



**POLITECNICO**  
**MILANO 1863**

SCUOLA DI INGEGNERIA INDUSTRIALE  
E DELL'INFORMAZIONE

# Design Document

**Version 1.0**

by  
Abdallah Alkhetiar  
Daniel Bonardi

December 3, 2024

# Document details

<b>Deliverable:</b>	DD
<b>Title:</b>	Design Document
<b>Authors:</b>	Abdallah Alkhetiar and Daniel Bonardi
<b>Version:</b>	1.0
<b>Date:</b>	December 3, 2024
<b>Download page:</b>	<a href="https://github.com/Zero3474/AlkhetiarBonardi.git">github.com/Zero3474/AlkhetiarBonardi.git</a>
<b>Copyright:</b>	Copyright © 2025, Abdallah Alkhetiar and Daniel Bonardi – All rights reserved

# Contents

<b>1</b>	<b>Introduction</b>	<b>3</b>
1.1	Purpose . . . . .	3
1.2	Scope . . . . .	3
1.3	Definitions, acronyms, abbreviations . . . . .	3
1.4	Revision history . . . . .	3
1.5	Reference documents . . . . .	3
1.6	Document structure . . . . .	3
<b>2</b>	<b>Architectural design</b>	<b>5</b>
2.1	Overview . . . . .	5
2.2	Component view . . . . .	5
2.2.1	High level component view . . . . .	5
2.2.2	Internship management diagram . . . . .	6
2.2.3	Account management diagram . . . . .	7
2.3	Deployment view . . . . .	8
2.4	Runtime view . . . . .	8
2.4.1	Login . . . . .	8
2.4.2	Sign up . . . . .	9
2.4.3	Internship creation . . . . .	10
2.4.4	Internship application . . . . .	11
2.4.5	Form assignment for internship . . . . .	12
2.4.6	Form compilation . . . . .	13
2.4.7	Form correction . . . . .	14
2.4.8	Student selection . . . . .	15
2.5	Component interfaces . . . . .	16
2.6	Selected architectural styles and patterns . . . . .	17
2.7	Other design decisions . . . . .	17
<b>3</b>	<b>User interface design</b>	<b>18</b>
<b>4</b>	<b>Requirement traceability</b>	<b>25</b>
<b>5</b>	<b>Implementation, integration and test plan</b>	<b>26</b>
<b>6</b>	<b>Effort spent</b>	<b>27</b>
<b>7</b>	<b>References</b>	<b>28</b>

# 1 Introduction

## 1.1 Purpose

This design document outlines the architectural framework, user interface design, and other system components necessary to meet the functional and non-functional requirements described in the Requirement Analysis and Specification Document.

It also serves as a foundation for developers, testers, and stakeholders to understand the system's structure and ensures that the final implementation adheres to the defined goals.

The main purpose of the S&C platform is to grant efficient and effective matchmaking between university students seeking experience through internships and companies offering them. The design aims to ensure:

- **Precision in matching:** The system adopts a sophisticated recommendation system to notify students about suitable internships, based on criteria such as skills and experiences.
- **Smooth user experience:** The system shall grant its users the best experience by smoothly managing all internship-related workflows, including searching, application, selection, feedback collection, and monitoring of ongoing internships.
- **Scalability:** The system's design shall accommodate future adjustments, such as advanced data analytics, integration with external tools, or even an expanded scope.

## 1.2 Scope

**Students&Companies** (S&C) is an internship platform designed to connect students with companies based on skills, experiences, and interests.

The platform offers two main ways to establish connections:

- **Recommendation system**
- **Proactive searching**

S&C also supports the selection process and gathers feedback on recommendations and provides spaces for users to address issues, share updates, and track internship progress. More details can be found in the RASD document.

## 1.3 Definitions, acronyms, abbreviations

TBD

## 1.4 Revision history

Version 1.0 - WIP

## 1.5 Reference documents

TBD

## 1.6 Document structure

### Section 1: Introduction

This section is presented with a brief overview of this document's content including the description of the purpose, the scope, and all the definitions, acronyms and abbreviations used.

## **Section 2: Architectural design**

This section is presented with a detailed description of the architectural choices for the system, including a high-level description of the system and its components. In addition it is presented a detailed description of the deployment view and the main runtime views of the system.

## **Section 3: User interface design**

This section is presented with a series of prototypes for the most important pages of the software to help graphic designers. Each image is accurately described to avoid any misunderstanding on the pages' available functionalities.

## **Section 4: Requirements traceability**

This section is highly dependent on the RASD, as it provides a complete mapping of both functional and non-functional requirements described in that document, with the modules introduced and described in this document.

## **Section 5: Implementation, integration and test plan**

The last section describes the procedures followed for implementing, integrating and testing the components of the platform.

## 2 Architectural design

### 2.1 Overview

The S&C design is based on two architectural choices:

- A 3 tier architecture
- A thin-client model

First of all the 3 tier architecture is built using 3 layers:

- **Presentation layer:** This layer is in charge of providing all the user interfaces for users to interact with the system.
- **Application layer:** This layer implements the core of the business logic and processes all the requests sent from the presentation layer.
- **Data layer:** This layer is responsible of managing all the data storage.

The choice of the 3 tier architecture what made to grant these key concepts:

- **Modularity:** Components are designed address a specific function required by the system, also making it easier to perform maintenance.
- **Scalability:** Since each module function independently from the others, adding or removing feature should be easier.
- **Security:** By implementing a secure communication protocol, users should not be worried about their sensitive data being stolen.

The thin-client approach instead, ensures the smoothest experience for the users, by keeping the client-side of the application as lightweight as possible. With this approach the client shall focus solely on the rendering of the user interface and the management of basic interactions, the rest of the computation happens on the server-side.

To sum it up, in the client there is only the presentation layer, while the application and data layer are in the server.

### 2.2 Component view

To make it easier to understand component view, we decided to divide it into 3 diagrams. The first represents a high-level view that distinguishes the client-side from the server-side components. The other 2 diagrams represent a more detailed version of the components presented in the server-side in the first diagram.

#### 2.2.1 High level component view

In the following diagram it is shown how the client-side interacts with the server side through the exposed interfaces.

We can clearly see the 2 sides:

- **Client-side:** Since the in the client-side we only have the presentation layer, we can see that on the left of the diagram there are the Users, that can either be accessing the application through a web browser or a mobile device.
- **Server-side:** On the server-side instead, we have both the application and data layer so we have the most important subsystems of the entire application:

- *Internship management*: Which is the actual biggest component which manages all the actions that involve internships.
- *Account management*: Which simply manages all the interactions that require some data regarding a user.

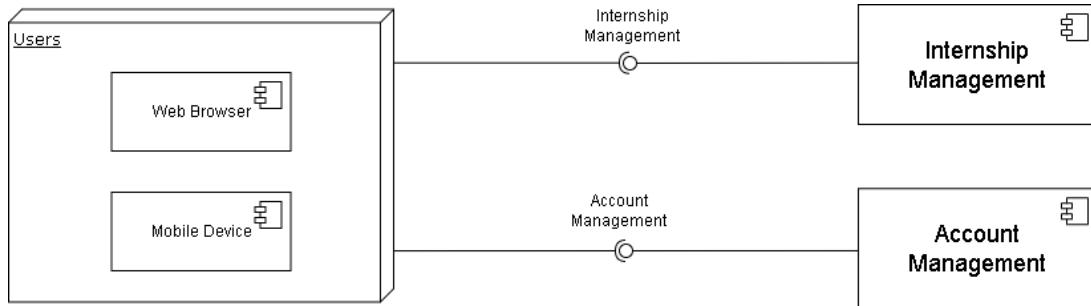


Figure 1: High level component view

### 2.2.2 Internship management diagram

The following diagram shows how the subsystem Internship Management is composed and how its modules communicate with external ones. This subsystem contains 6 modules:

- **Proactive search**: Allows student to search for specific internship.
- **Internship archival**: Allows companies to archive an internship, and show it on their profile.
- **Feedback**: Allows user to send a feedback.
- **Internship creation**: Allows companies to create an internship, and for this reason it needs to interact with the "Recommendation system" component to notify the compatible students.
- **Complaints**: Allows user to file a complaint, and it also needs to interact with the "Notification system" to notify the universities.
- **Form management**: Manages all the stages of a form, from its creation to the correction, and for this reason it needs to interact with the "Ranking system" component.

At the same time all components, external or internal to the subsystem, must interact with the DBMS to access all the data required for the interaction.

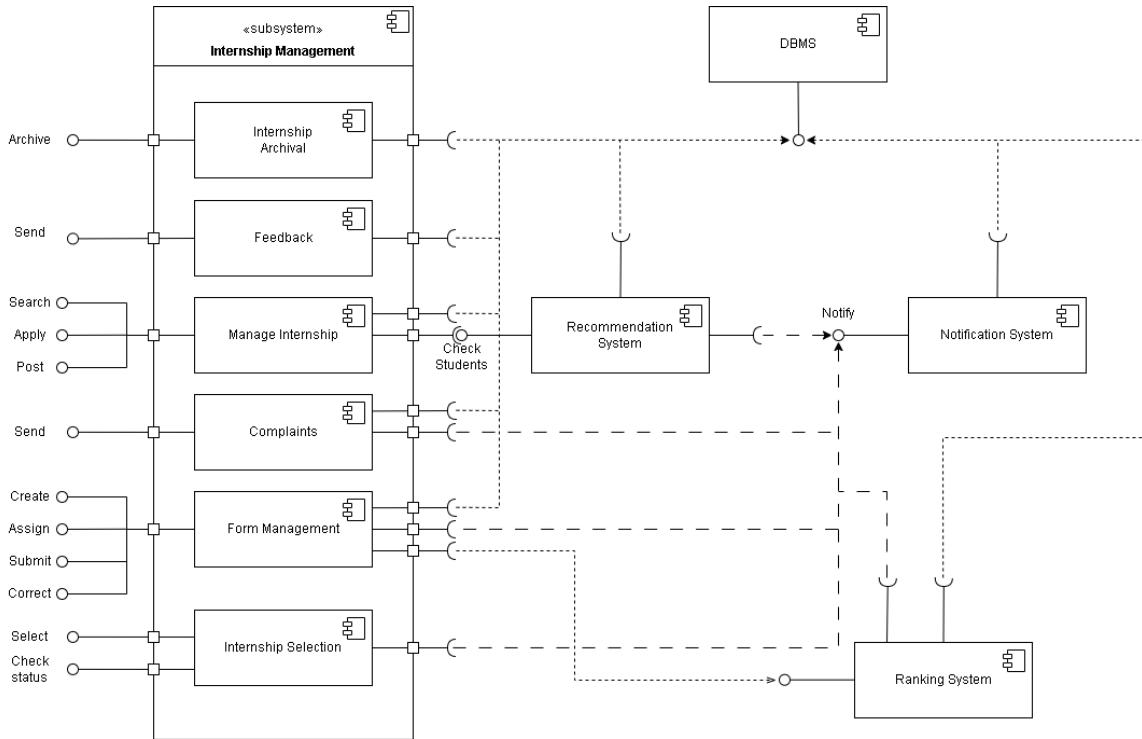


Figure 2: Internship management diagram

### 2.2.3 Account management diagram

The following diagram shows how the subsystem Account Management is composed and how its modules communicate with external ones. This subsystem contains only 2 modules:

- **Student manager:** Allows student to sign up, login, add a CV and update their profile.
- **Company manager:** Allows a company to sign up, login and update their profile.

In both cases the component need to access both the "Authentication" component to safely complete the login and sign up processes, and also the DBMS to manage all the required data.

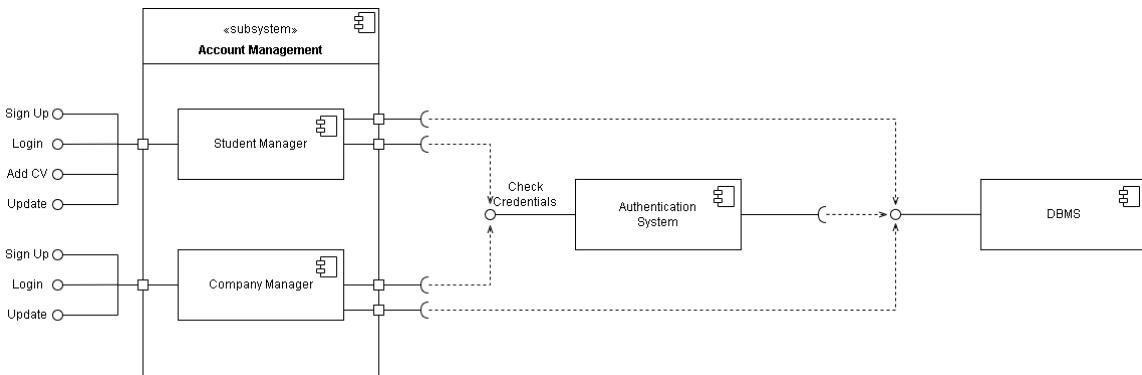


Figure 3: Account management diagram

## 2.3 Deployment view

## 2.4 Runtime view

With the following sequence diagrams, we are going to explain and represent all the interactions that happen between the components of the S&C system. But since this representation is still a high-level description of interactions, the function names, parameters, results, errors, and other smaller details will be modified or added during the development phase.

### 2.4.1 Login

In the login process, the user interacts with the *Web Application* to login in his account. After adding the required information, the data is sent to the *Account Manager* that checks if the credentials are correct through the *Authentication System* that accesses the *DBMS*. If the credentials are correct, the user is shown the home page, otherwise he is shown a message error and can try to login again.

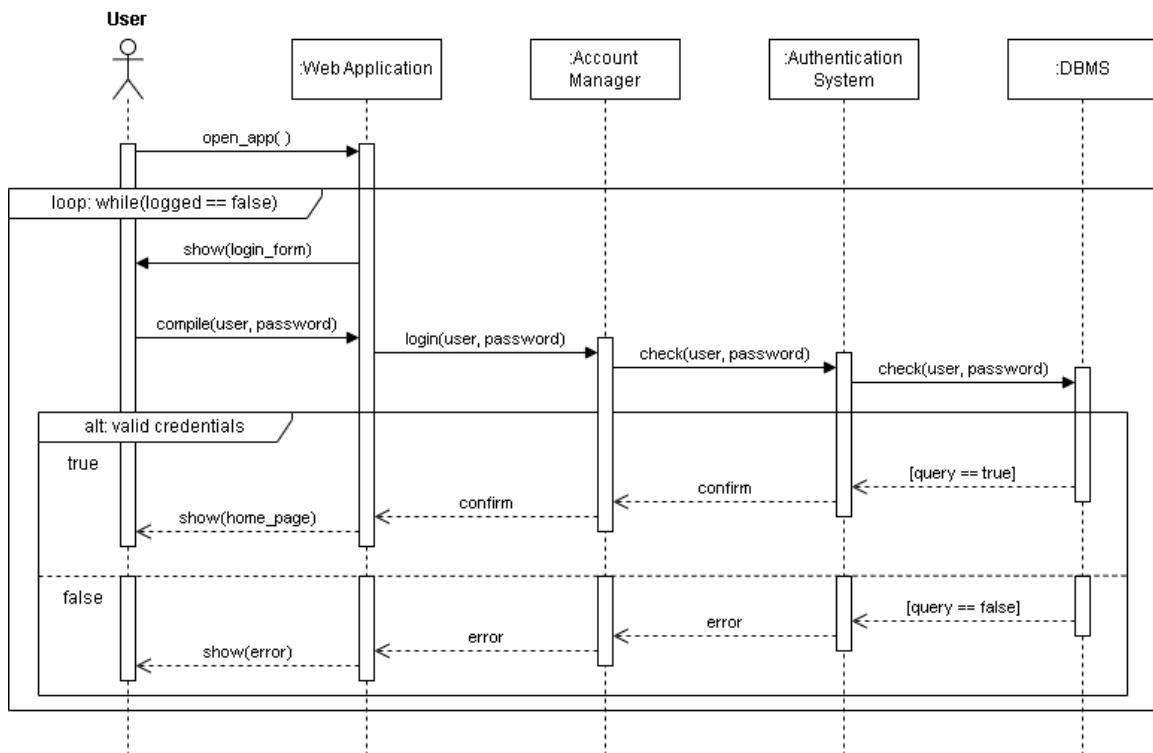


Figure 4: Login sequence diagram

### 2.4.2 Sign up

In the sign up process, the user interacts with the *Web Application* to create a new account. When entering the application, the user is presented with the login page, so he needs to request for the sign up page. After adding the required information, the data is sent to the *Account Manager* that checks if the credentials already exist in the data base. If the given information are correct, the user is shown the login page, otherwise he is shown a message error and can try to create his account again.

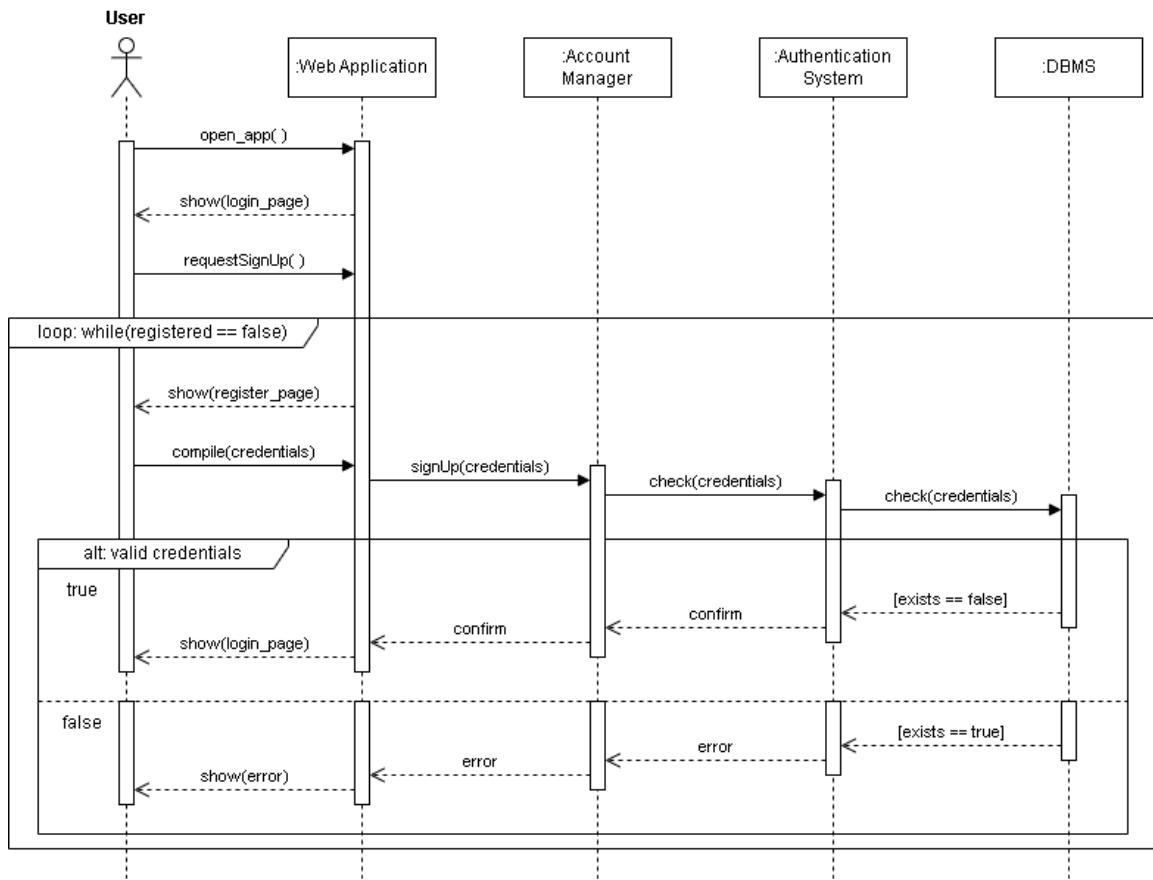


Figure 5: Sign up sequence diagram

### 2.4.3 Internship creation

In the internship creation process, a company interacts with the *Web Application* to post about a new internship. When entering all the correct information, the *Manage Internship* module interacts with the *Recommendation System* to identify all the student that are deemed compatible with the internship. Once the system has gotten the list, all the specified students are notified through the *Notification System*.

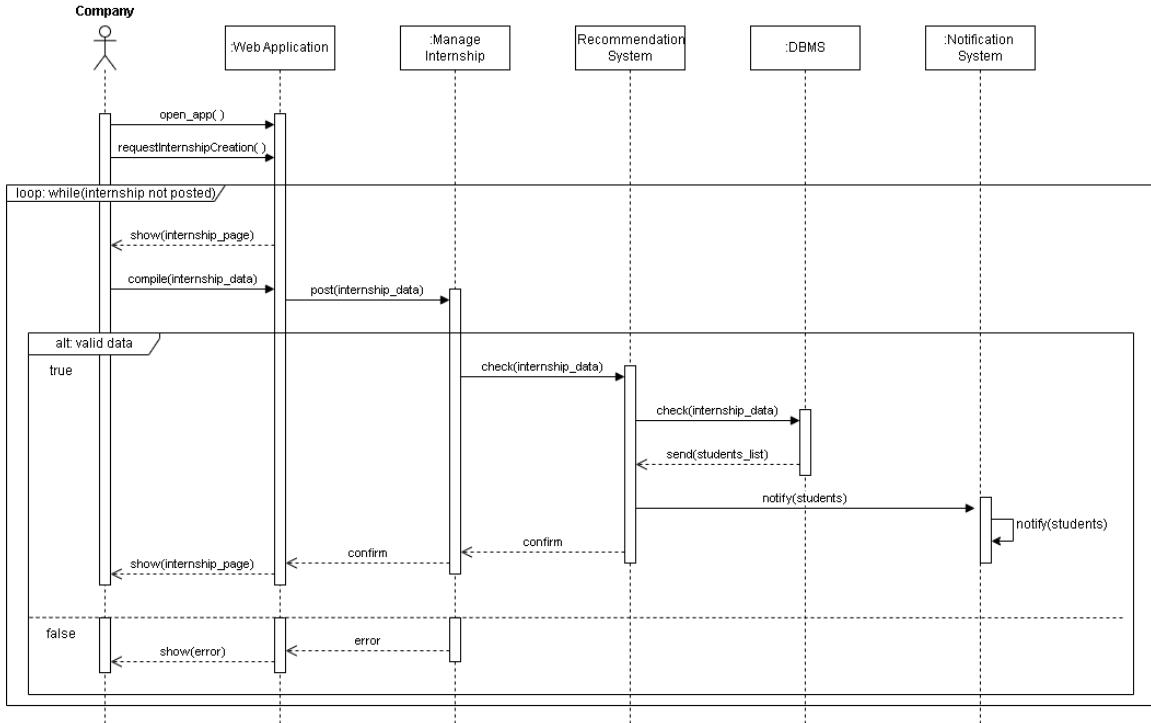


Figure 6: Internship creation sequence diagram

#### 2.4.4 Internship application

In the internship application process, the student interacts with the *Web Application* to search for a specific internship and apply for it. The student searches for specific keywords in the search bar, which activates the *Manage Internship module* and through the *DBMS* it retrieves the correct internships. The results are show on the *Web Application* and the student can apply for the ones that he prefers. When applying to the internship the *Manage Internship* module interacts with the *Notification System* to notify the company of the candidates.

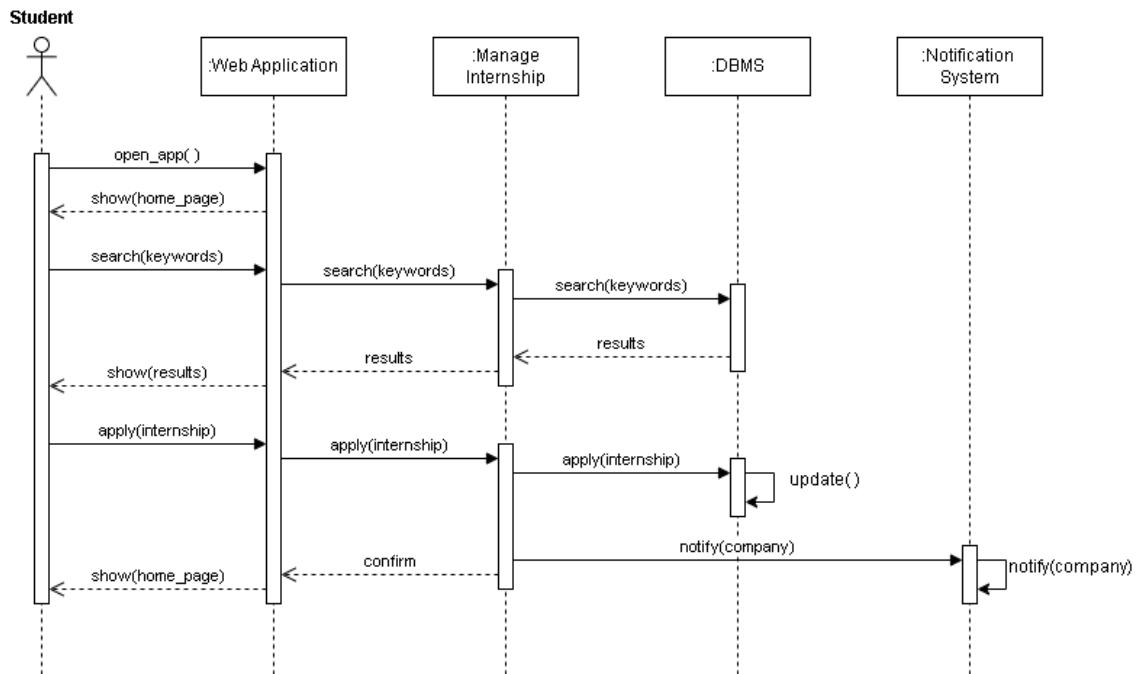


Figure 7: Internship application sequence diagram

#### 2.4.5 Form assignment for internship

In the form assignment process, a company interacts with the *Web Application* to assign a form to an already existing internship. In this case the *Web Application* needs to interact with the *Form Management* module to assign the form to the internship, and the *Notification System* to notify all the candidates of the form to fill. Notice that this flow happens only when assigning a valid form, trying to assign an empty form shows an error message.

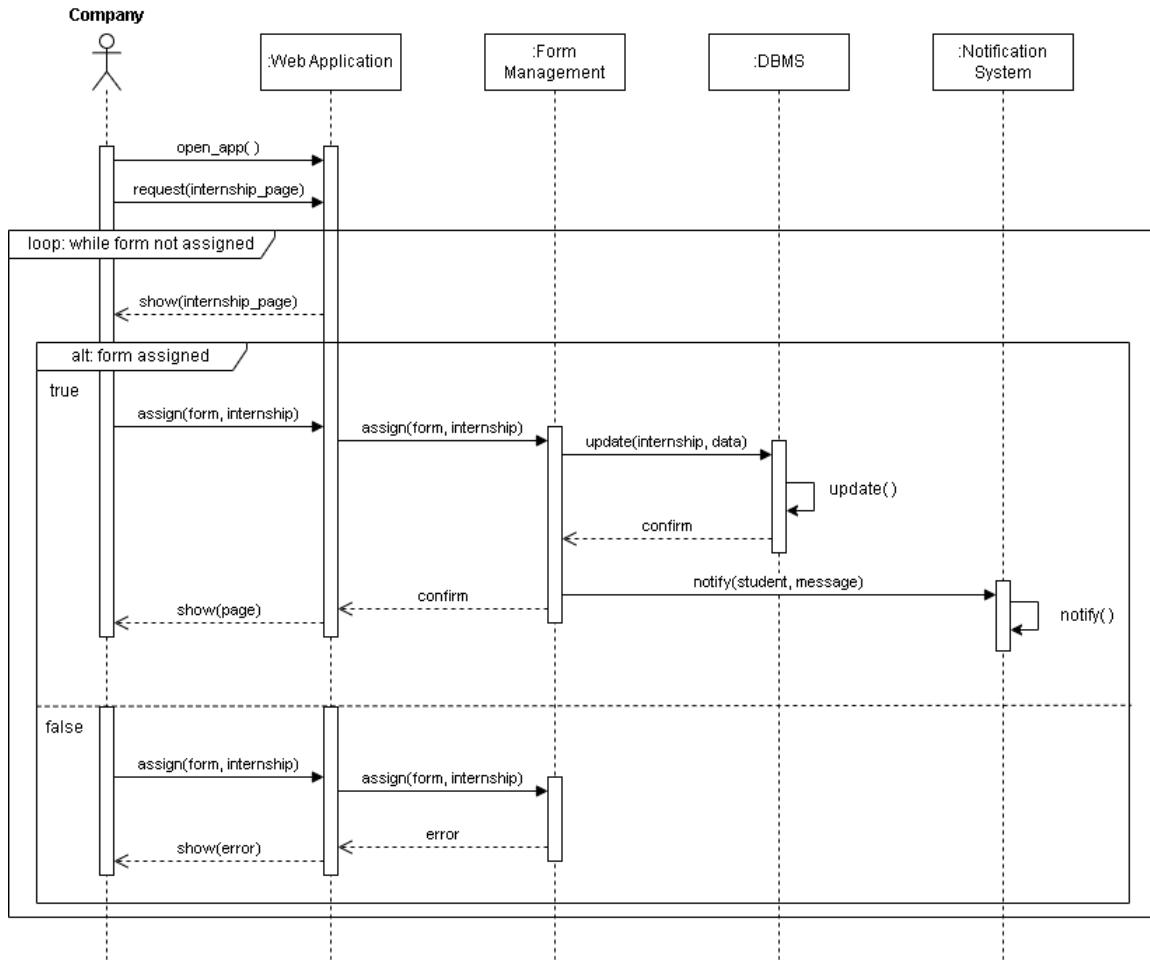


Figure 8: Form assignment sequence diagram

#### 2.4.6 Form compilation

In the form compilation process, the student interacts with the *Web Application* to fill and submit a form for an internship. In this case the *Web Application* needs to interact with the *Form Management* module to save the answer in the data base, and the *Notification System* to notify the company of the compiled forms they need to correct. Trying to submit an form with a missing answer will prevent the submission and show an error message instead.

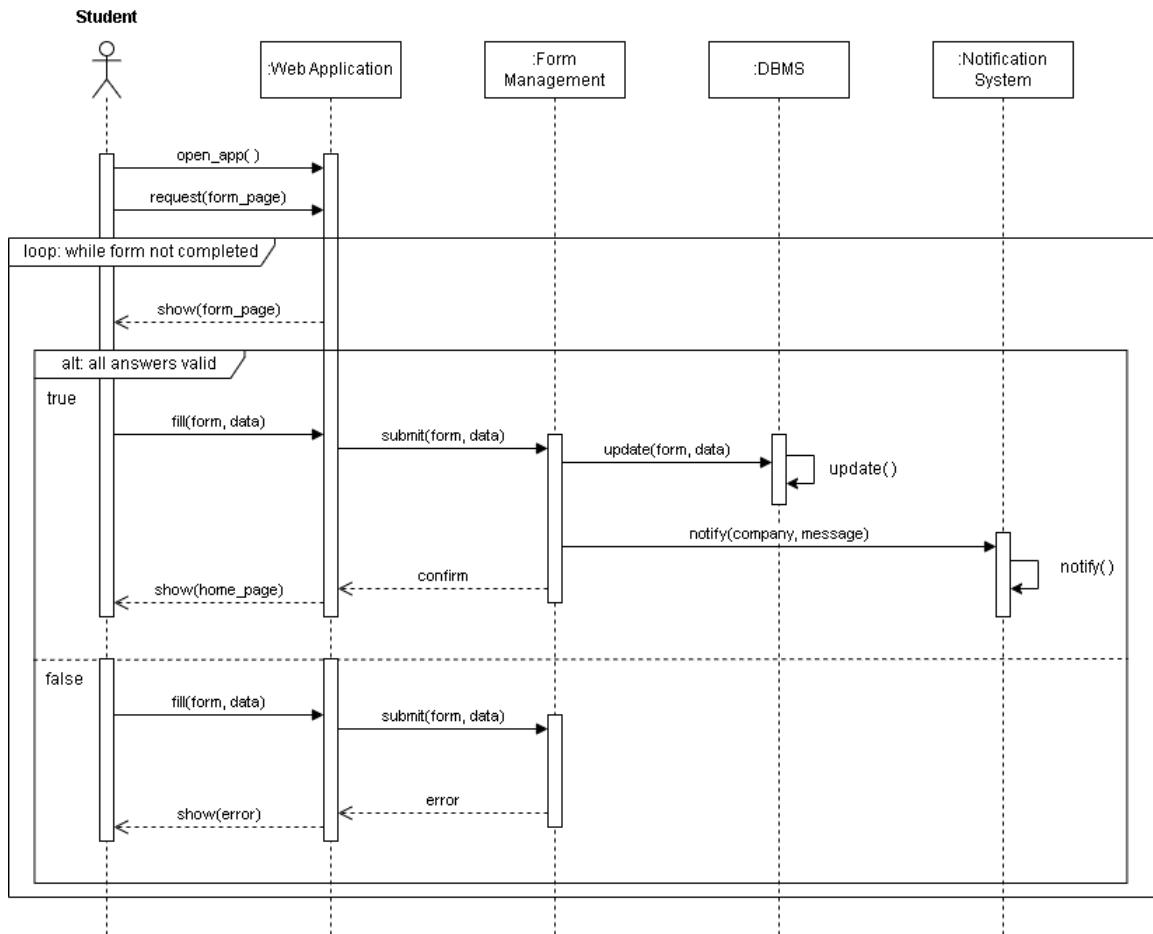


Figure 9: Form compilation sequence diagram

### 2.4.7 Form correction

In the form correction process a company interacts with the *Web Application* to correct a form submitted by a student. In this case the *Web Application* needs to interact with the *Form Management* module to save the score of the candidate for later use in the ranking system. The score can be submitted only when each individual question is given a valid score.

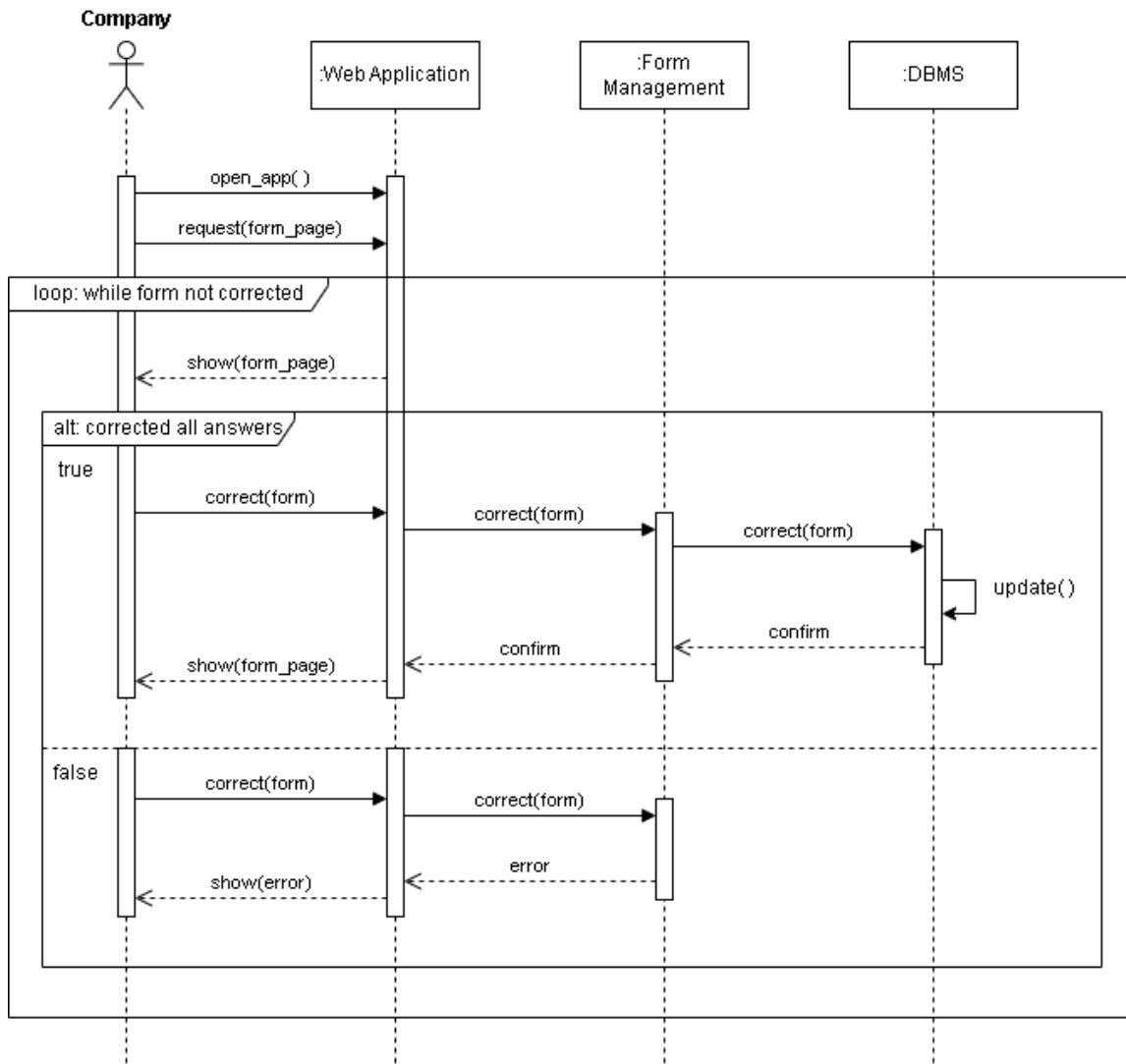


Figure 10: Form correction sequence diagram

#### 2.4.8 Student selection

In the student selection process a company interacts with the *Web Application* to select the candidates that can move to the next phase of the selection process or that were chosen to participate in the actual internship. In this case the *Web Application* needs to interact with the *Internship Selection* module and the *Ranking System* to retrieve from the data base the scores of all the candidates and format the in a readable way. The ranking is shown on the screen of the company application, and they can then choose a list of students that will advance to next phase. For this reason the *Internship Selection* module needs to also interact with the *Notification System*.

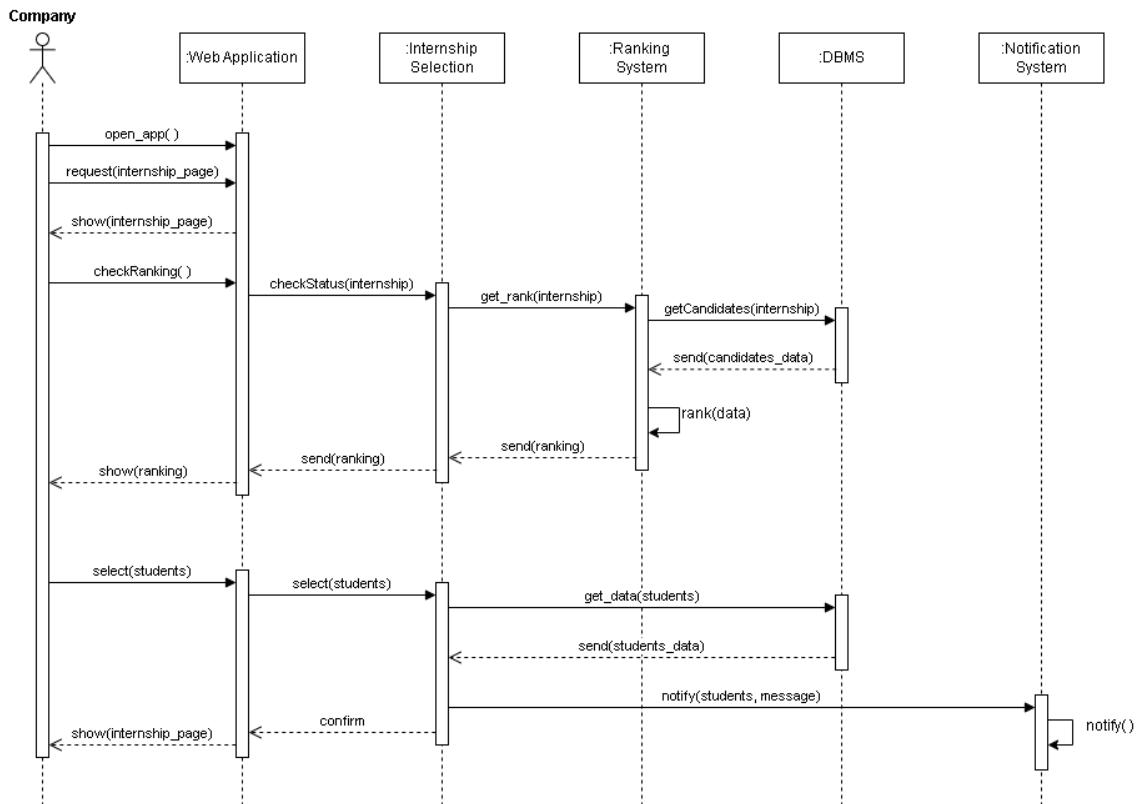


Figure 11: Student selection sequence diagram

## 2.5 Component interfaces

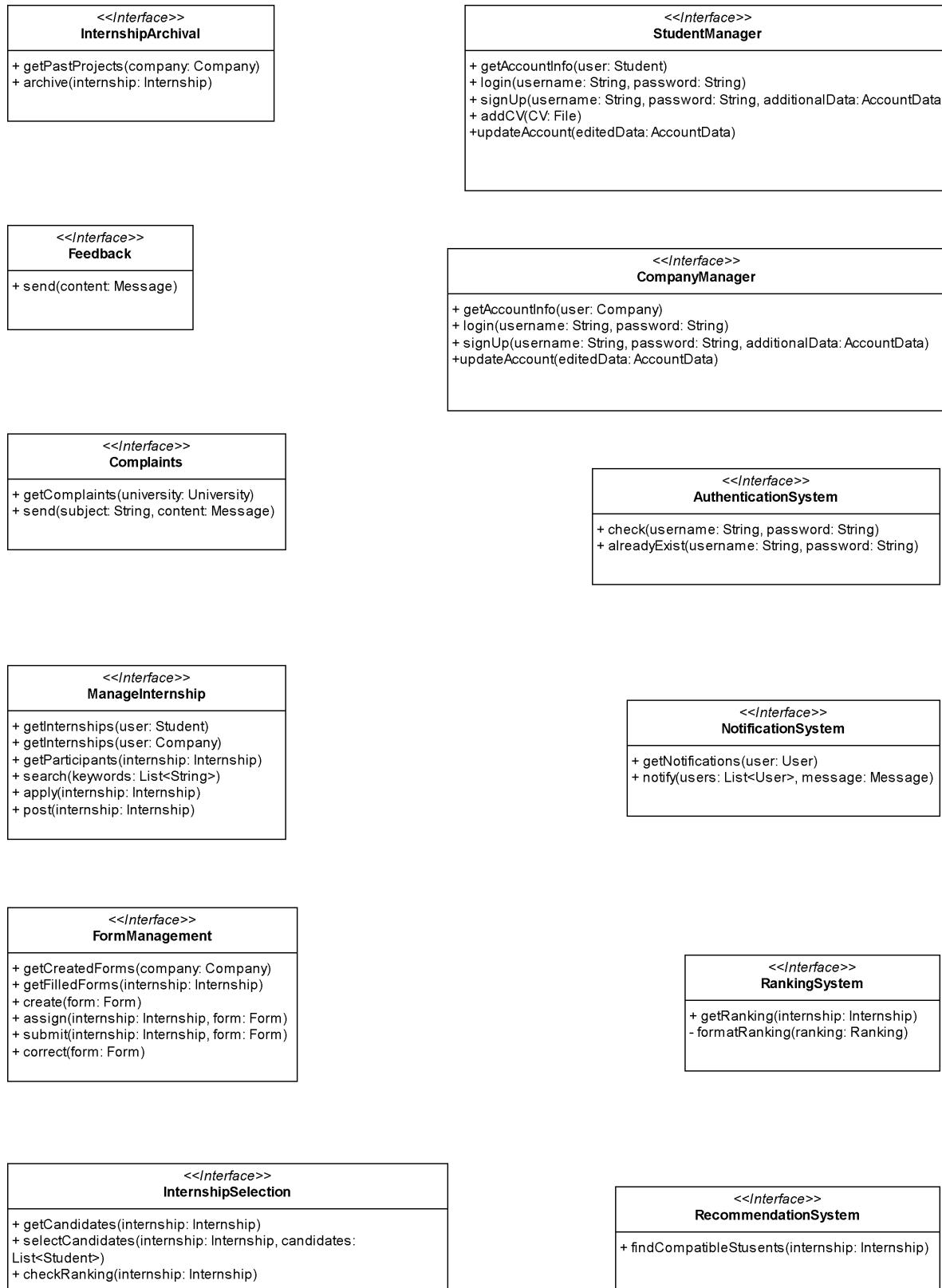


Figure 12: Component interfaces diagram

**2.6 Selected architectural styles and patterns**

**2.7 Other design decisions**

### 3 User interface design

The aim of this section is to present the prototypes of the most important pages of the application and explain the features by emulating scenarios of a user interacting with the system.

#### Login and Sign Up page

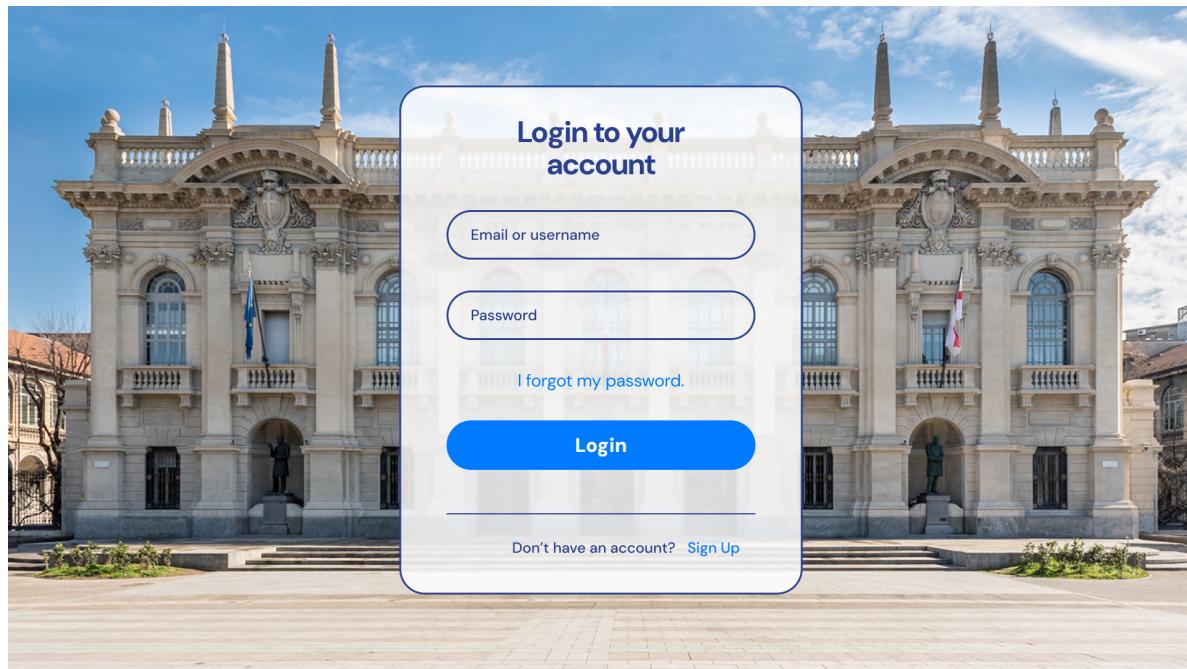


Figure 13: Login prototype page

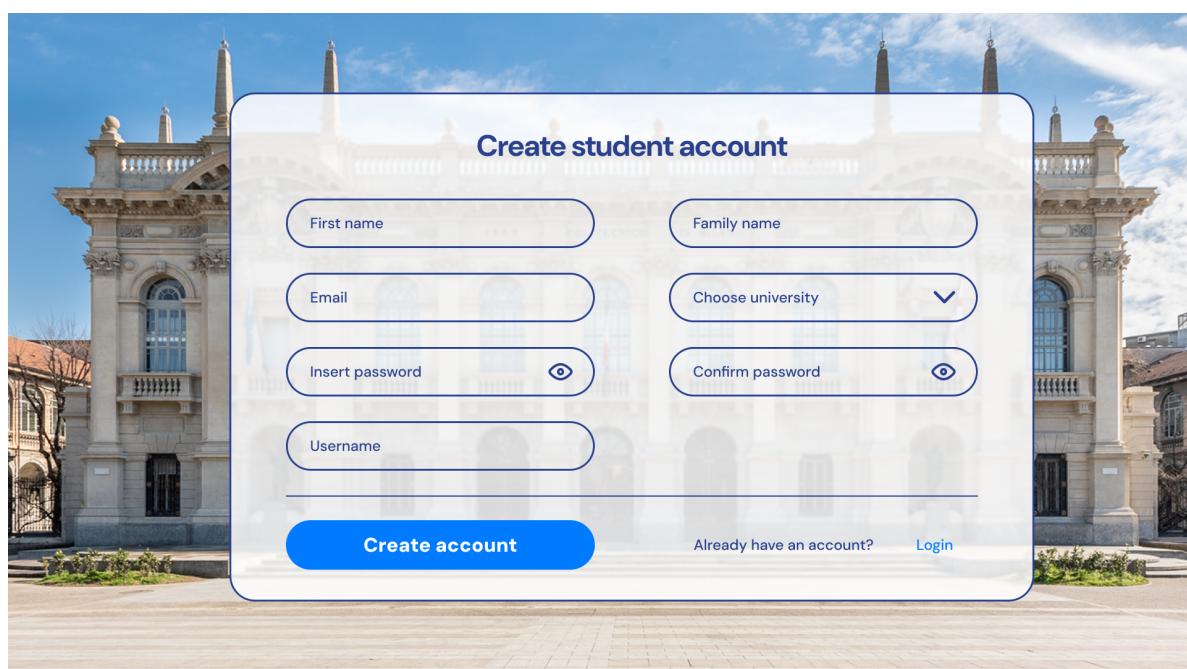


Figure 14: Sign up prototype page

The picture [13] shows the prototype of the login page, allowing to either login by entering the correct credentials, create an account in case the user doesn't have one yet, or reset the password in case it was forgotten.

The picture [14] shows the prototype of the sign up page for students, that allows students to create a new account by entering all the required information. In case the user already has an account there is a button that redirects to the login page. The case of a company creating an account is really similar to the one shown above with the simple difference that the user wont be required to choose a university.

## Profile page

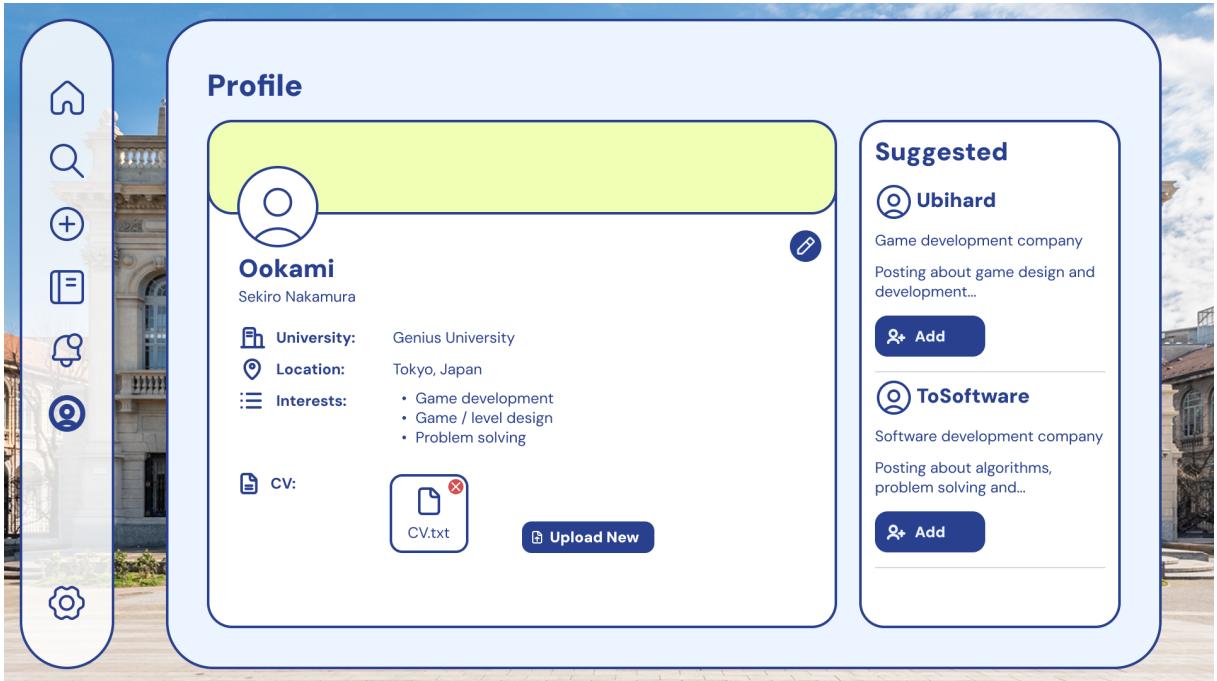


Figure 15: Student profile prototype page

The picture [15] shows the prototype of the profile page of a user logged in as a student. On the left, there is the section containing all the information provided by the user, and through the pencil button on the right, the user can modify the information. On the bottom there is the CV section. By clicking the file icon it opens the file allowing the user to check if the uploaded file is correct, otherwise the button on the right allows to upload a different file.

On the right side of the picture there is the "Suggested" area that show all the companies that might interest the user based on his interests.

## Notifications page

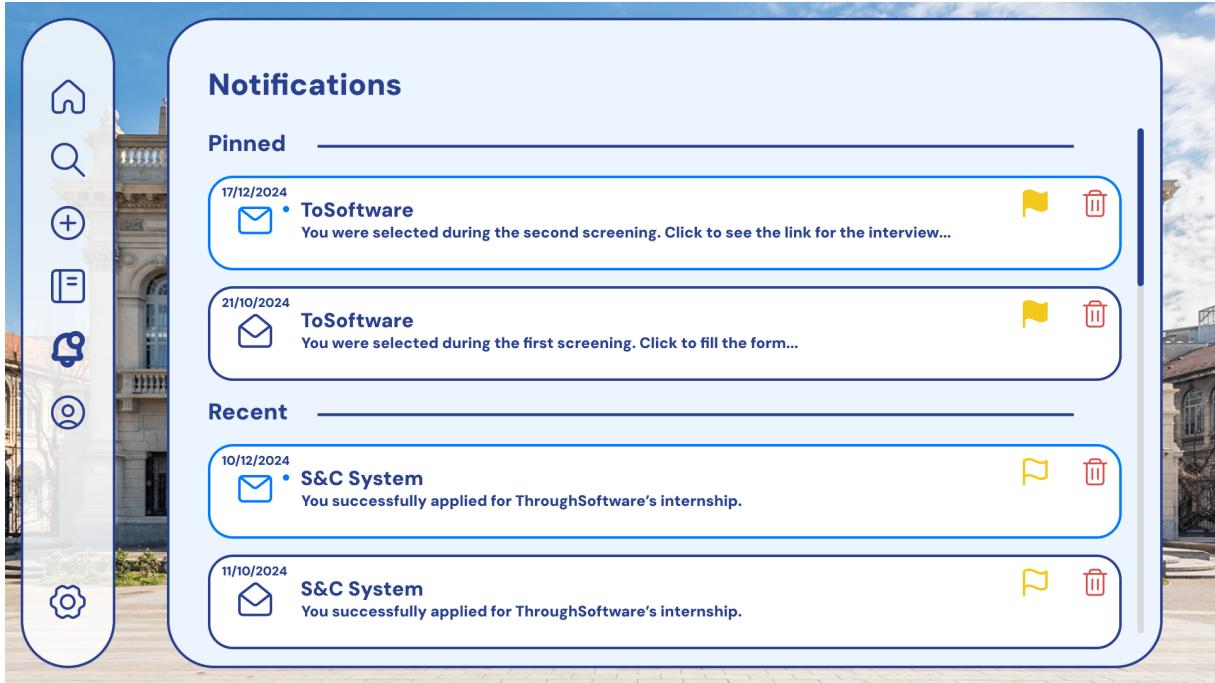


Figure 16: Notifications prototype page

The picture [16] shows the prototype of the notifications page and is shown to all three user categories. In general all notifications shown in the page render only a part of the whole message, and clicking on it allows the user to open a page that shows the whole text. In addition, notifications can be pinned to be always shown on top in their specific section by clicking on the yellow flag, or can be also permanently deleted by clicking on the trash can.

## Search page

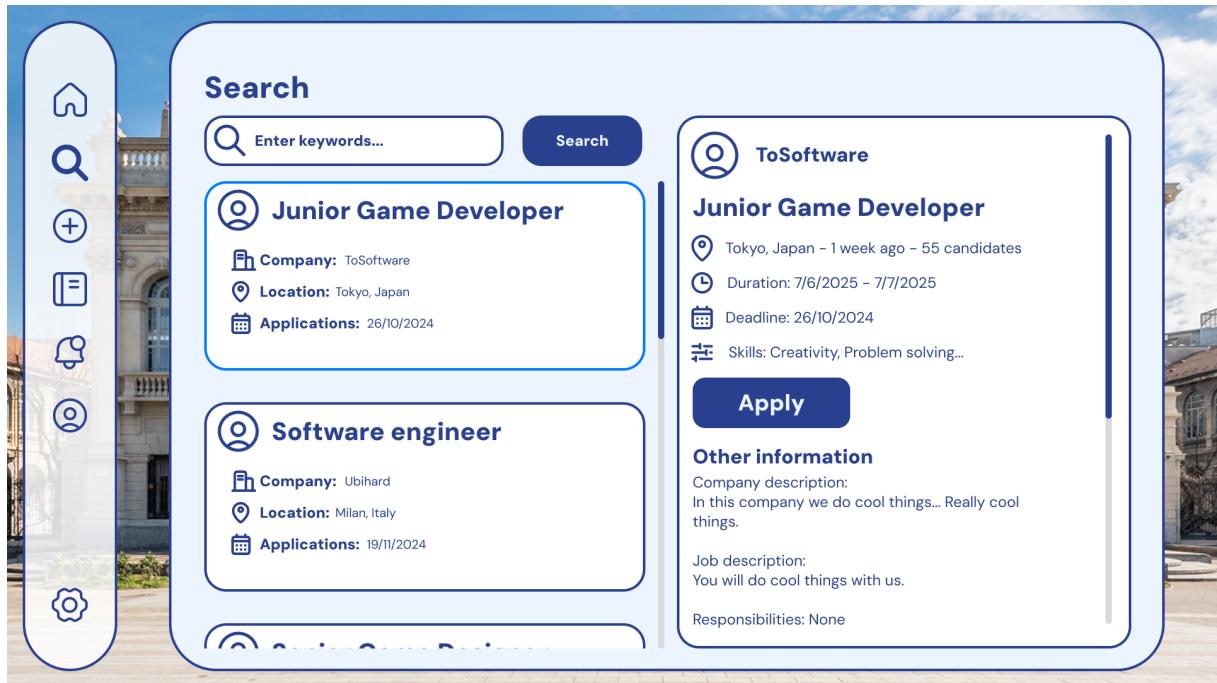


Figure 17: Search prototype page

The picture [17] shows the prototype of the search page, that allow students to proactively search for internship by entering keywords in the search bar.

Right after searching the keywords the internship found are displayed on the left side and only show the basic details. By clicking on the desired internship, all the additional information entered by the company are displayed on the right side. Only when visualizing the details of an internship it is possible to apply for said offer through a dedicated button.

## Form creation page

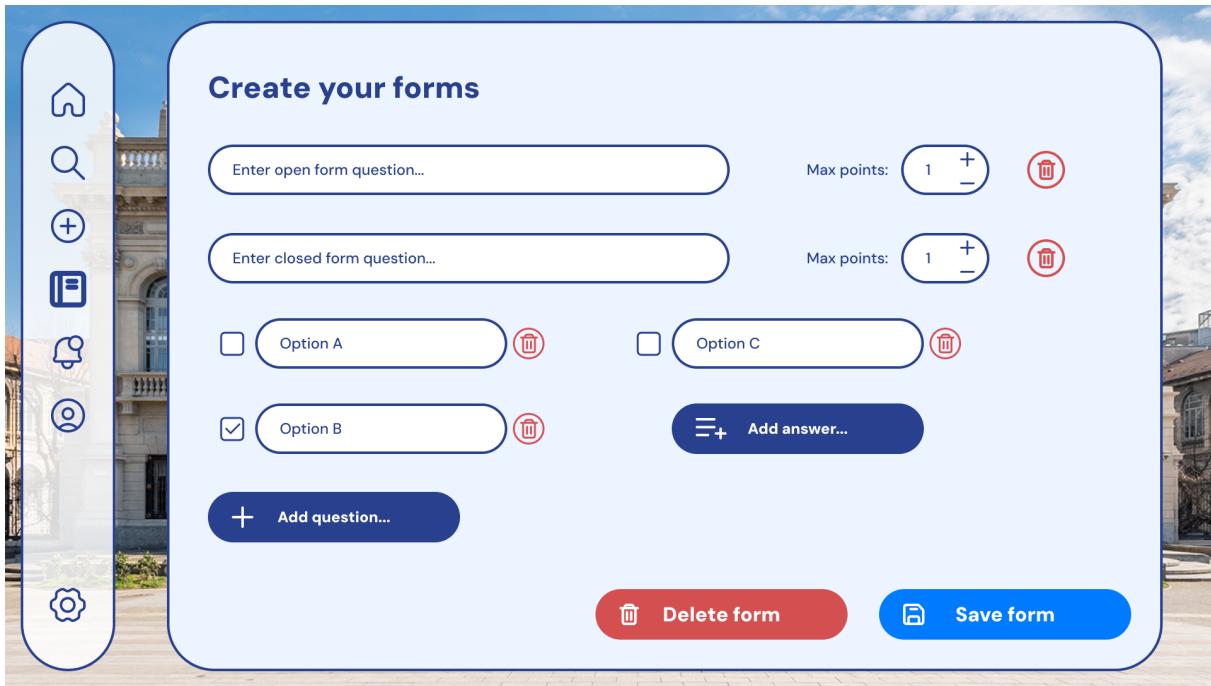


Figure 18: Form creation prototype page

The picture [18] shows the prototype of the form creation page that allows only companies to create custom forms for the candidates to fill during an application. In the form can be added two different types of questions: open and closed.

Open question only require the question text and the number of maximum point. The closed ones not only require the question text and the maximum points like the open ones, but they also need the options the candidates have to choose from. Each option can be either deleted, marked as correct with the checkbox on the left, or can have its text changed. In addition the company can choose to add more options to the closed question with the "*Add answer...*" button.

Lastly the company can decide to either add questions with the "*Add question...*" button or delete them with the trash can icon on the right.

Once the form has been completed it can be saved or deleted by using the dedicated buttons.

## Home page



Figure 19: Home prototype page

The picture [19] shows the prototype of the home page that allows students to see all the available internships posted by various companies.

The section on the left shows all the internships and by interacting with them the student can access a page that shows more details about the post and apply for the internship.

On the right there is a panel that shows the basic data regarding the user that is currently logged in.

## Internship creation page



Figure 20: Internship creation prototype page

The picture [20] shows the prototype of the page that allows companies to create and post about an internship.

In this page the company can add all the required information that allows the system to select the suitable students and notify them.

## **4 Requirement traceability**

## **5 Implementation, integration and test plan**

## 6 Effort spent

- Abdallah Alkhetiar

Chapter	Effort
1	0 h
2	0 h
3	0 h
4	0 h
5	0 h

- Daniel Bonardi

Chapter	Effort
1	0 h
2	0 h
3	0 h
4	0 h
5	0 h

## 7 References

### N-tier architectures

<https://dev.to/3bdelrahman>

<https://blog.nginx.org/blog>

<https://www.klipfolio.com>