**Test Plan**

**Created For: GreatCall**

**Created By: Mukesh**

**Date: April 09, 21**

**Version 1.0**

**VERSION CONTROL**

|  |  |  |  |
| --- | --- | --- | --- |
| **Date** | **Version** | **Description** | **Author** |
| April 09, 2021 | 1.0 | Initial draft | Mukesh |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

**TABLE OF CONTENTS**

1. Introduction 4

2. Purpose 4

3. In Scope 4

4. Out of Scope 4

5. Software/Network Requirements 4

6. Test Approach 4

7. Types of Testing 4

8. Testing tool to be used 5

9. Features to be tested 5

10. Features not to be tested 5

11. Estimated Time 5

12. Environment Setup 5

13. Establish Test Criteria 5

14. Roles and Responsibility 5

15. Questions to be asked during grooming call 6

16. Concerns to be raised before working 6

17. Release Control 6

18. Review and Approvals 6

19. Risk 6

20. Communication 7

# Introduction

GreatCall is a leader in connected health for active aging and a wholly-owned subsidiary of Best Buy. With health and safety solutions for older adults and their family caregivers, GreatCall’s innovative suite of easy-to-use mobile products and award-winning approach to customer care helps aging consumers live more independent lives.

Products and services include: Lively Mobile, Jitterbug Flip, Jitterbug Smart2, Lively Wearable, Lively Home and health, safety and wellness apps such as Urgent Care, GreatCall Link, MedCoach and 5Star Urgent Response Service

# Purpose

The Purpose of this document is to outline the requirements for the testing. This document will be used for the test planning and test strategy. This document shall be completed and used by the project test team to guide testing activities will be managed for this project. The test effort will be prioritized and executed based on the project priorities as defined in the Project.

# In Scope

The scope of this project is limited to manual testing. Functional testing is in scope and need to be tested

# Out of Scope

QA support for mobile application. Only web service will be tested

# Software/Network Requirements

* Browser(Chrome/Firefox/IE/Safari)
* Internet connectivity

# Test Approach

The test approach here defines the scope and general direction of the test effort. It is a high-level description of the overall process of testing. This includes the types of testing which will be performed or selecting the right test tool that fits the requirement.

# Types of Testing

* E2E Testing/ Functional
* Regression Testing
* Sanity Testing
* Retesting
* Smoke Testing/Build Acceptance Testing (BAT)
* Compatibility Testing

# Testing tool to be used

* Bug Reporting Tool

# Features to be tested

* User Story 1, 2 and additional stories in the epic
* Other features will be defined later

# Features not to be tested

To be defined later

# Estimated Time

* To test User Story 1, 2 and additional stories, it will require 1 hour including all positive and negative test cases
* Estimated time for rest of the features to be defined later.

# Environment Setup

To be defined later

# Establish Test Criteria

**Suspension Criteria:** It will define the benchmark for suspending all tests. If QA team found that 50% of all test cases have failed, then all testing will be suspended until the developers resolve all of the bugs that have been identified so far.

**Exit Criteria:** It will define the benchmark that will signify the successful completion of a test phase or project before a particular feature or portion of the software can be considered suitable for public use.

# Roles and Responsibility

The test roles and their responsibilities are detailed in the table below.

|  |  |
| --- | --- |
| **Role** | **Responsibilities** |
| QA Manager | Oversees **QA processes of the project, involved in Testing planning, monitoring, and control, reporting and collecting.** |
| Test Lead Software Quality Engineer | * Test Lead Software Quality Engineer Manages and tracks the software test planning and testing. Will be responsible for making sure that the team collectively understands what work needs to be delivered. * Will be responsible for determining what testing is appropriate. |
| Software Quality Engineer | Software Quality Engineer Creates and Execute cases and log defects. Validate repaired defects. |
| Software Engineer | Software Engineer is Software developer of the Project |

# Questions to be asked during grooming call

* What is the typical age of a user story in the product backlog?
* How often the product backlog is being groomed?
* in parallel to backlog grooming, will share the status of the user stories which is still in progress
* How long does the grooming of a typical user story take?
* How user stories are being created? (Is it a joint team effort with the PO or is the product owner writing the user stories and the team estimates them?)
* How likely effort of a user story is being estimated?
* What if the team shares different opinions for time estimation? How this process is handled
* Is poker planning meeting arranged to discuss estimations?
* What is a typical distribution of story sizes in the sprint backlogs?
* Are we re-estimating user stories at the end of a sprint?
* What type of work environment do we want to create on this project?
* What can we do as a team to make sure that we support each other to achieve our team goals?

# Concerns to be raised before working

* To make sure test plan is reviewed and approved by the PM
* To make sure the test case document is being reviewed and approved by the PM
* To share the time estimation before starting on any task
* To identify the risk in starting and share it with team
* To get the requirement clear in starting
* To share the functionality which should be tested and what not(If any)

# Release Control

Release management plan with appropriate version history will be maintained that will make sure test execution for all modification in that release.

# Review and Approvals

This document is to be reviewed by the QA team and the Project Manager, further PM will approve this document.

# Risk

Risk identification, includes anything that can possibly render the deployment ineffective, ultimately prevents loss of ROI. Risks may include:

* Tight timelines
* Undefined project scope
* Insufficient resources
* Continuously changing requirements
* Natural disasters

# Communication

Communication channels to be added here.