deti universidade de aveiro departamento de eletrónica, telecomunicações e informática

Software Engineering

1st Semester

2024/2025

Individual Project Proposal Document

Under the supervision of:

Nuno Sá Couto

Rafael Direito

Table of Contents

1 Document History 2
2 Description 3
3 Objectives 3
4 Key Technical Requirements 3
5 Base Architecture 4
6 Solution To Develop 4
7 General Project Guidelines 5
8 Final Deliverables 5
9 Useful Tutorials/Links 5
10 FAQ 5

1 Document History

Revision	Date	Updated By	Update Description
v0.1	13-09-2024	Rafael Direito	Initial Version

2 Description

This individual project proposal outlines the scope and objectives of a software development project to be undertaken by each student. The goal of this project is to enable students to gain practical experience in designing, developing, and deploying a software solution on Amazon Web Services (AWS).

3 Objectives

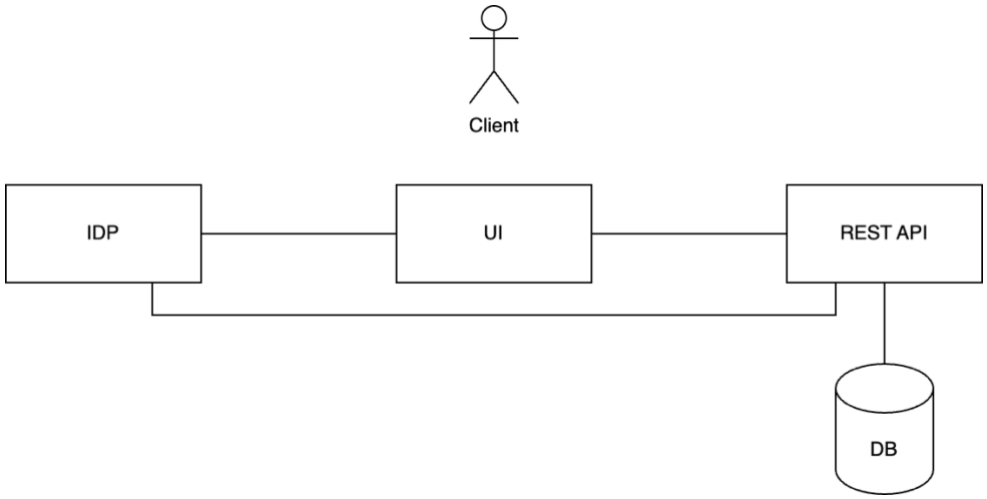
- **Software Development:** Develop a web-based software solution hosted on AWS that includes a RESTful API, a Web UI, and interacts with a relational database.
- **Agile Workflow:** Implement an agile development methodology with epics and user stories to manage project tasks efficiently and promote iterative development.
- **Authentication and Authorization:** Implement authentication, authorization, and accounting (AAA) mechanisms using, for instance, AWS Cognito for user management. Students should use an IDP.
- **Secure and Robust Deployment:** The developed solution must be deployed on AWS in a secure and robust approach.

4 Key Technical Requirements

All software solutions must adhere to the following technical requirements:

1. **REST API:** Develop a RESTful API to provide data and services to the Web UI.
2. **Web UI:** Create a user-friendly web interface that interacts with the REST API for user interaction.
3. **Relational Database:** Utilize a relational database system (e.g., PostgreSQL, MySQL) for data storage and retrieval.
4. **AAA Mechanisms**: Implement authentication, authorization, and accounting mechanisms using an IDP.
5. **Security:** The deployment of the developed security must follow good practices on security (do not expose ports that are not needed, etc.).
6. **Computing:** The developed solution can be deployed either in AWS EC2, or in AWS Elastic Container Service (ECS), as docker containers

5 Base Architecture



6 Solution To Develop

In the scope of the Individual course project, all students must develop the same software solution: a **To-Do List Application**. The **functionalities offered by the various solutions must be the same**. However, **students are free to choose their implementation tools and workflows**.

Functionalities

- Task Ownership
  - Each user must authenticate in the system.
  - The tasks created by a user can only be visible to them
- Task Management
  - Users should be able to add tasks with a title and description.
  - Users can mark tasks as completed.
  - Users can edit or delete existing tasks.
- Task Deadlines
  - Users can set deadlines for tasks.
- Sorting and Filtering
  - Provide options to sort tasks by creation date, deadline, or completion status.
  - Allow filtering tasks by category or completion status.
- Task Prioritization
  - Users should be able to assign priorities (e.g., low, medium, high) to tasks.

7 General Project Guidelines

In this project, you should:

- To deliver working software applying an **Agile/Scrum** software development methodology
- Use as much as possible the free tier AWS services
- Document the API using the **OpenAPI** specification
- Use FOSS tools

You must use:

- **Git**: Source Code Management
- **JIRA**: To manage and communicate project sprints, user stories and other relevant issues

You can use:

- **Docker**: To package most of the system's components

To kickstart your project, you must create a Jira account/workspace. This workspace should be accessible to all team members and course teachers. As such, you should provide access to the course teachers by adding them to your project. You may use the teachers' university e-mails (nuno.sacouto@ua.pt, rafael.neves.direito@ua.pt).
You shall follow the same approach for your GitHub accounts/organization.

8 Final Deliverables

1. Solution's **detailed architecture** diagram (APIs, databases, etc) + **workflows**
2. **User Stories** (implemented during the project)
3. **Application code** (Git repository)
4. **Demo** of the system (video) - to be discussed later
5. **Project Report** (should detail how you implemented your solution and made it available in AWS) More information on this project report will be provided during the semester.

9 Useful Tutorials/Links

TBD

10 FAQ

[Question]
TBD

[Answer]
TBD