Software Project Management Plan

for

IYTERN

Prepared by

GROUP 11

23.12.2023

Table of Contents

Revision History

Name	Date	Reason for Changes	Version
G11_SPMP	30.12.2023	Draft review	1.1

1 Overview

1.1 Purpose, scope, and objectives

Purpose:

- Facilitate the internship search process for students in the Computer Engineering department at Izmir Institute of Technology.
- Connect students with industry professionals to enhance their real-world experiences.
- Transform the internship search experience into a more efficient, effective, and user-friendly process.

Scope:

- Develop a platform named IYTERN to serve as a dedicated tool for internship searches.
- Collaborate with the IYTE student information system to integrate necessary data and streamline processes.
- Focus on the Computer Engineering department at Izmir Institute of Technology.
- Ensure the platform contributes to students' education by providing opportunities for practical, hands-on experiences.

Objectives:

- Create a user-friendly website as the primary interface for the IYTERN platform.
- Integrate the IYTERN platform with the IYTE student information system for seamless data access.
- Simplify the management of internship applications to save students time and effort.
- Enable students to efficiently navigate the internship search process with minimal hassle.
- Optimize students' time by providing easy access to required information for their internship search.

1.2 Assumptions and constraints

Assumptions:

- Facilitating internship search and application procedures for İzmir Institute of Technology (IZTECH) Computer Engineering students is a primary goal.
- Successful integration with IZTECH's student information system is assumed for efficient application management.
- Robust security measures are presupposed to safeguard student information within the system.
- Usability across various devices is assumed, enabling seamless access and utilization by students from different devices.

Constraints:

- Budgetary limitations dictate all expenditures within predefined limits, impacting resource allocation and acquisition.
- Time constraints necessitate adherence to a specific schedule for project milestones, including development phases and deployment.
- Technological constraints may involve specific programming languages or database management systems, influencing the project's technical architecture.
- Limitations in resources, encompassing human resources, hardware, and software availability, are considered.
- Ongoing demands on resources and time allocation are imposed by regular updates, maintenance requirements, and potential bug fixes, constraining the project's operational flow.

These assumptions and constraints are crucial in defining the project's boundaries, guiding its development, and managing expectations concerning the resources, timelines, and technological aspects involved in the creation and maintenance of the IYTERN platform.

1.3 Project deliverables

The primary deliverable of our project will be a fully functional operational website. Additionally, we will provide the user manual for the website along with the product. Alongside these, we will transfer the administrative permissions of the website to an account specified by the customer. In addition to these products, we will deliver the AS-IS report, TO-BE report, Project Charter report, SRS report, SPMP report, Design report and Test report prepared within the project to the customer. The delivery date and location will be decided jointly with the customer.

- **Internship Management System (Due Date: 7-May-2024):** The Internship Management System (IYTERN) is a comprehensive software solution aimed at streamlining, organizing, and optimizing the processes involved in managing internships within the computer engineering faculty in IYTE. This deliverable encompasses the development and deployment of a user-friendly platform designed to facilitate the end-to-end management of internships.

- **User Manual (Due Date: 7-May-2024):** The User Manual for the Internship Management System (IYTERN) is a comprehensive guide designed to assist users in understanding, navigating, and effectively utilizing the features and functionalities of the IYTERN platform. This manual provides step-by-step instructions to ensure a seamless experience in streamlining, organizing, and optimizing internship-related processes within the Computer Engineering faculty at Izmir Institute of Technology (IYTE).

- **AS-IS Report (Due Date: 07-Nov-2023):** This report will document the current state of the internship finding and application processes for İzmir Institute of Technology (IZTECH) Computer Engineering students before the implementation of the IYTERN platform. It will outline the existing procedures, challenges, and inefficiencies in the current system.

- **TO-BE Report (Due Date: 07-Nov-2023):** This report will present the envisioned or desired state of the internship finding and application processes after the implementation of the IYTERN platform. It will describe the improvements, functionalities, and efficiencies that the platform will bring to the process.

- **Project Charter Report (Due Date: 10-Oct-2023):** This document will define the project's purpose, objectives, stakeholders, scope, and high-level timelines. It will outline the project's goals and how the IYTERN platform aligns with the needs of IZTECH Computer Engineering students.

- **SRS Report (Software Requirements Specification) (Due Date: 28-Nov-2023):** This report will detail the functional and non-functional requirements of the IYTERN platform. It will define the features, functionalities, security protocols, and integration aspects required for the successful development and deployment of the website.

- **SPMP Report (Software Project Management Plan) (Due Date: 30-Dec-2023):** This comprehensive document will outline the overall management approach for the IYTERN project. It will include details on project scope, schedule, budget, resources, risks, communication plan, and quality assurance procedures.

- **Testing Reports (Due Date: 21-Apr-2024):** These reports will document the various tests conducted throughout the development lifecycle of the IYTERN platform. They will include test plans, test cases, test results, and any issues or bugs encountered, ensuring the platform's reliability, functionality, and security.

- **Design Report (Due Date: 12-Mar-2024):** This report will outline the architectural and design aspects of the IYTERN platform. It will detail the system architecture, database design, user interface design, and any other technical design considerations crucial for the development and implementation of the website.

1.4 Schedule and budget summary

Our project timeline spans **6 months**, with the goal of delivering the project by the end of the 6. month. The budget allocated for this endeavor is roughly **50.000** dollars.

We have divided the project into **3 main components**. The first part will be responsible for the development of the **website and user interaction**. The second part will involve the **completion of backend software components**. The third stage will **oversee interactions with the IYTE student information system**.

2 References

- Lecture-6_IEEE_PMP_V3.2.pdf
- G11_SPMP.pdf
- G11_SRS.pdf
- G11_Problem_Analysis_Report.pdf
- G11_Design_Report.pdf
- 1_TR_SummerPracticeApplicationLetter_2023.docx
- 2_TR_SummerPracticeApplicationForm2023.docx
- 3_TR_FirmaFormu.docx
- 4_SummerPracticeReportTemplate_Word2023.docx

3 Definitions

- **IYTE:** İzmir Yüksek Teknoloji Enstitüsü
- **IZTECH:** İzmir Institute of Technology
- **CENG323:** Project Management Course at IZTECH
- **IYTERN:** Internship Management System for IZTECH
- **SPMP:** Software Project Management Plan
- **SRS:** Software Requirement Specification

- **UAT:** User Acceptance Testing
- **MS Teams:** Microsoft Teams
- **IEEE:** Institute of Electrical and Electronics Engineers
- **PDF:** Portable Document Format
- **UI:** User Interface
- **FAQ:** Frequently Asked Questions

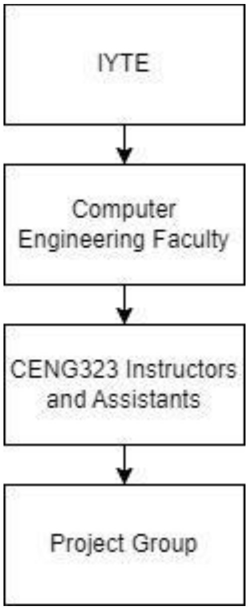
4 Project context

4.1 Product acceptance plan

- After the completion of every milestone, a gathering with the customers and project managers will be done for review.
- If the system accomplishes %95 of the requirements, the product is to be accepted.
- After the coding phase of the project there will be test phases that the IYTERN system must go through. If the error margin is lower than %5, the product is to be accepted.
- The IYTERN system will only do what is said in the scope. Any additional feature requests will not be accepted.

4.2 Project organization

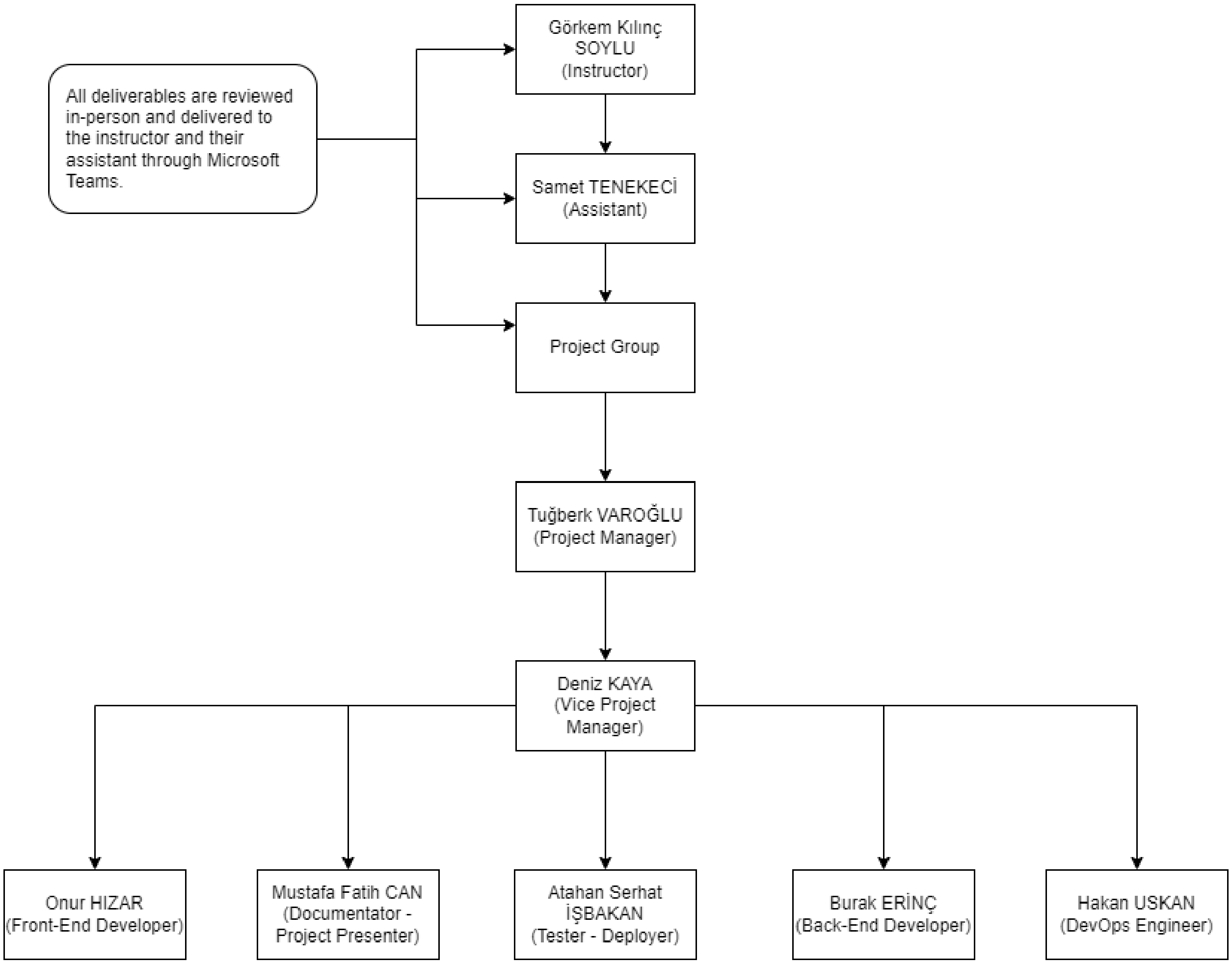
4.2.1 External interfaces


IYTE
Computer Engineering Faculty
CENG323 Instructors and Assistants
Project Group

The project group designs and implements the project's deliverables according to the schedule of CENG323 course which is constructed by the course's instructors and assistants and which is announced at the start of the term in MS Teams. The instructors and assistants are bound to the Computer Engineering Faculty of IYTE.

4.2.2 Internal structure

The diagram below shows the internal structure of the project organization.


All deliverables are reviewed in-person and delivered to the instructor and their assistant through Microsoft Teams.
Görkem Kılınç SOYLU (Instructor)
Samet TENEKECİ (Assistant)
Project Group
Tuğberk VAROĞLU (Project Manager)
Deniz KAYA (Vice Project Manager)
Onur HIZAR (Front-End Developer)
Mustafa Fatih CAN (Documentator - Project Presenter)
Atahan Serhat İŞBAKAN (Tester - Deployer)
Burak ERİNÇ (Back-End Developer)
Hakan USKAN (DevOps Engineer)

4.2.3 Roles and responsibilities

This sub clause of the SPMP shall identify and state the nature of each major work activity and supporting process and identify the organizational units that are responsible for those processes and activities. A matrix of work activities and supporting processes vs. organizational units may be used to depict project roles and responsibilities.

	Onur	Deniz	Fatih	Serhat	Tuğberk	Burak	Hakan	Görkem Kılınç SOYLU	Samet TENEKECİ
Project Management	R	R	R	R	AR	R	R	I	I
UI Design	AR	R	R	R	R	R	R	C	C
Frontend	AR	R	R	R	R	R	R	C	C
Backend	R	R	R	R	R	AR	R	C	C
Devops	R	R	R	R	R	R	AR	C	C
Database	R	AR	R	R	R	R	R	C	C
Testing	R	R	R	AR	R	R	R	C	C
Project Documentation	R	AR	R	R	R	R	R	C	C
Presentation	R	R	AR	R	R	R	R	C	C
Deployment	R	R	R	AR	R	R	R	C	C
Design	AR	R	R	R	R	R	R	C	C
Post-Implementation Review	R	R	R	R	R	R	AR	C	C

A: Accountable I: Informed C: Consulted R: Responsible

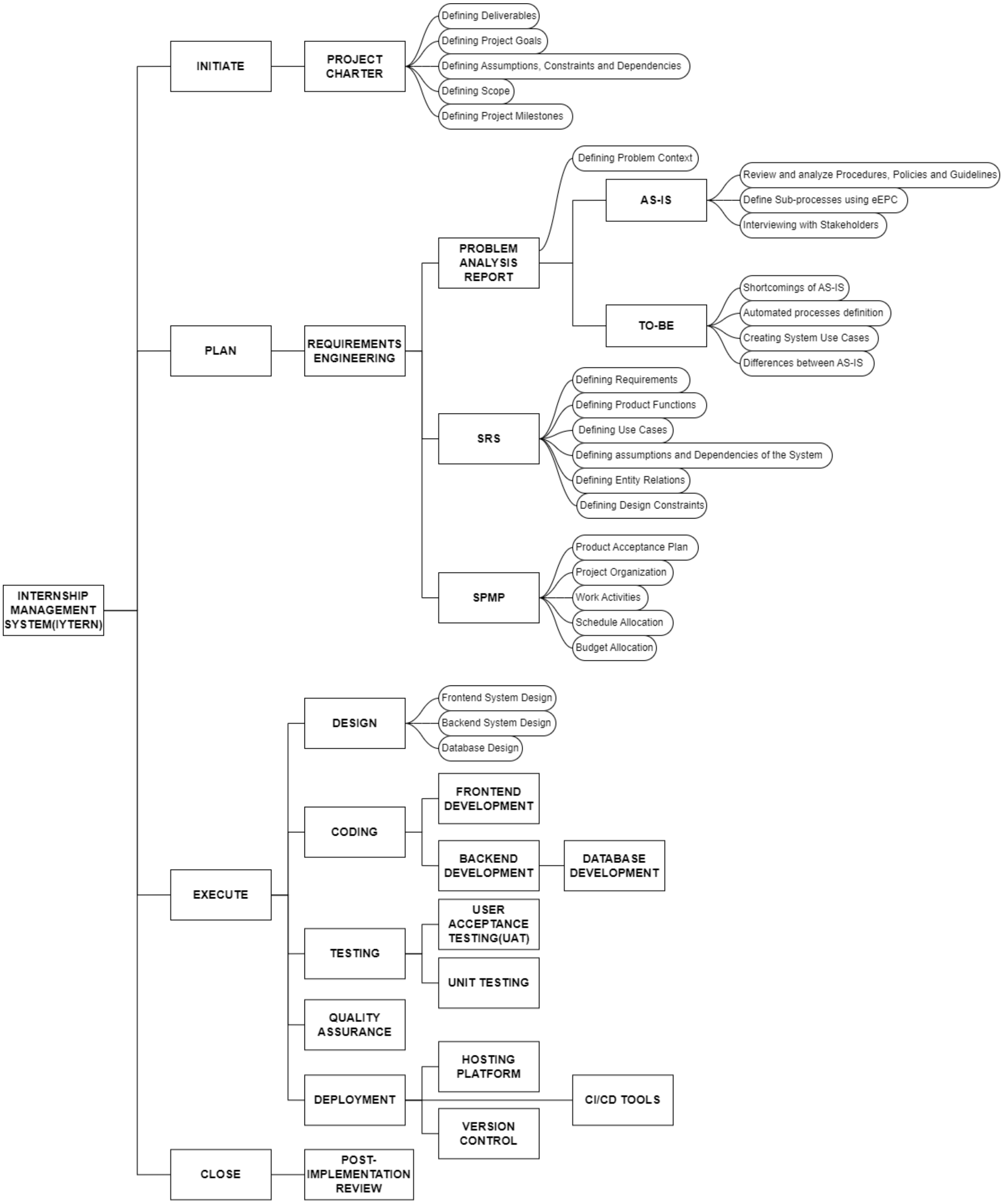
5 Project planning

This clause of the SPMP shall specify the work activities, schedule, resources, and budget details for the software project.

5.1 Project work plans

This clause of the SPMP shall specify the work activities, schedule, resources, and budget details for the software project.

5.1.1 Work activities



Work Package: Project Charter	
Description: The Project Charter serves as the cornerstone document that formally initiates a project by articulating its purpose, objectives, and scope. This foundational document provides a concise overview of the project's goals, identifies key stakeholders and their roles, outlines high-level timelines and budgetary considerations, and highlights potential risks and mitigation strategies. By establishing a common understanding among stakeholders and obtaining formal approval, the Project Charter lays the groundwork for successful project management, guiding decision-making and serving as a reference point throughout the project's lifecycle. It is a dynamic document that may evolve as the project progresses, providing a solid foundation for effective planning and execution.	
Predecessor: -	Successor: Problem Analysis Report
Acceptance Criteria: The Project Charter had to meet the specifications set by the stakeholders of the project.	
Work products: Project Charter	
Resources: Microsoft Word tools were used by the project team.	
Start Date:03.10.2023	End Date:10.10.2023

Work Package: Problem Analysis Report	
Description: A Problem Analysis Report is a comprehensive document designed to explore and analyze specific issues or challenges within a given context. This report delves into the root causes of the problem, providing an in-depth examination of its various dimensions. It typically includes a detailed problem statement, outlining the nature and scope of the issue, as well as its impact on relevant stakeholders. The report employs analytical tools, data, and evidence to dissect the problem, identifying contributing factors and potential obstacles to resolution. Moreover, it may propose viable solutions or recommendations based on the findings of the analysis. A Problem Analysis Report is an essential tool for decision-makers, offering valuable insights that inform strategic planning and guide the development of effective solutions to address the identified challenges.	
Predecessor: Project Charter	Successor: SRS
Acceptance Criteria: The Problem Analysis Report should analyze the problem very well and should come up with a solution to that problem.	
Work products: AS-IS and TO-BE reports	
Resources: draw.io, p11.microarc.net, Microsoft Word tools were used by the project team.	
Start Date: 08.10.2023	End Date: 07.11.2023

Work Package: SRS	
Description: A Software Requirements Specification (SRS) is a crucial document in the software development life cycle that serves as a comprehensive guide outlining the functional and non-functional requirements of a software system. It acts as a bridge between stakeholders, including clients and development teams, by clearly defining the features, functionalities, and constraints that the software must adhere to. The SRS document typically includes a detailed description of the system's purpose, scope, and objectives. It further breaks down specific requirements, such as user interactions, system interfaces, performance criteria, and security considerations. Through meticulous detailing of the software's behavior and constraints, the SRS becomes a reference point for developers, testers, and other project stakeholders, ensuring a shared understanding of the project's goals and characteristics. This document evolves throughout the development process, reflecting changes in requirements and providing a foundation for effective project management and successful software delivery.	
Predecessor: Problem Analysis Report	Successor: SPMP
Acceptance Criteria: The SRS should specify the requirements very well with defined use cases.	
Work products: SRS report	
Resources: draw.io, Microsoft Word tools were used by the project team.	
Start Date: 15.11.2023	End Date: 28.11.2023

Work Package: SPMP	
Description: The SPMP, or Software Project Management Plan, is a comprehensive document that outlines how a software development project will be planned, executed, monitored, and controlled. This plan serves as a guiding document for the project team and stakeholders, providing a roadmap for successfully managing and completing the software project. The SPMP typically includes information about project objectives, scope, schedule, budget, resource management, risk management, quality assurance, communication strategies, and other essential aspects of project management. By detailing the project management approach, the SPMP helps ensure that all team members and stakeholders are on the same page, fostering effective communication, collaboration, and successful project delivery.	
Predecessor: SRS	Successor: Design
Acceptance Criteria: The SPMP document should define the management of the project very well leaving no doubts behind.	
Work products: SPMP report	
Resources: draw.io, onlinegantt.com, Microsoft Word tools were used by the project team.	
Start Date: 06.12.2023	End Date: 07.01.2024

Work Package: Design	
Description: Design is a multifaceted and iterative process that involves conceiving, planning, and creating solutions to address specific needs or challenges. It spans various disciplines, including graphic design, product design, and user experience design, each with its unique approach and considerations. In essence, design encompasses the synthesis of creativity, functionality, and user-centricity, aiming to produce solutions that are not only aesthetically pleasing but also effective in fulfilling their intended purposes. The design process typically involves stages such as research, ideation, prototyping, and testing, emphasizing a user-centered approach to ensure the final product or solution aligns with user expectations and needs. Design is not merely about aesthetics; it's a strategic and problem-solving endeavor that plays a pivotal role in shaping the functionality and appeal of everything from digital interfaces to physical products, ultimately enhancing user experiences and driving innovation.	
Predecessor: SPMP	Successor: Frontend Development
Acceptance Criteria: The Design should design system architecture, UI/UX, database schemas, class and sequence diagrams adjusted for future developments.	
Work products: IYTERN system architecture, IYTERN database schema, IYTERN UI/UX design, IYTERN class and sequence diagrams	
Resources: Lucid Chart, Microsoft Word tools were used by the project team.	
Start Date: 06.03.2024	End Date: 26.04.2024

Work Package: Frontend Development	
Description: Frontend development is the process of creating and implementing the visual and interactive components of a website or web application that users directly engage with. Frontend developers use a combination of technologies, including HTML, CSS, and JavaScript, to structure and style web content and enhance user interactivity. This discipline focuses on crafting an intuitive and visually appealing user interface (UI), ensuring a seamless and engaging user experience across various devices and browsers. Frontend development involves translating design concepts into code, implementing responsive layouts, and integrating dynamic elements, such as animations and interactive features, to deliver a compelling and functional online interface. It is a crucial aspect of web development that bridges the gap between design and backend functionality, ultimately shaping how users perceive and interact with digital products on the client side.	
Predecessor: Design	Successor: Testing
Acceptance Criteria: The System should have user friendly UI and good optimization with the system's backend and should have a very small number of errors (1-25). %99 of users should be able to finish their process in the system without any confusion. The system has a 3-minute learning phase for %95 of the users.	
Work products: Frontend Development Document, UI	
Resources: ReactJS, VS Code	
Start Date: 26.03.2024	End Date: 01.05.2024

Work Package: Backend Development	
Description: Backend development refers to the creation and maintenance of the server-side logic, databases, and application architecture that power websites and web applications. Backend developers work on the server side, managing data storage, handling user requests, and ensuring the overall functionality and performance of the software. They use server-side programming languages like Python, Java, Ruby, or Node.js, along with frameworks such as Django, Flask, Spring, or Express, to build the server-side logic. Additionally, backend developers interact with databases, design APIs (Application Programming Interfaces), and implement security measures to protect data. The collaboration between frontend and backend developers ensures a seamless and robust web application with effective data processing, storage, and retrieval capabilities.	
Predecessor: Frontend Development	Successor: User Acceptance Testing
Acceptance Criteria: The backend system should have good optimization with the system's frontend. The system should send and receive the documents with %99 accuracy. Only the administrator should be able to grant privileges to users.	
Work products: Database Management System, Backend Development Document	
Resources: IntelliJ Idea, VS Code, Cloud Services, CI/CD Tools, Docker, Postman	
Start Date: 26.03.2024	End Date: 01.05.2024

Work Package: Unit Testing	
Description: Unit Testing is a fundamental software testing method where individual components or units of a software application are tested in isolation to ensure their correctness and functionality. These units can be functions, methods, or even entire modules. The primary goal of unit testing is to validate that each unit of code performs as expected, and it helps identify and fix bugs early in the development process. Unit tests are typically automated, making it easier for developers to run them frequently and consistently during the development cycle. Testing frameworks like JUnit for Java, NUnit for .NET, or Jest for JavaScript are commonly used to structure and execute unit tests. The isolation of units allows developers to focus on specific functionalities, promoting code reliability, maintainability, and overall software quality.	
Predecessor: Coding	Successor: Quality Assurance
Acceptance Criteria: The system should be able to fend off %95 of malicious users from disrupting the system. The System is %85 Fault Tolerant of the user mistakes. The system should send and receive the documents with %99 accuracy. The unit should meet performance expectations and should handle valid inputs correctly.	
Work products: Unit Testing Document	
Resources: JUnit, Mockito, Spring Boot Testing (including SpringBootTest, MockMvc, TestRestTemplate), Jest, Mocha.	
Start Date: 05.04.2024	End Date: 22.05.2024

Work Package: User Acceptance Testing	
Description: User Acceptance Testing (UAT) is the final phase in the software testing process where end-users evaluate the software to ensure that it meets their requirements and expectations before being deployed into the production environment. During UAT, real users interact with the system, assess its functionality, and validate that it aligns with the specified business requirements. This testing phase aims to identify any discrepancies between the intended and actual outcomes, as well as to uncover any issues that may have been missed in earlier testing stages. Successful completion of UAT is crucial for gaining user confidence, ensuring the software's usability, and minimizing the risk of post-deployment issues. It serves as the final checkpoint before the software is officially released to the user community, emphasizing the importance of thorough testing and collaboration between development teams and end-users.	
Predecessor: Coding	Successor: Quality Assurance
Acceptance Criteria: %99 of users should be able to finish their process without any confusion.The System has a 3 minute learning phase for %95 of the users.	
Work products: Test Report	
Resources: -	
Start Date:01.05.2024	End Date:22.05.2024

Work Package: Quality Assurance	
Description: Quality Assurance (QA) is a systematic process within the software development lifecycle that ensures the delivery of a high-quality product or service. It involves a set of activities, methodologies, and standards aimed at preventing defects, validating compliance with requirements, and enhancing overall product excellence. QA encompasses various stages, including requirement analysis, design, development, and testing, with the goal of identifying and addressing potential issues at each step. QA processes involve the creation and implementation of testing strategies, test plans, and test cases to verify that the software meets specified standards and requirements. Additionally, QA emphasizes continuous improvement by analyzing feedback, identifying areas for enhancement, and refining processes to achieve optimal results. Effective QA practices contribute to increased customer satisfaction, reduced defects, and a more robust and reliable software product.	
Predecessor: Deployment	Successor: Post Implementation Review
Acceptance Criteria: Our system should reach the designated quality that we designed.	
Work products: QA Report	
Resources: Computers, that we will check the quality of our system with.	
Start Date: 21.05.2024	End Date:29.05.2024

Work Package: Deployment	
Description: Deployment in software development is the pivotal phase where a developed application transitions from a testing or development environment to the live production environment. It involves transferring code, configuring settings, and ensuring that the application functions seamlessly in the production setting. Deployment processes may encompass database updates, environment testing, and the establishment of rollback procedures for contingencies. Whether done manually or automated through CI/CD pipelines, efficient deployment practices are critical to ensuring a smooth and reliable release, minimizing downtime, and ultimately delivering a high-quality software experience to end-users.	
Predecessor: Unit Testing	Successor: Quality Assurance
Acceptance Criteria: Application is online at Server	
Work products: Server and Docker Container	
Resources: Google Cloud, Docker	
Start Date: 15.04.2024	End Date: 22.05.2024

Work Package: Post Implementation Review	
Description: A Post-Implementation Review (PIR) is a systematic evaluation conducted after the completion and deployment of a project or a specific phase. It aims to assess the overall success of the implementation, identify lessons learned, and gather insights to improve future projects. During a PIR, key stakeholders, project managers, and team members analyze the project's outcomes, comparing them against the initial objectives and expectations. The review covers aspects such as project performance, budget adherence, timeline achievement, and the satisfaction of end-users. By examining both successes and challenges, a PIR provides valuable feedback to refine processes, enhance organizational learning, and inform decision-making for subsequent projects. The insights gained from a Post-Implementation Review contribute to continuous improvement, fostering a culture of adaptability and efficiency within the project management lifecycle.	
Predecessor: Quality Assurance	Successor: -
Acceptance Criteria: -	
Work products: Post Implementation Review Report	
Resources: Google Forms, Google Analytics, SharePoint, Microsoft Teams, Jira.	
Start Date: 22.05.2024	End Date: 03.06.2024

5.1.2 Schedule allocation

Scheduling Relationships:

- **Milestone 1: Collaborative Planning (Week 20)** Define collaboration protocols for frontend and backend teams.
  - Criteria: Established communication channels and task allocation strategy.

Constraints on Scheduling:

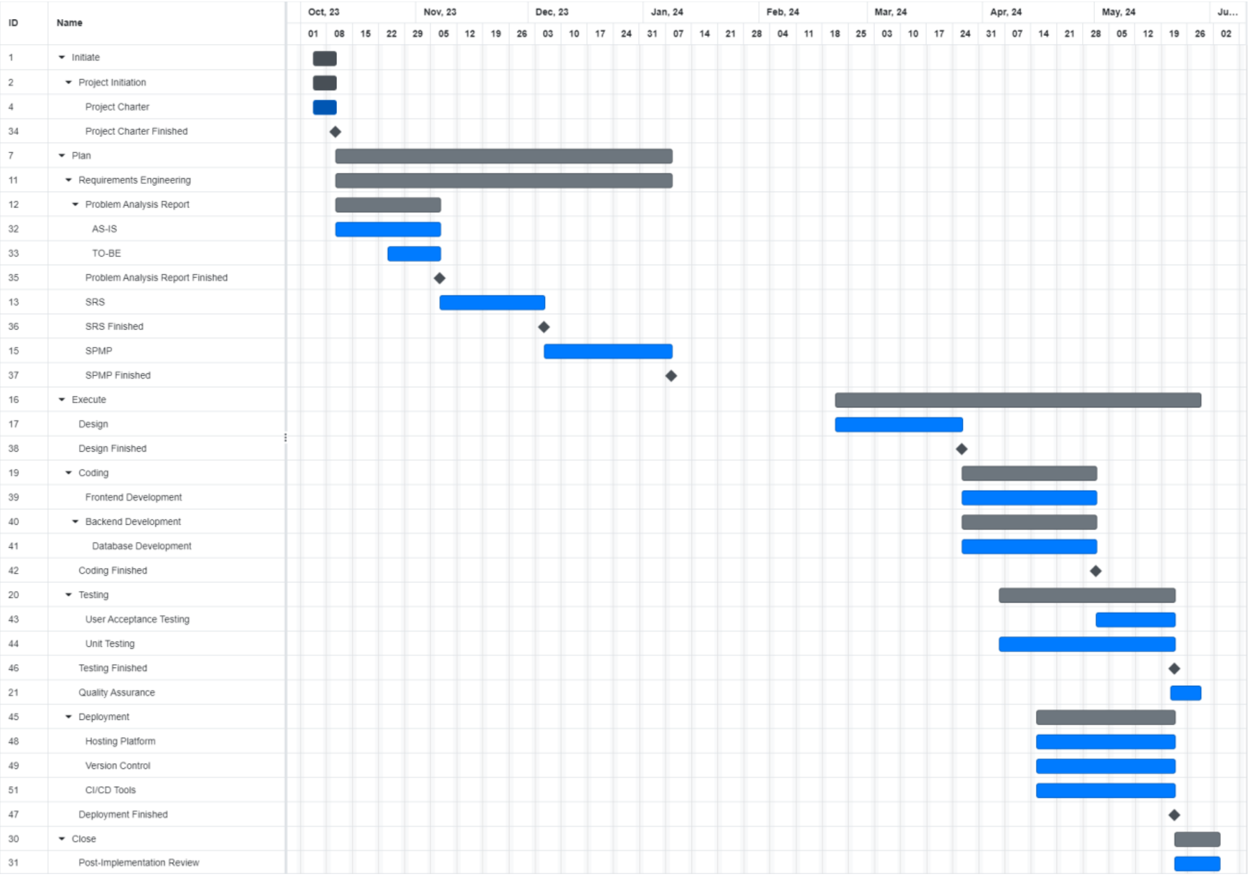
- **Milestone 2: Project Scope and Timeline Definition (Project charter) (Week 1-2)**
  - We need to finish the whole project in 30 weeks, by the end of the spring semester. This is the main rule for planning and completing our work.
  - Criteria: Documented project scope and timeline alignment.
- **Milestone 3: Problem Analysis Report (Week 2-6)**
  - This document will specify the AS-IS process, TO-Be process and the differences between the AS-IS and TO-BE process.
  - The improvements from the AS-IS process will be defined.
- **Milestone 4: Software Requirements Specification Report (Week 6-10)**
  - This document's main purpose is to specify the requirements of the project.
  - Use cases and the interfaces for these cases will be defined in this report.
- **Milestone 5: Software Project Management Plan (Week 10-14)**
  - This document will define both the internal and external management plan of the project.
  - Budget, time and acceptance constraints for all work activities are defined.
- **Milestone 6: Backend Preparation (Weeks 15-19)**
  - Review backend database entity model shapes.
  - Criteria: Approved backend database model for development.
  - Dependency: Completion required before backend API planning.
- **Milestone 7: Backend API Planning (Weeks 15-19)**
  - Assess and define backend API routes and return shapes.
  - Criteria: Finalized API routes documented and ready for implementation.
  - Dependency: Completion required before frontend planning.
- **Milestone 8: Frontend Planning (Weeks 15-19)**
  - Decide on front-end page names and layouts.
  - Criteria: Finalized frontend structure approved by UI/UX team.
  - Dependency: Completion needed for frontend and backend implementation.
- **Milestone 9: Frontend and Backend Implementation (Weeks 20-24)**
  - Commence frontend and backend implementation based on agreed structures.
  - Criteria: Deployment of barebones frontend with dummy data in agreed shape.

    - Dependency: Completion required before backend entities.
- **Milestone 10: Backend Entity Development (Weeks 20-24)**
    - Develop backend entities following the agreed data shapes.
    - Criteria: Deploy backend entities without file upload feature.
    - Dependency: Required before backend-frontend connection.
- **Milestone 11: Backend-Frontend Integration (Weeks 24-25)**
    - Establish connection between deployed backend and frontend versions.
    - Criteria: Frontend renders real data from connected backend.
    - Dependency: Completion needed after backend entity deployment.
- **Milestone 12: Data Validation Implementation (Weeks 20-24)**
    - Implement data validation protocols for frontend and backend.
    - Criteria: Implemented validation for critical data fields.
    - Dependency: Required before PDF upload feature.
- **Milestone 13: PDF Upload Feature Development (Weeks 24-25)**
    - Develop and deploy backend and frontend PDF upload features.
    - Criteria: Feature deployment without affecting existing functionalities.
    - Dependency: Completion needed after data validation.
- **Milestone 14: MVP Design Enhancement (Weeks 23-24)**
    - Enhance design for the Minimum Viable Product (MVP).
    - Criteria: UI/UX improvements aligned with project goals.
    - Dependency: None but precedes testing phase.
- **Milestone 15: Testing Phase (Weeks 21-27)**
    - Conduct security, UI, and load tests.
    - Criteria: Test reports with identified issues and solutions.
    - Dependency: Completion required after MVP design enhancement.
- **Milestone 16: Iterative Improvements (Weeks 21-27)**
    - Address identified issues from the testing phase.
    - Criteria: Implemented fixes and enhancements.
    - Dependency: Required after the testing phase if needed.
- **Milestone 17: Final Design and Backend Improvements (Week 28-30)**
    - Address any pending design or backend improvements.
    - Criteria: Implemented final improvements.
    - Dependency: None but follows the completion of the development phase.
- **Milestone 18: Project Presentation Preparation (Week 30)**
    - Prepare for the final project presentation.
    - Criteria: Presentation materials ready for submission.
    - Dependency: None but follows the completion of all development phases.

Techniques:

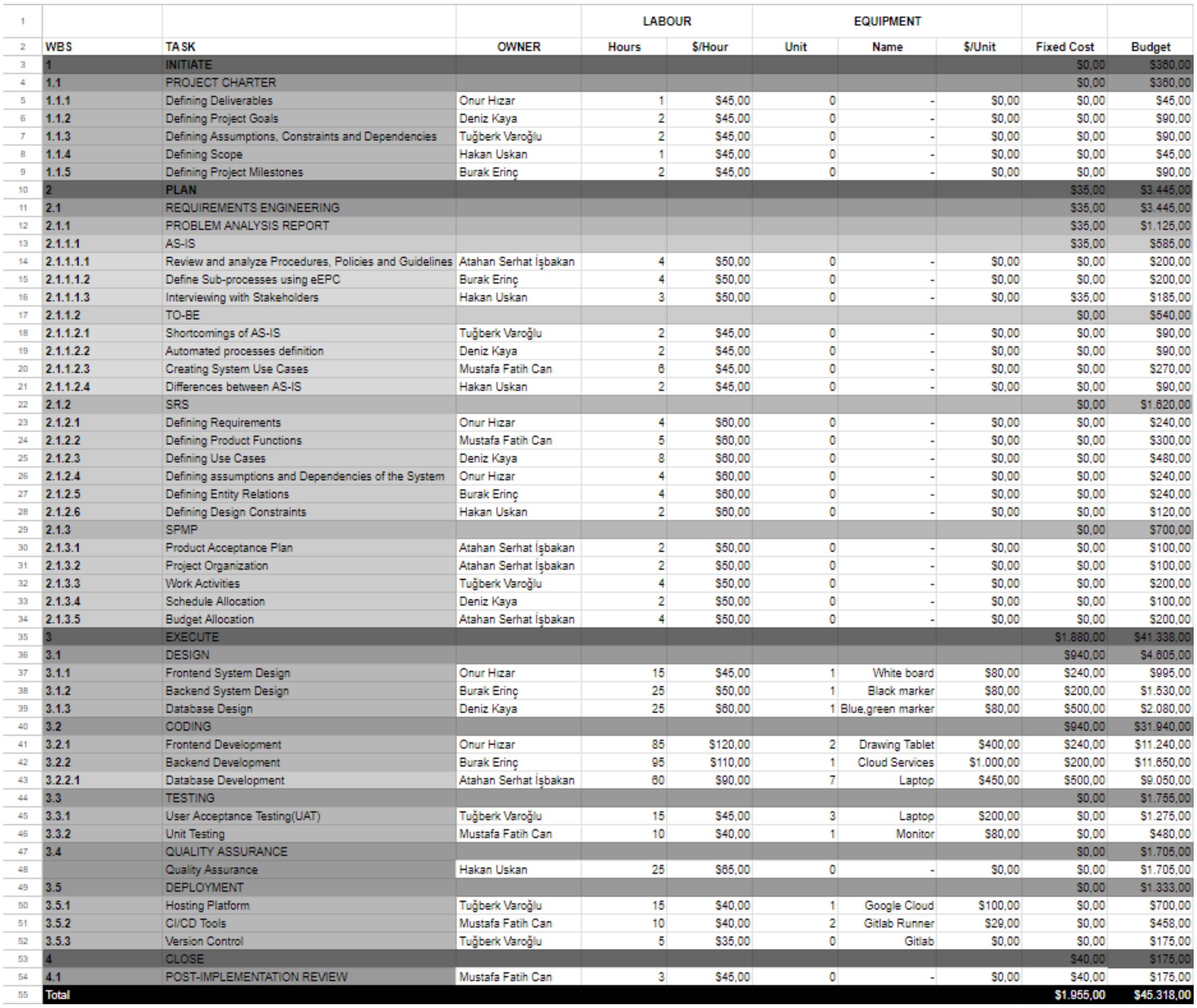
We will use **milestone charts**, **kanban tables** and **Gantt charts**.

Gantt Chart of the schedule:



ID	Name
1	▾ Initiate
2	▾ Project Initiation
4	Project Charter
34	Project Charter Finished
7	▾ Plan
11	▾ Requirements Engineering
12	▾ Problem Analysis Report
32	AS-IS
33	TO-BE
35	Problem Analysis Report Finished
13	SRS
36	SRS Finished
15	SPMP
37	SPMP Finished
16	▾ Execute
17	Design
38	Design Finished
19	▾ Coding
39	Frontend Development
40	▾ Backend Development
41	Database Development
42	Coding Finished
20	▾ Testing
43	User Acceptance Testing
44	Unit Testing
46	Testing Finished
21	Quality Assurance
45	▾ Deployment
48	Hosting Platform
49	Version Control
51	CI/CD Tools
47	Deployment Finished
30	▾ Close
31	Post-Implementation Review

5.1.3 Budget allocation



			LABOUR		EQUIPMENT				
WBS	TASK	OWNER	Hours	$/Hour	Unit	Name	$/Unit	Fixed Cost	Budget
1	INITIATE							$0,00	$360,00
1.1	PROJECT CHARTER							$0,00	$360,00
1.1.1	Defining Deliverables	Onur Hızar	1	$45,00	0	-	$0,00	$0,00	$45,00
1.1.2	Defining Project Goals	Deniz Kaya	2	$45,00	0	-	$0,00	$0,00	$90,00
1.1.3	Defining Assumptions, Constraints and Dependencies	Tuğberk Varoğlu	2	$45,00	0	-	$0,00	$0,00	$90,00
1.1.4	Defining Scope	Hakan Uskan	1	$45,00	0	-	$0,00	$0,00	$45,00
1.1.5	Defining Project Milestones	Burak Erinç	2	$45,00	0	-	$0,00	$0,00	$90,00
2	PLAN							$35,00	$3.445,00
2.1	REQUIREMENTS ENGINEERING							$35,00	$3.445,00
2.1.1	PROBLEM ANALYSIS REPORT							$35,00	$1.125,00
2.1.1.1	AS-IS							$35,00	$585,00
2.1.1.1.1	Review and analyze Procedures, Policies and Guidelines	Atahan Serhat İşbakan	4	$50,00	0	-	$0,00	$0,00	$200,00
2.1.1.1.2	Define Sub-processes using eEPC	Burak Erinç	4	$50,00	0	-	$0,00	$0,00	$200,00
2.1.1.1.3	Interviewing with Stakeholders	Hakan Uskan	3	$50,00	0	-	$0,00	$35,00	$185,00
2.1.1.2	TO-BE							$0,00	$540,00
2.1.1.2.1	Shortcomings of AS-IS	Tuğberk Varoğlu	2	$45,00	0	-	$0,00	$0,00	$90,00
2.1.1.2.2	Automated processes definition	Deniz Kaya	2	$45,00	0	-	$0,00	$0,00	$90,00
2.1.1.2.3	Creating System Use Cases	Mustafa Fatih Can	6	$45,00	0	-	$0,00	$0,00	$270,00
2.1.1.2.4	Differences between AS-IS	Hakan Uskan	2	$45,00	0	-	$0,00	$0,00	$90,00
2.1.2	SRS							$0,00	$1.620,00
2.1.2.1	Defining Requirements	Onur Hızar	4	$60,00	0	-	$0,00	$0,00	$240,00
2.1.2.2	Defining Product Functions	Mustafa Fatih Can	5	$60,00	0	-	$0,00	$0,00	$300,00
2.1.2.3	Defining Use Cases	Deniz Kaya	8	$60,00	0	-	$0,00	$0,00	$480,00
2.1.2.4	Defining assumptions and Dependencies of the System	Onur Hızar	4	$60,00	0	-	$0,00	$0,00	$240,00
2.1.2.5	Defining Entity Relations	Burak Erinç	4	$60,00	0	-	$0,00	$0,00	$240,00
2.1.2.6	Defining Design Constraints	Hakan Uskan	2	$60,00	0	-	$0,00	$0,00	$120,00
2.1.3	SPMP							$0,00	$700,00
2.1.3.1	Product Acceptance Plan	Atahan Serhat İşbakan	2	$50,00	0	-	$0,00	$0,00	$100,00
2.1.3.2	Project Organization	Atahan Serhat İşbakan	2	$50,00	0	-	$0,00	$0,00	$100,00
2.1.3.3	Work Activities	Tuğberk Varoğlu	4	$50,00	0	-	$0,00	$0,00	$200,00
2.1.3.4	Schedule Allocation	Deniz Kaya	2	$50,00	0	-	$0,00	$0,00	$100,00
2.1.3.5	Budget Allocation	Atahan Serhat İşbakan	4	$50,00	0	-	$0,00	$0,00	$200,00
3	EXECUTE							$1.880,00	$41.338,00
3.1	DESIGN							$940,00	$4.605,00
3.1.1	Frontend System Design	Onur Hızar	15	$45,00	1	White board	$80,00	$240,00	$995,00
3.1.2	Backend System Design	Burak Erinç	25	$50,00	1	Black marker	$80,00	$200,00	$1.530,00
3.1.3	Database Design	Deniz Kaya	25	$60,00	1	Blue,green marker	$80,00	$500,00	$2.080,00
3.2	CODING							$940,00	$31.940,00
3.2.1	Frontend Development	Onur Hızar	85	$120,00	2	Drawing Tablet	$400,00	$240,00	$11.240,00
3.2.2	Backend Development	Burak Erinç	95	$110,00	1	Cloud Services	$1.000,00	$200,00	$11.650,00
3.2.2.1	Database Development	Atahan Serhat İşbakan	60	$90,00	7	Laptop	$450,00	$500,00	$9.050,00
3.3	TESTING							$0,00	$1.755,00
3.3.1	User Acceptance Testing(UAT)	Tuğberk Varoğlu	15	$45,00	3	Laptop	$200,00	$0,00	$1.275,00
3.3.2	Unit Testing	Mustafa Fatih Can	10	$40,00	1	Monitor	$80,00	$0,00	$480,00
3.4	QUALITY ASSURANCE							$0,00	$1.705,00
	Quality Assurance	Hakan Uskan	25	$65,00	0	-	$0,00	$0,00	$1.705,00
3.5	DEPLOYMENT							$0,00	$1.333,00
3.5.1	Hosting Platform	Tuğberk Varoğlu	15	$40,00	1	Google Cloud	$100,00	$0,00	$700,00
3.5.2	CI/CD Tools	Mustafa Fatih Can	10	$40,00	2	Gitlab Runner	$29,00	$0,00	$458,00
3.5.3	Version Control	Tuğberk Varoğlu	5	$35,00	0	Gitlab	$0,00	$0,00	$175,00
4	CLOSE							$40,00	$175,00
4.1	POST-IMPLEMENTATION REVIEW	Mustafa Fatih Can	3	$45,00	0	-	$0,00	$40,00	$175,00
Total								$1.955,00	$45.318,00

6 Product delivery

The IYTERN System, developed within the IYTERN Project, has a team that uses Jira for reviewing the status of the project and for making new improvements. The project team performs regular status meetings on Microsoft Teams to keep all team members updated. The IYTERN project team additionally communicates with various future users, including companies and administrators, about the status, usage requirements and the delivery progress of the IYTERN internship management system also through various meetings done through Microsoft Teams during the development, usage and post-delivery phases. Communications within the project and between the team and future users are end-to-end encrypted also through Microsoft Teams.

The IYTERN System will be delivered as a live website deployed on a remote server managing internships between companies, students, deanery and other stakeholders. Depending on the user, usage instructions for creating, managing, and completing internships will be given on the manual section of the website.

The IYTERN system's website will also have FAQ, troubleshooting and contact information sections to guide all users and get issues fixed that users or the system can have. These sections will be easily accessible within the system's website.