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 | 12/6/15 | Initial Creation | 1.00 |
|  |  |  |  |

# 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

housePartyViewController.swift

* 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

## Features Not to be Tested

Other classes such as:

* babyShowerViewController.swift
* tailgateViewController.swift
* graduationViewController.swift
* bbqViewController.swift

These functionalities are to be implemented in next semester

# 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 conducted in which software elements, hardware elements, or both are combined and tested until the entire system has been integrated. The purpose of integration testing is to ensure that design objectives are met and ensures that the software, as a complete entity, complies with operational requirements. Integration testing is also called System Testing.>

**TODO: Next Semester**

## 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 to 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 done to ensure that that applied changes to the application have not adversely affected previously tested functionality.>

**TODO: Next Semester**

## Acceptance Testing

<Testing conducted to determine whether or not a system satisfies the acceptance criteria and to enable the customer to determine whether or not to accept the system. Acceptance testing ensures that customer requirements' objectives are met and that all components are correctly included in a customer package.>

**TODO: Next Semester**

## 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

<Identify the set of tasks necessary to prepare for and perform testing activities. Identify all intertask dependencies and any specific skills required.>

**TODO: Next Semester**

## 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

## Security

<Identify the testing environment security and asset protection requirements.>

**TODO: Next Semester**

## Tools

<Identify the special software tools, techniques, and methodologies employed in the testing efforts. The purpose and use of each tool shall be described. Plans for the acquisition, training, support, and qualification for each tool or technique.>

**TODO: Next Semester**

## Publications

<Identify the documents and publications that are required to support testing activities.>

**TODO: Next Semester**

## Risks and Assumptions

<Identify significant constraints on testing such as test item availability, test resource availability, and time constraints. Identify the risks and assumptions associated with testing tasks including schedule, resources, approach and documentation. Specify a contingency plan for each risk factor.>

**TODO: Next Semester**

# Change Management Procedures

<Identify the software test plan change management process. Define the change initiation, change review, and change authorization process.>

**TODO: Next Semester**

# 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** |