11/2/2017

Quality Assurance Plan

Group 11

Table of Contents

1 Revision History

TABLE 1 REVISION HISTORY

Version	Author	Publication / Revision Date	Description of Change
1	Gary (Sheung Yau)	Oct. 13th, 2017	Draft
1.1	Jia Hui (Mandy) Xiao	Oct. 16th, 2017	Draft - Reviewed
2	Ryan Kiew Ruelt Yean	Nov. 2nd, 2017	Added version 1 testing in Appendix

2 Introduction

Prep is an iPad-based mobile application that helps children with Autism adapt to a new environment. This app is written with iOS frameworks using swift language. The QA team is responsible for testing the product and ensuring it meets the requirements.

The test plan has been created to describe the test approach to team members. It includes the objectives, schedule, tools, and approach. This document also includes specific times and dates for the test deliverables.

2.1 Quality Objectives

The objective of testing before each release is to assure that the system meets the full functional and non-functional requirements. At the end of each development cycle, the project manager should find that the features planned for development are complete as specified in the functional requirements. Additional objective of testing is to assure that all issues and risks associate with the new version has been identified and exposed by the tester. Test results should be delivered to the team prior to the start of next cycle. This requires careful testing of the system and good communication from the QA side.

3 Test Approach

The team is using an agile approach to test the product. A week before each release date, a new version will be delivered to the team and will be tested against the functional requirements.

Tests will be created in accordance with the planned functionality and each test case will be marked as *pass* or *failed* on a google spreadsheet. We will call it the *requirements checklist*. When a bug is found during test activities, the tester will create a bug on Microsoft Planner and assign it to the developer that is responsible for fixing the bug.

As *Experience View* is created using 3d engines, the team will rely heavily on manual black box testing along with their experience with similar technologies to discover performance issues and any defects that cannot be caught with test tools. Exploratory testing will be carried out at the end of each cycle.

3.1 Automatic Unit Testing

The XCode project is set up to have the Unit Testing Bundle included. No automated tests are planned at this time but the goal is to write tests for all methods to achieve full code coverage. Test script will validate extreme scenarios one at a time. We will use automatic unit tests to check for boundary conditions and exercise each method with different input values.

3.2 Integration Testing

The purpose of integration testing is to assure that each module within the application is functioning correctly at all levels. We will be using a top-down approach to test the system incrementally, meaning that we will make sure that new features introduced to the system interacts well with other existing features. This is done by testing the overall system and validating the data sent by new modules through the user interface.

4 Testing Tools

TABLE 2: TOOLS FOR TEST ACTIVITIES

Test Activities	Tools
Bug Reporting	Microsoft Planner
Bug Tracking	Google Sheets
Test case planning	Google Sheets
Checklist creating	Google Sheets
Project size measuring	Cloc – Library on Github

5 Roles and Responsibilities

TABLE 3: TEAM MEMBERS

| Role	Staff member	Responsibilities
Project Manager	Jia Hui (Mandy) Xiao	1. Acts as product owner 2. Responsible for project schedule
QA Lead	John Kyung Joon Ko	1. Coordinate test activities 2. Report progress to PM 3. Create test plan
QA	Ryan Kiew Ruelt Yean	1. Understand requirements 2. Writing and Executing Test Cases 3. Bug reporting and tracking 4. Regressing Testing

6 Project Size and Complexity

An estimation of the number of lines of code to be written in each iteration is shown below.

TABLE 4: A BREAKDOWN OF THE NUMBER OF LINES OF CODE TO BE WRITTEN FOR EACH SCREEN

Version	Screen	# of lines of code to implement	# of lines of code to test	Total for each version
1	Experience Viewer	300	200	500
2	Home Page	150	50	500
	Experience Editor Start Page	200	100	
Final	Experience Editor Page	300	100	650
	Login Page	150	100	

FIGURE 1: CUMULATIVE NUMBER OF lines of original code written by the team

FIGURE 1: CUMULATIVE NUMBER OF lines of original code written by the team
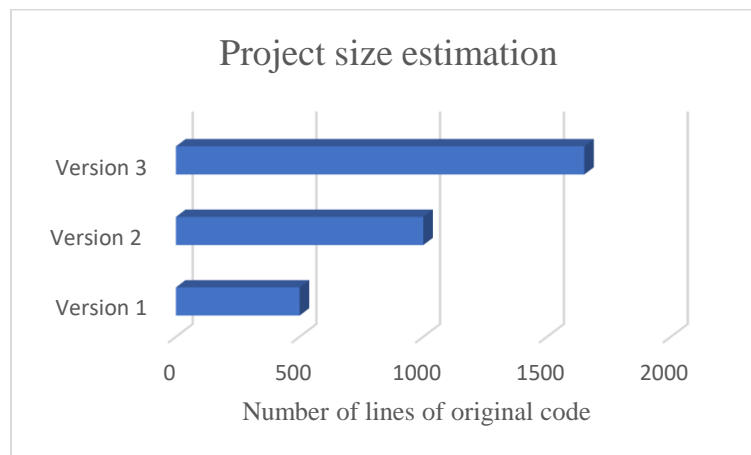
7 Bug Reporting

A critical problem, which prevents the user from completing the flow must be reported immediately to the PM and to the team in order to implement fixes as soon as possible. A major or minor bug must be assigned to the developer who is responsible for fixing the bug after it has been created on Microsoft Planner. All bugs should be entered and tracked via Google Sheets by the QA team.

## 7.1 Bug Severity and Priority Definition

TABLE 5: BUG SEVERITY

Severity	Description
Critical	the bug causes non-recoverable errors (crashes); there is no workaround to the problem, user cannot continue with flow
Major	the bug may only affect certain parts of the system; the bug can have a work around but needs to be fixed soon otherwise requirement is not met
Minor	the bug will impact the product in a way negative way but does not hamper its functionality
Trivial	The bug impacts user experience or ties to cosmetic issues

TABLE 6: BUG PRIORITY

Priority	Description
Critical	the bug must be fixed immediately; the product cannot be shipped
High	the bug must be fixed in the next release
Medium	The problem can be fixed in later versions
Low	The problem is out of the current scope, it is an enhancement

7.2 Bug Life Cycle
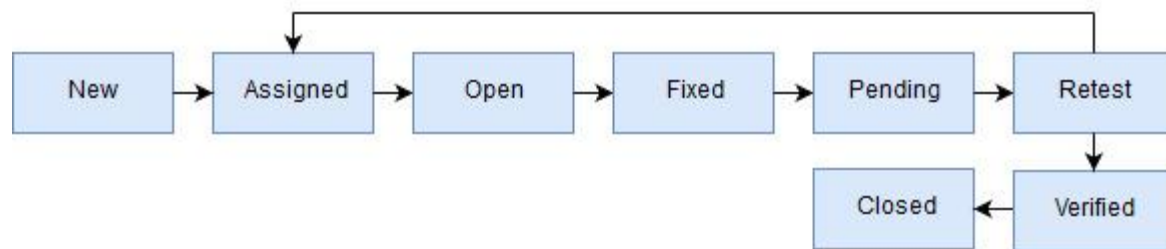
8 Test Environment

- Support level:
  - iPad Devices with iOS11+: iPad Air, iPad Air 2, iPad (2017), iPad mini 2, iPad mini 3, iPad mini 4, iPad Pro, iPad Pro (2017)
- Available Resource: one iPad Air with iOS 11 available for tester on the team

9 Test Schedule

TABLE 7: TEST SCHEDULE

Tasks	Start Date (mm/dd/yy)	End Date (mm/dd/yy)	Effort	Comments
Test Planning	10/21/17	10/27/17	6 d	
Review requirements documents	10/21/17	10/22/17	2 d	
Create test basis	10/23/17	11/24/17	2 d	
Test on Staging	10/30/17	11/02/17	4 d	
Version 1 release	11/06/17			
Test on Staging	11/13/17	11/16/17	4 d	
Version 2 Release	11/20/17			
System testing	11/21/17	11/24/17	4 d	
User Acceptance Testing	11/21/17	11/21/17	1 d	
Performance testing	11/25/17	11/26/17	2 d	
Regression testing	11/27/17	12/02/2017	6 d	
Final Release	12/04/2017			

10 User Acceptance Test

The purpose of User Acceptance Testing is to receive feedback from users to make final adjustments to the system and to assure that the system satisfies the use case scenarios.

The UAT will be done only when all the functional requirements have been completed and no outstanding critical defects has been found. A checklist for ALL pages in the user flow to be reviewed must be created prior to testing – including Login Page, Home Page, Experience Viewer Page, Experience Editor Start Page, and Experience Editor Page. Exit criteria is the following: all UAT tests run to completion and all defects found during the UAT are recorded using the Defect Tracking Form below.

TABLE 8: DEFECT TRACKING FORM

Defect #	Tester	Description	Date	Severity (1-5)	Status

As an example, a checklist for the Home Page is provided below.

TABLE 9: ACCEPTANCE TESTING CHECKLIST FOR HOME PAGE

Acceptance Testing Checklist	Pass/Fail	Date	Initials
Does each button navigate to the correct page?			
Does each icon appear clearly and correctly?			
Does the menu bar appear at the top of the page?			
Are the appropriate menu items available?			
Does edited experience appear?			
Does all text appear in correct format?			

11 Terms and Acronyms

TABLE 10: TERMS AND ACRONYMS

Term or Acronym	Definition
PM	Project Manager
QA	Quality Assurance
UAT	User Acceptance Testing

12 Approvers List

Name	Role	Approver / Reviewer	Approval / Review Date
Jia Hui (Mandy) Xiao	Project Manager	Reviewer	Oct. 18th, 2017