

# **Software Engineering 2**

A.Y. 2024/2025



**POLITECNICO**  
**MILANO 1863**

## **Students&Companies**

### **Design Document**

Lo Presti Irene  
Lussana Matteo

Release date: 07/01/2025  
Version: 1.0



# Table Of Contents

<b>Table Of Contents</b>	<b>3</b>
<b>1 Introduction</b>	<b>5</b>
1.1 Purpose . . . . .	5
1.2 Scope . . . . .	5
1.3 Definitions, Acronyms, Abbreviations . . . . .	6
1.3.1 Definitions . . . . .	6
1.3.2 Acronyms . . . . .	6
1.3.3 Abbreviations . . . . .	7
1.4 Revision history . . . . .	7
1.5 Reference Documents . . . . .	7
1.6 Document Structure . . . . .	7
<b>2 Architectural Design</b>	<b>9</b>
2.1 Overview . . . . .	9
2.2 Component view . . . . .	10
2.3 Deployment view . . . . .	12
2.3.1 Architecture nodes . . . . .	13
2.3.2 Components and services . . . . .	14
2.4 Runtime view . . . . .	20
2.5 Component interfaces . . . . .	36
2.6 Selected architectural styles and patterns . . . . .	39
2.6.1 Micro-service Architecture . . . . .	40
2.6.2 3-Tier Architecture . . . . .	40
2.6.3 Client-Server Model . . . . .	40
2.7 Other design decisions . . . . .	40
2.7.1 Scalability . . . . .	40
2.7.2 Ease of Deployment . . . . .	41
<b>3 User Interface Design</b>	<b>43</b>
3.1 Overview . . . . .	43
3.2 Mock-up pages . . . . .	44
3.2.1 Login Page . . . . .	44
3.2.2 Sign Up Page . . . . .	45
3.2.3 Home Page . . . . .	46
3.2.4 Student Profile Page . . . . .	47
3.2.5 Company Profile Page . . . . .	49

3.2.6	Company's ADVs Page . . . . .	50
3.2.7	Company's ADV Page . . . . .	51
3.2.8	ADV Page . . . . .	52
3.2.9	Search Page . . . . .	53
3.2.10	Monitor Page . . . . .	54
3.2.11	Inbox Page . . . . .	55
<b>4</b>	<b>Requirements Traceability</b>	<b>57</b>
<b>5</b>	<b>Implementation, Integration and Test Plan</b>	<b>59</b>
5.1	Overview and Implementation Plan . . . . .	59
5.2	Integration Strategy . . . . .	59
5.3	System Testing Strategy . . . . .	64
<b>6</b>	<b>Effort Spent</b>	<b>65</b>
<b>7</b>	<b>References</b>	<b>67</b>
7.1	References . . . . .	67
7.2	Used Tools . . . . .	67
	<b>List of Figures</b>	<b>69</b>

# 1 | Introduction

## 1.1. Purpose

The **S&C system** is a platform designed to connect university students seeking **internships** with companies offering internship opportunities. The platform helps students and companies connect and use tools designed for their needs.

**Students** are able to log in to the platform via their university credentials. Once they are logged in, they can **personalise their profile** by uploading their CV and inserting their skills, experiences and attitudes. They can **search for an internship** by using the search bar, or by waiting for personalised recommendations sent by the system. If they find a suitable internship, they can **contact** the company. They might also be contacted by a company, so they can accept or decline the offer. If a company accepts a student's application, the Selection Process begins, which involves an interview.

**Companies** can publish **internship advertisements**, look actively for suitable candidates via the search bar or use the recommendations sent by the system. They can, as well, **contact students**, or they can accept or decline internship requests. They can **setup an interview** with possible candidates, and create customized forms for use during interviews linked to specific internship advertisements.

Both students and companies can monitor the interactions regarding internships by the **Monitoring Section**. They can also provide **feedback** on their experiences with one another, whether after an interview or during an internship. These comments are published on the user's profile. The **Recommendation Process** is based on an analysis by the system that considers the criteria used in the search bar by the student, the candidate profile inserted in the internship advertisement, and the feedback provided by both parties.

The target of this document is the development team. It could also be used as a baseline for implementation activities and for integration and Quality Assurance. It refines the plan and previous estimations done in the RASD document.

## 1.2. Scope

In this document, an in-depth description of the development of the system is done. A combination of three design architectures: Micro-Service, 3-Tier and Client-Server is implemented (to read it more in details see section 2.6)

## 1.3. Definitions, Acronyms, Abbreviations

### 1.3.1. Definitions

**Single Sign-on** is an authentication method that allows users to log into multiple applications and websites using a single set of credentials. In the S&C system, Students use their university credentials to log in.

**Application Programming Interface** is a collection of functions and methods that allow the development of applications to interact with the features or data of an operating system, software, or other services.

**RESTful APIs**: specific standardised architectural style for APIs.

A **Database Management System (DBMS)** is a software system for creating and managing databases. A DBMS enables end users to create, protect, read, update and delete data in a database. It also manages security, data integrity and concurrency for databases. **Recommendation** is the process performed by the system that recommends to students potential internship positions that they might be interested in and to companies potential candidates.

The **Chamber of Commerce Certificate** is an official document provided by the Chamber of Commerce that holds a legal certification value. It confirms the company's registration in the Business Register and includes its name, legal structure, and registration details.

The **Revenue Agency** is a public entity that is not focused on profit but is dedicated to ensuring optimal tax compliance. Its primary responsibilities include collecting tax revenues, offering services and support to taxpayers, and conducting assessments and inspections to combat tax evasion.

### 1.3.2. Acronyms

DD: Design Document

ADV: advertisement

SSO: Single Sign-on

API: Application Programming Interface

REST: Representational State Transfer

HTTPS: HyperText Transfer Protocol Secure

TLS: Transport Layer Security

DBMS: Database Management System DB: Database

DMZ: Demilitarized Zone

OS: Operative System

SD: Sequence Diagram

UI: User Interface

VAT number: value-added tax identification number (*partita IVA*)

CEO: Chief Executive Officer

HR: Human Resources

CV: Curriculum Vitae

### 1.3.3. Abbreviations

R#: Functional Requirement

## 1.4. Revision history

Version 1.0: 07/01/2025

## 1.5. Reference Documents

- Specification document: Assignment RDD AY 2024-2025.
- Slides of the course "Software Engineering 2" held at Politecnico di Milano by Professor Rossi (a.y. 2024-25).
- "Fondamenti di Sistemi Informativi per il Settore dell'Informazione" (Cappiello, Fugini, Grefen, Pernici, Plebani, Vitali): book of the course "Sistemi Informativi" held at Politecnico di Milano by Professor Fugini (a.y. 2022-23), used to implement the Deployment View.
- Abbreviations section are taken from researches done on the Internet.

## 1.6. Document Structure

1. Introduction: analysis of the domain and product, summary of key architectural styles and decisions, and an overview of the contents and structure of this DD.
2. Architectural Design: general overview of the architecture, detailed explanation of the main interfaces, component diagrams, deployment diagrams, and dynamic of interaction (sequence diagrams).
3. User Interface Design: overview of user interfaces accompanied by mock-ups.
4. Requirements Traceability: mapping between requirements and design elements.
5. Implementation, Integration and Test Plan: sequence of subsystem and components implementation, along with the integration and test plan.
6. Effort Spent: information on the time spent drafting the document divided per group members.
7. References: this section lists the documents used to draft the project.

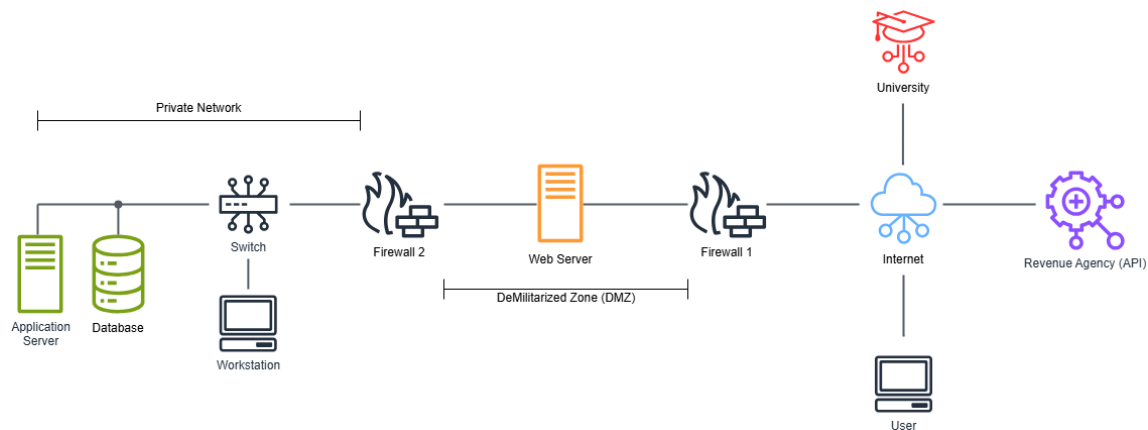




# 2 | Architectural Design

## 2.1. Overview

The architecture of the S&C system is presented in this section.



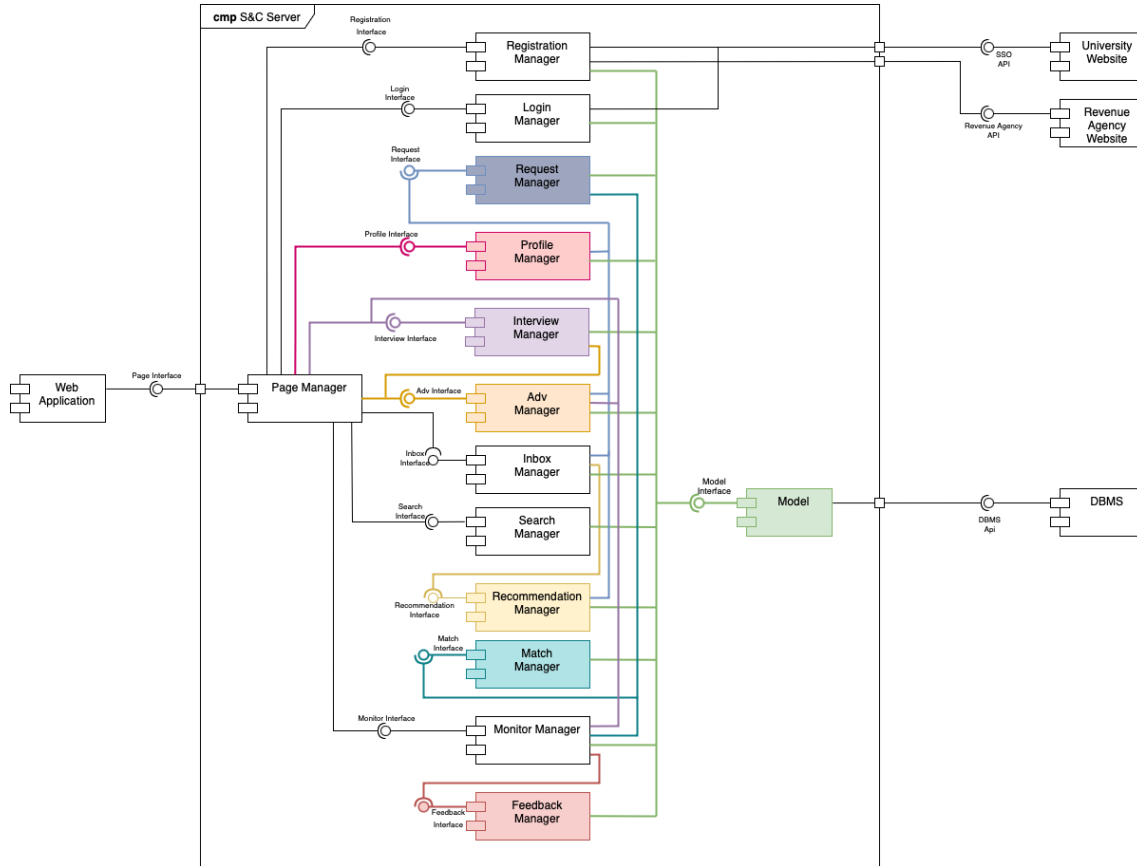
**Figure 2.1:** Overview of the S&C system

This diagram illustrates the platform's entire architecture, highlighting how the components are interconnected and how security is ensured. The **Private Network** is the backbone of the platform. It hosts the **Application Server**, which runs the system's logic, and the **Database** where all the data is stored. These components are connected via a switch to a Workstation used for internal administration, such as company registrations.

The **Demilitarized Zone (DMZ)** contains the Web Server, which handles user requests. The DMZ is placed between the two firewalls. This is standard practice to ensure that external users can only access the components within the DMZ (in this case, the Web Server) while maintaining the security of components inside the Private Network.

On the **Internet**-facing side, the platform interacts with users and integrates with external services, such as the university SSO systems and the Revenue Agency APIs.

## 2.2. Component view



**Figure 2.2:** Component view: Low Level Diagram

The figure 2.2 represent the S&C architecture the Server's components:

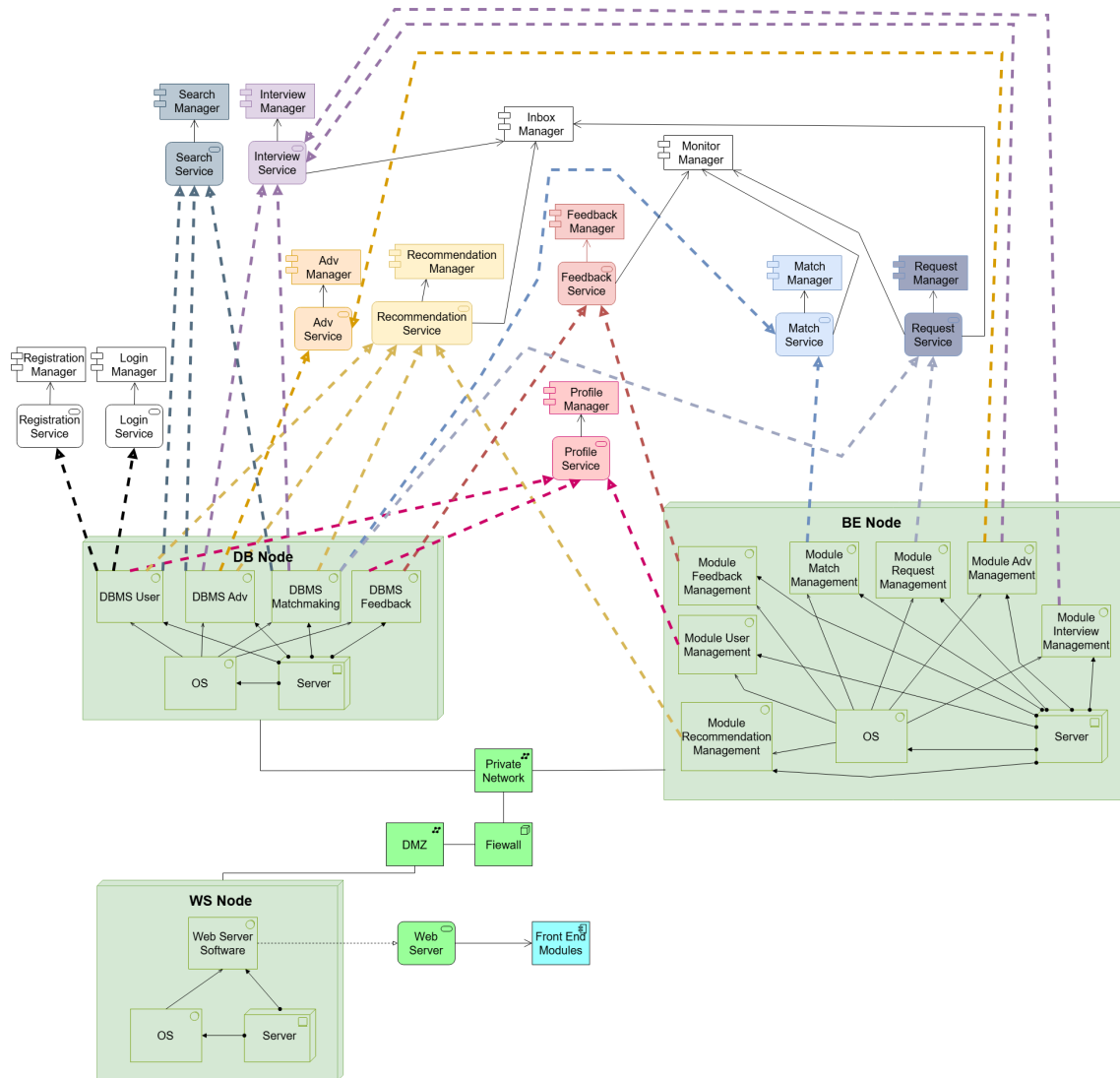
- **Page Manager:** This component directs the User's requests through the Page Interface to the correct component. It acts like a switch for whole the server.
- **Registration Manager:** This component handles new User registration, receiving the request from the Page Manager, and communicating with the university SSO API, if the user is a Student or with the Revenue Agency API, if the user is a Company. This component communicates also with the Users DBMS saving information about the new User.
- **Login Manager:** This component handles the login of an existing User, it communicates with the Users DBMS in order to check the validity of the credentials used by the User.
- **Request Manager:** This component is responsible for managing the Request of a Company to contact a Student and vice versa. If a Request is accepted the creation of a Match will be triggered.
- **Profile Manager:** This component loads the Profile Page of both Students and Companies by communicating with the Users DBMS. This component, also, communicates

with Request Manager in order to create a Request in case a visitor of the profile wants to contact the owner.

- **Interview Manager:** This component handles the request from the Adv Manager regarding the creation of a new Interview Form by the Users (C) and manages the interview requests.
- **Adv Manager:** This component handles the requests regarding the Adv Page: it loads the page by communicating with the DBMS, uses Request's interface to allow User (S) to send Requests to the Company proposing the Adv, and allow User (C) to create the Interview form thanks to Interview Manager's interface.
- **Inbox Manager:** This component handles the requests by the Page Manager regarding the Inbox Page that is loaded communicating with the DBMS. Also, it communicates with Request and Recommendation Manager to allow User to answer an internship request and contact the Users suggested in the Recommendation.
- **Search Manager:** This component handles the Page Manager's request about Search Page, and collects the results of the research made by the User by communicating with the DBMS.
- **Recommendation Manager:** This component handles the creation of recommendations using the data in the DBMS, uses the interface of Request Manager to let Users contact the suggestions made in the recommendation, and supplies to the Inbox Manager an interface to load the recommendation page.
- **Match Manager:** This component handles the request from the Monitor Manager regarding the information about a Match state by communicating with the DBMS.
- **Monitor Manager:** This component handles the requests by the Page Manager regarding the Monitor Page that is loaded communicating with the DBMS, Match Manager and Interview Manager. Also, it communicates with Feedback Manager in order to allow User to leave a feedback in certain states of the Match.
- **Feedback Manager:** This component handles the request from the Monitor Manager regarding the creation of a new Feedback from the Users.
- **Model:** This component handles the communication with the DBMS to fulfill all the requests of the other components about saving, deleting, getting or modifying data.

## 2.3. Deployment view

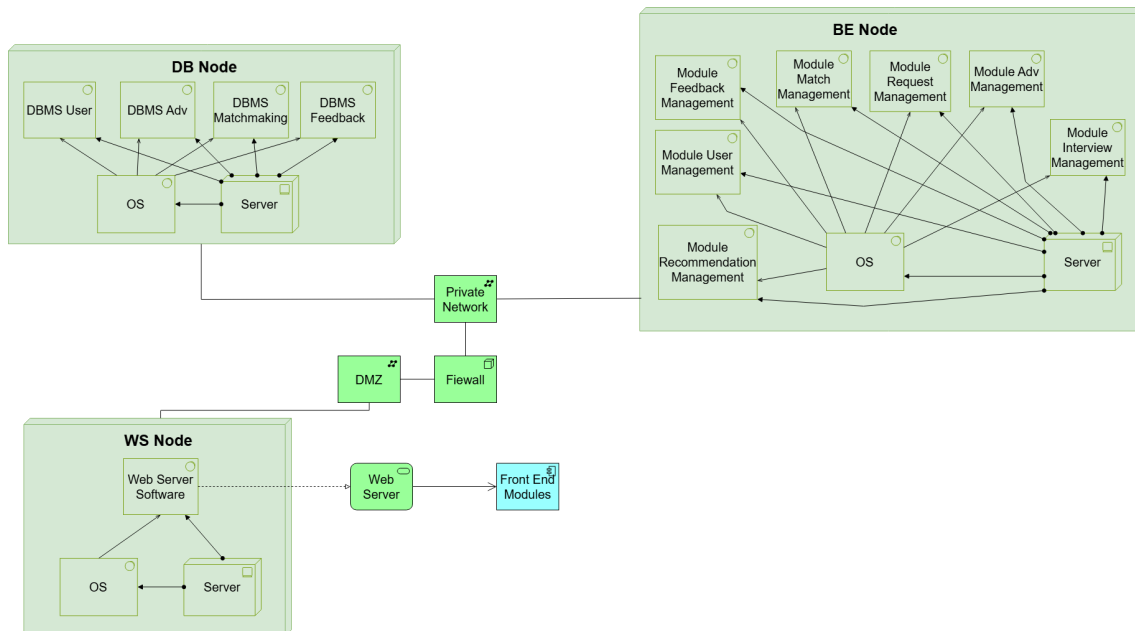
Here follows the **Deployment Diagram** of the S&C system.



**Figure 2.3:** Deployment Diagram

This is an overview of the whole diagram, each node and module is going to be explained in detail with a zoomed image.

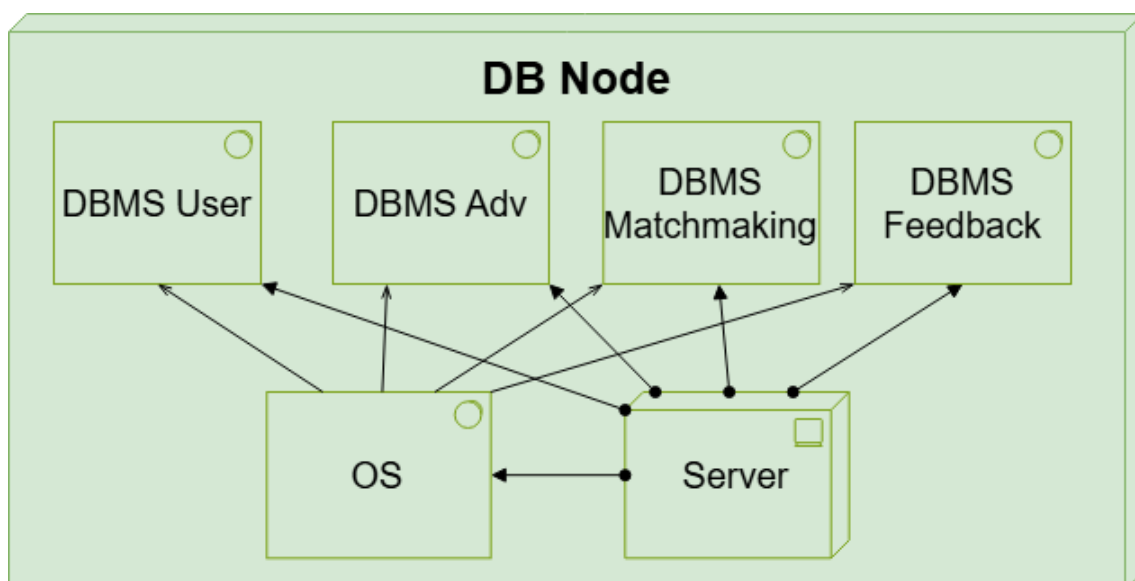
### 2.3.1. Architecture nodes



**Figure 2.4:** Architecture nodes in Deployment Diagram

This is the architecture shown in the Overview Chapter, with the modules and components shown.

### Database Node



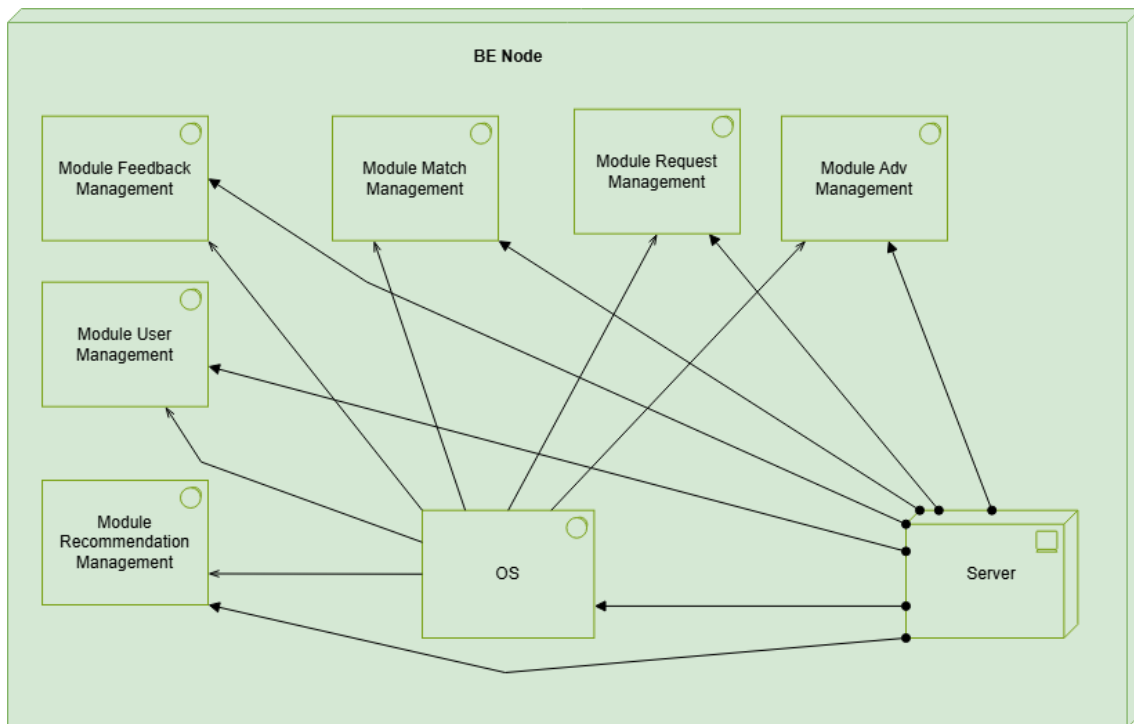
**Figure 2.5:** Database Node in Deployment Diagram

This node is designed to centralise database management, it divides responsibilities across modules:

- **User:** the DBs contain the information of both kinds of users, students, and companies.
- **Adv:** it organises the data of each internship advertisement, is related to the Company DB via external key.
- **Matchmaking:** it collects data about the matchmaking process: request messages, interview requests, matches' status.
- **Feedback:** feedback of companies and students is collected and connected to the respective user via external key.

The Operative System (OS) provides the base system for managing hardware and software resources on the DB Node and ensures the proper execution of database management systems and server processes. The Server facilitates communication between the DB Node and components.

## Back-End Node

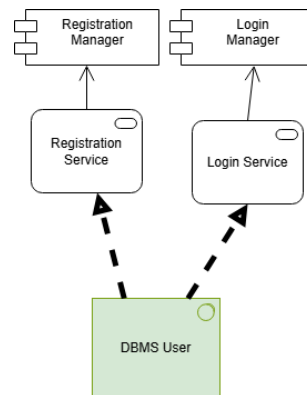


**Figure 2.6:** Back-End Node in Deployment Diagram

This node contains the business logic of the system, every module present runs specific functionalities related to the components, each module is seen in detail with the respective components and services.

### 2.3.2. Components and services

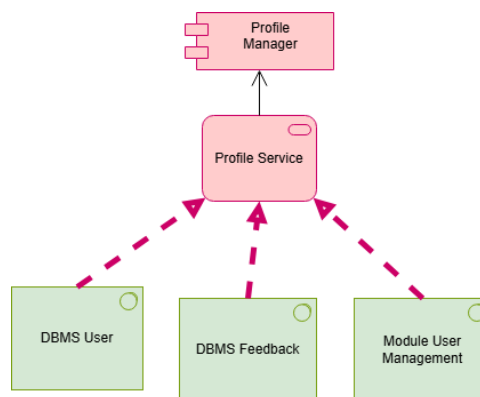
## Sign up and Login



**Figure 2.7:** Deployment Diagram: Login and Sign up

Users sign up and logins are managed through database's query, so their services relies entirely on the User DBMS.

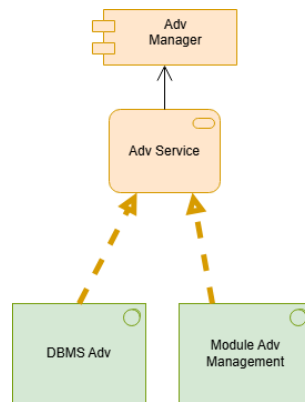
## Profile Management



**Figure 2.8:** Deployment Diagram: Profile Management

Users' data is stored inside the DB so profiles are managed through queries to the DB and the functions inside the **Module Profile Manager**.

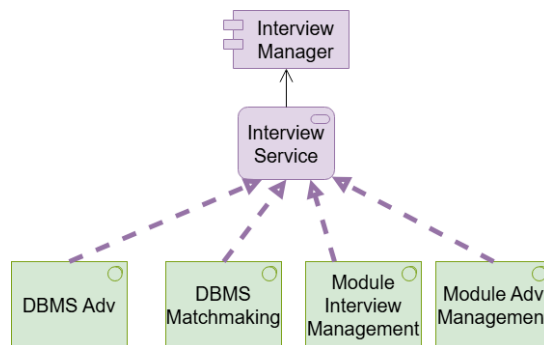
## ADV Management



**Figure 2.9:** Deployment Diagram: ADV Management

ADV's data is stored inside the DB, while the functions to manage them are found in the **Manage ADV Module**.

## Interview Management

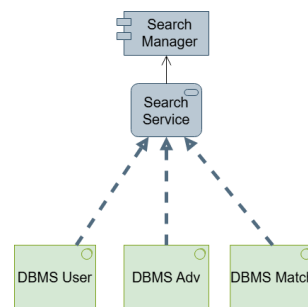


**Figure 2.10:** Deployment Diagram: Interview Management

Interview Manager handles interview forms and interview requests. Interview forms are stored in the **Adv DBMS** and managed by both **Module Adv Management** and **Module Interview Manager**, while interview requests are stored inside the **Matchmaking DBMS** and managed by the **Module Interview Management**.



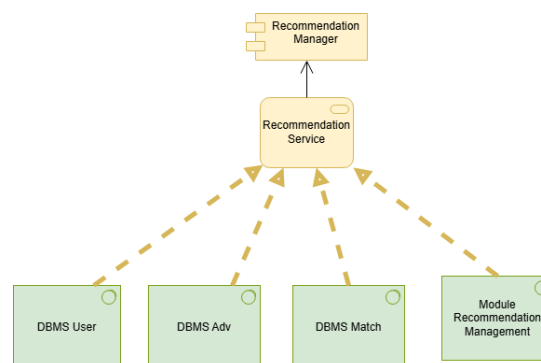
## Search Management



**Figure 2.11:** Deployment Diagram: Search Management

User searches rely on the Users DB and on the ADV, they are managed through queries. Also, the fields inserted by the students are stored inside the correspondent DB (in the User DBMS) in order to make proper recommendations. The search results cannot be already matched with the user searching, so the results are checked with the matches inside the DB.

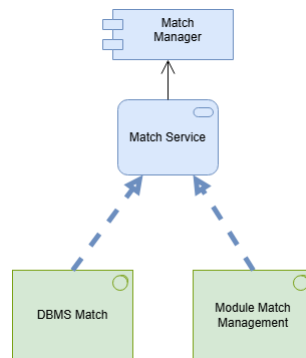
## Recommendation Management



**Figure 2.12:** Deployment Diagram: Recommendation Management

Recommendations are based on Users and ADVs, so they are based on the data in the DBs and managed through the functions in the **Recommendation Manager**. They can also queries the Matchmaking DBMS to make sure that the recommended profiles are not already matched.

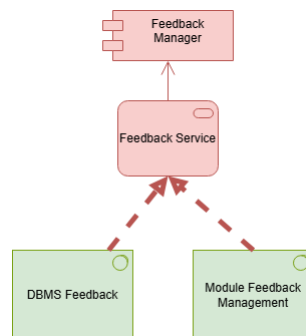
## Match Management



**Figure 2.13:** Deployment Diagram: Match Management

Matches are stored in the DB and are managed through the functions of **Module Match Management**.

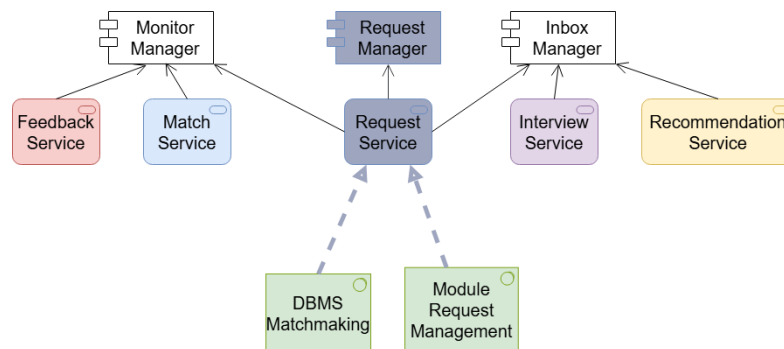
## Feedback Management



**Figure 2.14:** Deployment Diagram: Feedback Management

Feedbacks are stored inside the DB and are managed through the **Module Feedback Management**.

## Monitor and Inbox Management



**Figure 2.15:** Deployment Diagram: Monitor and Inbox Management

The Monitor Section shows the user's matches, and it is where users can leave feedbacks, so it is based on the **Match Service**, the **Request Service**, and the **Feedback Service**. The Inbox Section shows match and interview requests, and recommendations sent to the user, so it is based on the **Request Service** that is managed through the **Module Request Manager**, the **Recommendation Service**, and the **Interview Service**.

## 2.4. Runtime view

### Student Sign up

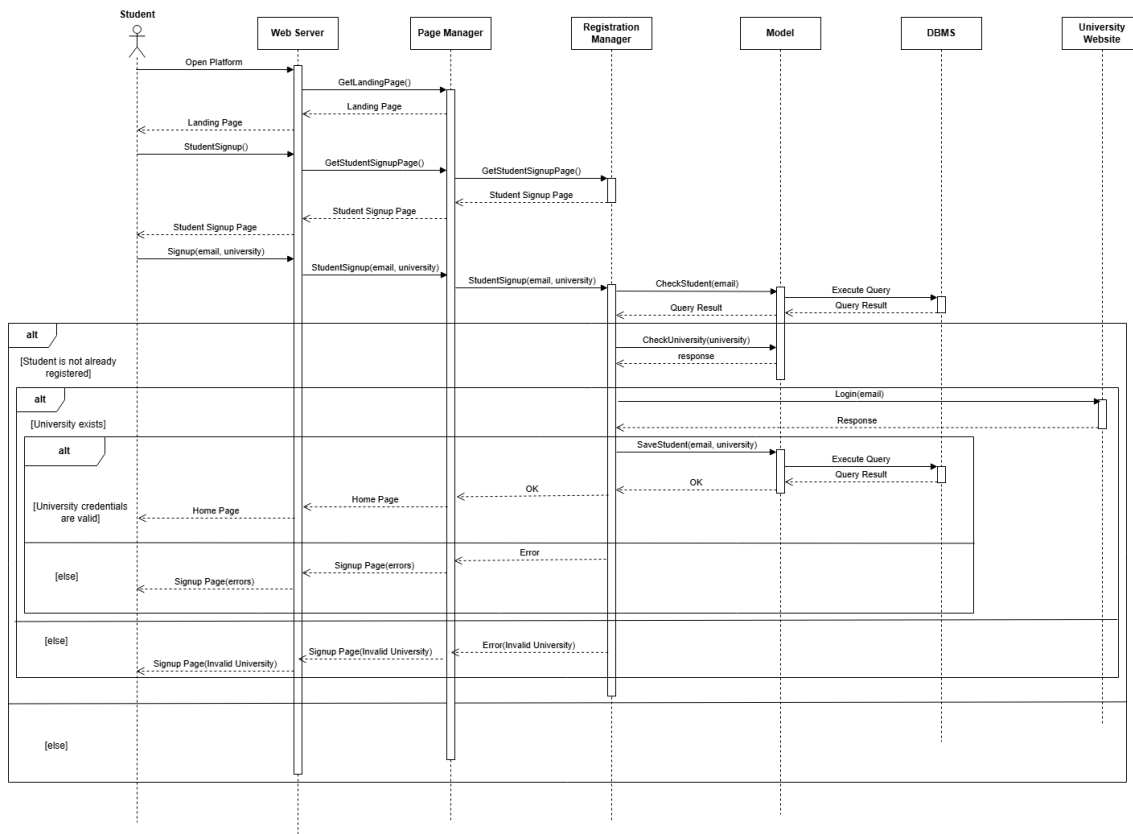
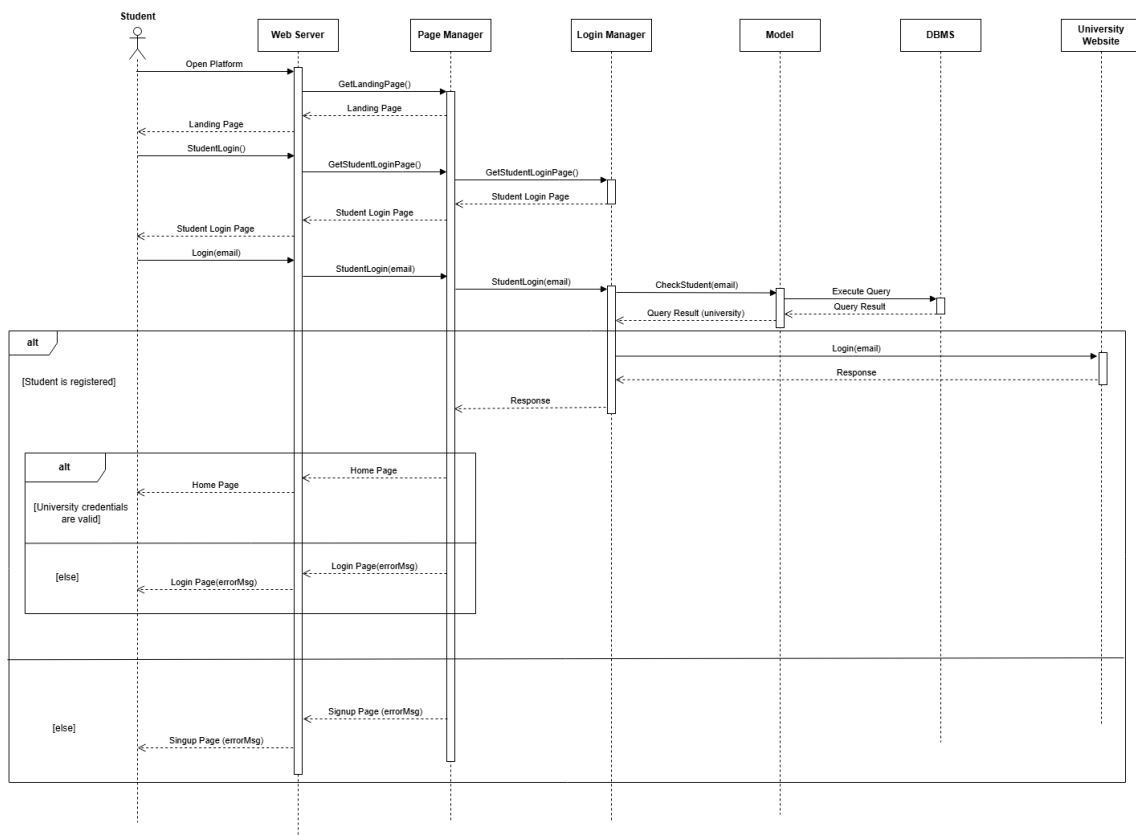


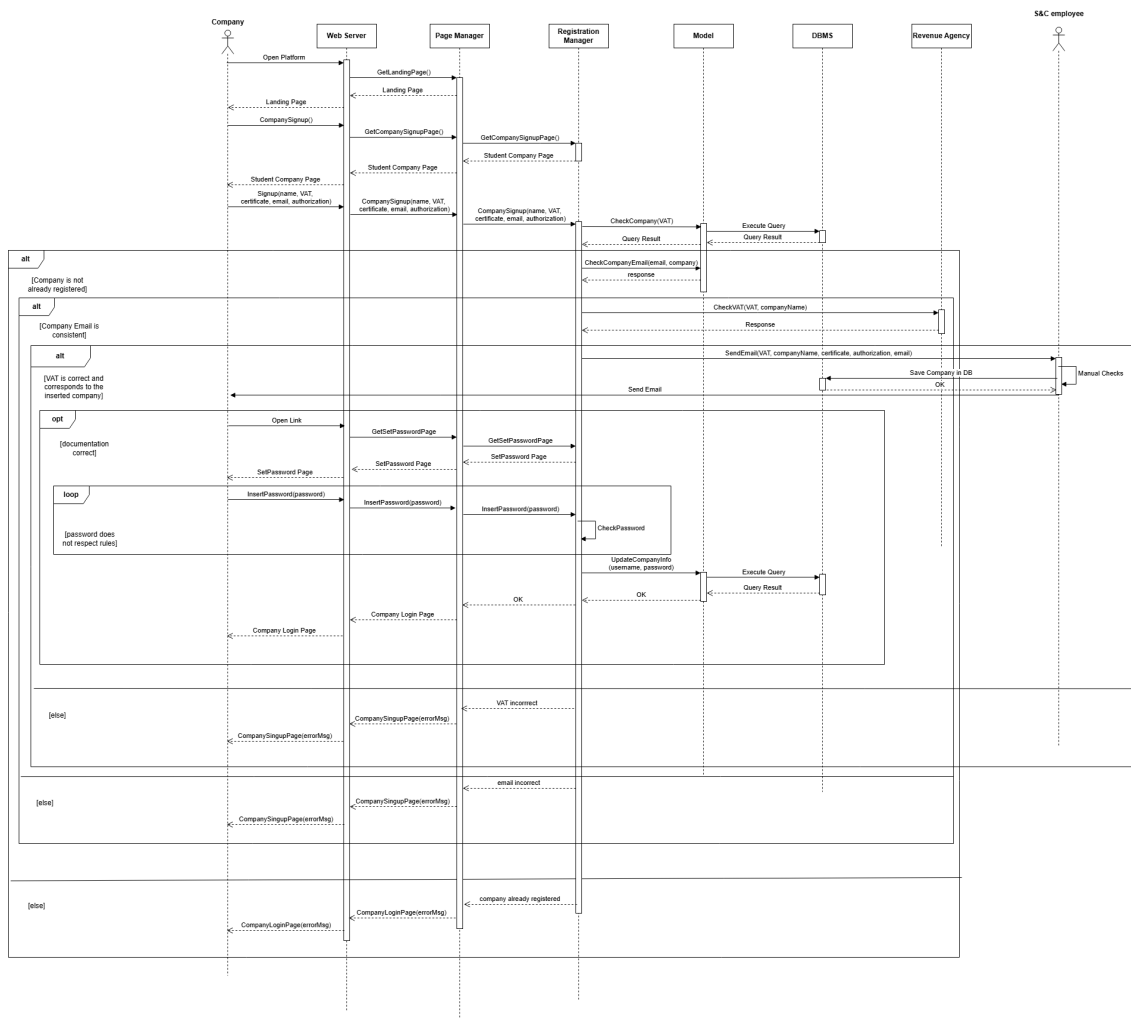
Figure 2.16: SD: Student Sign up

## Student Login



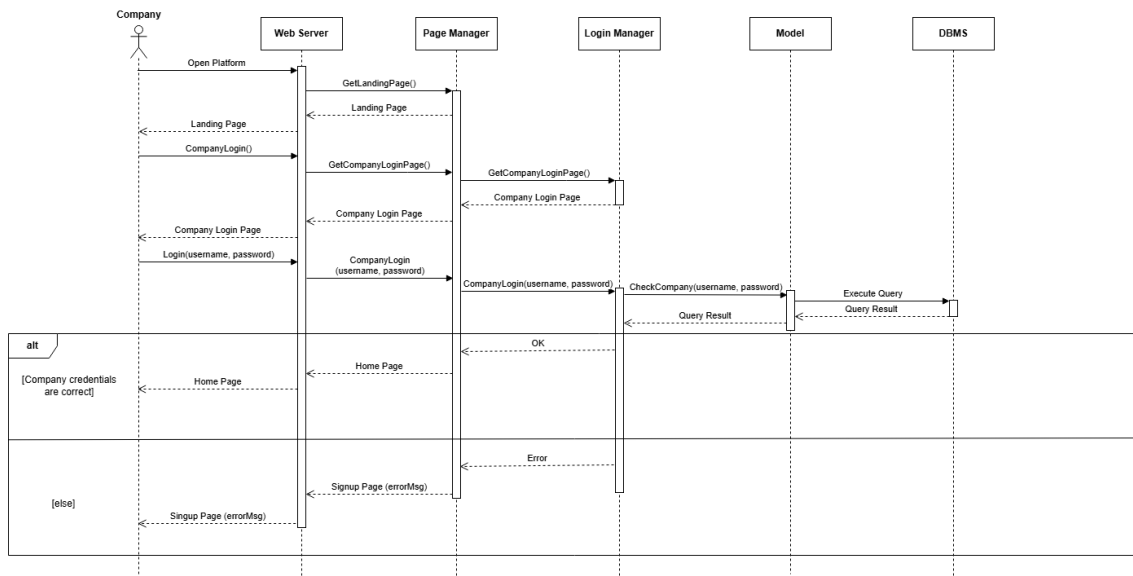
**Figure 2.17:** SD: Student Login

## Company Sign up



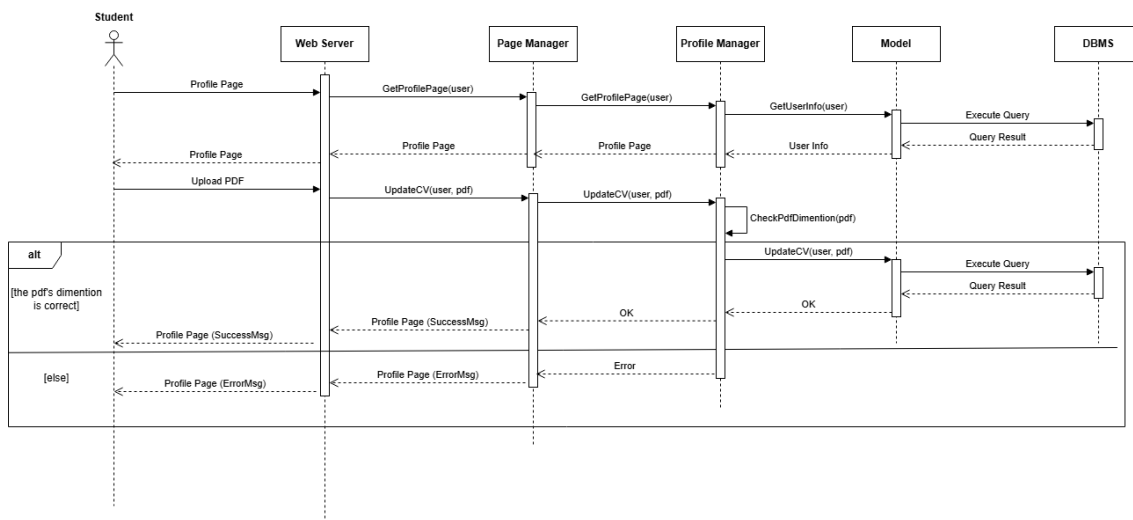
**Figure 2.18:** SD: Company Sign up

## Company Login



**Figure 2.19: SD: Company Login**

## Upload Student CV



**Figure 2.20: SD: Upload Student CV**

## Add new competence

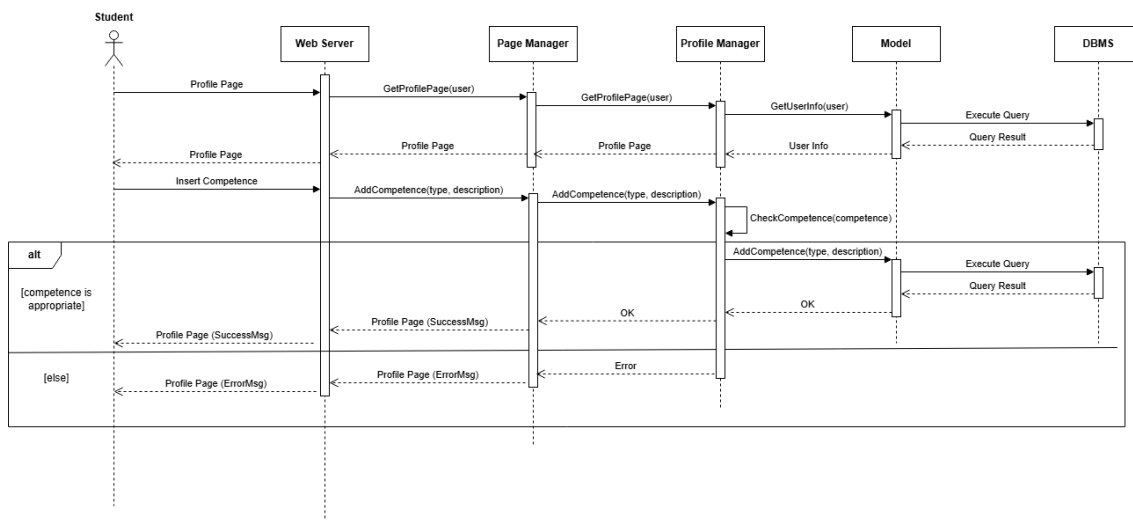


Figure 2.21: SD: Add new competence

## Edit competence

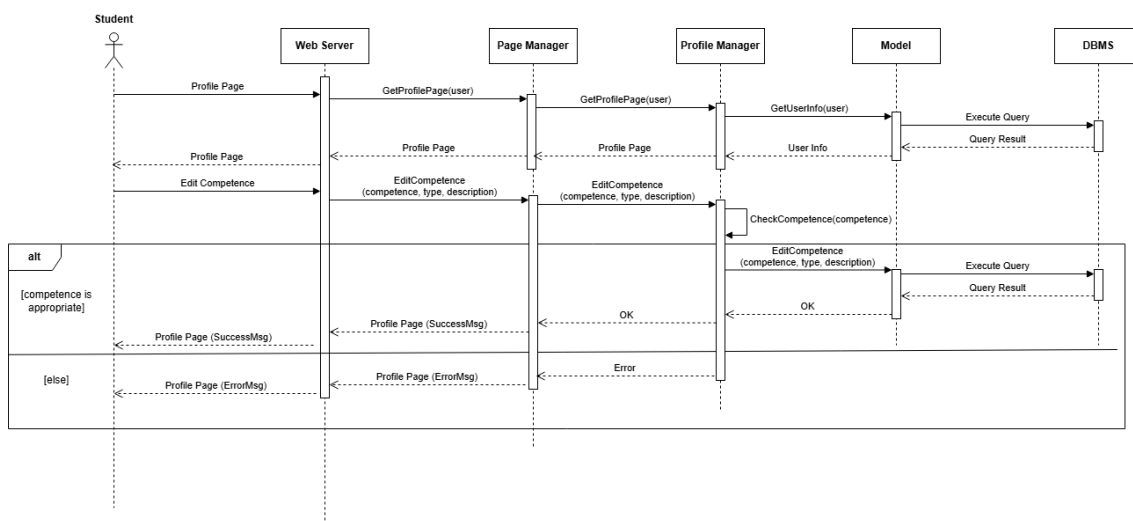


Figure 2.22: SD: Edit competence



## Create New ADV

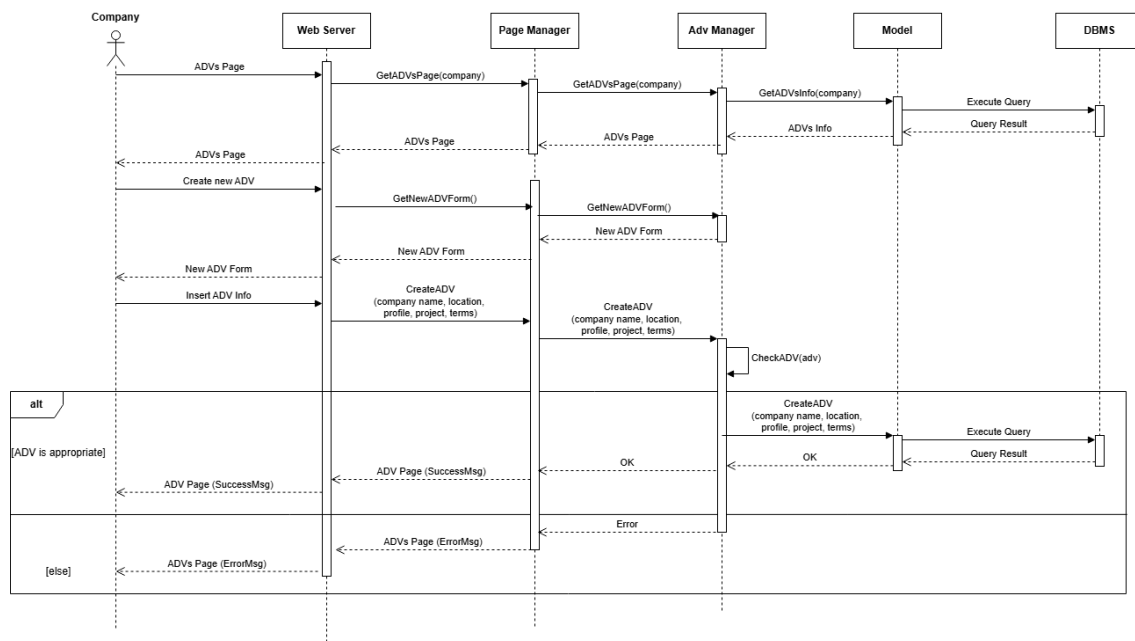


Figure 2.23: SD: Create New ADV

## Edit ADV

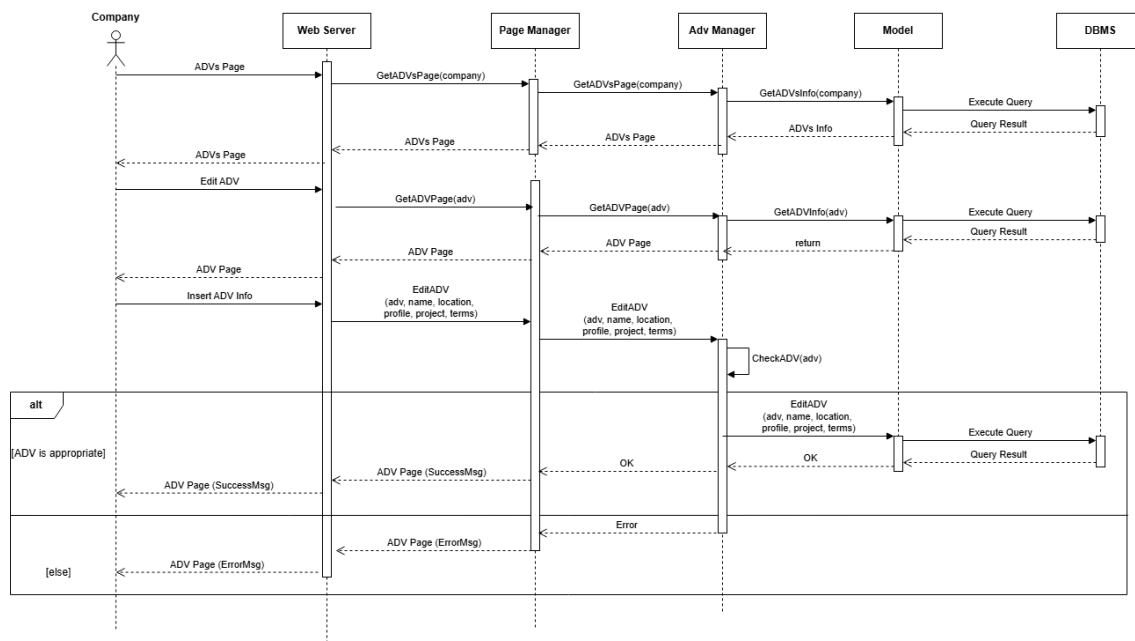
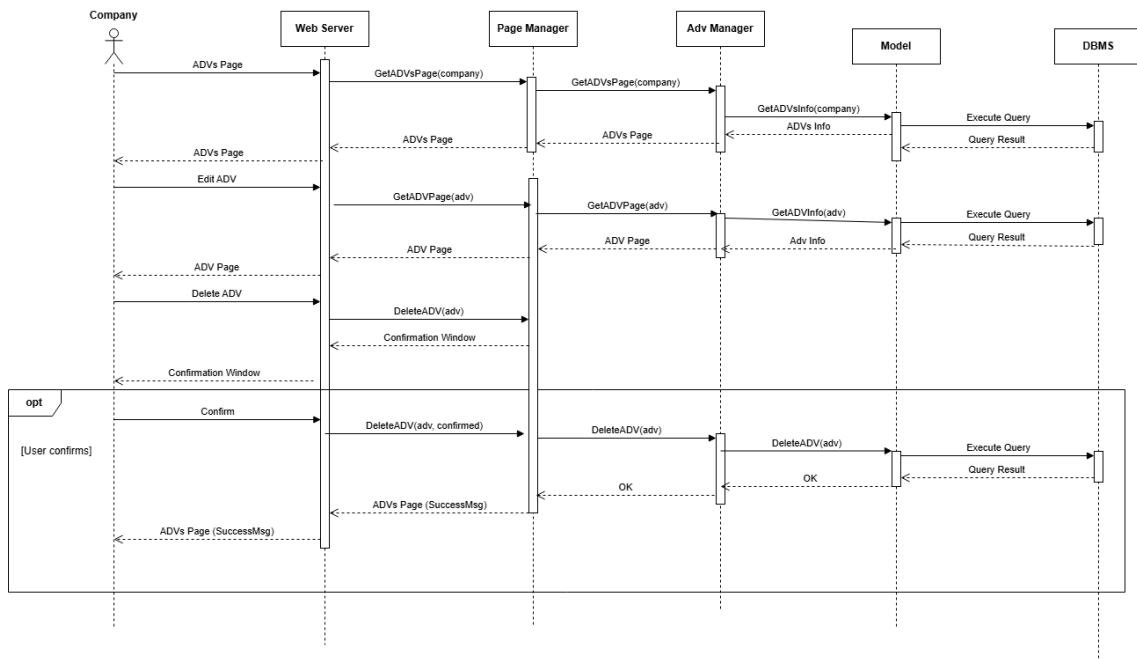


Figure 2.24: SD: Edit ADV

**Delete ADV****Figure 2.25: SD: Delete ADV**

## Proactive Internship Search

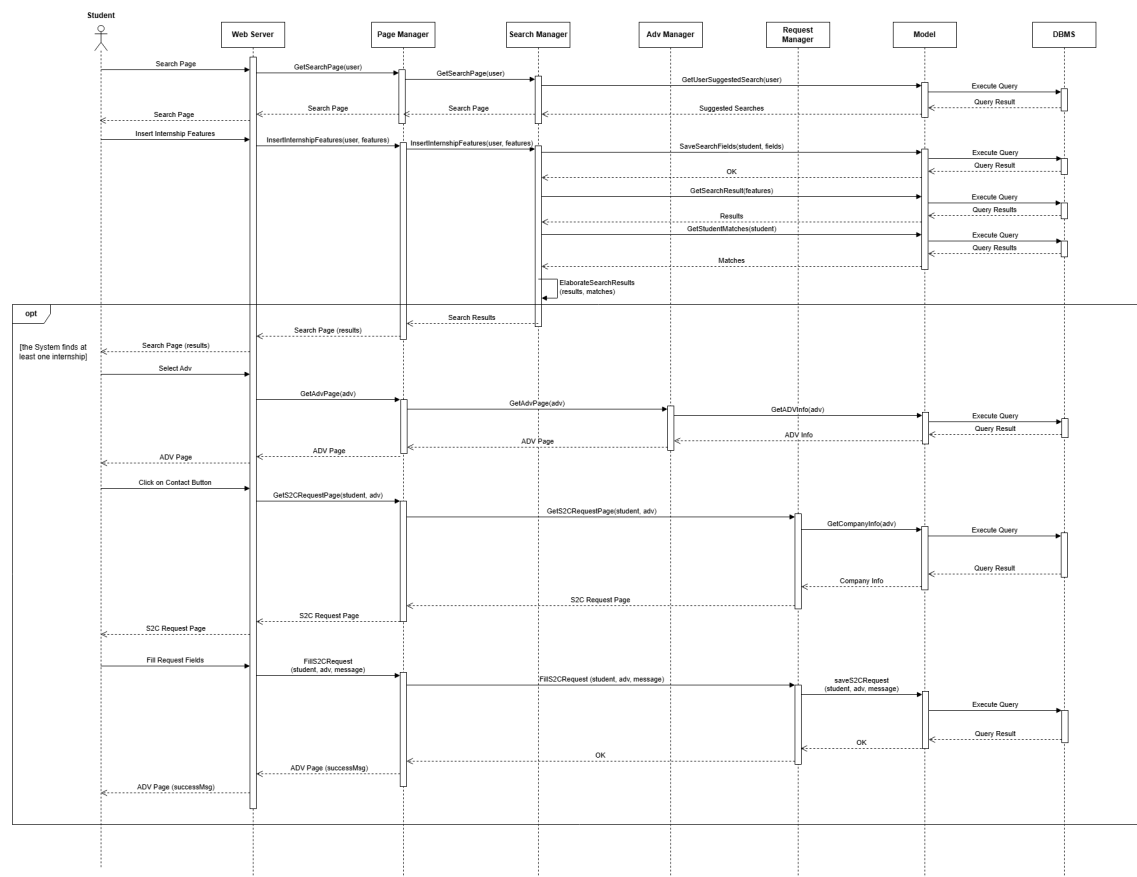


Figure 2.26: SD: Proactive Internship Search

## Proactive Candidate Search

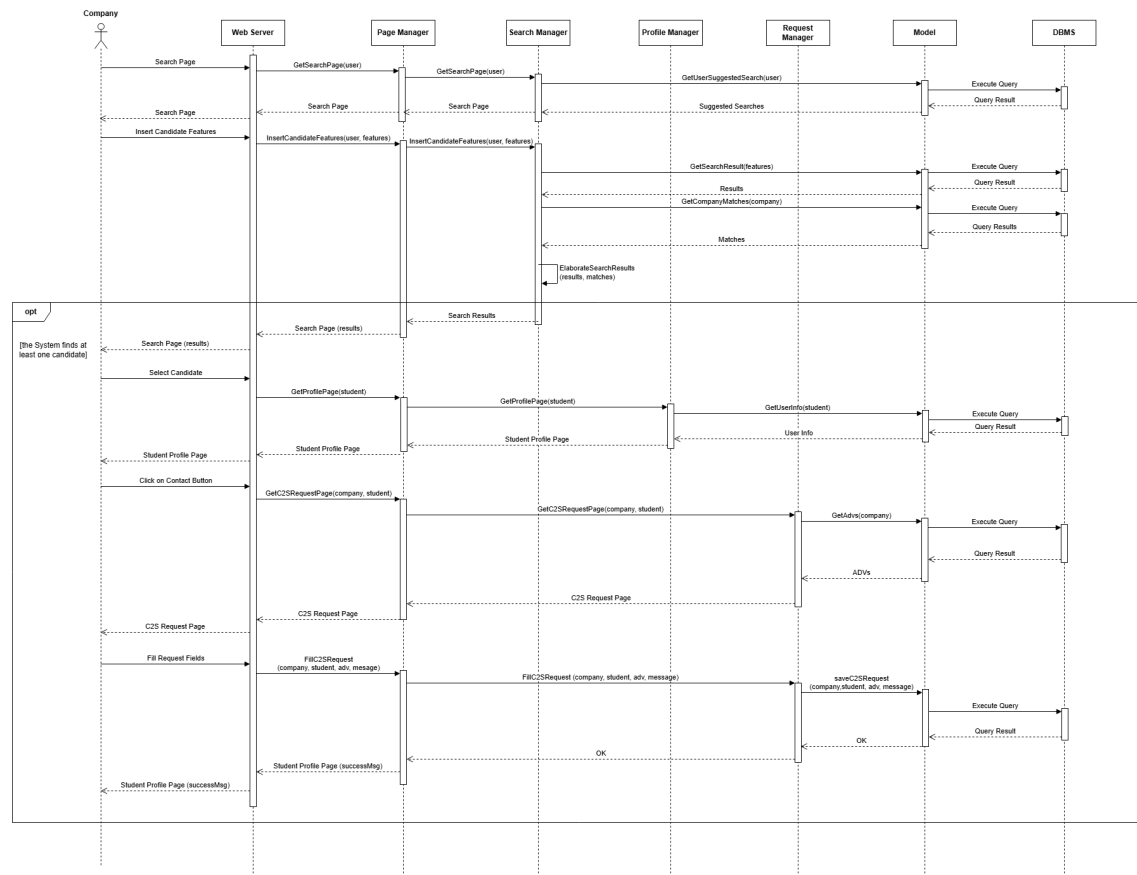
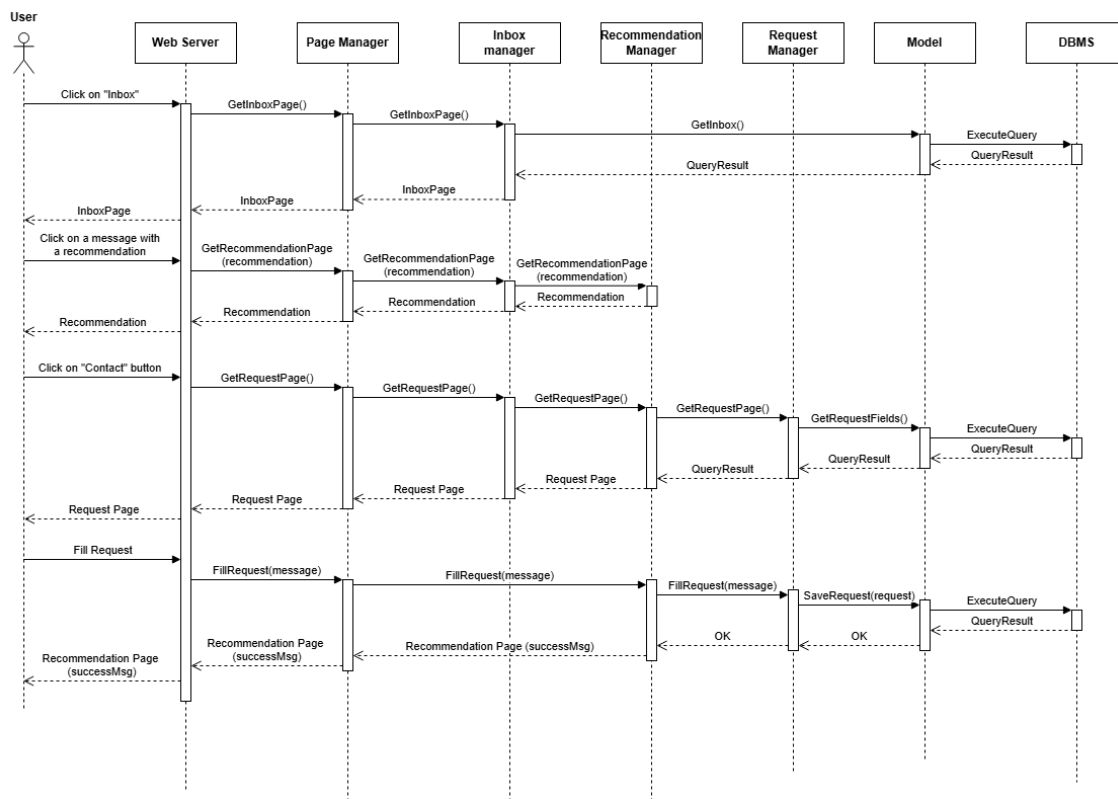


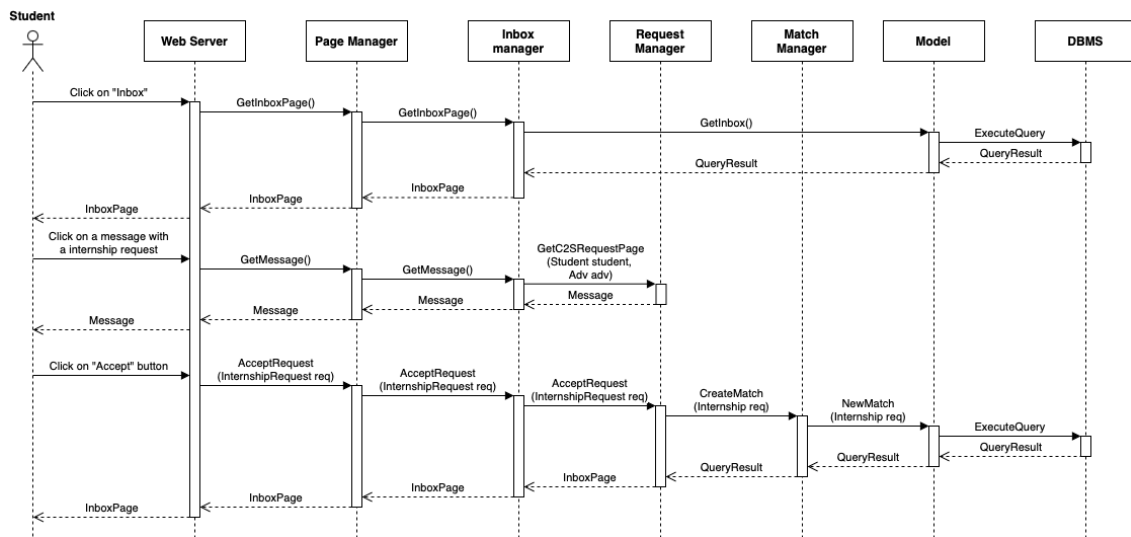
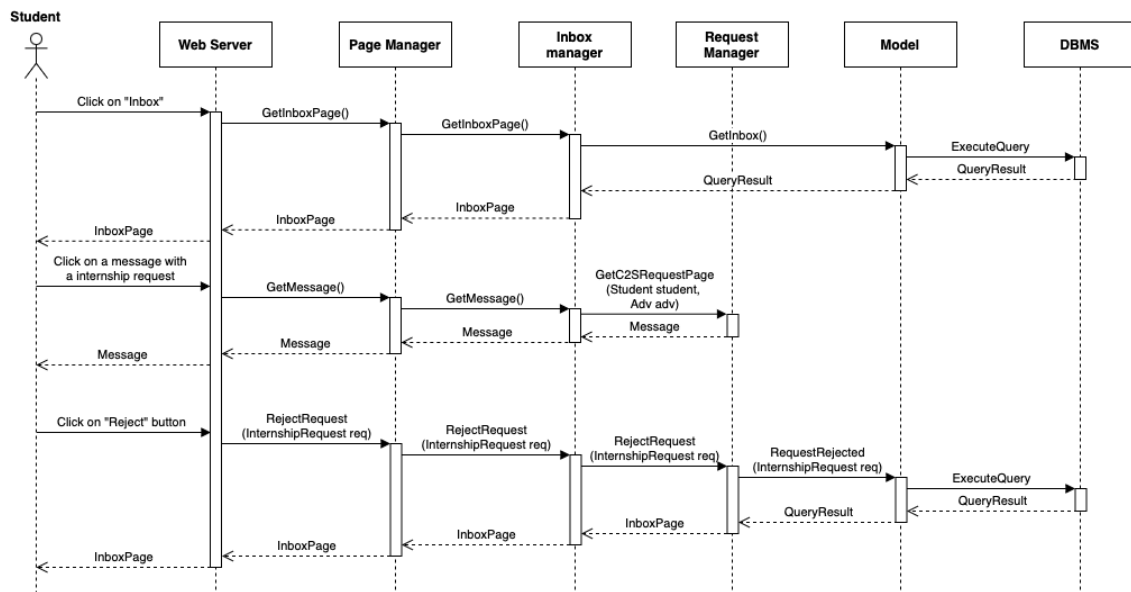
Figure 2.27: SD: Proactive Candidate Search

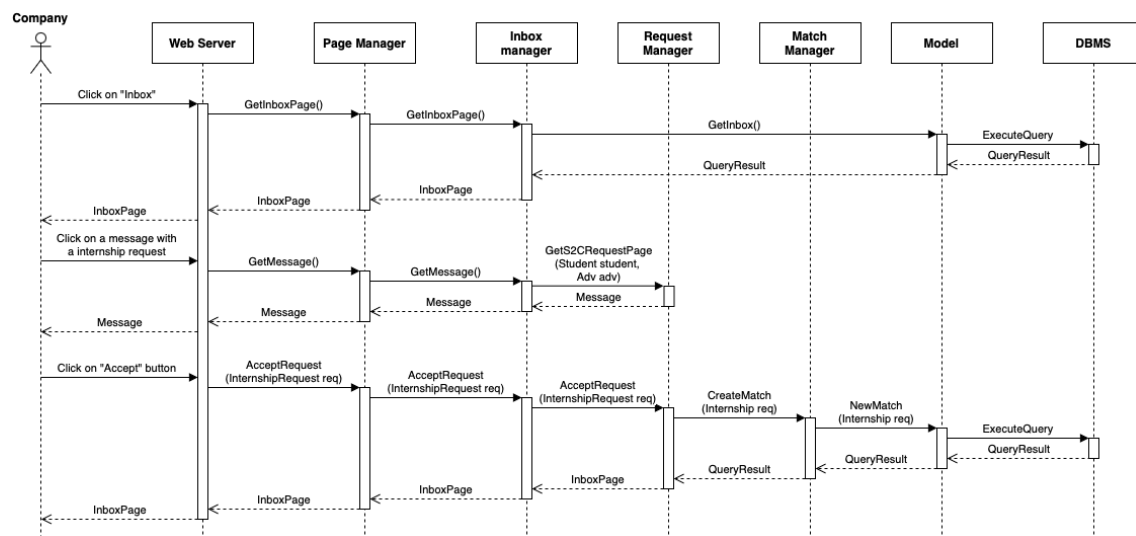
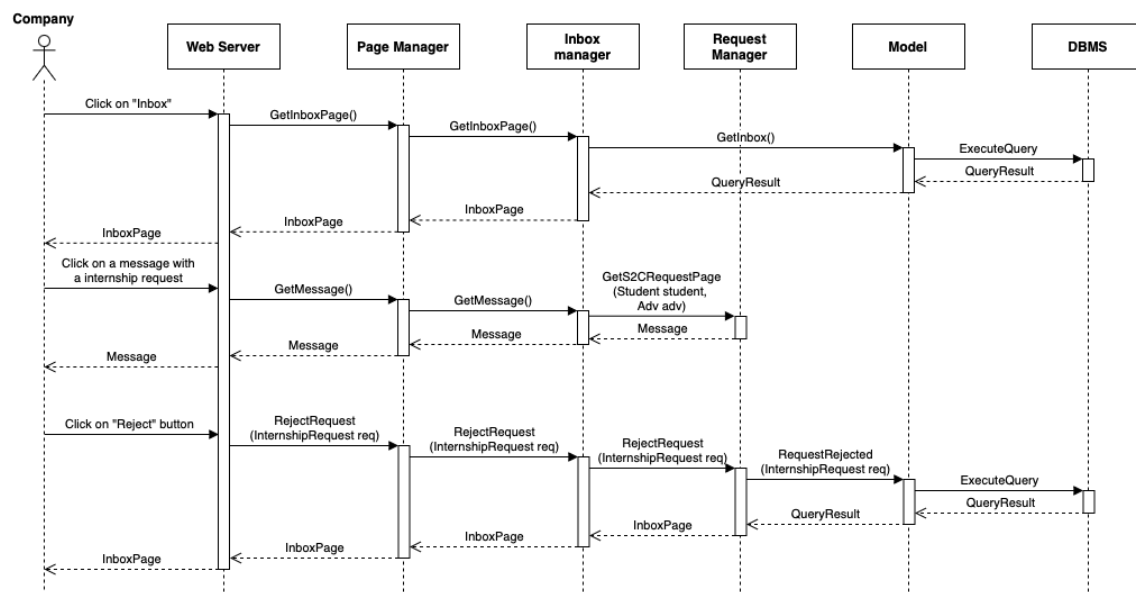
## Recommendation



**Figure 2.28:** SD: Recommendation

*Note:* This diagram is generalised for every type of user, so the request's functions are generalised too. The specific ones are used in the search diagrams.

**Accept internship request (Student)****Figure 2.29: SD: Accept internship request (Student)****Reject internship request (Student)****Figure 2.30: SD: Reject internship request (Student)**

**Accept internship request (Company)****Figure 2.31: SD: Accept internship request (Company)****Reject internship request (Company)****Figure 2.32: SD: Reject internship request (Company)**

### Check request outcome (Student)

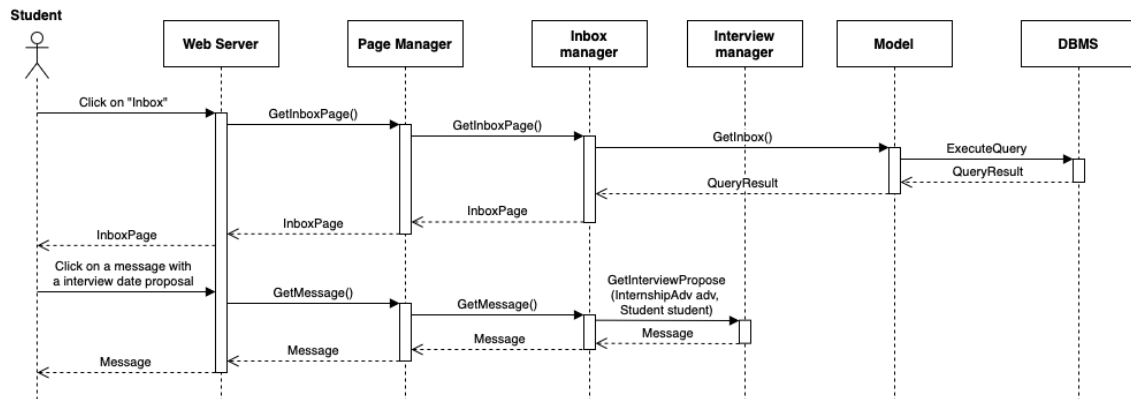


Figure 2.33: SD: Check request outcome (Student)

### Check request outcome (Company)

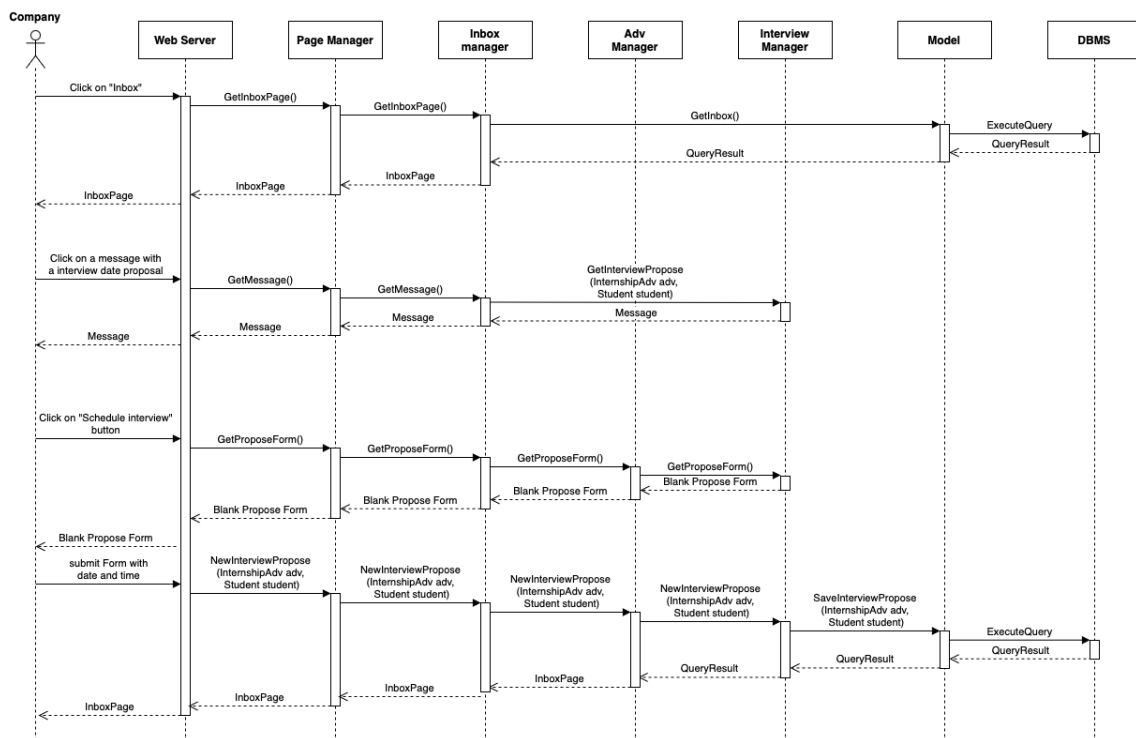


Figure 2.34: SD: Check request outcome (Company)



### Propose a date for an interview

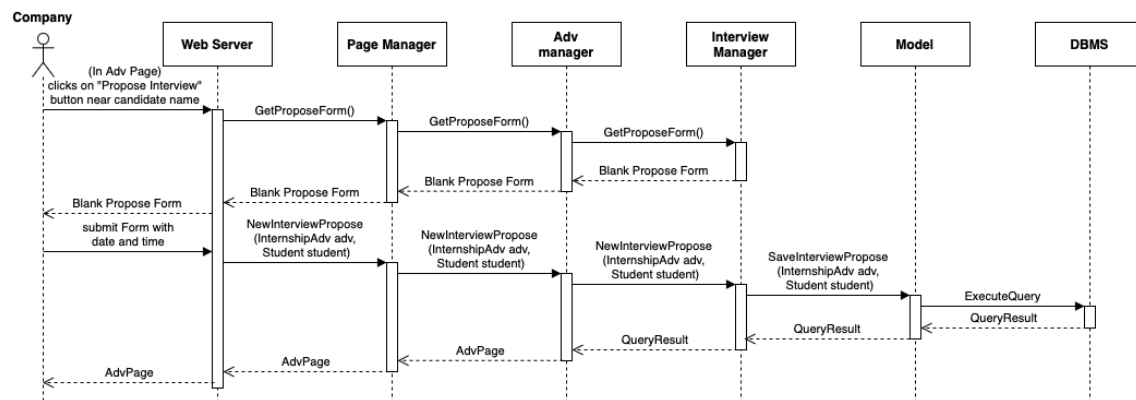


Figure 2.35: SD: Propose a date for an interview

### Accept a date for an interview

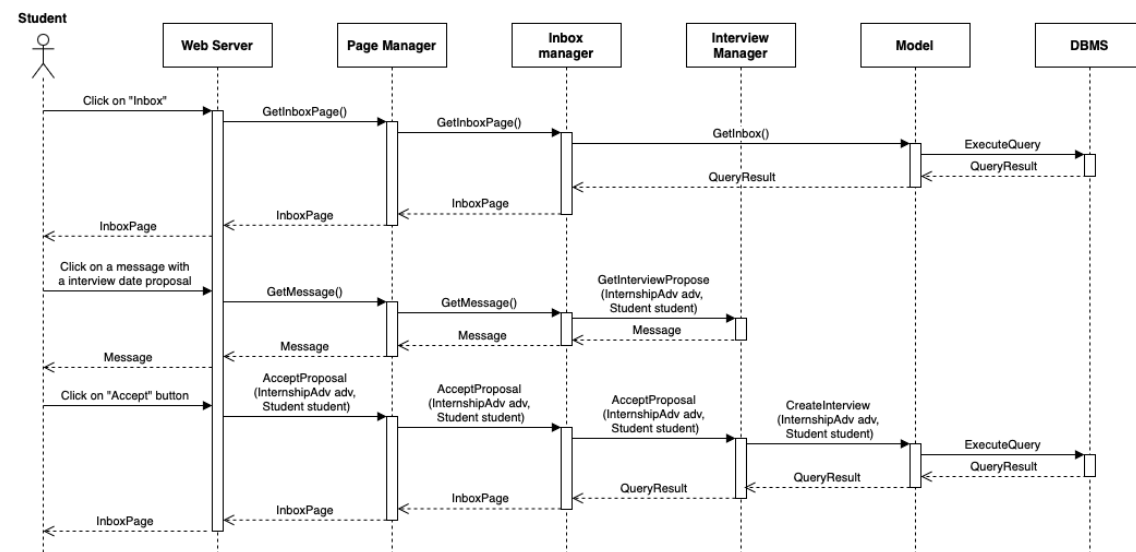


Figure 2.36: SD: Accept a date for an interview

### Reject a date for an interview

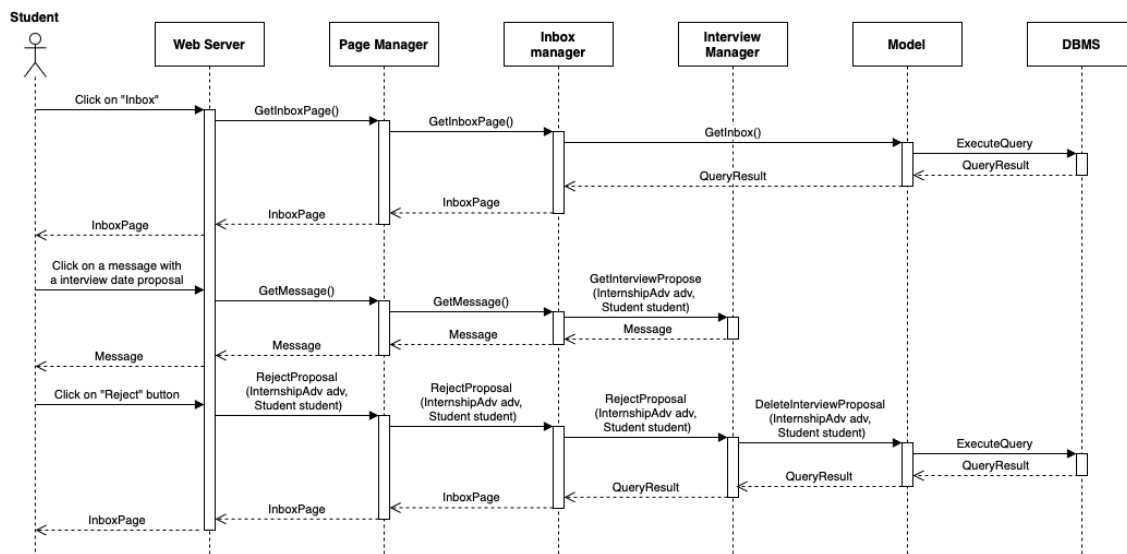


Figure 2.37: SD: Reject a date for an interview

### Creating form for interview

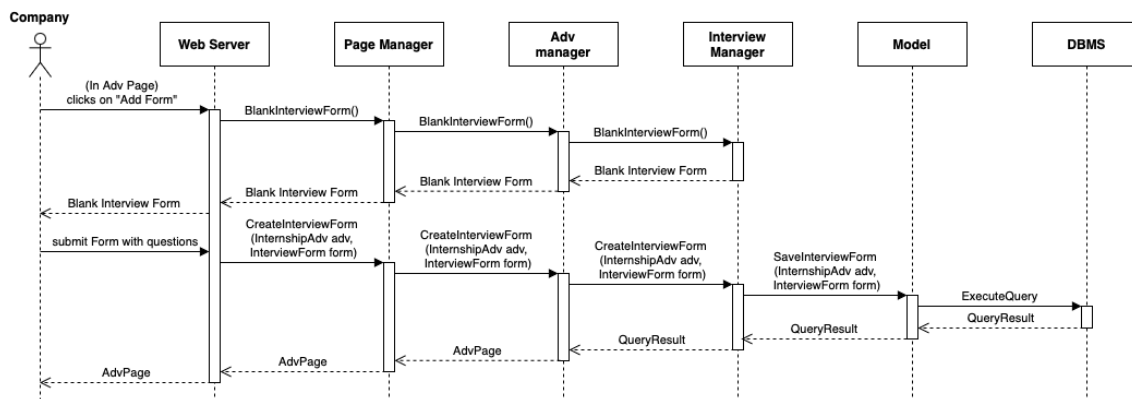


Figure 2.38: SD: Creating form for interview

### Deleting form for interview

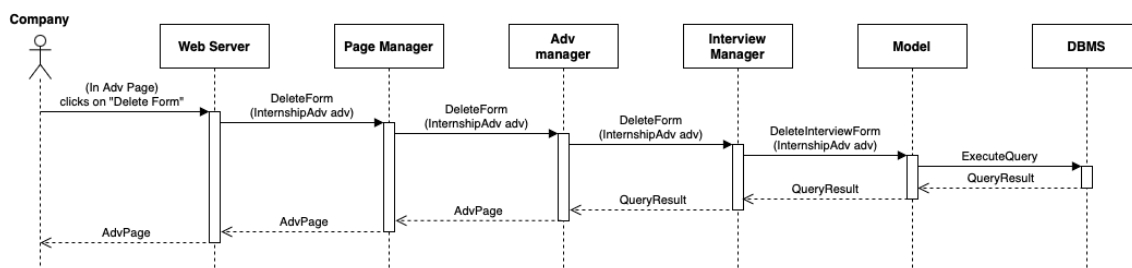


Figure 2.39: SD: Deleting form for interview

### Modify form for interview

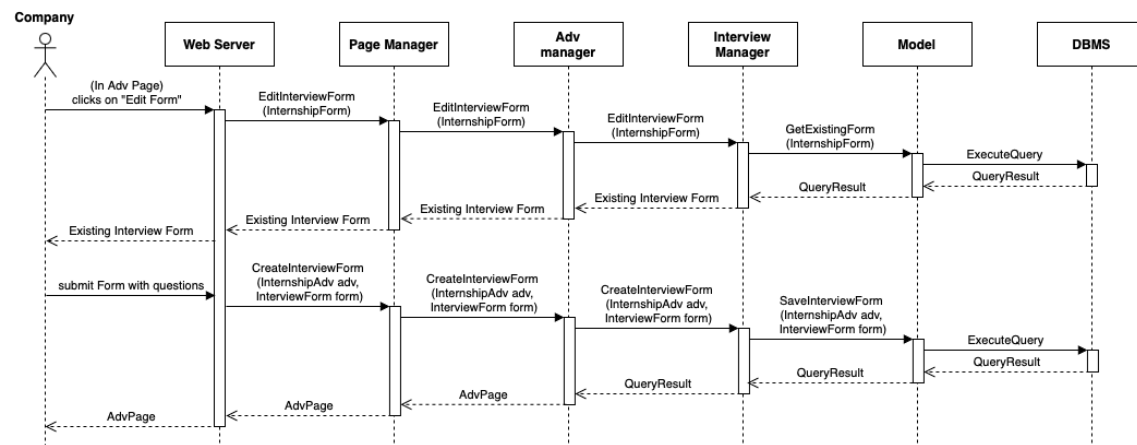


Figure 2.40: SD: Modify form for interview

### Fill the form during interview

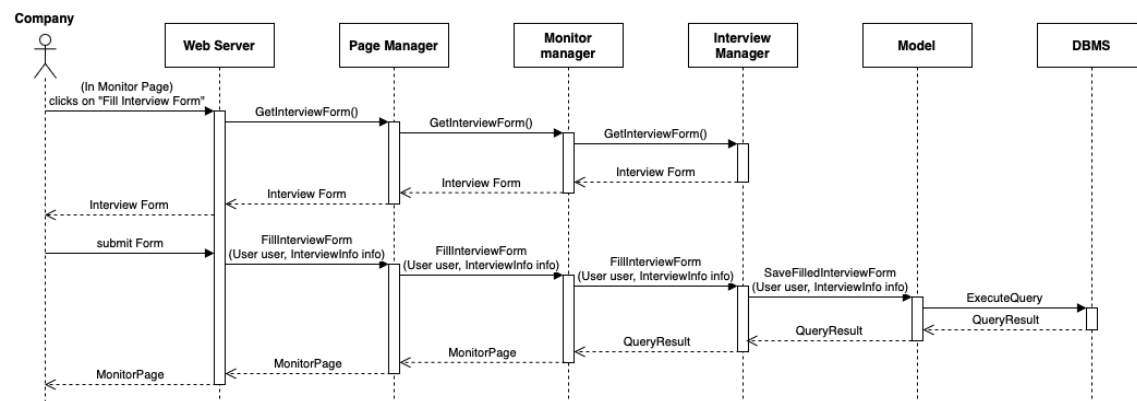


Figure 2.41: SD: Fill the form during interview

## Feedback

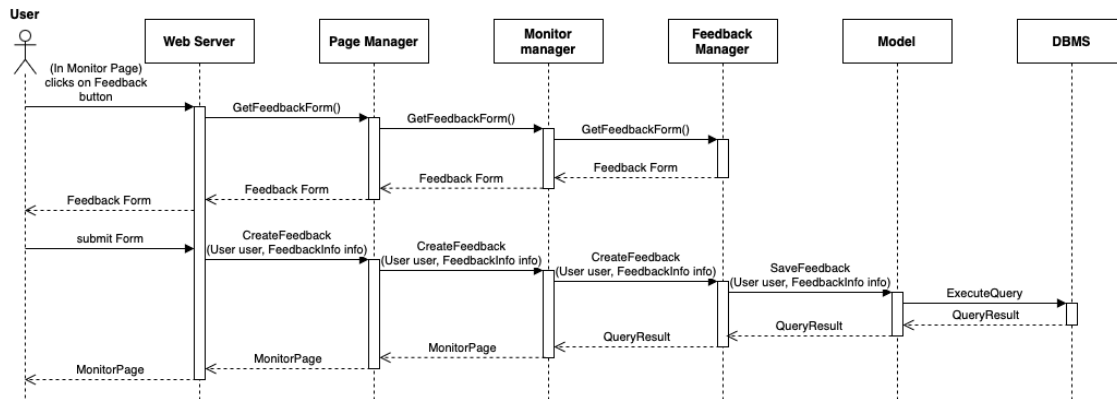


Figure 2.42: SD: Feedback

## Monitoring

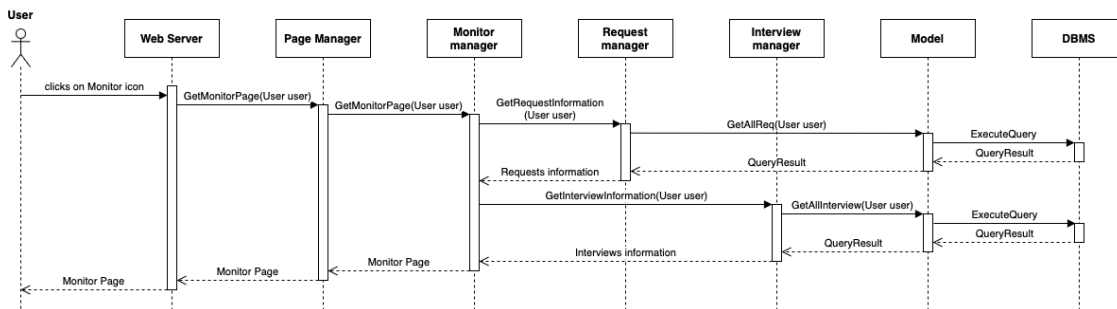


Figure 2.43: SD: Monitoring

## 2.5. Component interfaces

- **Page Manager**
  - It contains all the interfaces of all the others pages
- **Registration Manager**
  - GetStudentSingupPage()
  - StudentSingup(String email, String university)
  - StudentLogin(String email)
  - GetCompanySingupPage()
  - CompanySignup(String name, String VAT, File certificate, String email, File authorization)
  - GetSetPasswordPage()
  - InsertPassword(String password)

- **Login Manager**
  - GetStudentLoginPage()
  - StudentLogin(String email)
  - GetCompanyLoginPage()
  - CompanyLogin(String username, String password)
- **Request Manager**
  - GetS2CRequestPage(Student student, Adv adv)
  - FillS2CRequest(Student student, Adv adv, String message)
  - GetC2SRequestPage(Company company, Student student)
  - FillC2SRequest(Company company, Student student, Adv adv, String message)
  - GetRequestInformation(User user)
  - AcceptRequest(InternshipRequest req)
  - RejectRequest(InternshipRequest req)
- **Profile Manager**
  - GetProfilePage(User user)
  - UpdateCV(User user, File pdf)
  - CheckPdfDimention(File pdf)
  - AddCompetence(Map<CompetenceEnum, String> competences)
  - EditCompetence(Competence competence, EnumCompetence type, String description)
- **Interview Manager**
  - GetInterviewInformation(User user)
  - GetInterviewForm()
  - FillInterviewForm(User user, InterviewInfo info)
  - EditInterviewForm(InternshipForm)
  - BlankInterviewForm()
  - CreateInterviewForm(InternshipAdv adv, InterviewForm form)
  - DeleteForm(InternshipAdv adv)
  - NewInterviewPropose(InternshipAdv adv, Student student)
  - GetProposeForm()
  - GetInterviewPropose(InternshipAdv adv, Student student)
  - AcceptProposal(InternshipAdv adv, Student student)
  - RejectProposal(InternshipAdv adv, Student student)
- **Adv Manager**

- GetADVPage(Company company)
- GetNewADVForm()
- CreateADV(Company company, String name, String location, String profile, String project, String terms)
- CheckADV(Adv adv)
- GetADVPage(Adv adv)
- CreateADV(Company company, String name, String location, String profile, String project, String terms)
- EditCompetence(Competence competence, EnumCompetence type, String description)
- DeleteADV(Adv adv)
- **Inbox Manager**
  - GetInboxPage()
- **Search Manager**
  - GetSearchPage(User user)
  - InsertInternshipFeatures(User user, List<String> features)
  - InsertCandidateFeatures(User user, List<String> features)
  - ElaborateSearchResults(List<Adv> results, List<Match> matches)
- **Recommendation Manager**
  - GetRecommendationPage(Recommendation recommendation)
- **Match Manager**
  - CreateMatch(Internship req)
- **Monitor Manager**
  - GetMonitorPage(User user)
- **Feedback Manager**
  - GetFeedbackForm()
  - CreateFeedback(User user, FeedbackInfo info)
- **Model**
  - CheckStudent(String email)
  - CheckUniversity(String university)
  - CheckCompany(String VAT)
  - UpdateCompanyInfo(String username, String password)
  - CheckCompany(String username, String password)
  - GetUserInfo(User user)
  - UpdateCV(User user, File pdf)

- AddCompetence(Map<CompetenceEnum, String> competences)
- EditCompetence(Competence competence, CompetenceEnum type, String description)
- GetADVInfo(Company company)
- CreateADV(Company company, String name, String location, String profile, String project, String terms)
- GetADVInfo(Adv adv)
- EditCompetence(Competence competence, EnumCompetence type, String description)
- DeleteADV(Adv adv)
- GetUserSuggestedSearch(User user)
- GetSearchResult(List<String> features)
- GetStudentMatches(Student student)
- GetCompanyInfo(Adv adv)
- SaveS2CRequest(Student student, Adv adv, String message)
- SaveC2SRequest(Company company, Student student, Adv adv, String message)
- GetCompanyMatches(Company company)
- SaveSearchFields(Student student, List<String> fields)
- GetAllInterview(User user)
- GetAllReq(User user)
- SaveFeedback(User user, FeedbackInfo info)
- SaveInterviewForm(InternshipAdv adv, InterviewForm form)
- GetExistingForm(InternshipForm)
- SaveFilledInterviewForm(User user, InterviewInfo info)
- DeleteInterviewForm(InternshipAdv adv)
- GetInbox
- SaveInterviewPropose(InternshipAdv adv, Student student)
- DeleteInterviewProposal(InternshipAdv adv, Student student)
- NewMatch(Internship req)
- RequestRejected(InternshipRequest req)

## 2.6. Selected architectural styles and patterns

The S&C system uses a combination of architectural styles and design patterns to ensure modularity, scalability, and maintainability. The core of the system is built around **Micro-service Architecture** and is integrated with a **3-Tier Architecture** and a **Client-Server Model**. These enable the system to meet performance and goals while maintaining flexibility.

### 2.6.1. Micro-service Architecture

The Micro-service Architecture divides the application into independent service, each one responsible for a specific functionality. It enables efficient development, testing, deployment, and scaling thanks to its modularity, each micro-service operates independently, communicating with other through APIs. A unified **API Gateway** serves as a centralised entry point for client requests, reducing direct interactions between clients and individual micro-services.

### 2.6.2. 3-Tier Architecture

The 3-Tier Architecture is a software design model that organises the application into three layers. This model ensures modularity (each tier can be developed individually), scalability, maintainability (the separation of concerns allows for easier debugging), and security (it isolates the data, so the direct access to it is restricted). The three layers are:

1. **Presentation Tier**, i.e. the user interface layer: this tier displays information to users and collects their input. In the S&C platform, it is represented by the web pages and is located on the Web Server node of the physical architecture. This layer interacts with the micro-service through the unified API Gateway.
2. **Application Tier**, i.e. the business logic layer: This tier contains the function to process input, execute decisions, and perform computations. It works as a bridge between the first and third layers. In the S&C system is represented through the Model and is placed on the Application Server node. It ensures efficient data processing and coordination among micro-services.
3. **Data Tier**: this tier stores and manages data through databases.

### 2.6.3. Client-Server Model

The communication framework is developed following the Client-Server Model. The client is the web browser, it sends requests to the server using HTTPS protocol. The server processes the client's requests and provides responses. The Presentation Tier formats the server's responses while the Application Tier processes the client's requests.

## 2.7. Other design decisions

### 2.7.1. Scalability

The micro-services architecture implementation, as opposite at the monolithic one, enable a fine-gained scaling strategies, allowing a flexible deployment and more over a selective replication. Accordingly, with this approach we are able to scale the individual components rather than the entire system.



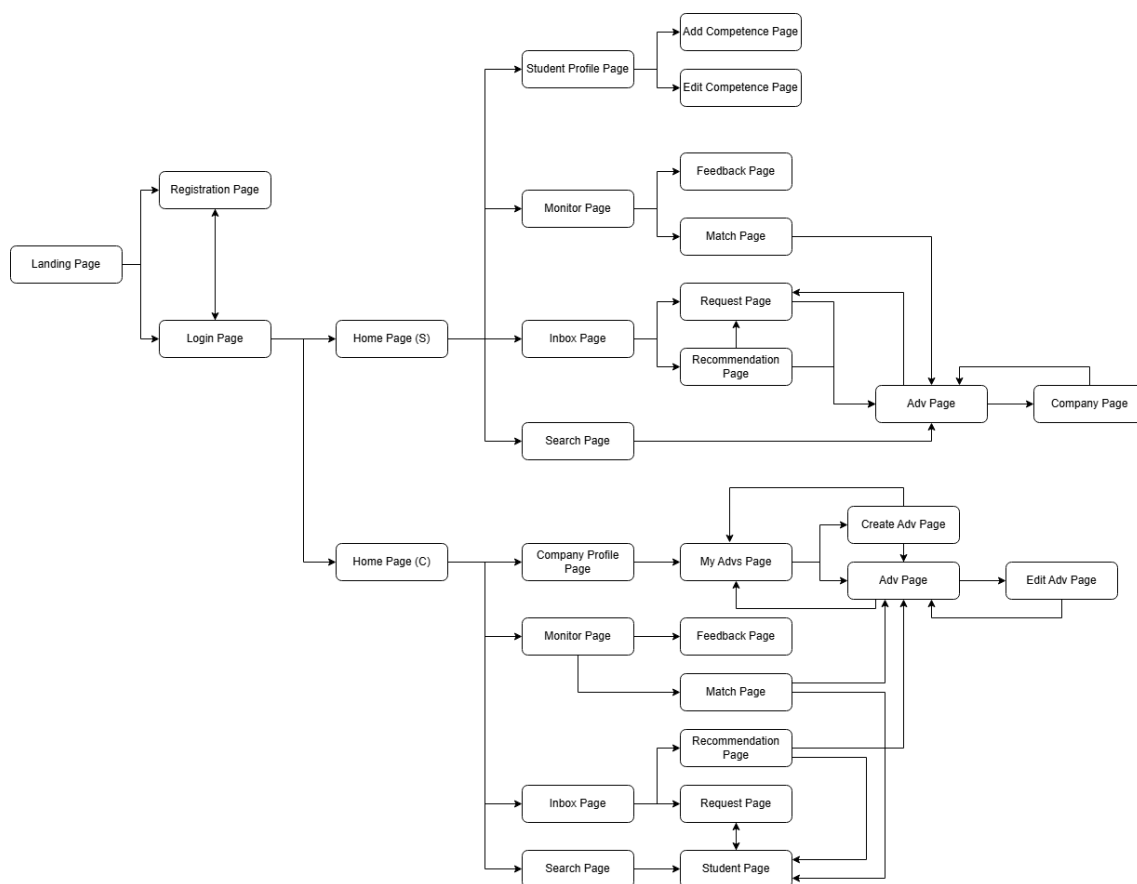
### 2.7.2. Ease of Deployment

The micro-services approach allows to enhance the ease of deployment, working with many smaller components, rather than a single big one, we can easily update them, test them and deploy them independently and, more over we can update and deploy the single components in parallel. Furthermore, with this approach we reduce the risks of critical errors, which can also be repaired easily in case they occur.



# 3 | User Interface Design

## 3.1. Overview



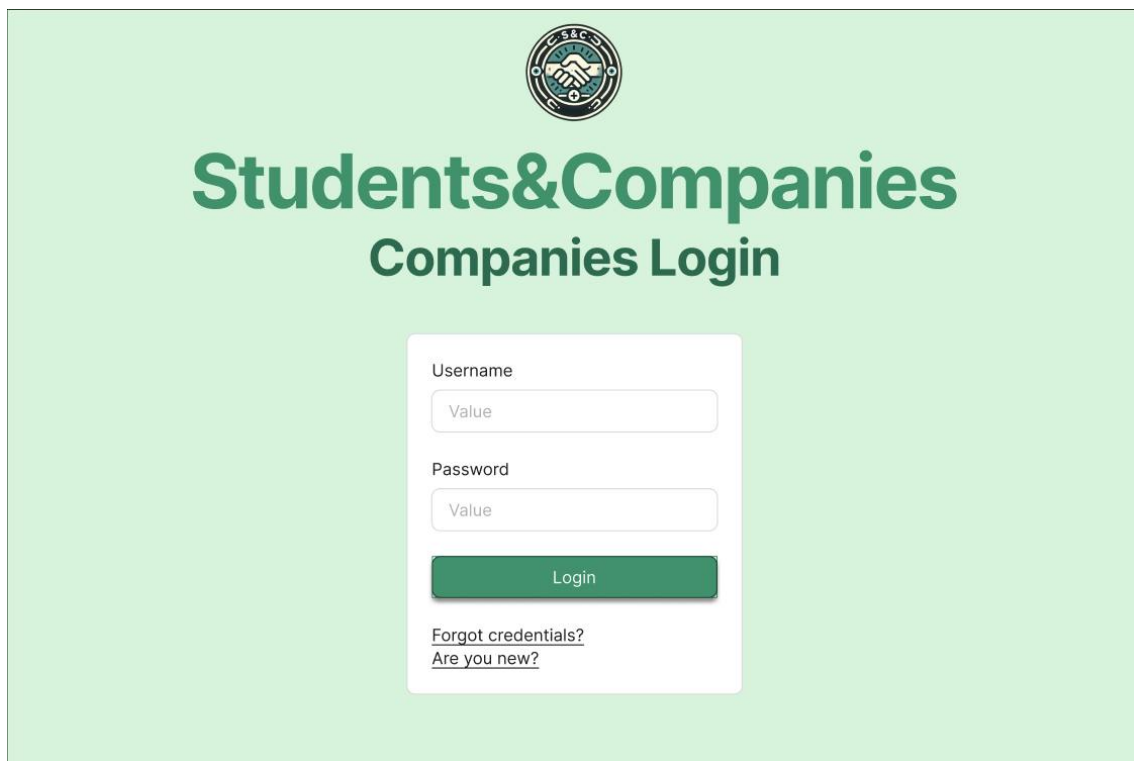
**Figure 3.1:** User Interface Overview

This diagram shows how users can navigate the platform. Upon entering the website's URL, they are directed to the **Landing Page**, where they can choose between Registration and Login. Both pages allow users to switch to the other if needed. After logging in, users are redirected to their personalised **Home Page**, which varies depending on their role (S - Student, C - Company). A navigation bar is present on all pages, enabling users to move across the most relevant sections, i.e. **Profile Page**, **Monitor Page**, **Inbox Page**, and **Search Page**. The arrows in the diagram represent links between pages, but not all links are shown to maintain diagram clarity. Additionally, back buttons are available on each page to facilitate navigation. When a student

visits an **ADV Page**, they can send a request regarding that ADV (as shown by the arrow linking the Adv Page to the Request Page). Similarly, companies can perform the same action when viewing a student's profile page. Lastly, as discussed in the RASD document, users can provide feedback via the **Monitor Page**.

## 3.2. Mock-up pages

### 3.2.1. Login Page



Students&Companies  
Companies Login

Username  
Value

Password  
Value

Login

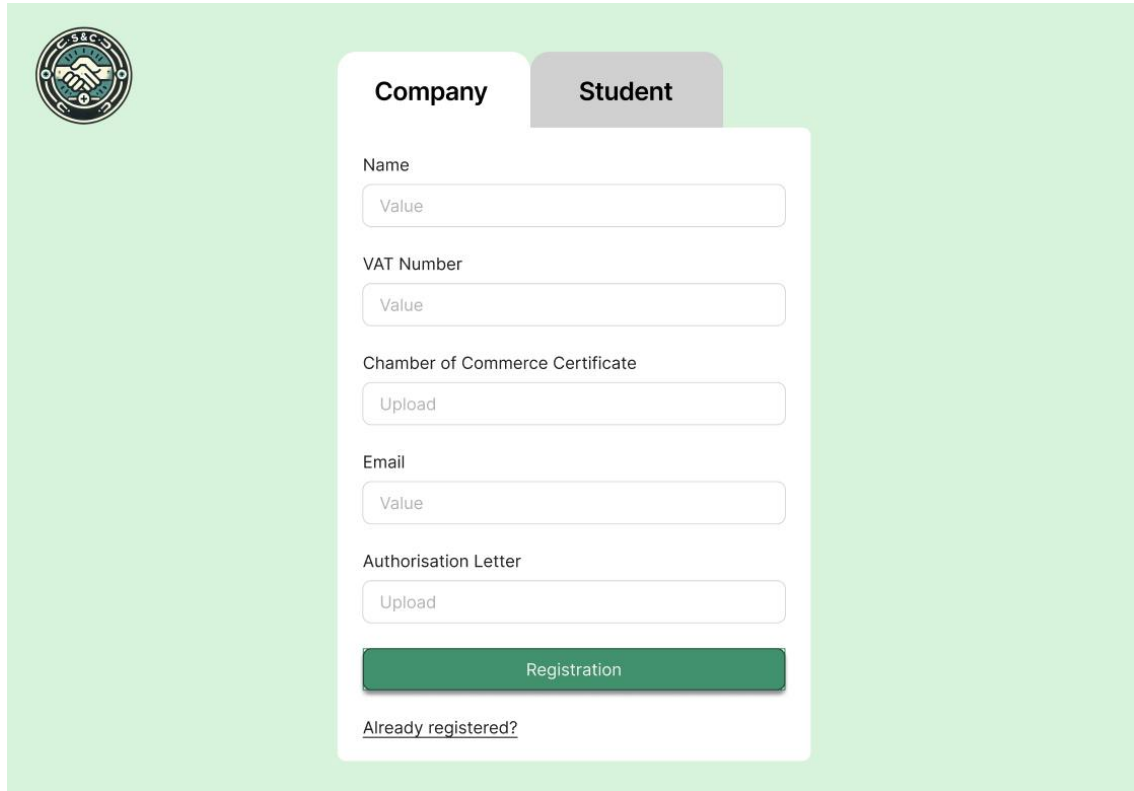
[Forgot credentials?](#)  
[Are you new?](#)

**Figure 3.2:** UI: Login Page

This is the Login Page for companies. As precedent discussed, the company's representative can insert the username and the password given at the moment of registration. If they do not remember the password or the username, they can click on the "Forgot credentials?" link, they will have to insert their company email and they will be able to reset the credentials. If the company is not registered, the representative can click on the "Are you new?" link to be redirected to the Sign up Page.

The student's login page is not shown because similar.

### 3.2.2. Sign Up Page

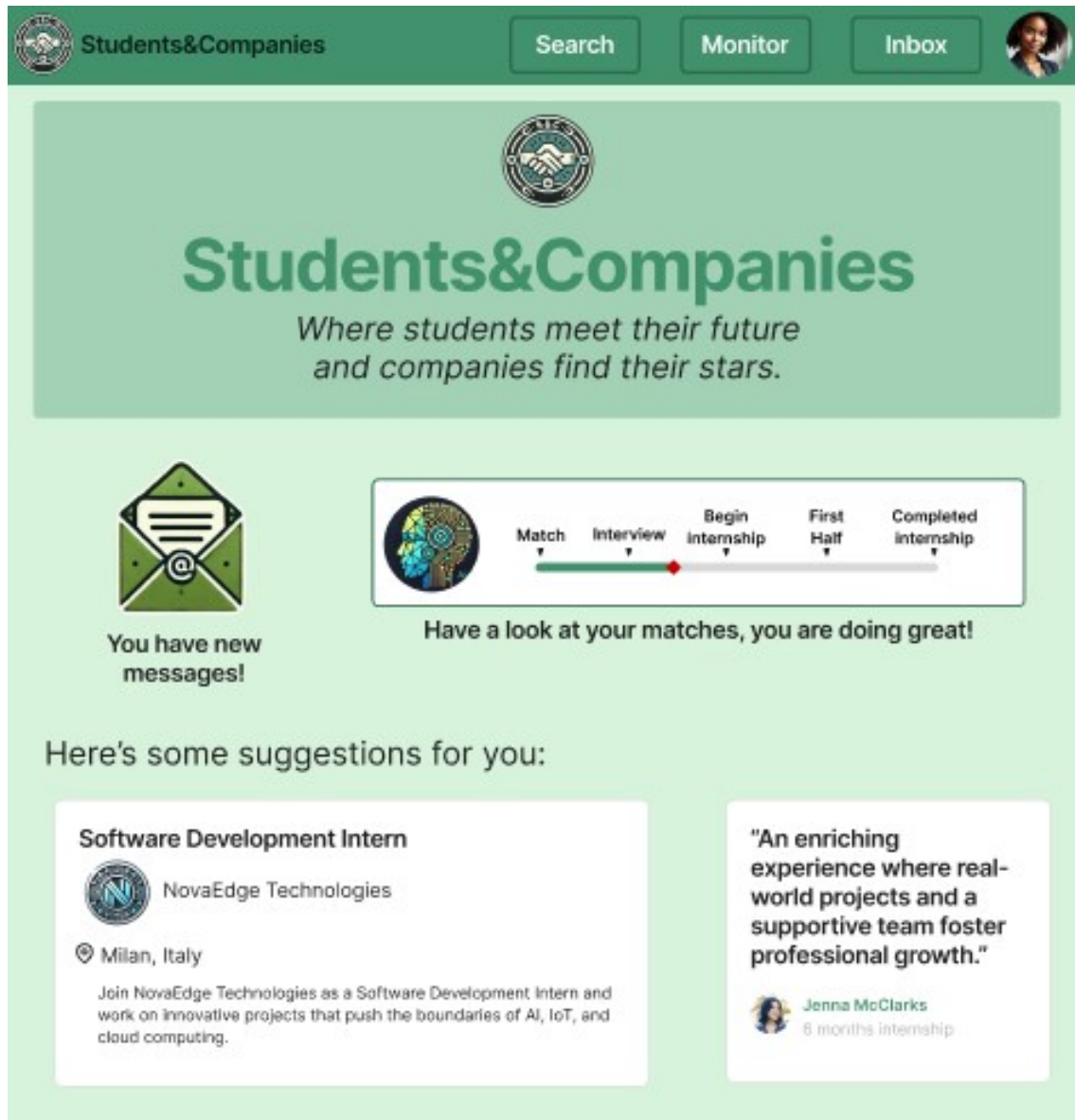


The mock-up shows a sign-up page with a light green background. In the top left corner is a circular logo with a handshake and the letters 'S&C'. The page features two tabs: 'Company' (active) and 'Student'. The 'Company' tab contains a form with the following fields: 'Name' (text input), 'VAT Number' (text input), 'Chamber of Commerce Certificate' (upload button), 'Email' (text input), and 'Authorisation Letter' (upload button). Below these fields is a green 'Registration' button and a link that says 'Already registered?'. The 'Student' tab is currently inactive.

**Figure 3.3:** UI: Sign up Page

This is the form shown to companies that want to register to the platform. If someone is already registered, they can click on the "Already registered?" link and will be redirected to the Login Page. Students can click on the "Student" section to compile their form. This is not shown because is similar.

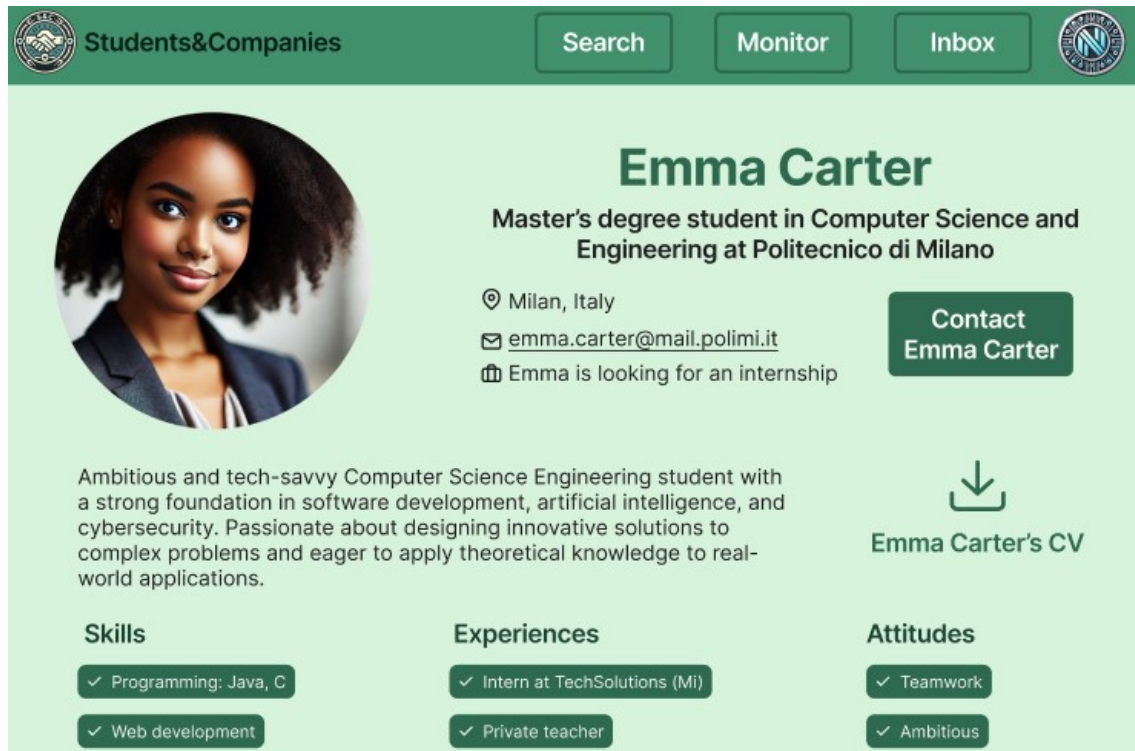
### 3.2.3. Home Page



**Figure 3.4:** UI: Home Page

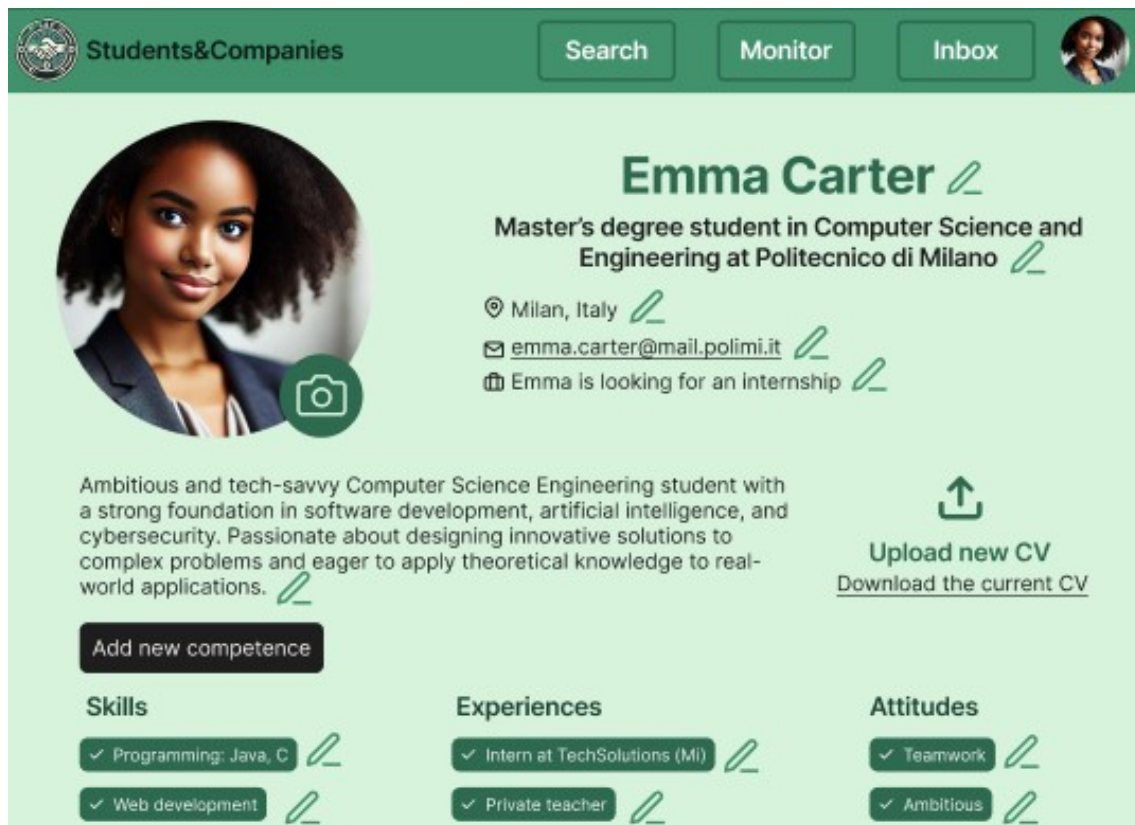
The **navigation bar** contains the logo and the name of the platform, they are clickable and directs to the Home Page. The other buttons direct to the correspondent pages, and the profile picture directs to the profile. The home page is personalised: in this case there is the envelope image that directs the user to the Inbox Page, and the monitor bar that redirects to the Monitor Page. The suggestions are linked to the Company Profile and to the ADV.

### 3.2.4. Student Profile Page



**Figure 3.5:** UI: Student Profile Page

This is a student's Profile Page seen by a company. By clicking on the download icon or on the "Emma Carter's CV" link, a user can download the student's CV. By clicking on one of the shown competence, a pop-up window opens and shows the description inserted by the student. If a company is interested, they can click on the "Contact Emma Carter" button to send an Internship Request to the student. By scrolling the page all the competences are shown, as well as relevant comments left by the companies.

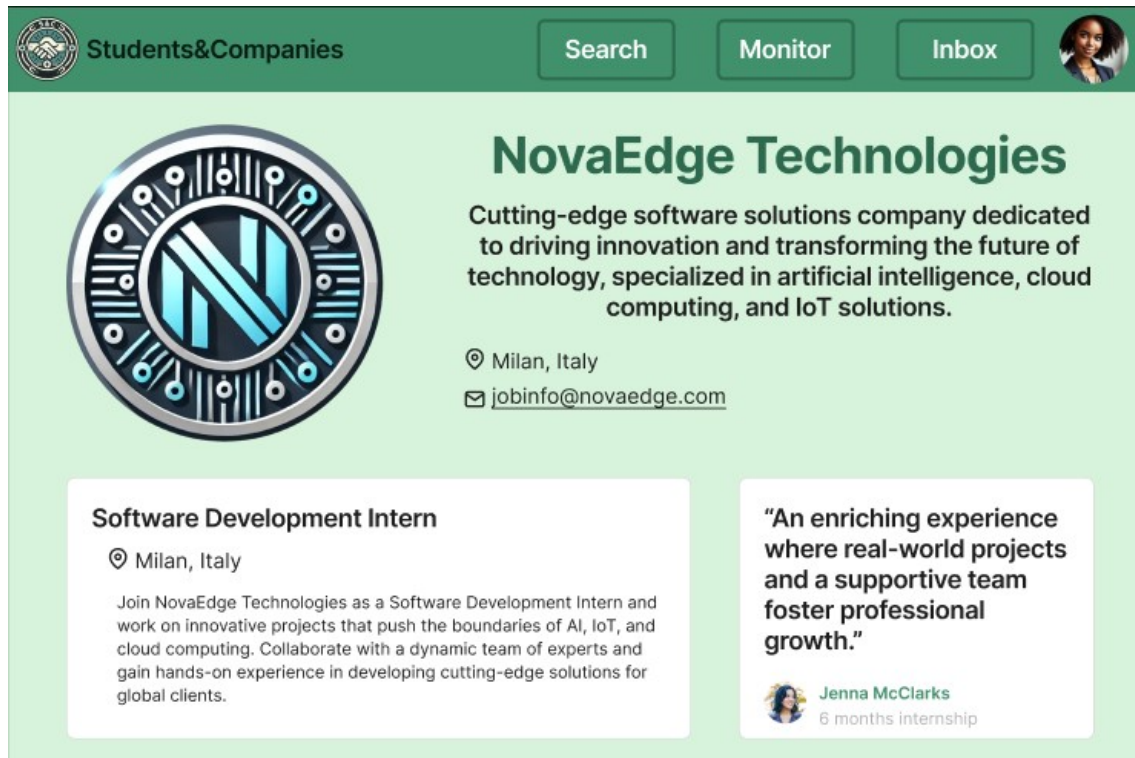


**Figure 3.6:** UI: Student Personal Profile Page

This image shows how the student sees their own personal page. They can modify all the information by clicking on the pen icon, they can upload a new profile image by clicking on the camera icon, they can upload a new CV by clicking on the upload icon but also they can download the current one, and they can add a new competence by clicking on the button.



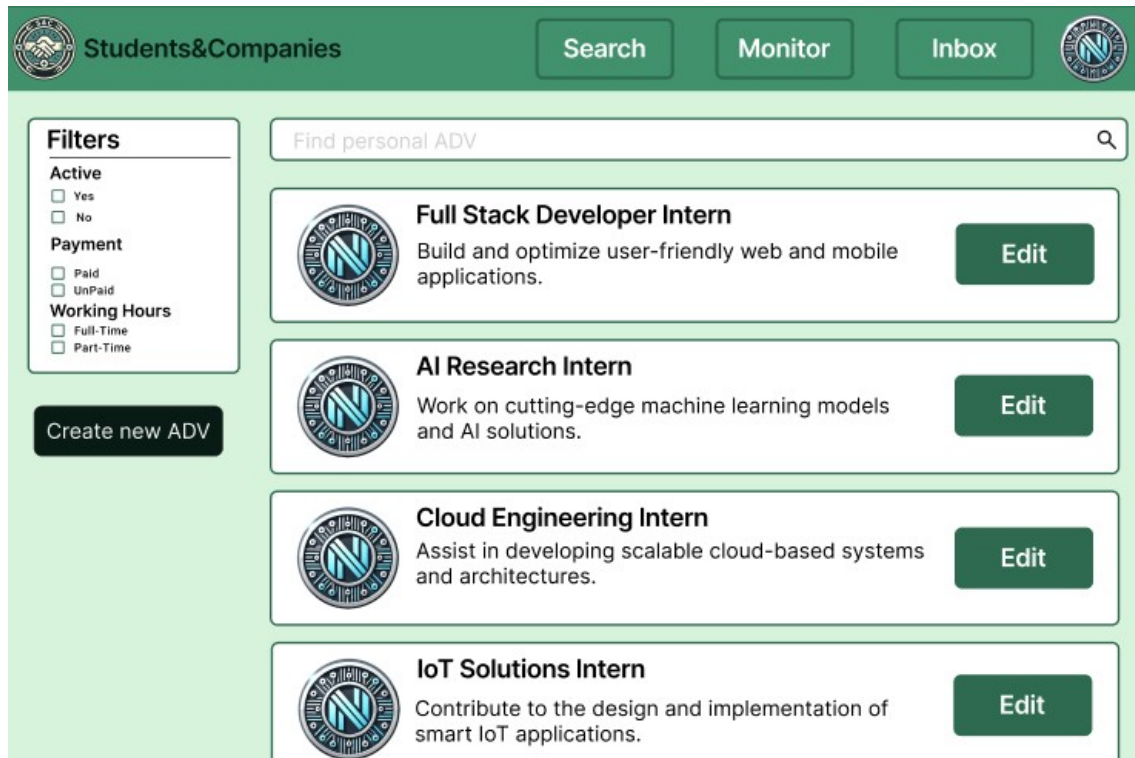
### 3.2.5. Company Profile Page



**Figure 3.7:** UI: Company Profile Page

This is a company's profile seen by students. All the company's active ADVs are shown in the boxes on the left, they are clickable and bring the user to the ADV page. On the right, relevant comments by students are shown.

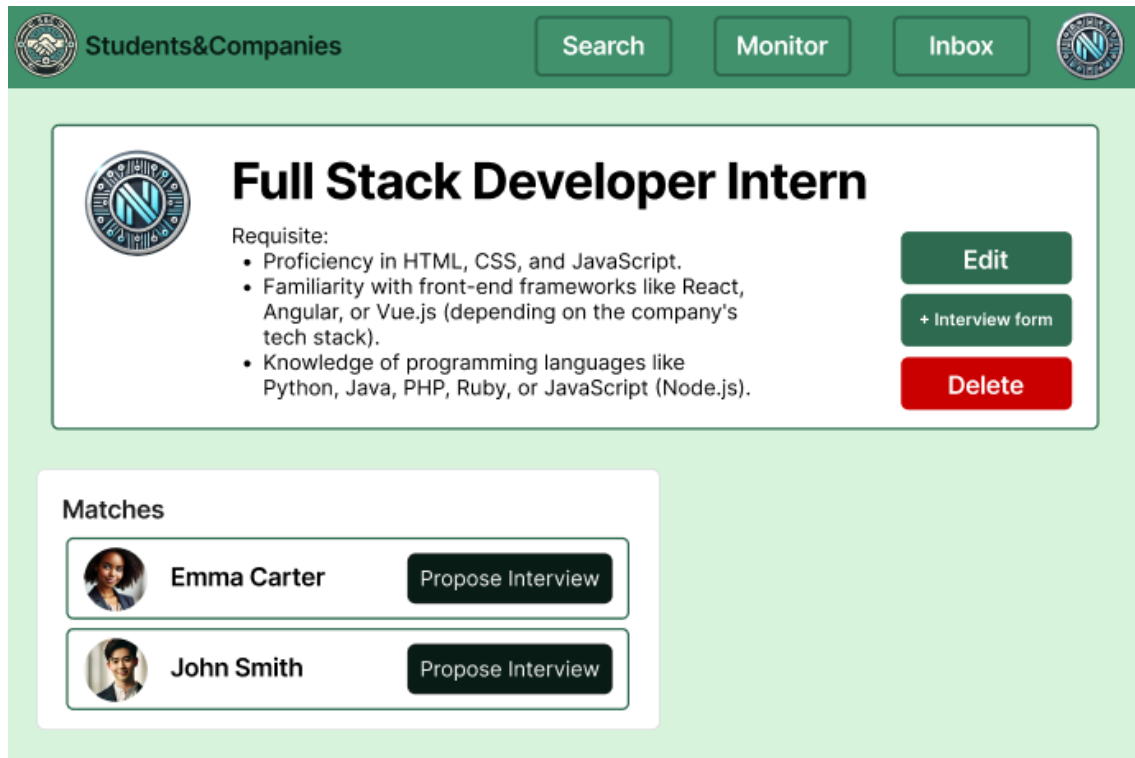
### 3.2.6. Company's ADVs Page



**Figure 3.8:** UI: Company's ADVs Page

When a Company is logged in, the representative can reach the ADVs Page from the personal profile. A new ADV can be created by clicking on the button, and the other ones can be modified by clicking on the corresponding "Edit" button. ADVs can be found via the search bar and/or applying filters. The Interview Form can be created by clicking on the "Edit" button.

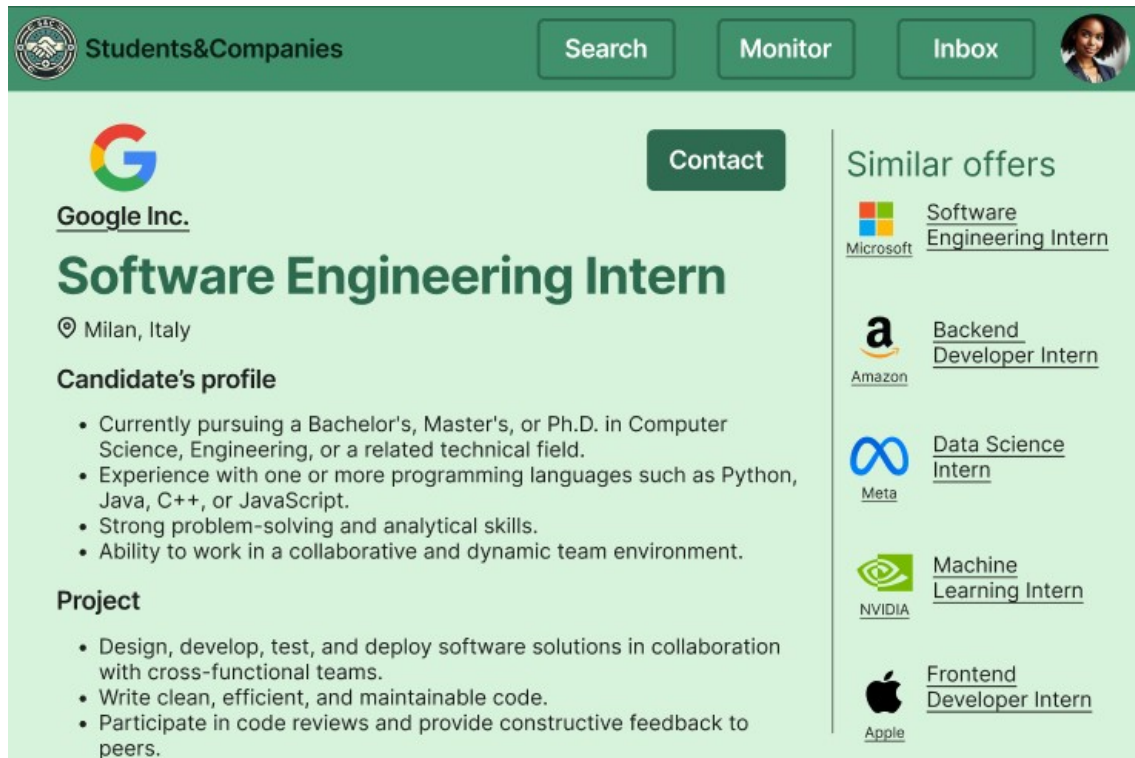
### 3.2.7. Company's ADV Page



**Figure 3.9:** UI: Company's ADV Page

When a Company is logged in, the representative can visualize ADV Page from the personal profile. The ADV can be edited by clicking on the "Edit" button, the representative can add a form to help the evaluation of the candidates during the interview by clicking on "+ Interview Form", or can delete the advertisement by clicking on "Delete" button. From this page the representative can also see all the students that has done a match with this Adv and setting up an interview with them pressing on "Propose Interview" button near the candidate name.

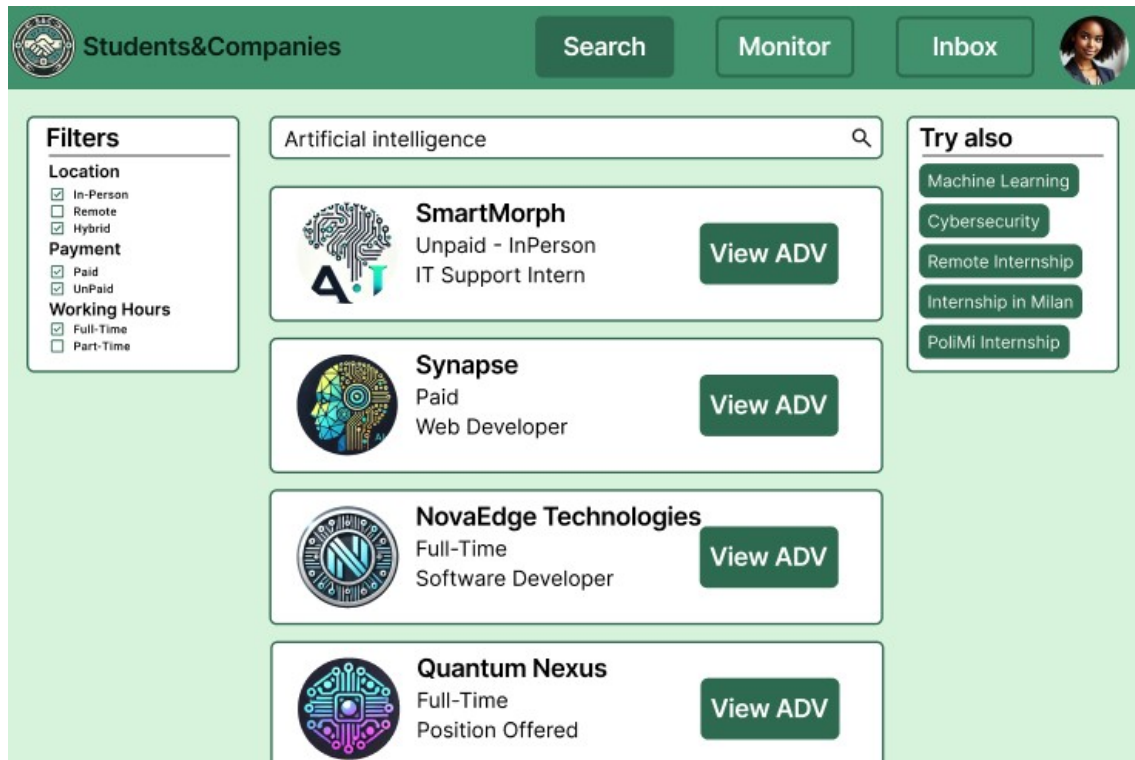
### 3.2.8. ADV Page



**Figure 3.10:** UI: ADV Page

This is the ADV Page seen by students. It contains all the information about the Internship position (all viewable by scrolling the page). By clicking on the company's logo or the name, the student is redirected to the company's profile. A student can contact the company regarding this ADV by clicking on the "Contact" button. On the left links to similar offers are shown.

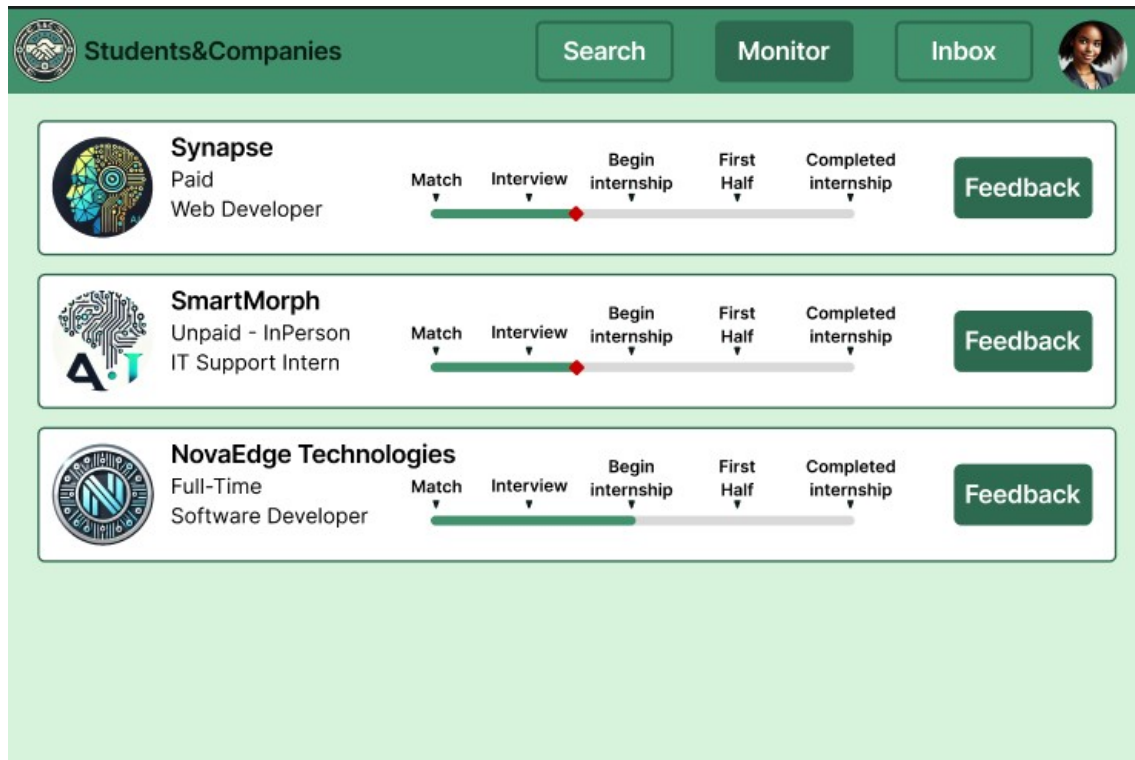
### 3.2.9. Search Page



**Figure 3.11:** UI: Search Page

The Search Page contains a search bar where the user (in this case a student) can insert key words. The user can also use the filters on the left or try the suggested key words on the right. When clicking on the "View ADV" button the student is redirected to the ADV page, while by clicking on the company's name they are redirected to the company's profile.

### 3.2.10. Monitor Page



**Figure 3.12:** UI: Monitor Page

The Monitor Page contains the match done, in this case it contains the ADV with which the student matched. For each ADV the student can see the progress and click on the "Feedback" button to leave one. The companies Monitor Page first has the list of the ADVs, when clicking on one of them the company is redirected to the monitor page of that ADV so they will see a list of student that have match with that adv, just as the image above.

## 3.2.11. Inbox Page

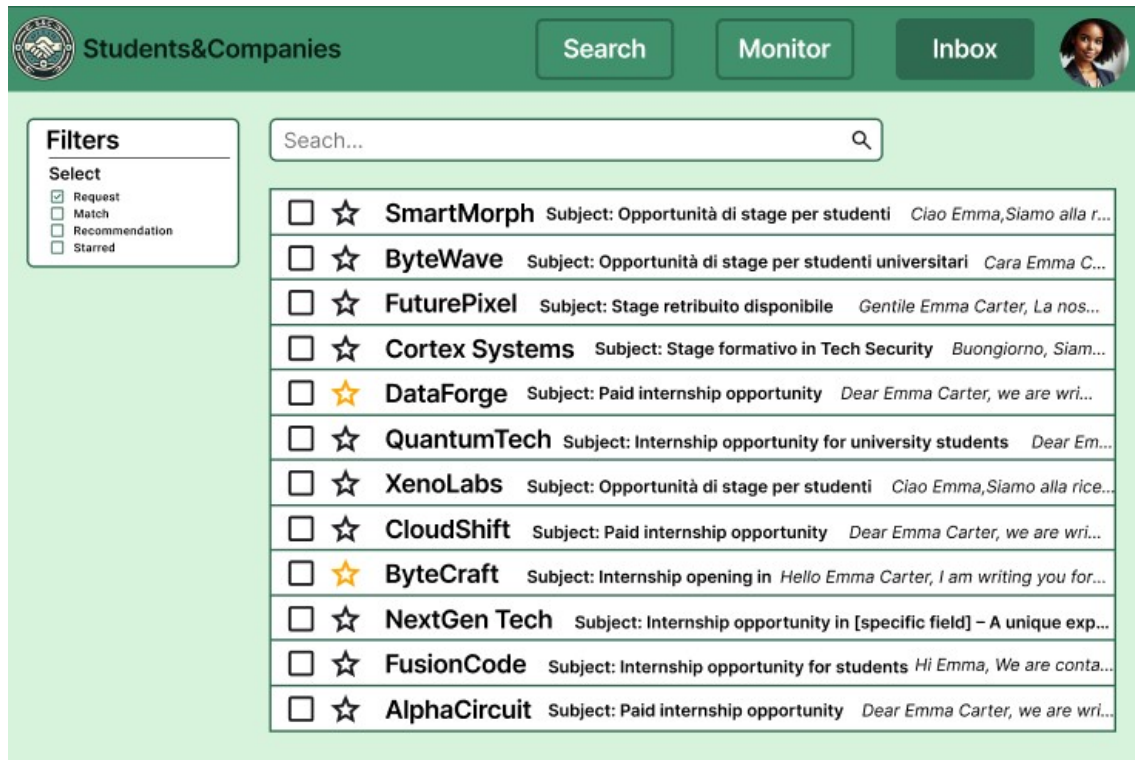


Figure 3.13: UI: Inbox Page

This is the Inbox Page, it works as a standard mail box.





# 4 | Requirements Traceability

- **Registration Manager**

- [R1] S&C allows Students to sign up to the platform through their university credentials using SSO access.
- [R3] S&C allows Companies to sign up to the platform by asking the representative of the company to insert their company email, the VAT number of the company, a Certificate of registration at the Chamber of Commerce, and a letter of authorisation or official assignment signed by a CEO or a HR referent. They have to choose a password.

- **Login Manager**

- [R2] S&C allows Student to login via their university credentials.
- [R4] S&C allows Companies to login to the platform via password and username (created by the system).

- **Profile Manager**

- [R5] S&C allows Students to upload the PDF of their CV (both for the first time and other times if there have been some changes).
- [R6] S&C allows Students to insert new experiences, skills, and attitudes on their profile.
- [R7] S&C allows Students to edit or delete experiences, skills, and attitudes present on their profile.

- **Adv Manager**

- [R8] S&C allows Companies to create an advertisement regarding an internship position they are offering, explaining the project of the internship and the terms offered.
- [R9] S&C allows Companies to modify and remove an advertisement posted on their page.

- **Search Manager**

- [R10] S&C allows Students to use a research bar to find appropriate internship positions by inserting specific requirements about the project and/or the terms offered.
- [R11] S&C allows Companies to use a research bar to find appropriate candidates for their internship positions by inserting specific experiences, skills and/or attitudes the candidates should have.

- **Recommendation Manager**

[R12] S&C informs students when an internship that might interest them becomes available.

[R13] S&C informs companies about the availability of a student whose CV corresponds to their needs.

- **Request Manager**

[R14] S&C allows Students to contact a company they are interested in.

[R15] S&C allows Companies to contact Students they are interested in.

- **Inbox Manager**

[R16] S&C informs a Student when a Company shows interest in their profile by sending the Student the Company's offer. The Student can accept or decline the offer.

[R17] S&C informs a Company when a Student shows interest in its offer by sending to the Company the Student's CV, the Company can accept or decline the offer.

[R18] S&C communicates to Students or Companies if their requests are successful.

- **Match Manager**

[R18] S&C communicates to Students or Companies if their requests are successful.

[R19] S&C allows Companies to setup an interview with the candidates.

- **Interview Manager**

[R19] S&C allows Companies to setup an interview with the candidates.

[R20] S&C allows Companies to write a form with all the aspects that will be considered during the interviews.

[R21] S&C allows Companies to modify or delete the interview form created.

[R22] S&C allows Companies fill the form during the interview.

- **Feedback Manager**

[R23] S&C allows Students to give personal feedback during the selection period about the Company's selection method or during the internship period about the internship itself.

[R24] S&C allows Companies to give feedback after the interview or during the internship about a single Student.

- **Monitor Manager**

[R25] S&C offers Students the possibility to monitor their matchmaking process with a Company (state of the matchmaking, scheduled interview, interview results).

[R26] S&C offers Companies the possibility to monitor their matchmaking process with a Student.

[R27] S&C offers Students and Companies the possibility to monitor the state of the internship.

# 5 | Implementation, Integration and Test Plan

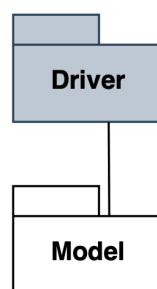
## 5.1. Overview and Implementation Plan

The Implementation, Integration and Test phases will follow a **Bottom-Up** strategy. This approach consists in develop and evaluate the components individually, and then integrate them into a larger system. Working with micro-services, this strategy will be followed starting from the low level components (model, login, and registration) and then continuing building the entire ecosystem around them. At each step some driver will be created in order to test the components. This strategy will allow developers to keep tracks of eventual bugs or errors, and furthermore will allow a parallel developing workflow.

## 5.2. Integration Strategy

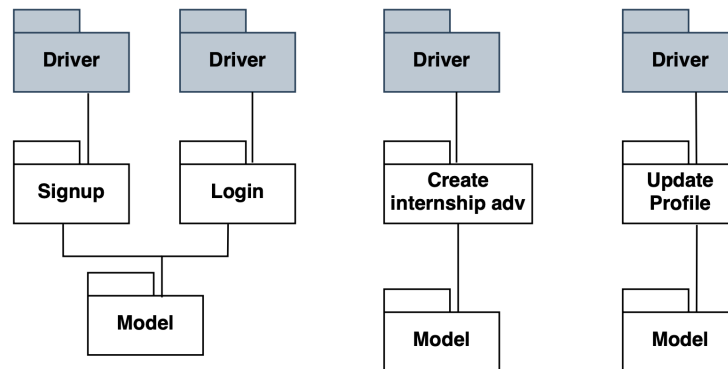
The implementation will be executed in a bottom-up way, starting from the basic feature and component, and after them, with their related test, the above components will be implemented.

The implementation starts with the model component and its test.



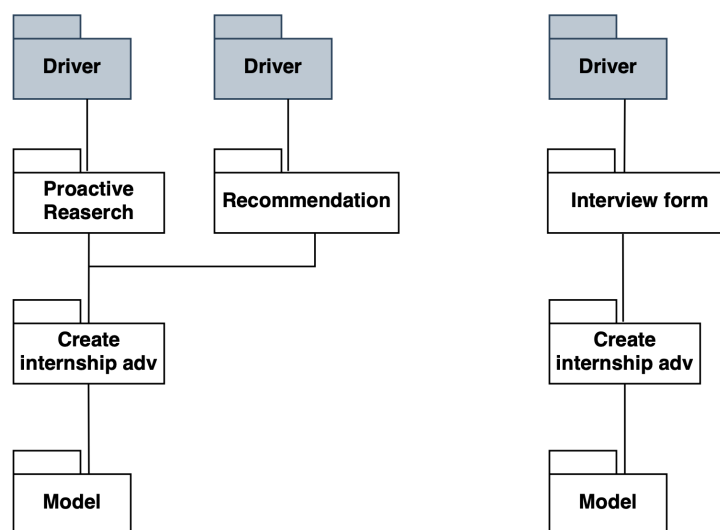
**Figure 5.1:** Model implementation

The implementation continues with the Login and Signup components, with a particular attention to the university and the Revenue Agency API interaction. At the same time the creation of an Adv of an internship and the Update of a profile can be implemented.



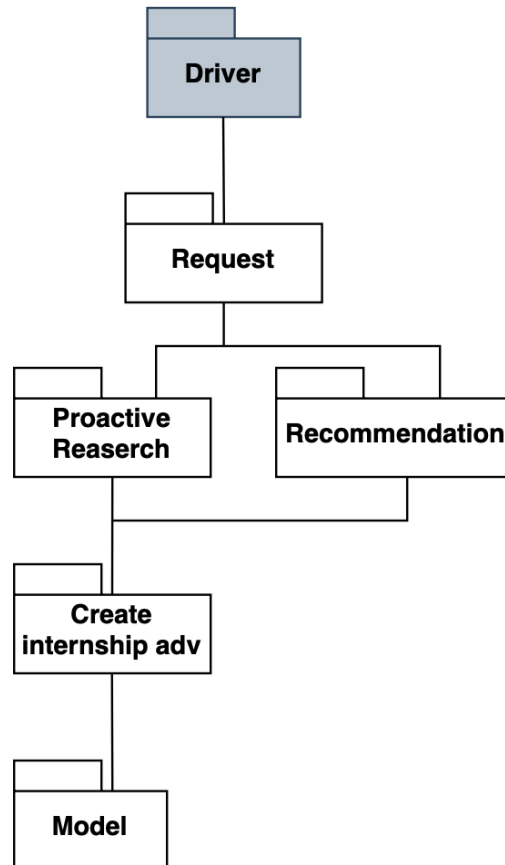
**Figure 5.2:** Login, Signup, creation of an adv and the update of a profile implementation

After Adv creation, Proactive research and Recommendation can be implemented, as long as the creation of the Interview form.



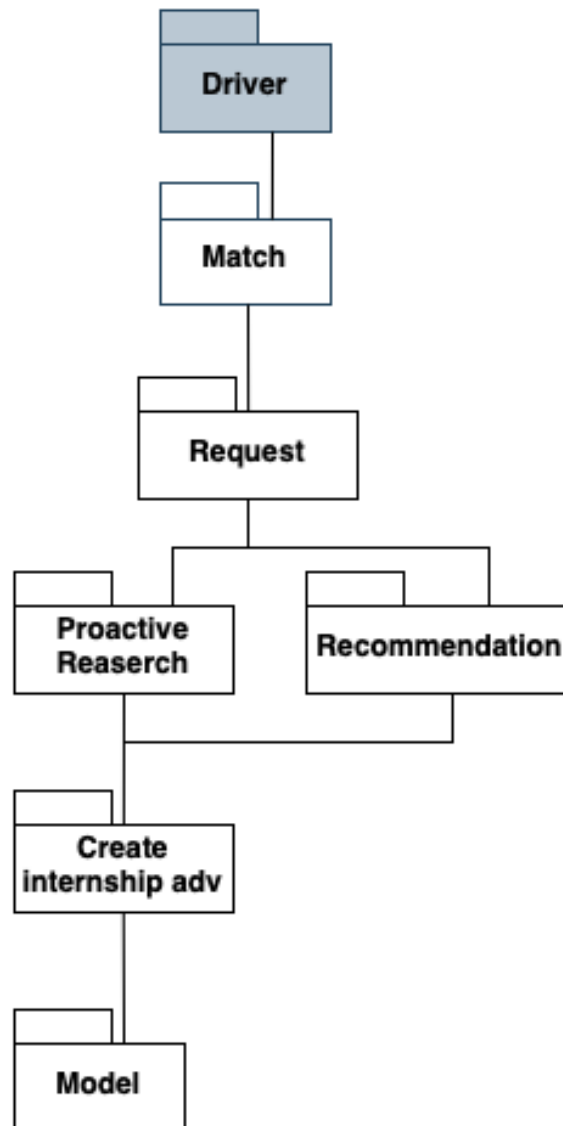
**Figure 5.3:** Proactive Research, Recommendation and interview form implementation

The implementation continues with the Request component, which is directly linked to the Recommendation and to the Proactive Research.



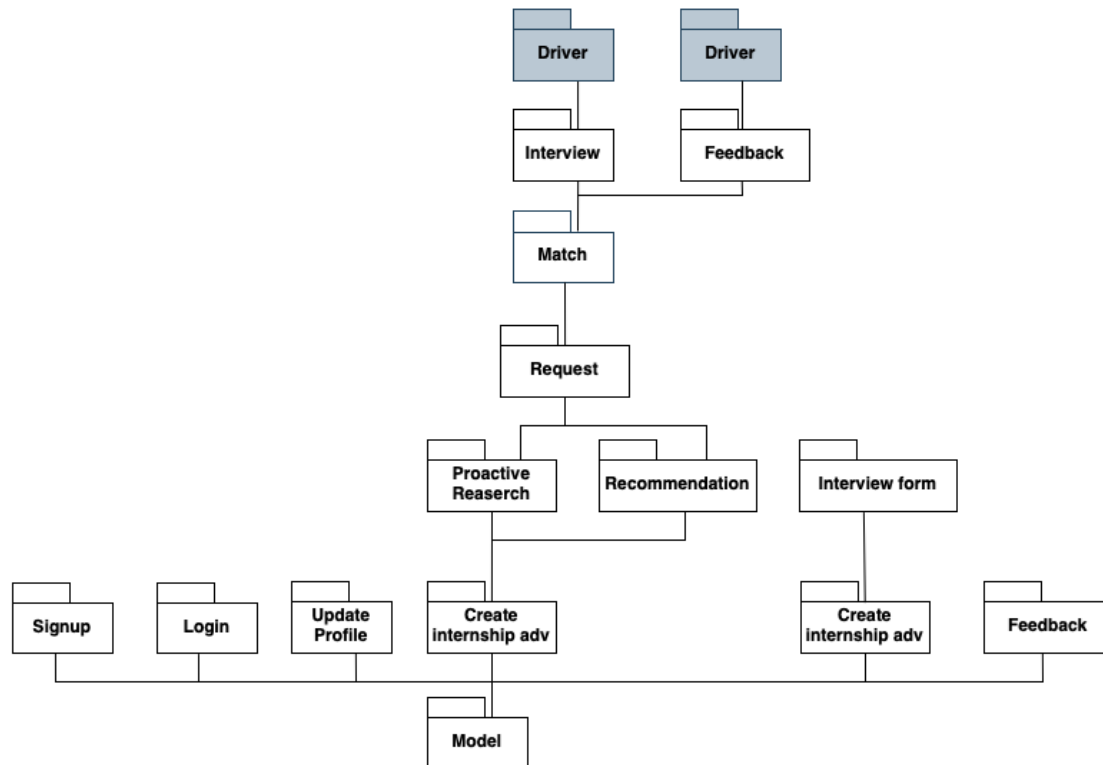
**Figure 5.4:** Request implementation

After Request implementation, the Match component can be implemented.



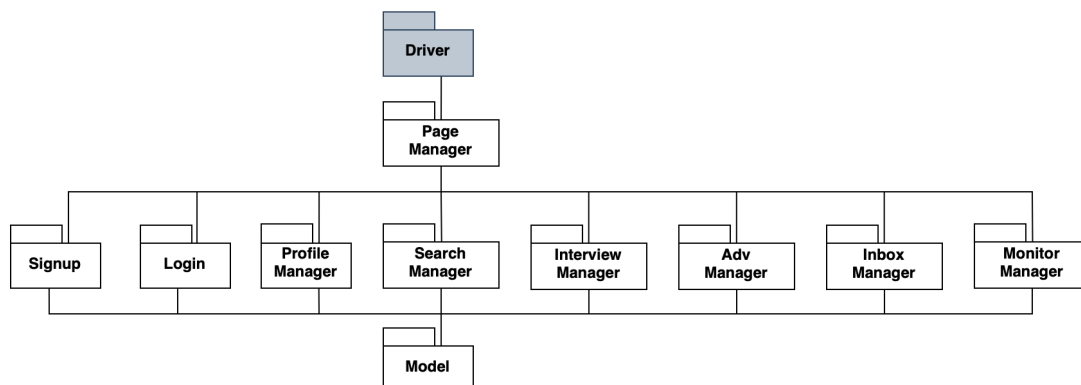
**Figure 5.5:** Match and Feedback implementation

Then the Interview is implemented and at the same time the Feedback section can be implemented and, during these final implementations, the system can be joined in order to test it in its entirety.



**Figure 5.6:** Interview and Feedback implementation and join all the system

The latter implementation section will involve the Page Manager and all the pages of the Application.



**Figure 5.7:** View implementation

### 5.3. System Testing Strategy

Following the Bottom-Up strategy, every component will be tested with a Driver during the implementation phase but also during the integration phase, in order to assure its functionality independently but also its communication features with all the other components of the system. The test that will be performed are:

- **Functional test:** This test sections will ensure that every components fulfill the designated job, accordingly with the system goals, requirement and functionality of the system.
- **Performance test:** This type of test are designated to finds bottlenecks and identify some critical situation that may occur during an heavy workload periods. This tests are also useful to get some idea of possible future update and enhancements.
- **Load test:** This type of test will cover the discover of memory problems such as leak or overflows.
- **UI test:** This type of test will evaluate the Usability of the System for all the users, and will establish if the System works on different type of browser.



## 6 | Effort Spent

This is an overview of the number of hours spent on each chapter. Most of the work was done in pairs. However, an approximation of the number of hours contributed by each member is provided.

-	<b>Irene Lo Presti</b>	<b>Matteo Lussana</b>
Introduction	2h	0.5h
Architectural Design	18h	17h
User Interface Design	15h	13h
Requirements Traceability	0.5h	0.5h
Implementation, Integration, and Test Plan	0.5h	3h



# 7 | References

## 7.1. References

- Specification document: Assignment RDD AY 2024-2025.
- Slides of the course "Software Engineering 2" held at Politecnico di Milano by Professor Rossi (a.y. 2024-25).
- Some definitions from the Definitions, Acronyms, Abbreviations section are taken from researches done on the Internet.

## 7.2. Used Tools

- GitHub for project versioning and sharing.
- $\text{\LaTeX}$  and *Overleaf* as editor for writing this document.
- *draw.io* for diagrams design.
- *Notion* for collaborative organisation.



## List of Figures

2.1	Overview of the S&C system . . . . .	9
2.2	Component view: Low Level Diagram . . . . .	10
2.3	Deployment Diagram . . . . .	12
2.4	Architecture nodes in Deployment Diagram . . . . .	13
2.5	Database Node in Deployment Diagram . . . . .	13
2.6	Back-End Node in Deployment Diagram . . . . .	14
2.7	Deployment Diagram: Login and Sign up . . . . .	15
2.8	Deployment Diagram: Profile Management . . . . .	15
2.9	Deployment Diagram: ADV Management . . . . .	16
2.10	Deployment Diagram: Interview Management . . . . .	16
2.11	Deployment Diagram: Search Management . . . . .	17
2.12	Deployment Diagram: Recommendation Management . . . . .	17
2.13	Deployment Diagram: Match Management . . . . .	18
2.14	Deployment Diagram: Feedback Management . . . . .	18
2.15	Deployment Diagram: Monitor and Inbox Management . . . . .	19
2.16	SD: Student Sign up . . . . .	20
2.17	SD: Student Login . . . . .	21
2.18	SD: Company Sign up . . . . .	22
2.19	SD: Company Login . . . . .	23
2.20	SD: Upload Student CV . . . . .	23
2.21	SD: Add new competence . . . . .	24
2.22	SD: Edit competence . . . . .	24
2.23	SD: Create New ADV . . . . .	25
2.24	SD: Edit ADV . . . . .	25
2.25	SD: Delete ADV . . . . .	26
2.26	SD: Proactive Internship Search . . . . .	27
2.27	SD: Proactive Candidate Search . . . . .	28
2.28	SD: Recommendation . . . . .	29
2.29	SD: Accept internship request (Student) . . . . .	30
2.30	SD: Reject internship request (Student) . . . . .	30
2.31	SD: Accept internship request (Company) . . . . .	31
2.32	SD: Reject internship request (Company) . . . . .	31
2.33	SD: Check request outcome (Student) . . . . .	32
2.34	SD: Check request outcome (Company) . . . . .	32
2.35	SD: Propose a date for an interview . . . . .	33
2.36	SD: Accept a date for an interview . . . . .	33
2.37	SD: Reject a date for an interview . . . . .	34

2.38	SD: Creating form for interview . . . . .	34
2.39	SD: Deleting form for interview . . . . .	34
2.40	SD: Modify form for interview . . . . .	35
2.41	SD: Fill the form during interview . . . . .	35
2.42	SD: Feedback . . . . .	36
2.43	SD: Monitoring . . . . .	36
3.1	User Interface Overview . . . . .	43
3.2	UI: Login Page . . . . .	44
3.3	UI: Sign up Page . . . . .	45
3.4	UI: Home Page . . . . .	46
3.5	UI: Student Profile Page . . . . .	47
3.6	UI: Student Personal Profile Page . . . . .	48
3.7	UI: Company Profile Page . . . . .	49
3.8	UI: Company's ADVs Page . . . . .	50
3.9	UI: Company's ADV Page . . . . .	51
3.10	UI: ADV Page . . . . .	52
3.11	UI: Search Page . . . . .	53
3.12	UI: Monitor Page . . . . .	54
3.13	UI: Inbox Page . . . . .	55
5.1	Model implementation . . . . .	59
5.2	Login, Signup, creation of an adv and the update of a profile implementation . .	60
5.3	Proactive Research, Recommendation and interview form implementation . . .	60
5.4	Request implementation . . . . .	61
5.5	Match and Feedback implementation . . . . .	62
5.6	Interview and Feedback implementation and join all the system . . . . .	63
5.7	View implementation . . . . .	63