PROJECT TITLE : FLIGHT DELAY MANAGEMENT SYSTEM

NAME : Sanchit Patil (23011041)

Aasawari Babar (23011042)

AIM :

In this experiment, we will learn how to identify functional and non-functional requirements from a given problem statement. These are the primary components of a Software Requirements Specification (SRS) for the Flight Delay Management System**.**

Introduction:

Requirement identification is the first step in any software project. For the Flight Delay Management System, it is essential to define what features the system should provide—such as logging delays, notifying passengers, and generating reports. Until these requirements are clearly understood and verified, development cannot proceed. Analysts usually work with airline staff and airport authorities to gather accurate and complete requirements.

THEORY :

Objectives :

- After completing this experiment you will be able to:
- Identify ambiguities, inconsistencies, and incompleteness from a requirements specification for a Flight Delay Management System
- Identify and state functional requirements specific to flight delay tracking and management
- Identify and state non-functional requirements such as system performance, reliability, and user accessibility for the flight delay system

Requirements :

A requirement specifies what the system should do without explaining how it will be done. For a Flight Delay Management System, requirements define expected behaviors such as tracking flight statuses, sending delay notifications, and handling rescheduling.

Requirements engineering involves understanding stakeholder needs and documenting them clearly. This becomes the foundation for system design, development, and testing.

It is essential to have a well-defined set of requirements before starting development. Incomplete or unclear requirements can cause issues later and may lead to user dissatisfaction, especially when timely flight updates are crucial.

Characteristics of Requirements :

Requirements gathered for the Flight Delay Management System should have the following properties:

1. Unambiguity

There should be no confusion about what the system should do.

Example: If the system is required to notify passengers about delays, the requirement should specify what counts as a delay (e.g., a delay of 15 minutes or more), not just say "notify on delays."

2. Consistency

Requirements should not contradict each other.

Example: One stakeholder might say “Notify passengers if delay exceeds 15 minutes,” while another says “Notify only if delay exceeds 30 minutes.” This inconsistency needs to be resolved.

3. Completeness

Requirements should specify what the system must do and what it must not do.

Example: If the system is to show real-time flight status, it should also specify that flights without updated data must not show incorrect or outdated information.

Categorization of Requirements :

User requirements:

Written in natural language for airline staff and passengers to understand, e.g., "Passengers should receive delay notifications via SMS or email."

System requirements:

Technical specifications for developers and testers, e.g., "The system shall poll the flight status API every 5 minutes."

Based on what they describe:

Functional Requirements (FRs):

Describe what the system should do.

Examples:

Track flight statuses in real-time.

Send delay notifications to affected passengers.

Allow airlines to update delay reasons.

Non-Functional Requirements (NFRs):

Describe how the system performs under certain conditions.

Examples:

The system shall be available 24x7.

Delay notification should be delivered within 2 minutes of a status update.

The system should support up to 10,000 simultaneous users.

Functional Requirements:

Identifying Functional Requirements for Flight Delay Management System:

Identify high-level functions:

Example: The system should be able to receive flight data and notify passengers of delays.

Identify use cases where users perform meaningful actions:

Example: Passengers can subscribe to flight updates, airlines can log delay reasons.

Think of the system as a black box with inputs and outputs:

Example: Input: Flight number and scheduled time; Output: Current flight status and delay alerts.

Decompose high-level requirements into sub-requirements:

Example: Notification could vary by user preference (email, SMS, mobile app).

Preparing Software Requirements Specifications

Once all functional and non-functional requirements for the **Flight Delay Management System** are identified and verified to be complete, consistent, and unambiguous, an SRS (Software Requirements Specification) is prepared. It follows IEEE standards and acts as a legal agreement between the client and the developer.

The SRS ensures that all expected features like delay tracking, notifications, and flight rescheduling are documented. If any feature is missing after development or requested later, the SRS helps resolve such disputes. However, writing the full SRS is beyond the scope of this experiment.

SIMULATION :

Functional Requirements

The Flight Delay Management System should allow users to register and log in based on their role, such as passenger or airline staff. Admin users must be able to add, update, or delete flight schedules as needed. The system should track real-time flight statuses, identifying whether a flight is on time, delayed, or cancelled, and record the reason for any delay (such as weather, technical issues, or air traffic).

In case of delays, the system must notify affected passengers through email, SMS, or in-app alerts. Passengers should also have the option to reschedule or cancel their bookings if their flight is significantly delayed. Additionally, the system should generate reports for admins showing delay trends and causes over time.

A user-friendly dashboard should display all upcoming delayed flights for passengers. The system must also maintain historical records of flights and their delay information. To enhance usability, users should be able to search or filter delay records by flight number, date, or status.

PROCEDURE :

General Instructions:

To perform the experiments in the Software Engineering Virtual Lab using the Flight Delay Management System (FDMS) as the chosen topic, follow these general steps:

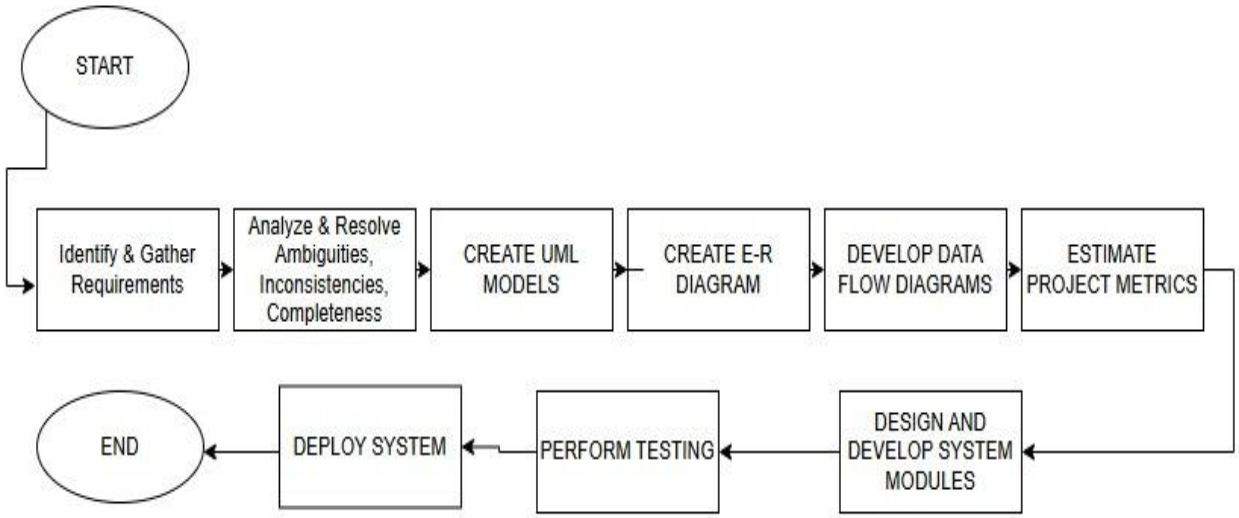
1. Read the theory related to the experiment.
2. Understand the simulation based on the FDMS problem statement.
3. (Optional) Take the self-evaluation to assess your understanding.
4. Attempt and solve the list of given exercises.

Experiment-Specific Instructions (for FDMS):

1. Carefully read the problem statement of the Flight Delay Management System**.**

2. Identify if there are any inconsistencies or missing information in the requirement specifications.
3. Clearly determine the functional and non-functional requirements of the system.
4. Select the appropriate checkboxes or fields (if using the virtual lab interface), and click on the **'Submit'** button to proceed.

FLOW DIAGRAM :



Rather than VLab words document for writeup and digram website for flow diagram.