Software Engineering- Project Plan, Validation & Testing

# Project Topic (e.g. Online Grocery Store Management System)

Group Members

|  |  |  |
| --- | --- | --- |
| Member Name | Member Surname | Student ID |
| Alican | Kaya | 49429 |
| Ege Bora | Güzel | 58351 |
| İrfan | Bag | 49279 |
| Cenker | Sule | 58566 |
| Efe Berk | Tancı | 59115 |
| Berke | Turk | 58169 |

# Introduction – Project Plan

This document outlines the roadmap for the Hotel Booking and Management Platform project. It defines the timeline, milestones, tasks, and deliverables to ensure team coordination, minimize risks, and clarify responsibilities.

## Overview

* Objective: Develop a web platform that allows guests to search rooms, make reservations, process payments, and cancel bookings, while administrators manage rooms, carousel items, features & facilities, and user inquiries.
* Methodology: Agile Scrum with two-week sprints.

## Project milestones

This table outlines the key milestones of the project, serving as high-level checkpoints to track progress. Each milestone is tied to a specific deliverable, ensuring that critical outputs are completed on time. It provides a clear timeline for stakeholders to monitor major achievements, such as the finalization of requirements or the delivery of the prototype. The planned dates align with the course schedule, reflecting the deadlines for major deliverables.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Milestone ID | Milestone Name | Description | Planned Date | Deliverables |
| M1 | Requirements finalized | Completion of the Software Requirements Specification (SRS) document, defining all functional and non-functional requirements. | Week 2 | SRS Document |
| M2 | UML Diagrams Completed | Creation of all UML diagrams: Use Case, Class, Sequence, State, and Activity diagrams | Week 4 | UML Diagrams (Draw.io XML + PNG) |
| M3 | Core Implementation | Development of reservation flow, payment integration, and admin panel | Week 10 | Beta Release |
| M4 | Testing & Deployment | Unit & integration testing, user acceptance testing (UAT), and production deployment | Week 14 | Final Application + Test Report |

# Task Breakdown (one milestone per group member)

The Task Breakdown table details each task required to complete the milestone, including descriptions, timelines, dependencies, and responsible roles. It ensures that all activities are clearly defined and sequenced, with dependencies reflecting the need for prior tasks (e.g., architecture design depends on requirements). The table assigns deliverables to each task, aligning with the course’s artifact-producing assignments. It serves as a practical guide for the team to manage workload and track progress.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Task ID | Task Name | Duration | Dependencies | Deliverables |
| M1.T1 | Identification of Stakeholders | 1 day | --- | List of stakeholders |
| M1.T2 | Identification of Functional Reuirements | 1 day | M1.T1 | List of functional requirements |
| M1.T3 | Identification of Functional Reuirements | 1 day | M1.T2 | List of non-functional requirements |
|  |  |  |  |  |

### Milestone M1 (Alican Kaya)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Task ID | Task Name | Duration | Dependencies | Deliverables |
| M1.T1 | Identify Stakeholders | 1 day | — | Stakeholder List |
| M1.T2 | Gather Functional Requirements | 2 days | M1.T1 | Functional Requirements Document |
| M1.T3 | Gather Non-functional Requirements | 1 day | M1.T2 | Non-functional Requirements Document |

### Milestone M2 (Ege Bora Güzel)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Task ID | Task Name | Duration | Dependencies | Deliverables |
| M2.T1 | Create Use Case Diagrams | 2 days | M1 deliverables | Use Case Diagram XML + PNG |
| M2.T2 | Create Class & Component Diagrams | 2 days | M2.T1 | Class/Component Diagram XML + PNG |
| M2.T3 | Create Sequence & State Diagrams | 2 days | M2.T2 | Sequence/State Diagram XML + PNG |
| M2.T4 | Create Activity Diagram | 1 day | M2.T3 | Activity Diagram XML + PNG |

### Milestone M3 (İrfan Bağ)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Task ID | Task Name | Duration | Dependencies | Deliverables |
| M3.T1 | Design UI in Figma | 3 days | M2 deliverables | Figma Prototype |
| M3.T2 | Review Prototype with Team | 1 day | M3.T1 | Review Feedback |
| M3.T3 | Refine Prototype | 1 day | M3.T2 | Final Figma Design |

### Milestone M4 (Cenker Şule & Efe Berk Tanrı)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Task ID | Task Name | Duration | Dependencies | Deliverables |
| M4.T1 | Develop Reservation Function | 3 days | M3 | Booking Module |
| M4.T2 | Implement Admin Panel | 3 days | M3 | Admin Module |
| M4.T3 | Integrate Payment Gateway | 2 days | M3 | Payment Module |

### Milestone M5 (Berke Türk)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Task ID | Task Name | Duration | Dependencies | Deliverables |
| M5.T1 | Create Unit Tests | 2 days | M4 | Unit Test Report |
| M5.T2 | Conduct User Acceptance Test | 2 days | M5.T1 | UAT Feedback |
| M5.T3 | Prepare Final Deployment | 1 day | M5.T2 | Deployment Checklist |

# Resource allocation (one milestone per group member)

The Resource Allocation table identifies the human and tool resources assigned to the project, specifying their roles and tasks. It ensures that each task has a designated responsible party, promoting accountability within the team. The availability column helps manage workload, noting whether team members are full-time or part-time. The inclusion of tools like Figma and Lucidchart reflects the practical needs of the project’s deliverables. This table supports efficient resource planning and helps avoid overloading team members.

|  |  |  |  |
| --- | --- | --- | --- |
| Resource Type | Resource Name | Assigned Tasks | Notes |
| Human | Alican Kaya | Milestone M1 | Full-time |
| Human | Ege Bora Güzel | Milestone M2 | Full-time |
| Human | İrfan Bağ | Milestone M3 | Full-time |
| Human | Cenker Şule | M4.T1, M4.T2 | Full-time |
| Human | Efe Berk Tanrı | M4.T3 | Full-time |
| Human | Berke Türk | Milestone M5 | Full-time |
| Tool | Figma | UI/UX Prototype | — |
| Tool | Draw.io | UML Diagrams | — |
| Tool | PHPStorm, MySQL | Application Development | — |

# Risk Management Integration

The Risk Management Integration table links the project plan to the Risk Analysis document, identifying key risks that could affect the schedule, specifying which tasks are most vulnerable.

|  |  |  |  |
| --- | --- | --- | --- |
| Risk ID | Description | Affected Tasks | Mitigation Strategy |
| R1 | Requirement changes during development | M1, M2 | Weekly reviews, change control |
| R2 | Issues with payment API integration | M4.T3 | Early sandbox testing, fallback gateway |
| R3 | Team member unavailability | All milestones | Redistribute tasks, set buffer days |

Validation and Testing

# Introduction – Validation and Testing

This section defines the testing strategies to verify that both functional and non-functional requirements are met.

# Test scope and exclusions

|  |  |  |  |
| --- | --- | --- | --- |
| Component/Feature | In Scope | Out of Scope | Related Requirements |
| User Interface | Navigation, form validation, menu functionality | Visual styling, responsive layout | FR-UI1, NFR-UI2 |
| Booking Functionality | Room search, reservation workflow | Load testing, stress testing | FR-BOOK1, FR-BOOK2 |
| Payment Processing | Payment flow, error handling | Penetration testing | FR-PAY1, NFR-SEC1 |

# Test cases (2 per person in group)

### Alican Kaya

TC1 – User Login

* Requirement: FR-LOGIN
* Precondition: User account registered
* Steps: 1) Enter valid username/password; 2) Click "Login"
* Expected Result: Redirect to user dashboard

TC2 – Search Rooms

* Requirement: FR-BOOK1
* Precondition: System operational with rooms defined
* Steps: 1) Select check-in and check-out dates; 2) Click "Search"
* Expected Result: Available rooms list displayed

### Ege Bora Güzel

TC3 – Make Reservation

* Requirement: FR-BOOK2
* Precondition: Logged-in user
* Steps: 1) Choose a room; 2) Fill reservation form; 3) Click "Reserve"
* Expected Result: Reservation confirmation message shown

TC4 – Cancel Reservation

* Requirement: FR-BOOK3
* Precondition: Active reservation exists
* Steps: 1) Navigate to reservation details; 2) Click "Cancel"
* Expected Result: Reservation status updated to cancelled

### İrfan Bağ

TC5 – Process Payment

* Requirement: FR-PAY1
* Precondition: Reservation in Pending state
* Steps: 1) Enter payment details; 2) Click "Pay"
* Expected Result: Display "Payment Success" notification

TC6 – View User Queries

* Requirement: FR-ADMIN1
* Precondition: At least one user query submitted
* Steps: 1) Login to admin panel; 2) Navigate to "User Queries"
* Expected Result: All submitted queries listed

# Evaluation Criteria (Project Plan)

**1. Timeline and Milestones (2 points)**

* **Completeness (1 point):** The timeline must include all major phases and milestones required to complete the project, covering **at least five key milestones**. Each milestone should have a clear description and a planned date.
* **Realism (1 point):** The timeline and milestones must be realistic, considering the project’s scope, team resources, and deadlines (e.g., aligned with the course schedule ending at Week 14). Unrealistic timelines (e.g., overly aggressive or vague dates) will reduce points.

**2. Task Breakdown and Dependencies (2 points)**

* **Task Identification (1 point):** The plan must list all critical tasks (at least 5) necessary to achieve the milestones, including descriptions, start/end dates, and deliverables (e.g., UML diagrams, prototype).
* **Dependencies and Sequencing (1 point):** Tasks must be logically sequenced with clearly defined dependencies (e.g., "Class Diagram depends on Architecture Design"). Dependencies should align with the workflow established by prior deliverables.

**3. Integration with Previous Deliverables (1 point)**

The plan must reference and incorporate elements from the Requirements Specification (e.g., tasks to meet functional requirements), Risk List (e.g., mitigation tasks), System Architecture Proposal (e.g., design tasks), and UML Diagrams (e.g., modeling tasks). Inconsistencies (e.g., omitting a key deliverable like the prototype) will reduce points.

**4. Clarity and Presentation (1 point)**

* **Organization (0.5 points):** The document must be well-structured, with clear sections (e.g., timeline, task breakdown, resource allocation) and tables for easy navigation. Poor organization (e.g., missing headings, cluttered tables) will reduce points.
* **Clarity (0.5 points):** Descriptions of milestones, tasks, and dependencies must be concise and unambiguous, using consistent terminology aligned with prior deliverables. Vague or overly complex language will result in deductions.

**5. Resource Allocation (1 point)**

* **Resource Identification (0.5 points):** The plan must identify resources (e.g., team members, tools like Figma or Lucidchart) assigned to each task
* **Feasibility (0.5 points):** Resource assignments must be realistic, ensuring no team member is overburdened and tools are appropriate for the tasks (e.g., Figma for prototyping). Overloading resources or missing tool assignments will reduce points.

**6. Timely Submission (1 point)**

# Evaluation Criteria (Testing Plan)

**1. Test Scope and Exclusions - Completeness and Relevance (2 points)**

* **Scope Definition (1 point):** The Test Scope and Exclusions table must clearly identify at least three system components or features to be tested (e.g., user interface, core functionality) and specify what is out of scope (e.g., stress testing, aesthetic design). Each in-scope item should be linked to specific requirements from the Requirements Specification (e.g., FR-1, NFR-2).
* **Relevance and Alignment (1 point):** The scope must align with prior artifacts, such as the Use Case Diagram (for functional testing), Class Diagram (for data components), and Prototype (for UI testing). Exclusions should be justified (e.g., "Penetration testing is out of scope due to limited resources").

**2. High-Level Test Cases - Coverage and Specificity (3 points)**

* **Individual Contribution (2 points):** Each group member must contribute exactly 2 test cases, clearly attributed to them (e.g., labeled with their name or ID). The total number of test cases must match the group size (e.g., 6 test cases for a 3-person group).
* **Test Case Quality (1 point):** Each test case must include a clear test scenario, the requirement tested (e.g., FR-1), preconditions, steps, expected outcome, and priority (e.g., high, medium). Test cases should be specific and testable (e.g., "User enters valid credentials and logs in" rather than "Test login").

**3. Consistency and Integration with Prior Artifacts (1 point)**

Both tables must demonstrate traceability to prior artifacts. The Test Scope table should reference components from the Component Diagram and use cases from the Use Case Diagram, while test cases should map to requirements and prototype features. Lack of traceability (e.g., testing undefined features) will result in deductions.

**4. Clarity and Presentation (1 point)**

* **Table Organization (0.5 points):** Both table and a test-case list must be well-organized, with clear headings, consistent formatting, and logical structure. The Test Scope table should separate in-scope and out-of-scope items clearly, and the Test Cases list should list test cases in a readable format. Disorganized or cluttered tables will lose points.
* **Clarity and Precision (0.5 points):** Descriptions in both tables must be concise, unambiguous, and use terminology consistent with prior artifacts. Vague language or inconsistent terms (e.g., using "login" in one table and "sign-in" in another) will result in deductions.

**5. Timely Submission (1 point)**