Logo, icon

Description automatically generated

**GOLDFOLKS**

***RELEASE PLAN***

Version 1.0

*18/10/2021*

*Prepared by Team ElevenDegree*

Anil Ankitha

Chan Shao Jing

Chong Yow Lim

Lionel Wong Zhi Neng

Low Jin Teng Jackson

Ng Chi Hui

Zachary Varella Lee Zheyu

# APPROVALS

### Submitting Organization’s Approving Authority:

Signature Printed Name Date Email

 Chan Shao Jing 18/10/2021 CHAN0924@e.ntu.edu.sg

# REVISION HISTORY

|  |  |  |  |
| --- | --- | --- | --- |
| **Version** | **Date** | **Organization/Point of Contact** | **Description of Changes** |
| 0.1 | 04/10/2021 | ElevenDegree/Ng Chi Hui | Initial Version |
| 0.2 | 09/10/2021 | ElevenDegree/Ng Chi Hui | Added Introduction, Referenced Documents, Overview |
| 0.3 | 16/10/2021 | ElevenDegree/Ng Chi Hui | Added Assumptions, Constraints, Risks, Release Approach, Glossary, Acronyms and Appendices |
| 0.4 | 16/10/2021 | ElevenDegree/Chong Yow Lim | Added Release Approach |
| 1.0 | 18/10/2021 | ElevenDegree/Chan Shao Jing | Approved Baseline Version |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

# TABLE OF CONTENTS

1. [INTRODUCTION 1](#_bookmark0)
2. [REFERENCED DOCUMENTS 1](#_bookmark1)
3. [OVERVIEW 1](#_bookmark3)
4. [ASSUMPTIONS, CONSTRAINTS, RISKS 2](#_bookmark4)
   1. [Assumptions 2](#_bookmark5)
   2. [Constraints 2](#_bookmark6)
   3. [Risks 2](#_bookmark7)
5. [RELEASE APPROACH 2](#_bookmark8)
   1. [Rationale 2](#_bookmark9)
   2. [Release Strategy 3](#_bookmark10)
      1. [Release Content 3](#_bookmark11)
      2. [Release Schedule 3](#_bookmark12)
      3. [Release Impacts 3](#_bookmark13)
      4. [Release Notification 3](#_bookmark14)
6. [GLOSSARY 3](#_bookmark15)
7. [ACRONYMS 4](#_bookmark16)
8. [APPENDICES 4](#_bookmark17)

**LIST OF FIGURES**

Figure 1: Use Case Diagram …................10

Figure 2: Context Diagram …................11

Figure 3: System Architecture …................12

# LIST OF TABLES

[Insert a List of Tables appearing within the Release Plan along with a page reference for each identified table as appropriate. Labels of Table titles and descriptions are to be placed centered, above the table within the main body of the document.]

<Table #: Table Title or Description Page Number>

Table 1: Referenced Documents 9

Table 2: Risk Management Log 14

Table 3: Release Content Table 17

Table 4: Release Schedule 18

Table 5: Release Impact 18

Table 6: Release Notification 18

Table 7: List of terms and definitions 18

Table 8: List of acronyms and definitions 18

Table 9: List of referenced documents links 18

# INTRODUCTION

This document details the release plan specifications of Team ElevenDegree’s GoldFolks project.

## 1.1 Purpose

This document shall lay out the release plan for GoldFolks, which is an all-in-one elderly software application and the relevant documentation needed. The purpose of the release plan serves to manage, plan, schedule and control the software build through different stages and environments, including testing and deploying software releases. This also helps create an actionable plan built on specific features, enhancements and fixes to build.

## 1.2. Scope

This document consists of the release strategy of GoldFolks and will provide relevant details for all past, present, and future releases of the application and documents. This document shall be updated whenever there is a new build or release by the developmental team or changes made to the various documents by the QA team, which will allow us to record and keep track of all releases of the application and documents throughout the lifecycle of the project.

The release engineer shall work closely with the project manager, developmental team and QA team to plan release windows and cycles and document all version changes.

## 1.3. Intended Audience

The intended audience of GoldFolks’ release plan will likely be the internal developmental team and the product team (i.e., Key Business Stakeholders, Marketing and Sales Team). The document will help team members understand past and future progress of the application and make key decisions. The quality assurance team can refer to the release plan to ensure testing can be conducted accordingly and on time with the project phases.

## 1.4. Expected Evolution

The expected evolution of the documentation will likely include issues such as fixing bugs and possibly new functionalities arising from continuous user feedback. Additionally, maintenance must be done to ensure the system is up-to-date with new technology advances.

## 1.5. Security & Privacy Considerations

The release plan shall not be made known to the public (unless intended to), as it may contain sensitive information about future release. User information such as medical information, email, and password must remain well secured and encrypted to protect user privacy.

# REFERENCED DOCUMENTS

Table 1: Referenced Documents

|  |  |  |
| --- | --- | --- |
| Document Name | Document Number | Issuance Date |
| System Requirement Specifications | SRS | 11 September 2021 |
| Quality Plan | QP | 11 September 2021 |
| Risk Management Plan | RMP | 26 September 2021 |
| Project Plan | PP | 26 September 2021 |

# OVERVIEW

GoldFolks is designed as a cross-platform (iOS, Android) all-in-one mobile application for the elderly audience. It comprises of 3 main features, namely the medication reminder feature, the cognitive game feature, and the exercise feature. These functionalities are the most used in the daily life of the elderly thus have been incorporated into the application. Prior to the development of GoldFolks, the elderly must switch between many applications which causes a lot of inconvenience especially when they are less technology savvy compared to younger users.

The mobile application is developed by a group of CZ3002 Advanced Software Engineering students in NTU under Team ElevenDegree. It is developed using the Flutter framework for the front-end and Google Firebase for the backend. This application adopts the design pattern of the Model, Controller, View framework also known as MVC. The MVC design pattern comprises of 3 components: the Model, which stores and manages data, the Controller which controls data flow between the Model and the View components and the View which is the visual representation of the data output.

The design, implementation and testing are developed incrementally until the requirements are met and the product is delivered without error. Agile methodology was adopted in the development of the application.

The first release of version 0.1 was developed internally consisting of the skeleton of the application with the initial UI design acting as the baseline of the project. Since the Agile methodology is used, our developers will finish and test each of their assigned functional features independently. Then, all the features will be merged into Version 1.0 to be released as a functioning prototype for our QA team to test and identify bugs that occur within the application. Version 1.1 will capture the feedback of our QA team and be released to our external intended target audience for testing. The team will work on refining the application with bug fixes and feature extension in oich the final version will be released as Version 2.0.

The use case diagram which provides a high-level context of the system is as follows:

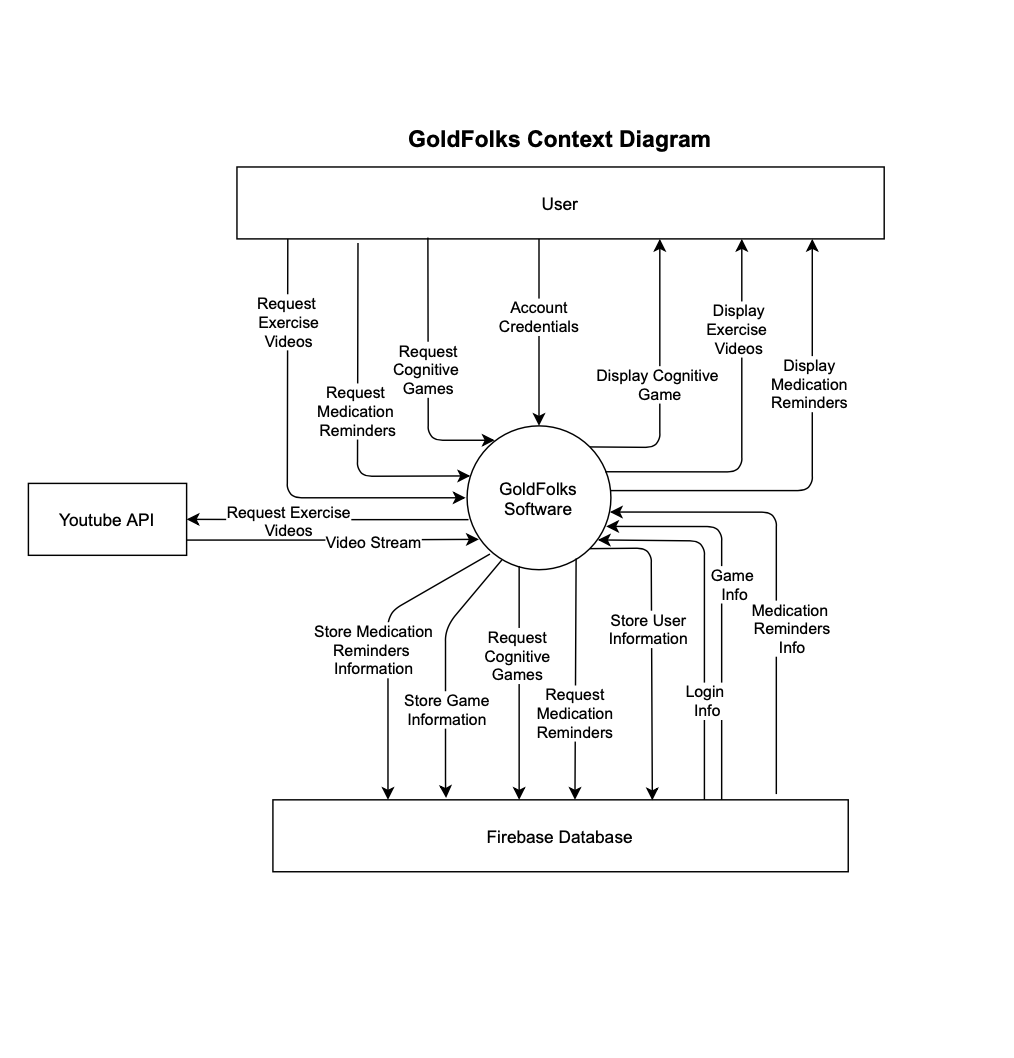
Figure 1: Use Case Diagram

Diagram

Description automatically generated

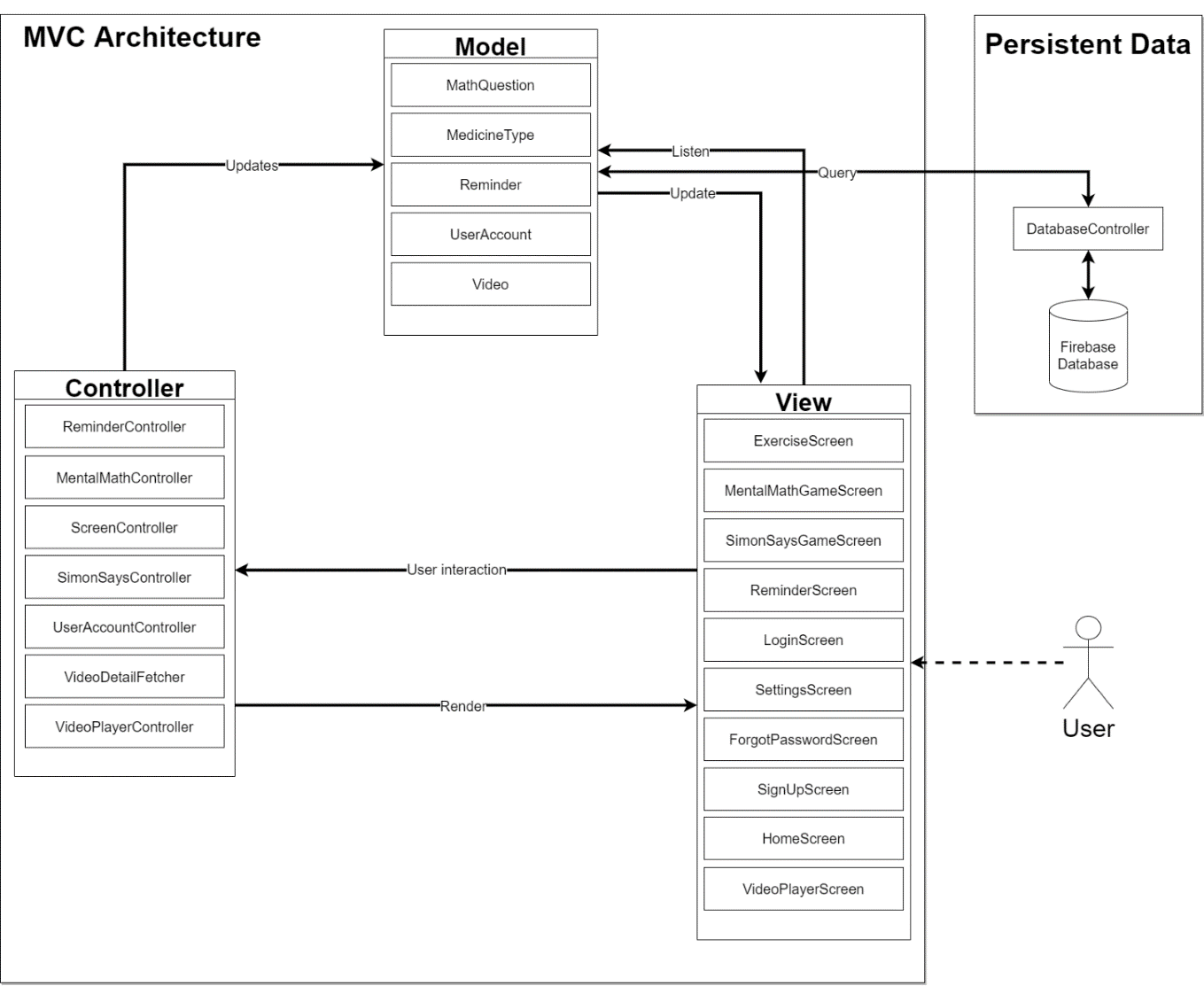
The context diagram has also been drawn to define and clarify the boundaries of the software system. The entire software system can be shown in a single process. It helps the developmental team better identify the flow of information between the system and external entities.

Figure 2: Context Diagram



The software architecture has been updated on the 9th October 2021 to ensure it’s more aligned with the MVC Architecture the team has been using in the development of the application. It also further clarifies the internal components with the Model, View and Controller.

Figure 3: Software Architecture



# ASSUMPTIONS, CONSTRAINTS, RISKS

## Assumptions

This section identifies the statements believed to be true for the successful release of the product and completion of the project.

**Technology**

* The main framework of the system is to remain unchanged after release.
* The required external dependencies, YouTube API and Firebase, are assumed to continue to be functional after the release of our application.

**Budget and Funding**

* Sufficient funds, either from GoldFolks’ parent organization or possible investors (e.g., Health Promotion Board), are acquired on time for development.
* The estimation of cost and resource allocation is accurate, and no sudden or additional large amount of cost is incurred.

**Development**

* GoldFolks’ developers are appropriately skilled and able to meet expectations and timeline of the project development lifecycle.

**Marketing**

* GoldFolks requires advertising support from the Health Promotion Board for the application to be well-known and publicized to be downloaded by our target audience, the elderly.

## Constraints

This section identifies any limitations that must be taken into consideration prior to the release of the product.

**Resources**

Users’ data on the application are still owned by the users. Thus, all the rights of using data must be clearly given prior approval by the users before being used or stored in external database or software.

**Time**

The entire project from the planning to release stage, aims to be completed in a short time frame, between August to November 2021. To ensure no delays, project planning must be cautious and correctly estimated.

**Budget**

With limited budget, proper planning and budgeting must be carried out to ensure processes do not cause a budget overrun.

**Manpower**

The team size is relatively small with only 8 members. Careful work assignment and task planning must be carried out. The project manager must exercise a structured and flexible manpower allocation to tasks to easily move developers around the different sub teams to ensure the project tasks can be completed on time.

**Functionality**

The app requires iOS 9 and above, and Android 4.4 (KitKat) and above. The Exercise Video functionality is dependent on a connection to the YouTube API and the user’s internet connection. Additionally, proper offline functionality must be maintained for medical reminders such that it is not solely dependent on the connection to the cloud Firebase storage.

## Risks

Table 2: Risk Management Log

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| No. | Risk Description | Probability | Impact | Ramifications | Mitigation of Risk |
| 1 | Project development duration is underestimated, unable to meet Project Deadline | Medium | High | Reduced functionalities or delay the deployment date. | * Potentially consider rescope of the project to prioritise important tasks to be completed first * Identify critical path and revise project schedule accordingly with additional buffer time, if possible * Follow up closely with developer team progress. |
| 2 | Hardware failure e.g., Hard Drive corrupted | Low | High | Unable to meet deadlines/release as codes cannot be written or ran. | * Maintain backups via version control software, e.g., git to minimize the impact. |
| 3 | Feature creep | Medium | High | Will delay the project and may not be able to meet the set deadlines. | * Maintain traceability between documentation and code to not go beyond the specified requirements. * Follow up closely with developers' team progress. |
| 4 | Teamwork and Communication Problems  (e.g., Updates to code not communicated to every group member) | High | Medium | Overlapping of codes which may lead to errors while running the program, and documentation may be inaccurate. | * Hold frequent meetings and discussions between members and team leads to enhance communication. * Ensure transparency of work done by maintaining a consistently updated Development Log for all members to keep up to date with the project progress. |
| 5 | Staff Unavailability | Medium | Medium | Staff's work will not be completed. | * Familiarize all team members to the project’s progress and keep a good documentation of the project’s progress and activity so the work can be easily handed to other members. * Transfer the workload to other staff workers in order to ensure that work is still completed by deadlines. |
| 6 | System not meeting requirements, developmental errors and roadblocks | Low | High | Scheduled releases may be affected as that would lead to more debugging and testing. | * Ensure that testing is done early so that the system can be checked against requirements with ample spare time. * Development log needs to be updated consistently. |
| 7 | Poor code design (e.g., high coupling) | Low | Medium | Modules cannot be separated | * Design a system architecture and class diagram and ensure that it is followed when coding the application. |
| 8 | COVID-19 | Low | Medium | Unable to meet up due to restrictions, teammates contracting the virus. | * Ensure that we follow safe distancing measures and minimize physical meetings. * Project meetings will all be held on Microsoft Teams if restrictions were to be tightened. * If a team member contracts the virus, all team members should perform a swab test and the project manager will delegate the workload of the affected team member accordingly. |
| 9 | Misunderstanding/Miscommunication of requirements from stakeholders | Low | High | Requires us to redesign our application which delays the project | * Establish a clear set of requirements and ensure that it is signed and agreed by both parties beforehand. * Before changes are to be made, derive traceability information to assess each requirement's change impact (CI). |
| 10 | Sensitive data being leaked/hacked | Low | High | Loss in customer trust and brand reputation | * Encrypt personal information given to the application so that hackers will not be able to access it. * Ensure proper crisis management procedures if data leaks were to occur. |
| 11 | Incomplete Documentation | Low | Low | May result in miscommunication between team members and may create issues in the future when maintaining the software | * Perform consistent review of existing documentation and create a list of documentation to be maintained. * Ensure that version control is in place for each document so that it is easy to track changes and verify information. |
| 12 | Inconsistent Programming Style | Low | Medium | Code will be harder to read and understand and it will be difficult for another person to maintain the software | * Code reviews to ensure adherence to proper programming style and principles e.g., names, alignment, locality, comments |

# RELEASE APPROACH

This section describes the approach to be taken for the releases of the GoldFolks application.

## Rationale

As mentioned in the Project Plan document and earlier in this document, the Agile methodology is used to develop the GoldFolks application. The main idea of Agile is to develop the program with small incremental releases in iterations by breaking down a bigger problem into multiple sub-problems. Changes, such as the implementation of new features, can be made in an incremental manner and delivered through collaboration between self-organizing cross-functional teams.

Agile also provides a constant feedback mechanism while the product is delivered. Additionally, the Agile approach testing is integrated with the project lifecycle which begins very early in the software development process. The increased frequent testing allows less effort to resolve the bugs later and a higher quality product delivered at the end of the GoldFolks project.

The Agile development approach is also selected due to the short time frame given for the project. This method will allow for changes to be easily made within the limited duration of the project, which is extremely crucial to meet and deliver the complete product by the deadline planned. Even when most requirements are well-defined at the initial stages of the project, addition and changes of functionality are not uncommon. Moreover, due to the independent nature of our individual functionalities, it would be more efficient for us to split the development of the functionality for sprint planning. Hence, the Agile methodology helps us to complete the project within the time constraint specified in the previous section.

Therefore, the Agile development methodology is the most ideal methodology and chosen over the other methodologies to ensure that the releases delivered to users are reliable and satisfy requirements. Compared to the traditional waterfall methodology, the Agile approach is a much more flexible and appropriate model to follow.

## Release Strategy

The release strategy will employ a phased function rollout. In a phased rollout, implementation of functions happens in stages. Information gleaned from earlier implementations can assist subsequent implementations so that potential issues can be minimized. This aligns well with the Agile methodology as mentioned in the above rationale as iterations are developed incrementally.

To facilitate the release process, continuous integration and the Build-Package-Deploy process will be used. Continuous integration involves merging individual components alongside development, as opposed to merging many components at once. This is a good practice that allows all team members to keep up with the current latest build, reducing time spent on miscommunication and development conflicts. The Build-Package-Deploy process includes running unit tests, generating release notes, and running integration tests before distribution of the product.

Version control software like SVN and Git are also used to manage releases. Such software provides a convenient way to view the history of changes to the product as well as to revert to a previous version of the product if need be. Branches can also be created to allow team members to work in parallel - which reduces time spent - and merge their changes in the future.

Alpha and Beta testing will be used to verify that the product satisfies requirements without issues or bugs. Feedback gained from these tests will be used to make final improvements to the product before release.

An automatic approach is used for release distribution. Installation of the product will be handled by Google Play Store and Apple’s App Store.

After alpha and beta testing, the first major release will be made. It will include some of the main functionalities of the application, which are specified in the System Requirements Specification document. Subsequent minor releases will be made based on feedback from users, which will include bug fixes and improvements in user experience. Subsequent major releases will include more main functionalities on top of the previous major release. Additionally, all releases will be internally tested by team members before they are distributed to users.

## Release Content

The following table showcases what our team will release in each version of the software application:

Table 3: Release Content Table

|  |  |  |
| --- | --- | --- |
| **Release Version** | **Release Content** | **Remarks** |
| 0.1 | Skeleton of application with basic UI design and layout. | - |
| 1.0 | Main functionality of Reminder system, Game system, and Exercise Video system. | Fully implemented as defined in SRS |
| 1.1 | Bug fixes and improvement in UI. | - |
| 2.0 | Bug fixes and notification system. | - |

## Release Schedule

The Release Schedule is an important timeline that our team will strictly adhere to in order to maintain a consistent frequency of releases. This allows us to uphold user satisfaction through constant quality maintenance and regular updates of new features. The following table states the expected delivery dates for the different release versions after thorough discussions with our clients.

Table 4: Release Schedule

|  |  |
| --- | --- |
| **Release Version** | **Expected Delivery Date** |
| 0.1 | 20/9/2021 |
| 1.0 | 4/10/2021 |
| 1.1 | 14/10/2021 |
| 2.0 | 18/10/2021 |

## Release Impacts

With every release, several components of our project are impacted. The following table summarizes how our business process and system are impacted with each release. We also state the benefits, goals and objectives that our project aims to achieve for every release.

Table 5: Release Impacts

|  |  |  |  |
| --- | --- | --- | --- |
| **Release Version** | **Impact on Business Process** | **System Impact** | **Benefits, Goals and Objectives** |
| 0.1 |  | * Initial UI screen with no functionality. | * Implementation of UI * Provides a working interface that helps the team to understand the flow of navigation through the app. |
| 1.0 | * Complete essential features of GoldFolks as a product * Announcement of product and rollout to users * Setup of channel for receiving user feedback | * Main branch now ready for new changes. * UI elements must be functional. Elements may be changed to better suit the functionality specified. * Requires establishment of cloud Firebase. Proper encryption of user email and password must be implemented. | * Implementation of main functionality according to functional requirements. * First major release to users. * Feedback from users used for bug fixes and improvement. |
| 1.1 | * Announcement of minor release to users * Improvement of product value through bug fixes | * Minor release branch tagged, and release deployed to user * Improve reliability of the application. | * Bug fixes and UI enhancements. * Achieves a higher level of usability and clarity. * Increase user satisfaction |
| 2.0 | * Further refine product value through new features * Announcement of major release to users | * Parallel update of main and release trunks in Git version control. * Requires system notification permissions on the user’s device. | * Implement notification system for Medication Reminder. |

## Release Notification

Table 6: Release Notification

|  |  |  |  |
| --- | --- | --- | --- |
| **Stakeholders** | **Method of Notification** | **Notification Information** | **Timeframe for Receipt of Notification** |
| Users | Future updates will be listed ahead of time on the application page description.  Push notifications will be handled by the iOS and Android app stores when the new version is pushed to the respective stores. | Changes to be made, including bug fixes, changes to UI, and new features, if any. | 7 days prior to release version for description update. Push notification will be done when the update is available. |
| Development Team | Email, instant messaging, and meetings | Same information as provided to users. Additional information regarding reasons for changes and any changes to documentation, architecture etc. | Immediately after the plan for a release is approved. |

# GLOSSARY

# Table 7: List of terms and definitions

|  |  |
| --- | --- |
| **Term** | **Definition** |
| Agile | A set of practices centered around the idea of iterative development, where requirements and solutions evolve through collaboration between self-organizing cross-functional teams. |
| Build-Package-Deploy | A standard procedure involving building the application, packaging the application into a software suitable for users to install and deploying it to the users. |
| Firebase | A platform created by Google for developing mobile and web applications. |
| Flutter | An open-source UI software development toolkit created by Google. |
| Git | A free and open-source distributed version control system. |
| GoldFolks | The name of the mobile application being developed as part of this project. |
| Release | A defined version of the application made available for users. |

# ACRONYMS

# Table 8: List of acronyms and definitions

|  |  |
| --- | --- |
| **Acronym** | **Definition** |
| SVN | Subversion |
| UI | User Interface |

# 

# APPENDICES

# Table 9: List of referenced documents links

|  |  |
| --- | --- |
| Document Name | Links |
| System Requirement Specifications | <https://172.21.149.196/svn/3002/TS3/ElevenDegree/Lab%202/GoldFolks%20SRS.pdf> |
| Quality Plan | <https://172.21.149.196/svn/3002/TS3/ElevenDegree/Lab%202/Quality_Plan_(Lab_2).pdf> |
| Risk Management Plan | <https://172.21.149.196/svn/3002/TS3/ElevenDegree/Lab%203/GoldFolks%20Risk%20Management%20Plan.pdf> |
| Project Plan | <https://172.21.149.196/svn/3002/TS3/ElevenDegree/Lab%203/GoldFolks%20Project%20Plan.pdf> |