

**TOBB ETU**

**Economy & Technology University**

**BIL 481**

**Project Management Plan (PMP)**

**Reference:** IEEE 1058 / ISO/IEC 12207

**TOBBLU HIZMET**Enes Kerem GOKSU  
Mohammadsafa KHORMALY  
Mehmet Alp OGUZTEKIN  
Furkan UZUNPOSTALCI

## 1. Project Overview

The primary goal of the "TOBBLU HIZMET" project is to develop a modern, web-based freelancing platform designed to efficiently connect clients with skilled professionals. The project addresses the common challenge in the gig economy where finding the right local talent is often disorganized and time-consuming. The platform aims to eliminate this friction by providing a centralized marketplace with a core matching system that filters freelancers by location, skills, and price. To foster security and commitment, the project will also implement a demo payment system creating a reliable community for hiring.

The specific deliverables for the BIL 481 project are structured across three assignments and a final demo. They consist of project documentation and a functional implementation of four selected use cases.

* Assignment 1 (Planning Documents):
  + Project Definition Document
  + Requirements Document
  + Project Plan Document
* Assignment 2 (Design Documents):
  + Design Document (detailing the architecture for 4 selected use cases)
  + Quality Assurance (QA) Plan
* Assignment 3 (Implementation & Test Documents):
  + Deployment Plan
  + Acceptance Tests and Acceptance Criteria (for the 4 use cases)
  + Test Results Report (for the 4 use cases)
* Final Code and Demonstration:
  + The complete, well-commented source code for the project, which must be fully functional for the four selected use cases.
  + A final project presentation and demo of these four implemented use cases.

## 2. Organization and Roles

## The "TOBBLU HIZMET" project team consists of four members. Responsibilities are divided based on project needs and member skills, primarily splitting into front-end and back-end development tasks.

| **Team Member** | **Role(s)** | **Primary Responsibilities** |
| --- | --- | --- |
| **Enes Kerem GOKSU**  **AND**  **Furkan UZUNPOSTALCI** | Front-end Developers |  **User Accounts & Login**  Build the Login and Sign-up screen pages.   **User Profile Page**  Build the main profile page layout. Create the UI for adding, removing, and displaying a profile picture. Create the forms for adding/editing name, surname, and location (country/city). Create a selector for user type (client or freelancer). Create the UI for adding/editing a list of skills or needs. Display the user's calculated rating. Create the "Invitation button". Create the "Messenger button".   **Matching & Search**  Build the UI for the search/matching system. Create the filter inputs for skills, location, and price. Create the UI for the client to choose how to sort results (by price or rating). Build the "results" page to display the list of matched freelancers.   **Chat & Confirmation**  Build the basic chatting page UI. Implement the real-time display of messages (without saving history). Create the "Done" button within the chat for both users.   **Payment & Rating**  Build the static "Payment" page that just says "coming soon". Create the UI (e.g., a pop-up or page) for a client to leave a rating and review for a freelancer. |
| **Mohammadsafa KHORMALY**  **AND**  **Mehmet Alp OGUZTEKIN** | Back-end Developers |  **Project Setup**  Set up the Git and GitHub repository. Choose and set up the database. Set up the free cloud service for deployment.   **User Accounts & Authentication**  Create the API endpoint for user registration (creating a new user). Create the API endpoint for user login (authenticating a user).   **User Profile API**  Create API endpoints for updating and getting profile data.   **Core Matching System Logic**  Implement the matching algorithm that finds users. Implement the filtering logic for skills and price. Implement the sorting logic for the results (by price or rating).   **Chat & Confirmation API**  Set up the WebSocket server for real-time, non-persistent chatting. Create the logic for the "Invitation button" (to start a chat). Create the logic for the "Messenger button" (to open a chat). Create the API endpoint for the "Done" button.   **Payment & Rating API**  Implement the demo subscription logic (check if a user has a plan to use the platform). Implement the demo payment logic. Create the API endpoint for clients to submit a rating/review. Implement the logic to calculate and update a freelancer's rating based on points received. |

## Shared Responsibilities: All team members are responsible for contributing to the creation, review, and submission of all project documentation as required by Assignments 1, 2, and 3.

## 3. Project Schedule

This project is divided into three main phases. The timeline aligns with the three-assignment structure of the BIL 481 course.

| **Phase** | **Phase Name** | **Key Tasks & Activities** | **Estimated Start Date** | **Estimated End Date (Milestone)** |
| --- | --- | --- | --- | --- |
| **Phase 1** | Project Planning & Requirements | • Define project scope, objectives, and risks.  • Identify all functional and non-functional requirements.  • Create the initial project plan and assign roles. | Oct 23, 2025 | Nov 7, 2025 |
| **Phase 2** | System Design & QA Planning | • Design the system architecture and technology stack.  • Select 4 key use cases.  • Create detailed design for the 4 use cases (data flow, UIs).  • Develop the Quality Assurance (QA) Plan. |  |  |
| **Phase 3** | Implementation & Testing | • Develop the code for the 4 selected use cases.  • Conduct unit and integration testing as per the QA plan.  • Create the Deployment Plan.  • Design and document Acceptance Tests.  • Run tests and create the Test Results Report. |  |  |

## 4. Budget and Resources

1. Human Resources

* Project Team: This is the project team of four members (Enes Kerem GOKSU, Mohammadsafa KHORMALY, Mehmet Alp OGUZTEKIN, Furkan UZUNPOSTALCI), who will contribute their time and skills in project management, front-end, and back-end development.

2. Software & Tools

* Version Control: Git and GitHub for collaborative code management and submission.
* Development Environment: Standard IDEs.
* Technology Stack:
  + Database:
  + Backend: Java Spring Boot
  + Frontend:
* Hosting: The project will be deployed for the demo using free-tier cloud services.

3. Hardware Resources

* Development Laptops: No specialized hardware is required. All development and testing will be performed on the team members' standard laptops.
* Learning Resources
* Official documentation for the chosen technology stack, online tutorials, and AI-powered assistants will be used to research and implement best practices.

4. Budget

* Budget: The project is designed to be completed with a zero-dollar budget.
* Resource Allocation: All software, tools, and hosting services will be open-source or fall within the free-tier usage limits of cloud providers. The primary resource investment is the time and effort of the team members, which will be allocated across the project phases as defined in the Project Schedule.

## 5. Risk Management

* Risk 1:
  + Description: The team may be tempted to add new, unplanned features during development, which could delay the project.
  + Mitigation: All proposed changes must be reviewed and approved by the team.
* Risk 2:
  + Description: The team may encounter unexpected challenges with new technologies.
  + Mitigation: Allocate time for research and learning. Encourage pair programming to solve complex problems and select technologies that the team has some familiarity with.
* Risk 3:
  + Description: A team member may fall behind due to unforeseen circumstances.
  + Mitigation: Hold regular, brief check-in meetings to monitor progress and identify issues early. Foster an environment where team members can ask for help.

## 6. Quality Assurance

This section defines the project's overall quality assurance strategy, the high-level quality constraints, and the specific metrics and plans for testing the "TOBBLU HIZMET" platform, in line with the requirements for the Quality Assurance (QA) Plan.

1. Quality Assurance Strategy

Our strategy integrates quality assurance throughout the project lifecycle.

* Testing Methodologies:
  + Unit Testing
  + Integration Testing
  + Usability Testing
  + Acceptance Testing
* Automated vs. Manual Testing:
  + Automated: Unit tests for back-end logic will be automated.
  + Manual: All usability, integration, and final acceptance testing for the 4 use cases will be conducted manually to ensure real-world scenarios are met.

2. Software Quality Factors & Metrics

As required by the QA Plan, we will measure the project's success using the four quality factors Usability, Correctness, Performance and Maintainability.

3. Test Plan and Bug Tracking

* Test Cases: The full QA Plan document will contain detailed test cases for core functionalities.
* Bug Tracking: We will use the built-in "Issues" feature on our GitHub repository to report, track, and manage all bugs found during testing.

## 8. References

*(This section will be populated during the project's development with citations for any external documentation, frameworks, or research used.)*