Bilkent University

Department of Computer Engineering

CS 413

Software Engineering Project Management

2023-24 Spring Semester

Project Management Plan

Project Name: CODED.

Group Name: Blank Space

Project Team Members:

Zeynep Hanife Akgül	(22003356)
Beyza Çağlar	(22003394)
Alper Göçmen	(22002948)
Deniz Tuna Onguner	(22001788)
İlayda Zehra Yılmaz	(22001769)

Instructor: Onur Karabulut

7 April 2024

Table of Contents

1. Executive Summary

CODED. is a website application that assists students, tutors, and instructors in CS (Computer Engineering) and CTIS (Information Systems and Technologies) labs at Bilkent University throughout the entire lab process. The application has four features: a chatbot, a code analyzer, a test case runner, and a grading bot. The chatbot assistant answers students' questions related to their code without revealing the "exact" answer but giving clues as to how to reach a possible solution in order to encourage the student to learn. The similarity check feature does a similarity check among codes on Github/sections/all past codes/classes, based on the instructor's preferred settings. The clean code evaluator checks students' code quality, gives them feedback on the quality of their code, and helps them in the debugging process. Finally, the test case runner feature does automated tests on student codes to provide an auto-grading system. Overall, CODED. tutors lab students while minimizing the effort of lab assistants and instructors.

2. Front Matter

The scope, budget, objectives, deliverables, timeframes, and resource requirements of the project are all outlined in this project management plan. Its main goal is to provide a thorough understanding of the project's context, scope, and the procedures and guidelines that must be followed during its entire lifecycle. This plan is meant for the project team, sponsors, stakeholders, CEO, VP of Engineering, and anyone else involved in the planning, execution, and supervision of the project. As a preliminary framework, the PMP provides stakeholders with an outline of the goal, parameters, and intended audience of the project, laying the foundation for the subsequent specific information.

Project Name: CODED.
Date of Issue: 07.04.2024
Draft Number: CDD-2024-001
Issuing Organization: BlankSpace Inc.

3. Project Overview

3.1. Project Summary

3.1.1. Purpose, Scope, and Objectives

The purpose of this project, named **CODED.**, is to address the issue of a shortage of lab tutors in the CS and CTIS departments at Bilkent University. The project aims to develop a website application that will assist students, tutors, TAs, and instructors throughout the lab process.

The application will have several key features:

1. A chatbot that can answer student questions about lab assignments and guide them in writing clean code.
2. A plagiarism checker that allows instructors to compare student code against various sources.
3. A grading system that checks the logic of the code, runs methods separately for correctness, and performs static checks for partial point calculations.

By implementing these features, the application aims to reduce the time it takes to grade code and lighten the workload of tutors, TAs, and instructors. This will ultimately improve the efficiency of the lab sessions and enhance the learning experience for students.

The scope of the project encompasses the entire development lifecycle of the web application. This includes project initiation, requirements gathering, design, development, testing and quality assurance, deployment and launch, and project closure. It also incorporates data management processes and the integration of necessary backend systems, ensuring compliance with data protection regulations. The project will support specific CTIS and CS courses, leveraging a diverse technology stack and open-source frameworks. The scope also includes the development of key features such as a chatbot, automated code quality checks, plagiarism detection, and an advanced grading tool.

The project will deliver a fully functional web application designed to ease the workload of instructors, TAs, and tutors while maintaining the quality of education for

students. The application will provide students with immediate answers to their basic code-related questions, allowing academic personnel to handle more complex issues. The project will follow an agile methodology with regular deliverables to ensure alignment with stakeholder requirements. The planned deliverables are as follows:

- Project Charter and Sprint Backlog
- Project Management Plan (PMP)
- Requirements Analysis Document
- System Design Document
- Mockup
- Codebase and Documentation
- Functional Prototype
- Minimum Viable Product (MVP)
- Test Plans
- Alpha Version
- Alpha Version Testing and Code Quality Report
- Beta Version
- Beta Version Testing and Code Quality Report
- User Acceptance Testing (UAT)
- Final Release
- User Manual
- Final Budget Report
- Project Closure Report

The web application addresses the pressing need for a comprehensive and efficient solution to assist in the lab process at Bilkent University. The current system relies heavily on the availability and capacity of tutors, TAs, and instructors, which often leads to a high student-to-tutor ratio and limits the quality of education. This project aims to bridge this gap by providing an automated platform that can handle basic code-related questions, thereby freeing up academic personnel to handle more complex issues. The system will consider the requirements of the lab assignments and the coding language to provide accurate and helpful responses. By combining a chatbot for immediate answers, automated code quality checks, plagiarism detection, and an advanced grading tool,

CODED. aims to become an indispensable tool in the CS and CTIS labs. The system will ensure that the quality of education is maintained while reducing the workload of the academic personnel. The satisfaction of the project objectives will be determined through user feedback, the efficiency of the grading process, and the reduction in workload for the academic personnel. The ultimate goal is to enhance the lab experience for both the students and the academic personnel.

The quality goals for the CODED. are aimed at ensuring that the software components meet best practices and quality standards, achieve high code coverage, satisfy user expectations, maintain system uptime, reduce errors, and design a scalable system. To assess and control quality, key performance metrics are measured, including compliance, code coverage, defect density, user satisfaction, uptime, and scalability. Various mechanisms such as assessments, reviews, and audits are implemented to maintain and improve quality. Reports are generated to provide transparency and support decision-making. Overall, the project's quality assurance plan ensures that work processes and deliverables align with the defined quality goals, leading to user satisfaction and project success.

Please refer to the **'Requirements Analysis Document'** for the official statement of product requirements for this project. This document, approved by the stakeholders, outlines the functional and non-functional requirements of the application. It provides detailed descriptions of the system features, performance requirements, design constraints, and other specifications necessary for the development of the application."

3.1.2. Assumptions and Constraints

This project is based on several assumptions and constraints that will guide the project's execution and management.

Assumptions:

1. **Availability of Stable Internet Connection**: It is assumed that stable internet connectivity will be provided in the labs, enabling students to access the web app's services during lab sessions.

2. **Availability of Requisite Hardware**: It is assumed that students in labs will have access to devices/computers capable of running a web browser to utilize the web app.
3. **Compatibility with Primary Web Browsers**: The web app is assumed to be compatible with commonly used web browsers such as Safari, Chrome, and Firefox.
4. **Active User Engagement**: It is assumed that students in labs will actively engage with the web app during lab sessions.
5. **Exclusive Use of the Web App**: It is assumed that students in labs will solely use the web app and no other tools or services to enhance consistency and control over lab assignments.
6. **Systematic User Training**: It is assumed that necessary training and support will be provided to students and instructors, especially those in non-engineering majors, to use the web app as expected.
7. **User Proficiency**: Assuming that users have a basic understanding of the coding languages supported by the application.
8. **Instructor Involvement**: Assuming that instructors will actively use the application for grading and providing feedback.

Constraints:

1. **Time Constraints**: The web app needs to be fully developed and ready for deployment before the start of the next semester to align with the academic calendar.
2. **Performance Constraints**: The web app needs to handle a varying number of users/students during lab sessions without experiencing performance issues.
3. **Instructor Discretion**: Instructors should have the ability to disable any of the web app's services (i.e., chatbot, plagiarism check, code quality measurement, and pass-failure of test cases) as they see fit.
4. **Privacy Constraints**: Given that the system will store both personal and academic data, the web app needs to comply with data privacy regulations such as GDPR.

5. **Regulatory Compliance**: The project must comply with any intellectual property rights and licenses related to the use of third-party data sources or software.

3.1.3. Project Deliverables

- **Project Charter and Backlog (delivery in the 1st week):** The project charter and backlog serve as a comprehensive outline to the project, including its scope, objectives, stakeholders, and requirements, thereby laying the basis to plan the initial sprint.
- **Project Management Plan (delivery in the 2nd week):** The PMP defines the scope, objectives, goals, budget, timeline, and resource management of the project, providing a comprehensive framework for the project execution.
- **Requirement Analysis Report (delivery in weeks 2nd to 3rd):** The requirement analysis report documents both stakeholders and software requirements, containing both functional and non-functional dependencies.
- **System Design Document (delivery in the 3rd week):** The system design document serves as a guide for implementing the system by demonstrating its general architecture and components.
- **Mockups (delivery in weeks 3rd to 4th):** In order to give the stakeholders a visual demonstration of the proposed/planned user interface layout, mockups are used to showcase the initial User Interface (UI) design.
- **Codebase and Documentation (delivery in the 4th week):** The codebase and documentation encompass the initial source code and the generic repository structure, laying the foundation for the development and the maintenance of the project.
- **Functional Prototype (delivery in weeks 4th to 5th):** The functional prototype is the initial working version of the software, providing a physical representation to the project for the initial validation and testing.
- **Minimum Viable Product (delivery in the 6th week):** The minimum viable product, or shortly MVP, is the most basic, but operating, version of the final product that provides value for customers and renders possible the incremental development depending on feedback from customers.

- **Test Plans (delivery in the 7th week):** Test plans assure extensive testing as well as quality assurance throughout the application's development lifecycle by identifying the test strategy for the future versions.
- **Alpha Version (delivery in weeks 8th to 9th):** The Alpha Version of the software is for internal testing purposes within the development team.
- **Alpha Version Testing and Code Quality Report (delivery in the 9th week):** The report on the results of the Alpha Version tests presents the outcomes, providing insights to the team regarding the software's performance.
- **Beta Version (delivery in the 12th week):** The Beta Version represents the iteration designated for testing with stakeholders and, possibly, with a selected group of users.
- **Beta Version Testing and Code Quality Report (delivery in the 15th week):** The report on the results of the Beta Version tests presents the outcomes, providing insights to the team and to the stakeholders regarding the software's approximate performance.
- **User Acceptance Testing (delivery in weeks 17th to 18th):** User Acceptance Testing (UAT) is to determine whether the application is ready to be deployed.
- **Final Release (delivery in the 21st week):** The end product that is ready to be deployed.
- **User Manual (delivery in the 22nd week):** The documentation for end users, explaining how the application is needed to be used properly.
- **Final Budget Report (delivery in weeks 22nd to 23rd):** The report for the finalization of expenditures and budget allocations.
- **Project Closure Report (delivery in the week 24th):** The Project Closure Report provides a comprehensive summary of the project, including its overall outcomes, the lessons learned by the development team, and the relevant future plans, serving as a valuable reference for the stakeholders aiming to inform them for the possible future projects.

3.1.4. Schedule and Budget Summary

Our project is designed to be completed over a span of 24 weeks, divided into 8 sprints each lasting three weeks. Below is an itemization of major work activities and processes as per the work breakdown structure:

- **Sprint 1 (Weeks 1-3):** Project Charter and Sprint Backlog, Project Management Plan (PMP), Requirements Analysis Document, System Design Document
- **Sprint 2 (Weeks 4-6):** Mockup, Codebase and Documentation, Functional Prototype, Minimum Viable Product (MVP)
- **Sprint 3 (Weeks 7-9):** Test Plans, Alpha Version, Alpha Version Testing and Code Quality Report
- **Sprint 4 (Weeks 10-12):** Beta Version
- **Sprint 5 (Weeks 13-15):** Beta Version Testing and Code Quality Report
- **Sprint 6 (Weeks 16-18):** User Acceptance Testing (UAT)
- **Sprint 7 (Weeks 19-21):** Final Release
- **Sprint 8 (Weeks 21-24):** User Manual, Final Budget Report, Project Closure Report

A sprint review will be held after each sprint to analyze the work done and make changes to the product backlog, if needed.

In terms of the budget, we have analyzed the costs related to every component of the project. There are six budget components which are Engineering, Business, Hardware, Software, Services and Other costs. The total budgets are given as follows:

- Engineering Person-Months Cost: $1,392,000
- Business Person-Months Cost: $360,000
- Hardware Cost: $20,000
- Software Cost: $39,000
- Services Cost: $6,000
- Other Cost: $7,500

The overall cost, before profit, is estimated at $1,824,500. The profit margin is 25%, so the profit will be approximately $456,125. Thus, the total budget for the project, including profit, is approximately $2,280,625.

Payments will be provided based on some major project milestones. The payment schedule is as follows (assuming T0 is starting time):

Initial Payment (T_0): 20% of the total budget, amounting to $364,900. This covers the project initiation and planning phases, including the creation of the Project Charter, Sprint Backlog, and Project Management Plan.

Post-Requirements Analysis (T_0 + 2-3 weeks): 20% of the total budget, amounting to $364,900. This payment is due after the completion of the Requirements Analysis Document and System Design Document, marking the end of the detailed planning phase.

Post-Prototype Development (T_0 + 4-5 weeks): 20% of the total budget, amounting to $364,900. This payment is due after the development of the Functional Prototype and Minimum Viable Product (MVP), indicating the transition from planning to development.

Post-Beta Version (T_0 + 12 weeks): 20% of the total budget, amounting to $364,900. This payment is due after the release of the Beta Version, marking the beginning of the testing phase.

Final Payment (T_0 + 21 weeks): 20% of the total budget, amounting to $364,900. This final payment is due after the Final Release and the completion of the User Manual, Final Budget Report, and Project Closure Report, marking the successful completion of the project.

This payment schedule is tied to significant project milestones, ensuring that payments are made in line with project progress. It should be noted that this is a suggested payment schedule and may need to be adjusted based on occurring projects risks, and other factors. In that case, any such modifications will be reported to all stakeholders in advance.

3.1.5. Evolution of the Plan

The effectiveness of the project plan is critical to its success. Given that projects are dynamic, this section describes how the project plan will change over time to meet altering requirements and conditions.

The initial project plan was prepared by the development team with the inputs provided by the stakeholders. Consequently, the Agile approach is preferred by the team for the overall development process, which allows iterative development.

To manage changes effectively within the team and with stakeholders, a project management software application will be utilized. All changes and updates on the project, either scheduled or unscheduled, will be deployed to the team members and stakeholders via this application.

In addition, ensuring that all the stakeholders are notified of modifications to the project plan requires effective communication. There will be regular updates on the status of the project via communication channels.

Using documentation tools, the project plan will be documented, and version control will be upheld to track alterations over time. All the members of the project team and stakeholders will have access to a central repository containing all project material, including the project plan.

4. References

[1] Lecture slides for CS 413 Software Engineering Project Management on Moodle.

5. Definitions

AWS: Amazon Web Services

CI/CD: Continuous Integration/Continuous Deployment

GDPR: General Data Protection Regulation

MVP: Minimum Viable Product

PMP: Project Management Professional

SEE: Software Engineering Environment

UI: User Interface

UI/UX: User Interface/User Experience

UAT: User Acceptance Testing

UML: Unified Modeling Language

VPN: Virtual Private Network

6. Project Context

6.1. Process Model

The project utilizes the Agile Scrum method, which is an incremental development method encouraging collaboration and flexibility in response to changes in the project. The total duration of the project is planned to be 24 weeks divided into 8 3-week sprints.

Sprint 1 (Weeks 1-3): Project Kick Off

- **Objective:** Initiate the project by defining the project objectives, scope, requirements, and stakeholders.
- **Activities:**
    - Describe the project objectives, scope, requirements, and stakeholders.
    - Meet with stakeholders for the detailed requirements analysis.
    - Design the system architecture with respect to the PMP.
    - Prepare the project environment with necessary tools such as hardware or software tools.
- **Responsibilities:**
    - The project manager should supervise the project framework composition.

- The product owner and the development team should start working on the project infrastructure and plan the stakeholder requirements.

Sprint 2 (Weeks 4-6): Design of the Primary Functionalities

- **Objective:** Determine and construct the main functionalities and features of the project.
- **Activities:**
  - Review user stories.
  - Identify the key features and functionalities.
  - Outline the system design.
  - Prepare a mockup of the end product.
  - Develop a functional prototype of the software.
  - Improve the functional prototype and create a Minimum Viable Product (MVP).
- **Responsibilities:**
  - The product owner should assist in reviewing all the user stories and clarify any ambiguity.
  - The project manager should ensure that the activities are completed within the time and budget constraints while maintaining healthy communication between the product owner, the stakeholders, and the development team.
  - The project manager should ensure that the identified functionalities, the functional prototype and the MVP is within the scope of the project.
  - The developers should understand the user stories well, design the appropriate system, identify main functionalities, and create an MVP within the project constraints.

Sprint 3 (Weeks 7-9): Enhancement of the Primary Functionalities

- **Objective:** Improve the MVP for an Alpha version release.
- **Activities:**
  - Create testing strategies and plans for future versions and the end product.
  - Release the Alpha version of the software.
  - Conduct testing of the Alpha version and address any issues.
- **Responsibilities:**

    - The product owner should assist in the creation of testing strategies and the implementation of the required changes.
    - The project manager should organize the Alpha version release, give feedback on the testing scenarios, and facilitate coordination between different teams.
    - The development team should enhance the MVP and implement the changes urged by the product owner to the Alpha version.
    - The quality assurance and testing team should conduct the planned tests on the Alpha version.

Sprint 4 (Weeks 10-12): Implementation of the Secondary Functionalities

- **Objective:** Determine and construct the main functionalities and features of the project.
- **Activities:**
    - Review user stories.
    - Identify the secondary features and functionalities.
    - Improve the Alpha release to create a Beta version.
- **Responsibilities:**
    - The product owner should ensure that the stakeholder requirements are followed by reviewing the user stories with the development team.
    - The project manager should ensure that the project constraints are followed such as the deadlines and the budget limits.
    - The development team should review user stories again and implement the necessary features within the project constraints.

Sprint 5 (Weeks 13-15): Enhancement of the Secondary Functionalities

- **Objective:** Test the Beta release and make improvements based on the tests.
- **Activities:**
    - Test the Beta version.
    - Identify the changes to be made to improve the final product.
- **Responsibilities:**
    - The product owner should monitor the changes planned so that the end product is in line with the stakeholder requirements.

    - The project manager should oversee the Beta version release, evaluate the tests, and expedite the communication between teams.
    - The development team should implement the additional features and create a viable product for the Beta release.
    - The quality assurance and testing team should handle the Beta version tests.

Sprint 6 (Weeks 16-18): Advanced Testing and Quality Assurance

- **Objective:** Prepare the Beta version to be released as the final product via User Acceptance Testing (UAT).
- **Activities:**
    - Implement the identified changes to the Beta version.
    - Conduct UAT on the improved product.
    - Identify any further changes to the software based on the feedback from the stakeholders and the users.
- **Responsibilities:**
    - The product owner should assist the development team in preparing the UATs and regulate the final updates to the product with regard to the stakeholder expectations.
    - The project manager should supervise the effective communication between the stakeholders and the team.
    - The project manager should ensure that the end product satisfies the necessary constraints in the PMP.
    - The development team should implement the improvements to the Beta version and prepare the UAT scenarios to be run afterward.
    - The quality assurance team should help with the preparation and running of the UATs, analyze the feedback from the stakeholders and the users, and ensure that the application adheres to the quality standards of the project.

Sprint 7 (Weeks 19-21): Release of the Final Product

- **Objective:** Release the final product to the end users.
- **Activities:**

    - Make the final adjustments to the product before release.
    - Release the end product.
- **Responsibilities:**
    - The product owner should make sure that the final product meets the stakeholder requirements.
    - The project manager should ensure that the final adjustments to the product align with the project timelines, budget, and quality standards.
    - The project manager should monitor the feedback loop post-release for a successful product launch.
    - The development should implement the final adjustments to the product in line with the product owner's and the project manager's requirements.
    - The development team should fix any bugs or issues with the product after they test the product.

Sprint 8 (Weeks 22-24): Project Finalization

- **Objective:** Finalize the project documents.
- **Activities:**
    - Prepare the user manual to guide users on how to use the product.
    - Assemble the final budget report.
    - Create the project closure report, which contains information on possible future improvements, lessons learned, and outcomes.
- **Responsibilities:**
    - The product owner should ensure that the product and the user manual correctly represents the stakeholder requirements.
    - The project manager should monitor the user manual.
    - The project manager should prepare the final budget report and the project closure report, with optional insight from the other team members.
    - The development team should prepare a comprehensive user manual that correctly represents the end product and its use cases.

6.2. Process Improvement

Since the project adopts an agile approach, continuous process improvement is crucial. There are several improvements that can be listed as periodic assessments, root cause identification, incremental improvement.

The project will follow a systematic approach to process improvement, using periodic assessments throughout its duration. Periodic assessments will be conducted at key milestones, evaluating project performance and identifying areas for improvement. These assessments will include various aspects of the project, including the functionality of the website application, user interactions, and overall project management processes.

Along with periodic assessments, the problem-resolution planning will be closely related to process improvement planning. Root causes of occurring problems will be identified with a root cause analysis as this identification can lead to simple process improvements that can significantly reduce rework during the remainder of the project.

Furthermore, an incremental process improvement will be applied such that without causing any disruptions to the project activities, appropriate changes will be done. This allows for a continuous integration of changes to the workflow and seamless adaptation of the project to current requirements.

6.3. Infrastructure and Enabling Systems

This section specifies the plan for production systems, software engineering environments (SEE), maintenance systems, and project support systems, encompassing hardware, operating systems, network infrastructure, and software tools.

As infrastructure, the project will have workstations with cutting-edge technology, providing computational capacity and speed which is crucial especially for the chatbot feature. The production systems within the infrastructure will include high-powered deployment environments, for both the frontend and backend components. Known for their powers for writing, debugging, and testing code efficiently, IntelliJ and VScode will be used by the developers, which are part of the SEE's used in the project. Additionally, GitHub will be used as the central repository for version control, enabling undisturbed collaboration and code management among team members. CODED. consists of the combination of hardware, operating

systems, and network infrastructure to support development and deployment activities. Developers will use classical laptops equipped with suitable hardware configurations to code, test, and debug the application locally. The operating systems that will be used are Windows, macOS, or Linux distributions, tailored to individual developer preferences. Network connectivity will be established through secure VPN connections to ensure secure access to project resources and services.

As for enabling systems, software tools such as development and management tools, and several licenses are included. These tools empower the project team with a complete framework for project design, implementation, and testing. Along with development tools and licenses, the project will use project management tools in order to track progress, allocate resources, manage risks, and provide effective communication within the team and with the stakeholders. Securely storing data and deploying the project is crucial, so the project will use AWS cloud services provider, chosen for its scalability and accessibility while maintaining the security.

6.4. Methods, Tools, and Techniques

The project will follow the Agile software development methodology, specifically Scrum, to continuously and incrementally deliver value to stakeholders. This approach allows for flexibility, collaboration, and adaptability to changing requirements. Sprints will be held in short terms, usually in two weeks, with regular sprint planning, review, and retrospective meetings to ensure continuous improvement.

CODED. will be developed using JavaScript for frontend development with React.js framework and Java for backend development with Spring Boot framework. Additionally, Python will be used for the chatbot development. For data management, PostgreSQL will be used for its scalability and versatility. For high quality code production, continuous deployment (CI/CD) pipelines will be used with appropriate tools like Jenkis.

Platforms like Jira and Notion will be used for project management to help keep track of project activities, timelines, deadlines, work done, to dos, and such components related to project tracking. Furthermore, Telegram is planned to be used for effective communication among teams, when the team is not working face to face. Unified Modeling Language (UML) diagrams will be created before the implementation process for specification and design, including object and class models, sequence diagrams, and use case diagrams to visualize system architecture and

requirements. For creating these diagrams, Draw.io will be used for its user-friendly interface and comprehensive features. For the UI/UX designs, Figma will be used by the team.

6.5. Product Acceptance

The project should meet the objectives for it to be an acceptable product. To meet the project requirements, every deliverable of the project should work toward a mutual goal. The planned sprints of the project divides the workload into a total of eight 3-week-periods. Each deliverable in every single sprint has a set of criteria to be accepted as a feasible product. Thus, this section explains the acceptance criterion for each deliverable and sprint of the project.

Sprint 1 (Week 1-3): Project Kick Off

The first sprint aims to set up the environment for a productive development process. Therefore, the deliverables of the first sprint should define the basic constraints of the project such as the scope, the requirements, the budget, and the timeline. A meeting with the stakeholders should be held for comprehensive requirements analysis. The tools such as the hardware, software and the communication tools that will be used during the project should be decided and prepared in this sprint as well. A document demonstrating the relationship between different tools will be encouraged for product acceptance. Overall, this sprint can be completed when the product backlog has been assembled based on the requirements' analysis, the system design has been prepped, and the project infrastructure has been initiated.

Sprint 2 (Week 4-6): Design of the Primary Functions

The main goal of the second sprint is to extract the core functionalities of the application based on the requirements' analysis and user stories. Mockups of the application design, a functional prototype, and an MVP should be produced at the end of this sprint. The product owner should approve each deliverable for the product acceptance. The deliverables should meet the quality standards set by the project manager and the product owner. The sprint can be completed after the MVP is approved by the product owner, the stakeholders and the project manager.

Sprint 3 (Week 7-9): Enhancement of the Primary Functions

In the third sprint, the MVP will be tested and improved for an Alpha version release. The Alpha version only includes the core functionalities of the application. It allows for exhaustive testing of the implemented features by the internal team members. The sprint can be completed once the released Alpha version is tested and approved.

Sprint 4 (Week 10-12): Implementation of the Secondary Functionalities

This sprint aims for a Beta version release. The Alpha version will be enhanced with the secondary functionalities and bug fixes to create the Beta version. The Beta version may take longer to implement compared to the Alpha version as it is more expansive. This sprint can be concluded when the Beta version is approved by the team and released.

Sprint 5 (Week 13-15): Enhancement of the Secondary Functionalities

The fifth sprint includes the testing and quality assurance of the Beta release. After the code quality report for the Beta version is created, the improvements to the Beta version should be determined. This sprint can be completed once the improvements are accepted to be within the scope of the project by the product owner.

Sprint 6 (Week 16-18): Advanced Testing and Quality Assurance

In this sprint, the improvements to the Beta version should be implemented and the new product should go through UATs. The UATs will be conducted by the stakeholders and the users as well as the development team. Therefore, their review of the product will also affect the quality assurance. This sprint can be concluded once the stakeholders, the users, the product owner, and the project manager approves the product's functionalities via the UATs.

Sprint 7 (Week 19-21): Release of the Final Product

This sprint includes the release of the final version of the project. The Beta version will be enhanced based on the feedback given by the stakeholders in the UATs. After the enhancements, the end product should also undergo testing. Once the final tests have been finished and the end product is released with approval from the product owner and the project manager, this sprint can be completed.

Sprint 8 (Week 22-24): Project Finalization

In this sprint, the final documents for the project closıre will be created. The project manager should finalize the project if every aspect of the project satisfies the project quality standards. The end product should provide the users with a secure, intuitive, easy-to-use, and beneficial experience. All the deliverables produced during the project should be comprehensive, proper and well-structured.

6.6. Project Organization

6.6.1. General

This section is added to identify interfaces to organizational entities external to the project, describe the project's internal organizational structure, and specify roles and responsibilities for the project.

6.6.2. External Interfaces

The "CODED." project interfaces with several external organizational entities, including Bilkent University's Computer Engineering and CTIS departments, potential industry partners, and other universities. Communication with these will occur through various channels such as email, virtual meetings, and collaboration platforms like email applications, Discord, Google Teams, and Zoom. Responsibilities for managing these interfaces are distributed among project team members. These roles are mentioned below, under the *Authorities and Responsibilities* heading.

6.6.3. Internal Interfaces

The internal organizational structure of the CODED. project is a cross-functional team framework. Reporting relationships are agile, which creates a collaborative and quick decision-making environment. The project is divided into different areas, such as development, testing, design, and project management. Each area is led by a team leader responsible for coordinating activities, managing resources, and ensuring that the deliverables align with project goals.

6.6.4. Authorities and Responsibilities

Key roles and responsibilities within the project include:

Project Manager: Responsible for stakeholder management, risk mitigation, and resource allocation. Also monitors the project throughout the planning and execution. Engages with the external interfaces through communication strategies.

Technical Leads: Responsible for ensuring technical feasibility, and alignment with architectural standards. They lead the development processes, and design efforts.

Developers: Developers are responsible for implementing and coding the features of the project so that it meets the requirements in the specification and design documentation.

Testers: Responsible for running tests on software components in order to find and identify the issues and defects so that the product fulfills quality requirements.

UX/UI Designers: Responsible for designing and creating visually appealing, user-friendly, intuitive, and creative interfaces while taking the user's needs into consideration.

Business Analysts: Responsible for ensuring a healthy communication between the stakeholders and the development teams. They define user stories and analyze the requirements.

Each team member's responsibilities are clearly defined with project objectives, ensuring a collaborative and efficient working environment.

7. Project planning

7.1. General

This section shall specify the project management processes of the CODED. project. This part includes the plans to ensure the smooth execution of project initiation, work, acquisition, and supply, as well as the project’s assessment and control plans and, finally, project closeout

plan. This planning is done so that the project can perform according to the project's stakeholder's expectations, every stakeholder is aware of their roles, and the team can perform their duties in an efficient manner.

7.2. Project Initiation

7.2.1. General

This section defines the project's initiation using the constraints estimation of project scope, the staffing, and plans for the acquisition of resources that will be used to support the staff of the project. All the entry criteria must be satisfied regarding estimation, staffing, resource acquisition, and project staff training to start the project. The estimation will be made based on the cost, schedule, and resource baselines. The staffing part will include the resources and requirements for the project staffing regarding staff number, skill level and duration needed. Resource acquisition will examine the rest of the resources other than the staff members and their acquisition and realization. Project staff training will be examined on training needed to ensure the necessary training level.

7.2.2. Estimation

There are several categories in terms of CODED.'s estimations. As estimations are important for both the development team and stakeholders, this section is planned as realistic as possible.

7.2.2.1. Cost Estimation

The estimated total cost is $1,824,500. To predict this data, data from previous projects that have similar development teams and project duration is used. This way, a more consistent estimation can be made. However, it is not so possible for a project to be the same as another one, so the estimation has to regard the differences as well. Similarities and differences with the previous projects are listed, and an average estimation is made using the comparison technique. This estimation allows for monitoring the budget and staying within it during the development process.

7.2.2.2. Schedule Estimation

The project duration is estimated to be 24 weeks. The bottom-up strategy is used to estimate the schedule. In this approach, the schedule is created by gathering together the estimations for individual activities. Since this method breaks down the project scope into smaller components, they become more manageable, allowing for a detailed and more accurate estimation.

First, the tasks and the activities are defined. The tasks are identified from specifications documentation, project deliverables, and requirements. At this point, the activities are tried to be defined in small sizes so they fit the bottom-up strategy. After their identification, the project team analyzes each individual task, to estimate their duration. This duration is set based on the complexity of the task, available resources, and historical data. Once the durations are determined, individual estimations are then gathered together to find out the duration of one deliverable of the project. Like so, the duration of deliverables are added up, to find the duration of bigger parts. To define the whole schedule, dependencies and links between the activities are identified. Tasks that are dependent on each are planned to be executed in a sequential way, so their durations are added to the overall duration. There are also several uncertainties and risks related to some of the individual tasks. For these cases, buffers are allocated for the project to tolerate potential delays, without corrupting the execution of activities according to the schedule. Although a schedule is defined before the development process starts, it is important to note that this process is meant to be iterative, that the schedule can be readjusted when needed, along with the evolving project requirements and resources.

7.2.2.3. Resource Estimation

The project includes a resource estimation that stands for estimations of resources needed to deliver the whole project. To estimate the resources, experts, team leads, developers should collaborate to evaluate what kinds of activities need what kinds of and volumes of resources. Similar to schedule and cost estimation, previous data will be used for more accurate guesses. There are several processes of estimating the resources.

Regarding the requirements of the project, the resources are first identified. The identification process is held through taking experts, stakeholders, team leads and putting them together. Additionally, historical data is used for similar cases for a more accurate identification.

After deciding on the resources, resource requirements are set. Each resource has different requirements that show changes based on the project's needs, complexity and volume. Availability of these requirements is checked both internally and externally. Contingency plans are also made in cases of resource shortages or unforeseen developments. Then, resource allocation is done. To ensure a proper allocation, the resources' availability, complexities and urgency are checked and analyzed. As the project continues to develop, revision of these allocations may be required. Finally, the resource management techniques are applied to guarantee a seamless flow of resources.

7.2.3. Staffing

This part shows the number of required staff with their responsibilities. The sources of a specific staff can be an internal transfer or new hire. The internally transferred staff are the personnel already working in the company, and given a new responsibility within the project. The new hire personnel are the employees that are newly hired and will start their work in the company with CODED.. Below tables show the number of staff needed, their responsibilities and sources.

Staff Personnel	Number of Staff	Source of Personnel
Project Initiation		
Project Manager	1	Internal Transfer
Product Manager	1	Internal Transfer
Requirements Analysis		
Business Analyst	3	2 Internal Transfer, 1 New Hire
Team Leads	4	Internal Transfer
UI/UX Designers	3	2 Internal Transfer, 1 New Hire
Development		

Software Architect	1	Internal Transfer
Software Engineers	10	7 Internal Transfer, 3 New Hire
Candidate Engineers	3	New Hire
Testing and Quality Assurance		
QA Engineer	5	Internal Transfer
Security Engineer	5	Internal Transfer
Deployment		
Project Manager	1	Internal Transfer
Marketing Manager	1	Internal Transfer
Finance Manager	1	Internal Transfer
Technology Manager	1	Internal Transfer

7.2.4. Resource Acquisition

The resource acquisition and release will be done in the following process:

1. Identifying the resources: The resources in the context of CODED. will be equipment such as
    - The office equipment: Computers, monitors, other office necessities.
    - Computer hardware and software: Hardware such as servers, networking equipment, workstations, and software that is necessary for the project like applications, open source codes etc.
    - Staff: Engineers, Team Leaders, Managers, and Architects who are working on the project.
2. Evaluate the cost, quality, and availability of the resources and allocate them:

- The office equipment: The equipment mentioned above is easily acquired, some might need more budget like computers and monitors but acquiring them is easy and quality will be easy to find for our project.
- The accommodation resources: The office where the project will be developed will be rented for as long as the project takes (6 months)
- Computer hardware and software: The hardware systems will be included in the project budget like software applications and will be paid for as long as the project application is not removed or terminated.
- Staff: All of the staff will be paid for in accordance with their job, which necessitates them. The core team of software developers that might be used for fixing bugs or adding new features might be kept as long as the application is not removed or terminated.

3. Monitor the resources in the project life cycle and release them if they are no longer necessary:
    - The office equipment: These will be necessary until the project is terminated and will be kept by the company during the project life cycle. After that the company might decide to release resources if necessary but can use the resources for future projects.
    - The accommodation resources: After the project is finished in X months, the office will be released.
    - Computer hardware and software: These systems must be kept as long as the application is maintained and paid for accordingly without being released.
    - Staff: Because the core team of software developers might be necessary for fixing bugs or adding new features, they will be kept as long as the application is maintained, and they will not be released. However, project managers might be released as the team will be self-sufficient as an Agile team on its own.

7.2.5. Project Staff Training

- **Project Management Training:** Teaching the project management principles and best practices.
    - **Training Types:** Lectures, computer-assisted training, mentoring
    - **Number of Personnel:** 1 person

    - **Entry Criteria:** Basic understanding of project management concepts
    - **Exit Criteria:** Expansive comprehension of project management concepts
- **UI/UX Design Training:** Teaching the UI/UX design concepts and best practices.
    - **Training Types:** Lectures, online tutorials, mentoring
    - **Number of Personnel:** 3-4 people
    - **Entry Criteria:** Basic understanding of and experience in UI/UX design
    - **Exit Criteria:** Expansive comprehension of UI/UX design concepts
- **ML Engineering Training:** Teaching the ML concepts and best practices.
    - **Training Types:** Lectures, online tutorials.
    - **Number of Personnel:** 4-5 people
    - **Entry Criteria:** Basic understanding of ML concepts.
    - **Exit Criteria:** Expansive comprehension of ML concepts.
- **QA and Testing Engineering Training:** Teaching the QA and Testing concepts and best practices.
    - **Training Types:** Lectures, mentoring, consultations.
    - **Number of Personnel:** 4-5 people
    - **Entry Criteria:** Basic understanding of QA and Testing concepts.
    - **Exit Criteria:** Expansive comprehension of QA and Testing concepts.
- **Software Engineering Training:** Teaching the Software Engineering tools and practices used in the company.
    - **Training Types:** Lectures, mentoring, consultations.
    - **Number of Personnel:** 5-6 people
    - **Entry Criteria:** Basic understanding of software engineering concepts.

    - **Exit Criteria:** Expansive comprehension of the company's software engineering practices.
- **Security Engineering Training:** Teaching the Security Engineering concepts.
    - **Training Types:** Lectures, online tutorials, mentoring.
    - **Number of Personnel:** 4-5 people
    - **Entry Criteria:** Basic understanding of Security Engineering concepts.
    - **Exit Criteria:** Expansive comprehension of Security Engineering concepts.
- **Team Lead Training:** Teaching the team leading best practies, leadership qualities and company principles.
    - **Training Types:** Lectures, mentoring.
    - **Number of Personnel:** 5 people (1 person per team)
    - **Entry Criteria:** Basic understanding of team leading concepts and displaying leadership behavior.
    - **Exit Criteria:** Expansive comprehension of team leading concepts.
- **General Management Training:** Teaching the management best practices, tools, company principles, and leadership qualities.
    - **Training Types:** Lectures, mentoring, consultations.
    - **Number of Personnel:** 5-6 people
    - **Entry Criteria:** Basic understanding of management concepts and displaying leadership behavior.
    - **Exit Criteria:** Expansive comprehension of management concepts.

7.3. Project Work Plans

7.3.1. General

This section of the PMP lays out the blueprint for the project. The work activities, schedule, resource, budget allocation, procurement, and disposal are

included in the following parts of this section. All team members are expected to adhere to the plans explained here for the successful completion of the project.

7.3.2. Work Activities

7.3.2.1. Project Initiation

Duration: 1 week

7.3.2.1.1. Identify Goals and Stakeholders

- Resources: Project team
- Work Products: Project goals and stakeholders are documented.
- Acceptance Criteria: Project goals and stakeholders are approved by stakeholders.

7.3.2.1.2. Create Project Charter

- Resources: Project team, Project manager
- Work Products: Project Charter is created.
- Acceptance Criteria: The project's risks, constraints, scope, budget, milestones, and team contributions are reviewed, approved, and documented by stakeholders.

7.3.2.1.3. Allocate Budget & Resources

- Resources: Project manager, Finance department
- Work Products: Budget Plan, Resource Allocation Plan
- Acceptance Criteria: The necessary budget and resources are documented by stakeholders.

7.3.2.2. Requirements

7.3.2.2.1. Identify Stakeholder & Software Requirements

- Duration: in the 1 week that takes to do the Requirement Analysis Document
- Resources: Project team
- Work Products: Stakeholder and software requirements documented
- Acceptance Criteria: The project's requirements are documented by stakeholders.

7.3.2.2.2. Risk Analysis on Requirements

- Duration: in the 1 week that takes to do the Requirement Analysis Document
- Resources: Project team, Project Manager
- Work Products: Risk assessment of requirements is documented.
- Acceptance Criteria: The project's risk assessment of requirements is documented by stakeholders.

7.3.2.2.3. Requirement Analysis Document

- Duration: 1 week
- Resources: Project team
- Work Products: Requirement Analysis Document
- Acceptance Criteria: The non-functional and functional requirements of software and stakeholders are finalized by stakeholders.

7.3.2.2.4. Project Management Plan (PMP)

- Duration: 1 week
- Resources: Project team
- Work Products: Project Management Plan (PMP)
- Acceptance Criteria: The project's objectives, goals, budget, resources, and time management are approved by stakeholders.

7.3.2.3. Design

7.3.2.3.1. Define System's Design

- Duration: in the 1 week that takes to do System Design Document
- Resources: Project team, system architects
- Work Products: The system's architecture and components are decided.
- Acceptance Criteria: The project's design is decided by stakeholders.

7.3.2.3.2. Risk Assessment on Design Process

- Duration: in the 1 week that it takes to do System Design Document

- Resources: Project team, Project Manager
- Work Products: Risk assessment of design decisions like system architecture and components are documented.
- Acceptance Criteria: The project's risk assessment of system architecture and components is documented by stakeholders.

7.3.2.3.3. System Design Document

- Duration: 1 week
- Resources: Project team, Project Manager, UI/UX Team Leaders
- Work Products: The system Design Document is finalized.
- Acceptance Criteria: Design decisions are documented and approved by stakeholders.

7.3.2.3.4. Mockup Creation

- Duration: 1 week
- Resources: Project team, UI/UX Designers and Team Leaders
- Work Products: The system user interface is created.
- Acceptance Criteria: A mockup that includes the user interface is approved by stakeholders and presented to them.

7.3.2.4. Development

7.3.2.4.1. Implement Initial Codebase

- Duration: ~1 week
- Resources: Project team, Developers
- Work Products: The system' initial codebase and repository is created and documented.
- Acceptance Criteria: Initial Codebase is successfully integrated and approved by stakeholders.

7.3.2.4.2. Implement Functional Prototype

- Duration: ~1 week
- Resources: Project team, Developers
- Work Products: The system's functional prototype is created.
- Acceptance Criteria: Functional Prototype is successfully integrated and approved by stakeholders.

7.3.2.4.3. Risk Assessment of Development Process

- Duration: During the Development Process
- Resources: Project team, Project Manager,
- Work Products: Risk assessment of development is documented.
- Acceptance Criteria: Risk assessment of development is approved by stakeholders.

7.3.2.4.4. Produce Minimum Viable Product (MVP)

- Duration: 1 week
- Resources: Project team, Developers
- Work Products: The system's Minimum Viable Product (MVP) is created.
- Acceptance Criteria: The simplest version of the end product that provides value to the users is assessed and the Minimum Viable Product (MVP) is successfully integrated and approved by stakeholders.

7.3.2.5. Quality Assurance and Testing

7.3.2.5.1. Create Test Plans

- Duration: 1 week
- Resources: Test engineers, Test and QA Leads
- Work Products: Test plans and test cases are created.
- Acceptance Criteria: Test plans and test cases are approved by stakeholders.

7.3.2.5.2. Internal Testing and Quality Check

- Duration: 2 weeks
- Resources: Developers, Test engineers, Test and QA Leads
- Work Products: Alpha version for internal software testing has been created. Its Code Quality Report is delivered.
- Acceptance Criteria: The internal testing version and quality check documentation were approved and reviewed by stakeholders.

7.3.2.5.3. Stakeholders and Small Groups Testing and Quality Check

- Duration: 6 weeks

- Resources: Developers, Test engineers, Test and QA Leads
- Work Products: Beta version for the stakeholders and a small group of users software testing has been created. Its Code Quality Report is delivered.
- Acceptance Criteria: The stakeholders and a small group of users testing the version of the code and quality check documentation were approved and reviewed by the stakeholders.

7.3.2.5.4. Risk Assessment of Quality Assurance and Testing

- Duration: During the Quality Assurance and Testing Process
- Resources: Project team, Project Manager, Test and QA Lead
- Work Products: Risk assessment of quality assurance and testing is documented.
- Acceptance Criteria: Risk assessment of quality assurance and testing is approved by stakeholders.

7.3.2.5.4. User Acceptance Testing (UAT)

- Duration: 2-3 weeks
- Resources: Test and QA Lead, Project team, Project Manager
- Work Products: The user acceptance testing documentation.
- Acceptance Criteria: User acceptance testing is reviewed and approved by the stakeholders. Stakeholders assess whether the application is ready to be deployed.

7.3.2.6. Deployment

7.3.2.6.1. Preparation for Deployment

- Duration: 1 week
- Resources: Project team, Project Manager
- Work Products: Preparation for deployment documentation.
- Acceptance Criteria: Stakeholders approve and review the application’s preparation for deployment.

7.3.2.6.2. Risk Assessment of Final Release

- Duration: During the Deployment Process
- Resources: Project team, Project Manager

- Work Products: Risk assessment of deployment is documented.
- Acceptance Criteria: Risk assessment of deployment is approved by stakeholders.

7.3.2.6.3. Release of the End Product

- Duration: 2 weeks
- Resources: Project team, Project Manager
- Work Products: The end product is released.
- Acceptance Criteria: Stakeholders approve and review the deployment of the end product.

7.3.2.6.4. Create User Manual

- Duration: 1 week
- Resources: Project team, Project Manager
- Work Products: User Manual is released.
- Acceptance Criteria: Stakeholders approve and review the product's user manual.

7.3.2.7. Invoice & Payment

7.3.2.7.1. Invoice and Payment Planning and System Generation

- Duration: ~1 week
- Resources: Project Manager, Payment processing team, Vice president of the finance department, Finance Manager
- Work Products: Invoice and Payment Planning is released. Payment Processing is established.
- Acceptance Criteria: Stakeholders approve and review the product's invoice and payment processing.

7.3.2.7.2. Risk Assessment of Invoice & Payment

- Duration: During the Invoice & Payment Process
- Resources: Project team, Project Manager, vice president of finance, Finance Manager
- Work Products: Risk assessment of invoice & payment is documented.

- Acceptance Criteria: Risk assessment of invoice & payment is approved by stakeholders.

7.3.2.7.3. Final Budget Report

- Duration: ~1 week
- Resources: Project Manager, Project team, Payment processing team, Vice president of the finance department, Finance Manager
- Work Products: Final Budget report is released.
- Acceptance Criteria: Stakeholders approve and review the final budget report.

7.3.2.8. Closure

7.3.2.8.1. Assess Final Project Evaluation

- Duration: less than a week
- Resources: Project Manager, Project team
- Work Products: The project's final evaluation with the outcomes, lessons learned and future plans is done.
- Acceptance Criteria: Stakeholders approve and review the final project evaluation.

7.3.2.8.2. Project Closure Report

- Duration: less than a week
- Resources: Project Manager, Project team
- Work Products: Project Closure Report is delivered and the project's final evaluation of the outcomes, lessons learned, and future plans is documented.
- Acceptance Criteria: Stakeholders approve and review the Project Closure Report.

7.3.3. Schedule Allocation

The schedule allocation describes the schedule of the project using milestones and their start and end dates.

The project schedule is below, T_0 represents the start of the project:

Milestone	**Start Date**	**End Date**

Project Initiation	T_0	T_0 + 1 week
Identify Goals and Stakeholders	T_0	T_0 + 1 day
Create Project Charter	T_0 + 1 day	T_0 + 4 days
Allocate Budget & Resources	T_0 + 4 days	T_0 + 1 week
Requirements	T_0 + 1 week	T_0 + 2-3 weeks
Identify Stakeholder & Software Requirements	T_0 + 1 week	T_0 + 1 week + 1-2 days
Risk Analysis on Requirements	T_0 + 1 week + 1-2 days	T_0 + 1 week + 2-3 days
Requirement Analysis Document	T_0 + 1 week + 2-3 days	T_0 + 2 weeks
Project Management Plan (PMP)	T_0 + 2 weeks	T_0 + 2-3 weeks
Design	T_0 + 2-3 weeks	T_0 + 3-4 weeks
Define System Design	T_0 + 2-3 weeks	T_0 + 2-3 weeks + 2 days
Risk Assessment on Design Process	T_0 + 2-3 weeks + 2 days	T_0 + 2-3 weeks + 3 days
System Design Document	T_0 + 2-3 weeks + 3 days	T_0 + 3 weeks
Mockup Creation	T_0 + 3 weeks	T_0 + 3-4 weeks
Development	T_0 + 3-4 weeks	T_0 + 6 weeks

Implement Initial Codebase	T_0 + 3-4 weeks	T_0 + 3-4 weeks + 6 days
Implement Functional Prototype	T_0 + 3-4 weeks + 6 days	T_0 + 4-5 weeks + 6 days
Risk Assessment of Development Process	T_0 + 4-5 weeks + 6 days	T_0 + 5 weeks
Produce Minimum Viable Product (MVP)	T_0 + 5 weeks	T_0 + 6 weeks
Quality Assurance and Testing	T_0 + 6 weeks	T_0 + 17-18 weeks
Create Test Plans	T_0 + 6 weeks	T_0 + 7 weeks
Internal Testing and Quality Check	T_0 + 7 weeks	T_0 + 9 weeks
Stakeholders and Small Groups Testing and Quality Check	T_0 + 9 weeks	T_0 + 15 weeks
Risk Assessment of Quality Assurance and Testing	T_0 + 15 weeks	T_0 + 15 weeks + 2 days
User Acceptance Testing (UAT)	T_0 + 15 weeks + 2 days	T_0 + 17-18 weeks
Deployment	T_0 + 17-18 weeks	T_0 + 22 weeks
Preparation for Deployment	T_0 + 17-18 weeks	T_0 + 18-19 weeks
Risk Assessment of Final Release	T_0 + 18-19 weeks	T_0 + 18-19 weeks + 2 days
Release of the End Product	T_0 + 18-19 weeks + 2 days	T_0 + 21 weeks
Create User Manual	T_0 + 21 weeks	T_0 + 22 weeks

Invoice & Payment	T_0 + 22 weeks	T_0 + 23-24 weeks
Invoice and Payment Planning and System Generation	T_0 + 22 weeks	T_0 + 22 weeks + 5 days
Risk Assessment of Invoice & Payment	T_0 + 22 weeks + 5 days	T_0 + 22-23 weeks
Final Budget Report	T_0 + 22-23 weeks	T_0 + 23 - 24 weeks
Closure	T_0 + 23-24 weeks	T_0 + 24 weeks
Assess Final Project Evaluation	T_0 + 23-24 weeks	T_0 + 23-24 weeks +3 days
Project Closure Report	T_0 + 23-24 weeks +3 days	T_0 + 24 weeks

7.3.4. Resource Allocation

The resource allocation describes how the resources of the project are allocated.

Work Activity	**Resource Type**	**Quantity**	**Skill Level**
Project Initiation	Project Manager	1	Senior
Project Initiation	Staff Software Engineer	1	Senior
Project Initiation	Staff Security Engineer	1	Senior
Project Initiation	Staff Test Engineer	1	Senior
Project Initiation	Staff ML Engineer	1	Senior
Requirements	Product Manager	1	Senior

Requirements	Software Architect	1	Senior
Design	UI/UX Manager	1	Senior
Design	Lead UI Designer	1	Senior
Design	UX Designer	1	Senior
Design	UX Researcher	1	Senior
Design	Software Architect	2	Junior
Development	Software Engineer	2	Senior
Development	Lead Data Scientist	1	Senior
Development	ML Engineer	1	Senior
Development	ML Engineer	1	Junior
Development	Data Engineer	1	Senior
Development	NLP Engineer	1	Senior
Development	Security Engineer	2	Senior
Development	Security Engineer	2	Junior
Quality Assurance and Testing	Test and QA Lead	1	Senior
Quality Assurance and Testing	QA Engineer	1	Senior
Quality Assurance and Testing	QA Engineer	1	Junior
Quality Assurance	Automation	1	Senior

and Testing	Engineer		
Deployment	Candidate Engineer	3	Junior
Deployment	Marketing Manager	1	Senior
Deployment	Technology Manager	1	Senior
Deployment	Legal Affairs Manager	1	Senior
Invoice & Payment	Finance Manager	1	Senior
Project Closure	Executive Manager - CEO	1	Senior

7.3.5. Budget Allocation

Work Activity	**Resource Type**	**Quantity**	**Skill Level**	**Total Cost ($)**
Project Initiation	Project Manager	1	Senior	51,000
Project Initiation	Staff Software Engineer	1	Senior	54,000
Project Initiation	Staff Security Engineer	1	Senior	60,000
Project Initiation	Staff Test Engineer	1	Senior	42,000
Project Initiation	Staff ML Engineer	1	Senior	57,000

Requirements	Product Manager	1	Senior	48,000
Requirements	Software Architect	1	Senior	60,000
Design	UI/UX Manager	1	Senior	45,000
Design	Lead UI Designer	1	Senior	45,000
Design	UX Designer	1	Senior	39,000
Design	UX Researcher	1	Senior	39,000
Design	Software Architect	2	Junior	108,000
Development	Software Engineer	2	Senior	96,000
Development	Lead Data Scientist	1	Senior	60,000
Development	ML Engineer	1	Senior	57,000
Development	ML Engineer	1	Junior	48,000
Development	Data Engineer	1	Senior	57,000
Development	NLP Engineer	1	Senior	57,000
Development	Security Engineer	2	Senior	114,000
Development	Security	2	Junior	96,000

	Engineer			
Quality Assurance and Testing	Test and QA Lead	1	Senior	48,000
Quality Assurance and Testing	QA Engineer	1	Senior	45,000
Quality Assurance and Testing	QA Engineer	1	Junior	42,000
Quality Assurance and Testing	Automation Engineer	1	Senior	51,000
Deployment	Candidate Engineer	3	Junior	18,000
Deployment	Marketing Manager	1	Senior	48,000
Deployment	Technology Manager	1	Senior	48,000
Deployment	Legal Affairs Manager	1	Senior	48,000
Invoice & Payment	Finance Manager	1	Senior	48,000
Project Closure	Executive Manager -	1	Senior	120,000

	CEO			

7.3.6. Procurement

T_0 represents the start of the project:

Goods & Services	Type of Contract	Person Responsible	Source of Request	Deadline	Lead Time
Development Hardware (Laptops, Monitors, etc.)	Purchase	Technology Manager	Fixed Supplier	T_0 + 1 week	1 week
Development IDEs, environments, licenses	Subscription	Technology Manager	Fixed Supplier	T_0 + 2-3 weeks	2-3 weeks
Management Tools	Purchase	Project Manager	Fixed Supplier	T_0 + 2-3 weeks	2-3 weeks
AWS RDS (Relational Database Service)	Subscription	Technology Manager	Fixed Supplier	T_0 + 4 weeks	4 weeks
AWS EC2 (Elastic Compute Cloud)	Subscription	Technology Manager	Fixed Supplier	T_0 + 4 weeks	4 weeks
OpenAI API	Subscription	Technology Manager	Fixed Supplier	T_0 + 4 weeks	4 weeks

7.3.7. Disposal

The resources to be demolished, disposed or decommissioned are:

- OpenAI API subscription should be ended by Technology Manager.
- AWS EC2 (Elastic Compute Cloud) subscription should be ended by Technology Manager.

- AWS RDS (Relational Database Service) subscription should be ended by Technology Manager.
- Management Tools should be decommissioned by Project Manager.
- Development Hardware (Laptops, Monitors, etc.) should be decommissioned by Technology Manager.
- Development IDEs, environments, licenses should be unsubscribed by Technology Manager.

8. Project assessment and control

8.1. General

The following procedures detail how to assess and oversee various aspects of a project, including product specifications, project scope, timeframe, budget, resource allocation, procurement of subcontracted services, and the quality of work processes and deliverables. All control activities will adhere to the organization's established standards, policies, and procedures, as well as any contractual agreements related to project control. These control measures encompass managing requirements, regulating changes to project scope, monitoring and adjusting schedules, overseeing budget allocations, ensuring quality assurance, managing subcontractor relationships, and concluding the project effectively.

8.2. Requirements Management

Similar to many other software systems, CODED. anticipates future changes, thus necessitating configuration management. Requests for modifications from users and developers will be systematically tracked, with careful consideration given to the potential impacts of these changes before adoption. Employing an agile approach inC development will facilitate the incorporation of modifications into the requirements [1].

The initial phase of configuration management involves identifying key configuration items (CIs) pertinent to requirements, such as user stories, features, or epics. Each CI will be assigned a unique identifier and stored in a centralized repository or tool for organization. Subsequently, a configuration plan will be maintained, outlining a strategy for handling change requests. For CODED., the team will assess the priority of changes relative to planned features

for upcoming increments. Prototyping and modeling will aid in evaluating the impact of proposed changes, providing insights into system capabilities during implementation. Decision-making regarding change adoption will consider factors such as risk assessment, benefits, user impact, costs, and release cycles [1]. Change requests will be approved or rejected based on these criteria, with appropriate modifications made to the software and documentation if approved.

Configuration status accounting entails monitoring and reporting on configuration items throughout the project. A traceability matrix will ensure alignment between project requirements and deliverables, with regular updates reflecting evolving needs. User feedback from prototypes will inform requirement refinement. Finally, configuration auditing verifies adherence to configuration management processes, including a review of the requirements document to ensure alignment.

8.3. Scope Change Control

Utilizing configuration management processes, CODED. will monitor activities that fall outside the project's defined scope. Initially, key configuration items (CIs) related to scope, such as the scope statement, scope baseline, and Work Breakdown Structure (WBS), will be identified. Subsequently, a configuration plan will outline procedures for handling scope change requests. When a change beyond the project's current scope is proposed, the project team and stakeholders will collaboratively evaluate its impact on various criteria, including scope, schedule, cost, feasibility, alignment with project objectives, risks, and stakeholder implications. These criteria will guide the decision-making process regarding the adoption of the change. Approved scope change requests will prompt modifications to the software and associated documentation, such as the scope baseline and WBS, aligning with the configuration management process.

The next step, configuration status accounting, involves updating corresponding documentation to track configuration items. This ensures alignment with project objectives and scope. Configuration auditing ensures adherence to configuration management processes, verifying that documentation, such as the WBS, scope baseline, and scope statement, accurately reflect the project's objectives and scope, and that all configuration management steps are being followed correctly.

8.4. Schedule Control

A baseline schedule outlining the estimated start and finish times for each activity, will be established to serve as a reference point for monitoring and managing schedule progression. Major and minor project milestones will be identified to track development progress and assess achievements. Objective criteria will be utilized to evaluate the scope and quality of completed work at each milestone, ensuring project adherence to schedule and milestone deadlines.

Earned value measurement (EVM) techniques will be employed to compare planned and completed work, with key metrics such as Schedule Variance (SV) and Schedule Performance Index (SPI) calculated to assess schedule performance and identify any deviations. Schedule compression methods, such as fast-tracking or crashing, will be utilized if necessary to mitigate schedule delays. Earned value analysis will be used to demonstrate the schedule plan and track progress of completed work.

Various earned value measures, including SV and SPI, will provide insights into project schedule performance, guiding decision-making and corrective actions as needed. Control mechanisms, including a well-defined Work Breakdown Structure (WBS), Change Control Process, and regular Progress Reports, will be employed to ensure effective schedule management. The WBS will break down project deliverables into manageable tasks, aiding in scope definition and scheduling accuracy. The Change Control Process will assess scope changes for alignment with project objectives, while Progress Reports will provide stakeholders with updates on project status and highlight any schedule deviations and scope-related risks.

In the event of schedule deviations, corrective actions such as revising the project schedule, reallocating resources, adjusting task priorities, or rearranging activities will be implemented promptly to maintain alignment with project scope and timeline expectations. The primary objective is to address schedule alterations promptly and ensure alignment with project development goals.

8.5. Budget Control

To ensure a comprehensive understanding of the project's financial status, cost measurement will encompass all relevant direct and indirect expenses and utilized resources throughout project implementation. Comparisons between planned and budgeted costs will be

conducted, with any disparities examined to identify underlying reasons. This analysis will enable the project team to take corrective action to align the project's financial performance with planned expenditures.

Prompt corrective action will be taken if actual costs exceed budgeted costs or if deviations potentially impact the project's financial performance. Investigations into the causes of these discrepancies will inform the development of corresponding strategies for their elimination, with documented reports forwarded to the project team and stakeholders.The budget control plan will determine cost reporting intervals, considering project complexity, duration, and the need for timely financial information. Reporting frequency will balance the provision of up-to-date information with minimizing administrative burden.

Appropriate methods and tools, such as budget estimation software, project management platforms like Jira and Asana, financial management systems, and collaboration tools like Trello and Slack, will be utilized for budget management and cost calculation. Microsoft Excel will also be employed to create custom budget templates and track expenses.

The budget plan will incorporate frequent milestones evaluated using clear indicators to measure progress and work quality. These milestones will assess the scope and quality of completed work items across project phases, utilizing objective indicators such as predetermined metrics or performance criteria.

Earned value analysis will be employed to demonstrate the budget and schedule plan, track schedule progress, and calculate the cost of completed work. Derived from actual cost (AC) and earned value (EV), the earned value provides insights into cost management through measures such as Cost Variance (CV) and Cost Performance Index (CPI), indicating project performance relative to the budget. Positive CV and CPI values signify favorable progress, guiding informed decision-making and corrective actions as needed.

8.6. Quality Assurance

Quality objectives have been set for the CODED. project to ensure that work processes and outcomes meet predefined standards and objectives. These objectives cover various facets, including adherence to best practices and quality benchmarks, code coverage, customer satisfaction, system reliability, error reduction, and scalability. These objectives are as follows:

- Ensure all software components are developed, implemented, and maintained in accordance with best practices and quality standards.
- Attain a minimum code coverage of 90% for all developed software modules.
- Achieve a customer satisfaction rating of 4 out of 5 or higher based on user feedback.
- Maintain a system uptime of at least 99.9% and reduce critical errors to less than 1% per month.
- Design a scalable system capable of accommodating a minimum of 200 concurrent users.

To evaluate the quality of work processes and outcomes, specific quality metrics have been selected as key indicators. These metrics include compliance with best practices and quality standards, code coverage, defect density, customer satisfaction, system reliability, error reduction, and scalability.

To uphold and control quality, several mechanisms have been implemented. Regular assessments ensure quality assurance of work processes, allowing for identification of improvement areas and implementation of corrective measures. Verification and validation processes ensure that work products and outcomes meet required standards, with regular reviews and inspections ensuring accuracy, correctness, and compliance with quality benchmarks.

Joint reviews involving project team members, stakeholders, and subject matter experts are conducted to collaboratively evaluate work outcomes, incorporating valuable feedback and suggestions to enhance overall standards. Periodic audits are also carried out to assess compliance with quality standards, processes, and procedures, with recommendations and corrective actions implemented as needed.

The quality assurance plan includes provisions for monitoring and evaluating the performance of project vendors. Clear guidance on quality expectations is provided to vendors, and their compliance with quality standards is regularly assessed. Ongoing communication and coordination with vendors ensure consistent quality throughout the project.

Detailed records are maintained for all quality assurance activities, encompassing measurement outcomes, quality control procedures, and any adjustments made. Reliable reports are generated to provide project stakeholders with transparent insights into the project's current quality performance, supporting informed decision-making.

By implementing these quality control mechanisms and maintaining an effective quality assurance plan, the CODED. project ensures that work processes and outcomes consistently align with defined quality objectives, ultimately promoting customer satisfaction and project success.

8.7. Subcontractor Management

The project will not involve subcontractors, meaning that all work items and services will be handled internally by members of the project team. This ensures direct oversight and accountability over project deliverables.

The absence of subcontractors simplifies the management structure as there is no need for separate management plans or coordination with external entities. The project team will take full responsibility for requirements management, monitoring technical progress, controlling schedule and budget, defining product acceptance criteria, ensuring quality assurance and measurement, and implementing risk management methods.

8.8. Project Closeout

This section, the PMP's project closeout, includes all of the necessary plans for assuring a seamless project closeout. This closeout plan's main elements include project materials archiving standards and the creation of a final report that includes experiences acquired and a breakdown of project goals met.

The project charter, the project management plan, a variety of project papers, acceptable deliverables, business documents, agreements, procurement documentation, and the source code are all included in the plan's provisions for protecting project resources. Keeping these resources up to date can guarantee the continuity and well-informed decision-making by allowing future audits, knowledge transfer, and initiatives to benefit from the whole project record's insights and information.

Participants' opinions and views will be gathered through post-project recaps. Documentation of their project-related experiences, difficulties, and experiences gained will be made. Future project management procedures, technological applications, teamwork initiatives, and stakeholder engagement tactics will all benefit from this input.

9. Product Delivery

A phased delivery method will be used to deliver the project in a well-organized and structured way. Through its development process, the development team will be in constant touch with the quality assurance specialists to prevent defects and issues. When a phase is completed and is agreed to be free of errors and ready to be published, the deliverables are released to the customer.

Information Flow:

- Internal to the Project: A robust internal communication framework will be used for a smooth information flow among team members. Regular meetings, progress reports, and communication channels will allow aligned team members in terms of objectives and the project schedule.
- External Organizations: With stakeholders, clients, and vendors, a clear communication is aimed. Regular updates and progress reports will be delivered to these external organizations to keep them engaged through the whole project development process.

Packaging and Physical Delivery Plans:

- Packaging: The deliverables will be packaged in an alignment with the industry standards and the project objectives.
- Physical Delivery: Although the project does not have physical deliverables, any will be delivered according to the delivery logistics when needed.

Customer Documentation:

- Manuals: Detailed operation manuals will be delivered, including functionalities and the operations. Maintenance manuals will also be provided to the customer, including servicing of the products. Finally, training materials and manuals will be provided to guide the customer on how to use the project.

10. Supporting Processes

10.1. General

This section of the Project Management Plan (PMP) encompasses plans for the various supporting processes that extend throughout the project's duration. These plans cover a range of aspects, including project supervision and work environment, decision management, risk

management, configuration management, information management, quality assurance, and measurement.

10.2. Project Supervision and Work Environment

An environment is created by the project manager so that every team member work collaboratively, sharing a common goal. Daily meetinga are held, and daily tasks, advises, and tasks are given to the team. When tasks are done well, the success is appreciated while any effort is recognized. Any potential errors are safely shared, and are addressed as soon as possible. Along with the daily meetings, regular meetings are also being held. The project manager follows the performance targets, encourages different opinions, and resolves arising problems within the team and the project. There is no physical working environment, instead team members work remotely. Cloud-based tools and seamless communication platforms are used among team members to eliminate communication problems. The communication is vital for both synchronous and asynchronous dialogues, so platforms are chosen based on this.

10.3. Decision Management

Specific decision categories are developed based on the conditions and demands of the CODED. initiative.

Decision Categories:

Strategic Decisions:

These are pivotal choices that significantly impact the project's overall direction, including defining and potentially revising the project's scope, budget, timelines, or key performance indicators (KPIs). These decisions require input from the Project Manager and other top-level stakeholders.

Technical Decisions:

Technical decisions encompass choices related to software architecture, programming language selection, code management strategies, API integrations, security measures, QA methodologies, and UX/UI design principles. Developers propose these decisions, which are then approved by the Project Manager.

Operational Decisions:

Operational decisions pertain to the day-to-day functioning of the project, such as task assignments, resource allocation, meeting schedules, internal communication protocols, and addressing minor technical issues. Typically, these decisions are handled at the team level, although the Project Manager may intervene in cases of disagreements or resource constraints.

Risk Management Decisions:

This category involves decisions concerning the identification, assessment, prioritization, and mitigation of potential risks. These decisions require a collaborative approach involving input from all team leaders and the Project Manager.

Decision-Making Approaches:

Strategic Decisions:

The decision-making process commences with Steering Committee meetings, where thorough discussions are held to assess the potential impacts of each option. This ensures that all strategic decisions are reached through consensus, are well-documented, and ultimately receive final approval from the Project Manager.

Technical Decisions:

Technical decisions involve consultation with technical leads and the Project Manager, who carefully evaluate their suitability for project requirements and potential impacts on scope or timeline. If a decision carries significant consequences, it is escalated to the Steering Committee for further deliberation.

Operational Decisions:

Operational decisions are primarily made by team members and leaders. If a decision alters the project plan or team structure, the Project Manager is informed to maintain alignment with the project's overarching strategy.

Risk management decisions are reached collectively during dedicated meetings focused on risk assessment. These meetings involve the Project Manager, all team leads, and possibly the Steering Committee, ensuring comprehensive consideration of all perspectives in addressing risks.

Method of Involving All Relevant Parties

Strategic Decisions: Involves the Project Manager, Stakeholders, Steering Committee, Marketing Specialists.

Technical Decisions: Developers, ML Engineers, Designers and the Project Manager.

Operational Decisions: Team members and Team leads, and the Project Manager when needed.

Risk Management Decisions: Project Manager, Stakeholders, team leads, and potentially the Steering Committee during high-risk situations.

Success Criteria:

This structured decision-making process is designed to guarantee that the project remains aligned with defined objectives and requirements, fosters transparent and efficient communication, and effectively mitigates risks. The criteria for success are outlined as follows:

Timeliness: Decisions must be made within specified timelines to prevent project delays.

Effectiveness: Each decision should advance the desired outcome without compromising the overall project goals.

Stakeholder Satisfaction: The decision-making process should promote transparency and engagement, leading to increased satisfaction among stakeholders.

Tracking and Evaluation:

A decision log will be maintained to record all decisions made throughout the project's duration. This log will serve to document decision outcomes and facilitate continuous improvement in the decision-making process. The responsibility for maintaining and updating this log lies with the Project Manager.

Documentation and Reporting:

Every decision, particularly those pertaining to strategy and technology, will be comprehensively documented. This report will include details such as the nature of the decision, individuals involved, rationale, consequences, and final resolution. Distributing this document to relevant stakeholders will promote transparency and ensure all team members are kept informed about the project's status.

10.4. Risk Management

For our project to succeed, it is critical to manage risks well in order to reduce any potential roadblocks that might obstruct our development. We have a thorough plan in place to address every potential obstacle, including those pertaining to staffing and technology.

The project manager will supervise the full risk management procedure and guarantee that all work is finished on time. Within the purview of their own areas of expertise, each team member will participate in the identification and monitoring of risks.

1. Risk Identification

The process of identifying risks in our project is ongoing and iterative. It is the duty of the team to identify possible risks in respective domains.

We use a variety of risk assessment approaches and procedures, such as brainstorming, the Delphi Method, Root Cause Analysis, and SWOT Analysis. We also take into account outside variables including modifications to legal or regulatory frameworks, improvements in technology, and changes in market dynamics.

2. Risk Analysis & Prioritization

After identifying risks, we conduct both qualitative and quantitative risk analyses. In qualitative analysis, risks are assessed on a scale of 1 to 10 based on their likelihood and impact. These ratings are then used to calculate a Risk Priority Number (RPN), which aids in prioritizing risks. Quantitative analysis, on the other hand, involves a more detailed examination of the potential cost or schedule impact of risks using techniques like Decision Tree analysis.

3. Contingency Planning

Contingency planning ensures the presence of a backup strategy, referred to as 'strategy B,' for each high-priority risk. These plans are comprehensive and detailed, encompassing early warning indicators of the risk event, strategies to mitigate its impact, necessary resources, assigned roles and responsibilities, and a communication strategy. For instance, if a key team

member is on the brink of departure, the contingency plan would involve identifying potential successors, providing cross-training to other team members, and ensuring accurate documentation of the departing individual's work.

4. Risk Tracking & Evaluation

We employ a risk register to monitor all identified risks and their current status. This register is continuously updated and accessible to all stakeholders. It includes details such as risk description, owner, likelihood, impact, Risk Priority Number (RPN), mitigation plan, contingency plan, and current status. Additionally, we conduct weekly risk review meetings to assess any newly identified risks, the status of existing risks, and the effectiveness of mitigation and contingency measures.

5. Risk Communication

Effective risk communication is vital to ensure that everyone is informed about potential issues and equipped with strategies to address them. Regular risk updates are provided to all key stakeholders, including the client, suppliers, and internal teams, through emails, meetings, and dashboards. Moreover, escalation processes are implemented to ensure that high-priority issues are promptly addressed by the appropriate individuals.

6. Risk Factors

- **Acquirer-Supplier Relationship Risks:** Our dependence on suppliers for components like software libraries and cloud services increases the risk of delays, quality issues, and supply disruptions. To mitigate these risks, we maintain frequent communication with suppliers, manage partnerships diligently, and establish contingency plans.
- **Contractual Risks:** We mitigate this risk by ensuring that all contracts are clear, transparent, and legally sound.
- **Technological Risks:** We assess and plan for risks associated with technology choices, software architecture, integration challenges, and potential changes in technological standards.
- **Development & Target Environment Risks:** We identify and control risks present in both the target environment (such as network issues and user device compatibility issues) and our development environment (such as system failures and software compatibility issues).

- **Schedule & Budget Risks:** We continuously evaluate risks that could lead to schedule or budget overruns.
- **Acquirer Acceptance Risks:** We maintain regular communication with the acquirer during the development phase to ensure their needs are addressed and mitigate this risk. Agile development techniques allow us to receive ongoing feedback from the acquirers, enabling prompt resolution of issues and risk mitigation.

10.5. Configuration Management

Configuration Identification:

The CODED. project will employ a systematic approach to accurately define the project scope and identify all pertinent configuration items. These configuration elements may encompass software modules, hardware components, documentation, and various other artifacts. For instance, separate configuration items such as the recommendation engine, user profile management, and menu generation will be identified as distinct software modules. By identifying these components, the project team gains a comprehensive understanding of the project scope and the interdependencies between different elements.

Configuration Control:

Every proposed modification will undergo a formal change control process. Change requests will be evaluated considering the project's objectives, schedule, and budget. The Change Control Board (CCB), composed of key stakeholders and subject matter experts, will review each change request. For instance, if a request entails adding an extra feature to the client ordering system. The CCB will scrutinize the change request, assess its implications, and decide whether to proceed with its implementation. The configuration management plan will specify the necessary documentation for change requests, including change forms, impact analysis reports, and updated configuration baselines.

Status accounting:

The project will manage an accounting system to track the status of configuration items and their associated attributes. Regular status reports will record details such as the version, status, and location of each item. These reports enable the project team to monitor changes, identify discrepancies, and furnish precise information on the configuration status of the project.

Evaluation and Release Management:

This configuration management plan outlines procedures for evaluating the effectiveness, usability, and quality of configuration items to ensure alignment with project requirements and objectives. Various evaluation techniques, such as testing, auditing, and inspections, will be employed.

Furthermore, the plan details procedures for packaging and deploying configured items.. Only authorized and rigorously tested configurations will be released. Version control methods will be established to ensure reproducibility and traceability of previous setups.

Automated Configuration Management Tools:

Automated configuration management tools streamline communication among project team members, simplify version control, facilitate change management, and offer a central repository for storing configuration components. For precise version tracking and documentation, alterations to software code and configuration files will be monitored using a version control system such as Git. Furthermore, collaboration platforms like Jira or Notion can be employed by the project team to coordinate change requests, allocate tasks, and uphold transparency in monitoring project progress.

10.6. Information Management

10.6.1. General

This section of the report explains how the information will be managed in the application. The information management, representation forms, responsibility distribution, information storage and availability sections are described in detail. The plans for information security are also included in this section.

10.6.2. Project Information Management

The purpose of this section is to outline the plan for managing information throughout the project lifecycle. There are various types of information that will be

managed all which can either be classified as deliverable or non-deliverable work products.

Non-Deliverables: The types of products that are not delivered to the stakeholders and the users but are required for the project's completion.

- **Requirements Specifications:** The functional and non-functional requirements needed by the stakeholders that serve as a foundation to the application's use cases and features.
- **Design Documents:** All of the diagrams and models describing the system architecture.
- **Test Plans:** Documents defining the tests that will be conducted to assure project's succes.
- **Meeting Logs:** Notes and summaries of the meetings both among the team and with the stakeholders.
- **Review Reports:** Documentation for the project reviews with information on updates, changes, bug findings and recommendations from the stakeholders.

Deliverables: The products that are delivered to the end users and the stakeholders.

- **The Source Code and Versions:** The software codebase, including the previous application versions such as the functional prototype, the Alpha version and the Beta version.
- **The Web Application:** The final product which will be used by the end users.
- **The User Manual:** The document describing the workflow of the application.
- **Test Results:** Documentation of the test cases and their respective results.
- **Customer Satisfaction Surveys:** Feedback document collected from the end users and the stakeholders regarding the advantages and the disadvantages of the product.
- **Configuration Library:** Document for managing application versions and configurations.

- **Principles of Operation:** Guideline for outlining the functionality and operation of the system.

The Information Management Plan: The information management plan includes the list of information items, controlling templates/standards, responsible parties specification, review and baseline dates, and distribution list.

- **Information Items:**
    - **Project Charter:** Inital document specifying the project scope and objectives.
    - **Requirements Specifications:** The detailed documentation of stakeholder requirements.
    - **Design Documents:** The detailed document describing the system architecture such as mockups and diagrams.
    - **Source Code:** The source code for the web application.
    - **Test Plans and Test Reports:** The plans and results for the conducted tests including the inputs and outputs for each test.
    - **The User Manual:** The guidelines for using the application prepared for the end users.
    - **Meeting Logs:** The notes and summaries of the meetings held with the stakeholders and within the team.
    - **Project Status Reports:** The reports stating the current status of the along with accomplished and upcoming milestones and planned changes.
    - **Post-Implementation Review Report:** The evaluation of the application including the lessons learned and future plans.
- **Controlling Templates/Standards:**
    - The source code will comply with the best coding practices for Spring Boot Java, React JS, and Python.
    - The user manual will comply with the IBM's User Guide standards.
    - The meeting minutes will comply with the IBM's Meeting Minutes guides.

    - For any project management related documents such as the Project Charter, PMBOK conventions will be followed.
- **Responsible Parties:**
    - **Project Manager:** Directly responsible for the Project Charter. Will be in charge of ensuring each information item is delivered within the time constraints.
    - **Product Owner:** Responsible for the Requirements Specifications, Project Status Reports, and Post-Implementation Review Reports.
    - **Security Engineer:** Responsible for the security of the Source Code, Test Plans, and Test Reports.
    - **Software Engineer:** Responsible for the Source Code, the Requirements Specification, the Design Documents, the User Manual, the Project Status Report, and the Post-Implementation Review Report.
    - **Test and QA Engineer:** Responsible for the Test Plans, Test Reports, the Project Status Reports, and the
    - **ML Engineer:** Responsible for the Requirements Specifications, the Design Documents and the Source Code.
    - **UI/UX Designer:** Responsible for the Source Code, the Requirements Specifications, the Design Documents, and the User Manual.
    - **Development Team:** The team consists of all the developers. Their specific responsibilities have been explained above. However, it can be assumed that they are all responsible for at least some part of the information items.
- **Baseline and Review Dates:**
    - There is a specified baseline and review date for each of the information items. The baseline date represents the initial submission of the item, whereas the review date represents the submission for the updated version based on customer, stakeholder or management feedback.

- **Distribution List:**
    - **The Team:** The internal company. All documents are available for the team members, especially if they need them for the development process.
    - **Higher Management Team:** The C-Level managers. All documents are available for the high management. Documents such as the Project Charter, Project Status Report and Post-Implementation Review will be directly delivered to the management.
    - **External Stakeholders:** The documents such as the Project Charter, Requirements Specification report, User Manual, project Status Report and Post-Implementation Review will be available for the external stakeholder as long as they do not contain company-sensitive information. The legal affairs department can reduce some or all of these documents if needed.

10.6.3. Communication and Publicity

This section of the PMP for the CODED. project outlines methods and tactics to effectively disseminate information to all stakeholders involved in the development and utilization of the web application.

Stakeholders:

- Developers: The group accountable for coding web application
- Users: Individuals who will use CODED.
- Management Team: Those sponsoring, managing, and making decisions regarding the project.
- Marketing Team: Those tasked with advertising and raising awareness about web applications.

The stakeholders will receive updates on the following aspects:

- Progress Updates: Regular reports on development progress
- User Feedback: Input from users on web application usability and recommendations.
- Marketing Updates: Information on marketing strategies and promotions.

Different communication forms will be used based on stakeholders and content:

- Progress Reports: Detailed updates for developers and management teams.
- Online Feedback: Surveys for user opinions.
- Marketing Materials: Promotional content for the marketing team.

Responsibilities for communication:

- Development Team Lead: Updates management on technical progress.
- Customer Support Representative: Gathers user feedback for the developers.
- Marketing Manager: Develops marketing plans and informs stakeholders.

Communication methods and technologies:

- Project management software for tracking progress.
- Email and instant messaging for quick updates.
- Online surveys for gathering feedback.

Frequency of communication depends on stakeholder needs:

- Weekly Development Team Meetings.
- Monthly Stakeholder Reports.
- Continuous User Feedback Channels.

Escalation process:

- Development Team Lead resolves team issues.
- Project Manager handles larger concerns.

Documents may include:

- Progress Reports distributed to relevant teams.
- User Surveys available online.

Regular evaluation and revision of the communication plan will ensure effectiveness. Stakeholders will be updated via email or meetings. The Marketing Team will handle broader marketing efforts outside the project's scope, integrating their strategy into the overall communication plan.

10.7. Quality Assurance

Quality Assurance (QA) is vital for any project to ensure a high-quality product. To make sure of a project that is fully aligned with the objectives and the requirements, QA is done on

various processes throughout the development phase. The QA plan consists of analysis, reviews, audits, assessments, and inspections. From designing phase to delivering phase, this plan is followed, testing the quality of each process. Within the project, there are specialized engineers for this task, QA engineers that play a role in most of the project processes. However, ensuring the quality assurance is not limited to these engineers, instead, every team member makes sure of producing high-quality, requirements fulfilled deliverables.

10.8. Measurement

When evaluating the project, several metrics are defined as below. These metrics ensure a product delivery that meets the objectives and requirements of the stakeholders and customers.

User Engagement Metrics: User volume, usage frequency, average session duration.

User Satisfaction: Chatbot answer satisfaction, grade satisfaction, user-friendly interface, intuitive interface.

Engagement Duration: The amount of time users actively use the application.

Correctness of the Chatbot: Answers of the chatbot aligned with the instructor expectations, correctness of the answers.

User Surveys: There are several types of users and by the time the project is ready to be used by them, ethics-approved surveys will be distributed, measuring the user experience.

Social Media Analytics: Tracking the amount of people the product advertisements reach.

10.9. Reviews and Audits

The PMP's Reviews and Audits section outlines the CODED. project's timeline, resources, methods, and procedures for carrying out various project reviews and audits. It includes the subsequent components:

Management Progress Reviews:

- Schedule: Every month on the last Thursday, there will be reviews.
- Resources: Senior management team; project managers.
- Methods: Examining overall progress, project milestones, and budget status.

- Protocols: Holding management review meetings and using a template for consistent progress reports.

Developer Peer Reviews:

- Schedule: Thursdays are used for the weekly peer code reviews.
- Resources: Developers are the resources.
- Methods: Finding areas for improvement, checking code for errors, and adhering to coding standards.
- Protocols: Using a checklist for peer review, holding code review meetings, and recording results.

Quality Assurance Audits:

- Schedule: Monthly audits carried out annually.
- Resources: Auditors and the QA team.
- Techniques: Examining and rating project deliverables and procedures in a methodical manner.
- Methods: Making use of a sample strategy, recording conclusions and suggestions, and using an audit checklist.

Acquirer Conducted Reviews and Audits:

- Schedule: Ad hoc audits and reviews carried out at the acquirer of the project's request.
- Resources: Members of the project team and acquirer representatives.
- Techniques: Independent evaluations, on-site inspections, and functional and compliance checks.
- Procedures: Working with the acquirer to create a personalized review and audit plan based on particular needs.

External Organizations:

- Data Privacy Regulatory Body: Guarantees adherence to data privacy guidelines and protects user data.

The Reviews and Audits section ensures effective conduct of project reviews and audits, providing valuable insights, identifying improvement areas, and ensuring regulatory compliance. This is achieved by specifying the schedule, allocating appropriate resources, defining methods and procedures, and identifying external regulatory agencies.

10.10. Verification and Validation

This section of the PMP outlines the verification and validation plan for the project, encompassing the scope, tools, techniques, and responsibilities associated with these activities. It outlines the organizational connections and degrees of autonomy between tasks related to verification and validation and development activities. Techniques including traceability, milestone reviews, progress reviews, peer reviews, prototyping, simulation, and modeling must be identified as part of verification planning. Planning for validation entails defining methods including testing, demonstrating, analyzing, and inspecting. The strategy also specifies the automated technologies that will be used for validation and verification.

Verification Planning:

This plan describes how to ensure that the project's work products are verified and that all requirements are met. Among the methods are a few of them:

Traceability: Creating a link between requirements, design, implementation, and test cases by using tools such as ReqTrack or JIRA to make sure all requirements are met and implemented appropriately.

Milestone Reviews: Formal assessments carried out to assess the quality, completeness, and progress of work items at significant milestones, such as the completion of major deliverables. Microsoft Teams and Zoom are two examples of tools that can be used for virtual review meetings.

Progress Reviews: Monitoring the project's advancement and benchmarks on a regular basis to make sure it stays on track and spot any problems or deviances. Asana and Trello are two tools that can help in monitoring and controlling the project's progress.

Peer Reviews: Using online resources like GitHub or Bitbucket, peer code reviews are a great way to find bugs, verify that the code complies with standards, and point out areas that could use improvement.

Prototyping: the process of creating interactive prototypes with software such as Adobe XD or InVision to evaluate and improve functionality and design before a large-scale implementation.

Simulation and modeling: Simulink and other simulation and modeling applications are used to assess system behavior.

Validation Planning:

This plan describes how to validate the system to make sure it works as planned and satisfies user needs. Methods consist of:

Examination: Extensive testing is done to confirm that the system functions as intended, fulfills requirements, and satisfies expectations. This covers system testing, user acceptability testing, integration testing using tools like Cypress or Selenium, and unit testing using frameworks like JUnit or pytest.

Demonstration: Showing stakeholders the system's usability, functionality, and adherence to specifications. Platforms like Zoom or Microsoft Teams can be used for virtual demonstrations, and screen capture programs like Camtasia or OBS Studio can be used for documentation.

Analysis: Applying analytical methods to evaluate the dependability, performance, and behavior of a system. Using programs like MATLAB or R, this could entail statistical analysis, mathematical modeling, or simulation.

Inspection: Analyzing the system to find flaws, defects, or contradictions in the documentation, code, or design. Tools such as SonarQube or ESLint can be used for code inspection and static analysis.

Degrees of separation between development and activities related to validation and verification: The development team follows the specifications when designing and building the system. Next, the verification team checks their work to make sure it complies with guidelines and standards. The validation team evaluates the system to ensure that users are satisfied and that it is operating properly. The verification and validation teams function independently of the development team, even though they work closely together, with the validation team offering feedback and the development team handling any problems. They do, however, share findings to make sure everything is in working order.

11. Additional Plans

In order to ensure compliance with all product criteria and contractual obligations, it is imperative to develop and implement comprehensive additional plans. These supplementary plans are essential for addressing the specific and unique aspects of the project, including safety, privacy, security, integration, and other relevant factors.

Safety Plan:

During the development phase, coding security will be accomplished by strict adherence to standard secure coding principles. To make sure safety regulations are satisfied, robust risk assessment methods will be established, and regular audits will be conducted. Continuous safety measures will be done to ensure a prompt and effective response in the event of a security breach.

Privacy and Security Plans:

Data will be encrypted not only during storage but also during transmission. Data anonymization techniques will be used where necessary to protect user identities. Biometric access controls will be introduced as needed, and compliance with privacy regulations such as KVKK and GDPR will be mandatory. To assess the resilience of our security protocols, regular penetration testing will be conducted.

Software and Equipment Plan:

This plan will be undertaken for all necessary hardware and software, including servers, workstations, testing equipment, and network infrastructure. Multiple redundant systems will be deployed to make sure a highly available and fault-tolerant environment is satisfied.

Integration Plans:

We will conduct these plans thoroughly and evaluate all systems that our application needs to interface with. Detailed API documentation, encompassing all endpoints, request/response formats, error codes, and illustrative examples, will be meticulously crafted. Extensive testing will be applied on all APIs to ensure straightforward and dependable integration.

Product Support Plans:

We will establish multiple channels for product assistance, including a dedicated phone line, email support, and a live chat option. Additionally, a comprehensive FAQ section will be developed, along with a community forum where users can seek assistance from fellow members.

Product Maintenance Plans:

A structured framework for corrective, proactive, adaptive, and continuous improvement activities will be established. Real-time performance monitoring that will be triggered by anomalies will be implemented. To uphold customer satisfaction, feedback from users will be regularly gathered and analyzed.

12. End Matter

Agile: A project management approach known for its flexibility, self sufficient teams and team collaboration.

Scum: An Agile project management approach that uses Sprints and an iterative development approach that utilizes Scrum methodology.

UI/UX: User Interface/User Experience

ML: Machine Learning

Sprint: Scrum Approach's short periods that the different activities are accomplished.

Project Charter: The document that includes the project's outline of the project scope, objectives, stakeholders, and requirements.

Project Management Plan (PMP): Document that defines the project scope, objectives, goals, budget, timeline, and resource management in detail.

Artificial Intelligence (AI): The computer systems that can do complex tasks that humans can do.

Change Control Board (CCB): The group in a project that is responsible for making decisions when a change happens.

Configuration Items (CI): Individuals items or components in a system that are considered as a single entity for configuration management.

PMBOK: Project Management Body of Knowledge

Git: Distributed version control system

General Data Protection Regulation (GDPR): Regulations that protect the privacy of individuals' data within the European Union

Kişisel Verileri Koruma Kanunu (KVKK): Regulations that protect the privacy of individuals' data within Turkey.

Baseline: The fixed reference points for comparison of the project.

Software: The set of instructions to operate a specific task on computers.

13. Contributions

Zeynep Hanife Akgül: Definitions, Project Context (Process improvement, Infrastructure and enabling systems, Methods, Tools, and Techniques, Project organization, External interfaces, Internal interfaces, Authorities and responsibilities), Project Planning (Estimation, Staffing), Product Delivery, Supporting Processes (Project supervision and work environment, Quality Assurance, Measurements).

Beyza Çağlar: Project Context (Process Model, Product Acceptance), Project Planning (General, Project Staff Training), Supporting Processes (Information Management).

Alper Göçmen: Project Overview (Project Summary, Purpose, Scope and Objectives, Schedule and Budget Summary), Project Assessment and Control (General, Requirements Management, Scope Change Control, Schedule Control, Budget Control, Quality Assurance, Subcontractor Management), Supporting Processes (General, Project Supervision and Work Environment, Decision Management, Risk Management, Quality Assurance, Measurement, Reviews and Audits, Verification and Validation), Additional Plans.

Deniz Tuna Onguner: Project Overview (Assumptions and Constraints, Project Deliverables), Project Assessment and Control (Project Closeout).

İlayda Zehra Yılmaz: Project Planning (General, Project initiation, Resource acquisition, Project work plans, Work activities, Schedule allocation, Resource allocation, Budget allocation, Procurement, Disposal), End Matter.