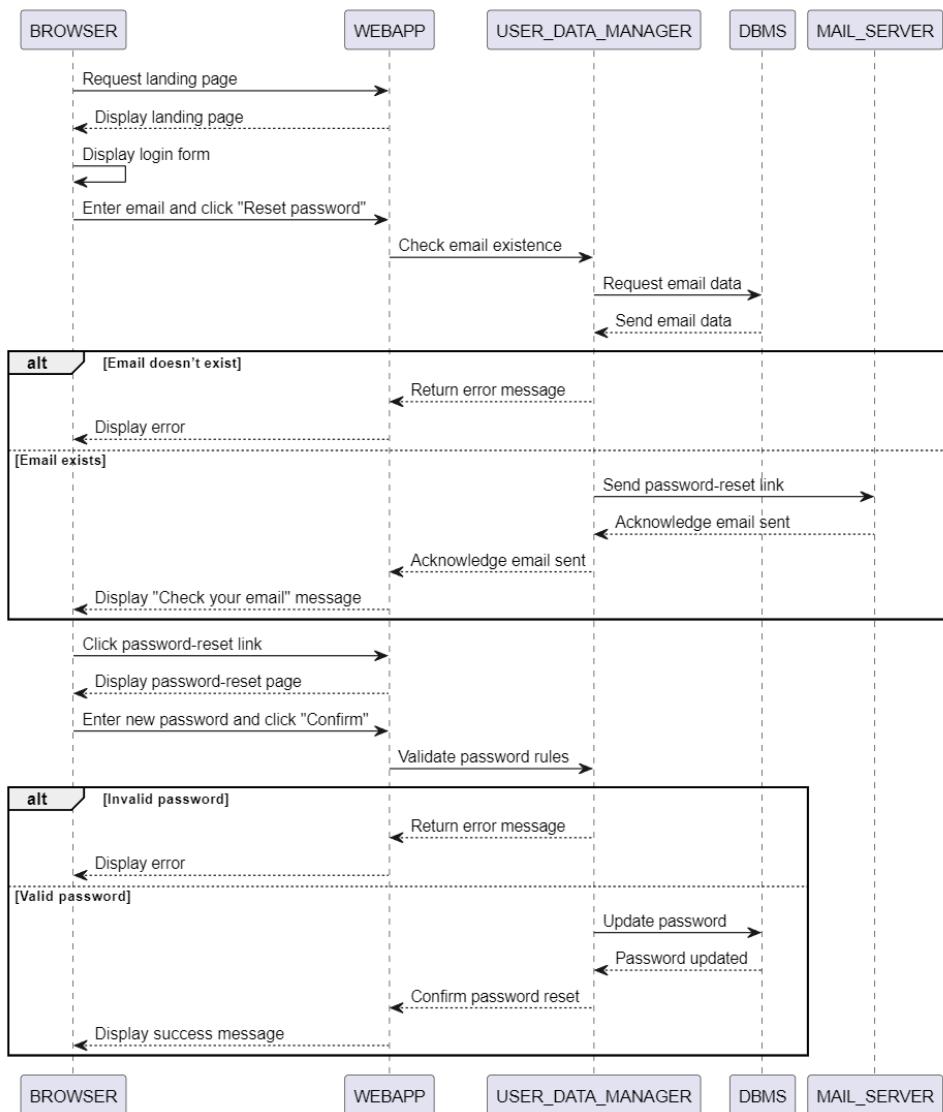


Figure 15: Student or company resets account's password

This diagram represents the company's candidates interview process. Once the company operator has accessed the single chat with the student to interview (all data is fetched from the DBMS by the Chat Manager), he is able to initiate a questionnaire creation through the dedicated button, which data is provided by chat Chat Manager. Once the user has finished, clicking "Confirm", the form data is forwarded to the DBMS by the Chat Manager, which also sends an email to the candidate notifying the form creation thanks to the external Mail Server. The student, accessing his matches page and the single match details, opens the questionnaire and fills it. Then, clicking "Confirm", he sends all data to the Chat Manager and, finally, to the DBMS. Also, S&C gives the opportunity to start a video call with the dedicated button and a dedicated request is made to the Chat Manager.

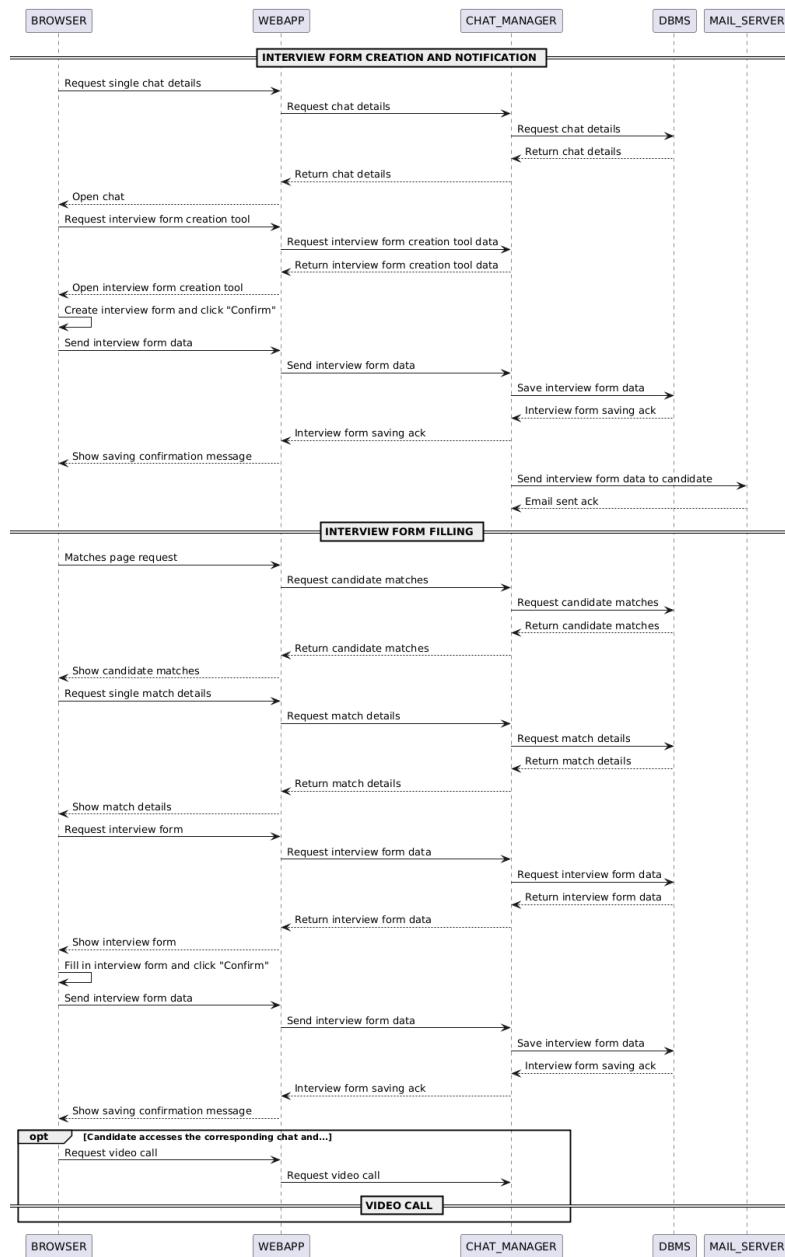


Figure 16: Company interviews a candidate

This diagram represents the process of accepting or refusing a system's recommendation by a student. The user, once accessed the home page, can view the recommendations (retrieved from the DBMS by the Recommendation Manager). More details can be viewed clicking on a single one and loading data requested from the User Data Manager from the database. If the student clicks on "Like" or "Dislike", the recommendation status is updated in the DBMS by the Recommendation Manager. If the user clicks "Back", the browser reloads the home page.

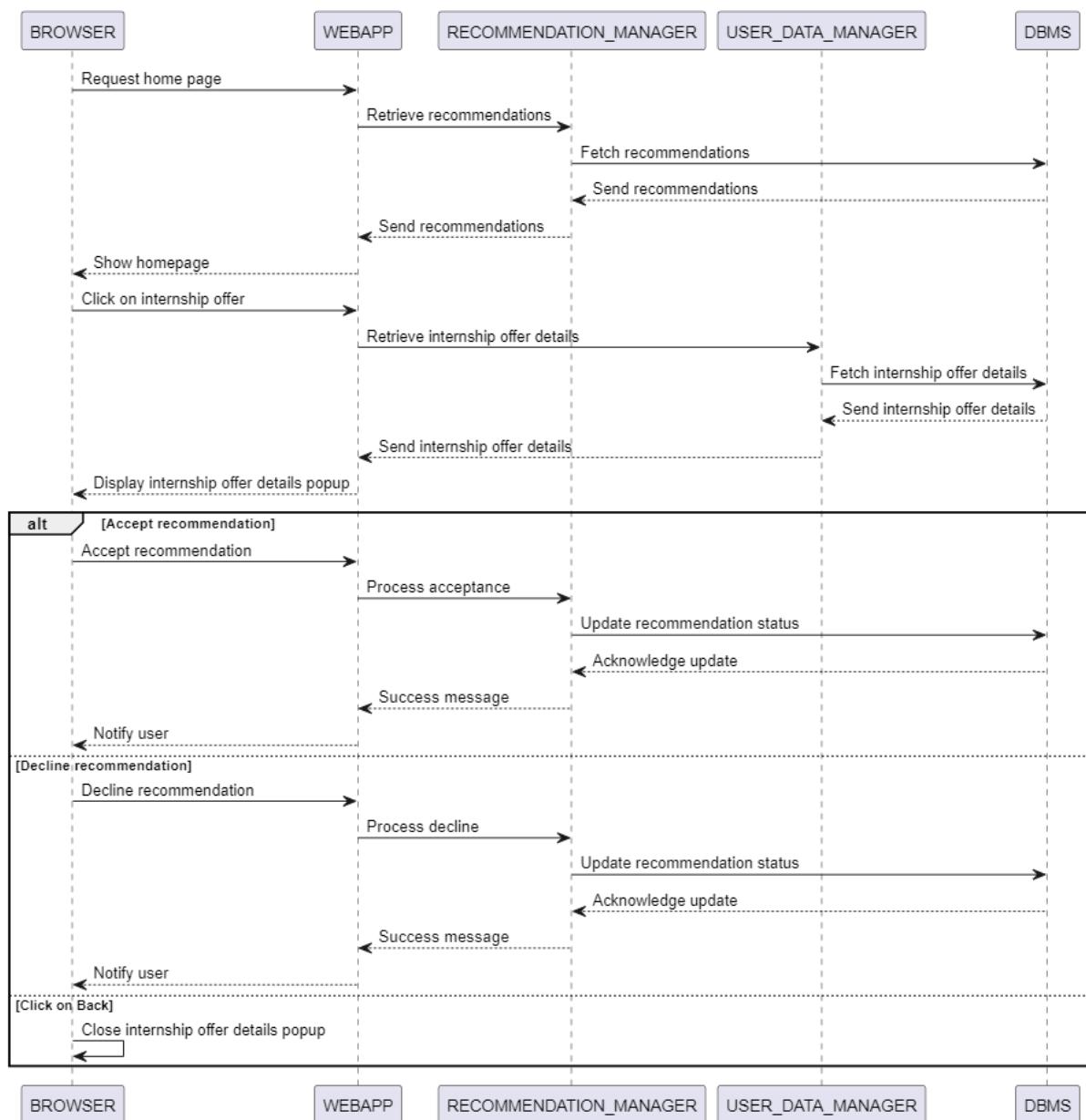


Figure 17: Student accepts/refuses system's recommendations

This diagram represents a new internship offer creation by a company operator. Once the user has accessed its profile page and clicked on the "New internship" button, it's now able to insert the new offer data in the forms shown. While the operator compiles the requested fields, S&C provides AI generated hints about how to write the offer through the User Data Manager interacting with the external AI service. Once the user has completed the forms filling in, clicks on the "Save" button. If all mandatory files have been filled (check performed by the User Data Manager), the data is saved in the DBMS, otherwise, an error message is shown.

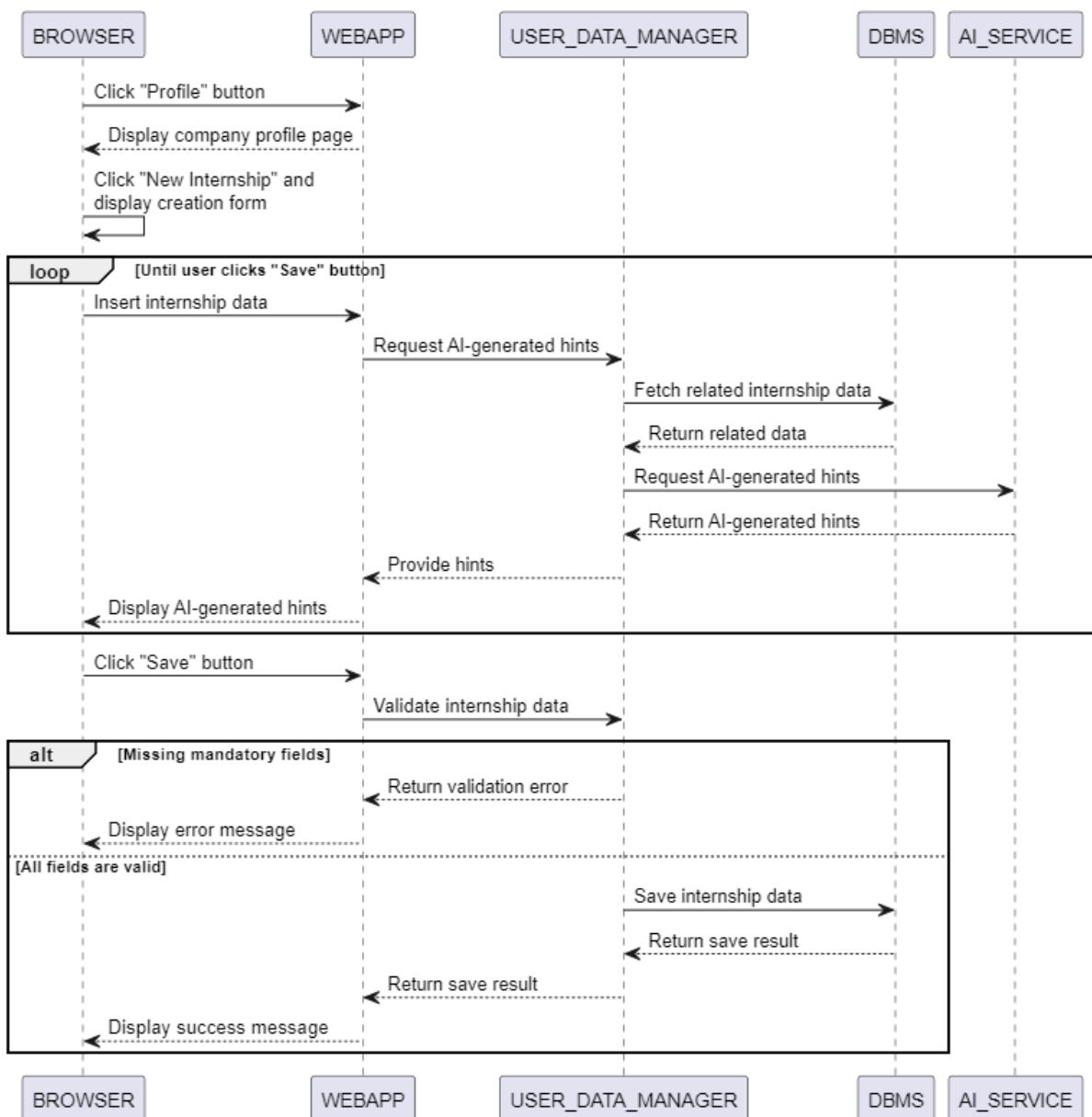


Figure 18: Company creates new internship

This diagram represents the process of acceptance/refusal of a candidate by a company. Once accessed the home page, the company operator can click on the paper icon related to a single student in order to see his details (all data is fetched from the DBMS through the User Data Manager). Also, the user can click on the green check icon (acceptance) or the "X" icon (refusal): so, the User Data Manager sends through the Mail Server an email to the candidate informing the result of the interview and updates the status on the DBMS.

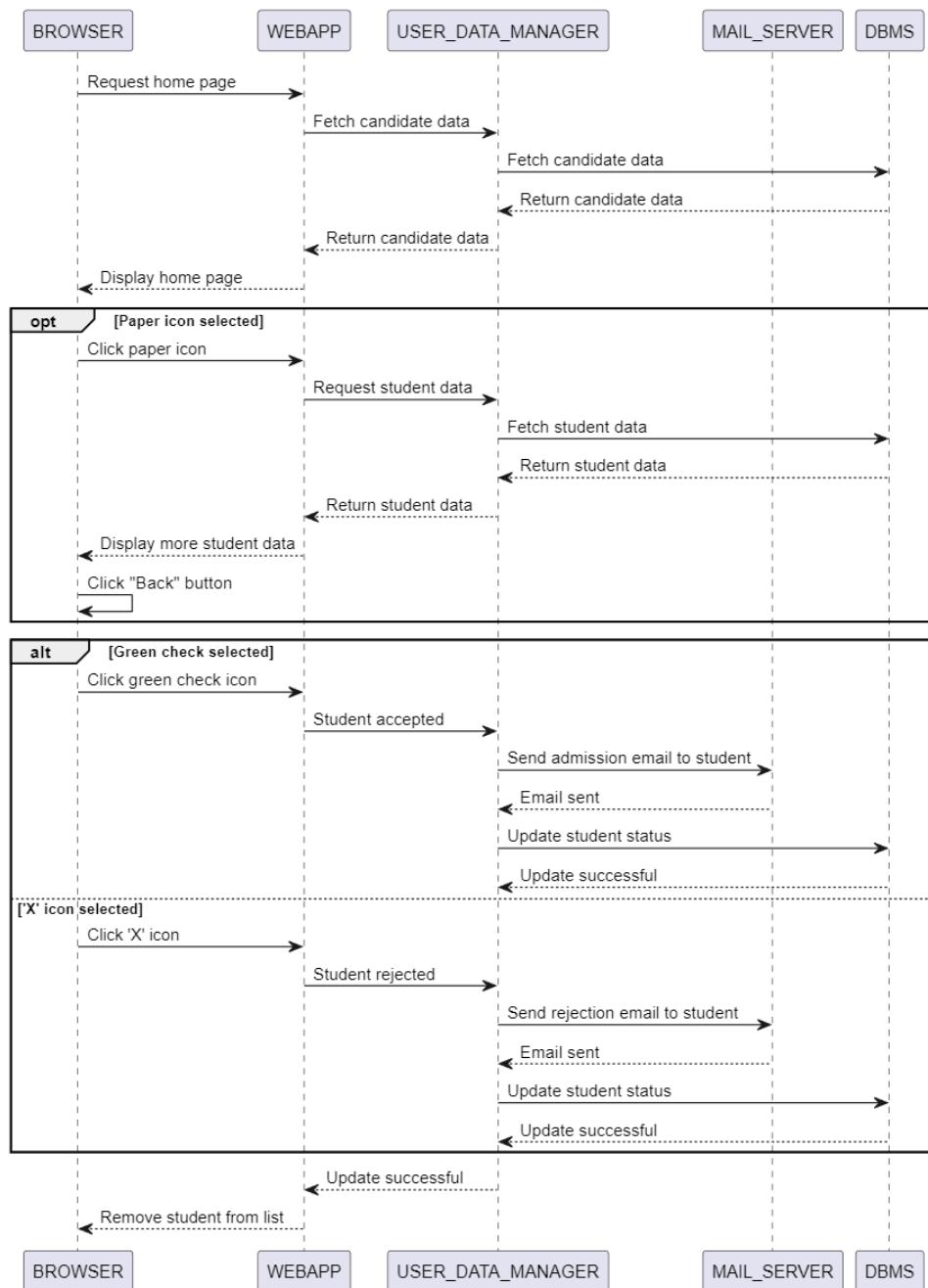


Figure 19: Company accepts/rejects candidate

This diagram represents the process of editing/opening/closing/deleting an internship offer by a company operator. Once the user has accessed the company profile page and clicked the "Edit internships" button, the operator is able to perform all these listed actions by clicking on the right button, which correspond to the internship offer which he wants to modify (the data concerning the internship offers list is fetched from the DBMS by the User Data Manager). If the operator clicks on "Open", "Close" or "Delete", the user Data Manager updates the internship status on the database and the user can then view the updated offers list. If the user clicks on "Edit", he can modify the internship details and then click "Save": the User Data Manager uploads the new data to the DBMS and the user is able to see the updated offers list.

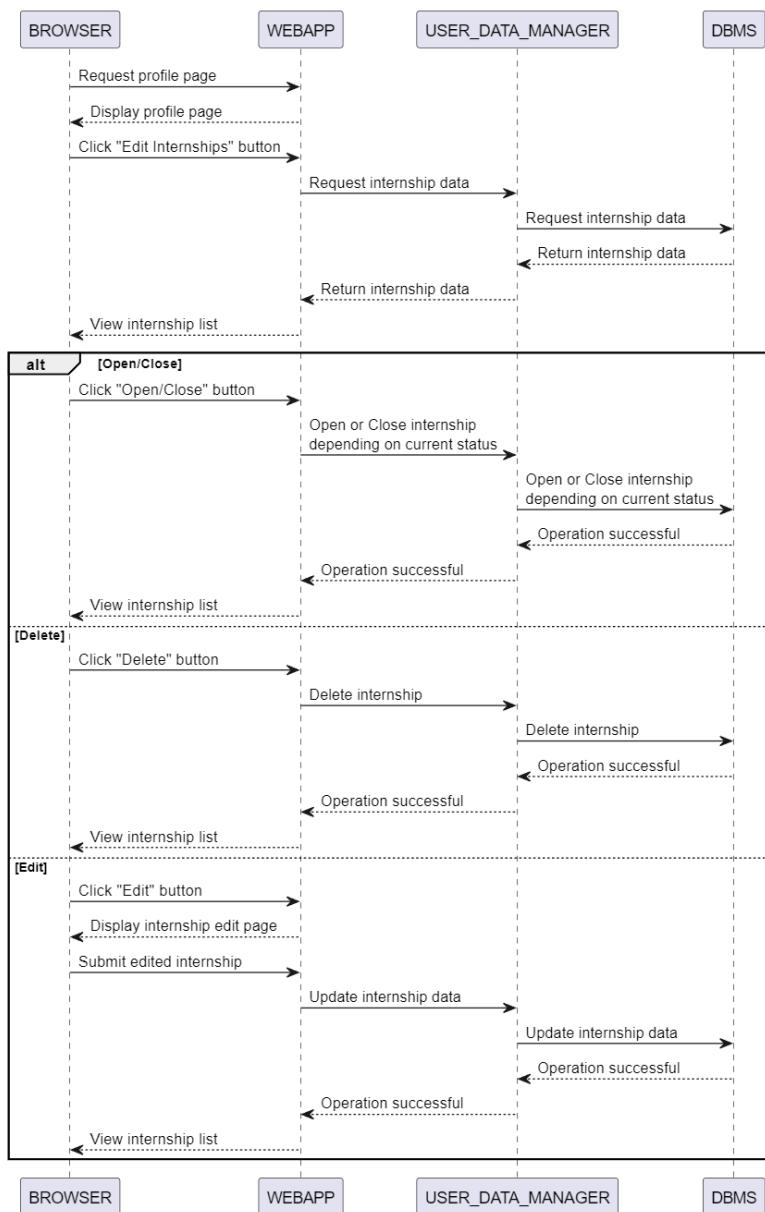


Figure 20: Company edits/opens/closes/deletes internship

In order to provide AI-suggested recommendations service, S&C performs periodic queries to the AI service through the Recommendation Manager and receives back potential matching pairs, which are stored in the DBMS.

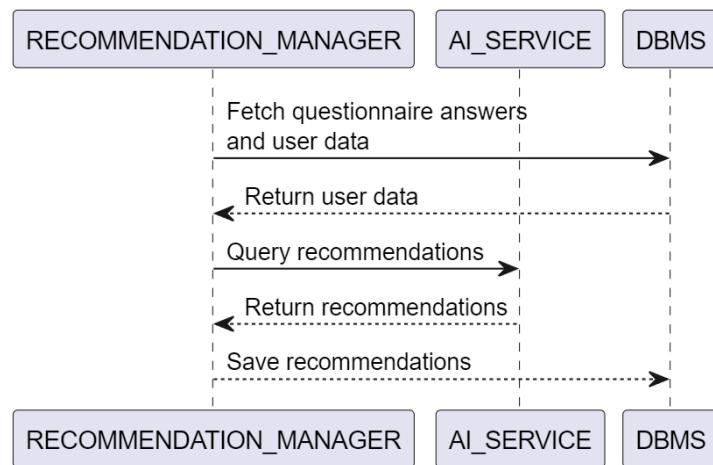


Figure 21: Periodic AI service query for recommendations

## 2.5 Component interfaces

- **Login manager**

- login\_student\_company (*String* email, *String* password)
- login\_university (*String* email)
- login\_otp (*String* otp)
- request\_reset\_password (*String* email)
- reset\_password (*String* new\_password)

- **Registration manager**

- registration\_student (*String* name, *String* surname, *String* institutional\_email, *String* residential\_address, *String* password)
- registration\_company (*String* company\_name, *Int* taxID, *String* headquarters\_address, *String* email\_address, *String* password)
- company\_operator\_auth (*String* email)
- confirm\_registration (*String* email)

- **Profile manager**

- student\_upload\_cv (*File* cv)
- company\_open\_internship (*String* application\_domain, *String* tasks\_to\_perform, *String* technologies, *Float* salary, *String* workplace, *String* benefits, *String* candidates\_skills\_required)
- company\_close\_internship (*Internship* internship)
- company\_edit\_internship (*Internship* internship)
- accept\_reject\_candidate (*String* candidate\_email)
- get\_filtered\_internships\_offers (*Map<String, Int>* filters)
- file\_complaint (*Internship* internship, *String* complaints)
- monitor\_internship\_complaints (*String* id)
- view\_internship\_details (*String* id)
- like\_internship\_offer (*String* id)
- view\_profile (*String* id)
- view\_interviewed\_students (*String* internship\_offer\_id)
- get\_student\_details (*String* id)

- **Chat manager**

- search\_user (*String* name)
- view\_chats (*String* user)
- open\_chat (*String* user, *String* chat\_id)
- send\_message (*String* chat\_id, *String* text)
- start\_video\_call (*List<String>* user\_ids)

- create\_questionnaire ()
- save\_new\_questionnaire (*Questionnaire* data)

- **Recommendation manager**

- get\_new\_ai\_recom (*File*) data
- get\_recom ()
- student\_accept\_reject\_recom (*String* recom\_id)
- send\_feedback\_questionnaire (*String* email, *String* link)
- save\_feedback\_data (*String* data)
- get\_analytics ()

## 2.6 Selected architectural styles and patterns

S&C is developed over a 3-tier architecture with thick client, which is useful to organize applications into three logical computing layers, reducing application complexity and enhancing fault tolerance, flexibility and scalability. The three layers are:

- **Data layer:** it stores data and sends information to the logic tier for elaboration. It is executed on the server.
- **Business logic layer:** it performs computations with data fetched from the Data tier and handles logical evaluations. In a thick client configuration, it is executed on the client.
- **Presentation layer:** it interacts with the user, displaying information and collecting inputs. It is executed on the client.

This architectural design is a particular configuration of the generally defined "Client-Server architecture": a design in which the client is the point of meeting between the application and the final user, displaying the dynamic User Interface and collecting inputs, and the server is dedicated to elaborate those inputs and provide answers. In the pure concept of this architectural design, the server represents a passive entity which replies only when clients sends requests: in this particular case, the server have to take action even if not invoked, for example, in the act of sending emails about recommendations-feedback questionnaires. Because of this, it should be considered as an active entity whose behavior should be described as "event-driven".

Client-Server architecture communication is performed by the use of REST-API, in order to enhance server scalability. The data encoding language used is JSON.

## 2.7 Other design decisions

### 2.7.1 Availability

Availability refers to the system's ability to remain operational and accessible to users. A monolithic architecture simplifies error handling and fault tolerance since all components reside within a single deployable unit. This eliminates issues such as network latency and service coordination that commonly affect microservices. However, a key drawback is the single point of failure—if the application crashes, the entire system becomes unavailable. In contrast, a microservices architecture enables fault isolation, meaning a failure in one service does not necessarily impact the entire system.

To mitigate this risk, redundancy mechanisms such as load balancers and replicated database instances can improve availability without introducing the complexity of microservices.

### 2.7.2 Scalability

Scalability determines the system's ability to handle increasing load. A monolithic architecture scales vertically by increasing server resources (CPU, memory), which is simpler and requires minimal configuration. However, it lacks the fine-grained scalability of microservices, where individual components can scale independently based on demand.

### 2.7.3 Ease of Deployment

One of the strongest advantages of a monolithic architecture is its ease of deployment. With a single deployable unit, there is no need to manage inter-service communication, API contracts, or container orchestration, which are all complexities introduced by microservices.

In contrast, microservices require careful versioning, networking strategies, and service discovery mechanisms, all of which increase the time and effort required for deployment and maintenance.

### 2.7.4 Data Storage

A monolithic architecture benefits from a single, centralized database, simplifying data consistency and transaction management. Microservices, by contrast, often require distributed databases or data synchronization mechanisms to avoid inconsistencies between services. Moreover, maintaining strong ACID transactions and consistent data models is significantly easier in a monolithic system than in a microservices-based one.

### 3 User Interface Design

In this section, mockups of several S&C User Interface pages will be presented. These designs will be adapted for mobile device UI also.

#### 3.1 Overview

The graph describes S&C's pages structure, highlighting connections between them.

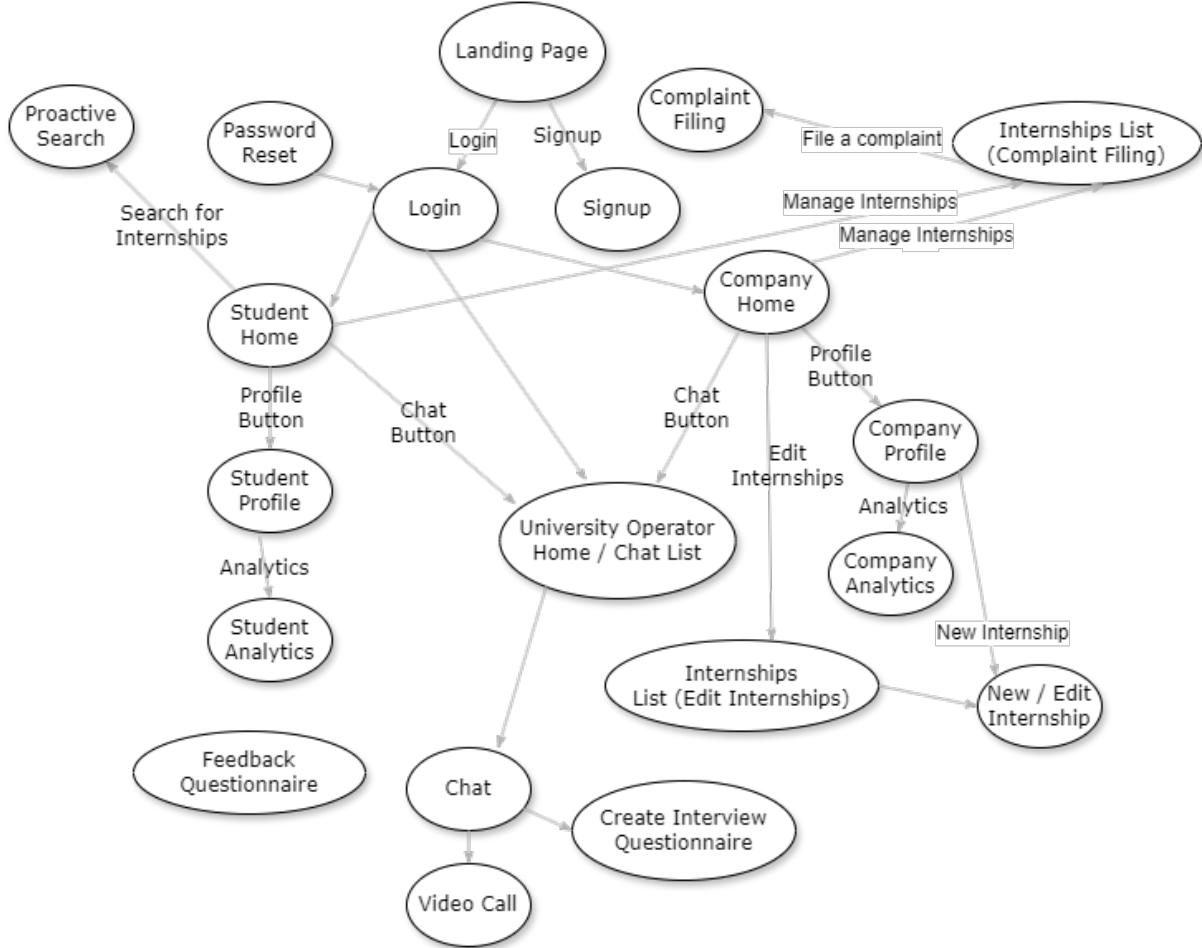


Figure 22: UI Map

### 3.2 UI Mockups

Any S&C page accessed without a valid login cookie will redirect to this page. The user here can either login or sign up.

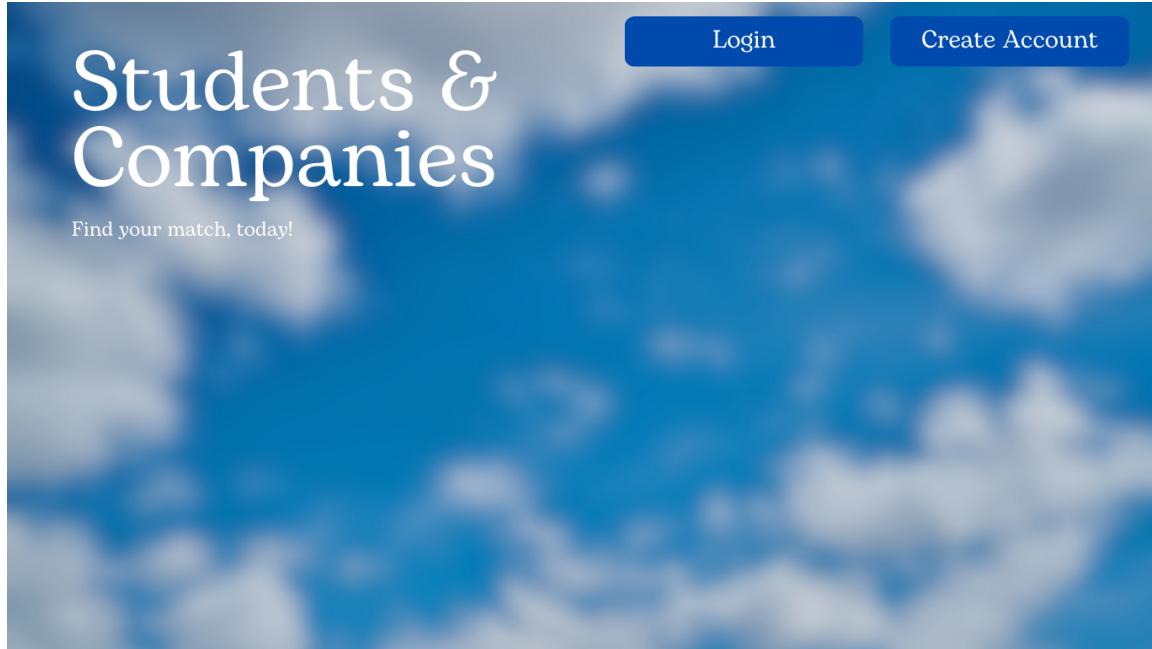


Figure 23: Landing Page

The user is able to sign-up to S&C by filling the right registration form, based on his own identity. Once the form is correctly and completely filled, the user is able to register and see its homepage.

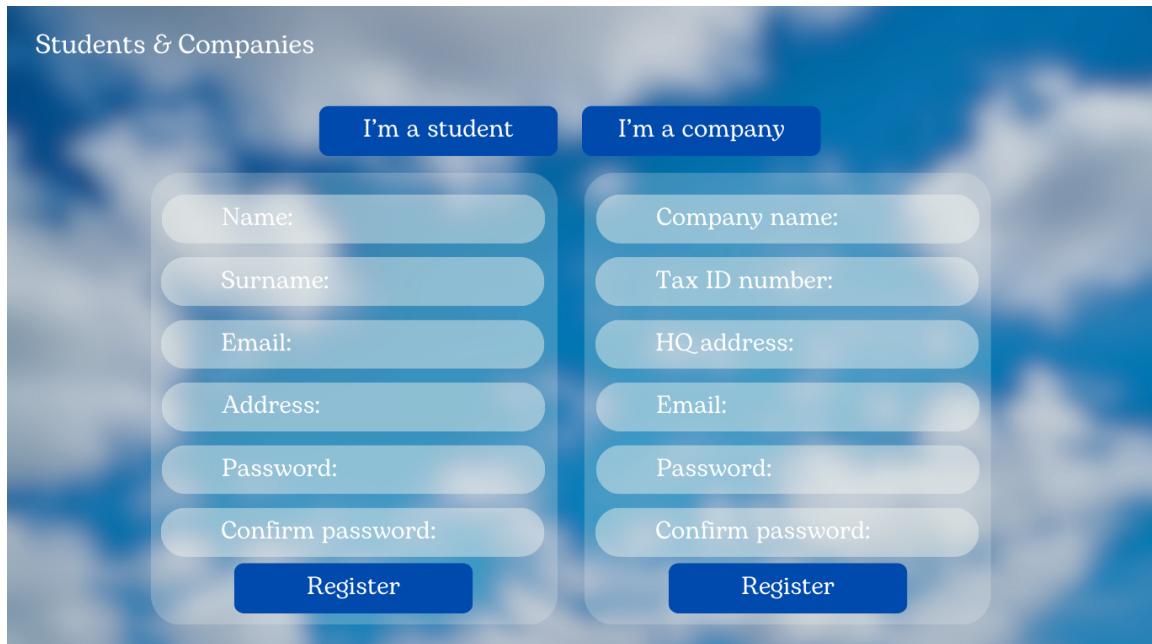


Figure 24: Sign Up Page

The user can fill in the form corresponding to his identity and then click the corresponding login or send OTP button to login to S&C.

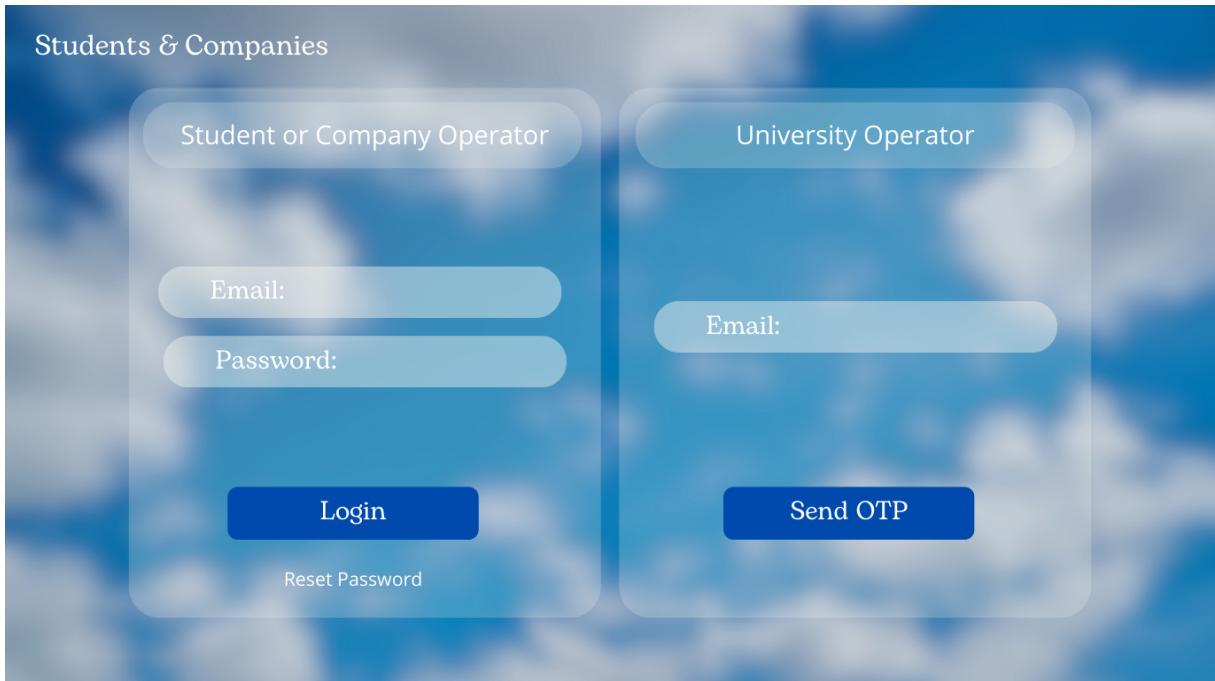


Figure 25: Login Page

From the homepage, the student is able to browse personalized internship offers or, alternatively, choose to manage his internships, start a "proactive search" or see his matches.

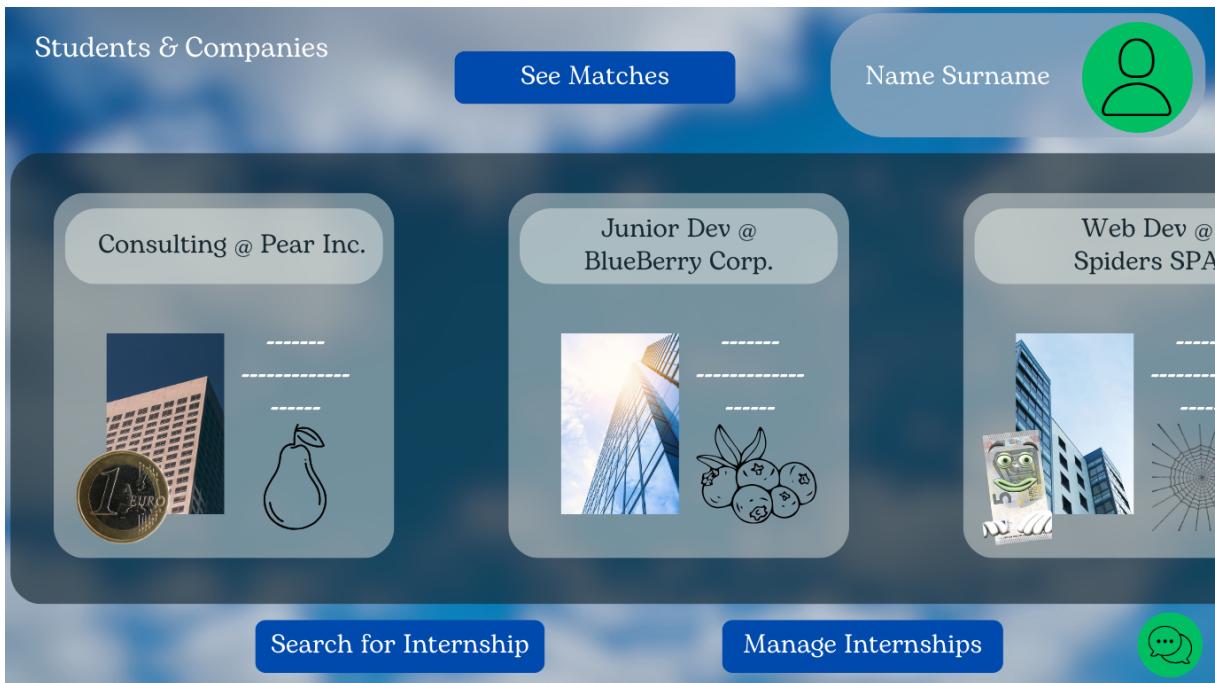


Figure 26: Student Home Page

In the internship searching page, the student is able to perform a "proactive search", by filtering search results and clicking on internships to see more details.



Figure 27: Student Proactive Search

In the student profile page, a student is able to edit his CV and see AI-generated suggestions about how to improve it, or to consult the profile statistics by clicking on the dedicated button.

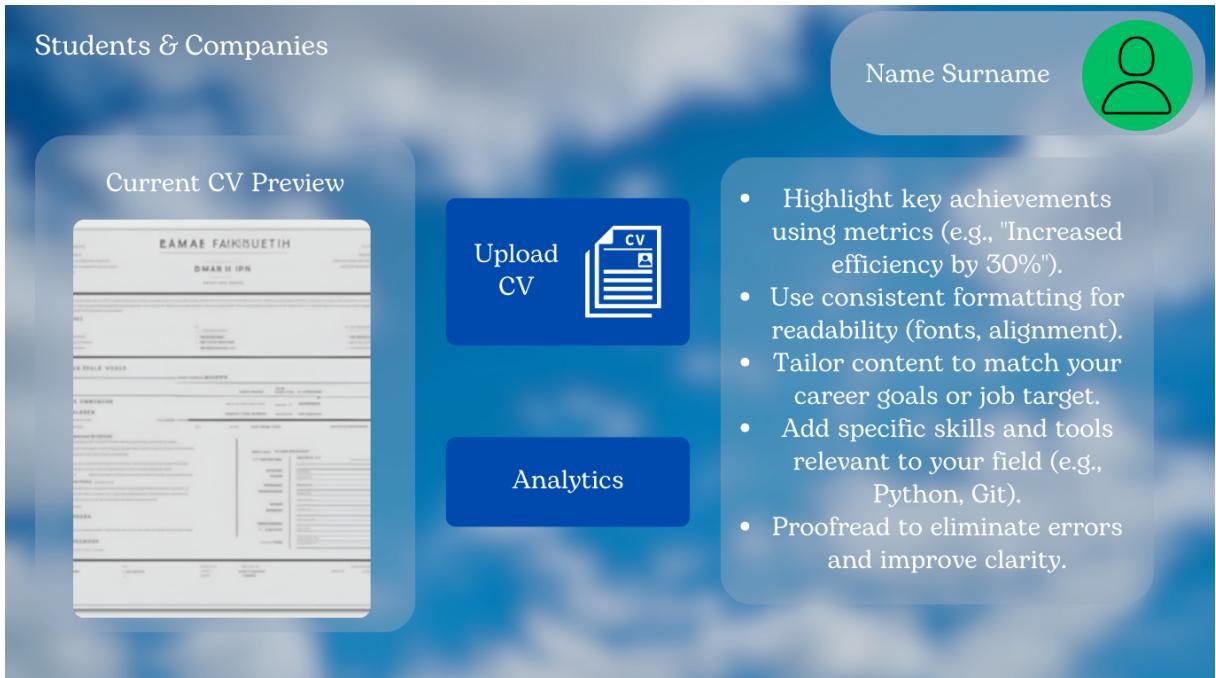


Figure 28: Student Profile Page

In the company homepage, a company operator is able to browse potential candidates or choose to manage the company's internships or see its matches by clicking on the corresponding buttons. He may scroll to the "Candidates selection" part of the homepage.

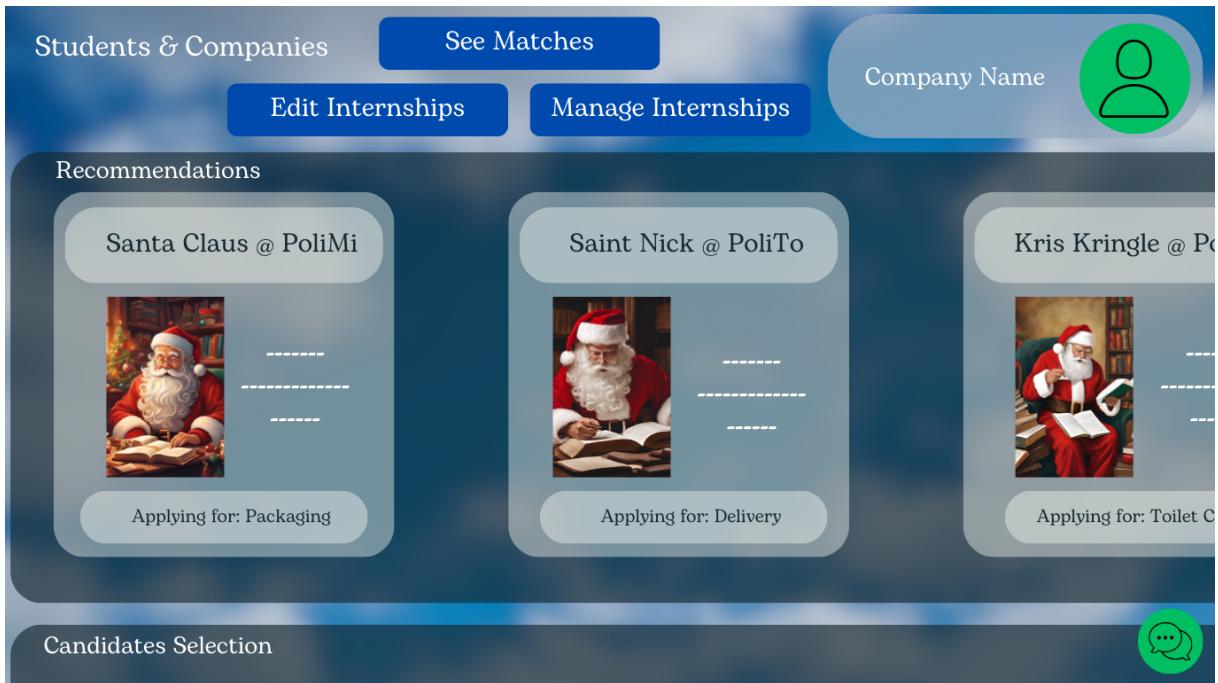


Figure 29: Company Home Page - Recommendations

In the company homepage, in the "Candidates selection" section, a Company operator is able to view students who applied to internships listed by its company and approve or reject them.

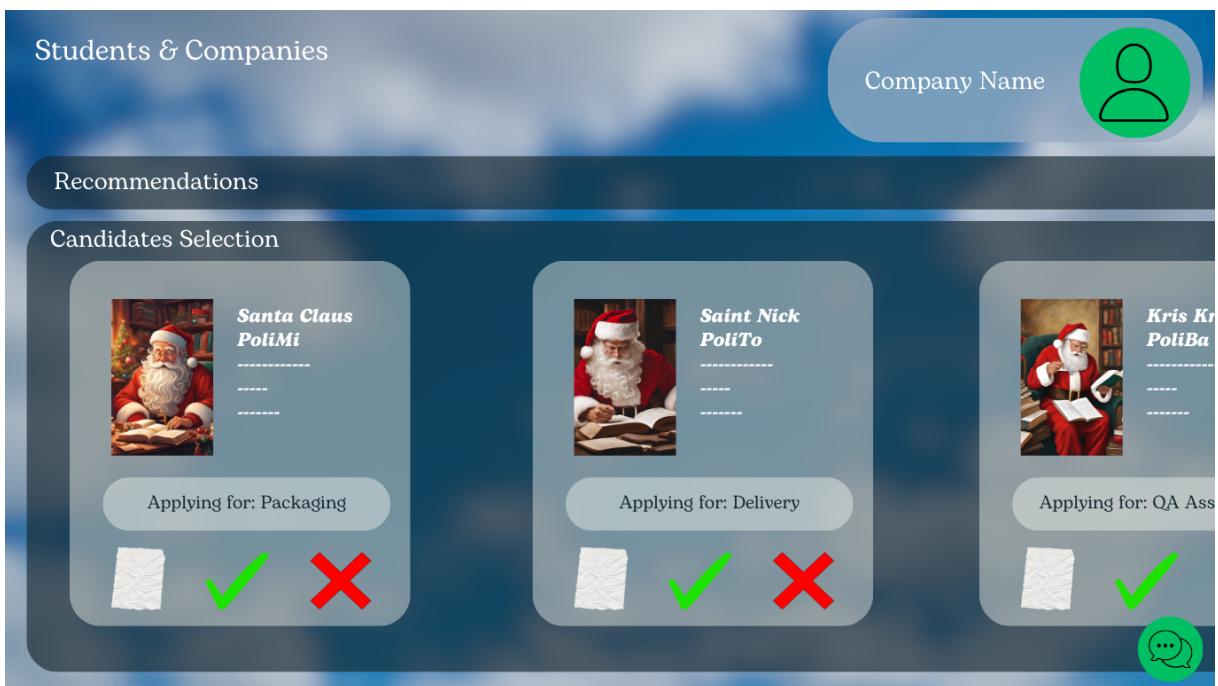


Figure 30: Company Home Page - Candidates Selection

In the company profile page, a Company operator is able to open a new internship position or edit an existing one, or consult profile statistics by clicking on the dedicated button.



Figure 31: Company Profile Page

In the internships management page, all the existing positions are listed and the operator is able to edit, open/close or delete each one of them.

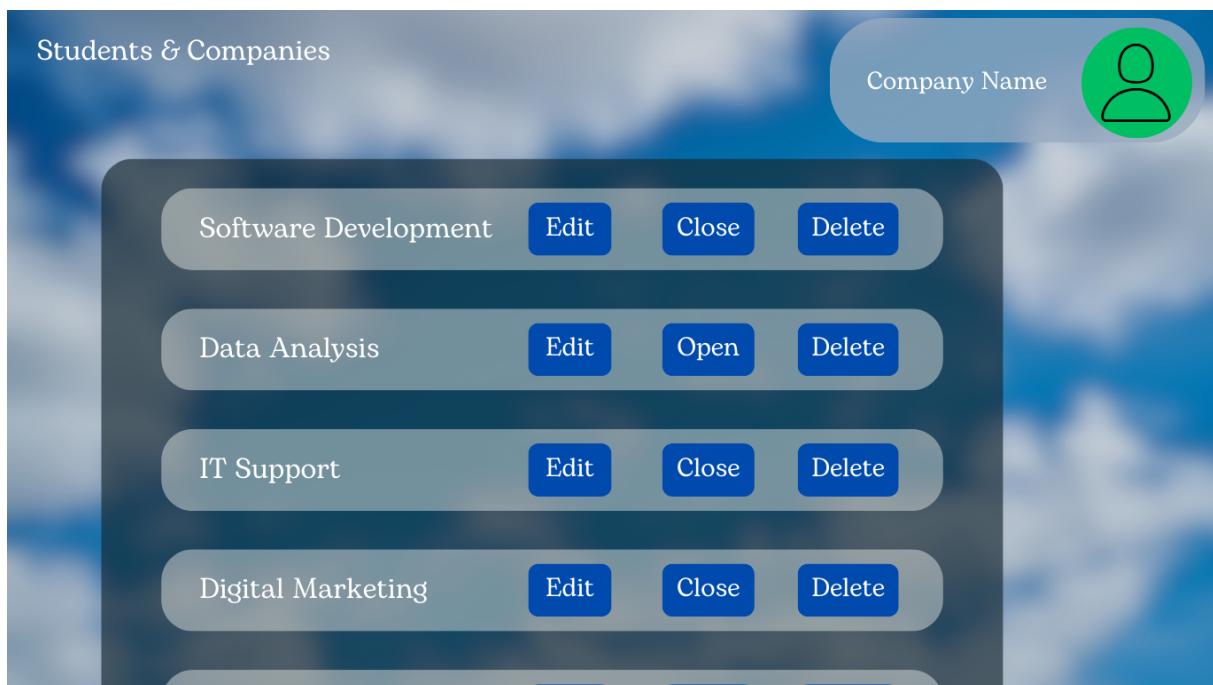


Figure 32: Company - Edit Internships

The internship editing page is displayed once the company operator has selected a single internship offer to edit, or in the case of opening a new one: in the first case the operator is able to edit the text fields, in the latter, the forms are displayed blank. AI suggestions on improving the current offer are shown on the right and updated in real time.



Figure 33: Company - New/Edit internship

In the chats list page, all active chats with other users are displayed. In order to open an existing chat, it is sufficient to click on the desired one, alternatively, it's possible to open a new one by clicking on the dedicated button.

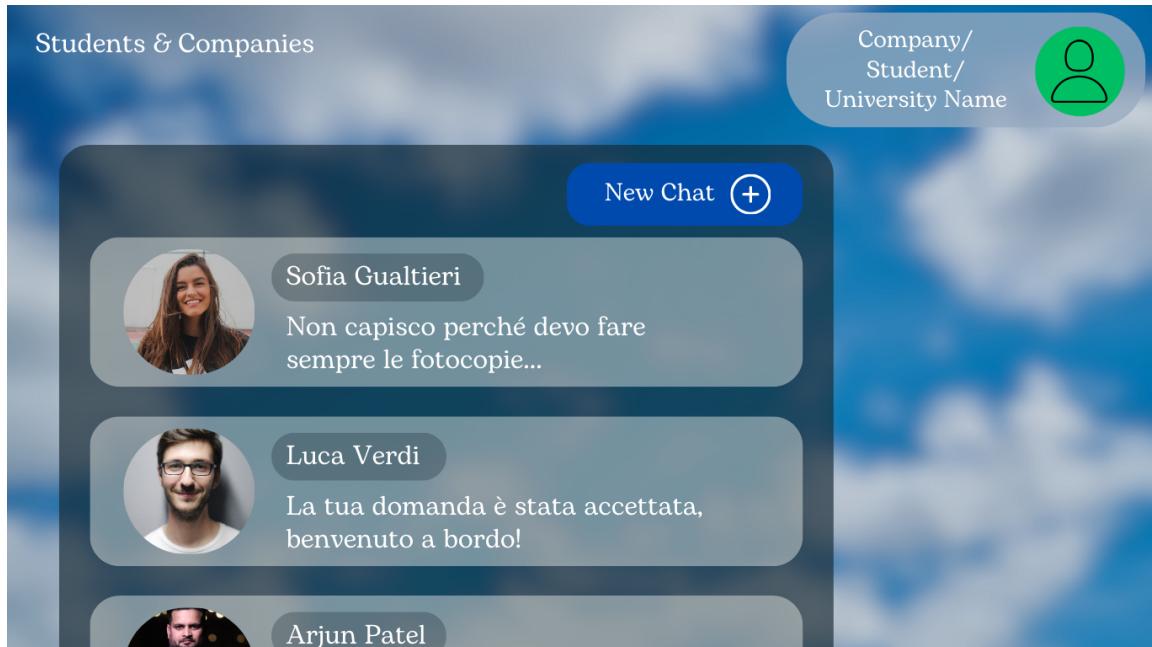


Figure 34: Chats List

The chat page allows the user to chat or video call another user. He may read their previous conversations.

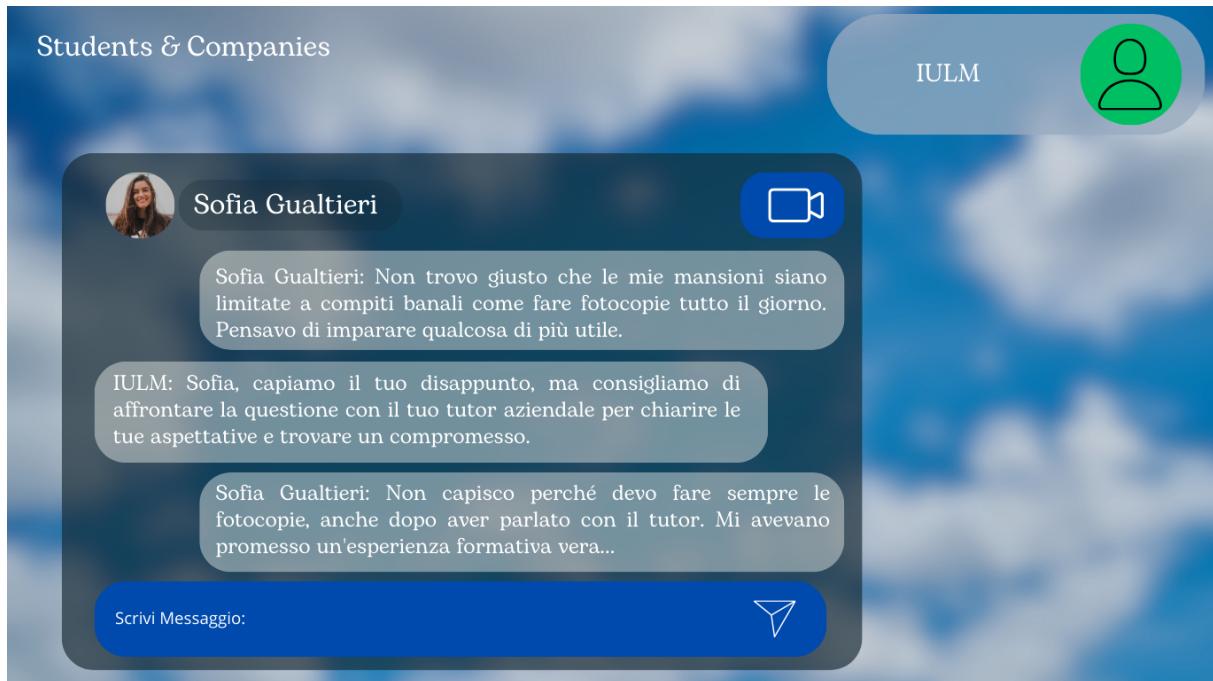


Figure 35: Chat

In the internships management page, the user can open a complaint about relevant internship(s) (just one if the user is a student, potentially multiple ones if it's a company) by clicking the appropriate button.

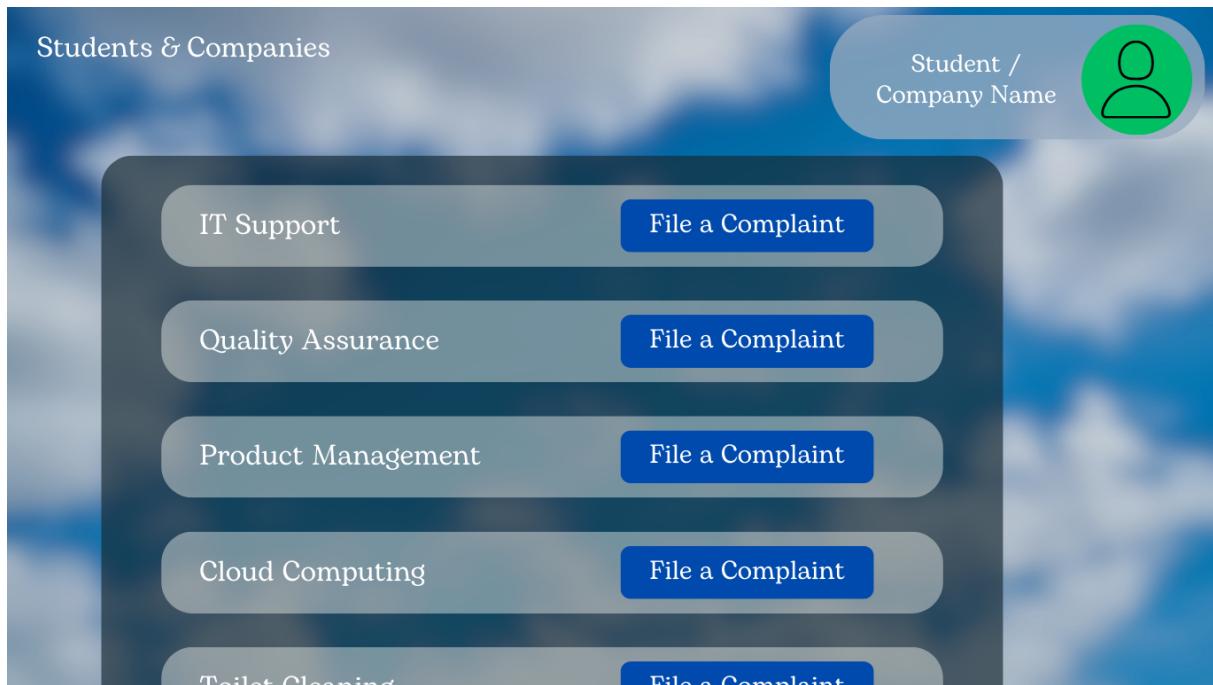


Figure 36: Internships Management Page

## 4 Requirements Traceability

### Login Manager

- R2: S&C allows registered students and companies to log in.
- R25: S&C allows university operators to log-in using an OTP mechanism.

### Registration Manager

- R1: S&C allows unregistered students and companies to sign up to the platform.

### Profile Manager

- R3: S&C allows students to upload/update their CV.
- R4: S&C allows registered companies to create, delete, activate or disable internships.
- R5: S&C allows registered students to proactively search for internships through appropriate filters.
- R13: S&C allows students or companies to file a complaint during internships.
- R14: S&C allows university operators to monitor internships and mediate complaints.
- R19: S&C allows students to view internships details.
- R17: S&C allows a user who has lost his password to recover it via traditional email-link.
- R18: S&C allows users to view their profile.
- R22: S&C allows companies to view all students who completed the interview process.
- R23: S&C allows companies to select students to start an internship.
- R24: S&C allows companies to reject students from their selection process.
- R27: S&C must provide students suggestions in order to improve their CVs.
- R28: S&C must provide companies suggestions in order to improve their internship offers.

### Recommendation Manager

- R6: S&C recommends internships to students based on their skills and interests.
- R7: S&C recommends candidates to companies based on students' skills and interests.

- R8: S&C allows students and companies to accept or decline system recommendations.
- R9: S&C automatically crunches all available data to provide better recommendations.
- R10: S&C collects feedback from students and companies.
- R12: S&C presents all data relevant to potential candidates to the companies.
- R15: S&C provides students suggestions about how to improve their CVs so as to be more appealing to companies.
- R16: S&C provides companies suggestions on how to improve project descriptions to be more appealing to students.
- R29: S&C allows students and companies to monitor the execution and the outcomes of the matchmaking process.
- R30: S&C must send questionnaires to students and companies about matchmaking performance and/or internship experience.

### **Chat Manager**

- R11: S&C provides companies with guided tools to interview candidates.
- R20: S&C allows students and companies to view their matches.
- R21: S&C allows students, companies and university operators to chat with/video-call.
- R26: S&C allows university operators to start a new chat with students or companies, in order to handle complaints.

# 5 Implementation, Integration and Test Plan

## 5.1 Overview and Implementation Plan

In this chapter, the system's implementation and integration and the test strategy are described. The adopted method is the "bottom-up" one: the implementation starts from the leaves of the "uses" hierarchy, from small functionalities which don't require any other one in order to work properly.

For each one of these modules, a driver is developed in order to permit testing. When a new module has been completely developed, it passes to the testing phase and then, finally, it can be integrated into the system (replacing the previously related driver). So, a brand new driver has to be developed in order to permit further testing. This strategy causes the separated development of more subsystems, which will be integrated all-in-one at the end of the process.

The "bottom-up" strategy results particularly useful to error and bugs tracking, ensuring effective testing. This strategy also allows independent teams to work in parallel on different functions.

## 5.2 Features Identification

**[F1] Login and Registration features** The system allows all users to login and register to the platform as the basic and first feature. This allows proper functioning of the other more advanced characteristics provided by S&C, such as recommendations, matching and chat.

**[F2] Suggestion feature** Thanks to the adoption of external AI tools, S&C is able to provide suggestions on how to improve the overall presentation of candidates and companies, which are crucial for the proper engagement of the platform.

**[F3] Chat/Video call features** In order to provide a complete service to users, communication is essential, that's why student-company and student-university chats and video calls are available.

**[F4] Monitoring features** In order to have a transparent vision of personal performance, S&C allows the access to analytics about internships and profile appeal. Also, in order to enhance the personalized AI-generated recommendations, S&C is able to autonomously reach students and companies with questionnaires about their matching experience.

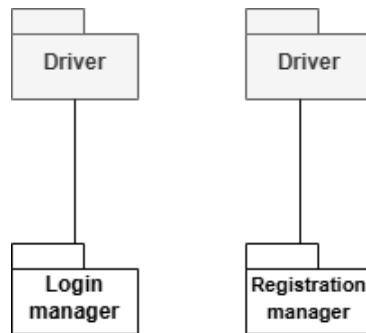
**[F5] Matching feature** Thanks to the use of appropriate filters, students can proactively match companies based on their interests, abilities and experiences, improving the overall quality of service. At the same time, AI tools provide possible matching pairs to students and companies, which are highlighted by S&C to users.

**[F6] Complaint feature** S&C allows students and companies to open a complaint about a specific ongoing internship, which is handled by a suitable university operator.

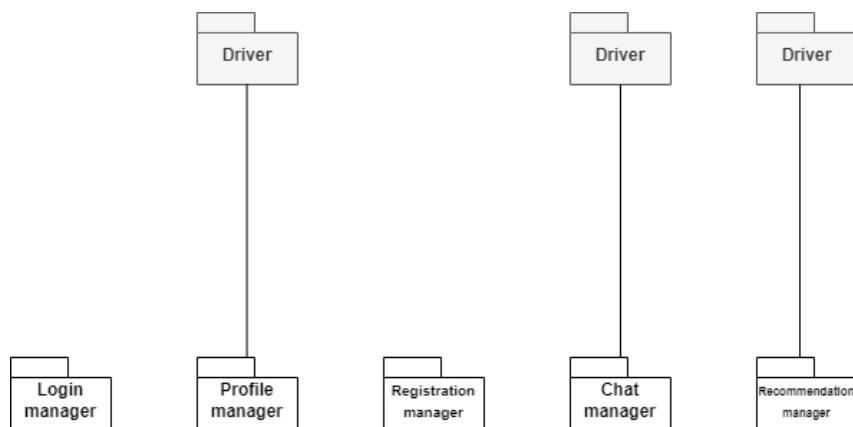
### 5.3 Integration Strategy

The system's components integration can start as soon as the hosting server and the DBMS are ready. The external Mail Server and AI Service can be integrated later: they will be necessary as soon as components that implement features which use these ones are integrated.

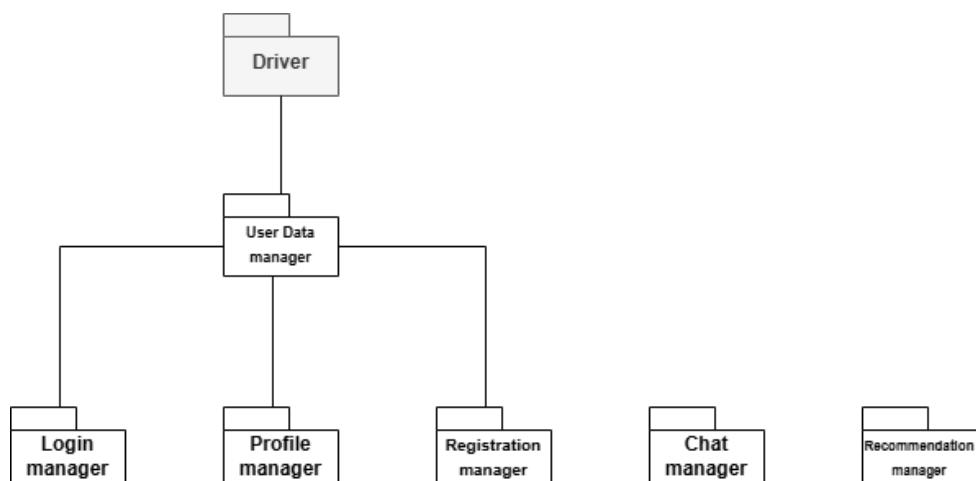
The first components to be integrated in the system are the Login Manager and the Registration Manager.



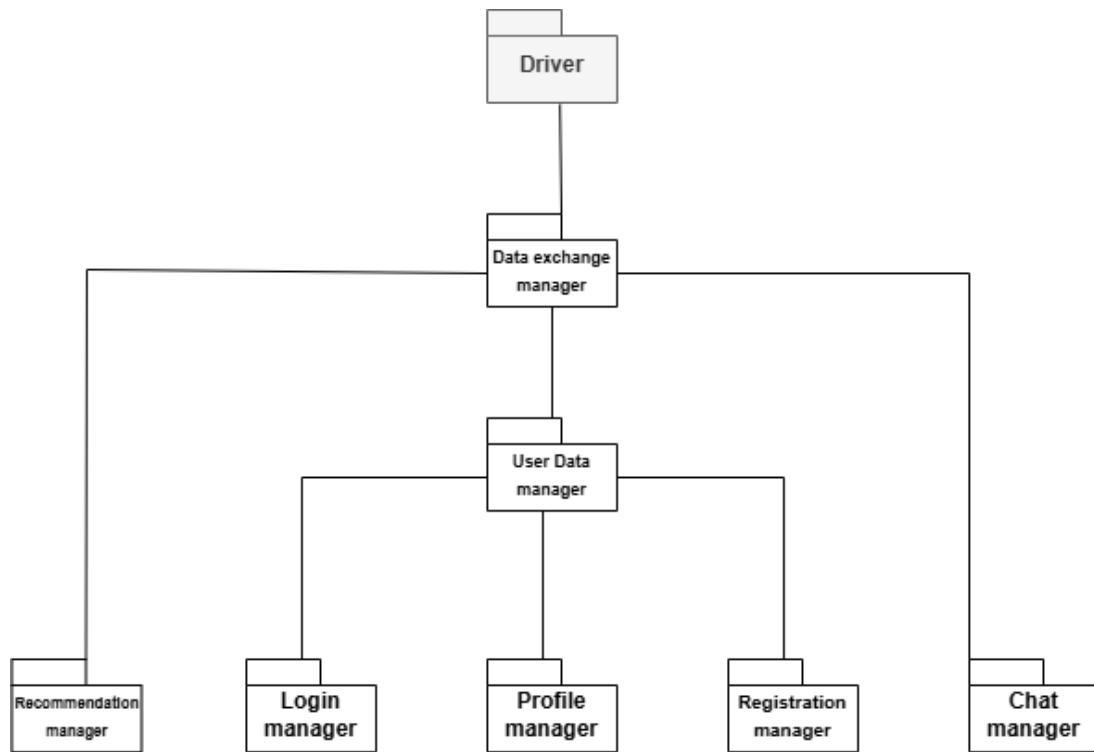
Then, Profile Manager, Chat Manager and Recommendation Manager are integrated.



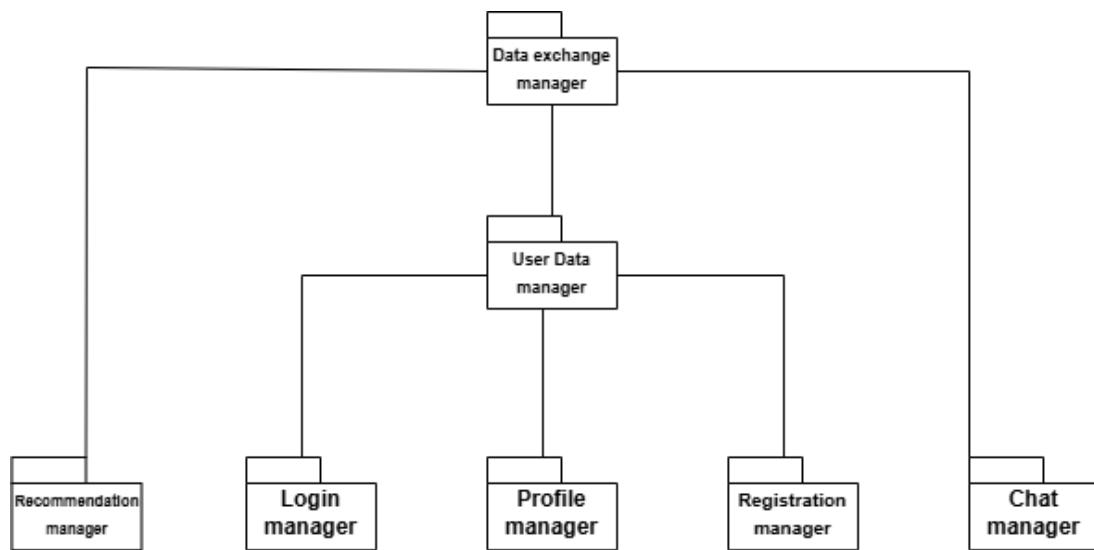
At this point, the User Data Manager is integrated. The Profile Manager, Login Manager and Registration Manager are linked to this new component.



Finally, the Data Exchange Manager is integrated, which will permit information exchange of all existing components with the WebApp.



No more drivers are required, having entirely built and tested the system, which is represented below.



## 5.4 System Testing Strategy

Before and after the integration into the system of every new developed component, a rigorous testing phase has to be performed, in order to grant requested system's properties, correct workflow and absence of bugs and errors. In this procedure, the use of drivers is primary. Pursuing these goals, multiple types of testing are required, described below.

- **Functional testing:** performed in order to grant the workflow correctness, required functionalities presence and alignment with the details contained in the RASD document, ensure system's goals and requirements reaching and fulfillment.
- **UI testing:** performed in order to ensure the user interface experience results optimal and accessible on every possible device type and browser.
- **Performance testing:** performed in order to look for possible bottlenecks and every type of performance issues, observing system's resiliency while subjected to heavy workload.
- **Stress testing:** performed in order to be able to ensure the possibility of fast and effective system recovery after an eventual failure event.
- **Load testing:** performed in order to check for memory leaks, buffer overflows and any other possible type of code misconduct.

## 6 Effort Spent

Member of group	Effort spent	
Falzoni Elia	Introduction	3h
	Architectural Design	13h
	User Interface Design	2h
	Requirements Traceability	1h
	Implementation Integration Test Plan	4h
	Reasoning / Planning	6h
Toffoli Andrea	Introduction	2h
	Architectural Design	16h
	User Interface Design	3h
	Requirements Traceability	1h
	Implementation Integration Test Plan	3h
	Reasoning / Planning	7h
Torti Andrea	Introduction	1h
	Architectural Design	14h
	User Interface Design	8h
	Requirements Traceability	1h
	Implementation Integration Test Plan	3h
	Reasoning / Planning	4h

Table 1: Effort spent (in hours) by each member of the group.

## 7 References

### 7.1 References

- The Requirement Engineering and Design Project specification document, A.Y. 2024/2025.

### 7.2 Used Tools

- GitHub for project versioning and sharing.
- L<sup>A</sup>T<sub>E</sub>X and *Overleaf.com* as editor to work on this project.
- *PlantUML* and *Draw.io* for diagrams' design.
- *Canva* for mockup's design.
- *Google Documents* for collaborative writing, roadmap schedule and progress tracking.

## List of Figures

1	S&C architectural design overview . . . . .	7
2	High-level S&C architectural design overview . . . . .	8
3	Low-level S&C component view overview . . . . .	9
4	Deployment Diagram . . . . .	10
5	Student or Company Operator Login . . . . .	12
6	Student or Company sign-up . . . . .	13
7	Company accepts/declines system's recommendations . . . . .	14
8	Student updates CV . . . . .	15
9	Student Proactively Looks For Internships . . . . .	16
10	Student or Company looks at analytics . . . . .	17
11	Student or Company files a complaint . . . . .	18
12	Student or Company Operator fills feedback questionnaire . . . . .	19
13	University Operator Login . . . . .	20
14	Student or Company or University Operator chats . . . . .	21
15	Student or company resets account's password . . . . .	22
16	Company interviews a candidate . . . . .	23
17	Student accepts/refuses system's recommendations . . . . .	24
18	Company creates new internship . . . . .	25
19	Company accepts/rejects candidate . . . . .	26
20	Company edits/opens/closes/deletes internship . . . . .	27
21	Periodic AI service query for recommendations . . . . .	28
22	UI Map . . . . .	32
23	Landing Page . . . . .	33
24	Sign Up Page . . . . .	33
25	Login Page . . . . .	34
26	Student Home Page . . . . .	34
27	Student Proactive Search . . . . .	35
28	Student Profile Page . . . . .	35
29	Company Home Page - Recommendations . . . . .	36
30	Company Home Page - Candidates Selection . . . . .	36
31	Company Profile Page . . . . .	37
32	Company - Edit Internships . . . . .	37
33	Company - New/Edit internship . . . . .	38

34	Chats List . . . . .	38
35	Chat . . . . .	39
36	Internships Management Page . . . . .	39

## List of Tables

1	Effort spent (in hours) by each member of the group. . . . .	46
---	--	----