Software Requirements Specification

Project Title: Medication Tracker
Version: 1.0
Project Code: MT-2025
Supervisor: Fizza Aqeel
Co-Supervisor: Not applicable
Project Team:

Abdul Qadir (22i-2127) Laiba Khan (22k-4610) Hassan Rizvi (22k-4539)

Submission Date: May 6, 2025

Document History

Version	Name of Person	Date	Description of Change
1.0	Abdul Qadir	20-04-2025	Initial draft created with basic project details
1.1	Laiba	24-04-2025	Added functional requirements and use cases
1.2	Hassan	28-04-2025	Finalized non-functional requirements and reviewed document structure

---

Distribution List

Name	Role
Fizza Aqeel	Supervisor

Document Sign-Off

Version	Sign-off Authority	Sign-off Date

1.0	Fizza Aqeel	06-05-2025

Table of Contents

1. Introduction

1.1 Purpose of Document

This document outlines the functional and non-functional requirements for the Medication Tracker application. It serves as a guide for the development team and stakeholders to ensure a clear understanding of the system's capabilities and constraints.

1.2 Intended Audience

- **Project Supervisor:** Fizza Aqeel
- **Development Team:** Laiba, Abdul Qadir, Hassan
- **Potential Users:** Individuals managing multiple medications daily

1.3 Abbreviations

- **MT:** Medication Tracker
- **UI:** User Interface
- **UX:** User Experience
- **API:** Application Programming Interface

1.4 Document Convention

This document uses the following conventions:

- **Font:** Arial
- **Font Size:** 12pt
- **Headings:** Bold
- **Subheadings:** Italic

---

2. Overall System Description

2.1 Project Background

Managing multiple medications can be challenging for individuals, especially those with chronic illnesses. The Medication Tracker application aims to simplify this process by providing a user-friendly platform to track medication schedules, dosages, and inventory.

2.2 Project Scope

The core functionalities include:

- **User Accounts:**
    - Sign up, log in, and manage secure accounts
- **Medicine Entry:**
    - Add medicine name, dosage, time, start & end date
- **Daily Tracking:**
    - Mark each dose as Taken or Skipped
    - Track per scheduled time
- **Upcoming Medicines:**
    - View medicines due in the next 24 hours
- **History:**
    - Show a 7-day log of taken/skipped actions
    - Daily breakdown of medication adherence

2.3 Not In Scope

- Integration with external healthcare systems
- Integration with third-party health record systems or advanced analytics
- Support for multiple user profiles

2.4 Project Objectives

- Develop an intuitive Android application for medication management
- Ensure accurate tracking of medication inventory and schedules
- Provide timely reminders to users for medication intake and restocking

2.5 Stakeholders

- **Primary Users:** Individuals managing multiple medications
- **Development Team:** Laiba, Abdul Qadir, Hassan
- **Supervisor:** Fizza Aqeel

2.6 Operating Environment

- **Platform:** Android OS
- **Development Tools:** React Native
- **Third-party Libraries:** CompactCalendarView for calendar functionalities

2.7 System Constraints

- **Cloud Storage:**
    - Data is stored remotely using Firebase
    - Ensures real-time sync and secure cloud-based data persistence

- **Cross-Platform Support:**
    - React Native supports both Android and iOS
- **Timestamp Handling:**
    - No manual Unix timestamp input required
    - User-friendly input (e.g., date/time pickers)

2.8 Assumptions & Dependencies

- Users will input accurate medication details and timestamps
- Application relies on the CompactCalendarView library for calendar functionalities

---

3. External Interface Requirements

3.1 Hardware Interfaces

- Supports Android and iOS smartphones and tablets
- **Minimum Android requirement:** API level 21 (Lollipop) and above
- **iOS requirement:** iOS 11 and above (React Native default compatibility)

3.2 Software Interfaces

The application integrates with the following components:

- **Firebase Services:**
    - Authentication – user sign-up, sign-in, and password recovery
    - Cloud Firestore – real-time database for medication records

    - Firebase Rules – secure access enforcement

- **React Native Libraries:**
    - `@react-native-firebase/app`, `@auth`, and `@firestore`
    - `@react-native-community/datetimepicker`
    - `react-native-calendars`

3.3 Communications Interfaces

- Uses **HTTPS** to interact with Firebase
- Requires stable internet for data sync
- Offline support via Firebase's local caching

---

4. Functional Requirements

4.1 Functional Hierarchy

1. **User Account Management**
    - Registration, Login, Logout, Password Recovery
    - Firebase Authentication Integration
2. **Medicine Management**
    - Add Medicine: name, dosage, time, date
    - Store and Sync with Firebase
3. **Daily Tracking**

- Display Today’s Medicines
- Mark as Taken/Skipped
- Save Daily Status

4. **Upcoming Medicines**
    - List medicines due in coming month
5. **Medication History**
    - Maintain 7-Day History
    - Display Taken/Skipped Logs
6. **User Interface**
    - React Native UI
    - Date/Time Picker
    - Optional Calendar View

4.2 Use Cases

4.2.1 Add Medication

- **Use Case ID:** UC-01
- **Actors:** User
- **Feature:** Treatment Plan Management
- **Pre-condition:** User has launched the application and navigated to the settings interface

Scenarios:

Step	Action	Software Reaction
1	User selects "Add Medication"	Application displays input fields

2	User enters name, pills per day, start and end dates	Application validates input
3	User inputs current Unix timestamp	System stores timestamp
4	User saves medication	Schedule is recalculated

Alternate Scenarios:

- **1a:** User enters invalid data → App prompts correction

Post Conditions:

- Medication is added to treatment plan
- Calendar is updated

Cross-referenced Use Cases:

- UC-02 (Mark Medication as Taken)
- UC-03 (View Upcoming Medications)

---

5. Non-functional Requirements

5.1 Performance Requirements

- App responds to input within 2 seconds
- Medication schedules updated daily

5.2 Safety Requirements

- Uses Firebase Auth for secure access

- HTTPS communication
- Firebase Rules enforce user-level data access
- No sensitive data stored on device
- Secure error handling

5.3 Security Requirements

- Local storage to maintain privacy
- No third-party data sharing

5.4 User Documentation

- User Manual
- In-app Help Section
- FAQs

---

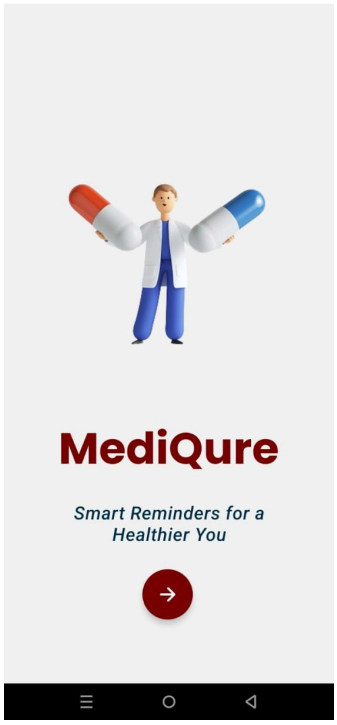
6. References

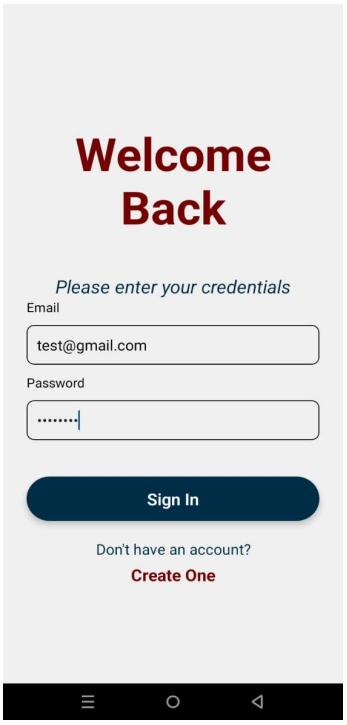
1. *IEEE Std 830-1998 – IEEE Recommended Practice for Software Requirements Specifications*
2. *Firebase Documentation – https://firebase.google.com/docs*
3. *React Native Documentation – https://reactnative.dev/docs*
4. *Android Developers Guide – https://developer.android.com/guide*
5. *Tuberjee. React Native Firebase Tutorial https://www.youtube.com/@tuberjee*

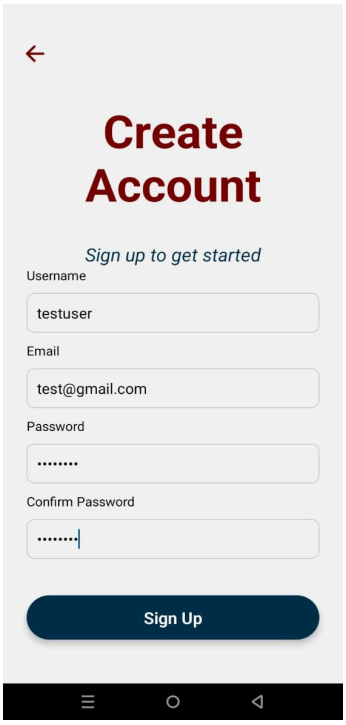
---

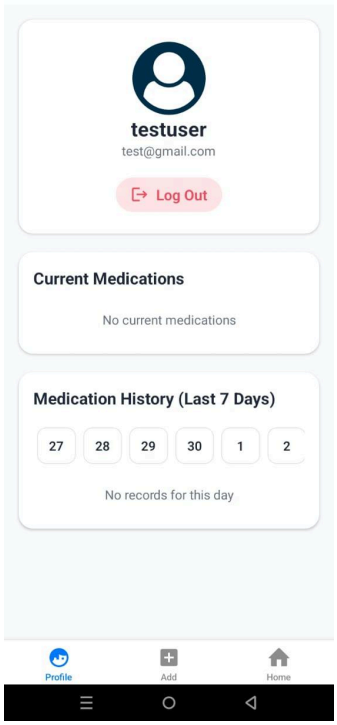
7. Appendices

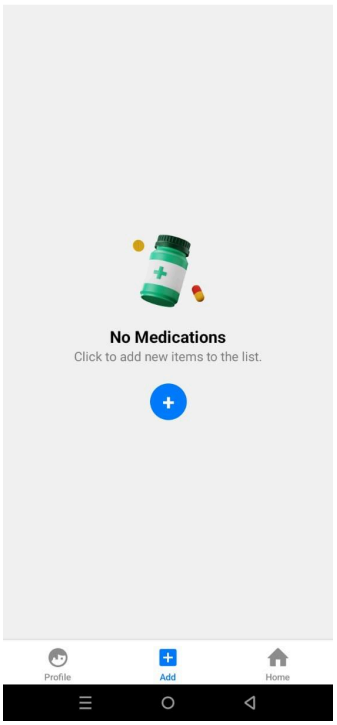
Appendix A: Screenshots of Application Interfaces

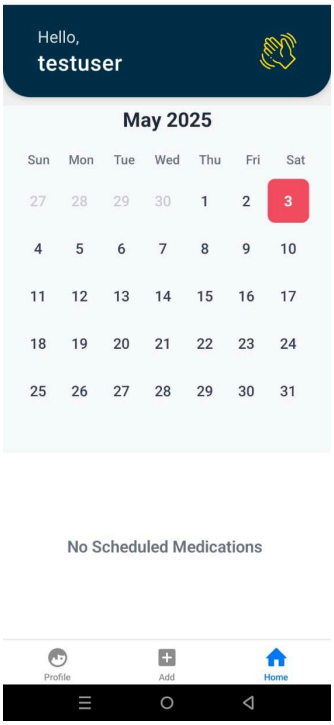












Appendix B: Sample Medication Schedule Calculations