



POLITECNICO
MILANO 1863

SCUOLA DI INGEGNERIA INDUSTRIALE
E DELL'INFORMAZIONE

SOFTWARE ENGINEERING II
COMPUTER SCIENCE AND ENGINEERING

Design Document

Students & Companies

Author:

Name Surname

Student ID:

XXXXXX

Academic Year:

2024-25

Contents

Contents

1	Introduction	1
1.1	Purpose	1
1.2	Scope	2
1.3	Definitions, Acronyms, Abbreviations	4
1.3.1	Definitions	4
1.3.2	Acronyms	4
1.3.3	Abbreviations	4
1.4	Reference Documents	5
1.5	Document Structure	6
2	Architectural Design	7
2.1	Overview	7
2.1.1	High Level View	7
2.2	Component View	7
2.2.1	RESTful APIs Component Diagram	7
2.2.2	Service Discovery Component Diagram	7
2.2.3	Event-Driven Pattern Components	7
2.2.4	Data Layer Access Component Diagram	7
2.2.5	User Interfaces Component Diagram	7
2.3	Deployment View	7
2.3.1	High-Level Deployment View	7
2.3.2	Detailed Deployment View	7
2.4	Run Time View	7
2.5	Component Interfaces	7
2.6	Selected Architectural Styles and Patterns	7
2.6.1	Database Management	7
3	Specific Requirements	9
3.1	External Interface Requirements	9
3.1.1	User Interface	9
3.1.2	Hardware Interfaces	9
3.1.3	Software Interfaces	9
3.2	Functional Requirements	9

3.2.1	Use Case Diagrams	9
3.2.2	Use Cases	9
3.2.3	Sequence Diagrams	10
4	Requirements Traceability	11
5	Implementation, Integration and Test plan	13
5.1	Overview	13
5.2	Implementation Plan	13
5.2.1	Features Identification	13
5.2.2	Components Integration and Testing	13
5.3	System Testing	13
6	Effort Spent	15
7	References	17
8	Per fare prove	19
	Bibliography	21
	List of Figures	25
	List of Tables	27

1 | Introduction

1.1. Purpose

The *Students&Companies (S&C)* platform bridges the gap between university students seeking internships and companies offering them. It simplifies the process of matching students with internship opportunities based on their skills, experiences, and preferences, as well as companies' requirements and offered benefits.

The software involves three main actors: **students**, **companies**, and **universities**.

- **Students** use the platform to search and apply for internships, submit their CVs, and receive recommendations tailored to their profiles.
- **Companies** advertise internships, specify requirements, and manage the selection process for suitable candidates.
- **Universities** monitor the execution of internships and handle complaints or issues that may arise.

S&C features a **recommendation system** that matches students and internships using mechanisms ranging from keyword-based searches to advanced statistical analyses. The platform also facilitates communication, supports the selection process, and tracks internship progress to ensure transparency for all involved parties.

1.2. Scope

The Students&Companies (S&C) platform is a web application designed to facilitate communication and matchmaking between university students seeking internships and companies offering them. The platform simplifies and automates the process by enabling students to explore and apply for internships, while also allowing companies to advertise their openings and identify suitable candidates. Additionally, a sophisticated recommendation system enhances the user experience by automatically suggesting relevant matches to both students and companies based on their preferences and requirements.

This document aims to outline the key architectural decisions behind the design and implementation of the S&C platform. Given the diverse user base, which includes students, companies, and universities, and the need for simultaneous interaction among these parties, a web application was chosen as the foundation. Its accessibility and ease of use ensure a seamless experience for users across various locations and devices.

The complexity of the platform, along with the distinct functionalities it provides—such as recommendations, selection processes, and feedback collection—led to the choice of a microservices architecture. This architectural style was selected due to its ability to offer scalability, flexibility, resilience, and modularity. Each microservice operates independently, allowing for targeted scaling based on demand, individual updates and deployments, and clear separation of responsibilities. The result is a system that is both maintainable and adaptable to evolving requirements.

From a deployment perspective, the system adopts a three-tier architecture. The user client layer represents the web and mobile interfaces used by students, companies, and universities. The server layer hosts all microservices, which manage the business logic and application functionality. Finally, the shared database layer ensures consistency in data storage, maintaining information about users, internships, recommendations, and feedback.

To manage interactions between microservices, the platform uses a combination of communication patterns based on specific functional needs. For real-time or asynchronous interactions, an event-driven communication model is employed. In this approach, some microservices act as event publishers while others function as consumers. For example, the Notification Microservice processes events related to complaints, messages, and new recommendations, while other services publish events to reflect changes in system states, such as the publication of a new message on the communication platform or the acceptance of a student for an interview.

For functionalities that do not require immediate interactions, synchronous communication mechanisms are used. This includes scenarios such as retrieving a list of internships or submitting CVs, where sequential processing and immediate feedback are necessary. The combination of event-driven and synchronous communication ensures that the system remains both responsive and straightforward to use, catering to the dynamic needs of its users.

All these architectural choices are just mentioned here to provide an overview of the system; they will be better explained and unpacked down the line of this document. The

following image shows the major components of the Students&Companies system.



1.3. Definitions, Acronyms, Abbreviations

1.3.1. Definitions

1.3.2. Acronyms

1.3.3. Abbreviations

1.4. Reference Documents

1.5. Document Structure

2 | Architectural Design

2.1. Overview

2.1.1. High Level View

2.2. Component View

2.2.1. RESTful APIs Component Diagram

2.2.2. Service Discovery Component Diagram

2.2.3. Event-Driven Pattern Components

2.2.4. Data Layer Access Component Diagram

2.2.5. User Interfaces Component Diagram

2.3. Deployment View

2.3.1. High-Level Deployment View

2.3.2. Detailed Deployment View

2.4. Run Time View

2.5. Component Interfaces

2.6. Selected Architectural Styles and Patterns

2.6.1. Database Management

3 | Specific Requirements

3.1. External Interface Requirements

3.1.1. User Interface

3.1.2. Hardware Interfaces

3.1.3. Software Interfaces

3.2. Functional Requirements

3.2.1. Use Case Diagrams

3.2.2. Use Cases

3.2.3. Sequence Diagrams

4 | Requirements Traceability

5 | Implementation, Integration and Test plan

5.1. Overview

5.2. Implementation Plan

5.2.1. Features Identification

5.2.2. Components Integration and Testing

5.3. System Testing

6 | Effort Spent

7 | References

8 | Per fare prove

Ciao ragazzi come va?

Guardate questo link importantissimo: [1]

Questo lo ho aggiunto dopo.

Questo aggiunto dopo da VS code direttamente.

modifica in chimata

Bibliography

- [1] Simone. provabibliografia, 2024.

List of Figures

List of Tables

