# NAME: ASHLEY TAN WACHE

MATRICULE: FE22A154

COURSE: INTERNET PROGRAMMING AND MOBILE PROGRAMMINGS

# Collecting and Analyzing Users Requirements for a Mobile Application

## Introduction

Requirement engineering is a crucial process in software development that ensures a mobile application meets the needs and expectations of its users. It involves collecting, documenting, and analyzing user requirements to create a well-defined set of functional and non-functional specifications. This document outlines the key steps involved in collecting and analyzing user requirements for a mobile application.

## 1. Collecting User Requirements

Collecting user requirements involves gathering information from stakeholders, including end users, clients, and business owners. Several techniques can be used:

### 1.1 Interviews

- Conduct structured or unstructured interviews with potential users to understand their needs.  
- Ask open-ended questions to gather insights on user expectations and pain points.

### 1.2 Surveys and Questionnaires

- Distribute online or paper-based surveys to collect responses from a broad user base.  
- Use Likert scales, multiple-choice questions, and open-ended questions to gain both qualitative and quantitative data.

### 1.3 Focus Groups

- Gather a small group of target users to discuss their needs and preferences.  
- Encourage users to share their opinions and experiences regarding similar applications.

### 1.4 Observations

- Observe how users interact with similar mobile applications.  
- Identify usability challenges and common usage patterns.

### 1.5 Competitive Analysis

- Study competitors’ applications to understand existing features, strengths, and weaknesses.  
- Identify gaps that can be addressed in the new mobile application.

## 2. Documenting User Requirements

After gathering the requirements, they must be documented in a structured format:

### 2.1 Functional Requirements

- Define the specific features and functionalities the application must have.  
- Example: "The app should allow users to register using their email or social media accounts."

### 2.2 Non-Functional Requirements

- Specify performance, security, and usability expectations.  
- Example: "The app should load within three seconds under normal network conditions."

### 2.3 Use Cases and User Stories

- Use Cases: Define how users will interact with the system.  
- User Stories: Short descriptions of user goals in the following format:  
   
 Example: "As a user, I want to reset my password so that I can regain access to my account."

### 2.4 Wireframes and Prototypes

- Create sketches or digital wireframes to visualize the application's layout and navigation.  
- Use tools like Figma, Adobe XD, or Balsamiq for digital prototypes.

## 3. Analyzing User Requirements

Once requirements are collected and documented, they must be analyzed to ensure feasibility and completeness.

### 3.1 Prioritization

- Categorize requirements based on their importance using frameworks like MoSCoW (Must-have, Should-have, Could-have, Won't-have).

### 3.2 Feasibility Analysis

- Assess technical, financial, and operational feasibility to determine whether the requirements can be implemented.

### 3.3 Conflict Resolution

- Identify conflicting requirements from different stakeholders and resolve them through discussions and trade-offs.

### 3.4 Validation and Verification

- Conduct review sessions with stakeholders to validate requirements.  
- Ensure that documented requirements accurately reflect user needs.

## Conclusion

Collecting and analyzing user requirements is a fundamental step in developing a successful mobile application. By employing various techniques to gather insights, documenting them in a structured manner, and thoroughly analyzing their feasibility and priority, development teams can ensure they build a user-centric and efficient application.