POLITECNICO
MILANO 1863

SCUOLA DI INGEGNERIA INDUSTRIALE
E DELL'INFORMAZIONE

S&C - ATD

Software Engineering 2 Project
- MSc Computer Science and engineering

Authors: **Vaccarino - Vacis - Palladino**

Student IDs: 10779331 - 10764243 - 10765405
Reference Professor: Prof. Matteo Camilli
GitHub-Repo:https://github.com/giovanni-vaccarino/VaccarinoPalladinoVacis
Academic Year: 2024-25

1 | Analyzed Project

- **Authors**: Marta Ravazzoni, Diego Terzi, Matteo Barbieri
- **Github Repo**: https://github.com/DiTiEu/BarbieriRavazzoniTerzi
- **Reference material**: All documents and code found in the repository, including:
  - RASD document
  - DD document
  - ITD document

2 | Installation and Setup

The installation and the setup process was specified in the ITD document. We have not experienced any issues in this phase and following the steps specified, we managed to successfully install and start the application. The requirements and the dependecies specified were enough to don't cause issues.

3 | Acceptance Test Cases

Below you can find firstly a list of general comments regarding the tests we have performed. At section 3.8 you can find the detailed list of tests performed, associated with the functional requirements that we've found in the RASD and ITD documents.

3.1. Authentication and User Profile Management

Various authentication-related requirements were tested, revealing several issues:

- There is no dedicated user table in the database. Instead, email and password fields are separately stored in the student, company, and university tables. This results in an authentication conflict, allowing the same email to be registered for multiple roles.
- As a consequence, if a user registers as both a student and a company, they cannot choose their login role, leading to authentication ambiguity.

3.2. Recommendation System

To test the recommendation system, it was necessary to manually insert skills into the database. Without this manual intervention, the matching system could not be tested. Despite the manual insertion we had many trouble testing it. Furthermore it was necessary to stay in touch with the developers of the application. As regards this, they help us to clarify some doubts but many others remained unsolved.

3.3. Internship Management

Similar to the recommendation system, skills had to be manually inserted into the database. Additionally:

- Skills were not visible unless they were inserted into the "internship_skills" table.
- The system only functioned correctly if an entry with ID 0 and the name "Select a skill" was present in the skills table.
- No confirmation or error messages were displayed, making debugging difficult.
- A page listing all student applications is missing.

- The process for companies to accept applications was unclear.
- Different application states are missing, preventing users from understanding application progress.

3.4. Selection and Interview Management

The following issues were observed:

- A student can apply multiple times to the same application due to a lack of backend validation.
- Controls to ensure that a student has uploaded a CV before applying are missing.
- Once a CV is sent to the company, there is no way to decline the application, leaving it permanently in a pending state.
- Missing redirects send companies back to the same confirmation page.
- Due to a lack of backend control, companies can submit responses multiple times.
- Users are unable to upload a new CV or modify an existing one.

3.5. Feedback Management

- When no feedback is available, the feedback page incorrectly redirects to the homepage.
- The process of adding feedback was obstructed since the process of accepting applications was not possible.

3.6. Automated Suggestions System

The implementation of this feature was not required.

3.7. Monitoring and Issue Resolution

The implementation of this feature was not required.

3.8. Functional Requirements

Issues: It was required to the student to add manually data inside the database, which increase the workload required to test the web application and needed to contact directly the developer of the app.

[R1]: The System allows students to create an account by providing personal information and uploading their CVs

User Registration:

- Creation of a Student account.

Description: This test verifies that students can register by providing valid personal information and compile their CVs. Upon successful registration, students should be presented with their profile and options to manage their CV.

Test type	Test input	Pass/Fail
Register Student	Name: Mario Surname: Rossi Email: mariorossi@gmail.com Password: Pass Pass	Passed

Table 3.1: Test Case for Student Registration

Test type	Test input	Pass/Fail
Submit CV	Bio: Ciao sono una bio Graduation Grade: 110 Graduation Title: Computer Engineer Field of Study: Computer Study	Passed, but we had to manually add the skills to the database, including a strange skill with ID 0 and name "Select a skill".

Table 3.2: Test Case for CV Submission

[R2]: The System allows companies to create an account by providing company details and uploading internship descriptions

User Registration:

- Creation of a Company account.

Description: This test ensures that companies can register by providing valid details and uploading internship descriptions. The system should validate input and prevent duplicate company accounts. Upon registration, companies should access features for internship management.

Test type	Test input	Pass/Fail
Register Company	Name: Company Email: company1@gmail.com Password: Pass	**Passed**

Table 3.3: Test Case for Company Registration

Test type	Test input	Pass/Fail
Internship Creation	Title: Software Eng Intern Description: Internship description Begin Date: 20/03/2025 End Date: 20/07/2025 Question: What is your favourite programming language? Field of Study: Computer Science Total Positions: 10 Skills: C++, C	**Passed**

Table 3.4: Test Case for Internship Creation

[R3]: The System allows universities to create accounts to manage internship issues

User Registration:

- Creation of a University account.

Description: This test ensures that universities can register.

Test type	Test input	Pass/Fail
Register University	Name: University Of Madagascar Email: unimadagascar@gmail.com Password: Pass	**Passed**

Table 3.5: Test Case for University Registration

[R4]: The System allows all users to log in to their respective accounts using valid credentials

User Authentication:

- Login for Students, Companies, and Universities.

Description: This test ensures that users can log in using valid credentials. The system should handle authentication securely and deny access to invalid credentials.

Outcome: The system **does not** guarantee this requirement. Infact the same email can be registered as a student, a company and a university, but when logging in, you can't decide which type of account you should login with.

Test type	Test input	Pass/Fail
Login	Email: mariorossi@gmail.com Password: Pass	**Passed**

Table 3.6: Test Case for Login

Test type	Test input	Pass/Fail
Login	Email: company1@gmail.com Password: Pass	**Fail**, since the system doesn't allow to choose with what type of account to log in, and we have lost the company account associated to company1@gmail.com.

Table 3.7: Test Case for Login

[R5]: The System allows students to upload, modify, and delete their CVs

CV Management:

- Upload, modify, and delete CVs.

Description: This test verifies that students can upload, modify, and delete their CVs. The system should confirm each action and update the student's profile accordingly.

Outcome: The system successfully allows students to create it for the first time, but it **does not** allow for modification or deletion.

Test type	Test input	Pass/Fail
Modify CV	N.A	**Fail**, since it is not possible to update your existing CV, it is only possible to recreate a new one.

Table 3.8: Test Case for Modify CV

Test type	Test input	Pass/Fail
Delete CV	N.A	**Fail**, since it is not possible to remove your CV.

Table 3.9: Test Case for Delete CV

3.9. Recommendation System

[R6]: The System extracts keywords from students' CVs and self-assessments

Keyword Extraction:

- Extraction of keywords from CVs and self-assessments.

Description: This test has not been performed, since we have not managed a way to workout properly the action.

Outcome: N.A.

[R8]: The System suggests internships to students based on a match between their skills, preferences, and the requirements of internships

Internship Suggestions:

- Suggesting internships to students based on matching criteria.

Description: This test ensures that the system suggests internships based on keyword matches, student skills, and preferences.

Outcome: The system **does not** provide a dedicated section where lists the suggested internships. However in the internship listing, there is a sorting method for matching, but the result is not clear.

[R9]: The System allows companies to view recommended students for their internships

Student Recommendations:

- Viewing recommended students.

Description: This test verifies that companies can view a list of students recommended for their internships.

Outcome: The system displays a list of recommended students for internships.

Test type	Test input	Pass/Fail
View Recommended Students	N.A	**Fail**, since if we go to the company dashboard and click on the "Browse CVs" button to explore the recommended students, we are redirected to the company's dashboard page, without any error message or feedback.

Table 3.10: Test Case for Viewing Recommended Students for Internships

---

[R10]: The System allows students to browse internships suggested by the recommendation system

Internship Browsing:

- Browsing suggested internships.

Description: This test ensures that students can browse internships suggested by the recommendation system.

Outcome: The system provides in the global listing internship page a sorting filter for the matched internships.

3.10. Internship Management

[R11]: The System allows companies to upload, modify, and delete internship descriptions

Internship Management:

- Upload, modify, and delete internship descriptions.

Description: This test verifies that companies can manage internship descriptions, including uploading, modifying, and deleting them.

Outcome: The system allows companies to manage internship descriptions.

Test type	Test input	Pass/Fail
Create Project (New Internship)	Title: Dev Available Positions: 100 Question1: question1 Skill1: skill1	**Passed**, works correctly, only open question.
Update Project (Update Internship)	Title: New Dev Question1: New Question1	**Fail**, the update does not work. When updating the project, the skills are no longer visible. No alert message to verify what is wrong.

Table 3.11: Test Case for Creating and Updating an Internship

---

[R12]: The System allows students to browse all internships from a global list

Global Internship Browsing:

- Browsing all internships.

Description: This test ensures that students can browse all internships from a global list.

Outcome: The system allows students to browse a comprehensive list of internships.

[R13]: The System allows students to apply for internships by completing the associated questionnaire

Internship Application:

- Applying for internships by completing questionnaires.

Description: This test verifies that students can apply for internships by completing the required questionnaires.

Outcome: The system successfully processes internship applications and stores questionnaire responses. But there are no message to verifity the status of the application.

Test Type:
The system allows students to apply for internships by completing the associated questionnaire

Test Input:
Question: What is your favourite programming language?
Answer: Javascript for sure

Pass/Fail:
Passed

3.11. Selection and Interview Management

[R14]: The System does not allows companies to review questionnaire responses submitted by students

Response Review:

- Reviewing student responses to questionnaires.

Description: This test ensures that companies can view and review the responses submitted by students during the application process.

Outcome: The system **does not** provide a way to investigate the answers provided by the students to the questionnaries.

[R15]: The System allows companies to approve or reject applications based on submitted responses

Application Decision:

- Approving or rejecting internship applications.

Description: This test verifies that companies can approve or reject student applications based on their responses to the questionnaire.

Outcome: The system allows companies to approve or reject applications and updates the application status accordingly.

[R16]: The System marks internships as officially started on the start date specified in the internship details

Internship Start:

- Marking internships as started.

Description: This test ensures that the system marks internships as officially started on the specified start date.

Outcome: There was no marking signs of the officaly started internship.

[R17]: The System marks internships as completed after the declared duration has elapsed

Internship Completion:

- Marking internships as completed.

Description: This test verifies that the system marks internships as completed after the specified duration has elapsed.

Outcome: There was no way to verify it is working or not, since there are no page dedicated to track the status of the internship both on the student and company page. In general we can say that there were no marking signs of the completed internship.

3.12. Feedback Management

[R18]: The System allows companies to provide feedback on students' performance at the conclusion of an internship

Company Feedback:

- Providing feedback on student performance.

Description: This test ensures that companies can provide feedback on students at the conclusion of an internship.

Outcome: There was no way to test whether an internship comes to an end and obiviously this requirement too.

[R19]: The System allows students to provide feedback on companies and their internship experiences

Student Feedback:

- Providing feedback on companies.

Description: This test verifies that students can provide feedback on their internship experience and the company.

Outcome: There was no way to test whether an internship comes to an end and obviously this requirement too.

3.13. Automated Suggestions System

We **have not** covered the automated suggestions system, since it was not in the list of the requirements to implement.

4 Documents and Code Quality

4.1. Documents Quality

As far as documents concerns, we've noticed the following:

4.1.1. RASD Document

Issues:

- The goals are not complete, and all of them could be splitted in different goals.
- The user interface section is empty.
- The use case diagram is confusionary and not clear.

Strengths:

- Accurate and complete descriptions for the use cases sequence diagrams.
- Although some functional requirements have the same issue of the goals-including more than one functional requirement in a single one-the requirements seem pretty complete and well divided in sections.

4.1.2. DD Document

Issues:

- The dispatcher's component interface and usage is wrong, since the dispatcher, following the SRP principle, should only route the requests that the server is receiving to the correct section of the application.
- The other components interfaces are not well detailed, not specifying the response of the requests, and the description of the purpose of the single elements.
- The user interface design section is completely empty.

Strengths:

- The use cases sequence diagrams look complete and coherent with the ones written in the RASD document.

4.1.3. ITD Document

Issues:

- All the testing strategies mentioned(including unit, integration and e2e testing) have not been implemented. We have indeed not found any file in the github repository related to testing.

4.2. Code Quality

Regarding the code, we've noticed the following:

- **Lack of Documentation:** The codebase lacks doc comments, making it difficult to understand the purpose and functionality of various components.
- **Static Analysis Results:** We conducted static analysis(using flask8) and found only some warnings in the source code.
- **Violation of the Single Responsibility Principle (SRP):** In the route directory files, we noticed that the Single Responsibility Principle has not been followed. The separation of concerns between controllers and services is unclear. Controllers, which are responsible for handling incoming requests, are mixed with business logic that should ideally reside within separate service components. This coupling makes it difficult to test the business logic independently of the routing endpoints.
- **Error Handling:** The code lacks error-handling: in many cases, even when a request does not result in a successful creation or update (e.g., the creation of an internship), the response still returns a status code of `200 OK` with no accompanying error messages. This impacts also a lot the usability of the application, where in many cases we've experienced the need to check manually in the database if the action was effectively applied.

5 | Effort Spent

Member of Group	Effort Spent	Hours
Giovanni Vaccarino	Introduction	0h
	Installation and Setup	0.5h
	Acceptance Test Cases	1h
	Documents and Code Quality	2.5h
Vittorio Palladino	Introduction	0h
	Installation and Setup	0.5h
	Acceptance Test Cases	4h
	Documents and Code Quality	0h
Nicolò Vacis	Introduction	0h
	Installation and Setup	0h
	Acceptance Test Cases	3h
	Documents and Code Quality	0.5h

Table 5.1: Effort spent by each member of the group

6 | References

6.1. References

- **GitHub Repository** - https://github.com/DiTiEu/BarbieriRavazzoniTerzi
- **Documents** - All documents (RASD, DD, ITD) found in the repository