****

**Software Project Management Plan**

**ST{AU}CKTION**

**Team 10**

**Ahmet Oğuz Ergin, 21903070**

**Ayşe Özge Erkan, 21902107**

**Bertan Özer, 21902475**

**Damla Köklü, 22003189**

**Göktuğ Yeşilyurt, 22003697**

Bilkent University

Department of Information Systems and Technologies

07.12.2024

Change History

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **File Name** | **Document Type** | **Deliverable** | **Version** | **Submission Date** |
| Team10\_SPMP\_FR1.docx | MS Word 2024 | 3 | 1 | 07.12.2024 |
| Team10\_SPMP\_FR2.docx | MS Word 2024 | 3 | 2 | 08.01.2025 |

Project Team

|  |  |  |
| --- | --- | --- |
| **Name, Surname** | **Student Id Number** | **e-mail** |
| Ahmet Oğuz Ergin | 21903070 | oguz.ergin@ug.bilkent.edu.tr |
| Ayşe Özge Erkan | 21902107 | ozge.erkan@ug.bilkent.edu.tr |
| Bertan Özer | 21902475 | bertan.ozer@ug.bilkent.edu.tr |
| Damla Köklü | 22003189 | damla.koklu@ug.bilkent.edu.tr |
| Göktuğ Yeşilyurt | 22003697 | goktug.yesilyurt@ug.bilkent.edu.tr |

Project Details

|  |  |
| --- | --- |
| **Project Name** | **ST{AU}CKTION** |
| **Software Name** | **ST{AU}CKTION** |
| **Company Name** | **No Sponsor Company** |
| **Academic Advisor** | **Dr. Cüneyt Sevgi** |
| **Github URL** | **https://github.com/staucktion** |
| **WEB page** |  |

Individual Contributions Overview

|  |  |
| --- | --- |
| **Name, Surname** | **Summary of Contributions to the SPMP Document** |
| Ahmet Oğuz Ergin | Main responsible of the Project Effort Estimation section. |
| Ayşe Özge Erkan | Main responsible of the Scope, Project Monitoring and Measuring sections and also formating. |
| Bertan Özer | Main responsible of the Product Verification and Validation section. |
| Damla Köklü | Main responsible of the Project Schedule section. |
| Göktuğ Yeşilyurt | Main responsible of the Executive Summary and Software Development Environment sections. |

Executive Summary

The Software Project Management Plan (SPMP) of our project shows which key management strategies, resource plannings, and progress tracking methods we are planning to use while developing our project. We used three different ways to estimate the effort needed for development of this project, which are Work Breakdown Structure (WBS), Use Case based estimation, and agile estimation methods. Upon the completion of each estimation, we came up with the following results: WBS provided a total effort of 2,090 hours by breaking tasks into smaller and manageable components. On the other hand, Use Case based estimation resulted in 2,522.52 person-hours where we factored in both technical and environmental influences. Finally, in agile estimation we used story points and came up to the result that there would be 4 XS, 7 S, 6 M, 4 L and 5 XL story points in total, measured in t-shirt sizes. Upon comparing results of each methodology, we concluded that Use Case based estimation is the most suitable method for estimating this project, since it considers both the details of the project while accounting for things like Environmental Factors as well. The SPMP also includes a structured project schedule with clear dependencies and deliverables, supported by financial planning using budget estimations and a Net Present Value (NPV) analysis to ensure cost efficiency, which we estimated as 23773.41 ₺ (Turkish Liras) in the end. We decided to track our project's progress via a combination of Kanban boards for task management and bi-weekly sprint reviews to reorganize priorities for the next sprint. We will be using product metrics such as defect density, process metrics such as cycle time, and project metrics such as effort variance to have a more grained overview of our project's progression. For verification and validation process of the project, we will be using a mix of unit, integration, and system testing. The tools will help us with confirming that the code we will be writing is "structurally" correct, and we will further do acceptance testing with our stakeholders, including our Product Owner Cüneyt Sevgi, to validate that the code we write actually meets the requirements. With the first incremental demo of our application, we will be showcasing the Use Cases related with Take Photo and Purchase Photo actions to show the main functionalities of our app while using the tools we have documented in this document.

Table of Contents

[Executive Summary i](#_Toc187195220)

[Table of Contents ii](#_Toc187195221)

[ii ii](#_Toc187195222)

[1 Scope 1](#_Toc187195223)

[2 Project Schedule 1](#_Toc187195224)

[2.1 Work Packages & Dependencies 1](#_Toc187195225)

[2.2 Resource Requirements 19](#_Toc187195226)

[2.3 Cost Estimation, Net Present Value, Budget and Resource Allocation 22](#_Toc187195227)

[3 Project Effort Estimation 23](#_Toc187195228)

[3.1 Work Breakdown Structure (WBS) - Decomposition-Based Estimation 23](#_Toc187195229)

[3.2 Use Case Based Estimation 26](#_Toc187195230)

[3.3 Agile Estimation 32](#_Toc187195231)

[3.4 Discussion 38](#_Toc187195232)

[4 Project Monitoring and Measuring 39](#_Toc187195233)

[5 Product Verification and Validation 42](#_Toc187195234)

[6 Software Development Environment 48](#_Toc187195235)

[6.1 Programming Languages 48](#_Toc187195236)

[6.2 Web User Interface 48](#_Toc187195237)

[6.3 Database 48](#_Toc187195238)

[6.4 APIs 48](#_Toc187195239)

[6.5 Cloud Services 48](#_Toc187195240)

[6.6 Web Server 48](#_Toc187195241)

[6.7 Verification and Validation 49](#_Toc187195242)

[6.8 Version Control 49](#_Toc187195243)

[7 Discussions 52](#_Toc187195244)

[7.1 Limitations and Constraints 52](#_Toc187195245)

[7.2 Health and Safety Issues 52](#_Toc187195246)

[7.3 Legal Issues 52](#_Toc187195247)

[7.4 Economic Issues and Constraints 52](#_Toc187195248)

[7.5 Sustainability 52](#_Toc187195249)

[7.6 Ethical Issues 52](#_Toc187195250)

[7.7 Multidisciplinary Collaboration 52](#_Toc187195251)

[8 References 53](#_Toc187195252)

List of Tables

[Table 1 Work Breakdown Structure 1](#_Toc187195305)

[Table 2 Fix Monthly expenses in TL 20](#_Toc187195306)

[Table 3 Fix Monthly expenses in Dollar 20](#_Toc187195307)

[Table 4 Cash outflow in TL 20](#_Toc187195308)

[Table 5 Cash outflow in Dollar 21](#_Toc187195309)

[Table 6 Unadjusted Actor Weighting Table 27](#_Toc187195310)

[Table 7 Transaction Counting 28](#_Toc187195311)

[Table 8 Unadjusted Use Case Weighting Table 29](#_Toc187195312)

[Table 9 Technical Complexity Factors 30](#_Toc187195313)

[Table 10 Environmental Factors 31](#_Toc187195314)

[Table 11 Correlation of User Stories and Story Points 36](#_Toc187195315)

[Table 12 distribution of story points 37](#_Toc187195316)

[Table 13 Tools and Frameworks to be Used with Their Versions 49](#_Toc187195317)

List of Figures

[Figure 1 Expenses and Total NPV Diagram with USD Expenses 22](#_Toc187195332)

[Figure 2 Updated NPV Diagram 23](#_Toc187195333)

[Figure 3 Use Case Diagram 26](#_Toc187195334)

Abbreviations

|  |  |
| --- | --- |
| API | Application Programming Interface |
| AWS | Amazon Web Services |
| EFactor | Environmental Factor Value |
| EP | Epic |
| FME | Fix Monthly Expense |
| N/A | Not Applicable |
| NPV  PH | Net Present Value  Person Hours |
| PHM | Person Hours Multiplier |
| SDK | Software Development Kit |
| SE | Confirmation Actions |
| SPMP | Software Project Management Plan |
| SR | System Response |
| SV | System Actions |
| TCF | Technical Complexity Factors |
| TFactor | Technical Factor Value |
| TL | Turkish Liras |
| U | User Action |
| UAW | Unadjusted Actor Weight |
| UCP | Use Case Points |
| UI | User Interface |
| US | User Story |
| UUCP | Unadjusted Use Case Points |
| UUCW | Unadjusted Use Case Weight |
| WBS | Work Breakdown Structure |
| WP | Work Package |

# Scope

The Software Project Management Plan (SPMP) outlines the methodologies, resources, timelines, and tasks essential for managing the development of St{au}cktion. This project involves building a robust platform that enables photographers to upload photos, participate in auctions, and sell their content to stock marketers or regular users. Key functionalities include user management, photo management, location-based features, voting system, bidding and auction systems, and lastly secure transaction handling. The scope of the project management plan activities is as follows.

1. **Planning and Scheduling:** Deliverables and dependencies were divided among group members based on their abilities and experience and written in Work Breakdown Structure.
2. **Effort Estimation:** To ensure precise effort forecasting, three estimation techniques are applied:
3. **Work Breakdown Structure (WBS):** Tasks are broken into manageable sub-tasks and effort allocated based on complexity and priority. Payment and Transactions and Testing and Validation are the most resource-intensive tasks.
4. **Use Case Based Estimation:** Assesses effort based on use case complexity and technical/environmental factors, yielding 2716,56 person-hours.
5. **Agile Estimation:** Uses T-shirt sizing (e.g., XS to XL) to estimate the effort for each user story. Factors such as complexity, risk, and repetition are considered, making it ideal for iterative progress tracking.
6. Monitoring and Metrics: To track progress and ensure timely delivery, the following monitoring strategies and metrics are implemented:
7. **Progress Tracking:** A Kanban board is used to visualize and update task statuses (e.g., "To-Do," "In Progress," "Done") while sprint reviews are conducted bi-weekly to assess progress and adjust priorities.
8. **Key Metrics (Project, Product, Process):**
9. **Planned vs. Actual Task Completion:** Tracks whether deliverables are on schedule.
10. **Defect Density:** Monitors the number of defects per module, ensuring quality control.
11. **Effort Variance:** Compares estimated effort with actual effort to refine future estimations.
12. **Cycle Time:** Measures the time required to complete individual tasks, identifying potential delays.
13. **Verification and Validation:** To ensure product quality and alignment with requirements, comprehensive testing strategies are employed.
14. **Unit Testing:** Focuses on individual modules like user authentication and auction setup to validate functionality.
15. **Integration Testing:** Tests the interaction between system components, such as APIs and the database, to ensure seamless functionality.
16. **System Testing**: Validates the entire system against functional and non-functional requirements to ensure readiness for deployment.
17. **Acceptance Testing:** Engages stakeholders and end-users to verify that the product meets business and user needs.

Testing employs tools like Postman for API validation and SonarQube for static code analysis.

1. **Software Development Environment**: The project utilizes a robust development setup:
2. **APIs and Services:** Google Maps API for location features; simulated banking API for payments.
3. **Cloud Services:** AWS Lambda is employed for computationally intensive tasks like watermarking photos, reducing server load.
4. **Testing Tools:** JUnit for Android unit testing, Mocha for backend testing, and SonarQube for static code analysis.
5. **Team Assignments:** Tasks are distributed based on expertise, optimizing efficiency and reducing delays.

# Project Schedule

## Work Packages & Dependencies

Below Work Breakdown Structure states the deliverables, work packages, and dependencies within the work packages. Duration of each of the work package is also indicated in the hours column of the table.

Table 1 Work Breakdown Structure

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | | | | | | | | | | | |
|  | **Breakdown St{au}cktion** | **Description** | **WBS** | **Code** | **Dependencies** | **Assigned to** | | | | | **PH** |
|  |  |  |  | **1.0** |  |  |  |  |  |  |  |
|  |  | **Initial Plan + Business**  **Model** |  | **1.1** |  |  |  |  |  |  |  |
|  | WP1.1 | Deciding on Internal  Stakeholders (Role  Assessment). | 1.1.1 |  | 0 | Ahmet | Ayşe | Bertan | Damla | Göktuğ | 5ph |
|  | WP1.2 | Meeting with PO (Cüneyt  Sevgi) for creating a use case diagram of St{au}cktion. | 1.1.2 |  | 0 | Ahmet | Ayşe | Bertan | Damla | Göktuğ | 5ph |
|  | WP1.3 | Discussing requirements,  project scope and assessing  risks. | 1.1.3 |  | 0 |  | Ayşe |  |  |  | 3ph |
|  | WP1.4 | Allocating tasks on the Initial  Plan report and Business  Model. | 1.1.4 |  | 0 | Ahmet | Ayşe | Bertan | Damla | Göktuğ | 5ph |
|  | WP1.5 | Creating a GANTT Chart. | 1.1.5 |  | WP1.4,1.3,1.2 |  |  |  | Damla |  | 8ph |
|  | WP1.6 | Creating Work Breakdown Structure. | 1.1.6 |  | WP1.4,1.3,1.2 |  |  |  | Damla |  | 8ph |
|  | WP1.7 | Discussing the Legal, Ethical, Health constraints while preparing this project. | 1.1.7 |  | WP1.6,1.5,1.4,1.3,1.2,1.1 | Ahmet | Ayşe | Bertan | Damla | Göktuğ | 5ph |
|  | WP1.8 | Getting feedback about first draft of the initial plan document. | 1.1.8 |  | WP1.6,1.5,1.4,1.3,1.2 | Ahmet | Ayşe | Bertan | Damla | Göktuğ | 5ph |
|  | WP1.9 | Creating a change list for the initial plan document. | 1.1.9 |  | WP1.6,1.5,1.4,1.3,1 | Ahmet | Ayşe | Bertan | Damla | Göktuğ | 10ph |
|  | WP1.10 | Changing the initial plan’s final document considering those changes. | 1.1.10 |  | WP1.9 | Ahmet | Ayşe | Bertan | Damla | Göktuğ | 20ph |
|  | WP1.11 | Formatting the final document | 1.1.11 |  | WP1.10 |  |  |  | Damla |  | 24ph |
|  |  | **SRS + Requirements Prototype** |  | **1.2** |  |  |  |  | Damla |  |  |
|  | WP2.1 | Choosing the Requirements,  Creating a requirements  prototype. | 1.2.1 |  | Deliverable 1 | Ahmet | Ayşe | Bertan | Damla | Göktuğ | 5ph |
|  | WP2.2 | Allocating tasks on the SRS + Requirements prototype. | 1.2.2 |  | Deliverable 1 | Ahmet | Ayşe | Bertan | Damla | Göktuğ | 4ph |
|  | WP2.3 | Finalizing the expected end software products list with identifying database operations that will be used | 1.2.3 |  | Deliverable 1 |  |  |  | Damla | Göktuğ | 12ph |
|  | W2.P4 | Defining the final version of the functional and non functional requirements. | 1.2.4 |  | Deliverable 1 | Ahmet |  |  |  |  | 8ph |
|  | WP2.5 | Creating the use case models. | 1.2.5 |  | Deliverable 1 | Ahmet |  |  |  |  | 8ph |
|  | WP2.6 | Discussing the two most complex use cases with the product owner. | 1.2.6 |  | WP2.5,2.4 | Ahmet | Ayşe | Bertan | Damla | Göktuğ | 10ph |
|  | WP2.7 | Defining system models. | 1.2.7 |  | WP2.5,2.4 |  | Ayşe |  |  |  | 8ph |
|  | WP2.8 | Identifying user  characteristics. | 1.2.8 |  | WP2.5,2.4 |  | Ayşe |  |  |  | 4ph |
|  | WP2.9 | Getting feedback about first draft of the initial plan document. | 1.2.9 |  | WP2.1,2.2,2.3,2.4,2.5,2.6,2.7 | Ahmet | Ayşe | Bertan | Damla | Göktuğ | 5ph |
|  | WP2.10 | Creating a change list for the SRS document. | 1.2.10 |  | WP2.9 | Ahmet | Ayşe | Bertan | Damla | Göktuğ | 10ph |
|  | WP2.11 | Changing the initial plan’s final document considering those changes. | 1.2.11 |  | WP2.10 | Ahmet | Ayşe | Bertan | Damla | Göktuğ | 20ph |
|  | WP2.12 | Formatting the final document. | 1.2.12 |  | WP2.11 | Ahmet |  |  |  |  | 12ph |
|  |  | **SPMP** |  | **1.3** |  |  |  |  |  |  |  |
|  | WP3.1 | Estimating effort, making a   schedule. | 1.3.1 |  | Deliverable 2 | Ahmet | Ayşe | Bertan | Damla | Göktuğ | 10ph |
|  | WP3.2 | Defining all of the cash flows in the project from start to the end, and calculating the NPV. | 1.3.2 |  | Deliverable1,2 | Ahmet |  |  | Damla |  | 20ph |
|  | WP3.3 | Updating the work-breakdown structure. | 1.3.3 |  | Deliverable 1,2 |  |  |  | Damla |  | 12ph |
|  | WP3.4 | Working on project  monitoring structure and  Measuring. | 1.3.4 |  | Deliverable 2 |  | Ayşe |  |  |  | 5ph |
|  | WP3.5 | Determining final  Development Environment,  and resources. | 1.3.5 |  | Deliverable 2 |  |  |  |  | Göktuğ | 10ph |
|  | WP36 | Identifying the test cases. | 1.3.6 |  | Deliverable 2 |  |  | Bertan |  |  | 8ph |
|  | WP3.7 | Going to the course instructor’s room for getting an idea about the work breakdown structure. | 1.3.7 |  | WP,3.4,3.5,3.6 |  |  |  | Damla |  | 10ph |
|  | WP3.8 | Getting feedback about SRS document. | 1.3.8 |  | WP3.4,3.5,3.6 | Ahmet | Ayşe | Bertan | Damla | Göktuğ | 5ph |
|  | WP3.9 | Creating a change list for the SRS document. | 1.3.9 |  | WP3.8 | Ahmet | Ayşe | Bertan | Damla | Göktuğ | 10ph |
|  | WP3.10 | Changing the SRS final document considering those changes. | 1.3.10 |  | WP3.9 | Ahmet | Ayşe | Bertan | Damla | Göktuğ | 20ph |
|  | WP3.11 | Getting feedback about first draft of the initial plan document. | 1.3.11 |  | WP3.10 | Ahmet | Ayşe | Bertan | Damla | Göktuğ | 5ph |
|  | WP3.12 | Formatting the final document. | 1.3.12 |  | WP3.11 |  | Ayşe |  |  |  | 18ph |
|  |  | **SDD** |  | **1.4** |  |  |  |  |  |  |  |
|  | WP4.1 | Finalizing the Use case diagram and Requirements, using the latest version of them. | 1.4.1 |  | Deliverable 2,3 | Ahmet |  |  |  |  | 8ph |
|  | WP4.2 | Finalizing, or updating software increment descriptions in SPMP. | 1.4.2 |  | Deliverable 3 | Ahmet |  |  |  |  | 5ph |
|  | WP4.3 | Mentioning about the frameworks, APIs, libraries, services, and databases that will be used, and explaining how they will contribute to the project. | 1.4.3 |  | Deliverable 3 |  |  |  |  | Göktuğ | 5ph |
|  | WP4.4 | Making high level architecture design. | 1.4.4 |  | Deliverable 3 |  |  | Bertan | Damla |  | 6ph |
|  | WP4.5 | Making low level design. | 1.4.5 |  | Deliverable 3 |  | Ayşe |  |  |  | 6ph |
|  | WP4.6 | Go over the document for making sure the informations are correct and will be helpful for developing the project. | 1.4.6 |  | WP4.4,4.5 | Ahmet | Ayşe | Bertan | Damla | Göktuğ | 5ph |
|  | WP4.7 | Discuss the document with PO, and project advisor Cüneyt Sevgi for making sure the document is highlighting the coding stage clearly. | 1.4.7 |  | Wp4.4,4.5,4.6 | Ahmet | Ayşe | Bertan | Damla | Göktuğ | 5ph |
|  | WP4.8 | Getting feedback about first draft of the SDD report. | 1.4.8 |  | WP4.1,4.2,4.3,4.4,4.5,4.6,4.7 | Ahmet | Ayşe | Bertan | Damla | Göktuğ | 5ph |
|  | WP4.9 | Creating a change list for SDD’s final document. | 1.4.9 |  | WP4.8 | Ahmet | Ayşe | Bertan | Damla | Göktuğ | 10ph |
|  | WP4.10 | Changing the SDD’s final document considering those changes. | 1.4.10 |  | WP4.9 | Ahmet | Ayşe | Bertan | Damla | Göktuğ | 10ph |
|  | WP4.11 | Formatting the final document. | 1.4.11 |  | WP4.10 |  |  |  |  | Göktuğ | 6ph |
|  |  | **1st Increment Product + Presentation** |  | **1.5** |  |  |  |  |  |  |  |
|  | WP5.1 | Discuss the First Increment  product’s requirements with  the PO. | 1.5.1 |  | 0 | Ahmet | Ayşe | Bertan | Damla | Göktuğ | 5ph |
|  | WP5.2 | Allocating tasks of developers  and lead roles. | 1.5.2 |  | Deliverable 1 | Ahmet | Ayşe | Bertan | Damla | Göktuğ | 10ph |
|  | WP5.3 | Designing, and writing the code of taking the photo operation. | 1.5.3 |  | WP5.2 | Ahmet | Ayşe | Bertan | Damla | Göktuğ | 40ph |
|  | WP5.4 | Designing and writing the codes for storing the taken photos. | 1.5.4 |  | WP5.3 | Ahmet |  |  |  | Göktuğ | 50ph |
|  | WP5.5 | Generating watermark for photos that are validated. | 1.5.5 |  | 0 | Ahmet |  |  |  |  | 30ph |
|  | WP5.6 | Generate another requirements report for the coding stage. | 1.5.6 |  | 0 |  | Ayşe |  | Damla |  | 20ph |
|  | WP5.7 | Testing First Increment  Product. | 1.5.7 |  | WP5.3,5.4,5.5,5.6 |  |  | Bertan |  |  | 10ph |
|  | WP5.8 | Preparing a presentation  about 1st Increment Product. | 1.5.8 |  | WP5.7 | Ahmet | Ayşe | Bertan | Damla | Göktuğ | 50ph |
|  | WP5.9 | Presenting the 1st Increment Product. | 1.5.9 |  | WP5.8 | Ahmet | Ayşe | Bertan | Damla | Göktuğ | 5ph |
|  |  | **1st Sprint** |  | **1.6** |  |  |  |  |  |  |  |
|  | WP6.1 | Designing the user interfaces for the project. | 1.6.1 |  | 0 |  | Ayşe |  | Damla |  | 60ph |
|  | WP6.2 | Creating the host DBMS. | 1.6.2 |  | 0 |  |  |  |  | Göktuğ | 50ph |
|  | WP6.3 | Developing a website as  Prototype. | 1.6.3 |  | 0 | Ahmet | Ayşe | Bertan | Damla | Göktuğ | 70ph |
|  | WP6.4 | Allowing users to register to the system. | 1.6.4 |  | 0 | Ahmet |  |  |  |  | 40ph |
|  | WP6.5 | Allowing users to login and logout to the system. | 1.6.5 |  | 0 | Ahmet |  |  |  |  | 18ph |
|  | WP6.6 | Allowing system admin to manage user information. | 1.6.6 |  | 0 |  |  |  | Damla |  | 20ph |
|  | WP6.7 | Allowing registered users to manage their own information. | 1.6.7 |  | 0 |  |  |  | Damla |  | 20ph |
|  | WP6.8 | Implementing the API for accessing the map information. | 1.6.8 |  | 0 |  |  |  |  | Göktuğ | 50ph |
|  | WP6.9 | Allowing System admin to manage location categories. | 1.6.9 |  | 0 |  | Ayşe |  |  |  | 50ph |
|  | WP6.10 | Managing location information while users are taking photo, and send upload request. | 1.6.10 |  | 0 |  | Ayşe |  |  |  | 50ph |
|  | WP6.11 | Sending the photos to the validating agent for verification before uploading the images. | 1.6.11 |  | 0 |  |  |  |  | Göktuğ | 50ph |
|  | WP6.12 | Meeting with PO (Cüneyt Sevgi) for ensuring the project is going according to the plan. | 1.6.12 |  | 0 | Ahmet | Ayşe | Bertan | Damla | Göktuğ | 30ph |
|  | WP6.13 | Testing, Verification, and validation of the project. | 1.6.13 |  | 0 |  |  | Bertan |  |  | 20ph |
|  |  | **2nd Sprint** |  | **1.7** |  |  |  |  |  |  |  |
|  | WP7.1 | Allowing photographers to to take photos using the system’s user interface, while location is on. | 1.7.1 |  | 0 |  | Ayşe |  |  |  | 40ph |
|  | WP7.2 | Allowing validating agent to approve the location category requests. | 1.7.2 |  | 0 |  |  |  |  | Göktuğ | 40ph |
|  | WP7.3 | Allowing users to request a new location category. | 1.7.3 |  | 0 |  |  | Bertan |  |  | 40ph |
|  | WP7.4 | Using Google API for the application’s map that includes the location categories. | 1.7.4 |  | 0 |  |  |  |  | Göktuğ | 30ph |
|  | WP7.5 | Allowing registered users to vote for the photos in the relevant location categories. | 1.7.5 |  | 0 |  |  |  | Damla |  | 30ph |
|  | WP7.6 | Allowing photographers to update, delete and manage their photos. | 1.7.6 |  | 0 |  |  |  | Damla |  | 40ph |
|  | WP7.7 | Meeting with PO (Cüneyt Sevgi) for ensuring the project is going according to the plan. | 1.7.7 |  | 0 | Ahmet | Ayşe | Bertan | Damla | Göktuğ | 30ph |
|  | WP7.8 | Working on the mobile side of the application. | 1.7.8 |  | 0 | Ahmet | Ayşe | Bertan | Damla | Göktuğ | 100ph |
|  | WP7.9 | Creating the simulated banking application. | 1.7.9 |  | 0 |  | Ayşe |  | Damla |  | 40ph |
|  | WP7.10 | Testing, Verification, and validation of the Simulated banking application. | 1.7.10 |  | 0 |  |  | Bertan |  |  | 30ph |
|  | WP7.11 | Testing, Verification, and validation of the project. | 1.7.11 |  | 0 |  |  | Bertan |  |  | 20ph |
|  |  | **3rd Sprint** |  | **1.8** |  |  |  |  |  |  |  |
|  | WP8.1 | Creating a auction functionality for images that are voted the highest. | 1.8.1 |  | 0 |  |  |  |  | Göktuğ | 50ph |
|  | WP8.2 | Selling photos to the person who put the highest bid. | 1.8.2 |  | 0 |  |  |  |  | Göktuğ | 20ph |
|  | WP8.3 | Allowing application to manage bids. | 1.8.3 |  | 0 |  |  |  |  | Göktuğ | 20ph |
|  | WP8.4 | Putting a timer for the auction. | 1.8.4 |  | 0 | Ahmet |  |  |  |  | 30ph |
|  | WP8.5 | Enabling the selling option for images that are not selected for the auction. | 1.8.5 |  | 0 |  |  |  | Damla |  | 40ph |
|  | WP8.6 | Testing of the bidding functionality. | 1.8.6 |  | 0 |  |  | Bertan |  |  | 30ph |
|  | WP8.7 | Testing, Verification, and validation of the Entire project. | 1.8.7 |  | 0 | Ahmet | Ayşe | Bertan | Damla | Göktuğ | 200ph |
|  | WP8.8 | Getting ready for the Jury. Presentation and full code coverage. | 1.8.8 |  | Deliverable 5,6,7 | Ahmet | Ayşe | Bertan | Damla | Göktuğ | 100ph |
|  | WP8.9 | Buying some sweat treats for people visiting the project presentation. | 1.8.9 |  | WP8.8 | Ahmet | Ayşe | Bertan | Damla | Göktuğ | 5ph |
|  | WP8.10 | Proposing the project to the Jury and the PO (Cüneyt Sevgi). | 1.8.10 |  | WP8.10 | Ahmet | Ayşe | Bertan | Damla | Göktuğ | 60ph |

## Resource Requirements

The resources that are needed in this project are given in the below tables with their corresponding costs. In below tables, the cash outflows in Turkish Liras, and US Dollars are shown. By looking at the cost of total resources, FMEs are calculated. The tables of fix monthly expenses, show the fix minimal cost for each month, and the tables of cash outflow show the overall cost, and expected cost of the project broken down into months with additional costs.

Table 2 Fix Monthly expenses in TL

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Resources** | **Fuel for face to face meetings, lectures** | **Electricity bill for each of the group members** | **Coffee before each of the meeting** | **Total resources** |
| **Cost** | 1.000,00TL | 500,00TL | 800,00TL | 2.300,00TL |

Table 3 Fix Monthly expenses in Dollar

|  |  |  |  |
| --- | --- | --- | --- |
| **Resources** | **Chatgpt 0.4** | **Office 365 Subscription** | **Total resources** |
| **Cost** | $20,00 | $9,00 | $29,00 |

Table 4 Cash outflow in TL

|  |  |  |
| --- | --- | --- |
|  |  |  |
| **Month** | **Expenses in TL** | **Description** |
| **Sep-24** | 2.300,00TL | FME |
| **Oct-24** | 4.300,00TL | FME + Psychological help for one of the group members due to heavy workload. |
| **Nov-24** | 2.300,00TL | FME |
| **Dec-24** | 2.700,00TL | FME + extra internet due to long meeting hours. |
| **Jan-25** | 2.500,00TL | FME + muscle relaxant, due to a sudden shock a team member had in one of the 411 Course. |
| **Feb-25** | 2.300,00TL | FME |
| **Mar-25** | 2.300,00TL | FME |
| **Apr-25** | 4.300,00TL | FME + Shirts for presentation of the project. |
| **May-25** | 2.300,00TL | FME |
| **Jun-25** | 4.300,00TL | FME + celebration dinner for the project |

Table 5 Cash outflow in Dollar

|  |  |  |
| --- | --- | --- |
| **Month** | **Expenses in Dollar** | **Description** |
| **September 2024** | $29,00 | FME |
| **October 2024** | $29,00 | FME |
| **November 2024** | $29,00 | FME |
| **December 2024** | $29,00 | FME |
| **January 2025** | $29,00 | FME |
| **February 2025** | $29,00 | FME |
| **March 2025** | $29,00 | FME |
| **April 2025** | $29,00 | FME |
| **May 2025** | $29,00 | FME |
| **June 2025** | $29,00 | FME |
| **July 2025** | $29,00 | FME |

## Cost Estimation, Net Present Value, Budget and Resource Allocation

According to the Resource requirements, below diagram displays both cashflow of dollars and Turkish Liras. This states the projects main cash outflow within each month, starting from September 2024, ending June 2025. Total NPV for this project, ending in June is 23773.41, as shown in the below figure.

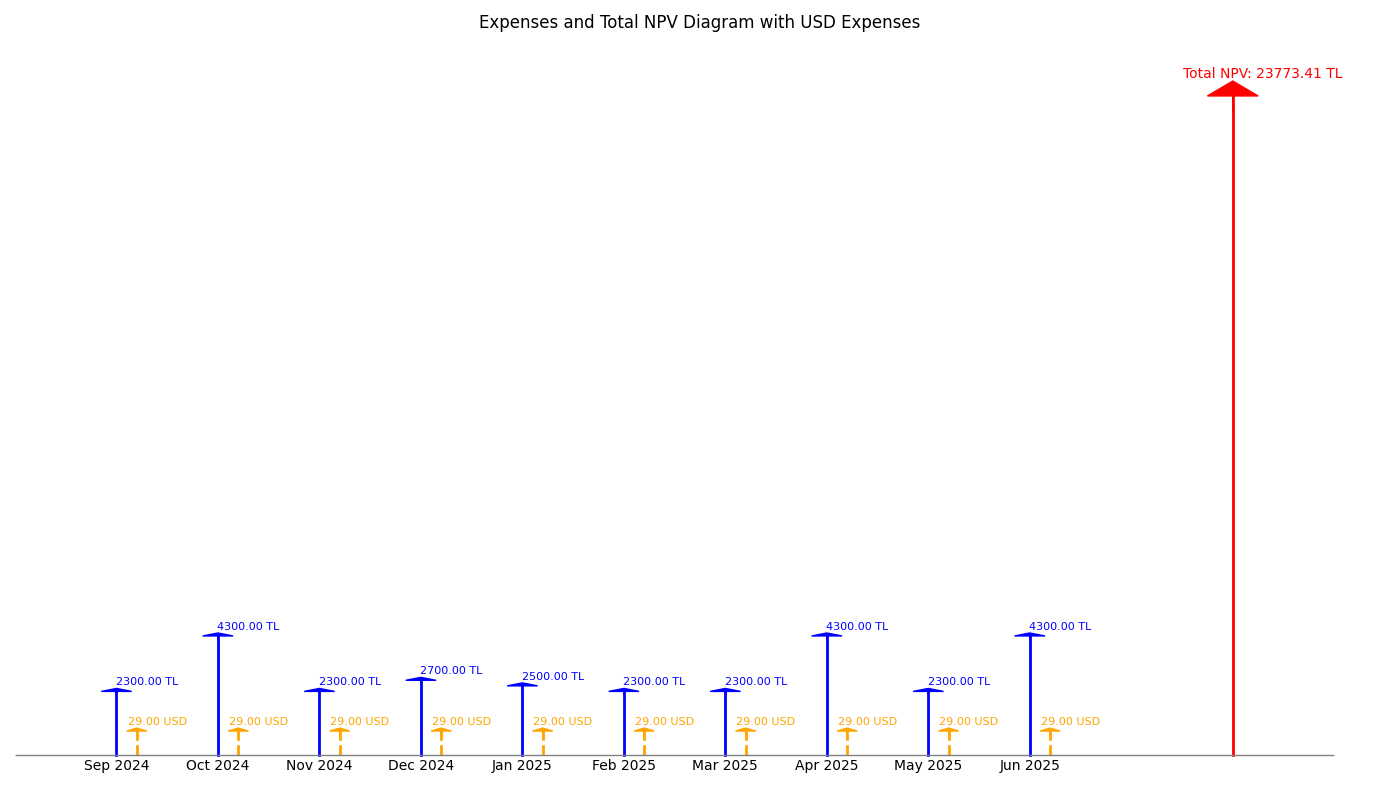


Figure 1 Expenses and Total NPV Diagram with USD Expenses

By calculating the NPV of dollars and Turkish Liras separately for each month, by calculating the NPV of each month is calculated, and displayed within two separate tables in the below.

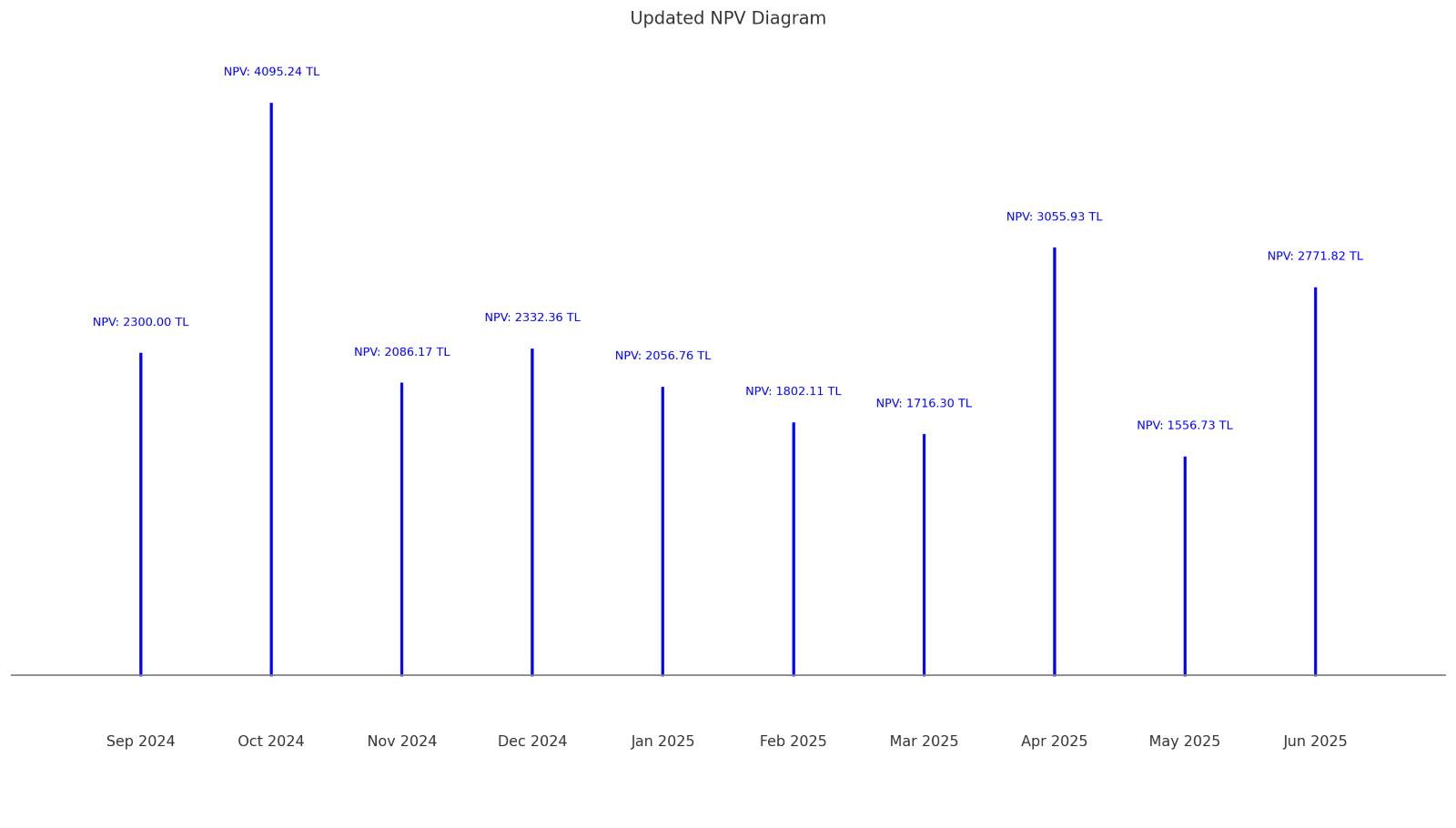


Figure 2 Updated NPV Diagram

By looking at the NPV analysis of the above diagrams, some comments can be made. Over time, the value of TL expenses decreases due to the 5% monthly discount rate. This reflects how the purchasing power of money decreases over time. For example, TL costs in the future (July 2025) are worth significantly less today, so their impact on the project is reduced when calculated in today's terms. Monthly interest rate of dollar is a lot less than the Turkish Liras, therefore Dollars impact to the project is higher, as its value is preserved more than the Turkish liras.

# Project Effort Estimation

## Work Breakdown Structure (WBS) - Decomposition-Based Estimation

The Work Breakdown Structure (WBS) decomposes the system into manageable components and tasks. This approach provides a clear understanding of the project scope, identifying both functional modules and the supporting software engineering processes. Each component is assigned an effort estimate based on its complexity and importance to the overall system.[[1]](#footnote-1)

**Product Functionalities**

1. **User Management**
2. User Registration and Authentication
3. Role-based access for photographers, stock companies, and system admin
4. Effort: 80 ph
5. **Photo Management**
6. Uploading, editing, deleting, and managing photos
7. Applying watermarks to uploaded photos
8. Validation by agents
9. Effort: 120 ph
10. **Location-Based Features**
11. Map integration using Google Maps API
12. Categorizing photos based on locations
13. Location category management (user requests and admin approvals)
14. Effort: 300 ph
15. **Voting System**
16. Voting functionality for users on location-based photos
17. Reward system for voters based on final bids
18. Effort: 150 ph
19. **Auction and Bidding**
20. Auction setup for top-voted photos
21. Timer for auctions and bid management
22. Direct sales for photos not included in auctions
23. Effort:  300 ph
24. **Payment and Transactions**
25. Integration of a simulated banking application for transactions
26. Effort: 400 ph

**Supporting Software Engineering Processes**

1. **Requirements Engineering**
   1. Requirements documentation
   2. Effort: 80 ph
2. **High-Level and Low-Level Design**
   1. Architectural design for system components
   2. Detailed design for critical modules (e.g., photo management, auction system)
   3. Effort: 80 ph
3. **Testing and Validation**
   1. Unit and integration testing for core functionalities
   2. Validation of system features with the Product Owner and advisor
   3. Effort: 400 ph
4. **Documentation**
   1. Preparing Software Design Document (SDD)
   2. Generating a change list and finalizing documentation
   3. Effort: 80 ph
5. **Presentations and Feedback**
   1. Preparing presentations for increments and sprints
   2. Discussing and incorporating feedback from Product Owner and jury
   3. Effort: 100 ph

Decomposition based estimation approach not only ensures that all aspects of the project are covered but also facilitates better management of time, resources, and milestones. With a total estimated effort of 2090 ph, the WBS serves as a vital tool for guiding the project toward successful completion, ensuring that all components are addressed in a balanced and efficient manner.[[2]](#footnote-2)

## Use Case Based Estimation

Use case-based estimation evaluates project effort by considering actors, use case transactions, technical factors, and environmental factors. Below is a detailed breakdown of the estimation process:

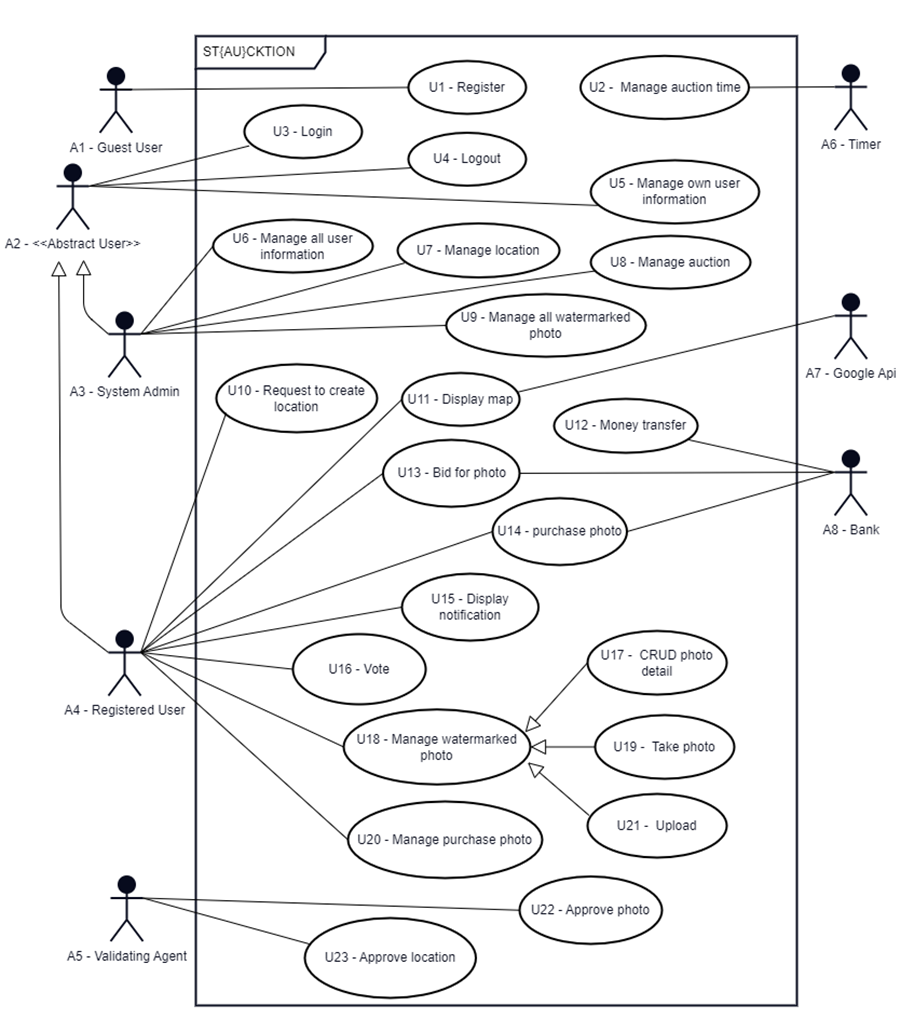


Figure 3 Use Case Diagram

Step 1: Unadjusted Actor Weight (UAW) Calculation

The Unadjusted Actor Weight (UAW) was calculated as 20 using the following table:

Table 6 Unadjusted Actor Weighting Table

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Actor Type | Description | Weighting Factor | Number | Result |
| Simple | External System with well-defined API | 1 | 1 | 1 |
| Average | External System using protocol-based interface | 2 | 2 | 4 |
| Complex | Human | 3 | 5 | 15 |
| Unadjusted Actor Weight Total (UAW) | | | | 20 |

Step 2: Use Case Transaction Counting

The number of transactions was determined based on the definition by Jacobson, which considers the following:

* User action (U): A request from the actor.
* System actions: Including data validation (SV), explicit operations (SE), and responses (SR).

Transactions are counted as a single unit from the actor's request to the system's response, with intermediate steps not being counted separately. For example, in the "Register" use case:

* User provides details (U).
* System validates details (SV).
* System creates a new user (SE).
* System confirms registration (SR).

This flow is counted as one transaction. Below is a summary of transaction counts for all use cases:

Table 7 Transaction Counting

|  |  |  |
| --- | --- | --- |
| Use Case ID | Use Case Name | Transaction count |
| 1 | Register | 1 |
| 2 | Manage auction time | 2 |
| 3 | Login | 1 |
| 4 | Logout | 1 |
| 5 | Manage own user information | 2 |
| 6 | Manage all user information | 2 |
| 7 | Manage location | 2 |
| 8 | Manage auction | 2 |
| 9 | Manage all watermarked photo | 2 |
| 10 | Request to create location | 1 |
| 11 | Display map | 2 |
| 12 | Money transfer | 1 |
| 13 | Bid for photo | 1 |
| 14 | Purchase photo | 1 |
| 15 | Display notification | 1 |
| 16 | Vote | 2 |
| 17 | CRUD photo detail | 3 |
| 18 | Manage watermarked photo | 5 |
| 19 | Take photo | 1 |
| 20 | Manage purchase photo | 2 |
| 21 | Upload | 2 |
| 22 | Approve photo | 1 |
| 23 | Approve location | 1 |

Step 3: Unadjusted Use Case Weight (UUCW) Calculation

The Unadjusted Use Case Weight (UUCW) was calculated as 110 based on the following table:

Table 8 Unadjusted Use Case Weighting Table

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Use-Case Type | Description | Weighting Factor | Number | Result |
| Simple | 1-3 transactions | 5 | 22 | 110 |
| Average | 4-7 transactions | 10 | 1 | 10 |
| Complex | >7 transactions | 15 | 0 | 0 |
| Unadjusted Use-Case Weight Total (UUCW) | | | | 120 |

Step 4: Calculating UUCP and Technical Complexity Factors

Unadjusted Use Case Points (UUCP): 140

Technical Complexity Factors (TCF): Derived from the table below as: 0.99

Table 9 Technical Complexity Factors

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Factor Number | Description | Weight | Assigned Value (0-5) | Weighted Value |
| T1 | Distributed System | 2.0 | 0 | 0 |
| T2 | Response time or throughput performance objectives | 1.0 | 2 | 2 |
| T3 | End-user online efficiency | 1.0 | 3 | 3 |
| T4 | Complex internal processing | 1.0 | 0 | 0 |
| T5 | Reusability of code | 1.0 | 5 | 5 |
| T6 | Ease of installation | 0.5 | 5 | 2.5 |
| T7 | Ease of use | 0.5 | 3 | 1.5 |
| T8 | Portability | 2.0 | 3 | 6 |
| T9 | Ease of change | 1.0 | 5 | 5 |
| T10 | Concurrency | 1.0 | 5 | 5 |
| T11 | Special security objectives included | 1.0 | 4 | 4 |
| T12 | Direct access for third parties | 1.0 | 5 | 5 |
| T13 | Special user training required | 1.0 | 0 | 0 |
| Technical Factor Value (TFactor) | | | | 39 |

Step 5: Environmental Factors and Adjustment

The Environmental Factor Value (EFactor) is 14, calculated from the table below:

Table 10 Environmental Factors

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Factor Number | Description | Weight | Assigned Value (0-5) | Weighted Value |
| E1 | Familiarity with system development process being used | 1.5 | 3 | 4.5 |
| E2 | Application experience | 0.5 | 3 | 1.5 |
| E3 | Object-oriented experience | 1.0 | 2 | 2 |
| E4 | Lead analyst capability | 0.5 | 2 | 1 |
| E5 | Motivation | 1.0 | 5 | 4 |
| E6 | Requirements stability | 2.0 | 3 | 6 |
| E7 | Part time staff | -1.0 | 3 | -3 |
| E8 | Difficulty of programming language | -1.0 | 2 | -2 |
| Environmental Factor Value (EFactor) | | | | 14 |

Final Calculations

Adjusted Use Case Points (UCP): 135.828

Between E1 and E6, there are two factors (E3 and E4) that are less than 3. Additionally, between E7 and E8, there are no factors greater than 3. As a result, the total sum of the EFactors is 2, leading to the determination of the Person Hours Multiplier (PHM) as 20.

Effort (in person-hours): 2716,56

## Agile Estimation

Although not mandatory, epics and user stories have been specified because, particularly in the analysis and design phases of a project adopting agile methodologies, evaluating project functionalities from different perspectives is crucial. Below are the defined epics and user stories:

1. **Epics of User Stories**

**EP-1:** Authentication and authorization

**EP-2:** User management

**EP-3:** Location management

**EP-4:** Photo management

**EP-5:** Auction management

**EP-6:** Purchase management

**EP-7:** Notification management

1. **User Stories**

**EP1-US1:** As a guest user, I want to register to system so that I will use the system functionalities which registered users can perform.

**EP1-US2:** As a registered user, I want to login to a system so that I can authenticate and perform functionalities which need authorization.

**EP1-US3:** As a registered user, I want to logout so that I can either login with a different account, register a new one, or simply end my session.

**EP2-US1:** As a system admin, I want to manage all user information so that I can ensure accuracy and enforce policies, such as removing inappropriate information.

**EP2-US2:** As a registered user, I want to manage my own information so that I can correct any outdated or incorrectly entered details.

**EP3-US1:** As a system admin, I want to manage location categories so that I can ensure accuracy and enforce policies, such as removing inappropriate or inaccurate information.

**EP3-US2**: As a validating agent, I want to approve location category requests so that the new location category will be accurate and appropriate.

**EP3-US3:** As a registered user, I want to make a request to create a new location category for a photo upload so that I will be able to upload photos for that category which does not exist.

**EP3-US4:** As a registered user, I want to display a map and click on location category markers on the map so that I can view and vote on the watermarked photos.

**EP4-US1:** As a system admin, I want to manage all watermarked photos so that I can ensure proper control, monitoring, and enforcement of policies regarding their display and sale.

**EP4-US2:** As a validating agent, I want to approve photos so that they can be included in the auction and made available for stock image companies to bid on.

**EP4-US3:** As a photographer, I want to take a photo using the system's user interface (instead of the default camera application) while GPS is on so that the photo's location cannot be mimicked.

**EP4-US4:** As a photographer, I want to upload a photo to the system so that the system can place it in the auction with a watermark.

**EP4-US5:** As a photographer, I want to manage my uploaded photos so that I can update, delete, or review them as needed before they are included in the auction.

**EP4-US6**: As a registered user, I want to display watermarked photos using a location filter so that I can view photos based on specific geographic areas.

**EP4-US7:** As a photographer, I want to vote for the watermarked photo of other users uploaded so that I would earn money if it the photo selected in the auction.

**EP5-US1**: As a timer, I want to start and end the auction at specified times so that the auction process is effectively managed.

**EP5-US2:** As a system admin, I want to manage auction so that I can ensure accuracy and enforce policies, such as terminating auction with inappropriate situation.

**EP5-US3:** As a stock image company, I want to bid for a watermarked photo so that I can purchase it at the end of the auction.

**EP5-US4:** As a stock image company, I want to purchase a photo so that I receive a watermark-free version with full rights.

**EP5-US5:** As a photographer, I want to receive a 10% commission for voting on photos that make it into the top 10%, so that I am rewarded for my participation in the voting process.

**EP5-US6:** As a system administrator, I want the system to automatically retain 20% commission from the payment of a photo purchase, so that the platform can take its share from successful transactions.

**EP5-US7:** As a photographer, I want to receive 70% of the payment for the sale of my photo, so that I am fairly compensated for my work.

**EP6-US1:** As a photographer, I want to assign a value to my photo even if it is not in the top 5% voted photos, so that stock image companies can still purchase it, allowing me to earn a profit.

**EP6-US2:** As a stock image company, I want to view photos that are not in the top 5% of votes, so that I can evaluate and purchase them as well.

**EP7-US1:** As a registered user, I will display system notifications so that I can display important information such as auction result gains.

1. **Correlation of User Stories and User Points**

When assigning points to user stories, factors such as risk, complexity, and repetition are considered. Points are assigned using a T-shirt sizing scale (XS, S, M, L, XL). Since there is no strict rule for assigning these points, examples of user stories with different point values are provided:

**M User Story:**

EP3-US3 is classified as M. It is not XL because this functionality is not the backbone of the project, and there are alternative ways to handle location requests. Additionally, it is less complex compared to other user stories that involve more internal and external interactions. It is not XS because, despite the alternatives, the location functionality is necessity and cannot be completely excluded.

**XS User Story:**

EP2-US1 falls under the XS category. It is not M because it involves repetition. Registered users will already manage their information, so the implementation is expected to be similar, with minimal additional effort required. Even if this user story is not implemented, the overall system functionality will not be significantly impacted.

**XL User Story:**

EP5-US3 is categorized as XL. It is not M because the auction functionality is a core component of the project. Without the auction and bidding system, the project would lose its innovative aspect, making it irreplaceable. The bidding flow is complex, requiring both prerequisites (e.g., verifying the balance of the stock company account before bidding) and post-requisites (e.g., managing the states of the bidding system). Additionally, the bidding process involves multiple internal and external actors.

The complete correlation between user stories and story points is shown below:

Table 11 Correlation of User Stories and Story Points

|  |  |
| --- | --- |
| User Story ID | Story Point |
| EP1-US1 | S |
| EP1-US2 | S |
| EP1-US3 | XS |
| EP2-US1 | XS |
| EP2-US2 | S |
| EP3-US1 | XS |
| EP3-US2 | M |
| EP3-US3 | M |
| EP3-US4 | XL |
| EP4-US1 | S |
| EP4-US2 | M |
| EP4-US3 | XL |
| EP4-US4 | L |
| EP4-US5 | L |
| EP4-US6 | M |
| EP4-US7 | XL |
| EP5-US1 | S |
| EP5-US2 | M |
| EP5-US3 | XL |
| EP5-US4 | L |
| EP5-US5 | L |
| EP5-US6 | M |
| EP5-US7 | S |
| EP6-US1 | XL |
| EP6-US2 | S |
| EP7-US1 | XS |

The distribution of story points is as follows:

Table 12 distribution of story points

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Size (T-shirt) | Size (Numeric) | Number of story points | | Result |
| XS | 1 | 4 | | 4 |
| S | 3 | 7 | | 21 |
| M | 5 | 6 | | 30 |
| L | 8 | 4 | | 32 |
| XL | 13 | 5 | | 65 |
| Total | | | 152 | |
|  |  |  |  |  |

The total numeric value of story points is 152. Assuming that 1 story point corresponds to 15 ph, the project will require a total of 2280 ph.

## Discussion

The effort estimations from WBS, Use Case-Based Estimation, and Agile Estimation differ due to their distinct approaches:

WBS focuses on breaking tasks into smaller components, providing detailed estimates. However, it does not account for environmental or technical factors, limiting its overall practicality. The estimated effort for WBS is 2090 person-hours (ph).

Use Case-Based Estimation evaluates effort based on user interactions while also considering environmental and technical factors. This broader perspective ensures a more comprehensive and realistic estimate. The estimated effort for Use Case-Based Estimation is 2716 ph, the highest among the three methods.

Agile Estimation uses story points and is flexible, adapting to iterative changes. However, it lacks precise time forecasting before the implementation phase, making it less suitable for early planning. The estimated effort for Agile Estimation is 2280 ph.

With the person-hour comparison across the three methods showing no significant differences, Use Case-Based Estimation emerges as the most suitable and realistic approach for this project, effectively balancing detail and broader considerations.

# Project Monitoring and Measuring

Effective monitoring is a crucial process for project success. To keep our project on track and aligned with objectives, we will use a hybrid approach that combines elements of the Pure Waterfall, Scrum, Kanban, and V-Model methodologies.  

**Monitoring Approach:**

* **Weekly Zoom Meetings:** These will serve as a platform for reviewing completed tasks, discussing ongoing activities, addressing blockers, and adjusting schedules as necessary. These meetings foster team alignment and ensure timely course corrections.
* **Digital Kanban Board:** Tools such as Jira or Trello will be used to track task progress with clear statuses like "To-Do," "In Progress," and "Done." Updates will be made daily, enabling a real-time view of progress and obstacles.
* **Bi-weekly Sprint Reviews:** Completed work will be assessed, deliverables validated, and priorities reorganized for the upcoming sprint.
* **Automated Notifications:** Project management tools will send alerts for overdue tasks and upcoming deadlines, maintaining team awareness and minimizing delays.
* **Stakeholder Meetings:** Bi-monthly Zoom meetings with the product owner, Cüneyt Sevgi, will ensure alignment with project objectives, facilitate feedback, and highlight potential delays or risks early.

This multi-faceted monitoring strategy ensures all stakeholders are informed, obstacles are addressed promptly, and the project remains aligned with its timeline and deliverables.

**Metrics Collection**

To monitor progress, quality, and efficiency, we have identified project, product, and process metrics tailored specifically for our project. Below, each metric is explained with its purpose, measurement method, and how it applies to our software development efforts.

* 1. **Project Metrics:**
     1. **Planned vs. Actual Task Completion:**
        1. **Definition:** Tracks the number of completed tasks compared to planned tasks for each sprint.
        2. **Purpose:** To identify schedule deviations and adjust priorities.
        3. **Measurement:** Weekly data collected from the Kanban board (e.g., Jira/Trello) by comparing tasks marked as "Done" against sprint goals.
        4. **Application to Our Project**: Helps us assess whether the team is on track with implementing core functionalities such as the auction system or photo management.
     2. **Effort Variance:**
        1. **Definition**: Compares the estimated effort (in story points) with the actual effort spent on tasks.
        2. **Purpose**: To improve estimation accuracy for future sprints and optimize resource allocation.
        3. **Measurement**: Collected at the end of every sprint by analyzing discrepancies between planned and actual story points in project management tools.
        4. **Application to Our Project**: Used to refine sprint planning for features like Google Maps integration and auction bidding.
  2. **Product Metrics:**
     1. **Customer Story Coverage:**
        1. **Definition**: Tracks the percentage of user stories implemented compared to the total planned.
        2. **Purpose**: To ensure that the product aligns with customer requirements and core functionalities.
        3. **Measurement**: Collected at the end of each sprint by analyzing completed user stories in the backlog.
        4. **Application to Our Project**: Ensures the implementation of high-priority features like the auction system and photo management while tracking unimplemented stories for future increments.
  3. **Process Metrics:**
     1. **Cycle Time:**
        1. **Definition**: Measures the time taken to complete individual tasks from the moment they are started to when they are marked as "Done."
        2. **Purpose**: To identify bottlenecks and improve task flow.
        3. **Measurement**: Data is tracked automatically in the Kanban board tool for each task, and the average cycle time is calculated weekly.
        4. **Application to Our Project**: Helps monitor delays in critical activities, such as integrating APIs or setting up the auction system.
     2. **Test Pass Rate:**
        1. **Definition**: Measures the percentage of test cases passed during each sprint.
        2. **Purpose**: To gauge testing effectiveness and product stability.
        3. **Measurement**: Automated testing tools (e.g., JUnit, Mocha) record the ratio of passed to total test cases after each testing phase.
        4. **Application to Our Project**: Focuses on critical functionalities, such as validating auction workflows and user authentication. Testing for non-essential features may be deferred until later increments.

# Product Verification and Validation

For the verification and validation stages of the project we will be using different techniques to validate that we are building the right product and verify that we are building the product correctly. For the verification process techniques like unit testing and integration testing will be conducted alongside code inspections, and static analysis of the code. While for the validation process system testing, and acceptance testing will be conducted to validate that the software fulfills its intended purpose.

*Code Inspections:*

With the code inspections, members of the team will be conducting reviews on the code written by their teammates. This way we can give feedback to each other on missing requirements, ways to make the code shorter and easier to read and find defects that the writer of the code may have missed. Code inspections are a key technique to avoid future problems in the development life cycle and will help the programming of the project go smoothly. Reviews on the design (design review) and the requirements (requirement review) will be done with our product owner, and project advisor, Cüneyt Sevgi to make sure the project is on the right track.

**Test Case ID:** CI001

**Test Case Description:** Ensure proper use of error handling and exception management.

**Test Date:** 2/2/25

**Expected Results:** Error and exception cases are handled optimally with appropriate logging and user feedback.

**Actual Results:** There is a function which gets an SQLSyntaxErrorException error

**Status (Pass/Fail):** Fail

**Additional Notes:** Verify that no unhandled exceptions are present.

*Static Analysis:*

With static analysis the code will be analyzed by automated tools without running the program to check for incorrect or inefficient code. SonarQube, an open-source platform for automated inspection of code quality, will perform static analysis of code to detect bugs and code smells. It will be used for static analysis of project components written in languages the platform supports, such as Java, JavaScript, HTML, and CSS.

**Test Case ID:** SA001

**Test Case Description:** Conduct a SonarQube scan for code smells and maintainability issues.

**Test Date:** 2/2/25

**Expected Results:** Code smells are minimized, and maintainability ratings meet acceptable thresholds.

**Actual Results:** Code smells are minimized, and maintainability ratings meet acceptable thresholds.

**Status (Pass/Fail):** Pass

**Additional Notes:** Ensure all identified refactoring suggestions are reviewed and implemented where appropriate.

*Unit Testing:*

Unit testing is a type of testing done to test individual units or components of a software. With unit testing we will be testing isolated portions of the code, or units of the code, to ensure they operate the way they are supposed to, and to check that they fulfill the functional and non-functional requirements of the project. In order to achieve this test cases will be written, team members will then conduct tests using using custom written test scripts or automated testing tools. For the website component of the project, Mocha, a feature-rich JavaScript test framework, will be utilized in test automation for unit testing. For the android application component of the project, Junit, a test automation framework for Java, will be utilized in test automation for unit testing. Espresso, a framework for automated testing of UI components and interactions used in Android application development, will be used for automated unit testing of UI elements in the development of the mobile app.

**Test Case ID:** UT001

**Test Case Description:** Validate user login functionality with valid credentials.

**Test Date:** 2/03/25

**Expected Results:** User successfully logs in and is redirected to their dashboard.

**Actual Results:** User is redirected to their dashboard

**Status (Pass/Fail):** Pass

**Additional Notes:** Test using valid and invalid inputs.

*Integration Testing:*

Integration testing is a form of software testing where modules, or components of the software are tested together as a unified system. It is used to ensure that when combined into a unified system, they communicate and collaborate as intended. This form of testing will be conducted to test the login and registration systems to see if the integration of frontend, backend, and database elements work as intended. On the mobile application side, integration testing will be utilized to check if the mobile app can correctly invoke the camera hardware through Android's APIs, process the captured image, and integrate it into the app's workflows. In addition, it will verify the integration of the GPS functionality, the watermarking function, and the Google Maps API. Postman is an API platform to test the functionality of APIs. It will be utilized to conduct the integration testing of different APIs, REST APIs, Google Maps API, and the simulated banking API used in the project.

**Test Case ID:** IT001

**Test Case Description:** Verify mobile app integration with Google Maps API for photo upload limitation

**Test Date:** 06/4/25

**Expected Results:** After uploading ten photos within a limited vicinity the user will not be able to upload photos from that location.

**Actual Results:** The user was able to upload 12 photos from the limited vicinity

**Status (Pass/Fail):** Fail

**Additional Notes:** Test with different locations and ensure proper coordinates are retrieved.

*System Testing:*

System testing is a form of software testing in which the complete system is tested to ensure it meets all functional and non-functional requirements. Because of its nature system testing will be done near the end of project development when there is a complete iteration of the software to test. It will be conducted manually by team members, and testers who agree to test the system.

**Test Case ID:** ST001

**Test Case Description:** Verify the complete auction process (creation to bid placement).

**Test Date:** 29/4/25

**Expected Results:** Users can create auctions, and others can place bids without issues.

**Actual Results:** The system crashed due to high user count

**Status (Pass/Fail):** Fail

**Additional Notes:** Ensure the system can handle multiple simultaneous users.

*Acceptance Testing:*

Acceptance testing is a form of testing that ensures the software meets the business and end users’ requirements. Similarly to system testing this will be conducted near the end of project development when there is a complete iteration of the software to test. However, it will differ from system testing by focusing on the inputs of end-users and the product owner, who in this instance serves as the baseline for qualifiying if the software meets business necessities. Like with system testing it will be conducted manually, but this time by outside users who qualify for the software’s end users and the product owner, Cüneyt Sevgi.

**Test Case ID:** AT001

**Test Case Description:** Verify that users can participate in an auction from the mobile app.

**Test Date:** 5/5/25

**Expected Results:** User places a bid, and it is reflected in real-time in the auction/reflected in the auction.

**Actual Results:** The bid placed by users is visible in the auction dashboard.

**Status (Pass/Fail):** Pass

**Additional Notes:** Test with different bid amounts and users.

*Schedule*:

*At the end of every SPRINT:*

Techniques: Code inspections, static analysis, and unit testing will be done.

Tools: Code inspections will be done manually and therefore won’t involve tools. SonarQube will be used to perform automated static code analysis to detect bugs, vulnerabilities, and code smells in project components written in Java, JavaScript, HTML, and CSS. Mocha will be used to conduct automated unit testing for the website code. Once the development on the Android application starts, JUnit, a test automation framework for Java will be utilized for automated unit testing, and Espresso will be used for automated unit testing of UI components in the Android application.

Following the 1st SPRINT:

Techniques: Integration testing will be conducted to see if the integration of frontend, backend, and database elements work as intended.

Tools: Postman will be used to test the integration and functionality of REST APIs between the frontend and backend.

Following the 2nd SPRINT:

Techniques: On the website side of the project, integration testing will be conducted on the simulated banking API, and the auction mechanics. For the mobile application, integration testing will be conducted to see if the camera functionality, GPS functionality, the watermarking function, and the Google Maps API integration work as intended. System testing will be conducted for the website.

Tools: Postman will be used to validate API functionality and integration on the website side of the project. The remainder of the integration testing on both the website side and Android application side will be done manually by team members. SonarQube will be used to ensure code quality and detect potential issues during the preparation phase for the website’s system testing. System testing for the website will be done manually by both team members and outside users who will test the system as end-users.

During the 3rd SPRINT:

Techniques: System testing and acceptance testing will be conducted.

Tools: Postman will be utilized to test API related functionalities in the system. Both system testing and acceptance testing will be done manually by team members and end-users.

# Software Development Environment

To develop our project, we selected the following tools, languages and frameworks that we believe would be the best candidates to build the best software development environment that suits our requirements.

## Programming Languages

For programming languages, JavaScript and TypeScript were chosen for the web side of the application; while Java and Kotlin were chosen for the Android application’s coding part. These languages will enable robust and dynamic application development.​[1]

## Web User Interface

The web user interface will be built using React, supported by libraries like Multer for handling file uploads and Sharp for image processing.​​ [1]

## Database

We will use PostgreSQL as the primary database, ensuring secure and scalable data storage for critical application elements such as user profiles, auctions, and transactions. [1]

## APIs

The Google Maps API is the main tool that will be used for location-based functionalities, and we will build our own simulated Banking API to simulate real life banking transactions easily. [1]

## Cloud Services

We are also planning on utilizing AWS Lambda for delegating off the processing power heavy jobs such as watermarking images instead of running it on the main server. [1]

## Web Server

We chose to use Nginx as web server for handling the incoming requests and routing them to their related endpoints. [1]

## Verification and Validation

To ensure our software's reliability and robustness, we will be using JUnit for unit testing of the Android application, and Espresso for UI testing of the Android application. Mocha will be used for testing Node.js based components of the backend code. Postman will be used for API testing and validation of HTTP requests and responses to ensure the backend code is working as intended, while SonarQube will be used for static code analysis to identify possible vulnerabilities and maintain code quality standards.​ [1]

## Version Control

Version control will still be streamlined with GitHub as we are already using it in the documentation phase.​​ [1]

We believe that this setup will be enough as the development environment we will need throughout the following phases of our project while promoting efficiency, scalability, and reliability.[[3]](#footnote-3) The following Table 13 summarizes the tools described above.

Table 13 Tools and Frameworks to be Used with Their Versions

|  |  |  |
| --- | --- | --- |
| **Tool** | **Category** | **Version** |
| JavaScript | Programming Language | ECMAScript 2020 |
| TypeScript | Programming Language | 5.7.2 |
| Java | Programming Language | 21.0.2 |
| Kotlin | Programming Language | 2.0.21 |
| Node.js | Runtime Environment | 22.12.0 |
| React | Framework | 18.3.1 |
| Multer | Library | 1.4.5-lts.1 |
| Sharp | Library | 0.33.5 |
| PostgreSQL | Database Management System | 16.6 |
| Google Maps API | API | N/A (Online service without explicit versioning) |
| Simulated Banking API | API | N/A (Has not been developed yet, so there is no available version) |
| AWS Lambda | Cloud Service | N/A (Online service without explicit versioning) |
| Nginx | Web Server | 1.27.3 |
| GitHub | Version Management Tool | N/A (Online service without explicit versioning) |
| Postman | Verification and Validation Tool | 11.21 |
| JUnit | Unit Testing Framework | 5.11.3 |
| Mocha | JavaScript Testing Framework | 11.0.1 |
| Espresso | UI Testing Tool | 3.6.1 |
| SonarQube | Static Code Analysis Tool | 10.7 |
| Notion | Project Management Tool | N/A (Online service without explicit versioning) |
| Jira | Project Management Tool | N/A (Online service without explicit versioning) |
| Android SDK | Software Development Kit | 35 |
| Android Studio | Integrated Development Environment | 2024.2.1 |

# Discussions

## Limitations and Constraints

Because of dormitory’s last entrance hour, one of our team members had to leave the meetings early.

## Health and Safety Issues

In the mental health extent, group members experienced huge stress for preparing the SPMP before the deadline. Because of cold allergies some team members suffer.

## Legal Issues

We did not experience any legal issues while preparing the SPMP.

## Economic Issues and Constraints

Because of the electricity cut, one of our team members had to buy an extra cellular service.

## Sustainability

We held our meetings on Zoom to avoid in-person gatherings, which would require everyone to use transportation that contributes excess carbon emissions to the environment. We also used digital tools like Notion and GitHub to minimize paper usage by keeping all documents and collaboration online.

## Ethical Issues

For the demonstration of the first increment of our project, we downloaded some photos from the internet without checking the license of the photograph.

## Multidisciplinary Collaboration

We did not collaborate with anyone from other departments, so this section is not applicable for us.

# References

|  |  |
| --- | --- |
| [1] | ​ ​Ergin, Erkan, Köklü, Özer and Yeşilyurt, "Second Version of Software Requirements Specification of Team 10," 2024. |

1. GenAI tool: ChatGPT 4o

   Prompt: Improve the English of the text.

   Rationale: Improve the English  [↑](#footnote-ref-1)
2. GenAI tool: ChatGPT 4o

   Prompt: Improve the English of the text.

   Rationale: Improve the English  [↑](#footnote-ref-2)
3. GenAI tool: ChatGPT 4o

   Prompt: “Please format this text to make it sound professional: [text here]”

   Rationale: To use better tone in the text  [↑](#footnote-ref-3)