
Software Requirements Specification

for

Digital Diary App

Version 1.0

Prepared by

Mohammed Hassan

202220173

Abdelrahman Adel

202220157

Takey Eldean

202220131

<student name>

<student code>

<student name>

<student code>

<student name>

<student code>

Instructor: *Mohammed karam*

Course: Software engineering

Teaching Assistant: *Mohammed karam*

Date: 14/11/2024

Table of Contents

Table of Contents.....	2
Revision History.....	2
1. Introduction.....	1
1.1 Purpose.....	1
1.2 Document Conventions.....	1
1.3 Intended Audience and Reading Suggestions.....	1
1.4 Product Scope.....	1
1.5 References.....	1
2. Overall Description.....	1
2.1 Product Perspective.....	1
2.2 Product Functions.....	2
2.3 User Classes and Characteristics.....	2
2.4 Operating Environment.....	2
2.5 Design and Implementation Constraints.....	2
2.6 User Documentation.....	2
2.7 Assumptions and Dependencies.....	2
3. External Interface Requirements.....	3
3.1 User Interfaces.....	3
3.2 Hardware Interfaces.....	3
3.3 Software Interfaces.....	3
3.4 Communications Interfaces.....	3
4. System Features.....	4
4.1 System Feature 1.....	4
4.2 System Feature 2 (and so on).....	4
5. Other Nonfunctional Requirements.....	4
5.1 Performance Requirements.....	4
5.2 Safety Requirements.....	5
5.3 Security Requirements.....	5
5.4 Software Quality Attributes.....	5
5.5 Business Rules.....	5
6. Other Requirements.....	5
Appendix A: Glossary.....	5
Appendix B: Analysis Models.....	6
Appendix C: To Be Determined List.....	6

Revision History

Name	Date	Reason For Changes	Version

1. Introduction

1.1 Purpose

The purpose of this Software Requirements Specification (SRS) document is to define the requirements for the **Digital Diary App**, developed using Flutter. This document provides a comprehensive overview of the app's functionalities, performance requirements, and constraints. It is intended to serve as a blueprint for the development team, ensuring that the final product aligns with the expectations of the stakeholders and meets the end-user needs.

1.2 Document Conventions

This document follows the IEEE SRS format. The following conventions are used:

- *Shall* indicates a mandatory requirement.
- *Should* indicates a recommended but not mandatory feature.
- *May* indicates an optional feature.
- References to figures, tables, or sections are in italics (e.g., see Section 2.3).

1.3 Intended Audience and Reading Suggestions

The intended audience for this document includes:

- **Developers:** To implement the app's features based on the requirements specified.
- **Project Managers:** To track project progress and ensure alignment with goals.
- **Testers:** To validate the app's functionalities against the specified requirements.
- **End-Users and Stakeholders:** To understand the scope and features of the Digital Diary App.

1.4 Product Scope

The **Digital Diary App** aims to provide users with a convenient platform to document their thoughts, events, and experiences in a secure and organized manner. The key features of the app include:

- User authentication for secure access.
- Creating, editing, and deleting diary entries.
- Categorizing and tagging entries for better organization.
- Support for multimedia content (images, videos).
- Data backup and synchronization for secure storage and easy retrieval.

The app is developed using Flutter to ensure compatibility with both Android and iOS platforms.

1.5 References

- *Flutter Documentation: Flutter Official Documentation*
- *IEEE Software Requirements Specification Standard: [IEEE 830-1998 SRS Standard](#)*
- *Sample SRS Document for E-commerce Application (provided in the course materials)*

2. Overall Description

2.1 Product Perspective

The **Digital Diary App** is a standalone mobile application that serves as a personal journaling tool for users. It is built using Flutter, enabling cross-platform compatibility on both Android and iOS devices. The app aims to replace traditional paper diaries and other note-taking apps by providing a secure, user-friendly, and feature-rich experience for documenting daily thoughts, experiences, and events.

The application will interact with:

- **Local Storage:** For storing entries offline on the user's device.
- **Cloud Services (optional):** For data backup and synchronization, allowing users to access their diary entries across multiple devices.
- **Authentication Services:** To ensure secure access through user login.

2.2 Product Functions

The Digital Diary App provides the following key functionalities:

- **User Authentication:** Allows users to sign up, log in, and reset passwords using email and password.
- **Diary Entry Management:** Users can create, edit, and delete diary entries with text input.
- **Multimedia Attachments:** Users can attach images, videos, and audio notes to their entries.
- **Search and Filter:** Users can search entries by date, keywords, or tags.
- **Categorization and Tagging:** Entries can be tagged or categorized for better organization.
- **Data Backup and Synchronization:** Option for users to back up their data to a cloud service (e.g., Firebase).
- **Security Features:** Data encryption to ensure user privacy and secure storage.

2.3 User Classes and Characteristics

The app targets the following user classes:

- **Basic Users:** General users who want to keep a digital diary of their daily thoughts and experiences. They have basic knowledge of mobile applications and expect a simple, intuitive interface.
- **Advanced Users:** Users who require additional features such as multimedia attachments, search capabilities, and data synchronization across devices. They may have experience with similar note-taking or journaling apps.

2.4 Operating Environment

The Digital Diary App will operate in the following environments:

- **Mobile Platforms:** Android (version 7.0 and above) and iOS (version 12.0 and above).
- **Development Environment:** Developed using Flutter SDK (version 3.0 or higher) with Dart programming language.
- **Cloud Services (optional):** Firebase for authentication and data backup.
- **Storage:** Uses local device storage with an option for cloud-based backup.

2.5 Design and Implementation Constraints

The following constraints are identified for the Digital Diary App:

- **Cross-Platform Compatibility:** The app must run seamlessly on both Android and iOS devices.
- **Data Security:** The app must ensure user data is encrypted both in transit (if using cloud backup) and at rest (stored on the device).
- **Performance:** The app must handle multimedia content efficiently without lag, even on lower-end devices.
- **Storage Limitations:** The app must handle limited device storage and provide users with warnings or options when storage is low.

2.6 User Documentation

The following documentation will be provided for the users:

- **User Guide:** A comprehensive guide detailing how to use the app, including features like creating entries, adding tags, and backing up data.
- **FAQs:** A section with frequently asked questions to assist users with common issues.

- **In-App Tutorials:** Step-by-step instructions displayed upon first-time use of the app to help users navigate key features.

2.7 Assumptions and Dependencies

- The app assumes users have an active internet connection for features like cloud backup and synchronization.
- The app depends on third-party services such as Firebase for authentication and data storage.
- The device must have a camera and microphone to support multimedia attachments.
- The app assumes that the user's device has sufficient storage for offline data and multimedia files.

3. External Interface Requirements

3.1 User Interfaces

The **Digital Diary App** will have an intuitive and user-friendly interface designed for ease of use across different user classes. The main components of the user interface include:

- **Login/Sign-Up Screen:** Users can log in with their email and password or create a new account.
- **Home/Dashboard Screen:** Displays a list of recent diary entries and allows users to navigate to create a new entry, view all entries, or search for specific entries.
- **Entry Creation Screen:** Provides a text editor for users to write their diary entries. It includes options to add multimedia (images, videos, audio) and tags.
- **Search and Filter Screen:** Users can search for specific entries using keywords, dates, or tags.
- **Settings Screen:** Allows users to manage their profile, adjust app preferences (e.g., theme, notifications), and enable cloud backup options.

UI Design Considerations:

- The UI will use a simple, clean design with a focus on accessibility, including options for larger text and dark mode.
- The app will use standard mobile gestures (e.g., swipe, tap) for navigation.

- *Error messages and prompts will be displayed for incorrect actions (e.g., failed login, storage issues).*

3.2 Hardware Interfaces

The **Digital Diary App** interacts with the following hardware components:

- **Camera:** *Used for capturing photos or videos to attach to diary entries.*
- **Microphone:** *Allows users to record audio notes as part of their diary entries.*
- **Storage:** *Utilizes the device's local storage for saving diary entries and multimedia files.*
- **Network Interface:** *Accesses the device's internet connection for cloud backup and synchronization.*

Hardware Requirements:

- *Android devices (version 7.0 and above) or iOS devices (version 12.0 and above).*
- *Devices should have a camera and microphone for multimedia features.*
- *At least 200 MB of free storage space is recommended for local data storage.*

3.3 Software Interfaces

The app will interact with the following software components:

- **Flutter SDK:** *The main framework for developing the cross-platform mobile application.*
- **Firebase Authentication:** *Provides secure user login and sign-up features.*
- **Firestore/Realtime Database (optional):** *Used for cloud-based storage and synchronization of diary entries.*
- **Local Database (SQLite):** *For offline storage of diary entries and metadata.*
- **Device APIs:** *Accessing device features like the camera, storage, and network.*

Third-Party Libraries:

- *The app may use additional third-party Flutter packages for features like multimedia handling, UI components, and data encryption.*

3.4 Communications Interfaces

The **Digital Diary App** will require communication interfaces for interacting with external services:

- **Internet Connectivity:** *The app will require an internet connection for user authentication, cloud backup, and data synchronization. The app should handle offline mode gracefully by storing entries locally until an internet connection is available.*

- **API Communication:** The app will communicate with Firebase services using secure REST APIs and the Firebase SDK for user authentication, data storage, and backup.
- **Data Transfer Protocols:** The app will use HTTPS for secure data transmission between the client (user's device) and the server (Firebase backend), ensuring encrypted communication.

Security Considerations:

- All API requests will be secured with HTTPS to protect user data.
- The app will use OAuth 2.0 for secure user authentication and authorization.
- Data synchronization will be handled with error checking and retries in case of network issues

4. System Features

4.1 User Authentication and Account Management

4.1.1 Description and Priority

*This feature allows users to securely sign up, log in, and manage their accounts. It is a **high-priority** feature since it provides access control and protects user data.*

4.1.2 Functional Requirements

- **FR-1:** The system shall allow users to create an account using an email and password.
- **FR-2:** The system shall allow users to log in using their registered email and password.
- **FR-3:** The system shall allow users to reset their password using a "Forgot Password" feature.
- **FR-4:** The system shall display error messages for failed login attempts (e.g., incorrect email or password).
- **FR-5:** The system shall ensure secure data transmission during login using HTTPS.

4.2 Diary Entry Management

4.2.1 Description and Priority

*This feature enables users to create, edit, and delete diary entries. It is the **core feature** of the application and has the highest priority.*

4.2.2 Functional Requirements

- **FR-6:** The system shall allow users to create new diary entries with a text editor.
- **FR-7:** The system shall allow users to edit existing diary entries.
- **FR-8:** The system shall allow users to delete unwanted diary entries.
- **FR-9:** The system shall auto-save entries while the user is typing to prevent data loss.
- **FR-10:** The system shall display a list of diary entries sorted by date.

4.3 Multimedia Attachments

4.3.1 Description and Priority

*This feature allows users to attach multimedia files (images, videos, audio notes) to their diary entries. It is a **medium-priority** feature, enhancing the user experience.*

4.3.2 Functional Requirements

- **FR-11:** The system shall allow users to attach images from the device gallery or capture new photos using the camera.
- **FR-12:** The system shall allow users to record audio notes and attach them to diary entries.
- **FR-13:** The system shall allow users to attach videos from the device gallery.
- **FR-14:** The system shall provide a preview option for multimedia attachments before saving.

4.4 Search and Filter

4.4.1 Description and Priority

*This feature enables users to search for specific entries using keywords, dates, or tags. It is a **high-priority** feature as it improves navigation and user experience.*

4.4.2 Functional Requirements

- **FR-15:** The system shall allow users to search diary entries using keywords from the entry content.
- **FR-16:** The system shall allow users to filter entries by date or tags.
- **FR-17:** The system shall display search results in a list format, sorted by relevance.

4.5 Data Backup and Synchronization

4.5.1 Description and Priority

*This feature provides users with the option to back up their data to a cloud service (e.g., Firebase) and synchronize entries across multiple devices. It is a **medium-priority** feature, focused on enhancing data security and accessibility.*

4.5.2 Functional Requirements

- **FR-18:** The system shall allow users to enable or disable automatic data backup to a cloud service.
- **FR-19:** The system shall synchronize diary entries across devices when the user logs in on a different device.
- **FR-20:** The system shall notify users if a backup or synchronization attempt fails due to network issues.

4.6 Security and Privacy

4.6.1 Description and Priority

*This feature ensures that all user data is securely stored and protected from unauthorized access. It is a **high-priority** feature given the sensitive nature of the content.*

4.6.2 Functional Requirements

- **FR-21:** The system shall encrypt all diary entries stored locally on the device.
- **FR-22:** The system shall use HTTPS for all data transmissions to and from the server.
- **FR-23:** The system shall log users out automatically after a period of inactivity (e.g., 15 minutes).
- **FR-24:** The system shall provide users with an option to enable biometric authentication (e.g., fingerprint, face recognition) for additional security.

5. Other Nonfunctional Requirements

5.1 Performance Requirements

- **PR-1:** The app shall load the home screen and display recent entries within 2 seconds on a typical mobile device.
- **PR-2:** The app shall handle up to 1,000 entries per user without significant performance degradation.
- **PR-3:** The app shall process search queries and display results within 1 second for a database containing up to 5,000 entries.
- **PR-4:** The app shall auto-save entries every 10 seconds during the editing process to prevent data loss.
- **PR-5:** The multimedia attachment feature shall support image files up to 10 MB and video files up to 100 MB without causing the app to crash.

5.2 Safety Requirements

- **SR-1:** The app shall not allow users to operate the camera feature while driving or in other situations where it could lead to unsafe behavior.
- **SR-2:** The app shall provide a warning if the user's device storage is below 100 MB to prevent potential data loss or crashes.
- **SR-3:** The app shall handle data backup errors gracefully, notifying users and providing options to retry or save locally without data loss.

5.3 Security Requirements

- **SecR-1:** The app shall encrypt all user data stored locally on the device using AES-256 encryption.
- **SecR-2:** The app shall use HTTPS for all communications between the client and server to protect against data interception.
- **SecR-3:** The app shall implement OAuth 2.0 for user authentication and authorization, ensuring secure login processes.

- **SecR-4:** The app shall support biometric authentication (e.g., fingerprint, facial recognition) as an optional layer of security for user login.
- **SecR-5:** The app shall log users out automatically after 15 minutes of inactivity to enhance security.
- **SecR-6:** The app shall not store user passwords in plaintext; instead, passwords will be hashed and salted before storage.

5.4 Software Quality Attributes

- **Reliability:** The app shall have an uptime of at least 99%, minimizing downtime and ensuring consistent availability.
- **Usability:** The app shall have an intuitive, user-friendly interface with accessible navigation. It shall support both light and dark themes for better user experience.
- **Scalability:** The app shall be designed to handle increased user data and entries as the user base grows without significant performance issues.
- **Maintainability:** The codebase shall be modular and follow clean code principles to facilitate easy updates and bug fixes.
- **Portability:** The app shall be compatible with both Android (version 7.0 and above) and iOS (version 12.0 and above) platforms, leveraging Flutter's cross-platform capabilities.

5.5 Business Rules

- **BR-1:** Users must be at least 13 years old to create an account, complying with general app privacy regulations.
- **BR-2:** The app shall not collect or share any personal data without explicit user consent, adhering to data privacy regulations (e.g., GDPR, CCPA).
- **BR-3:** Users can only create one account per email address to prevent duplicate accounts.
- **BR-4:** The app shall offer a premium feature (optional) for additional cloud backup and storage space, which users can subscribe to on a monthly or yearly basis.
- **BR-5:** In case of account deactivation or deletion, the app shall retain user data for a grace period of 30 days before permanent deletion, allowing users to recover their data if needed.

6. Other Requirements

This section includes any additional requirements that are not covered in the previous sections.

- **Regulatory Requirements:** The app shall comply with data protection laws and regulations such as the **General Data Protection Regulation (GDPR)** for users in the EU and the **California Consumer Privacy Act (CCPA)** for users in the USA.
- **Localization:** The app shall support multiple languages (e.g., English, Spanish, Arabic) based on the user's device settings.
- **Accessibility:** The app shall be designed with accessibility features in mind, including support for screen readers and high-contrast mode for users with visual impairments.

Appendix A: Glossary

Term	Definition
AES-256 Encryption	A strong encryption standard used to protect data at rest.
Cloud Backup	A feature that allows users to store their data on a cloud service for safe-keeping.
Flutter SDK	A software development kit used for building cross-platform mobile applications.
GDPR	General Data Protection Regulation, a legal framework for data protection in the EU.
OAuth 2.0	An industry-standard protocol for secure authorization.
SQLite	A lightweight, embedded database used for local data storage on the device.

Appendix B: Analysis Models

This section typically includes diagrams and models that provide a visual representation of the system's architecture and features. For the **Digital Diary App**, the following analysis models may be included:

1. **Use Case Diagrams:** Illustrate the main interactions between users and the app (e.g., creating an entry, attaching multimedia, searching entries).
2. **Data Flow Diagram (DFD):** Shows the flow of data between the user interface, local storage, and cloud services.
3. **Class Diagram:** Represents the structure of the app's data models, including classes for User, Entry, Tag, and Attachment.

These models can be created using diagram tools such as **draw.io**, **Lucidchart**, or **StarUML**.

Appendix C: To Be Determined (TBD) List

This section lists any requirements or decisions that are yet to be finalized.

Item	Description
<i>TBD-1: Cloud Backup Service</i>	<i>The specific cloud backup service (e.g., Firebase, AWS) is yet to be decided.</i>
<i>TBD-2: Monetization Strategy</i>	<i>Decision on whether the app will include in-app purchases or ads is pending.</i>
<i>TBD-3: Language Support</i>	<i>The complete list of languages to be supported is still under consideration.</i>
<i>TBD-4: Biometric Authentication</i>	<i>The types of biometric authentication (e.g., fingerprint, face ID) are yet to be finalized.</i>