EXPERIMENT 6

*Workbreakdown structure and scheduling using Gantt chart*

Ferry Ticketing System

Adwait Purao- 2021300101

Viraj Bhalerao – 2022301002

TE COMPS B - Batch B/B2

# AIM

To make a WBS and Gantt Chart for the problem statement selected.

# PROBLEM STATEMENT:

In the ferry ticketing system's class diagram, the central classes include **FerrySystem**, representing the overarching system with attributes like systemID and operators[]; **FerryOperator**, encapsulating operators' information and associating with routes[] and classes[]; **FerryRoute**, embodying details of ferry routes such as origin, destination, and schedule, connected to a specific ferry operator and a corresponding class; **FerryClass**, defining different classes with seat types and prices; **FerryTicket**, capturing ticket details like passenger information, route, class, seat number, and payment information; and **Passenger**, storing passenger information. Additionally, the **Payment** class manages payment-related attributes like paymentID, amount, and method. Relationships are established between these classes, enabling the representation of a cohesive ferry ticketing system. For instance, a passenger can search for available routes, select a route and class, and make a reservation, while the system handles payment processing and reservation cancellations with appropriate refund

mechanisms. This design ensures a comprehensive and adaptable framework for ferry ticket reservations across various operators and routes.

# THEORY

Work Breakdown Structure:

Work breakdown structure (WBS) in project management is a method for completing a complex, multi-step project. It's a way to divide and conquer large projects to get things done faster and more efficiently. The goal of a WBS is to make a large project more manageable. Breaking it down into smaller chunks means work can be done simultaneously by different team members, leading to better team productivity and easier project management.

The Project Management Institute (PMI) defines WBS as "a deliverable-oriented hierarchical decomposition of the work to be executed by the project team to accomplish the project objectives and create the required deliverables." Each WBS level represents a new and increasingly detailed definition of work needed to complete the project. PMI's definition adds that a WBS structure must be constructed in a way that each new level in the hierarchy includes all the work needed to complete its parent task. This means that every parent task element must have more than one child task within it to consider the parent task element complete.

Gantt Chart:

A Gantt chart, commonly used in project management, is one of the most popular and useful ways of showing activities (tasks or events) displayed against time. On the left of the chart is a list of the activities and along the top is a suitable time scale. Each activity is represented by a bar; the position and length of the bar reflect the start date, duration, and end date of the activity. This allows you to see at a glance:

1. What the various activities are
2. When each activity begins and ends
3. How long each activity is scheduled to last

# IMPLEMENTATION

Work Breakdown Structure:

A diagram of a company

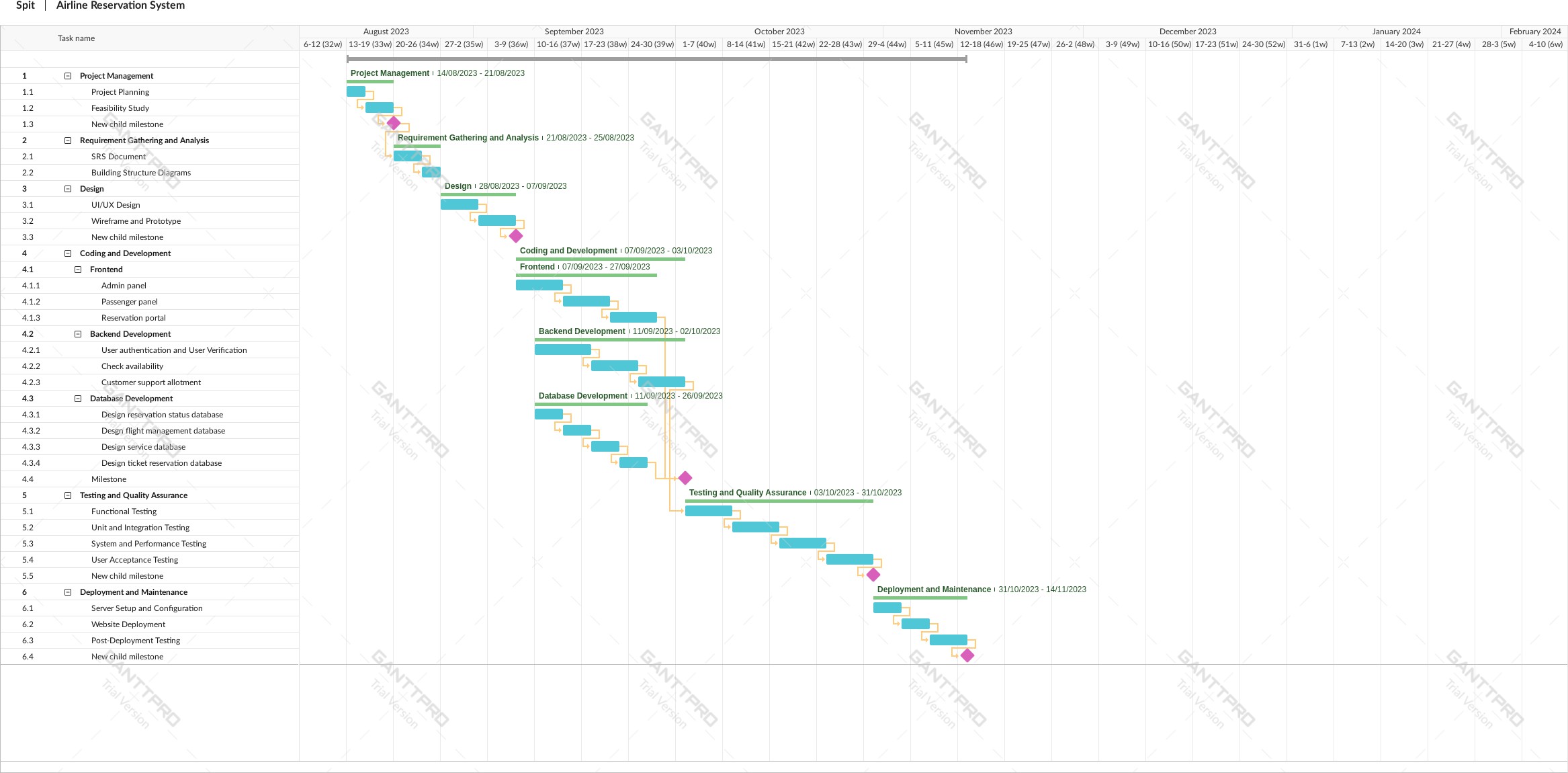
Description automatically generated

Gantt Chart:

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **WBS**  **Number** |  |  | **Task name / Title** | | **Planned start date** | **Planned end date** | | **Duration (hours)** | | **Predecessors** | |
| **1 Project Management** | | | | **14/08/23** | | | **21/08/23** | | **40** |  |
| 1.1 Project Planning | | | | 14/08/23 | | | 15/08/23 | | 16 |
| 1.2 Feasibility Study | | | | 16/08/23 | | | 18/08/23 | | 24 | 1.1 |
| 1.3 Milestone | | | | 21/08/23 | | | 21/08/23 | | 0 | 1.2 |
| **2 Requirement Gathering and Analysis** | | | | **21/08/23** | | | **25/08/23** | | **40** |  |
| 2.1 SRS Document | | | | 21/08/23 | | | 23/08/23 | | 24 | 1.3 |
| 2.2 Building Structure Diagrams | | | | 24/08/23 | | | 25/08/23 | | 16 | 2.1 |
| **3 Design** | | | | **28/08/23** | | | **07/09/23** | | **64** |  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 3.1 UI/UX Design | 28/08/23 | 31/08/23 | 32 |  |
| 3.2 Wireframe and Prototype | 01/09/23 | 06/09/23 | 32 | 3.1 |
| 3.3 Milestone | 07/09/23 | 07/09/23 | 0 | 3.2 |
| **4 Coding and Development** | **07/09/23** | **03/10/23** | **144** |  |
| **4.1 Frontend Development** | **07/09/23** | **27/09/23** | **120** |  |
| 4.1.1 Admin panel | 07/09/23 | 13/09/23 | 40 |  |
| 4.1.2 Passenger panel | 14/09/23 | 20/09/23 | 40 | 4.1.1 |
| 4.1.3 Reservation portal | 21/09/23 | 27/09/23 | 40 | 4.1.2 |
| **4.2 Backend Development** | **11/09/23** | **02/10/23** | **128** |  |
| User authentication and User  4.2.1 Verification | 11/09/23 | 18/09/23 | 48 |  |
| 4.2.2 Check availability | 19/09/23 | 25/09/23 | 40 | 4.2.1 |
| 4.2.3 Customer support allotment | 26/09/23 | 02/10/23 | 40 | 4.2.2 |
| **4.3 Database Development** | **11/09/23** | **26/09/23** | **96** |  |
| Design reservation status  4.3.1 database | 11/09/23 | 13/09/23 | 24 |  |
| Design ferry management  4.3.2 database | 14/09/23 | 18/09/23 | 24 | 4.3.1 |
| 4.3.3 Design service database | 19/09/23 | 21/09/23 | 24 | 4.3.2 |
| Design ticket reservation  4.3.4 database | 22/09/23 | 26/09/23 | 24 | 4.3.3 |
|  |  |  |  | 4.3.4, 4.2.3, |
| 4.4 Milestone | 03/10/23 | 03/10/23 | 0 | 4.1.3 |
| **5 Testing and Quality Assurance** | **03/10/23** | **31/10/23** | **160** |  |
| 5.1 Functional Testing | 03/10/23 | 09/10/23 | 40 | 4.4 |
| 5.2 Unit and Integration Testing | 10/10/23 | 16/10/23 | 40 | 5.1 |
| 5.3 System and Performance Testing | 17/10/23 | 23/10/23 | 40 | 5.2 |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 5.4 User Acceptance Testing | 24/10/23 | 30/10/23 | 40 | 5.3 |
| 5.5 New child milestone | 31/10/23 | 31/10/23 | 0 | 5.4 |
| **6 Deployment and Maintenance** | **31/10/23** | **14/11/23** | **80** |  |
| 6.1 Server Setup and Configuration | 31/10/23 | 02/11/23 | 24 |  |
| 6.2 Website Deployment | 03/11/23 | 07/11/23 | 24 | 6.1 |
| 6.3 Post-Deployment Testing | 08/11/23 | 13/11/23 | 32 | 6.2 |
| 6.4 Milestone | 14/11/23 | 14/11/23 | 0 | 6.3 |



# CONCLUSION

In our project timeline, we've planned for a streamlined and efficient development process. Starting on August 14, 2023, we aim to complete the project by November 14, 2023. We've allocated specific tasks and hours for each phase, considering weekends, to ensure continuous progress. By running certain phases in parallel, we aim to maximize efficiency and minimize downtime. This approach will help us deliver the project on time, ensuring thorough testing and quality assurance before deployment. We'll maintain a consistent 8-hour workday to meet our project goals and provide high-quality results to our clients.