

Your Company has awarded with an Online Train Ticketing System and assume you're assigned as Project Manager. This website will have all the train ticket booking and management facilities. The system has both web and mobile app platform. The system will open the tickets to book and purchase 5 days earlier to the departure date and customer can book and purchase the desired ticket item that time onward. Extra 10% discount will be provided to the customer only if they are register via the mobile app. Imagine you are assigned as a Project Manager to implement the "Online Train Ticketing System". Based on the High-Level scope

- a. Estimate the size of the project in terms of effort (man-days). Distribute the effort in different phases. (
- b. Prepare a Work plan or Project Plan for the project, showing all the planned tasks (minimum 2 levels) in different phases, duration (days), start date, end date, responsible.

**1. Requirement Gathering and Analysis:**

- Effort: 20% of the total project
- Man-days: Assuming a total project duration of 100 days, this phase would require around 20 days ( $100 * 0.20$ ).

**2. System Design:**

- Effort: 25% of the total project
- Man-days: Around 25 days ( $100 * 0.25$ ).

**3. Development:**

- Effort: 45% of the total project
- Man-days: Approximately 45 days ( $100 * 0.45$ ).

**4. Testing:**

- Effort: 15% of the total project
- Man-days: Roughly 15 days ( $100 * 0.15$ ).

**5. Deployment and Post-Deployment Activities:**

- Effort: 10% of the total project
- Man-days: Around 10 days ( $100 * 0.10$ ).

These estimations are based on a 100-day total project duration and a proportional breakdown of effort across each phase. Adjustments might be necessary based on the specific complexities and requirements of the project, as well as the size and expertise of the development team.

Part b

## **Phase 1: Requirement Gathering and Analysis**

Duration: 20 days (Start Date: DD/MM/YYYY, End Date: DD/MM/YYYY) Responsible: Project Manager

### **1. Project Kickoff and Planning**

- Duration: 5 days
- Start Date: DD/MM/YYYY
- End Date: DD/MM/YYYY
- Responsible: Project Manager

### **2. Client Meetings and Requirement Collection**

- Duration: 10 days
- Start Date: DD/MM/YYYY
- End Date: DD/MM/YYYY
- Responsible: Business Analyst

### **3. Requirements Documentation**

- Duration: 5 days
- Start Date: DD/MM/YYYY
- End Date: DD/MM/YYYY
- Responsible: Business Analyst

## **Phase 2: System Design**

Duration: 25 days (Start Date: DD/MM/YYYY, End Date: DD/MM/YYYY) Responsible: System Architect

### **1. Architecture Planning**

- Duration: 7 days
- Start Date: DD/MM/YYYY
- End Date: DD/MM/YYYY
- Responsible: System Architect

### **2. Database Design**

- Duration: 8 days
- Start Date: DD/MM/YYYY
- End Date: DD/MM/YYYY
- Responsible: Database Developer

### 3. UI/UX Design

- Duration: 10 days
- Start Date: DD/MM/YYYY
- End Date: DD/MM/YYYY
- Responsible: UI/UX Designer

## Phase 3: Development

Duration: 45 days (Start Date: DD/MM/YYYY, End Date: DD/MM/YYYY) Responsible: Development Team Lead

### 1. Backend Development

- Duration: 20 days
- Start Date: DD/MM/YYYY
- End Date: DD/MM/YYYY
- Responsible: Backend Developers

### 2. Frontend Development (Web)

- Duration: 15 days
- Start Date: DD/MM/YYYY
- End Date: DD/MM/YYYY
- Responsible: Frontend Developers

### 3. Mobile App Development

- Duration: 10 days
- Start Date: DD/MM/YYYY
- End Date: DD/MM/YYYY
- Responsible: Mobile App Developers

## Phase 4: Testing

Duration: 15 days (Start Date: DD/MM/YYYY, End Date: DD/MM/YYYY) Responsible: QA Team Lead

### 1. Unit Testing

- Duration: 5 days
- Start Date: DD/MM/YYYY
- End Date: DD/MM/YYYY

- Responsible: Developers and QA Team

## **2. Integration Testing**

- Duration: 5 days
- Start Date: DD/MM/YYYY
- End Date: DD/MM/YYYY
- Responsible: QA Team

## **3. User Acceptance Testing (UAT)**

- Duration: 5 days
- Start Date: DD/MM/YYYY
- End Date: DD/MM/YYYY
- Responsible: Client and QA Team

# **Phase 5: Deployment and Post-Deployment Activities**

Duration: 10 days (Start Date: DD/MM/YYYY, End Date: DD/MM/YYYY) Responsible: Operations Team

## **1. Deployment Planning**

- Duration: 3 days
- Start Date: DD/MM/YYYY
- End Date: DD/MM/YYYY
- Responsible: Project Manager and Operations Team

## **2. System Deployment**

- Duration: 5 days
- Start Date: DD/MM/YYYY
- End Date: DD/MM/YYYY
- Responsible: Operations Team

## **3. Post-Launch Support and Monitoring**

- Duration: 2 days
- Start Date: DD/MM/YYYY
- End Date: DD/MM/YYYY
- Responsible: Support Team and Project Manager

This is a basic outline to showcase tasks across various phases, each with estimated durations, start dates, end dates, and responsible individuals or teams. The actual project plan might include more detailed tasks, dependencies, and resources involved. Adjustments and additions can be made based on specific project requirements and team capacities.

#### Grantt Chartt

To calculate the total time needed to finish the project, we can add up the durations of all the tasks in the project plan: Total time = Duration of Task A + Duration of Task B + ... + Duration of Task L  
Let's calculate the total time needed to finish the project: Total time = 6 weeks + 6 weeks + 3 weeks + 9 weeks + 15 weeks + 1 week + 6 weeks + 3 weeks + 9 weeks + 3 weeks + 4 weeks + 3 weeks  
Total time = 68 weeks Therefore, the total time needed to finish the project is 68 week.

Black box and white box testing are two different approaches used in software testing to evaluate the functionality and performance of a software application.

#### Black Box Testing:

- **Definition:** Black box testing is a testing technique where the internal structure or workings of the software being tested are not known to the tester. The tester only focuses on the inputs and outputs without considering how the system processes these inputs to generate the outputs.
- **Approach:** Testers treat the software as a black box, testing the functionalities by providing inputs and observing outputs to ensure that the software behaves as expected.
- **Focus:** It concentrates on the functionalities, user interface, and overall behavior of the system without delving into the internal code or logic.
- **Testing Methods:** Equivalence partitioning, boundary value analysis, and various other functional testing methods are commonly used in black box testing.

#### White Box Testing:

- **Definition:** White box testing, also known as clear box testing or structural testing, involves testing the internal structure, code, and design of the software being developed.
- **Approach:** Testers have knowledge of the internal code structure and use this knowledge to design test cases that ensure each line of code is tested, branches are executed, and expected outcomes are achieved.
- **Focus:** It emphasizes verifying the flow of inputs and outputs through the software, examining the internal structures like control flow, code paths, and branches.
- **Testing Methods:** Techniques such as code coverage analysis, path testing, and statement coverage are commonly used in white box testing to ensure comprehensive code coverage.

#### Key Differences:

1. **Knowledge of Internal Structure:**

- Black box testing: Testers have no knowledge of the internal code or structure.
- White box testing: Testers possess knowledge of the internal code and use this knowledge for testing.

2. **Approach to Testing:**

- Black box testing: Focuses on functionalities and user perspectives.
- White box testing: Concentrates on code structure, paths, and logic.

3. **Testing Methods:**

- Black box testing: Employs functional and non-functional testing methods.
- White box testing: Uses code coverage and structural testing methods.

Both testing methods are essential in ensuring comprehensive testing coverage. Black box testing verifies if the software meets requirements and functions as expected from a user's perspective, while white box testing validates the internal logic, code structure, and paths within the software. Often, a combination of both approaches is used to achieve thorough test coverage in software testing projects.