Software Engineering 2 - RASD & DD Project- Prof. Elisabetta Di Nitto

Students & Companies

A platform for Seamless Internship Coordination

Acquadro Patrizio, Colosio Giacomo, Drugman Tito Nicola

Academic Year 2024-2025

POLITECNICO
MILANO 1863

Documentation on GitHub

S&C

RASD

Requirement Analysis and Specification Document

POLITECNICO
MILANO 1863

S&C

Goals

01 **Students would like to find and be matched with internships** that align with their skills, experiences and professional interest.

02 **Company tutors would like to reach and attract students** who have the right profile and are interested in their internship offering.

07 **Company tutors would like to manage their selection process** and track the status of applications and selections.

08 **University tutors would like to monitor their students' internships** and to address complaints or issues that might arise.

POLITECNICO
MILANO 1863

S&C

Scenarios and Scope

Lorenzo (University Student)

David (Company Tutor)

Claudio (University Tutor)

POLITECNICO
MILANO 1863

S&C

Scenarios and Scope

Student	A-4	The Student Credential Recovery
Student	A-5	The Student and the Chatbot Interaction
Student	A-6	Student Language Change
Student	A-7	The Student Enters the S&C Homepage
Student	A-8	The Student Searches for an Internship
Student	A8-1	Recommendation List in Matchmaking
Student	A8-2	Custom Search in Matchmaking
Student	A-9	The Student Navigates the Monitoring Page
Student	A-10	The Student Monitors the Progress of his Applications
Student	A-11	The Student Monitors the Status of Requests Sent to Him
Student	A-12	A Contact is Established - Selection Process (Student Overview)
Student	A-13	Lorenzo Monitors Active Stages
Student	A-14	The Student Completes the Final Evaluation Form
Student	A-15	The Student Navigates the Calendar Page
Student	A-16	The Student Interacts with the Messaging System
Student	A16-1	The Student Files a Complaint via Messaging
Company Tutor	B-1	The Company Tutor Opens the S&C Application
Company Tutor	B-2	Company Tutor Registration
Company Tutor	B2-1	Mandatory fields incomplete
[illegible]	[illegible]	[illegible]

Total scenarios = 63

Students -> 23

Company Tutors -> 23

University Tutors -> 17

POLITECNICO
MILANO 1863

S&C

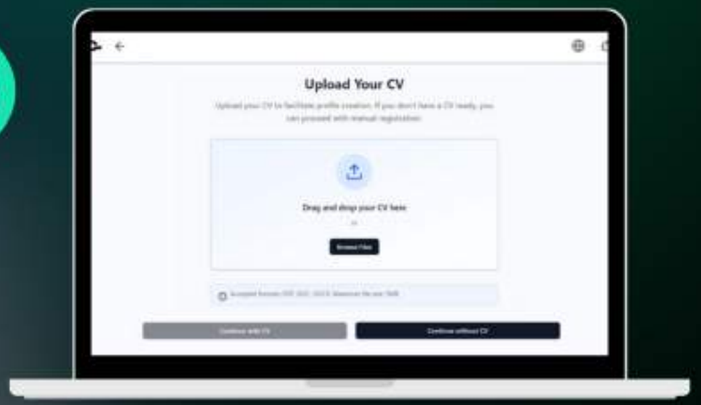
Scenarios and Scope

Scenario [A-2]: *Student Registration*

WP1

Students create their resumes with information about experiences, skills and attitudes.

F 3.3



POLITECNICO
MILANO 1863

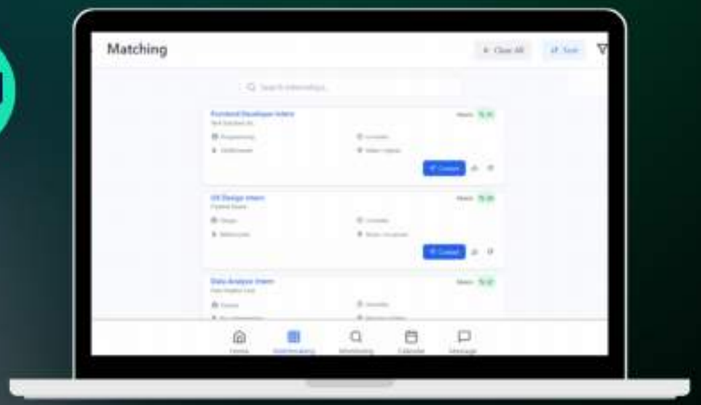
S&C

Scenarios and Scope

Scenario [A-8]: *The Student Searches for an Internship*

SP20 — The student browses internship offers without filters.

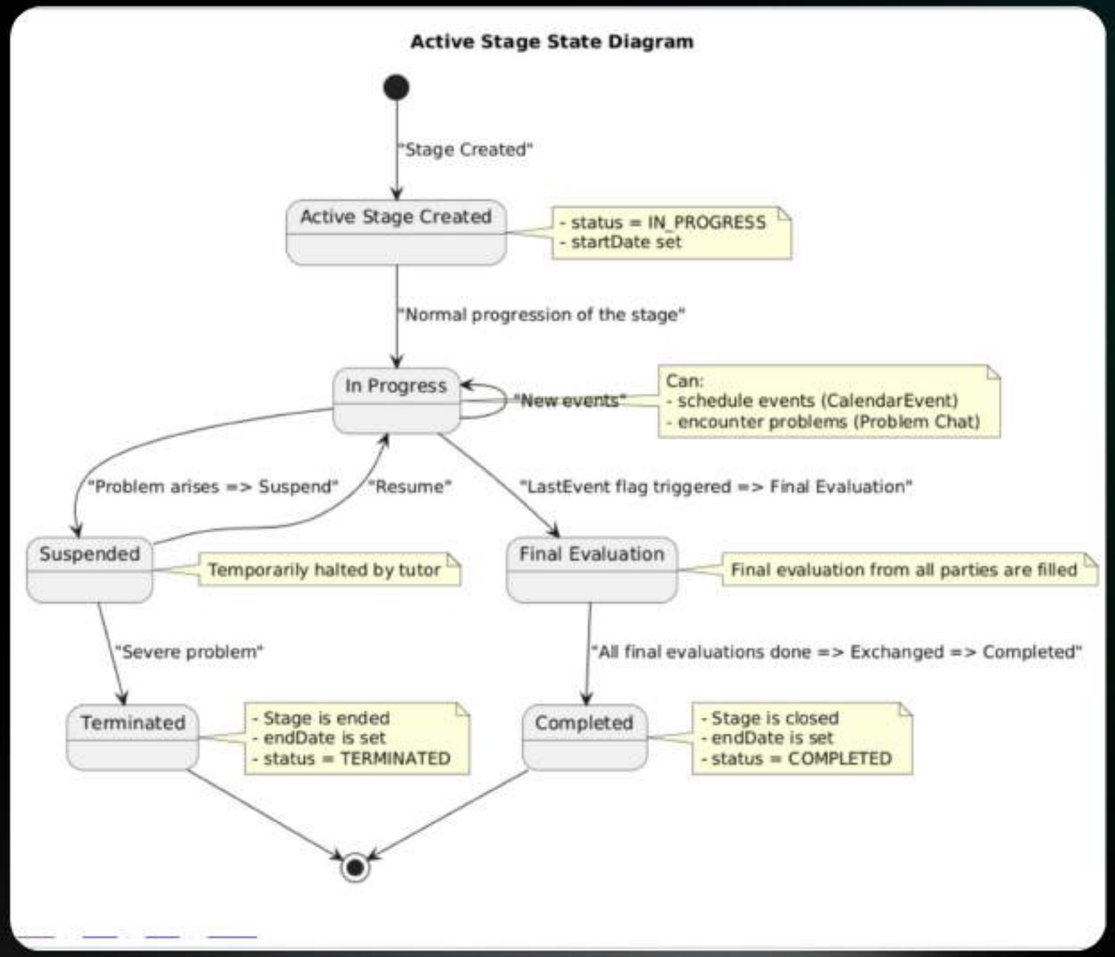
F 3.11



POLITECNICO
MILANO 1863

S&C

Class & State Diagrams



The Hearth of Analysis

Domain Class Diagrams = What

State Diagrams = How

UML Code for Design

PlantText

S&C

Class Diagrams

What They Are

Panoramic of the **entities** and **relations** in the system.

How We Built Them

1) From the **scenarios** we identified *entities* and *relations*.
2) Program in **UML.**

Enumerations

Enum, not subclasses

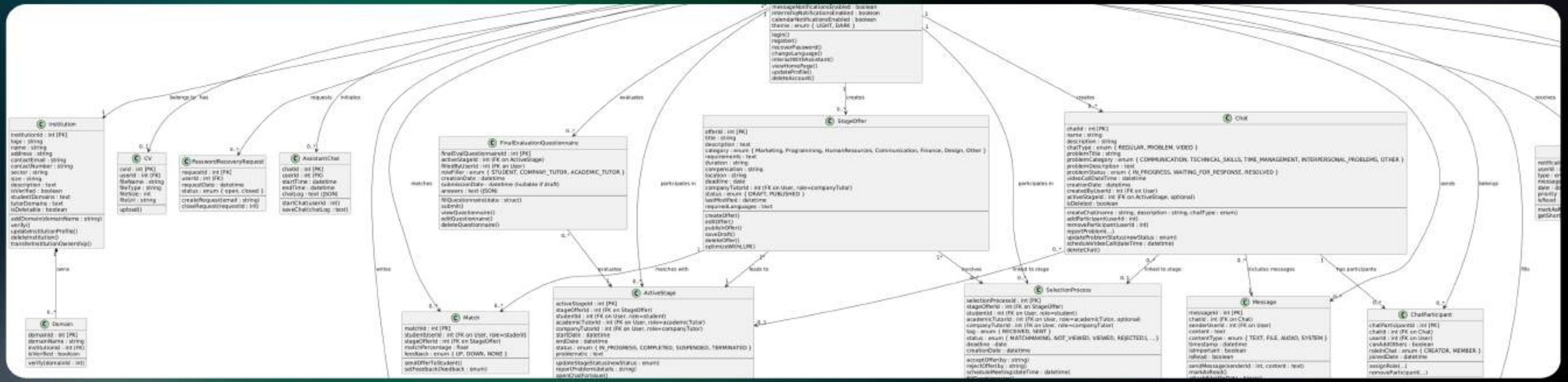
- Roles.
- Status.

TARGET ACHIEVED ✓

POLITECNICO
MILANO 1863

S&C

Class Diagrams - Result



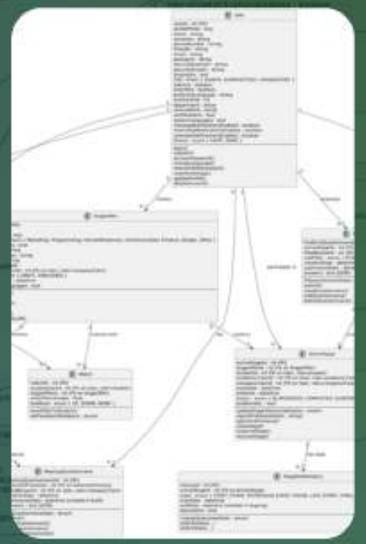
POLITECNICO
MILANO 1863

S&C

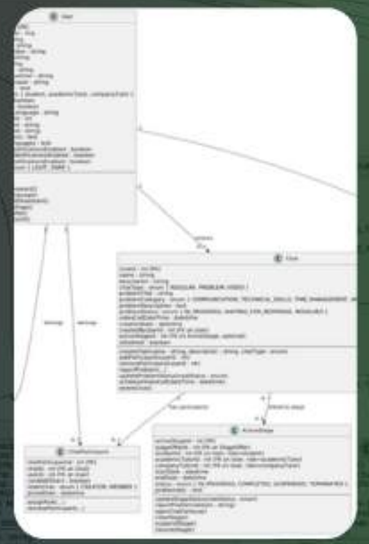
Class Diagrams - Results

Authentication

Match&Monitor



Chat&Calendar



POLITECNICO
MILANO 1863

S&C

State Diagrams

What They Are

Analyzing critical and complex **processes** with multiple **states**.

Usage

Easy understanding of their implementation.

How We Built Them

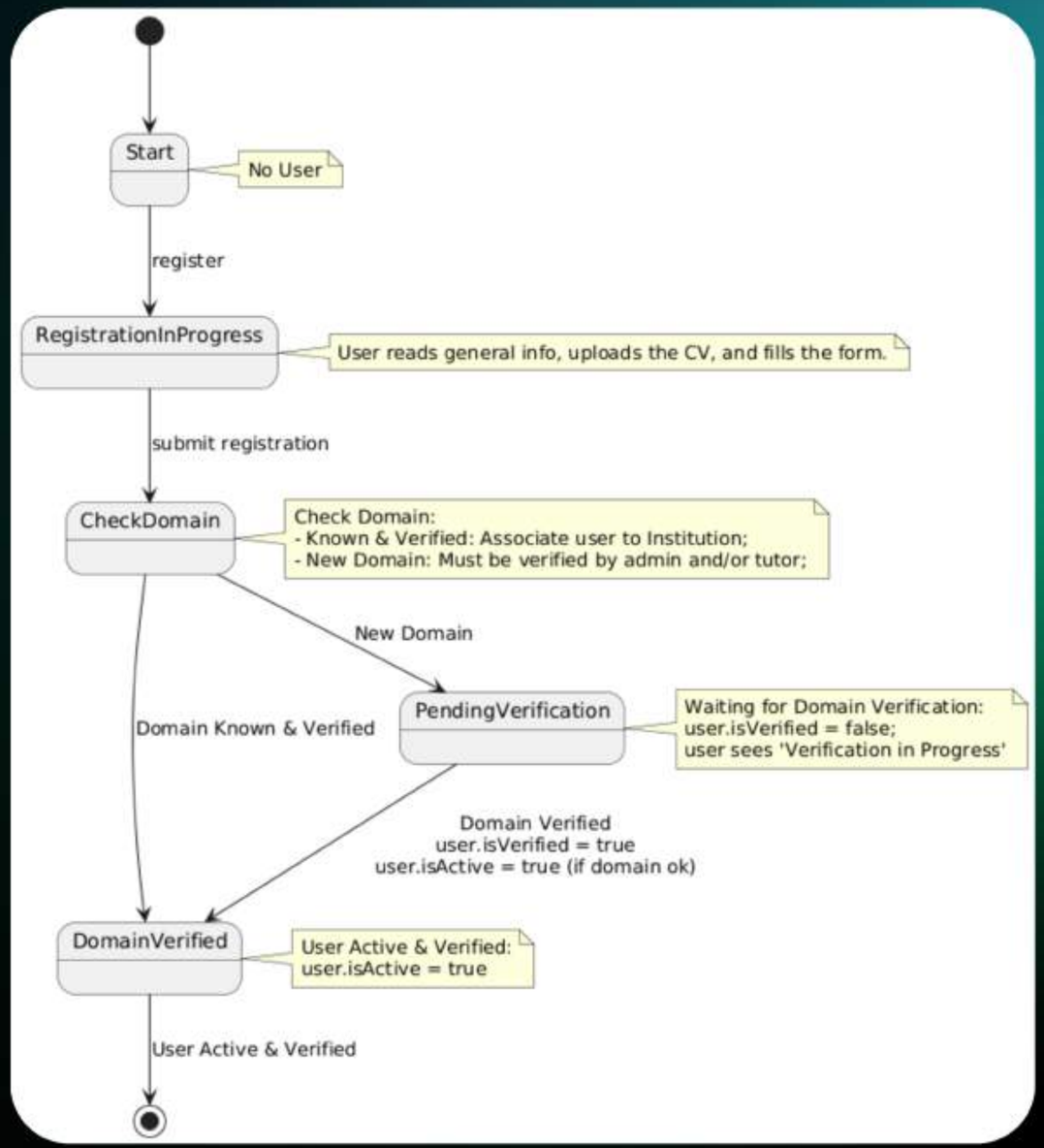
1) From the **scenarios** we identified *states*.
2) Program in **UML**.

TARGET ACHIEVED ✓

POLITECNICO
MILANO 1863

S&C

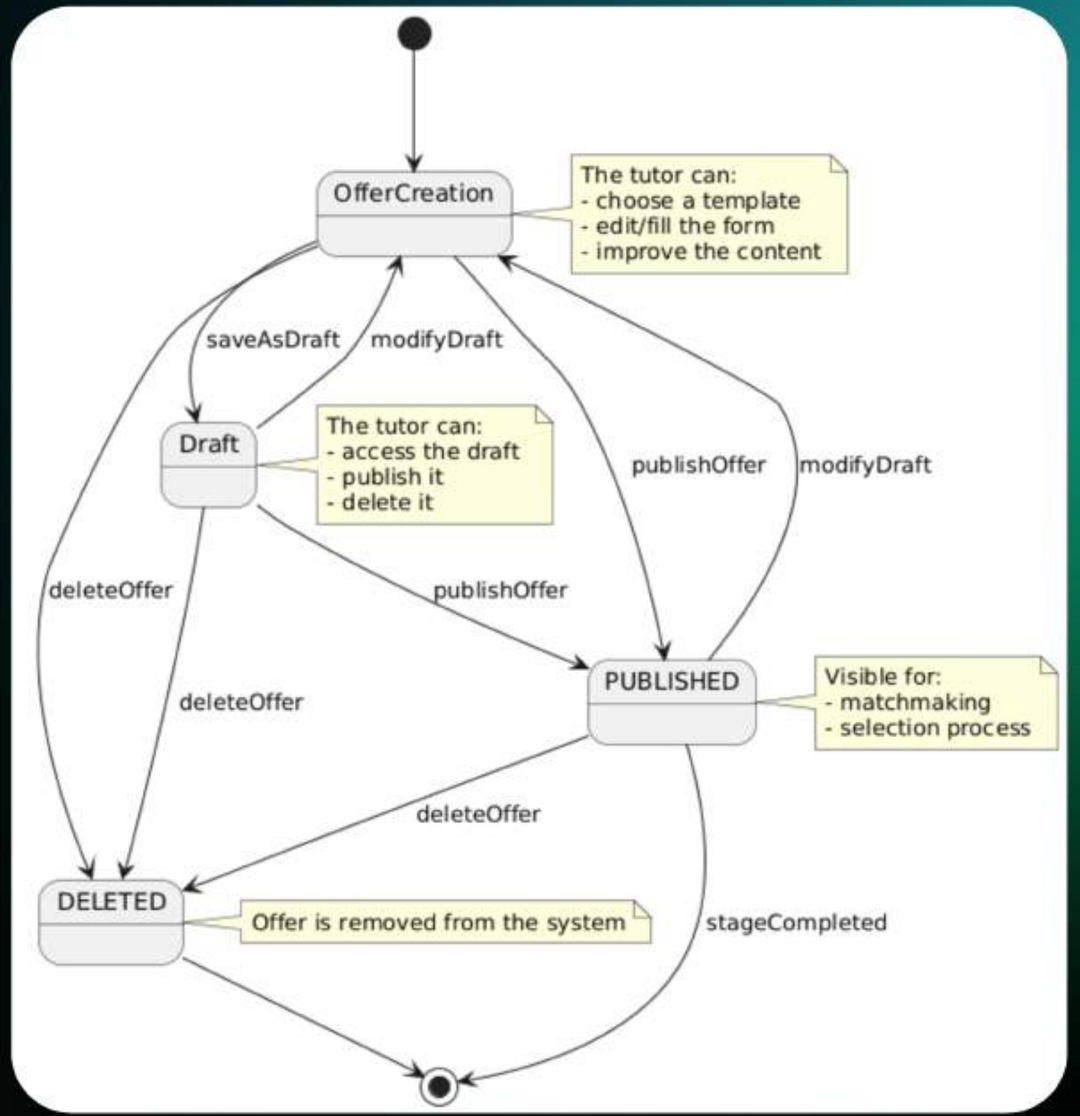
Domain Association



POLITECNICO
MILANO 1863

S&C

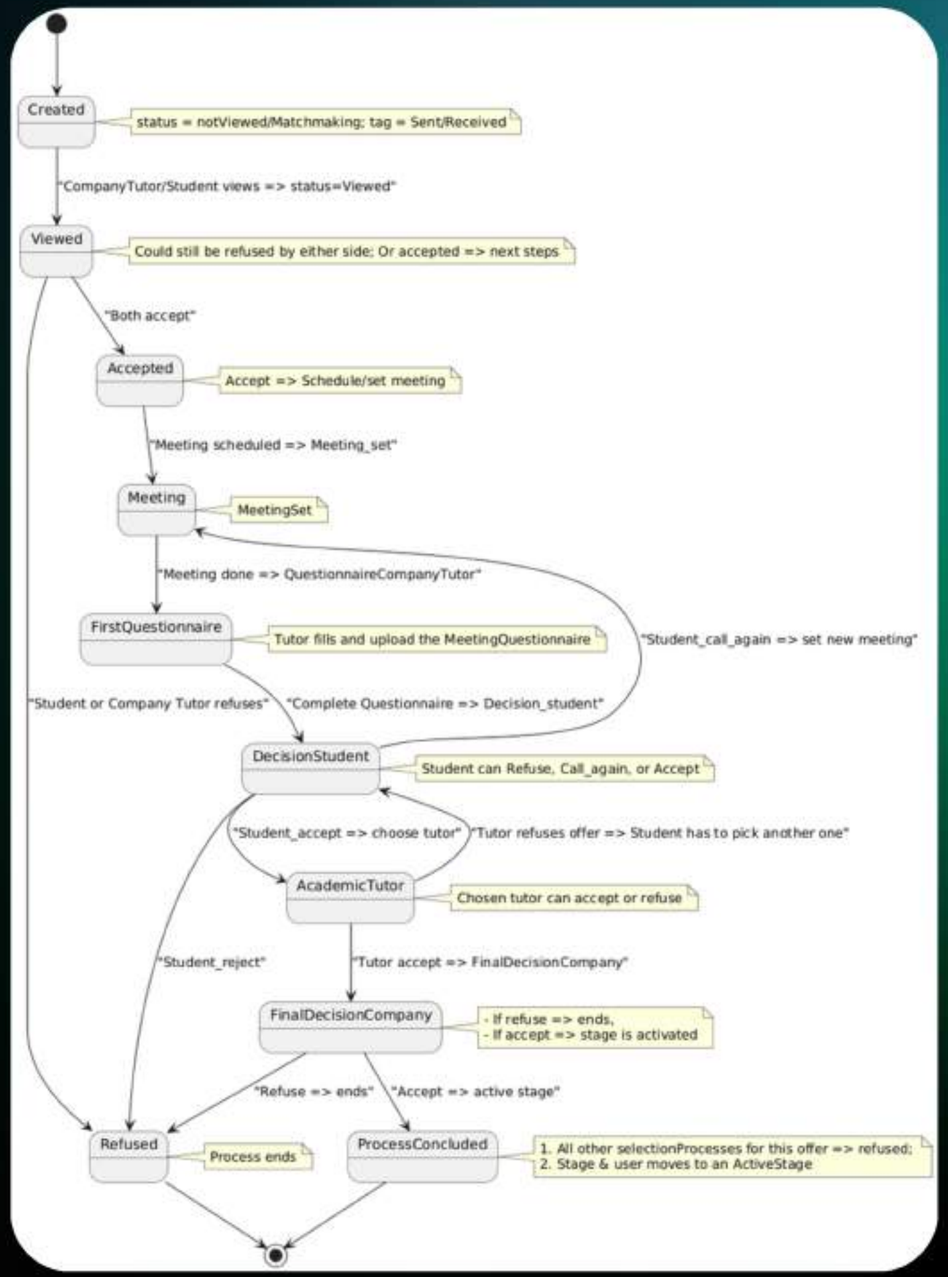
Stage Lifecycle



POLITECNICO
MILANO 1863

S&C

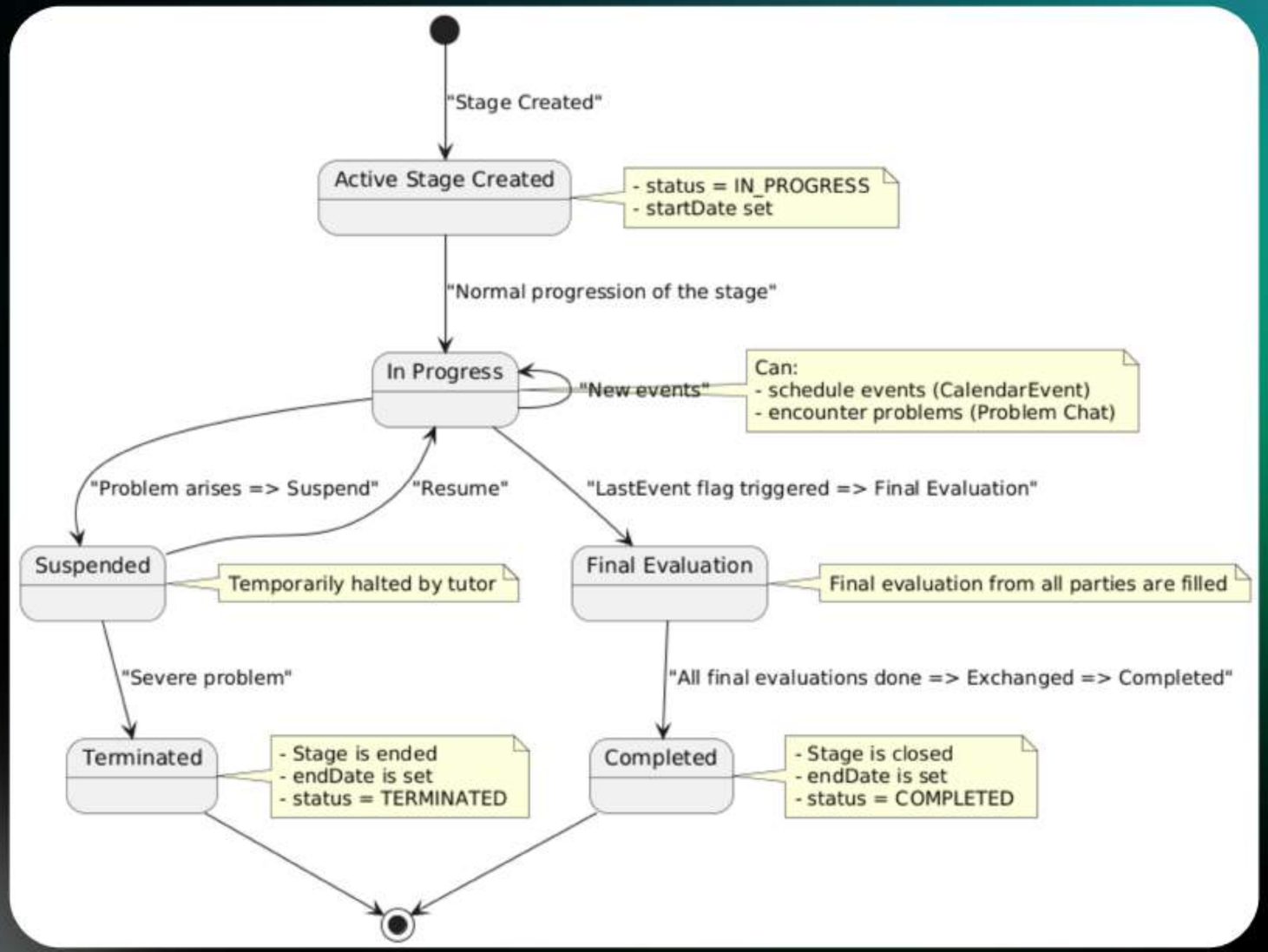
Selection Process



POLITECNICO
MILANO 1863

S&C

Active Stage



POLITECNICO
MILANO 1863

S&C

Use Cases & Sequence Diagrams

Combing what we have seen so far:

We have Obtain:

USE CASES

SEQUENCE DIAGRAMS

POLITECNICO
MILANO 1863

S&C

Use Cases & Sequence Diagrams

[UC3]: CV Upload

Name	CV Upload
Actors	• User • System
Entry Condition	The user has clicked the "Create Your Profile" button on the previous interface and reached the CV upload page.
Event flow	(a) The system displays the CV upload page. (b) The user performs one of the following actions: - Uploads a CV, by clicking on the "Browse Files" button or drags and drops a file into the upload area. - Continues without uploading a CV, so the user clicks the "Continue without CV" button. - Changes the system language using the "Change Language" option. - Opens the assistant interface for guidance through the "Assistant" button. - Returns to the previous screen by clicking "Go Back" . (c) Based on the user's action, the system: - Skips the upload process and redirects the user to the next screen. - Redirects the user to the appropriate interface (for language change, assistant support, or the previous screen) - Validates and processes the uploaded CV, so it checks if the file meets the format and size requirements (PDF, DOC, DOCX with max size 5MB). (d) According to the result of the validation: - If the CV is valid, the system uploads the file, displays a confirmation message and automatically redirects the user to the next screen. - If invalid, the system displays an error message prompting the user to upload a valid file.
Exit Condition	The user uploads a CV or continues without it, proceeding to the next screen, or navigates to a different screen using additional functionalities.
Exception	If the uploaded file fails validation (incorrect format or exceeds size limit), the system displays an error message, prompting the user to upload a valid file.

Table 3.3: Use Case: CV Upload

[UC13]: Matchmaking

Name	Matchmaking
Actors	• Student • Company Tutor • System
Entry Condition	The user (student or company tutor) has accessed the matchmaking screen via the navigation bar.
Event flow	(a) The system displays personalized recommendations based on the user's profile. For students, the system shows internships, while for company tutors it suggests student profiles. (b) The user interacts with the displayed recommendations and navigates the interface: - Filters results by criteria like skills, languages, or match percentage. - Sorts results by match percentage, suggestion date, or category. - Uses the advanced search feature or resets the criteria to return to default recommendations. - Selects a recommendation to access the detailed profile page. - Provides feedback (positive or negative) to improve the accuracy of future recommendations. - Selects an icon in the navigation bar for quick access to key sections. (c) The system processes the user's input by: - Updating the displayed recommendations based on the applied filters, sorting options, or search criteria. - Redirecting the user to the selected detailed profile when chosen. - Recording feedback provided by the user to adjust future recommendations. - Displaying the page related to the selected navigation bar icon.
Exit Condition	The user selects or interacts with an internship or student profile, refines the displayed results, or navigates to another screen via the navigation bar.
Exception	If the recommended internships or profiles fail to load due to a system error, an error message is displayed and the user remains on the current screen.

Table 3.13: Use Case: Matchmaking

[UC26]: Internship Creation and Management

Name	Internship Creation and Management
Actors	• Company Tutor • System
Entry Condition	The interface is accessed from the selection process screen.
Event flow	(a) The system displays the interface, which includes: - A menu with predefined or previously saved templates. - Editable fields for required details. (b) The company tutor interacts with the interface by: - Selecting a predefined or custom template to automatically populate fields. - Filling or modifying required fields manually. - Using "Improve" to optimize via an LLM. - Saving the current details as a draft for later editing by clicking the "Save as Draft" button. - Clicking "Save Template" to store the current configuration as a reusable template. - Navigating back to the previous screen without saving changes using the "Back" button. - Publishing the position to make it visible to students. - Delete the position, by clicking the bin button. (c) The system populates fields when a template is selected. (d) The system enhance the content of the internship. (e) The company tutor modifies the different fields. (f) The system validates and saves data when the "Publish Position" button is clicked. (g) The system publish the internship position. (h) The system sends notifications to students about the new position if published. (i) The system in alternative processes the tutor's actions by: - Saving the new template for reuse. - Redirecting the tutor back to the previous interface when the "Back" button is clicked. - Redirecting the tutor to the select page. - Saving the internship as a draft, for later modifications.
Exit Condition	The company tutor completes the internship creation, saves it as a draft, publishes it, or navigates away from the screen.
Exception	If required fields are incomplete or invalid, the system displays an error message, preventing the draft or publication from proceeding.

Table 3.26: Use Case: Internship Creation and Management

POLITECNICO
MILANO 1863

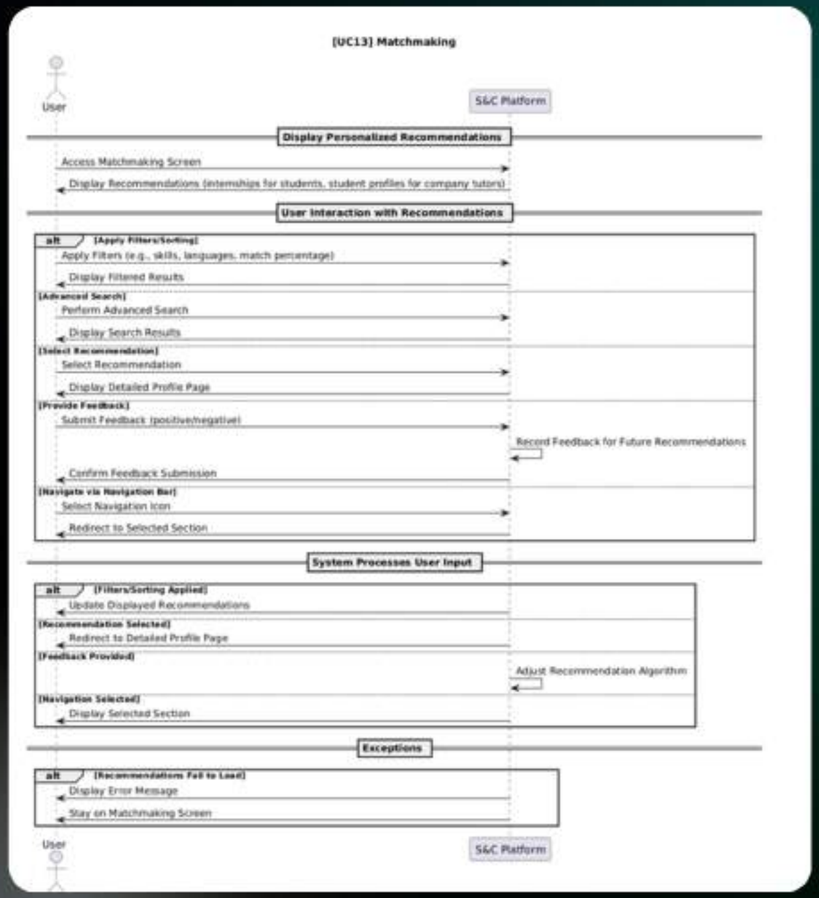
S&C

Use Cases & Sequence Diagrams

UC13]: Matchmaking

Name	Matchmaking
Actors	• Student • Company Tutor • System
Entry Condition	The user (student or company tutor) has accessed the matchmaking screen via the navigation bar.
Event flow	(a) The system displays personalized recommendations based on the user's profile. For students, the system shows internships, while for company tutors it suggests student profiles. (b) The user interacts with the displayed recommendations and navigates the interface: - Filters results by criteria like skills, languages, or match percentage. - Sorts results by match percentage, suggestion date, or category. - Uses the advanced search feature or resets the criteria to return to default recommendations. - Selects a recommendation to access the detailed profile page. - Provides feedback (positive or negative) to improve the accuracy of future recommendations. - Selects an icon in the navigation bar for quick access to key sections. (c) The system processes the user's input by: - Updating the displayed recommendations based on the applied filters, sorting options, or search criteria. - Redirecting the user to the selected detailed profile when chosen. - Recording feedback provided by the user to adjust future recommendations. - Displaying the page related to the selected navigation bar icon.
Exit Condition	The user selects or interacts with an internship or student profile, refines the displayed results, or navigates to another screen via the navigation bar.
Exception	If the recommended internships or profiles fail to load due to a system error, an error message is displayed and the user remains on the current screen.

Table 3.13: Use Case: Matchmaking



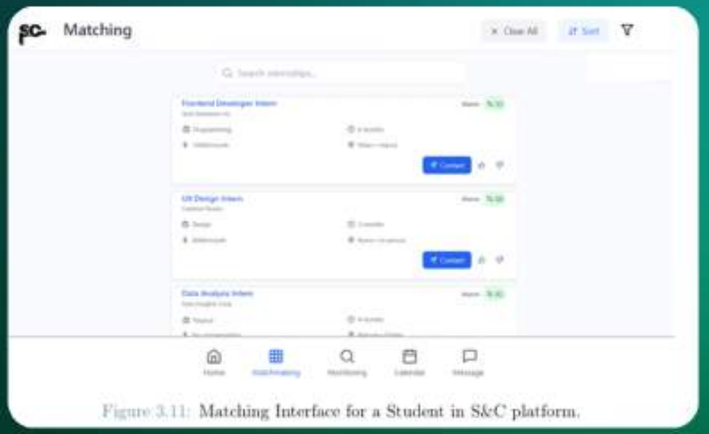


Figure 3.11: Matching Interface for a Student in S&C platform.

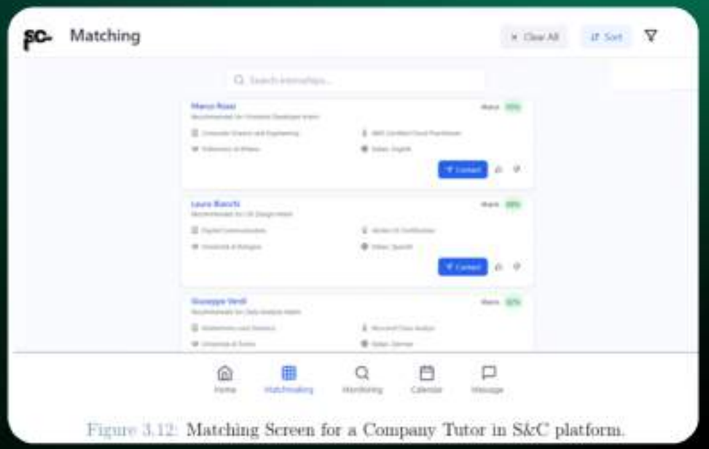


Figure 3.12: Matching Screen for a Company Tutor in S&C platform.

POLITECNICO
MILANO 1863

S&C

Domain Assumptions

04 **Company tutors are assumed to provide genuine internship opportunities**.

05 **University tutors are assumed to offer academic support** and oversight for students during their internships.

07 **Each user belongs to only one category at a time**; for example, *a student cannot simultaneously be a professor or a company tutor.*

POLITECNICO
MILANO 1863

Alloy

Simulates specifications, checks their validity, determines their satisfiability and identifies possible consequences.

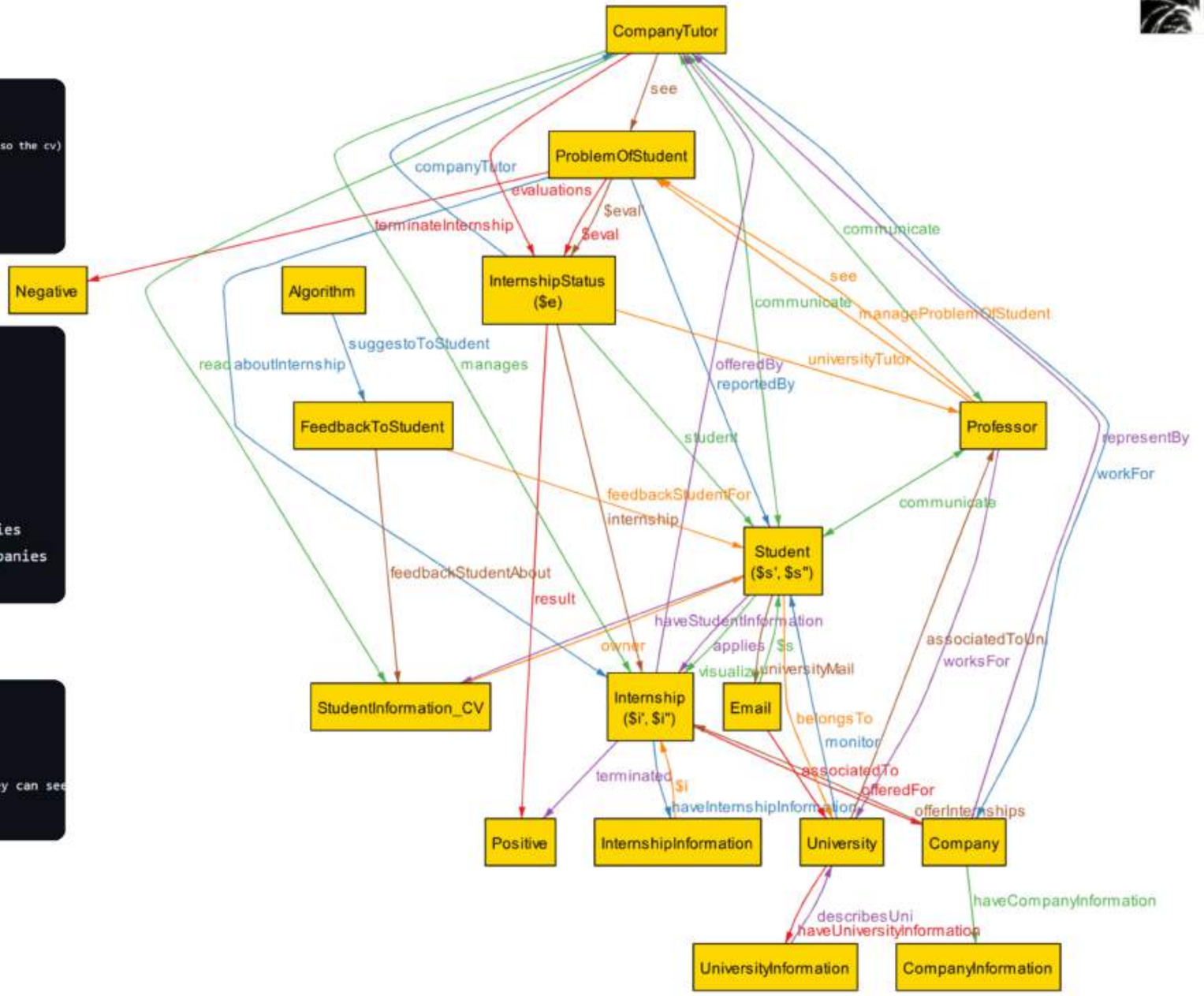
Two students can not finalize the same internship => if a student finalizes all the applications for that internship must be rejected (there can be only one positive evaluation).

If a student is enrolled in a university and he/she applies for an internship he/she must pick a professor of the university, but at first we did not consider as mandatory that each university had at least one professor.

```
// STUDENT
sig Student extends User {  // a student is an user
    belongsTo: one University,  // must be enrolled in exactly one university
    haveStudentInformation: one StudentInformation_CV,  // must have exactly one studentInformation (that contains also the cv)
    universityMail: one Email,  // must have exactly one university mail
    visualize: set Internship,  // can visualize the internships
    applies: set Internship,  // can apply to internships
    seeCompanyProblem: set ProblemOfCompany // can see the companyProblem
}
```

```
// COMPANYTUTOR
sig CompanyTutor extends User {  // a companyTutor is a user
    workFor: one Company,  // must work for exactly one company
    manages: some Internship,   // manage some internships
    read: set StudentInformation_CV,  // can read students' StudentInformation_CV
    evaluations: set InternshipStatus,  // can evaluate students
    see: set ProblemOfStudent,  // can see reported problems of students
    seeCompanyProblem: set ProblemOfCompany, // can see reported problems of companies
    manageProblemOfCompany: set ProblemOfCompany // can also manage problems of companies
}
```

```
// PROFESSOR
sig Professor extends User {
    worksFor: one University,  // must work for exactly one university
    see: set ProblemOfStudent, //  each professor has a set of problemsOfStudents they can see
    seeCompanyProblem: set ProblemOfCompany,  // each professor has a set of problemsOfSComapny they can see
    manageProblemOfStudent: set ProblemOfStudent // professors manage problems
}
```



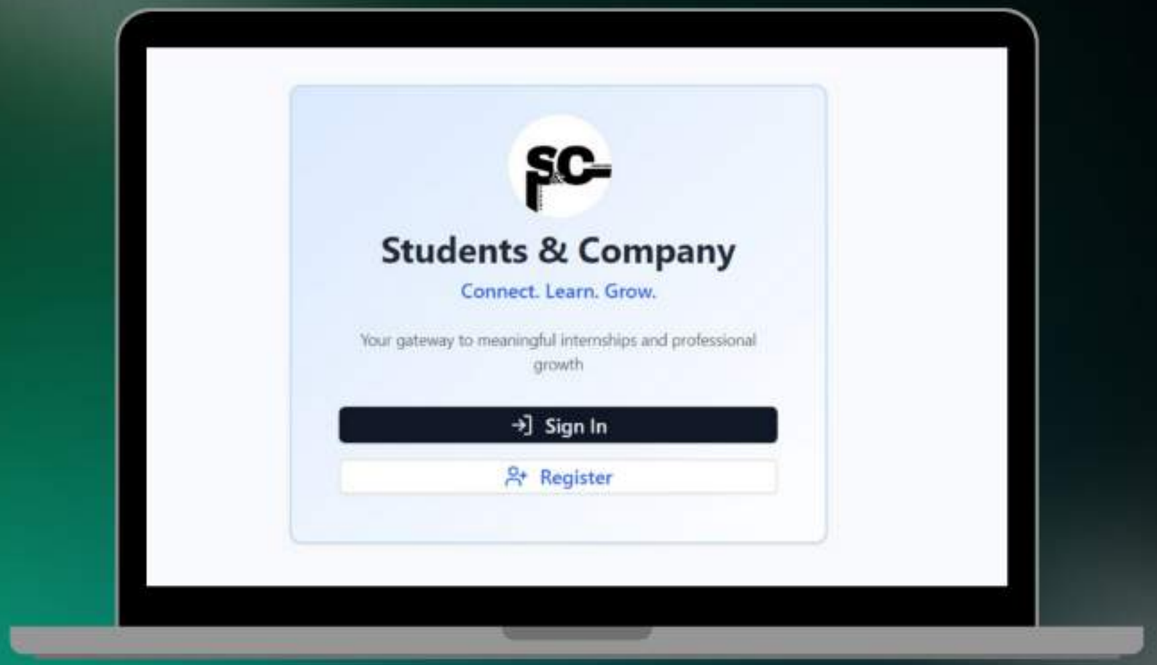
S&C

DD

Design Document

POLITECNICO
MILANO 1863

S&C



Components

Create several components to manage the platform's core functionalities and logic. Each component focus on a specific task. (*Login Manager, Profile Manager, Internship Manager,...*).

Dashboard Manager

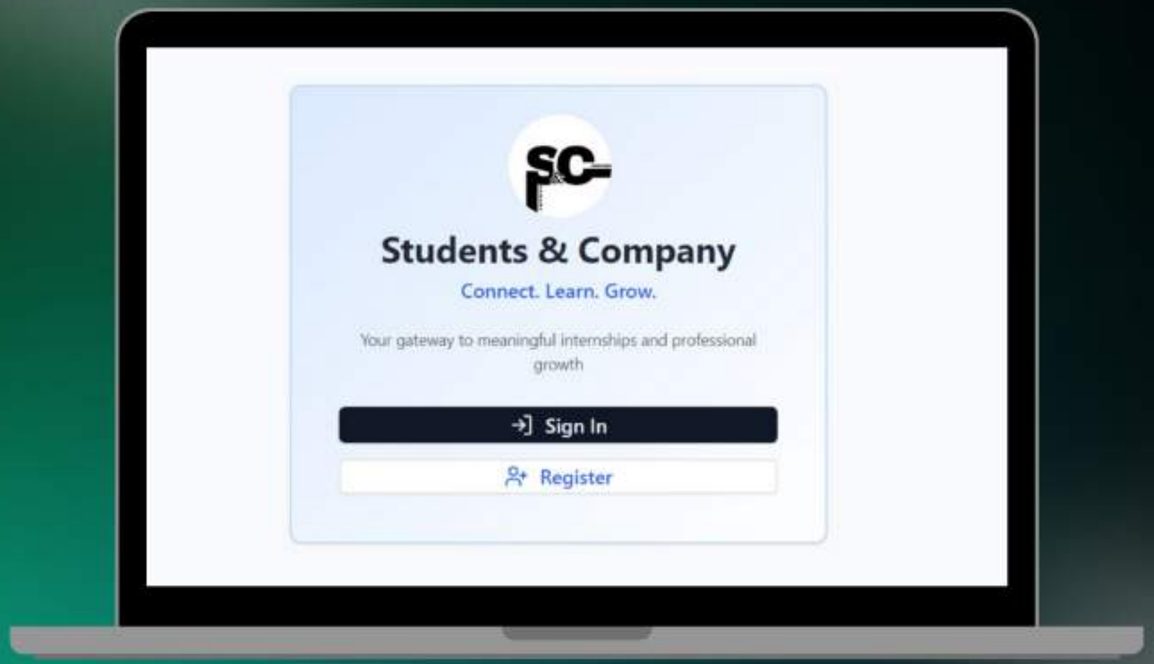
Orchestrates all communications between users and the S&C platform, direct all the requests of the users to the appropriate components.

Model

Represents the data on the server and acts as an interface to the database server. Every component needs to interface with Model to access data from the DBMS.

POLITECNICO
MILANO 1863

S&C



Components

Create several components to manage the platform's core functionalities and logic. Each component focus on a specific task. (*Login Manager, Profile Manager, Internship Manager,...*)

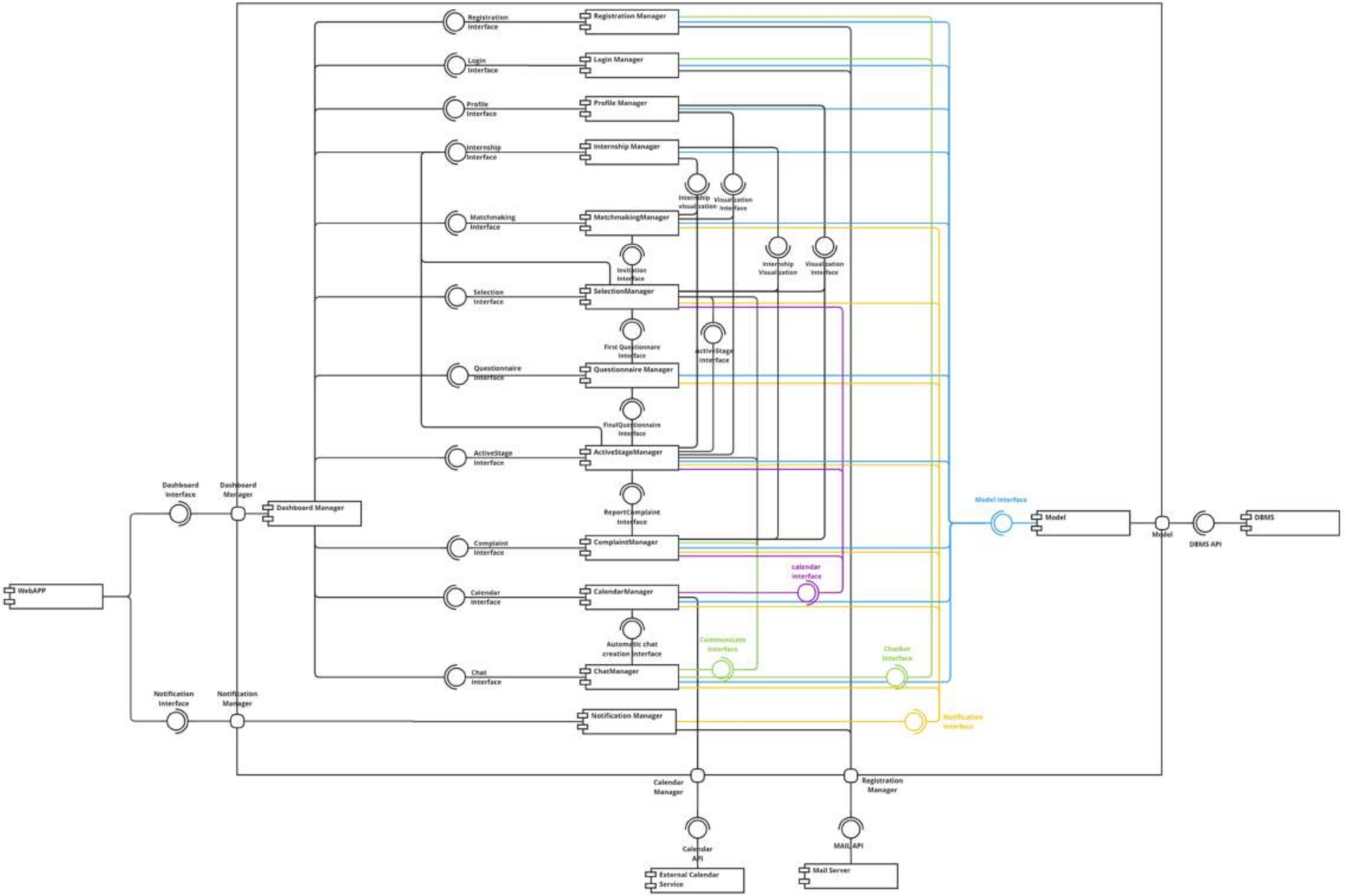
Dashboard Manager

Orchestrates all communications between users and the S&C platform, direct all the requests of the users to the appropriate components.

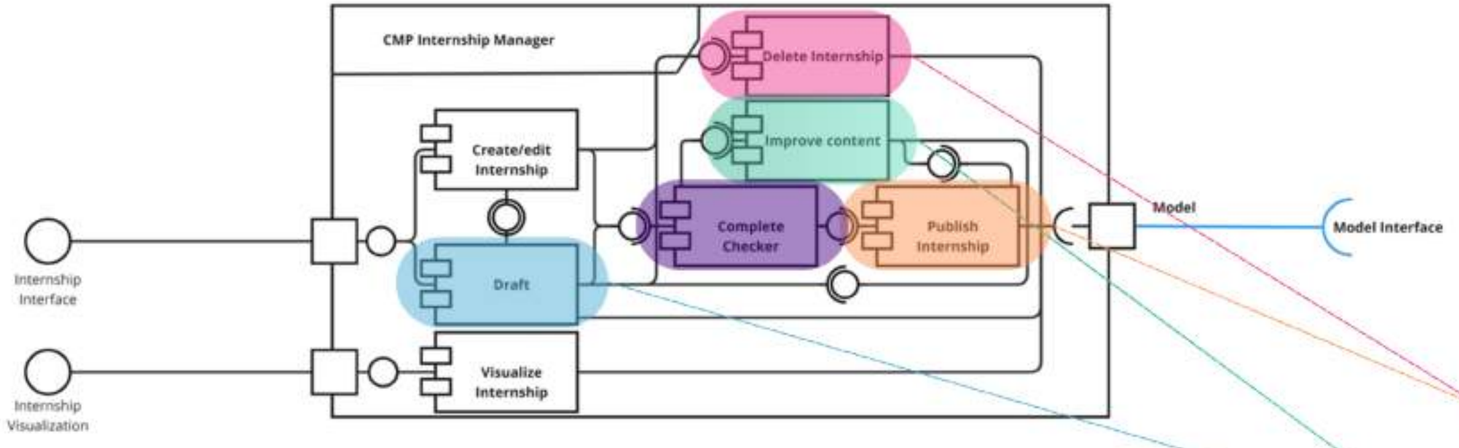
Model

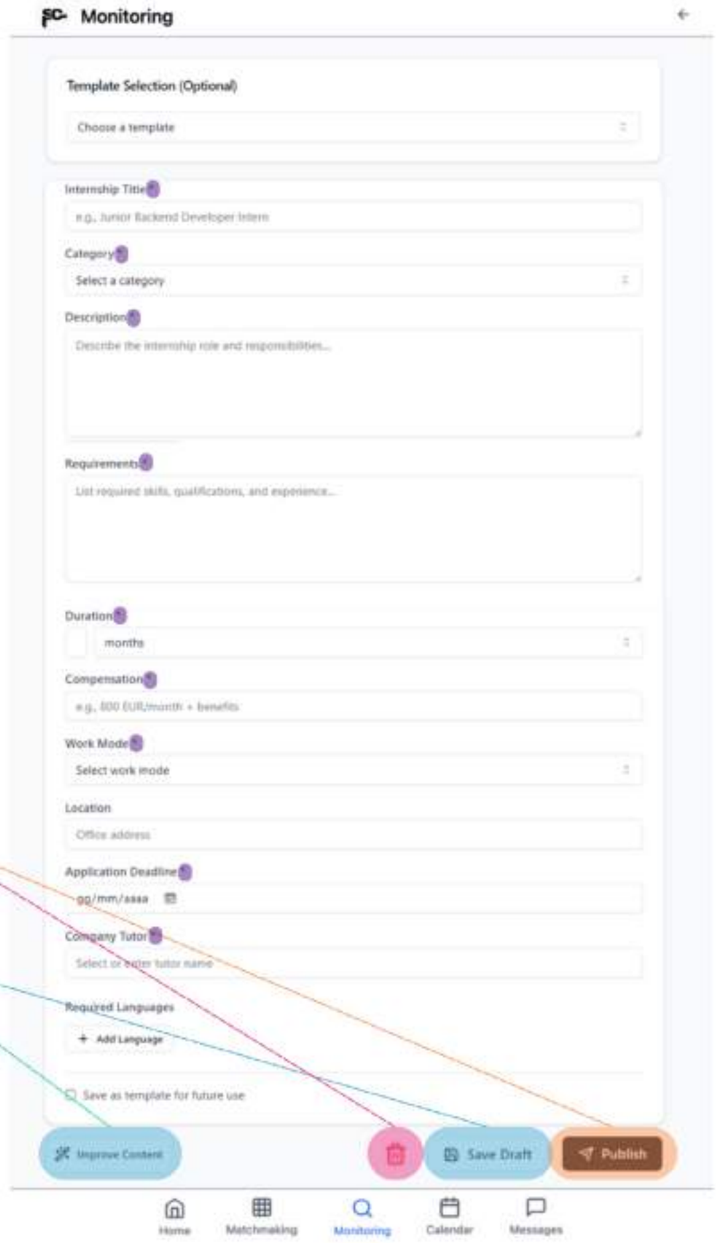
Represents the data on the server and acts as an interface to the database server. Every component needs to interface with Model to access data from the DBMS.

POLITECNICO
MILANO 1863


Registration Interface
Registration Manager
Login Interface
Login Manager
Profile Interface
Profile Manager
Internship Interface
Internship Manager
Internship visualization
Visualization Interface
Matchmaking Interface
MatchmakingManager
Invitation Interface
Internship Visualization
Visualization Interface
Selection Interface
SelectionManager
First Questionnare Interface
activeStage Interface
Questionnaire Interface
Questionnaire Manager
FinalQuestionnaire Interface
ActiveStage Interface
ActiveStageManager
Dashboard Interface
Dashboard Manager
Dashboard Manager
ReportComplaint Interface
Model Interface
Model
Model
DBMS API
DBMS
Complaint Interface
ComplaintManager
calendar interface
WebAPP
Calendar Interface
CalendarManager
Automatic chat creation Interface
Communicate Interface
Chatbot Interface
Chat Interface
ChatManager
Notification Interface
Notification Manager
Notification Manager
Notification Interface
Calendar Manager
Registration Manager
Calendar API
MAIL API
External Calendar Service
Mail Server

Internship Manager





S&C

Architectural Styles

- Three-Tier Architecture
- Client-Server
- REST API
- Model-View-Controller

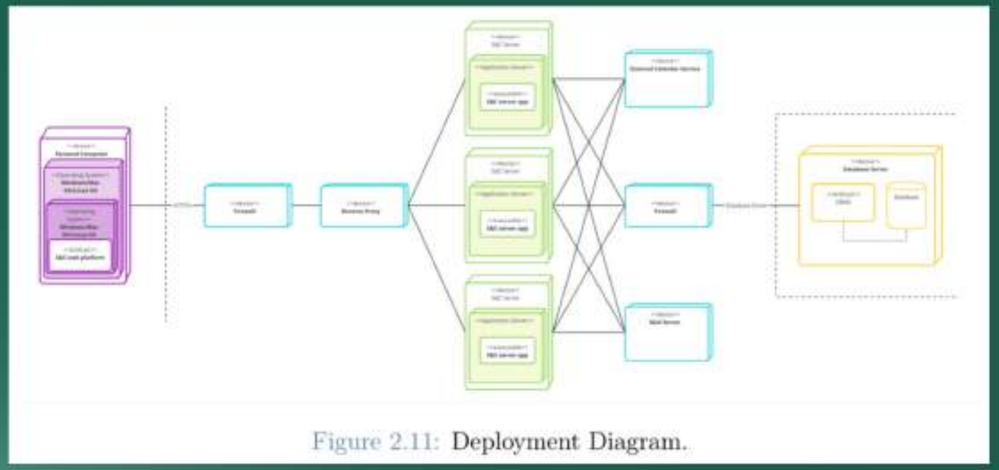


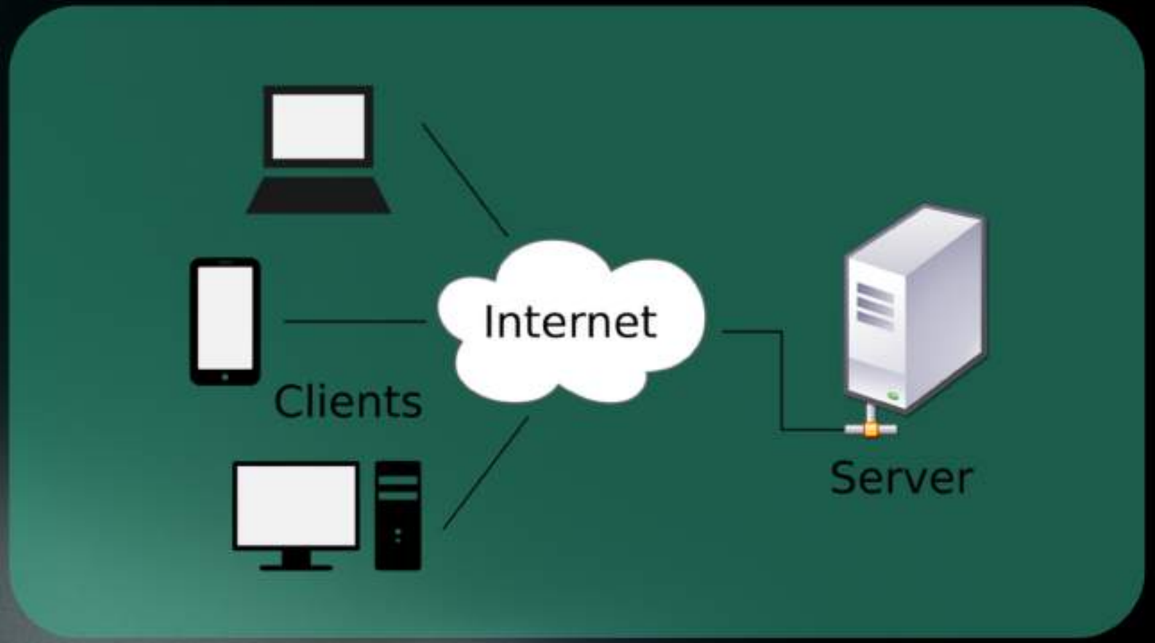
Figure 2.11: Deployment Diagram.

POLITECNICO
MILANO 1863

S&C

Architectural Styles

- Three-Tier Architecture
- Client-Server
- REST API
- Model-View-Controller



POLITECNICO
MILANO 1863

S&C

Architectural Styles

Three-Tier Architecture

Client-Server

REST API

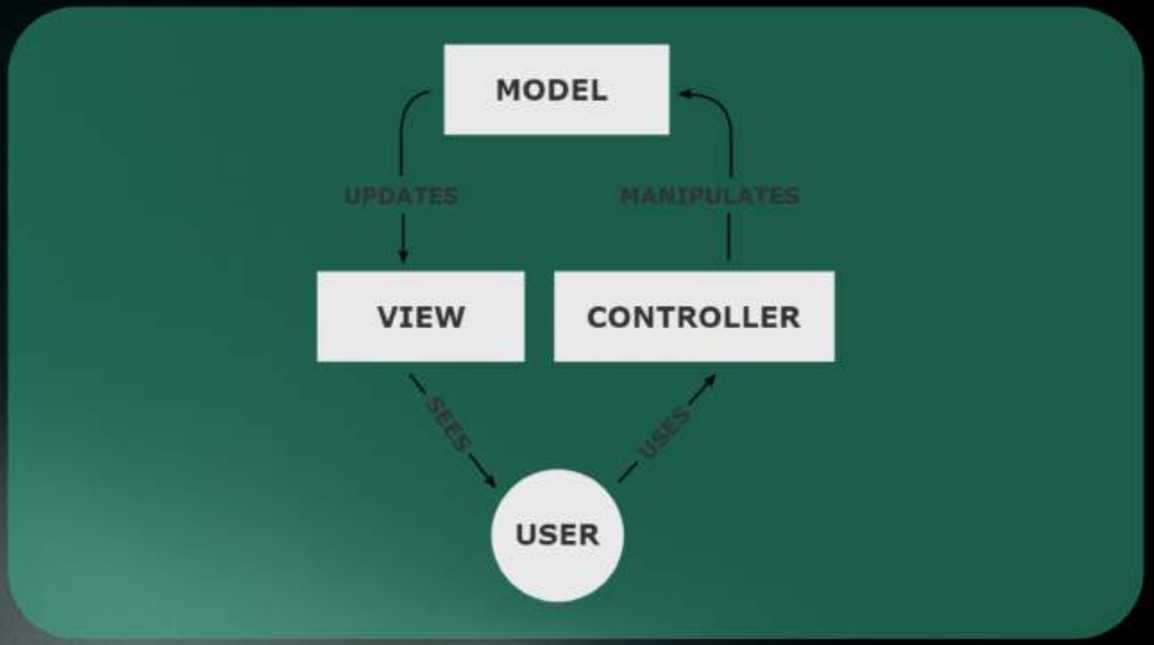
Model-View-Controller

- Facilitates communication between Client and Server.
- Based on the Representational State Transfer principles.
- Each transaction is independent and does not rely on prior interactions.
- No session information is stored; every request contains all necessary data.

S&C

Architectural Styles

- Three-Tier Architecture
- Client-Server
- REST API
- Model-View-Controller



POLITECNICO
MILANO 1863

S&C

Implementation Strategy

Thread-Based

Realization of **features**, implementing each of its parts (end-to-end).

Bottom-Up

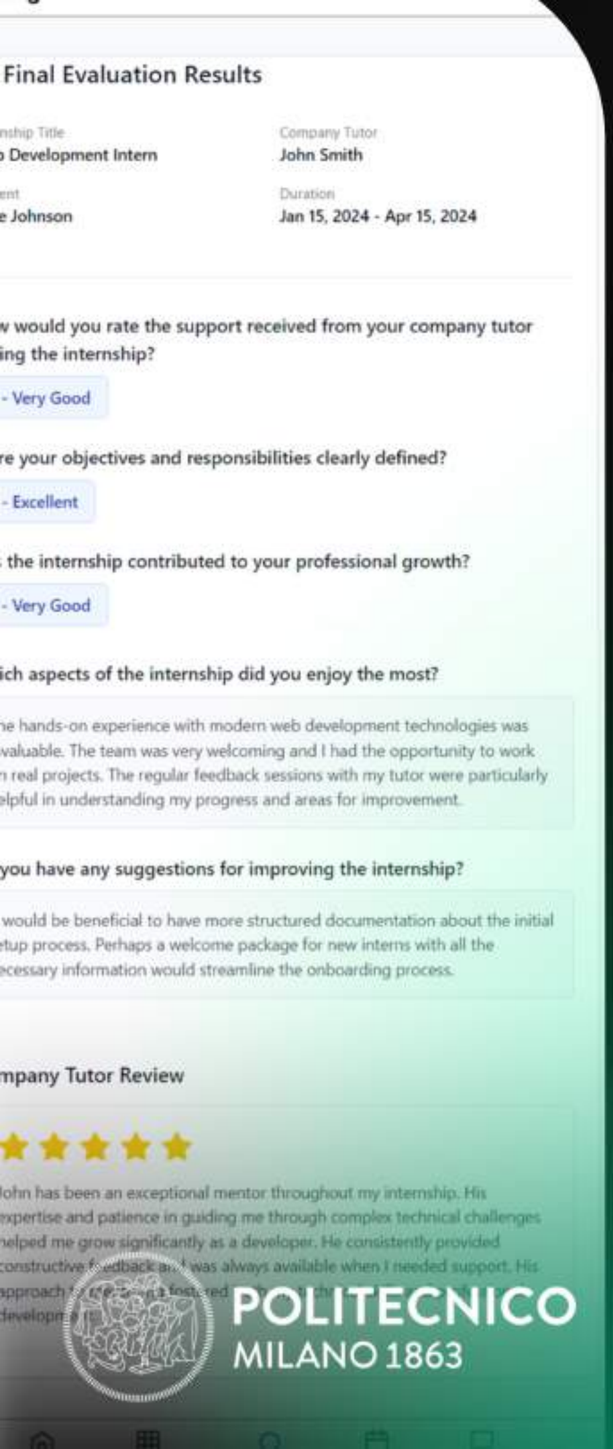
Incremental implementation (layer by layer).

Advantages

- Parallelization.
- Less Risks.
- Transparency.

TARGET ACHIEVED ✓

POLITECNICO
MILANO 1863



Features Identification

9 Features:

1. Authentication
2. Compose
3. Visualize
4. Matchmaking
5. Calendar
6. Messaging
7. Questionnaires
8. Finalization
9. Notification

Motivation

Feature = block of Req/Use cases.

Order = Logical dependencies for some cases.

Parallelization

- Develop subcomponents.

S&C

Component Integration & Testing

1

DBMS & HOST

Since these are the **core infrastructures** (lowest level)

2

DRIVER/STUBS & MODEL

Model=foundation of the system

Driver=simulate other modules to test model in isolation

3

ADDING COMPONENTS

Step-by-step integration:

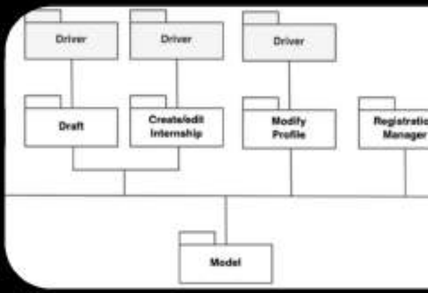
- following features order

1) Test in isolation (drivers/stubs)
2) Integration
3) Test with the rest of the system

4

SYSTEM TESTING

1) Evaluate the entire system:
   - Functional (Use cases?)
   - Performance (heavy tasks?)
   - Load (many users?)
   - Usability/UI (user interaction?)
   - Stress (extreme condition?)
2) **Alpha/Beta** Testing



POLITECNICO MILANO 1863

S&C

Component Integration & Testing

2

DRIVER/STUBS & MODEL

Model=foundation of the system
Driver=simulate other modules to test model in isolation

4

SYSTEM TESTING

1) Evaluate the entire system:
   - Functional (Use cases?)
   - Performance (heavy tasks?)
   - Load (many users?)
   - Usability/UI (user interaction?)
   - Stress (extreme condition?)
2) **Alpha/Beta** Testing

1

DBMS & HOST

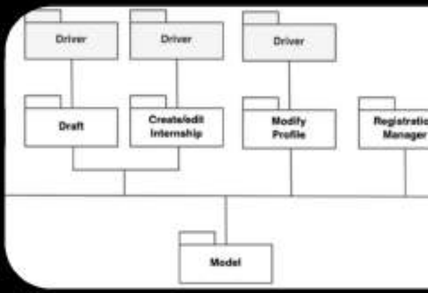
Since these are the **core infrastructures** (lowest level)

3

ADDING COMPONENTS

Step-by-step integration:
- following features order

1) Test in isolation (drivers/stubs
2) Integration
3) Test with the rest of the system



POLITECNICO
MILANO 1863

S&C

Component Integration & Testing

2

DRIVER/STUBS & MODEL

Model=foundation of the system
Driver=simulate other modules to test model in isolation

4

SYSTEM TESTING

1) Evaluate the entire system:
   - Functional (Use cases?)
   - Performance (heavy tasks?)
   - Load (many users?)
   - Usability/UI (user interaction?)
   - Stress (extreme condition?)
2) **Alpha/Beta** Testing

1

DBMS & HOST

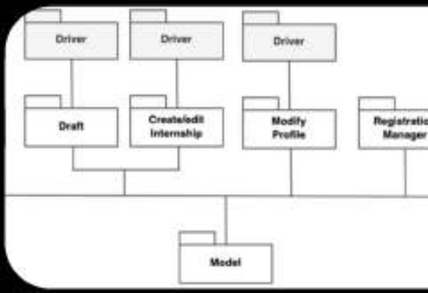
Since these are the **core infrastructures** (lowest level)

3

ADDING COMPONENTS

Step-by-step integration:
- following features order

1) Test in isolation (drivers/stubs)
2) Integration
3) Test with the rest of the system



POLITECNICO
MILANO 1863

S&C

Component Integration & Testing

1

DBMS & HOST

Since these are the **core infrastructures** (lowest level)

2

DRIVER/STUBS & MODEL

Model=foundation of the system
Driver/stubs=simulate other modules to test model in isolation

3

ADDING COMPONENTS

Step-by-step integration:

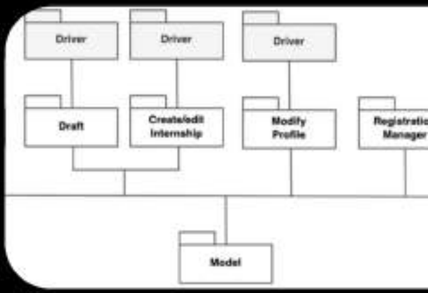
- following features order

1) Test in isolation (drivers/stubs)
2) Integration
3) Test with the rest of the system

4

SYSTEM TESTING

1) Evaluate the entire system:
   - Functional (Use cases?)
   - Performance (heavy tasks?)
   - Load (many users?)
   - Usability/UI (user interaction?)
   - Stress (extreme condition?)
2) **Alpha/Beta** Testing



POLITECNICO
MILANO 1863

POLITECNICO
MILANO 1863

Thank You

Questions?

studentipolimi1@gmail.com Milano, Piazza Leonardo da Vinci