**Software Requirements Specification**

**for**

Rewind App

**Version 1.0 approved**

**Prepared by Adithyan S Raj**

**Team-14**

**02/03/2021**

**Table of Contents**

**Table of Contents ii**

**Revision History ii**

**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 2**

2.1 Product Perspective 2

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 3

**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 5**

**Appendix C: To Be Determined List 6**

**Revision History**

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Date** | **Reason For Changes** | **Version** |
|  |  |  |  |
|  |  |  |  |

# **Introduction**

## **Purpose**

This is the initial release of the product [1.0]. This product is designed for productivity. With features like Journaling and task organization. But the best aspect of the app is that it aims to gamify our tasks thus making us more productive.

## **Document Conventions**

This document uses the following conventions:

|  |  |
| --- | --- |
| DB | Database |
| TBD | To Be Discovered |

## **Intended Audience and Reading Suggestions**

This project is intended for people of any age. This has been implemented under the guidance of college professors. This project is useful for anyone who wants to ramp up their productivity and organize their activities.

## **Product Scope**

Our product can help people improve their time management and other issues they may be facing. This app is built around the concepts of gamification which implies that the more we think of our life as a game the more likely we are to complete our tasks. So to do that we bring score tracking and other functionalities into our app which functions as a Journal and to-do list. But what separates us from other apps is that we make it seem more like a game. This app is cross-platform (Android and iOS) to reach a larger audience.

## **References**

1. [Gamification](https://en.wikipedia.org/wiki/Gamification#:~:text=Gamification%20is%20the%20application%20of,the%20characteristics%20of%20game%20elements.)
2. [Journaling](https://www.urmc.rochester.edu/encyclopedia/content.aspx?ContentID=4552&ContentTypeID=1)
3. [Cross-platform software](https://en.wikipedia.org/wiki/Cross-platform_software)

# **Overall Description**

## **Product Perspective**

The product is new and self-contained.

## **Product Functions**

*<Summarize the major functions the product must perform or must let the user perform. Details will be provided in Section 3, so only a high-level summary (such as a bullet list) is needed here. Organize the functions to make them understandable to any reader of the SRS. A picture of the major groups of related requirements and how they relate, such as a top-level data flow diagram or object class diagram, is often effective.>*

## **User Classes and Characteristics**

*<Identify the various user classes that you anticipate will use this product. User classes may be differentiated based on the frequency of use, a subset of product functions used, technical expertise, security or privilege levels, educational level, or experience. Describe the pertinent characteristics of each user class. Certain requirements may pertain only to certain user classes. Distinguish the most important user classes for this product from those that are less important to satisfy.>*

## **Operating Environment**

* Operating System: IOS, Android
* Database: Stored locally with upload functionalities
* Offline App
* Platform: Flutter

## **Design and Implementation Constraints**

There is a storage constraint as the app uses local storage. But we give the user the option to upload their files to google drive to rectify this.

## **User Documentation**

The help section inside the app provides how instructions to use the app and the various functionalities that go along with the app.

## **Assumptions and Dependencies**

*<List any assumed factors (as opposed to known facts) that could affect the requirements stated in the SRS. These could include third-party or commercial components that you plan to use, issues around the development or operating environment, or constraints. The project could be affected if these assumptions are incorrect, are not shared, or change. Also identify any dependencies the project has on external factors, such as software components that you intend to reuse from another project, unless they are already documented elsewhere (for example, in the vision and scope document or the project plan).>*

# **External Interface Requirements**

## **User Interfaces**

*<Describe the logical characteristics of each interface between the software product and the users. This may include sample screen images, any GUI standards or product family style guides that are to be followed, screen layout constraints, standard buttons and functions (e.g., help) that will appear on every screen, keyboard shortcuts, error message display standards, and so on. Define the software components for which a user interface is needed. Details of the user interface design should be documented in a separate user interface specification.>*

## **Hardware Interfaces**

*<Describe the logical and physical characteristics of each interface between the software product and the hardware components of the system. This may include the supported device types, the nature of the data and control interactions between the software and the hardware, and communication protocols to be used.>*

## **Software Interfaces**

*<Describe the connections between this product and other specific software components (name and version), including databases, operating systems, tools, libraries, and integrated commercial components. Identify the data items or messages coming into the system and going out and describe the purpose of each. Describe the services needed and the nature of communications. Refer to documents that describe detailed application programming interface protocols. Identify data that will be shared across software components. If the data sharing mechanism must be implemented in a specific way (for example, use of a global data area in a multitasking operating system), specify this as an implementation constraint.>*

## **Communications Interfaces**

*<Describe the requirements associated with any communications functions required by this product, including e-mail, web browser, network server communications protocols, electronic forms, and so on. Define any pertinent message formatting. Identify any communication standards that will be used, such as FTP or HTTP. Specify any communication security or encryption issues, data transfer rates, and synchronization mechanisms.>*

# **System Features**

## Basic functionalities like text, images, and videos for Journal

4.1.1 Description and Priority

This part is the part where we build the skeletal structure of our app with the text, images, and video uploading features. This is a high-priority feature.

4.1.2 Stimulus/Response Sequences

When the user selects the Journal section it will have an ‘+’ button which will allow the user to enter the journal for the day. Which can include text, images, and videos.

4.1.3 Functional Requirements

REQ-1: TBD

## App lock for security

4.2.1 Description and Priority

This protects the user data of the user that is currently logged in. This is a high-priority feature.

4.2.2 Stimulus/Response Sequences

When the user chooses to lock the app the app stays locked. It requires the user to enter the password or any means of biometric authentication to unlock the app again.

4.2.3 Functional Requirements

REQ-1: Fingerprint scanner in the app for biometric lock

* 1. **Storage and encryption of the data locally**

4.3.1 Description and Priority

The data is stored locally so to protect this data we encrypt it. This is a high-priority feature.

4.3.2 Stimulus/Response Sequences

When the user saves the journal for the day it gets encrypted and stored in the memory thus preventing malicious access to the data.

4.3.3 Functional Requirements

REQ-1: TBD

* 1. **Cloud backup feature**

4.4.1 Description and Priority

In-app cloud backup feature to store user data such as their journal, activity, and statistics. This is a medium-priority feature.

4.4.2 Stimulus/Response Sequences

If the user wants to backup their journal they can do so by uploading it to Google Drive using their Google account or even Onedrive using their Microsoft account. This cloud backup’ feature will be provided within the app.

4.4.3 Functional Requirements

REQ-1: TBD

* 1. **Gamified To-do list**

4.5.1 Description and Priority

Gamified to-do list where the user can enter the tasks they want to do. Based on whether the user completes the task they can gain experience or lose health if they neglected the task. This is a medium-priority feature.

4.5.2 Stimulus/Response Sequences

The User can add the various tasks by clicking the ‘+’ button. The user can specify the deadline of each task and if they didn’t tick the task within the deadline, the system assumes that the user hasn’t done it and the user will lose health. When the user loses all their health they lose a level. But if they complete their task they gain XP and can improve their level. The rate of leveling up will go up with each level.

4.5.3 Functional Requirements

REQ-1: TBD

* 1. **Statistics and game points for completing tasks**

4.6.1 Description and Priority

Statistics and game point to keep track and improve the user’s productivity. This is a low-priority feature.

4.6.2 Stimulus/Response Sequences

When the user completes a self assigned task their game points increase. Also, their game points will decrease if their activity is low or when they fail to complete their tasks.

4.6.3 Functional Requirements

REQ-1: TBD

# **Other Nonfunctional Requirements**

## **Performance Requirements**

*<If there are performance requirements for the product under various circumstances, state them here and explain their rationale, to help the developers understand the intent and make suitable design choices. Specify the timing relationships for real-time systems. Make such requirements as specific as possible. You may need to state performance requirements for individual functional requirements or features.>*

## **Safety Requirements**

*<Specify those requirements that are concerned with possible loss, damage, or harm that could result from the use of the product. Define any safeguards or actions that must be taken, as well as actions that must be prevented. Refer to any external policies or regulations that state safety issues that affect the product’s design or use. Define any safety certifications that must be satisfied.>*

## **Security Requirements**

*<Specify any requirements regarding security or privacy issues surrounding the use of the product or protection of the data used or created by the product. Define any user identity authentication requirements. Refer to any external policies or regulations containing security issues that affect the product. Define any security or privacy certifications that must be satisfied.>*

## **Software Quality Attributes**

*<Specify any additional quality characteristics for the product that will be important to either the customers or the developers. Some to consider are adaptability, availability, correctness, flexibility, interoperability, maintainability, portability, reliability, reusability, robustness, testability, and usability. Write these to be specific, quantitative, and verifiable when possible. At the least, clarify the relative preferences for various attributes, such as ease of use over ease of learning.>*

## **Business Rules**

*<List any operating principles about the product, such as which individuals or roles can perform which functions under specific circumstances. These are not functional requirements in themselves, but they may imply certain functional requirements to enforce the rules.>*

# **Other Requirements**

*<Define any other requirements not covered elsewhere in the SRS. This might include database requirements, internationalization requirements, legal requirements, reuse objectives for the project, and so on. Add any new sections that are pertinent to the project.>*

**Appendix A: Glossary**

*<Define all the terms necessary to properly interpret the SRS, including acronyms and abbreviations. You may wish to build a separate glossary that spans multiple projects or the entire organization, and just include terms specific to a single project in each SRS.>*

**Appendix B: Analysis Models**

*<Optionally, include any pertinent analysis models, such as data flow diagrams, class diagrams, state-transition diagrams, or entity-relationship diagrams*.>

**Appendix C: To Be Determined List**

*<Collect a numbered list of the TBD (to be determined) references that remain in the SRS so they can be tracked to closure.>*