Software Test Plan

for

Morning Star: Event Planner app

Version 1.00

Daniel Baker

DeMarcus Edwards

Kunj Patel

University of Virginia’s College at Wise

Department of Mathematics and Computer Science

December 6th 2016

Table of Contents

Table of Contents ii

Revision History iii

1. Introduction 1

1.1 Objectives 1

1.2 Testing Strategy 1

1.3 Scope 2

1.4 Reference Material 2

1.5 Definitions and Acronyms 3

2. Test Items 3

2.1 Program Modules 3

2.2 User Procedures 4

2.3 Operator Procedures 4

3. Features 4

3.1 Features to be Tested 4

3.2 Features Not to be Tested 4

4. Approach 4

4.1 Component Testing 4

4.2 Integration Testing 5

4.3 Interface Testing 5

4.4 Regression Testing 5

4.5 Acceptance Testing 5

4.6 Beta Testing 5

5. Pass/Fail Criteria 6

5.1 Suspension Criteria 6

5.2 Resumption Criteria 6

5.3 Approval Criteria 6

6. Testing Process 6

6.1 Test Deliverables 6

6.2 Testing Tasks 6

6.3 Responsibilities 7

6.4 Resources 7

6.5 Schedule 7

7. Environmental Requirements 7

7.1 Hardware 7

7.2 Software 7

7.3 Security 7

7.4 Tools 8

7.5 Publications 8

7.6 Risks and Assumptions 8

8. Change Management Procedures 8

9. Test Procedures 8

10. Requirements Matrix 9

Revision History

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Date** | **Reason For Changes** | **Version** |
| Kunj Patel | 12/6/16 | Initial Creation | 1.00 |
| DeMarcus Edwards | 03/01/17 | Core Data Base implemented | 2.00 |

# Introduction

This Software Test Plan (STP) will provide an overview of testing procedures and information for the Event Planner Application, also known as Morning Star. It will be referred to as Morning Star, or simply, the application, henceforth. The plan will identify items to be tested, the features to be tested, and the types of testing to be performed.

Primary focus of this testing plan will be to make sure that all the implemented functionality works correctly.

## Objectives

* Identify existing project information and the software that should be tested.
* List the recommended test requirements (high level).
* Recommend and describe the testing strategies to be employed.
* List the deliverable elements of the test activities.

## Testing Strategy

This application will be tested using white box testing method mostly. All the functionalities to be tested are per the test cases which will be provided by the testing team. Testers can test the app either on a physical iPhone, or a built-in simulator that xcode provides.

Refer to section 5 of this document for a detailed list of specific test plans.

## Scope

Scheduled and unscheduled changes will be managed via GitHub. GitHub is the remote hosting service used for the morning star project. Git is a source control tool.

Since apple requires only premium developer accounts to publish their app on the app store, we will be pushing out this app to the app store just yet. For the testing purposes, we will run this app either on iPhone simulator that xcode provides or we will use a physical iPhone and execute the app on it.

## Definitions and Acronyms

* API – Application Programming Interface. A set of routines, protocols, and tools that govern a software specification.
* GUI – Graphical User Interface. An interface that allows users to interact with electronic devices through icons and visual indicators.
* Morning Star – The name of the Event Palnner Application. The project that this STP is for.
* OS – Operating System. The software that the product runs on.
* SDD – Software Design Document
* SRS – Software Requirements Specification.
* STP – Software Test Plan

# Test Items

* Functionality off the app

## User Procedures

Due to the small size of the project, any user documentation will be re-read by the team and manually edited to reflect the changed or correct information.

# Features

## Features to be tested

* Is the button clickable?
* Does the button navigate to the correct view?
* Content displays correctly on the screen?
* User can scroll down the page?
* User should be able to click anywhere to hide the keyboard
* User should be able to enter in text fields
* User should be able to clear all data inside the text fields using the “clear button”
* User should be able to able to save all the entered information using the button “Save”
* User should be able to navigate through the app using navigation button items
* User should be able to access the add screen from outside the application(3d touch)
* User should be able to share events through 3rd party and apple mediums such as notes, mail, twitter etc

## Features Not to be Tested

* Whether the application can be downloaded wirelessly

# Approach

As mentioned in above sections, tester will be given test cases and they will follow along the list and test each individual requirement. User will user xcode simulator or a personal iPhone to test this application. Due to simplicity of this app, the detailed approach is not necessary since the user will be able to follow along the UI easily.

## Integration Testing

Testing done on project morning star shows that it is fully compatible with apple hardware that has been released after iOS 7. The original iPad however is not compatible with xcode therefore morning star cannot be ran on the IPAD. IPhone generation 5 and up should be able to run the application according to tests.

## Interface Testing

It has been determined that due to the limited GUI elements in the Morning star project, that most UI elements will be tested manually by the team. The focus of UI testing will be too utilized to verify GUI states (does it save its state when it is moved from background to foreground and visa-versa?) however.

## Regression Testing

Testing has shown updates to morning star especially 3D touch have be affected by changing the application to run on core data. Further tests have shown that this problem was solved by changing the schema and taking 3d touch out of the navigation controller.

## Acceptance Testing

The users have overwhelmingly accept as a replacement for calendar’s create event.

## Beta Testing

Since we cannot put the app on the app store, beta testing will not be offered for this project.

# Pass/Fail Criteria

If the functionality of an app does what is supposed to do, it is safe to say that it has pass the test. If the functionality partially works or fails completely, it will fail the test.

## Approval Criteria

All the functionality supposed to work the way it is supposed to. No partial correctness of the functionality is accepted.

# Testing Process

## Testing Tasks

Test the functionality of the application, constraints of the functions, also compatibility with older iOS devices.

## Responsibilities

Capstone Project group

* DeMarcus Edwards
* Daniel Baker
* Kunj Patel

# Environmental Requirements

<Specify both the necessary and desired properties of the test environment including the physical characteristics, communications, mode of usage, and testing supplies. Also provide the levels of security required to perform test activities. Identify special test tools needed and other testing needs (space, machine time, and stationary supplies. Identify the source of all needs that is not currently available to the test group.>

## Hardware

* An IOS device (phone or tablet) is needed for testing the application.
* The device must have touch support or mouse and keyboard support.
* Testing can be performed on an emulated device so long as it meets the above specifications as well as the software specifications below.

## Software

* The device must be running IOS 8 or higher.

## Tools

Software tools used to test Project morning star include xcode, and the hardware tools used to test the project include the iPad, iPhone 6, and the iPhone 7.

## Risks and Assumptions

It is assumed that the user is familiar with the iPhone and its functionality. For 3d touch compatibility it is assumed that the user is using iPhone 6s and above. It is also assumed that the user has enough storage space on their phone to run the application and create new events. Since the data is stored on the users phone the risk is that the user may not have the required storage space necessary to run the application.

# Change Management Procedures

The change mangagement process, is that our group meets to discuss problems with the application. After searching for solutions to the problem the team selects the most desirable solution and the pushes the new version of the application on github.

# Test Procedures

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Test Procedure Number:** | | **1.0** | | |
| **Date Tested:** | | **12/8/16** | | |
| **Test Performed By:** | | **Kunj Patel** | | |
| **Project Name:** | | **Morning Star** | | |
| **Software Version:** | | **1.0** | | |
| **Related Requirements:** | | **N/A** | | |
| # | Test Case | | Expected Result | Passed |
| **1** | **User shall be able to add new event** | | **User adds a new event** | **P** |
| **2** | **User shall be able to delete event** | | **User deletes an event** | **P** |
| **3** | **Keyboard should hide when pressed “Return” key** | | **Keyboard hides** | **P** |
| **4** | **Keyboard should hide when user scrolls the view controller** | | **Keyboard hides** | **P** |
| **5** | **Data should be stored until user deletes them.** | | **Data is stored permanently even if the user closes the app and reopens it** | **P** |
| **6** | **Clear button should clear all the text fields** | | **Upon that button pressed, all text field will go back to its initial state** | **P** |
| **7** | **Save button should create an event** | | **Save button will store the data in an array and save it** | **P** |
| **8** | **View event button should navigate to different controller** | | **It should navigate to the tableViewController and display events created** | **P** |
| **Test Procedure Number:** | | **2.0** | | |
| **Date Tested:** | | **12/8/16** | | |
| **Test Performed By:** | | **DeMarcus Edwards** | | |
| **Project Name:** | | **Morning Star** | | |
| **Software Version:** | | **2.01** | | |
| **Related Requirements:** | | **N/A** | | |
| # | Test Case | | Expected Result | Passed |
| **1** | **User shall be able to add new event** | | **User adds a new event** | **P** |
| **2** | **User shall be able to delete event** | | **User deletes an event** | **P** |
| **3** | **Keyboard should hide when pressed “Return” key** | | **Keyboard hides** | **P** |
| **4** | **Keyboard should hide when user scrolls the view controller** | | **Keyboard hides** | **P** |
| **5** | **Data should be stored until user deletes them.** | | **Data is stored permanently even if the user closes the app and reopens it** | **P** |
| **6** | **Clear button should clear all the text fields** | | **Upon that button pressed, all text field will go back to its initial state** | **P** |
| **7** | **Save button should create an event** | | **Save button will store the data in an array and save it** | **P** |
| **8** | **View event button should navigate to different controller** | | **It should navigate to the tableViewController and display events created** | **P** |
| **9** | **Date and Time should be scrolled** | | **User should be able to scroll to choose date & time** | **P** |
| **10** | **3d touch user shall access application from home screen** | | **User taken to create new event screen inside application by passing the table screen** | **F** |
| **11** | **Swipe left and select delete events** | | **Event is deleted** |  |
| **12** | **Share events** | | **Event is shared through the medium selected.** |  |
| **13** | **Constraints on Dates & Times preventing selection of past date & time** | | **Date & time is not allowed to be selected if it has already passed** |  |