Visits Application

Group 2: Jeff Curtis, David Polen, Derek Sappington, Roderick Barker

Final

**Table of Contents**

[**Overview**](#_Overview)**5**

[**Project Plan**](#_Project_Plan)**6**

**Introduction6**

Purpose of Plan6

Project Purpose6

Project Approach6

**Goals & Objectives7**

Project Goals and Objectives7

**Scope7**

Scope Definition7

Risk Assessment7

Milestones8

**Assumptions9**

Project Assumptions9

**Constraints9**

Project Constraints 9

Critical Project Barriers9

**Project Management Approach10**

Project Timeline10

Project Roles and Responsibilities10

Communications Plan11

**Approvals12**

Sign-Off Sheet12

[**Project Design**](#_Project_Design)**13**

**Introduction13**

Purpose of the Product Design Specification Document13

**General Overview and Design Guidelines/Approach……………………………………………………………………..13**

Assumptions / Constraints / Standards13

**Architecture Design13**

Logical View13

Hardware Architecture14

Software Architecture14

Communication Architecture15

**System Design15**

Use-Cases15

Datastore Design16

Data Conversions16

Application Program Interfaces16

User Interface Design17

**Product Design Specification Approval23**

[**Phase 1 Code**](#_Phase_I_Code)**24**

[**Phase 2 Code**](#_Phase_II_Code)**25**

[**Phase 3 Code**](#_Phase_III_Code)**26**

[**Project Testing**](#_Test_Plan)**27**

**Introduction27**

Purpose27

Project Overview27

Audience27

**Test Strategy27**

Test Objectives27

Test Assumptions28

Test Principals28

Data Approach29

Scope and Levels of Testing29

Exploratory29

[Functional Test](#_Functional_Test)29

Test Acceptance Criteria30

Test Deliverables30

Milestone List31

User Acceptance Test (UAT) 31

Test Deliverables31

Test Effort Estimate31

**Execution Strategy32**

Test Cycles32

Validation and Defect Management32

Test Metrics33

Defect Tracking & Reporting33

**Test Management Process34**

Test Design Process34

Test Execution Process34

Test Risks and Mitigation Factors35

Communications Plan36

Role Expectations36

Project Management37

Development Lead37

Test Lead37

Test Team37

Technical Writer38

**Test Environment38**

**Approvals38**

[**How to Run the Program**](#_How_to_Run)**39**

[**User’s Guide**](#_User’s_Guide)**40**

**Introduction40**

Scope and Purpose40

**First Time Run of Visits Application40**

Configure Location Settings40

Opening Visits Application40

Enable Locations41

Main Display43

Map Display44

Detail Display45

Web Page………………………………………………………………………………………………………………………………………46

[Master Document Revisions](#_Master_Document_Revisions)………………………………………………………………………………………………………………47

[**Conclusions**](#_Conclusions)**49**

**Member Contribution Report51**

# Overview

The proposed project is an iPhone application that uses key frameworks to automatically collect visit and location information in the background. This information is then displayed in various ways including a table view and a map view. The visit information is a collection of information where the phone has been.

Jeff Curtis was the project manager for group 2. He was able to lead the team to a successful end product that helped create the Visits application. Jeff was a member of the code writing team and test team. David Polen was the lead code writer for the group and played an integral role in designing the detailed view of the table in program. Derek Sappington was the lead code tester of the group. Derek was able to run multiple tests on the project and clean up all the issues the team faced within each phase. Roderick Barker was the technical writer for group 2. He made sure each version of the documentation was up to date and met the requirements of the week.

# Project Plan

**INTRODUCTION**

**Purpose of Plan**

The Group 2 iPhone App Project Plan will provide a definition of the project. This will include the project’s goals and objectives. Additionally, the Plan will serve as an agreement between the group, and other personnel associated with and/or affected by the project.

**The Project Plan defines the following:**

1. Project purpose
2. Project goals and objectives
3. Scope and expectations
4. Roles and responsibilities
5. Assumptions and constraints
6. Project management approach
7. Project timeline

**Project purpose**

The proposed project is an iPhone application that uses key frameworks to automatically collect visit and location information in the background. This information is then displayed in various ways including a table view and a map view. The visit information is a collection of information where the phone has been. For instance, if you drove to work and arrived at 730 am, then left for lunch at 1200, the first visit information would be the work address, including the arrival time, departure time, and coordinates. The second visit would be the arrival at the lunch location, with the same details minus the departure time.

From this basic information the application will first look at the visit for departure time, if not it will discard it. Then the application will take the coordinate information and perform a reverse geocode on it. This will perform a lookup and return as much information as possible, which may include street address, name, and areas of interest. This information will be used to create a custom place-mark object that will be stored in persistence storage. For example, a visit might tell you if you work at a historical building or some other known identity.

**Project Approach**

Project Phases

Phase I: Project Plan

Phase II: Project Design

Phase III: Users Guide & Test Plan

Phase IV: Phase 1 Source

Phase V: Phase 2 Source

Phase VI: Phase 3 Source

Phase VII: Final Report

**GOALS AND OBJECTIVES**

**Project Goals and Objectives**

The main goal of this project is to create a working iPhone application that shows the user detailed information about their daily movement. This information can help the user track their time more effectively, help them determine their past visits, and will open their eyes to the vast amount of data that is routinely collected from various applications that are given location information privileges.

**SCOPE**

**Scope Definition**

The hardware required to perform the coding is a Mac computer running the latest version of XCode and will utilize Apple’s Swift programming language and the iOS application programming frameworks. Running the application will require an iPhone and the user must allow the collection of location information.

**Risk Assessment**

**Initial Assessment**

| Risk | Risk Level  L/M/H | Likelihood of Event | Mitigation Strategy |
| --- | --- | --- | --- |
| Person Hours | H: Over 100 | Certainty | Assigned Project Manager, comprehensive project management approach and communications plan |
| Estimated Project Schedule | H: 6 weeks | Certainty | Created comprehensive project timeline with frequent baseline reviews |
| Narrow Knowledge Level of Users | M: Knowledgeable of most areas | Likely | Assigned roles based on what group members wanted to do |
| Available documentation | M: More than 75% complete/current | Likely | Balance of information to be gathered |
| Project Scope Creep | L: Scope generally defined, subject to revision | Unlikely | Scope initially defined in the project plan, reviewed weekly by group (Project Manager makes the final call) to prevent undetected scope creep |
| Consultant Project Deliverables unclear | L: Well defined | Unlikely | Included in the project plan, subject to amendment |
| Timeline Estimates Unrealistic | M: Timeline assumes no derailment | Somewhat likely | Timeline reviewed weekly by group (Project Manager makes the final call) to prevent undetected timeline departures |
| Number of Team Members Unknowledgeable of Business | L: Team well versed in operations impacted by technology | Unlikely | Full group knows how to complete each task and the duties |
|  |  |  |  |
| Schedules of individuals | H: Schedule can change based on the workload of current phase | Certainly | Comprehensive communications plan, frequent meetings, tight project management oversight |
| Code discrepancies | M: Project code could face errors when running | Somewhat likely | Code Writers, Code Testers, and Project Manager to review all code before it is finalized, the code can also be rolled back to the previous version to run |
|  |  |  |  |
| Team member connectivity | L: Members could face minor outages from the web | Somewhat likely | Members will back up all work and share access to all data for the group to work on |
|  |  |  |  |
| Team communication | H: Lack of communication from members could bring the project to a halt | Likely | Project Manager keeps communications with each member up and the group follows the meetup schedule |

**Milestones**

The following represent key project milestones, with estimated completion dates:

**Milestone Estimated Completion Date**

**Phase I: Project Plan**

Initial project plan 03/31/2019

**Phase II: Project Design**

Initial project plan, test plan, user’s guide 04/14/2019

**Phase III: Users Guide& Test Plan**

Initial test plan and user’s guide 04/07/2019

**Phase IV: Phase 1 Source**

Initial phase 1 source and comprehensive report 04/21/2019

**Phase V: Phase 2 Source**

Initial phase 2 source and comprehensive report 04/28/2019

**Phase VI: Phase 3 Source**

Initial phase 3 source and comprehensive report 05/05/2019

**Phase VII: Final Report**

Final overall product 05/12/2019

**ASSUMPTIONS**

**Project Assumptions**

The following assumptions were made in preparing the Project Plan:

1. Management will ensure that project team members are available as needed to complete project tasks and objectives.
2. Failure to identify changes to draft deliverables within the time specified in the project timeline will result in project delays.
3. Project team members will adhere to the Communications Plan.
4. All project participants will abide by the guidelines identified within this plan.
5. The Project Plan may change as new information and issues are revealed.

**CONSTRAINTS**

**Project Constraints**

1. Timeline of 6 weeks causes the amount of work to be limited.

**Critical Project Barriers**

1. Natural disasters

Should any of these events occur, the Project Plan would become invalid.

**PROJECT MANAGEMENT APPROACH**

**Project Timeline**

**Project Plan Due date for task:**

Initial project plan 03/31/2019

**Users Guide& Test Plan**

Initial test plan and user’s guide 04/07/2019

**Project Design**

Initial project plan, test plan, user’s guide 04/14/2019

**Phase 1 Source**

Initial phase 1 source and comprehensive report 04/21/2019

**Phase 2 Source**

Initial phase 2 source and comprehensive report 04/28/2019

**Phase 3 Source**

Initial phase 3 source and comprehensive report 05/05/2019

**Final Report**

Final overall product 05/12/2019

**Project Roles and Responsibilities**

| **Role** | **Responsibilities** | **Participant(s)** |
| --- | --- | --- |
| Project Manager | * Assign tasks for each portion of the project * Check progress throughout the week * Make final decisions if need be | Jeff Curtis |
| Code Writer | * Write code in languages needed for the project * Make code clear and easy to follow * Have code ready to test for each portion of the project | David Polen |
| Code Tester | * Test all completed code * Write Code * Test in progress code * Check code that has been written | Derek Sappington |
| Technical Writer | * Document plan for the project * Document test cases * Document user’s guide | Roderick Barker |

**Communications Plan**

The team will utilize a variety of communication methods to include Slack, e-mail, and text messages. The primary form of communication will be a dedicated Slack channel on which the team will meet multiple times per week dependent on the schedule. The team will meet on Saturdays at 10 AM to review and approve the deliverables for the following day. Additionally, the milestone for the following week will be reviewed and responsibilities assigned for the following week. Based on the workload for the given week the team will decide on how often and when to meet on Slack during the week. At a minimum, the team will then meet again on Tuesdays at 6:30 PM to discuss progress made on individual parts, assess any challenges presented, and how other team members can help each other.

Apart from the weekly meetings, members are encouraged to post on Slack at any time and communicate as often as needed. In the case of a last-minute meeting being required, the members can organize this via text message and let everyone know a time to get on Slack. In the event that a member misses a meeting, utilizing Slack allows them to quickly and efficiently read through the meeting and get acquainted with what was discussed.

**APPROVALS**

**Sign-off Sheet**

***I have read the above Project Plan and will abide by its terms and conditions and pledge my full commitment and support for the Project Plan.***

**Project Manager: Jeff Curtis 03/31/2019**

Date

**Code Writer: David Polen 03/31/2019**

Date

**Code Tester: Derek Sappington 03/31/2019**

Date

**Technical Writer: Roderick Barker 03/31/2019**

Date

# Project Design

**Introduction**

**Purpose of The Product Design Specification Document**

The Product Design Specification document documents and tracks the necessary information required to effectively define the architecture and system design in order to give the development team guidance on the architecture of the system to be developed.

**General Overview and Design Guidelines/Approach**

This section describes the principles and strategies to be used as guidelines when designing and implementing the system.

**Assumptions / Constraints / Standards**

***Assumptions:***

* Management will ensure that project team members are available as needed to complete project tasks and objectives.
* Failure to identify changes to draft deliverables within the time specified in the project timeline will result in project delays.
* Project team members will adhere to the Communications Plan.
* All project participants will abide by the guidelines identified within this plan.

***Constraints:***

* Timeline of 6 weeks causes the amount of work to be limited.

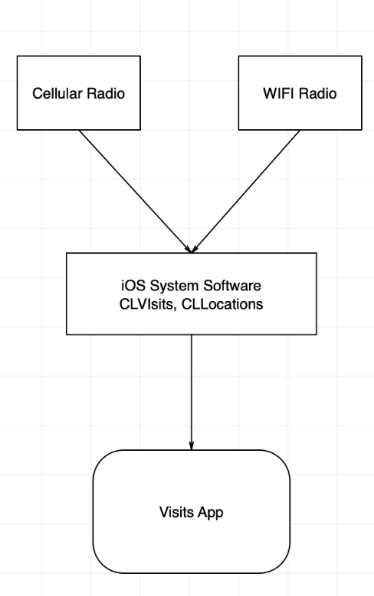
**Architecture Design**

The Visit application uses Apple’s iOS software development kit and associated frameworks. The application targets iPhones and uses the cellular and WIFI capabilities to harvest system-wide location events and bubbles these up for the user to see. The application is dependent on the user granting location services permissions to function correctly. The application is written using a model-view-controller approach

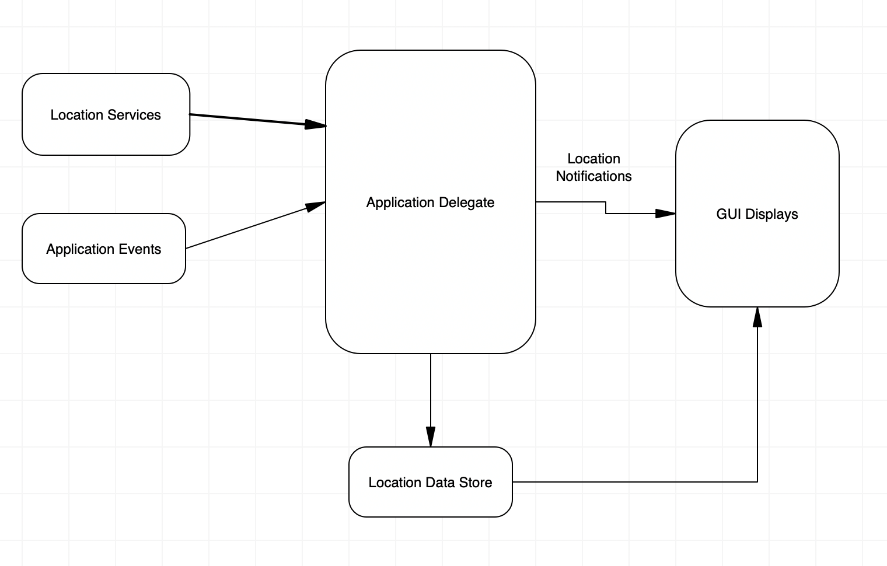
**Logical View**

****

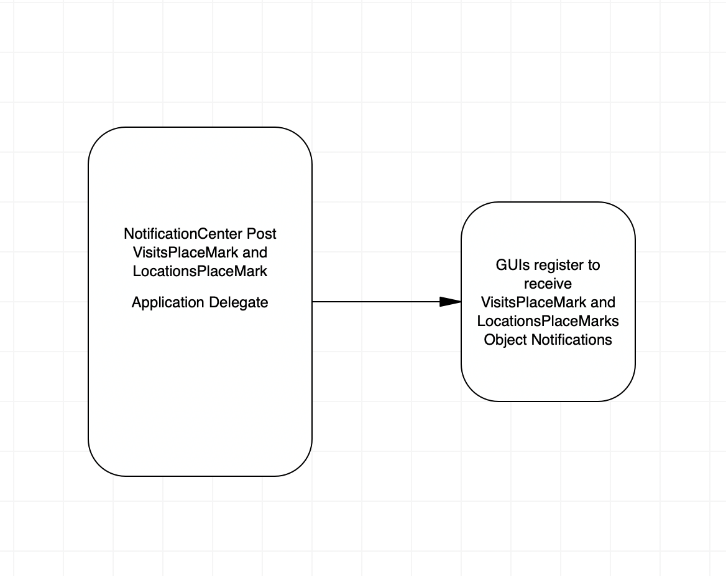
**Hardware Architecture**

**

**Software Architecture**

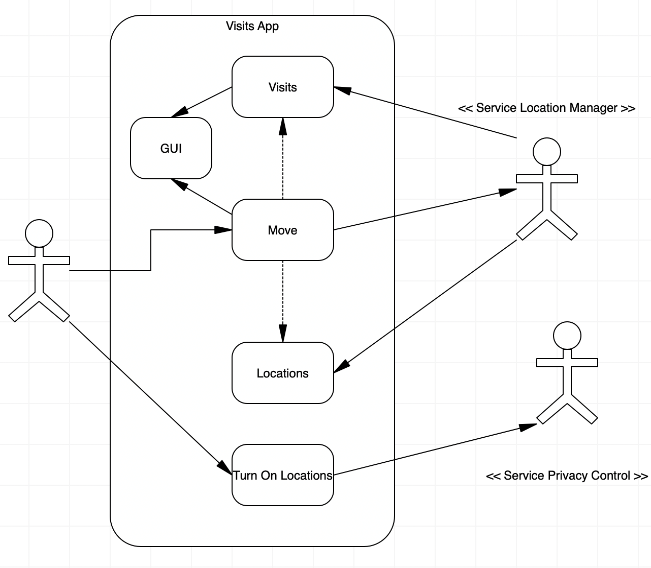
****

**Communication Architecture**

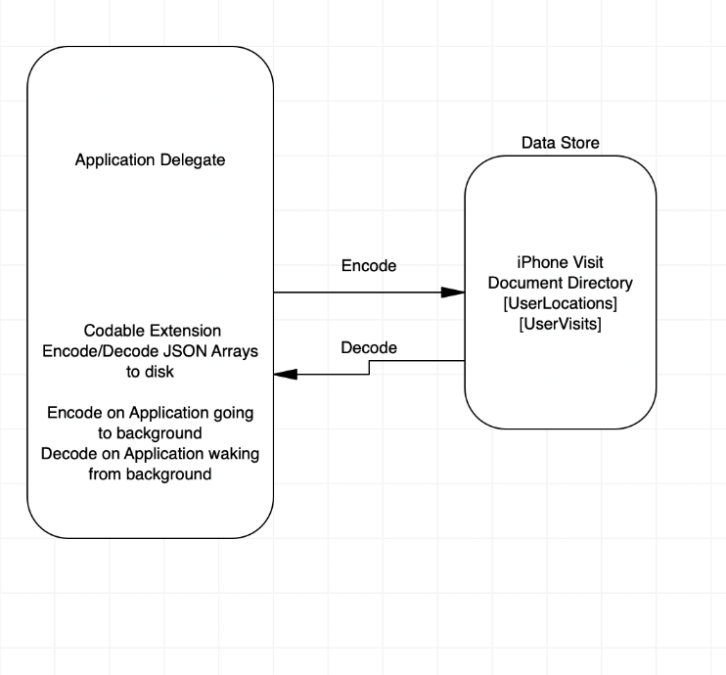


**System Design**

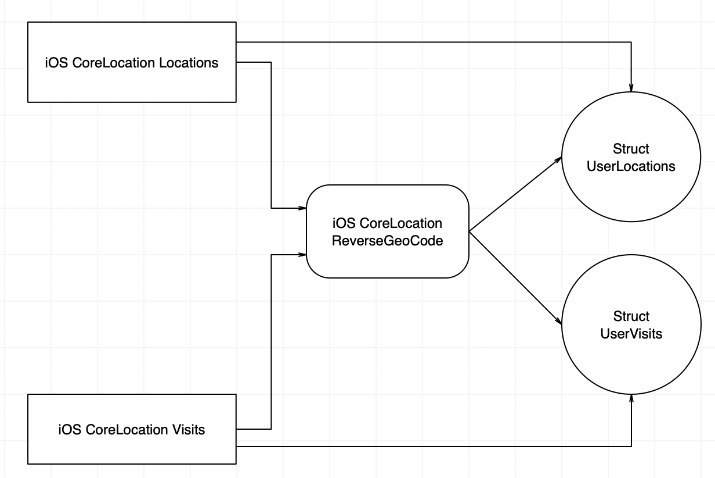
**Use-Cases**

****

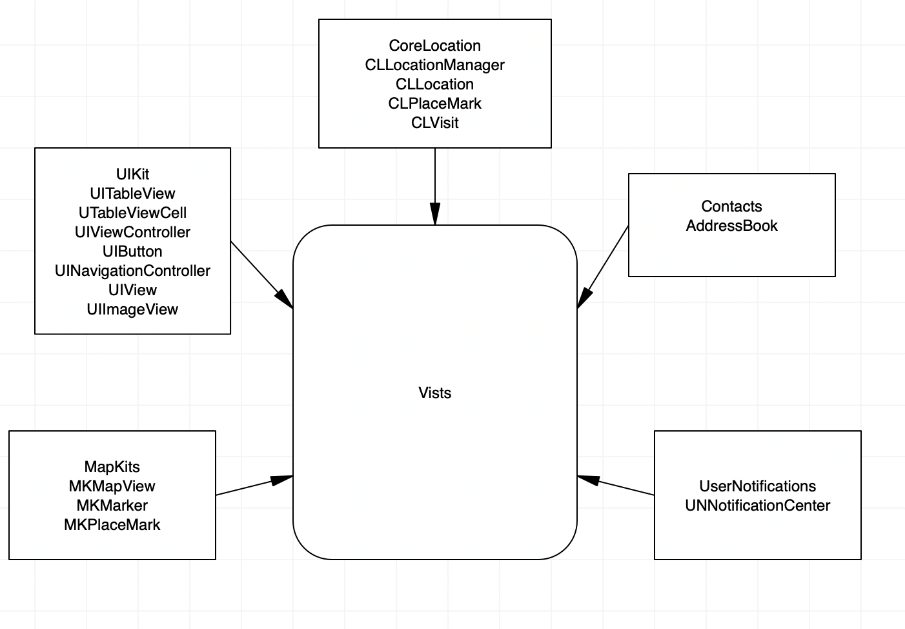
**DataStore Design**

****

**Data Conversions**

****

**Application Program Interfaces**

**

**User Interface Design**

|  |  |
| --- | --- |
| https://lh6.googleusercontent.com/a0GIGp6Rd6ZFLySM7Y6-NeVOIEzXOhi1E7M2OmnFrWfzmQLxnvEgYN01OCvi0-hEpYsjVbp9CeB2I01jvonaW2Y9ncj6Dj94n-CjFkOvrVXZWt0R4I9Kz-mADDD3ngOZnnaWw7X5 | https://lh6.googleusercontent.com/gWkrhNhID2fLe2hQLgOTvT77Mage-fWBvR5HlwT-3XRY3Mxeff1pkV5qkgW6gyR3iFC3y3LAWI23YUXHGgQsTzDkMjsYn-9t1tZjJ7MSKelP-KAENgmHrPMZIgC4T3RqPelvukhq |
| Main Screen Locations | Main Screen Visits |

|  |  |
| --- | --- |
| https://lh5.googleusercontent.com/U5nwHFIXh_sGIq_OL3fMl-_7RYvyO011TTgWePuMWn3sOJe7ONzKE2qB6rSbRVWcsaBR_lg3kAxkxN8yLJiJSU4ltCbv9YfPumBYrBT5zkOLtHl5k3vZQh6x66XPYrpdtfNWAPrN | https://lh4.googleusercontent.com/kLQng0kGN8sJyj5umt-RTewjc7owP3OI66xVQwoYMhCxqbWVlX4_N1DmhCckpd0PzKb1sl--96Bfgazy0ixyqS8W1jZ095RWvr9hkachdkVzsCvQ-_N4lwz3VWufCBV-la2I2FnM |
| Map Screen Locations | Map Screen Locations |

|  |  |
| --- | --- |
| https://lh5.googleusercontent.com/U5nwHFIXh_sGIq_OL3fMl-_7RYvyO011TTgWePuMWn3sOJe7ONzKE2qB6rSbRVWcsaBR_lg3kAxkxN8yLJiJSU4ltCbv9YfPumBYrBT5zkOLtHl5k3vZQh6x66XPYrpdtfNWAPrN | ~~\*~~https://lh4.googleusercontent.com/kLQng0kGN8sJyj5umt-RTewjc7owP3OI66xVQwoYMhCxqbWVlX4_N1DmhCckpd0PzKb1sl--96Bfgazy0ixyqS8W1jZ095RWvr9hkachdkVzsCvQ-_N4lwz3VWufCBV-la2I2FnM |
| Map Screen Locations | Map Screen Locations |
| https://lh5.googleusercontent.com/TeveRXO7BgGvPpbqxpEbmSs9G2lA6wGZqEeVmrcxgR2ZqVKNZ9-pTsZf1vKlp2ktMD5a49tmrLL1VQK4hjtSM5VpDYo1V1SCQvhJ5Io8t9n9Mz2_HjYMULmGUto9yC9mpdKcNZXJ | https://lh3.googleusercontent.com/G_HHhJjZ8hpumC1VexoVv9QT0EsbIuNXvTaj-T2kWGvsrJBkPj5OtE77-IuAbBdz36LoWk_lykYDwHZNJoi66PUy_wuUwtQ5wQdT2niNfX9_2roExhZhTSTQaDYOHPob049wPOK0aLtEBDJZww |
| Map Screen Visits | Tapping Address from Main Screen |

|  |  |
| --- | --- |
| https://lh4.googleusercontent.com/lZL9wB4X9KHi26y58MqgiJl37tn6YuLydkac4EEi0Z0uQdMel16WYv-B3N7n4dpjd83OMlDM5jg_UaZVJdMNqlVHKDxzLKI-BrlyXkCu5hpLUe6Hw6ttZ6M17glAfC6zjWZ_RusM | https://lh3.googleusercontent.com/fwARzbuWMmnqUSK5dWPcW99GMMKRyjgKe7fJkim7XlPYbhBARH5yioUbcAsVrS6LzRQdVHY69WLV8wLTMV_tyJNlJOF67t8FK4d6uXSZQQ0A-hCj69P-UWz9sOeL73pp_Jbd31xR |
| Detail Screen Location | Wiki View From Location |

|  |  |
| --- | --- |
| https://lh4.googleusercontent.com/ShE9bVrWF7TaLG39p3D9KZ8dbuJOj6Uy8lII71NZOK49lRa-bKuqBTlK0djl9coCdfCl2wVCKexZhaq1kgKlLDuauHCEV1nTG0ENHi8Izr_b-uJ0x6G019ZBacJORRt2UHfrzygU | https://lh5.googleusercontent.com/mfR4a4boB24XAj7hjo-nfqP9Vw1KFNL4Y_TeRO8ZEddf8wuOsrn_8vRLCxs4H-kMGHvy6EW1RPqY8hqzVy0LkgLPJOrOYgat6rbordPzdW52ASt2sLALaBHbxArX5IszFoo7bAea |
| Detail Screen With Speed Info | Detail Screen From Visit |

**Product Design Specification Approval**

The undersigned acknowledge they have reviewed the *Visits* **Product Design Specification** document and agree with the approach it presents. Any changes to this Requirements Definition will be coordinated with and approved by the undersigned or their designated representatives.

|  |  |  |  |
| --- | --- | --- | --- |
| Signature: | Jeff Curtis | Date: | 04/07/2019 |
| Title: | Project Manager |  |  |

|  |  |  |  |
| --- | --- | --- | --- |
| Signature: | David Polen | Date: | 04/07/2019 |
| Title: | Code Writer |  |  |

|  |  |  |  |
| --- | --- | --- | --- |
| Signature: | Derek Sappington | Date: | 04/07/2019 |
| Title: | Code Tester |  |  |

|  |  |  |  |
| --- | --- | --- | --- |
| Signature: | Roderick Barker | Date: | 04/07/2019 |
| Title: | Technical Writer |  |  |

# Phase I Code

PHASE 1 CODE

See zip folder Visits-dev-.0.7 for phase I code or at <https://github.com/jeffreyscurtis/Visits/tree/dev-0.7>

Tailor Report:

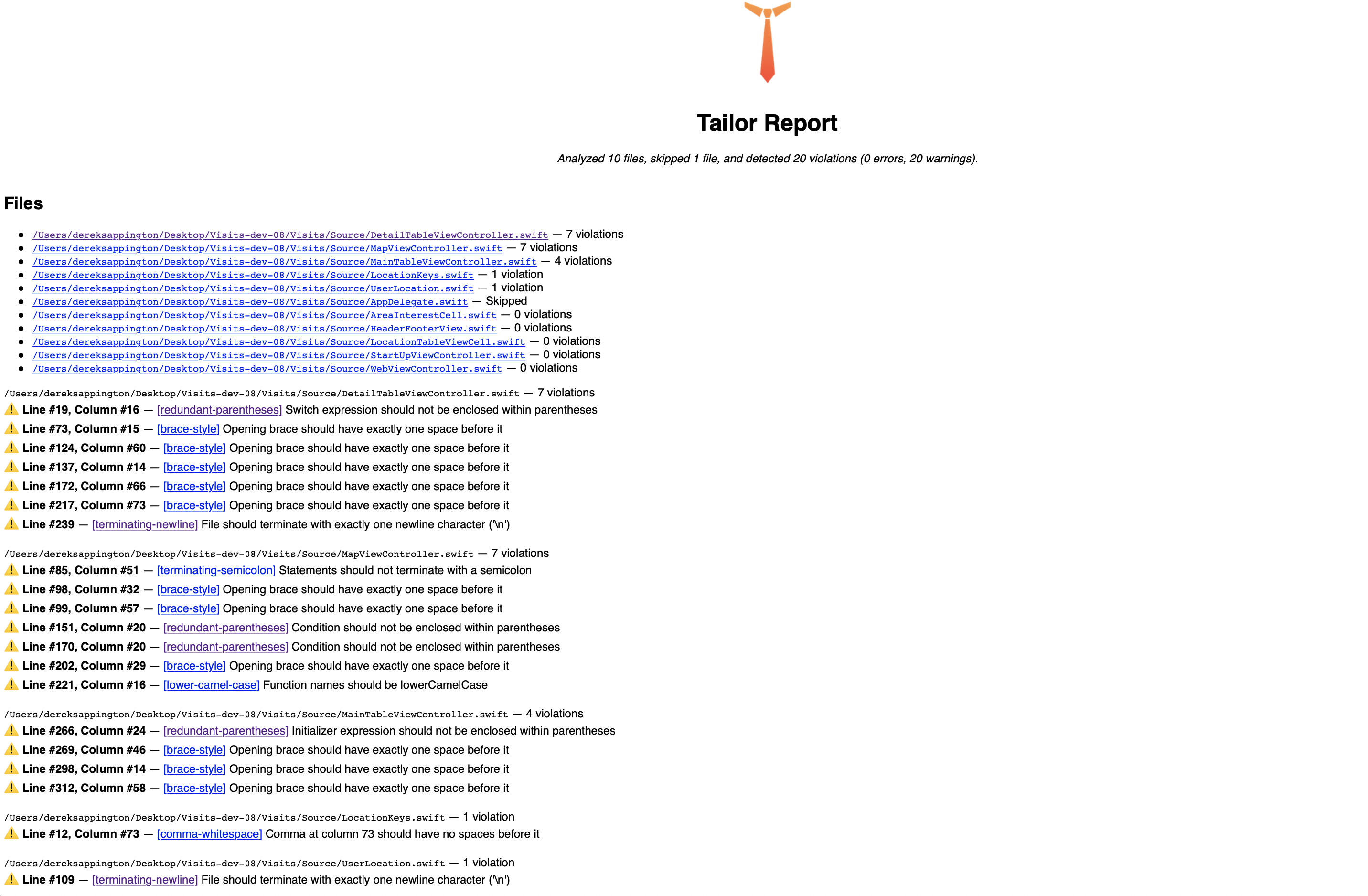


# Phase II Code

PHASE 2 CODE

See zip folder Visits-dev-0.8 for phase II code or at <https://github.com/jeffreyscurtis/Visits/tree/dev-08>

Tailor report:



# Phase III Code

PHASE 3 CODE

See zip folder Visits-dev-0.9 for phase III code or at <https://github.com/jeffreyscurtis/Visits/tree/release-1.0>

Phase III Tailor Report



# Test Plan

# INTRODUCTION

## Purpose

This test plan describes the testing approach and overall framework that will drive the testing of the iPhone App produced by group 2. The document introduces:

* Test Strategy: rules the test will be based on, including the givens of the project (e.g.: start / end dates, objectives, assumptions); description of the process to set up a valid test (e.g.: entry / exit criteria, creation of test cases, specific tasks to perform, scheduling, data strategy).
* Execution Strategy: describes how the test will be performed and process to identify and report defects, and to fix and implement fixes.
* Test Management: process to handle the logistics of the test and all the events that come up during execution (e.g.: communications, escalation procedures, risk, and mitigation, team roster)

## Project Overview

The Visits App is a powerful tool providing consumers with the ability to view relevant information such as their past visits to locations, current visits, dates and times of all visits.

## Audience

* Project team members perform tasks specified in this document and provide input and recommendations on this document.
* Project Manager Plans for the testing activities in the overall project schedule reviews the document, tracks the performance of the test according to the task herein specified, approves the document and is accountable for the results.
* Technical Team ensures that the test plan and deliverables are in line with the design, provides the environment for testing and follows the procedures related to the fixes of defects.

# TEST STRATEGY

## Test Objectives

The objective of the test is to verify the functionality of the iPhone app and to make sure it works according to the specifications.

The test will execute and verify the test scripts, identify, fix and retest all high and medium severity defects per the entrance criteria, prioritize lower severity defects for future fixing.

The final product of the test should guarantee a:

* A production-ready iPhone app

## Test Assumptions

**Key Assumptions**

* Production like data required and be available in the system prior to the start of Functional Testing

**General**

* Exploratory Testing would be carried out once the build is ready for testing
* Test case design activities will be performed by the code testing group
* Test environment and preparation activities will be owned by lead code tester
* Dev team will provide Defect fix plans based on the Defect meetings during each cycle to plan. The same will be informed to Test team prior to the start of Defect fix cycles
* Project Manager will review and sign-off all test deliverables
* The project will provide test planning, test design, and test execution support
* The project team has the knowledge and experience necessary or has received adequate training in the system, the project and the testing processes.
* There is no environment downtime during the test due to outages or defect fixes.
* The system will be treated as a black box; if the information shows correctly online and in the reports, it will be assumed that the database is working properly.

**Functional Testing**

* During Functional testing, the testing team will use preloaded data which is available on the system at the time of execution
* The Test Team will be performing Functional testing only on the iPhone app code

## Test Principles

* Testing will be focused on meeting the objectives and quality.
* There will be common, consistent procedures for all teams supporting testing activities.
* Testing processes will be well defined, yet flexible, with the ability to change as needed.
* Testing activities will build upon previous stages to avoid redundancy or duplication of effort.
* Testing environment and data will emulate a production environment as much as possible.
* Testing will be a repeatable, quantifiable, and measurable activity.
* Testing will be divided into distinct phases, each with clearly defined objectives and goals.

## Data Approach

* In functional testing, the iPhone app will contain pre-loaded test data and which is used for testing activities.

## Scope and Levels of Testing

### Exploratory

**PURPOSE**: the purpose of this test is to make sure critical defects are removed before the next levels of testing can start.

**SCOPE**: First level navigation

**TESTERS**: Jeff Curtis, David Polen, and Derek Sappington

**METHOD**: this exploratory testing is carried out in the application without any test scripts and documentation

### Functional Test

**PURPOSE:**  Functional testing will be performed to check the functions of the application. The functional testing is carried out by feeding the input and validates the output from the application.

**Scope:** The table below shows details about the scope of Functional test

|  |  |  |
| --- | --- | --- |
| **Case Number** | **Description** | **Pass/Fail** |
| **1** | Static Analyses of Code  Final [Report](https://slack-files.com/THFBY1JLE-FJ8CU56D7-95a5471943) – Project compilation report | Passed\* |
| **2** | Check initial application launch opens and loads as expected. | Passed |
| **3** | Check application navigation between different views works as expected. | Passed |
| **4** | Ensure all views are showing data that is of relevance to the view. | Passed |
| **5** | Check to ensure locations are updating as expected. | Passed |
| **6** | Ensure visits are being logged correctly and not being treated as locations, and that locations are not being treated as visits. | Passed |
| **7** | Ensure the difference is nominal in program functionality and performance over extended use of time. | Passed |
| **8** | Check that when user does not grant permission for locations to be “Always Allow” on device, the appropriate error screen is displayed. | Passed |

**TESTERS**: Jeff Curtis, David Polen, and Derek Sappington

**METHOD**: The test will be performed according to Functional scripts

#### TEST ACCEPTANCE CRITERIA

1. Approved Functional Specification document, Use case documents must be available prior to the start of the Test design phase.
2. Test cases approved and signed-off prior to the start of Test execution
3. Development completed, unit tested with pass status and results shared to Testing team to avoid duplicate defects
4. Test environment with the application installed, configured and ready to use state

#### TEST DELIVERABLES

|  |  |  |  |
| --- | --- | --- | --- |
| **S.No.** | **Deliverable Name** | **Author** | **Reviewer** |
| 1. | Test Plan | Roderick Barker | Jeff Curtis |
| 2. | Functional Test Cases | David Polen | Jeff Curtis |
| 3. | Logging Defects | Jeff Curtis, Derek Sappington | Jeff Curtis |
| 4. | Daily/weekly status report | David Polen, Derek Sappington | Jeff Curtis |
| 5. | Test Closure report | Roderick Barker | Jeff Curtis |

#### MILESTONE LIST

Due Date:

Phase 1 Code 04/18/2019

Phase 1 Code Test 04/19/2019

Phase 1 04/21/2019

Phase 2 Code 04/25/2019

Phase 2 Test 04/26/2019

Phase 2 04/28/2019

Phase 3 Code 05/02/2019

Phase 3 Test 05/03/2019

Phase 3 05/05/2019

### User Acceptance Test (UAT)

**PURPOSE**: this test focuses on validating business logic. It allows the end users to complete one final review of the system prior to deployment.

**TESTERS**: Project Manager

**METHOD**: Test team write the UAT test cases based on the inputs from the project manager

**TIMING**: After all other levels of testing (Exploratory and Functional) are done. Only after this test is completed the product can be released to production.

#### TEST DELIVERABLES

|  |  |  |  |
| --- | --- | --- | --- |
| **S.No.** | **Deliverable Name** | **Author** | **Reviewer** |
| 1. | UAT Test Cases | Code Writer / Code Tester | Jeff Curtis |

## Test Effort Estimate

|  |  |  |
| --- | --- | --- |
| Job | Team Member(s) | Estimated Completion Time |
| Phase 1 Source Code | David Polen, Jeff Curtis | 2 Days |
| Phase 1 Testing | Derek Sappington, Jeff Curtis | 1 Day |
| Phase 2 Source Code | David Polen, Jeff Curtis | 2 Days |
| Phase 2 Testing | Derek Sappington, Jeff Curtis | 1 Day |
| Phase 3 Source Code | David Polen, Jeff Curtis | 2 Days |
| Phase 3 Testing | Derek Sappington, Jeff Curtis | 1 Day |
| Final Phase Code | David Polen, Jeff Curtis | 2 Days |
| Final Phase Testing | Derek Sappington, Jeff Curtis | 1 Day |

# EXECUTION STRATEGY

## Test Cycles

* + There will be two cycles for functional testing. Each cycle will execute all the scripts.
  + The objective of the first cycle is to identify any blocking, critical defects, and most of the high defects. It is expected to use some workaround in order to get to all the scripts.
  + The objective of the second cycle is to identify remaining high and medium defects, remove the workaround from the first cycle, correct gaps in the scripts and obtain performance results.

## Validation and Defect Management

* It is expected that the testers execute all the scripts in each of the cycles described above. However, it is recognized that the testers could also do additional testing if they identify a possible gap in the scripts.
* It is the responsibility of the tester to open the defects, link them to the corresponding script, assign an initial severity and status, retest and close the defect; it is the responsibility of the Test lead to review the severity of the defects and facilitate with the technical team the fix and its implementation, communicate with testers when the test can continue or should be halted, request the tester to retest, and modify status as the defect progresses through the cycle.

Defects found during the Testing will be categorized according to severity.

|  |  |
| --- | --- |
| **Severity** | **Impact** |
| 1 (Critical) | * This bug is critical enough to crash the system, cause file corruption, or cause potential data loss * It causes an abnormal return to the operating system (crash or a system failure message appears). * It causes the application to hang and requires rebooting the system. |
| 2 (High) | * It causes a lack of vital program functionality with the workaround. |
| 3 (Medium) | * This Bug will degrade the quality of the System. However, there is an intelligent workaround for achieving the desired functionality - for example through another screen. * This bug prevents other areas of the product from being tested. However other areas can be independently tested. |
| 4 (Low) | * There is an insufficient or unclear error message, which has minimum impact on product use. |
| 5(Cosmetic) | * There is an insufficient or unclear error message that has no impact on product use. |

## Test Metrics

Test metrics to measure the progress and level of success of the test will be developed and shared with the project manager for approval. Below are some of the metrics

|  |  |  |
| --- | --- | --- |
| **Report** | **Description** | **Frequency** |
| Test preparation & Execution Status | To report on % complete, %WIP, % Pass, % Fail  Defects severity wise Status – Open, closed, any other Status | Weekly / Daily (optional) |
| Daily execution  status | To report on Pass, Fail, Total defects, highlight Showstopper/ Critical defects | Daily |
| Project Weekly Status report | Project driven reporting (As requested by PM) | Weekly – If project team needs weekly update apart from daily and there is a template available with the project team to use. |

## Defect Tracking & Reporting

Following flowchart depicts Defect Tracking Process:

**Approved?**

**Start**

**Tester:**

**Report defects**

**Dev Lead:**

**Assign defects**

**Developer:**

**Fixes defects**

**Tester:**

**Retests the product**

**No**

**Stop**

**Close defect**

**Yes**

**Test Lead:**

**Validate defects**

# TEST MANAGEMENT PROCESS

## Test Design Process

* The tester will understand each requirement and prepare a corresponding test case to ensure all requirements are covered.
* Each of the Test cases will undergo review by the project manager and the review defects are captured and shared to the Test team. The testers will rework on the review defects and finally obtain approval and sign-off.
* During the preparation phase, the tester will use the prototype, use case, and functional specification to write step by step test cases.
* Testers will maintain a clarification Tracker sheet and same will be shared periodically with the Requirements team and accordingly, the test case will be updated. The clarifications may sometimes lead to Change Requests or not in scope or detailing implicit requirements.
* Sign-off for the test cases would be done by project management.
* Any subsequent changes to the test case if any will be directly updated in the test plan.

## Test Execution Process

* Once all Test cases are approved and the test environment is ready for testing, the tester will start an exploratory test of the application to ensure the application is stable for testing.
* Each Tester is assigned Test cases directly in this document by project management.
* Testers to ensure necessary access to the testing environment If any issues, will be escalated to the Test Lead and in turn to the Project Manager as escalation.
* If any showstopper during exploratory testing will be escalated to the respective development teams for fixes.
* Each tester performs step by step execution and updates the status of the executions. The tester enters Pass or Fail Status for each of the steps directly.
* The testing team will participate in defect triage meetings in order to ensure all test cases are executed with either pass/fail category.
* This process is repeated until all test cases are executed fully with Pass/Fail status.
* During the subsequent cycle, any defects fixed applied will be tested and results will be updated in the next phase.

As per Process, final sign-off or project completion process will be followed

## Test Risks and Mitigation Factors

| Risk | Prob. | Impact | Mitigation Plan |
| --- | --- | --- | --- |
| **SCHEDULE**  The testing schedule is tight. If the start of the testing is delayed due to design tasks, the test cannot be extended beyond the UAT scheduled start date. | High | High | * The testing team can control the preparation tasks (in advance) and early communication with involved parties. * Some buffer has been added to the schedule for contingencies, although not as much as best practices advise. |
| **DEFECTS**  Defects are found at a late stage of the cycle or at a late cycle; defects discovered late are most likely be due to unclear specifications and are time-consuming to resolve. | Medium | High | Defect management plan is in place to ensure prompt communication and fixing of issues. |
| **SCOPE**  Scope completely defined | Medium | Medium | Scope is well defined but the changes are in the functionality is not yet finalized or keep on changing. |
| Natural disasters | Low | Medium | Teams and responsibilities have been spread to two different geographic areas. In a catastrophic event in one of the areas, there will resource in the other areas needed to continue (although at a slower pace) the testing activities. |
| Non-availability of Independent Test environment and accessibility | Medium | High | Due to non-availability of the environment, the schedule gets impacted and will lead to a delayed start of Test execution. |
| Delayed Testing Due To new Issues | Medium | High | During testing, there is a good chance that some “new” defects may be identified and may become an issue that will take time to resolve.  There are defects that can be raised during testing because of unclear document specification. These defects can yield to an issue that will need time to be resolved.  If these issues become showstoppers, it will greatly impact on the overall project schedule.  If new defects are discovered, the defect management and issue management procedures are in place to immediately provide a resolution. |

## Communications Plan

The primary form of communication will be a dedicated Slack channel on which the team will meet multiple times per week dependent on the schedule. The team will meet on Saturdays at 10 AM to review and approve the deliverables for the following day. Additionally, the milestone for the following week will be reviewed and responsibilities assigned for the following week. Based on the workload for the given week the team will decide on how often and when to meet on Slack during the week. At a minimum, the team will then meet again on Tuesdays at 6:30 PM to discuss progress made on individual parts, assess any challenges presented, and how other team members can help each other.

## Role Expectations

The following list defines in general terms the expectations related to the roles directly involved in the management, planning or execution of the test for the project.

| SN0. | Roles | Name |
| --- | --- | --- |
| 1. | Project Manager | Jeff Curtis |
| 2. | Development Lead | David Polen |
| 3. | Testing Lead | Derek Sappington |
| 4. | Technical Writer | Roderick Barker |

### Project Management

* Project Manager: reviews the content of the Test Plan, Test Strategy and Test Estimates signs off on it.

### Development Lead

* Review testing deliverables (test plan, cases, scripts, expected results, etc.) and provide timely feedback.
* Assist in the validation of results (if requested).
* Support the development and testing processes being used to support the project.
* Certify correct components have been delivered to the test environment at the points specified in the testing schedule.
* Keep project team and leadership informed of potential software delivery date slips based on the current schedule.
* Define processes/tools to facilitate the initial and ongoing migration of components.
* Conduct first line investigation into execution discrepancies and assist test executors in the creation of accurate defects.
* Implement fixes to defects according to schedule.

### Test Lead

* Acknowledge the completion of a section within a cycle.
* Give the OK to start the next level of testing.
* Facilitate defect communications between the testing team and technical / development team.

### Test Team

* Develop test conditions, test cases, expected results, and execution scripts.
* Perform execution and validation.
* Identify, document and prioritize defects according to the guidance provided by the Test lead.
* Re-test after software modifications have been made according to the schedule.
* Prepare testing metrics and provide regular status.

### Technical Writer

* Put a coherent document together of the test plan
* Summarize overall testing if necessary

# TEST ENVIRONMENT

Xcode is an integrated development environment, it will help the team with making the Visits iOS app. This test environment is essentially an app for making apps. Xcode is also a code editor that supports multiple languages. It’s also a user interface development tool for creating apps and games. This testing environment was picked due to its being the only officially-supported tool created by Apple for the creation of publishing apps to Apple's app stores.

# APPROVALS

**Approvers List**

|  |  |  |
| --- | --- | --- |
| Name | Role | Approval / Review Date |
| **Jeff Curtis** | **Project Manager** | 04/03/2019 |
| **David Polen** | **Code Writer** | 04/03/2019 |
| **Derek Sappington** | **Code Tester** | 04/03/2019 |
| **Roderick Barker** | **Technical Writer** | 04/03/2019 |

# 

# How to Run the Program

Instructions on running the source code:

1. Unzip source code and remember the saved location.

2. Open the latest version of Xcode

3. Use the file menu and select File -> Open

4. Navigate to the location where Visits-dev-08 was saved and select that folder

5. Select the file Visits.xcodeproj

6. If a scheme does not exist, create one for Visits and then select iPhone Xr for the simulator

7. Press the play button

At this point the program will launch on the simulator – in the simulator make sure you enable locations:

1. In the file, menu select Debug->Locations->Freeway Drive

The simulator will not simulate visits but will simulate locations. From this point on you should follow the user guide on how to operate the application. The main first step is to tap enable locations on the bottom button and in the pop up select “Always Allows”

To test the code, select Locations on the main screen

# User’s Guide

**Introduction**

**Scope and Purpose**

The Visits App is a powerful tool providing consumers with the ability to view relevant information such as their past visits to locations they have been to, current visits to these locations, and all the dates and times of the visits.

This user guide provided will help new user’s get a feel for how the visits app works and what the capabilities are. Each step of the initial set up will be covered and will explain to the user the overall purpose.

**First time run of Visits application**

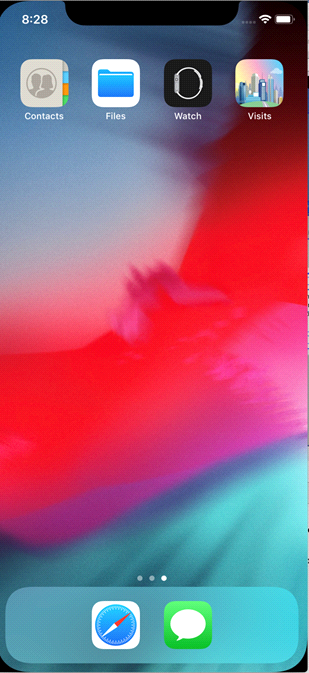
The following guide will show you how to run the application and all settings that should be configured for proper use and collection of location visits. Each step will tell you what to tap, enter, and what each function in each window does. Brief descriptions will be present to help explain in more detail.

**Configure Location Settings**

After the installation of the Visits application is complete, the first steps will be to configure the settings for the application to run on the device. The location settings will need to be configured to operate properly.

**Opening Visits Application:**

1. Tap Visits icon



**NOTE: This application is currently only available on Apple devices the run iOS…**

**Enable Locations:**

1. Upon opening of application:

Read the entry message on the bottom of the screen

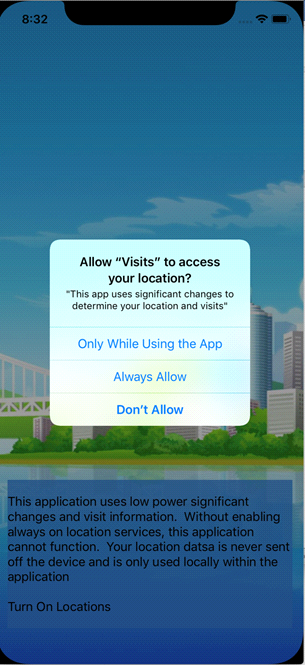
Tap “Turn On Locations”



Once the user has tapped “Turn on Locations”:

1. Read the pop up “Allow “Visits” to access your location?”

Tap “Always Allow”



**NOTE: Location services must be enabled, without enabling these services this application cannot function properly.**

**Main Display**

Once all location services and settings have been configured and enabled, the user will be brought to the main display. The main display will show a large map of the current country your location settings are configured to. From here you can change the map type, change between locations and places that have been visited, go to the map display page, and get a detailed informational display of locations.



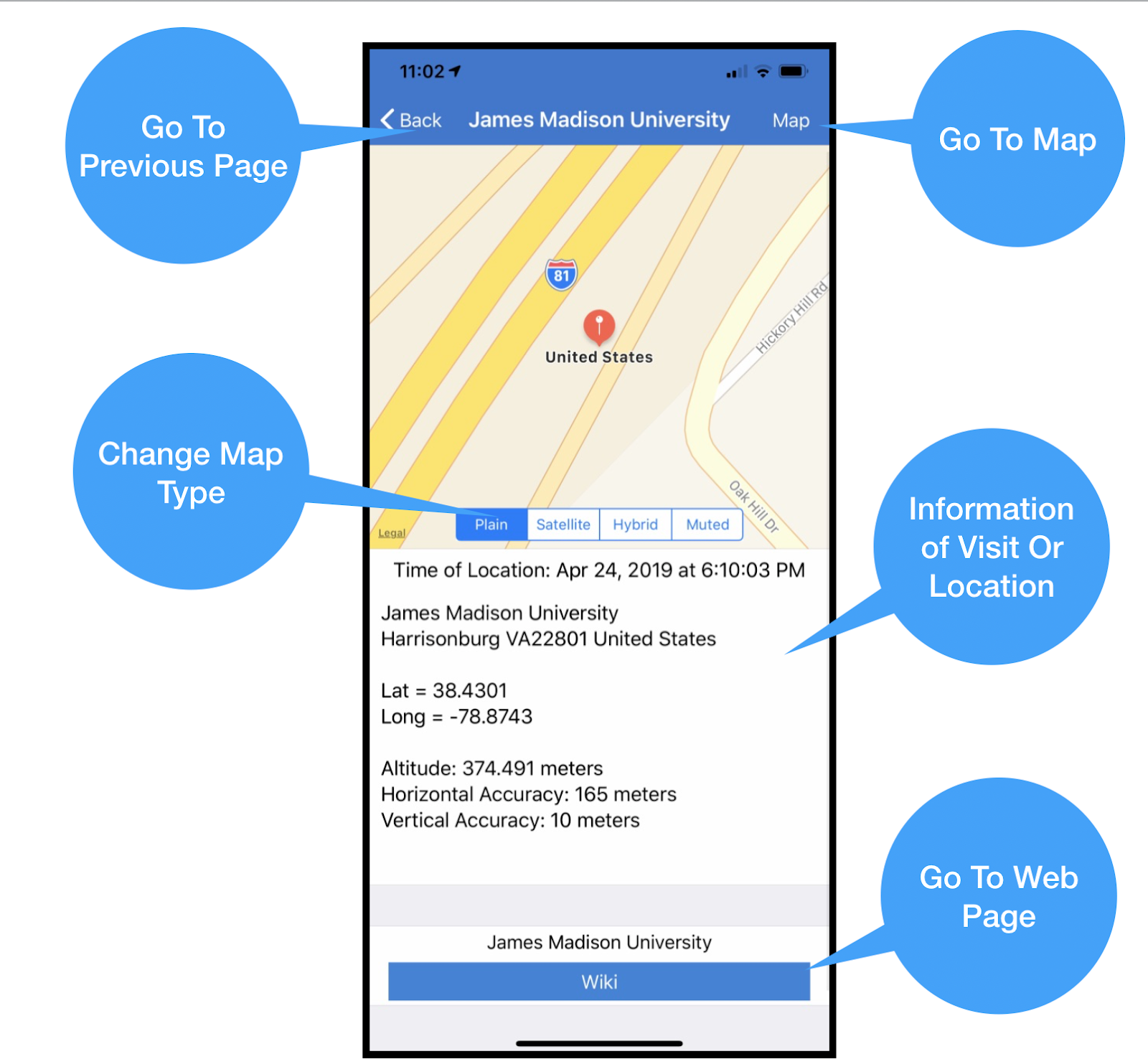
**Map Display**

To get a view of just the map and all visible locations and visited locations, tapping the “Map” tag at the top right-hand corner of the main display brings you to the map page. This page can be customized to show different map types and the locations and visited locations. To leave this page and get back to the main display, simply tap the back tag on the upper left-hand corner of the map page.



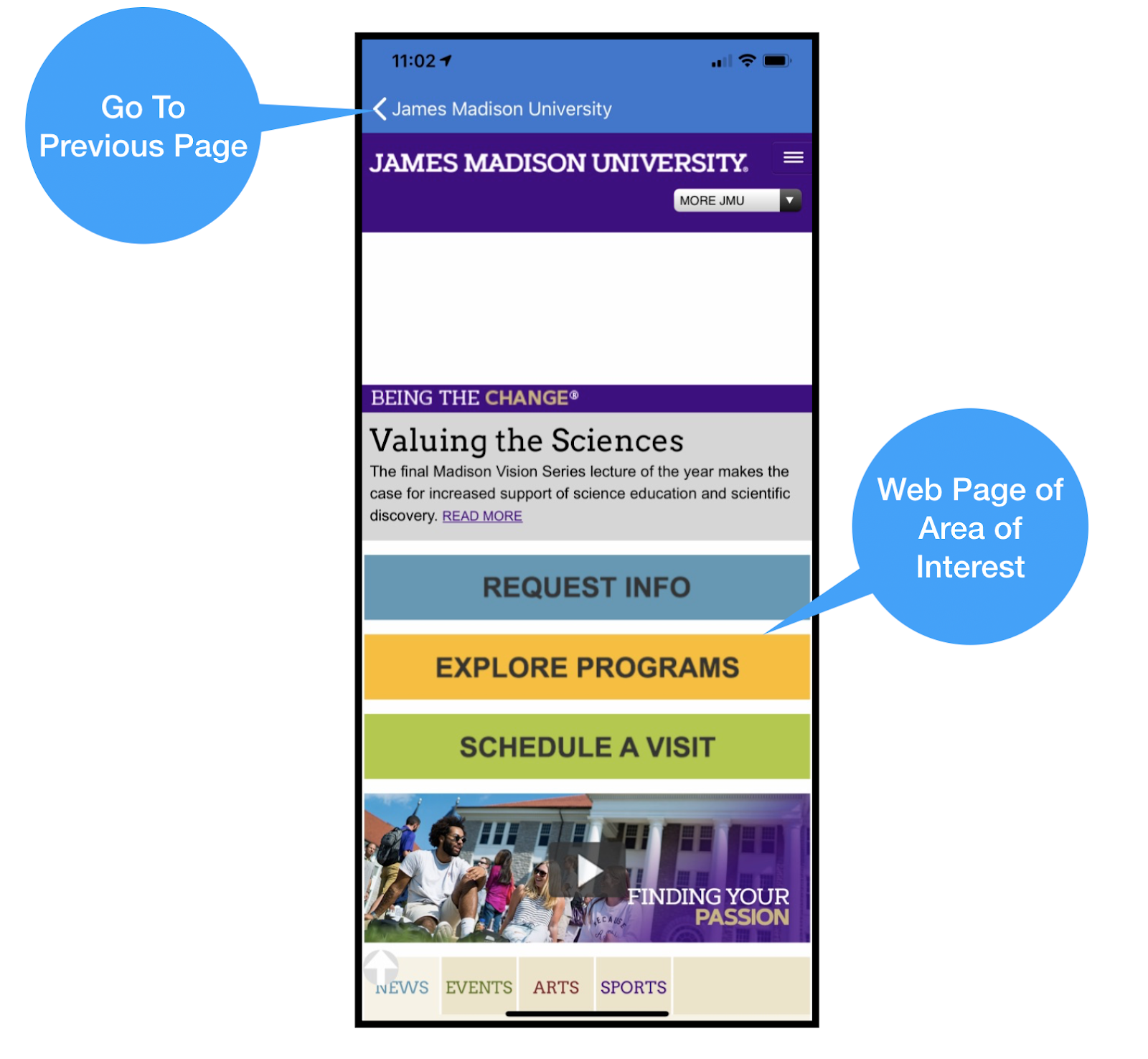
**Detail Display**

The Visits application also gives a detailed display of all locations and visits. This will pick up all key points and mark them on the map above the display. This information will provide you with information ranging from the last time you visited, a brief description of the landmark, and much more. Like the previous screen, the style of the map can be changed. To see just the map or the previous screen you simply tap the “Back” or “Map” tags. Additionally, if an area of interest was available near the location one or more wiki buttons will take you to the web page of this detection



**Web Page**

If any area of interest were found in your locations or visits, then a webpage result is available.  This page is simply a small browser with limited functionality.

****

## Master Document Revisions

|  |  |  |  |
| --- | --- | --- | --- |
| Visits Application Master Revision Table | | | |
| Project Design Document | | | |
| Version No. | Date | Author | Revision Description |
| 1.0 | 04/03/19 | Roderick Barker | Initial draft |
| 1.1 | 04/04/19 | Roderick Barker, Jeff Curtis | Revision 1 of draft |
| 1.2 | 04/13/19 | Jeff Curtis | Datastore revision |
| 1.3 | 04/20/19 | David Polen | Test cases updated and new table of contents added |
| 1.4 | 04/27/19 | Jeff Curtis | Updated screenshots and updated User’s Guide |
| 1.5 | 04/28/19 | Roderick Barker, Jeff Curtis | Formatted updates to match the current documentation |
| 1.6 | 05/06/19 | Roderick Barker | Final revision |
| Project Test Plan Documentation | | | |
| Version No. | Date | Author | Revision Description |
| 1.0 | 04/02/19 | Roderick Barker | Initial draft |
| 1.1 | 04/03/19 | Roderick Barker | Draft revision 1 |
| 1.2 | 04/20/19 | Roderick Barker | Merging of testing document to full document |
| 1.3 | 04/21/19 | Jeff Curtis, David Polen, Derek Sappington | Phase 1 Testing |
| 1.4 | 04/28/19 | Jeff Curtis, David Polen, Derek Sappington | Phase 2 testing |
| 1.5 | 04/28/19 | Jeff Curtis, Roderick Barker | Updated test tables documentation |
| 1.6 | 05/04/19 | Derek Sappington, Roderick Barker | Phase 3 testing and result updates |
| 1.7 | 05/06/19 | Roderick Barker | Final revision |
| Project User’s Guide Documentation | | | |
| Version No. | Date | Author | Revision Description |
| 1.0 | 04/02/19 | Roderick Barker | Initial draft |
| 1.1 | 04/03/19 | Roderick Barker, Jeff Curtis | Draft revision 1 |
| 1.2 | 04/14/19 | Roderick Barker, Jeff Curtis | Draft revision and merging of guide to main document |
| 1.3 | 04/28/19 | Roderick Barker, Jeff Curtis | Draft revision 2 |
| 1.4 | 05/06/19 | Roderick Barker | Final revision |

## Conclusions

For this project, we learned that the more we communicated, the faster the project rolled out. Our communications plan was very beneficial as we were able to stick to the schedule and get each phase done on time and get everything we wanted for that specific phase done. The one issue we faced and is probably true for most groups was time. Having only a handful of weeks to get a full project with corresponding documentation done was quite a task. Making sure we hit all the wants in our original plan was a bit of challenge. We were able to get it done and be somewhat ahead of schedule but with more time we could have done much more. Overall this project was a great learning experience, from seeing how it can be a bit of a challenge to get everyone to be able to sync up at the same time and to just making sure you do not overextend and have to cut back on your idea.

**Future Enhancements**

During the development of the visit applications, several areas were found that could benefit from future application development to enhance the product. These are:

1. Add database storage, local and cloud based. Adding this feature will allow data point stored in a relational manner giving the application much finer control over feature sets. For instance, using the appropriate tables, we could infer work locations, home locations, daily commutes and other interesting facts that require intelligence to the location data. Another benefit of adding cloud services for storage such as CloudKit is the ability to share locations with friends and incorporate a built-in secure messaging service.
2. Add additional data services such as Yelp, Foursquare and Google Places. This would allow us to present a more robust areas of interest feature. Currently we are limited to the small amount of data that Apple’s MapKit provides
3. Incorporate automatic trip and mileage detection. Since we store a user location and the visit, we can calculate the time required to move from one place to another with little additional effort, this in turn could be used to calculate mileage and time traveling. Since this is done automatically in the background, the user would not need to enter any additional data for this feature.

**Lesson Learned**

This application was written in Apple’s new programming language “Swift”, because of this, and the changing nature of the language, it was difficult at times to get the “Swift” way of programming. One of the key safety features of the language is the use of “optionals”, this made interfacing with Apple’s iOS frameworks more demanding than Objective-C. Since the language has gone through five major revisions, code example from previous version of language will not work, which can lead to frustration, especially for us new to the language. Finally, XCode’s static analysis tool was useless on Swift. This was solved by using the open source tool Tailor. Using this tool allowed our team to identify and fix over 400 language best practices, this made the code more readable and less prone to bugs.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Week 8 Report** | | | | | |
| **Team Members** | **Application Specifications** | **Program Documentation** | **Communications** | **Code Setup** | **Code Build and Familiarity** |
| **Jeff Curtis** | Outlined which portions of code went to each member | Checked documentation was finished | Contributed | Made sure full project was done and ran | Contributed |
| **David Polen** | Contributed | Checked documentation was finished | Contributed | Applied last minute corrections to code | Contributed |
| **Roderick Barker** | Contributed | Formatted all updates to match current documentation standards | Contributed | Contributed | Contributed |
| **Derek Sappington** | Performed final Static Analysis | Checked documentation was finished | Contributed | Corrected code errors from static analysis | Contributed |