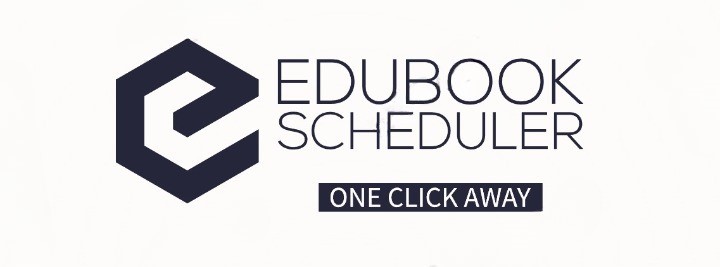


Project Plan



CONTENTS

[PROJECT ASSIGNMENT 2](#_Toc149764978)

[CONTEXT 2](#_Toc149764979)

[GOAL OF THE PROJECT 2](#_Toc149764980)

[SCOPE AND PRECONDITIONS 2](#_Toc149764981)

[STRATEGY 3](#_Toc149764982)

[RESEARCH QUESTIONS AND METHODOLOGY 3](#_Toc149764983)

[END PRODUCTS 4](#_Toc149764984)

[PROJECT ORGANIZATION 5](#_Toc149764985)

[STAKEHOLDERS AND TEAM MEMBERS 5](#_Toc149764986)

[COMMUNICATION 5](#_Toc149764987)

[ACTIVITIES AND TIME PLAN 5](#_Toc149764988)

[PHASES OF THE PROJECT 5](#_Toc149764989)

[TIME PLAN AND MILESTONES 6](#_Toc149764990)

[TESTING STRATEGY AND CONFIGURATION MANAGEMENT 8](#_Toc149764991)

[TESTING STRATEGY 8](#_Toc149764992)

[TEST ENVIRONMENT AND REQUIRED RESOURCES 9](#_Toc149764993)

[CONFIGURATION MANAGEMENT 10](#_Toc149764994)

[FINANCES AND RISK 11](#_Toc149764995)

[RISK AND MITIGATION 11](#_Toc149764996)

# PROJECT ASSIGNMENT

## CONTEXT

Our current project is the development of an application called "EduBook Scheduler," assigned by our client, SIMAC. This application aims to simplify the booking of meetings between students and tutors in the academic niche.

## GOAL OF THE PROJECT

**Problem Statement:** Traditional methods of scheduling academic meetings are very time-consuming, leading to ad-hoc arrangements, communication issues, and a lack of centralized information. Our goal is to provide a modern, user-friendly IT solution that solves these challenges.

**Advantages and Value Addition:**

* Centralized, innovative, user-friendly scheduling system
* Real-time updates and notifications
* Improved communication between tutors and students

## SCOPE AND PRECONDITIONS

**Inside Scope (What's Included):**

1. Development of a user-friendly web-based booking system.
2. User registration and authentication.
3. Meeting search and booking functionality.
4. Centralized storage for meeting outcomes.
5. Real-time updates and notifications.
6. Features for efficient communication.

**Outside Scope (What's Not Included):**

1. Integration with external IT systems.
2. Payment processing.
3. Development of a mobile app.
4. Extensive marketing and legal compliance.
5. Comprehensive customer support features.

**Preconditions:**

* Availability of necessary data.
* Access to required technology and tools.
* In accordance to legal and regulatory requirements.

## STRATEGY

**Project Management Approach:** Agile, specifically using the Scrum framework, will be used as the project management methodology. Our choice of Agile is justified by its adaptability, customer-centricity, risk management capabilities, and focus on delivering a fully functioning parts of the product each sprint. This approach allows for flexibility and continuous improvement throughout the project's lifecycle.

## RESEARCH QUESTIONS AND METHODOLOGY

**Research Questions:**

1. What are the current market trends?
2. What are the customer’s preferences in academic meeting booking?
3. What can we implement in our application to enhance user experience?
4. What is the best most convenient way to optimize the system’s performance and

reliability?

**Methodology:**

* + Market research, surveys, interviews.
  + Usability and A/B testing.
  + Load testing, security assessments.

## END PRODUCTS

# PROJECT ORGANIZATION

## STAKEHOLDERS AND TEAM MEMBERS

|  |  |  |  |
| --- | --- | --- | --- |
| **Role** | **Name** | **Role and Functions** | **Availability** |
| Project Manager | Ömer Faruk Koran | Overall project  management and coordination. | Full-time, throughout the project |
| Product Owner | Hristo Ganchev | Represents client's interests and priorities. | As needed, for requirement clarifications |
| Scrum Master | Nicole Genova (we will rotate each meeting) | Facilitates Scrum meetings and process adherence. | Full-time, throughout the project |
| Client | Christian van Deuren | Represents SIMAC company | As needed, for project updates and approvals |
| Tutor | Ruiter, Senne S.M. de | Provide academic guidance and mentorship | As needed, for academic inputs and feedback |

## COMMUNICATION

The communication with the stakeholder and tutor will be established primarily face-to-face. However, in cases of importance, our team has been provided with their emails if face-to-face communication is not possible. Communication with the tutor should be established at least once a week in order to receive feedback, made reflections on it and update our work. On the other hand, meetings with the client are scheduled for once every three weeks.

# ACTIVITIES AND TIME PLAN

## PHASES OF THE PROJECT

**Sprint A: Project Initialization**

Start Date: 29.09.2023

Finish Date: 13.10.2023

**Sprint B:**

Start Date: 13.10.2023 Finish Date: 13.11.2023

**Sprint C:**

Start Date: 13.11.2023 Finish Date: 04.12.2023

**Sprint D:**

Start Date: 04.12.2023 Finish Date: 18.12.2023

**Sprint E:**

Start Date: 08.01.2024

Finish Date: 26.01.2024

## TIME PLAN AND MILESTONES

**Sprint A: Project Initialization**

Goals:

Create a project plan.

Develop an initial product backlog.

Set up the basic backend infrastructure.

**Sprint B:**

Goals:

Improve the UI design.

Set up a database and connect it to the application.

Create unit tests for the application and connect repository to SonarQube.

Implement availability management system.

**Sprint C:**

Goals:

Email notification system.

Booking management system.

Tutor filling their availability.

Updated dashboard design

**Sprint D:**

Goals:

Websockets (user notifications) implementation

Making UI more user-friendly

Monthly availability and assigning of students to tutors

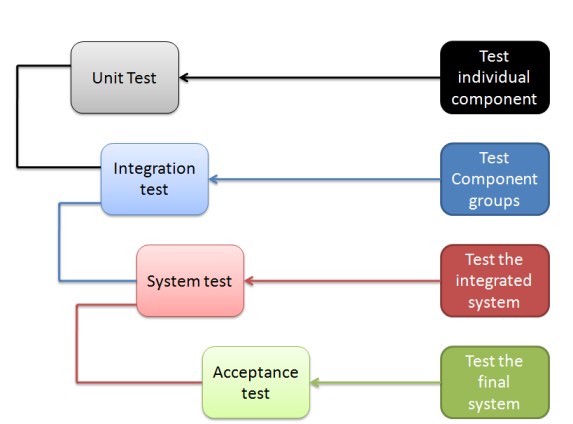
**Sprint E:**

Goals:

Polishing overall functionality based on stakeholder’s requirements

# TESTING STRATEGY AND CONFIGURATION MANAGEMENT

## TESTING STRATEGY



For this project, a comprehensive testing strategy that covers multiple levels and security are to be used:

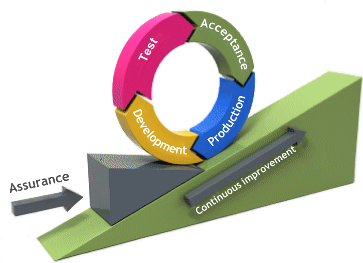
**Unit Testing:** We will conduct unit testing for individual components of our system, including the backend and frontend. The goal is to ensure that each component functions correctly in isolation. Since this is the final stage of testing, for Sprint E, we aim to achieve at least 80% code coverage for our unit tests, which will be automated.

**System Testing:** System testing will focus on testing the system as a whole. We will evaluate the end-to-end functionality of the booking system. This includes testing various user scenarios, such as searching for meetings, booking meetings, and user authentication.

System tests can be automated, but we will most likely consider testing functionality manually.

**Acceptance Testing:** Acceptance testing will involve testing the system against the acceptance criteria defined in the user stories. We will collaborate closely with stakeholders to validate that the system meets their requirements.

## TEST ENVIRONMENT AND REQUIRED RESOURCES

Our test environment will follow the DTAP pipeline, and it will consist of the following components:

**Development Environment:** Developers will have individual development environments for coding and unit testing.

**Testing Environment:** We will maintain a dedicated testing environment where system and acceptance testing will take place.

**Acceptance Environment:** This environment is specifically for user acceptance testing (UAT), where stakeholders can assess the software to ensure it meets their requirements and expectations.

**CI/CD Environment:** We will set up a continuous integration and continuous delivery (CI/CD) pipeline to automate the testing and deployment processes. This environment will include tools like Jenkins and GitLab CI.

## CONFIGURATION MANAGEMENT

We will use Git as our version control system. We'll follow a straightforward workflow, primarily using additional branches for development and later merging everything into the main branch for the final product. We'll document and manage change requests using Jira.

We'll establish baselines for various project artifacts, including code, documentation, and design. These baselines will serve as reference points for tracking changes and ensuring consistency throughout the project.

This configuration management approach will provide structure and control over the project's development and testing processes while maintaining traceability and accountability.

# FINANCES AND RISK

## RISK AND MITIGATION

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Risk** | **Impart** | **Probability** | **Prevention activities** | **Mitigation activities** |
| 1. The project may get deleted or lost | Huge | Low | Every project version will be available to download from GitLab | In case of the project getting lost, it can be downloaded again from GitLab |
| 1. Code may be lost or deleted | Huge | Medium | Every project version will be available to download from GitLab | In case of code getting lost, it can be downloaded again from GitLab |
| 1. The application may crash or restart, leading to loss of work | Huge | High | The project will constantly be saved in order to not lose work | In case the application crashes, the file will already be saved and no major changes will be lost |
| 1. The developer/s may experience a burnout | Huge | Medium | The developer/s will have a perfect balance between working time and rest | The developer/s will improve their focus in order to complete work efficiently and have more time to “cool down” |
| 1. Miscommunication may occur between the developers and the stakeholders | Huge | Medium | All of the stakeholders’ requirements will be written down | The requirements will be noted down in a short span of time after a discussion is made in order to not forget important details, features, or ideas |
| 1. The stakeholders may be absent when needed | Medium | Medium | Feedback from the stakeholders will be asked for constantly so that project development is not slowed down in case of stakeholders’ absence | Stakeholders can be contacted via email or Teams |
| 1. The developers may stumble upon a very difficult bug to fix | High | High | The developers will ask for assistance and look through online resources in order to fix the said bugs. | The developers can create a separate branch from the one where a bug has occurred so that they can continue working on other features. |