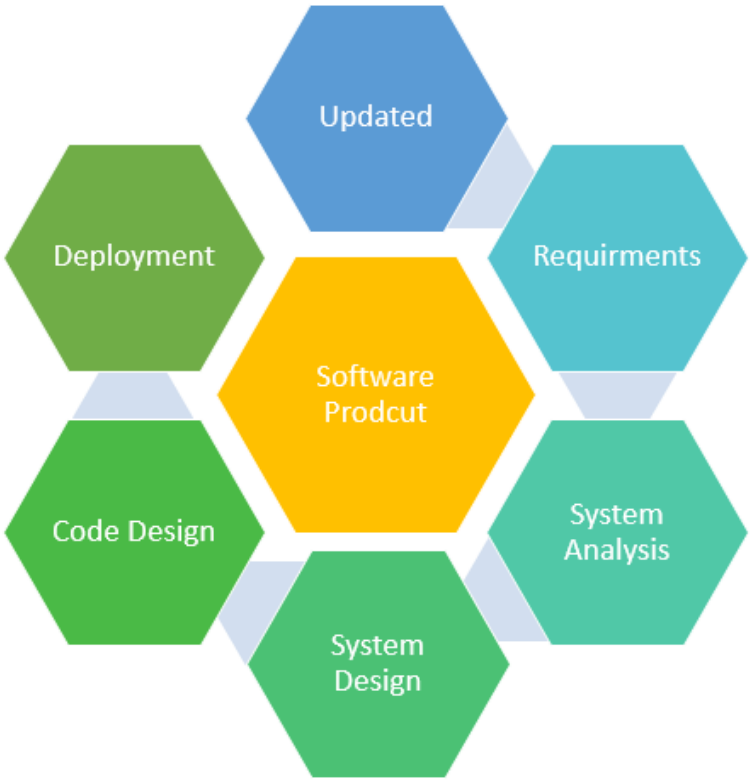
deti universidade de aveiro departamento de eletrónica, telecomunicações e informática

Software Engineering

1st Semester

2024/2025

Group Project Proposal Document



Under the supervision of:

Nuno Sá Couto

Rafael Direito

Table of Contents

1 Document History

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

2 Description

This project proposal outlines the scope and objectives of a collaborative software development project to be undertaken by college students. 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) while fostering creativity in choosing the software theme.

3 Objectives

- **Software Development:** Develop a web-based software solution that includes a RESTful API, a Web UI, and interacts with a relational database. Additionally, this solution should comprise at least 3 microservices and asynchronous communication mechanisms. This software solution is to be deployed on AWS.

- **Agile Workflow:** Implement an agile development methodology with weekly sprints, epics, and user stories to manage project tasks efficiently and promote iterative development.

- **Authentication and Authorization:** Implement authentication, authorization, and accounting (AAA) mechanisms using an IDP (e.g. AWS Cognito) for user management, ensuring at least two user scopes - admin and regular users.

- **Media Content Handling:** Enable the software to store and manage various types of media content, such as images, sound, or video files.

- **Centralized Logging and Monitoring:** The developed solution should be supported by monitoring and logging dashboards. These dashboards shall provide the means to monitor the solution and for a faster log analysis and troubleshooting.

4 Key Technical Requirements

All software solutions must adhere to the following technical requirements:

1. **Microservices:** At least 3 microservices must be implemented.

2. **Internal Communication:** At least 2 of the implemented microservices should communicate asynchronously.

3. **REST API:** Develop a RESTful API to provide data and services to the Web UI.

4. **Web UI:** Create a user-friendly web interface that interacts with the REST API for user interaction.

5. **Database:** Utilize a database system (e.g., PostgreSQL, MySQL, MongoDB, Redis) for data storage and retrieval.

6. **AAA Mechanisms**: Implement authentication, authorization, and accounting mechanisms using AWS Cognito, providing admin and user roles with appropriate access levels.

7. **Media Content Handling:** Incorporate features to upload, store, and manage media content (e.g., images, sound, video) securely within the application.

8. **Centralized Monitoring and Logging:** The developed solution should be supported by monitoring and logging dashboards. These dashboards shall provide the means to monitor the solution and for a faster log analysis and troubleshooting.

9. **Docker:** All solution components that are not offered by AWS must be packaged into Docker Images and deployed through AWS ECS.

5 General Project Guidelines

In this project, you should:

- To deliver working software applying an **Agile/Scrum** software development methodology
- Apply a **microservice architecture** and best practices
- Use as much as possible the free tier AWS services
- Document all APIs using the **OpenAPI** specification
- Having a **CI/CD pipeline** with Automated Unit Testing, and **integrated with the SCM**, is considered as plus
- Having **Terraform scripts** to create your AWS infrastructure is considered a plus

You must use:

- **Docker**: To package most of the system's components
- **Git**: Source Code Management
- **JIRA**: To manage and communicate project sprints, user stores and other relevant issues
- **Docusauros, Confluence, or similar**: To create a documentation page / wiki. In this page you should list all the deliverables (see next section)

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.
In regard to the Docusauros (or similar) website, we advise you to host it through GitHub Pages.

6 Deliverables

1. **Project Proposal:** Each student group will submit a detailed project proposal outlining their chosen software theme, project plan, and timeline.
2. Solution's **detailed architecture** diagram (APIs, databases, etc) + **workflows**
3. **User Stories detail** (for the ones implemented) by sprint – **to be presented in weekly sprint Reviews**
4. **Application code** (the course lecturers should have access to your Git repository)
5. **Demo** of the system (video that should be included in the final presentation) - to be discussed later
6. Final presentation

7 Useful Tutorials/Links

TBD

8 FAQ

[Question]
TBD

[Answer]
TBD