Logo

Description automatically generated

GoldFolks

*Test Plan*

***Version 1.0***

***30th October 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*

VERSION HISTORY

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Version #** | **Implemented**  **By** | **Revision**  **Date** | **Approved**  **By** | **Approval**  **Date** | **Reason** |
| 0.1 | Low Jin Teng Jackson, Zachary Varella Lee Zheyu, Anil Ankitha | 23/10/2021 | Chan Shao Jing | 23/10/2021 | Initial Test Plan Draft |
| 1.0 | Low Jin Teng Jackson, Zachary Varella Lee Zheyu, Anil Ankitha | 30/10/2021 | Chan Shao Jing | 30/10/2021 | Final Test Plan |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |

[VERSION HISTORY 2](#_Toc86448134)

[Test Plan Identifier 4](#_Toc86448135)

[Introduction 4](#_Toc86448136)

[Test Items (Functions) 4](#_Toc86448137)

[Software Risk Issues 5](#_Toc86448138)

[Features to be Tested 5](#_Toc86448139)

[Features not to be Tested 6](#_Toc86448140)

[Approach (Strategy) 6](#_Toc86448141)

[Item Pass/Fail Criteria 7](#_Toc86448142)

[Suspension Criteria and Resumption Requirements 9](#_Toc86448143)

[Test Deliverables 9](#_Toc86448144)

[Remaining Test Tasks 10](#_Toc86448145)

[Environmental Needs 10](#_Toc86448146)

[Staffing and Training Needs 10](#_Toc86448147)

[Responsibilities 10](#_Toc86448148)

[Schedule 11](#_Toc86448149)

[Planning Risks and Contingencies 11](#_Toc86448150)

[Approvals 12](#_Toc86448151)

[References 12](#_Toc86448152)

# Test Plan Identifier

This test plan’s identifier will be TestPlan-V1.0-M.

The test plan identifier contains three sections delimited by hyphens. The first section is the document identifier, “TestPlan”. The second subsection, “V1.0” indicates the test plan level. The last section indicates the type of test plan, where “M” stands for Master plan, “L” stands for Level plan, and “I’ stands for Integration plan.

# Introduction

The purpose of this plan is to identify and draft the procedures used in the testing of the GoldFolks application as part of the Master Test Plan. The following sections will discuss the test items, features to be tested, and the overall approach.

We will assign team members to the various types of testing listed to ensure each member has individual responsibility for their own sections.

Information on test cases and coverage can be found in a separate document.

# Test Items (Functions)

|  |  |
| --- | --- |
| **Functional Testing** | |
| **ID** | **Item** |
| 1 | Mental Math game |
| 2 | Simon Says game |
| 3 | Login |
| 4 | Medication reminders |
| 5 | Exercise video |
| 6 | Firebase backend |
| 7 | Navigation |
| **Non-functional Testing** | |
| **ID** | **Item** |
| 8 | Timing testing |
| 9 | Stress testing |
| 10 | Pilot testing |
| 11 | Acceptance testing |
| 12 | Installation testing |

# Software Risk Issues

The following is a list of possible software risk issues that we could encounter:

* Potential changes that might impact certain functionalities of the software application
* Ability to use and understand a new package/tool, etc.
* Poor maintenance of certain complex functions
* Modifications to components with a past history of failure
* Poorly documented modules or change requests

Unit testing can allow us to identify potentially risky components of our applications based on the quantity and magnitude of defects found. Thus, it is important to maintain records and pay special attention to these components to ensure the software risks are contained to a minimal level.

# Features to be Tested

The following is a list of features to be tested from the user’s perspective of what the system does.

|  |  |  |
| --- | --- | --- |
| **Feature** | **Level of Risk** | **Explanation** |
| Mental Math | High | Core feature of the application |
| Simon Says | High | Core feature of the application |
| Exercise videos | High | Core feature of the application |
| Medication reminders | High | Core feature of the application |
| Login | High | Required for users to access the application’s main features |
| Updating data to database | Medium | Not crucial for main functionality, mainly used for backup of data and updating game high scores |

# Features not to be Tested

The following is a list of features that will not be included in the test suite in this version of the test plan.

|  |  |
| --- | --- |
| **Feature** | **Explanation** |
| Sudoku Game | This feature will be tested and released in a future version of the product. |

# Approach (Strategy)

This section discusses the approach and standards used during the unit testing for the identified items. GoldFolks will use both white-box testing and black-box testing, before conducting acceptance and installation testing. White-box testing will be conducted for both functional testing and performance testing. Black-box testing will be used to conduct pilot testing to verify the functionality of the combined components. Finally, acceptance testing and installation testing will be conducted to ensure the application is user accepted and able to be compiled and deployed to different types of environments.

**White-Box Testing**

1. Functional Testing

Under functional testing, each specified component in Section 3 Test Items will be tested. Cyclomatic complexity will be used to identify the number of control flows to test. The testing will involve testing that each identified control flow, given a specified input, will return the correct output.

1. Performance Testing

Two aspects of performance testing will be covered: Timing Testing and Stress Testing. Timing testing will be conducted by measuring the response time of the application upon button presses during navigation. The response times will be logged and recorded to identify any unacceptably long wait times. For stress testing, we concentrate on stressing the application with the maximum number of users to find network issues and bottlenecks in performance. Metrics such as hit time and time to receive the first byte will be used to measure performance under stress testing.

**Black-Box Testing**

1. Pilot Testing

The app will be distributed to a set of end users and asked to evaluate the application in terms of each of the key functionalities identified in Features to be Tested. Feedback will be collected regarding usability and bugs.

1. Acceptance Testing

After the pilot testing, acceptance testing will be conducted. The application will be distributed to members of the team as well as people outside the development team. Users will be provided with a platform to provide feedback regarding the quality of the app as well as any bugs that occur.

1. Installation Testing

Following acceptance testing, once the QA team has validated the software and any new changes to the application, the development team will deploy the latest version of the application to the Google Play and Apple App Store for end users to install and utilize, collecting feedback on the installation process from reviews.

# Item Pass/Fail Criteria

|  |  |  |  |
| --- | --- | --- | --- |
| **Functional testing** | | | |
| **ID** | **Item** | **Pass criteria** | **Fail criteria** |
| 1 | Mental Math game | The game tutorial is displayed correctly. The game can be played and exited without issues. High scores and leaderboard are updated and displayed correctly. | Unable to display tutorial, unable to load the game, game bugs, unable to exit the game, unable to update or show high score and leaderboard. |
| 2 | Simon Says game | The game tutorial is displayed correctly. The game can be played and exited without issues. High scores and leaderboard are updated and displayed correctly. | Unable to display tutorial, unable to load the game, game bugs, unable to exit the game, unable to update or show high score and leaderboard. |
| 3 | Login | Creates new account on command, verifies login credentials correctly, sends emails for resetting passwords. | Fails to create and store new account in database, unable to verify login credentials correctly, fails to send email for resetting passwords, fails to store new password in database. |
| 4 | Medication reminders | Creates, edits, deletes reminders correctly. Displays a list of existing reminders correctly. | Unable to create, edit, or delete reminders. Unable to display existing reminders correctly, unable to store reminder data accurately. |
| 5 | Exercise videos | Displays list of available exercise videos. Fetches video from YouTube and plays it. Playback controls function as intended. | Unable to display list of videos, unable to display video, playback controls fail to work. |
| 6 | Firebase backend | Stores and updates (new) account data. Stores, updates, and deletes medication reminders appropriately. Stores and updates leaderboard data for games. | Fails to store, update, or delete data correctly. |
| 7 | Navigation | Navigation to selected screen functions as intended and display the correct screen every time. | Fails to display intended navigated screen. |
| **Non-functional testing** | | | |
| **ID** | **Item** | **Pass criteria** | **Fail criteria** |
| 8 | Timing testing | App navigates within 0.5 seconds in 10 trials. | App takes longer than 0.5 seconds on average to navigate in 10 trials. |
| 9 | Stress testing | Connection to app database is established within 25 seconds when under 85% load. | Connection is established in longer than 25 seconds. |
| 10 | Pilot testing | No major defects are reported during pilot testing. | When either of the functionality do not work as intended. Major defects that cause a failure of the application itself are reported. |
| 11 | Acceptance testing | No major defects are reported during acceptance testing. | When either of the functionality do not work as intended. Major defects that cause a failure of the application itself are reported. |
| 12 | Installation testing | App achieves at least a 4.0 rating on the app store. | App achieves less than a 4.0 rating on the app store. Failures in installation are reported by users on the app store. |

# Suspension Criteria and Resumption Requirements

Defects that affect navigation or the user interface to a large degree, such as preventing navigation to the main features of the application, warrant stoppage of testing. Testing should only resume once these navigational defects are fixed.

# Test Deliverables

The deliverables of this test plan include:

* Test plan document
* Test cases and requirement test cover report
* Test tools and outputs
* Error logs and execution logs
* Problem reports and corrective actions

# Remaining Test Tasks

For future versions of the product, the following tests will be done.

* User acceptance testing for improvements in user experience
* Functional testing on new features

# Environmental Needs

These are the special requirements for the test plan.

The Firebase server housing the database for the application must be online and running. The YouTube API used by the application and YouTube itself must also be functional during testing.

# Staffing and Training Needs

Training will be required for the staff to be familiar with the testing procedures and software required. Our Lead Developer has briefed the QA team on the current state of the application and the overall structure of our program to assist the QA team with writing the testing scripts and performing use-case validation.

The QA Engineer will subsequently be responsible for providing the training and disseminating the information on running the application and the structure of our code. During the User Acceptance Testing (UAT) phase, however, minimal training and guidance will be provided to the users for us to better understand the user flow and ways to improve on it.

# Responsibilities

The following is the list of responsibilities related to the Test Plan, their description, and the assigned team member in charge of assuring the responsibility is carried out.

|  |  |  |
| --- | --- | --- |
| **Responsibility** | **Description** | **Assignee** |
| Setting risks | Identifying and recording risks and risk levels of items to be tested. | Low Jin Teng Jackson |
| Selecting features to be tested | Identifying and sorting crucial features of the application to be tested. | Zachary Varella Lee Zheyu |
| Setting overall strategy | Designing an approach and process of testing procedures. | Chan Shao Jing |
| Ensuring tools required for testing are available | Ensuring external software/hardware is available for testing. | Chong Yow Lim |
| Resolution of scheduling conflicts | Ensuring that scheduling conflicts are minimized and resolved in a timely manner. | Ng Chi Hui |
| Providing training | Mentoring and training of team members regarding testing tools and the system. | Lionel Wong Zhi Neng |
| Critical decisions regarding test items | Deciding if items not covered in the test plan require testing or not. | Anil Ankitha |

# Schedule

The testing schedule is to be aligned with the plan according to the Gantt chart, updated by the project manager. The Gantt chart will be planned such that sufficient time is allocated for the team to test the application and its components. In the event of a delay, time will be allocated in the form of a buffer week to avoid delaying the product. Furthermore, the following measures will be undertaken to avoid and mitigate delays:

1. QA Manager will do routine check-ins weekly with the Lead Developer on progress and changes to components in order to ensure components are progressing according to schedule and are able to be tested according to the Test Plan.
2. Communicate holes in documentation or conflicting specifications to the Project Manager to prevent delays in development.
3. Lead Developer will ensure their team develops the product according to the SRS and avoids unnecessary feature creep.

# Planning Risks and Contingencies

The following are the identified risks that may impact the Test Plan and suggested contingencies should they occur.

|  |  |
| --- | --- |
| **Risk** | **Contingency** |
| Late delivery of software or testing tools | Additional buffer time will be used to account for the late delivery of testing tools. |
| Lack of personnel to cover the testing when it begins | Re-allocation of manpower who have completed developmental work/documentation work to assist the testing team. |
| Change to the original requirements or design | The test and development schedule will move out an appropriate number of days into the buffer week to account for the new changes. |
| Insufficient testing software, data, or tools | Testing team will ensure that redundant testing software/suites exist in the event one fails or exceeds budget. |

# Approvals

The testing team is required to deliver a full report on the status of the application after final testing. The Quality Assurance Manager is responsible for vetting this report before submission to the Project Manager for final approval. The report will then be shown to clients and various stakeholders to verify that the product meets business requirements.

All parties are required to provide their signatures to indicate their approval.

|  |  |
| --- | --- |
| **Stakeholder** | **Signature** |
| Quality Assurance Manager |  |
| Project Manager |  |
| Client |  |

# References

* Test Cases and Test Coverage Plan v1.0
* SRS
* Project Timeline (Gantt chart)