**PROJECT TITLE :FLIGHT DELAY MANAGEMENT SYSTEM**

**NAME :**SHARVARI MULEY (23011064)

KRUTIKA NAPHADE (23011060)

**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:**

* **Purpose**:  
  To build a system that efficiently tracks, manages, and communicates flight delays to relevant stakeholders including passengers, airlines, and airport authorities.
* **Scope**:  
  The system will handle delay logging, passenger notifications, and historical report generation. It aims to improve passenger experience and operational awareness.
* **Overview**:  
  This SRS outlines all functional and non-functional requirements for the system. It serves as a foundation for system design and testing.

**General Description:**

 **Functions**:

* Real-time tracking of flight statuses
* Logging of delay reasons
* Passenger notifications via SMS, Email, or App
* Admin dashboard for analytics and reports

 **User Community**:

* Passengers
* Airline staff
* Admin personnel

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."

### **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.

#### **3. Functional Requirements**

* **Possible Outcomes**:
  + On-time
  + Delayed
  + Cancelled
* **Ranked Order** (Example Priority):
  + Real-time delay notifications
  + Rescheduling support
  + Historical reports
* **Input-Output Relationship**:
  + **Input**: Flight ID, current status
  + **Output**: Updated flight info, delay alert, rescheduling options

#### **4. User Interface Requirements**

* **Software Interfaces**:
  + Admin dashboard
  + Passenger portal
  + Mobile App
* **Examples**:
  + Login/Registration screens
  + Delay notification settings
  + Flight search/filter page

#### **5. Performance Requirements**

* **Response Time**: Delay notifications should be sent within 2 minutes.
* **Throughput**: System must handle 10,000 users concurrently.
* **Scalability**: Must scale up during peak hours (e.g., festival seasons).

#### **6. Non-Functional Attributes**

* **Usability**:  
  Simple UI for passengers, detailed options for admins.
* **Reliability**:  
  99.9% uptime required for critical operations.
* **Security**:  
  Password encryption, role-based access control, and secure data storage.

#### **7. Schedule and Budget**

* **Timeline**:  
  Development – 3 months  
  Testing – 1 month  
  Deployment – 1 week
* **Cost Estimate**:  
  Approx. ₹15 lakhs (includes server, dev resources, marketing)

#### **8. Appendices**

* **Supplementary Information**:  
  API endpoints, message formats, sample notifications
* **Glossary**:
  + **FDMS** – Flight Delay Management System
  + **NFR** – Non-Functional Requirement
  + **FR** – Functional Requirement

**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:**

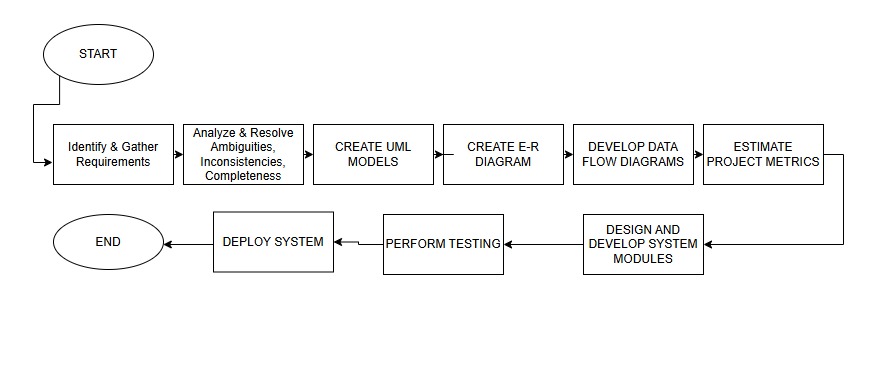
To perform the experiments in the Software Engineering Virtual Lab using the**Flight Delay Management System (FDMS)** as the chosen topic, follow these general steps:

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 writeupand digram website for flow diagram.

**Conclusion**

The **Flight Delay Management System (FDMS)** is designed to address the growing need for timely and accurate communication of flight delays to passengers and stakeholders. By clearly identifying both functional and non-functional requirements, this Software Requirements Specification ensures that the system will be reliable, scalable, and user-friendly.

Proper requirement analysis and documentation—such as what is outlined in this SRS—minimize the chances of project failure and ensure the software meets user expectations. Once implemented, the system will streamline operations, reduce passenger frustration, and provide useful analytics for airline decision-making.

**References**

1. IEEE Std 830-1998, IEEE Recommended Practice for Software Requirements Specifications.
2. Sommerville, Ian. Software Engineering, 10th Edition, Pearson Education.
3. Pressman, Roger S. Software Engineering: A Practitioner’s Approach, 7th Edition, McGraw-Hill.
4. SEPM Virtual Lab Manual – Software Engineering Lab (NPTEL/Virtual Labs)
5. Airline and Airport Authority Interviews (used for gathering real-world requirement insights)