*Florida International University*

*School of Computing and Information Sciences*

Software Engineering Focus

Final Deliverable

ADDIGY 6.0

**Team Members:**

Bhanu Boredha

Rupa Kotha

**Product Owner(s)**:

Jason Dettbarn.

**Mentor(s)**:

Mohsen Taheri

**Instructor**: Masoud Sadjadi

The MIT License (MIT)

Copyright (c) *2016 Florida International University*

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

***Abstract***

*Addigy 6.0*

Mobile phones have become one of the most valuable assets nowadays. IT intelligence to scale providers some sort of intelligence to assist when taking these assets. Addigy - Mobile Device Management. The purpose of this project is to create a system to be able to remotely monitor and apply configurations to mobile devices (iOS and Android). The project consists of two mobile applications (one for android and one for iOS) that communicate to a web service to report monitoring data and retrieve configurations. It also has a website component where an administrator can see the monitoring data and setup configurations.

**Table of Contents**

**Introduction** ………………………………………………………………………………………………………………………………..6

Current System ………………………………………………………………………………………………………………... 6

Purpose of New System ……………………………………………………………………………………………………………... 7

**User Stories**

Implemented User Stories …………………………………………………………………………………………………………..8

Pending User Stories ……………………………………………………………………………………………………..….. 16

**Project Plan**

Hardware and Software Resources ………………………………………………………………………………………….… 18

Sprints Plan ………………………………………………………………………………………………………………………. 20

*Sprint 1*  …………………………………………………………………………………………………………………………... 20

*Sprint 2*  …………………………………………………………………………………………………………………………... 21

*Sprint 3*  …………………………………………………………………………………………………………………………... 22

*Sprint 4*  …………………………………………………………………………………………………………………………... 24

*Sprint 5*  …………………………………………………………………………………………………………………………... 25

*Sprint 6*  …………………………………………………………………………………………………………………………... 27

**System Design**

Architectural Patterns …………………………………………………………………………………………………….. 28

**System Validation**  ……………………………………………………………………………………………………………….29

**Appendix**  ……………………………………………………………………………………………………………………………….34

Appendix A - UML Diagrams …………………………………………………………………………………………………….34

*Static UML Diagrams*  ……………………………………………………………………………………………………….34

*Dynamic UML Diagrams*  …………………………………………………………………………………………………..34

Appendix B - User Interface Design ……………………………………………………………………………………….…... 40

Appendix C - Sprint Review Reports ……………………………………………………………………………………...…… 46

Appendix D - User Manuals, Installation/Maintenance Document, Shortcomings/Wishlist Document and other documents …………………………………………………………………….…………… 48

**References** …………………………………………………………………………………………………………………...………...51

# 

# Introduction

The rapid growth in the IT monitoring market imposes the need for robust tools not only to monitor the current state of systems but to show historical information and timeline trends of the systems in their network.

The following sections provide a description of the characteristics and limitations of the current system as well as a description of a proposed system that will address some of these limitations.

## Current System

Addigy current system is a cloud based Mac Management platform that simplifies Mac IT Management. Which provides the following services.

* Asset Management – Encryption, Security & File Vault Keys.
* OS & Application management – OS versions, Updates & Apps.
* Policy Management/ Deployment -Take Control.

## 

## Purpose of New System:

Addigy - Mobile Device Management. The purpose of this project is to create a system to be able to remotely monitor and apply configurations to mobile devices (iOS and Android). The project consists of two mobile applications (one for android and one for iOS) that communicate to a web service to report monitoring data and retrieve configurations. It also has a website component where an administrator can see the monitoring data and setup configurations.

# User Stories

The following section provides the detailed user stories that were implemented in this iteration of the …. project. These user stories served as the basis for the implementation of the project’s features. This section also shows the user stories that are to be considered for future development.

## Implemented User Stories

**User Story #713: Preparation for the project:**

Setup and preparation for the project

### Acceptance Criteria:

1. meetings with the product owner
2. Knowing the project previous work
3. Installing the software’s required

**User Story #714: Preparation for the project:**

Setup and preparation for the project

### Acceptance Criteria:

### Meetings with the product owner

1. Knowing the project previous work
2. Installing the software’s required

**User Story #715:** **Installing python and Django:**

As a developer I need to install python and Django

### Acceptance Criteria:

1. Installation of python 3.5.2
2. Installation of Django

**User Story #716:** **Installing python and Django:**

As a developer I need to install python and Django

### Acceptance Criteria:

1. Installation of python 3.5.2
2. Installation of Django

**User Story #717: Develop a sample Django application**

Develop a sample Django application in which we can upload static files

**Acceptance Criteria:**

1. Creation of sample Django application.
2. Able to upload static files

**User Story #718: Develop a sample Django application**

Develop a sample Django application in which we can upload static files

**Acceptance Criteria:**

1. Creation of sample Django application.
2. Able to upload static files

## User Story #719: Installing Django on Ubuntu server

## As a developer I need to install a Django server on FIU Ubuntu server and run sample Django app that we have created from the Django server

## Acceptance Criteria:

## Login to the Django server.

## Install Python pip and Django on the server.

## Run sample Django App that we have created from the Django server

## User Story #720: Installing Django on Ubuntu server

## As a developer I need to install a Django server on FIU Ubuntu server and run sample Django app that we have created from the Django server

## Acceptance Criteria:

## Login to the Django server.

## Install Python pip and Django on the server.

## Run sample Django App that we have created from the Django server

## User Story #721: Installing Django on Ubuntu server

## As a developer I need to install a Django server on FIU Ubuntu server and run sample Django app that we have created from the Django server

## Acceptance Criteria:

## Login to the Django server.

## Install Python pip and Django on the server.

## Run sample Django App that we have created from the Django server

**User Story #722: Learn HTML, CSS, JavaScript and AngularJS.**

As a developer I need to learn HTML, CSS, JavaScript and AngularJS

### Acceptance Criteria:

1. Learn HTML and CSS.
2. Learn JavaScript
3. Learn AngularJS.

**User Story #723: Learn HTML, CSS, JavaScript and AngularJS.**

As a developer I need to learn HTML, CSS, JavaScript and AngularJS

### Acceptance Criteria:

1. Learn HTML and CSS.
2. Learn JavaScript
3. Learn AngularJS.

**User Story #725: Setting up Ionic Environment.**

As a Developer I need to setup Ionic Environment.

### Acceptance Criteria:

1. Install Node.js
2. Install cordova and ionic command line tools.
3. Install webstorm IDE.

**User Story #726: Setting up Ionic Environment.**

As a Developer I need to setup Ionic Environment.

### Acceptance Criteria:

1. Install Node.js
2. Install cordova and ionic command line tools.
3. Install webstorm IDE.

**User Story #731: Upload a picture to the server and set as wallpaper.**

As a user I need to upload a picture to the server, the server generates a configuration file to set the file, changing wallpaper of the device.

**Acceptance Criteria:**

1. Upload an image to the server.
2. Server should generate a configuration file to set the picture as wallpaper of the device.
3. The device consumes the file, changing the wallpaper of the device.

### Related Tasks:

​1. Research on .mobileconfig files.

**User Story #732: Upload a picture to the server and set as wallpaper.**

As a user I need to upload a picture to the server, the server generates a configuration file to set the file, changing wallpaper of the device.

**Acceptance Criteria:**

1. Upload an image to the server.
2. Server should generate a configuration file to set the picture as wallpaper of the device.
3. The device consumes the file, changing the wallpaper of the device.

### Related Tasks:

​1. Research on .mobileconfig files.

**User Story #733: Fetch contacts from device contacts and add contact.**

Develop an app that fetch and display the contacts from the mobile device contacts and add contact from the app.

### Acceptance Criteria:

1. Fetch and display the contacts.
2. Add contact to the mobile device.

### Related Tasks:

1. Install android studio.
2. Create an android virtual device.
3. Research on Ngcordova contacts plugin.

**User Story #734: Fetch contacts from device contacts and add contact.**

Develop an app that fetch and display the contacts from the mobile device contacts and add contact from the app.

### Acceptance Criteria:

1. Fetch and display the contacts.
2. Add contact to the mobile device.

### Related Tasks:

1. Install android studio.
2. Create an android virtual device.
3. Research on Ngcordova contacts plugin.

## User Story #735: Integrate Google Maps into ionic app.

Develop a hybrid app that integrates google maps into the app and which works on android and ios platforms.

### Acceptance Criteria:

1. Integrate google maps into the app.
2. App should run on android and IOS platforms.

### Related Tasks:

1. Install android studio.
2. Create an android virtual device.
3. Research on cordova google maps plugin.

## Pending User Stories

**…..**

# Project Plan

This section describes the planning that went into the realization of this project. This project incorporated the agile development techniques and as such required the sprints to be planned. These sprint planning are detailed in the section. This section also describes the components, both software and hardware, chosen for this project.

## Hardware and Software Resources

**IONIC FRAMEWORK:**

Ionic is a complete open-source SDK for hybrid mobile app development. Built on top of AngularJS and Apache Cordova, Ionic provides tools and services for developing hybrid mobile apps using Web technologies like CSS, HTML5, and Sass. Apps can be built with these Web technologies and then distributed through native app stores to be installed on devices by leveraging Cordova. Ionic was created by Max Lynch, Ben Sperry, and Adam Bradley of Drifty Co. in 2013.

**PYTHON:**

The Python framework, specifically the Django framework, is used on the backend as the project's web server. One of the primary reasons for choosing python was to be consistent with other existing server-side technologies used at Addigy. Other important reasons for using Python include:

* Quicker prototyping/development because due to high-level data types and dynamic typing. Existing web frameworks, like Django, that require minimal work to start serving requests from client applications
* Django is easily scalable

**AngularJS**

AngularJS allows for easy development of Views in an MVC pattern without having to build a complete MVC pipeline on the back-end. This project uses the backend to receive, process, store, and serve json data from client systems. AngularJS allows the project to make RESTful calls to the back end, receiving data for the front-end to display to the user on the dashboard view.

**MongoDB**

A NOSQL database simplifies storage and incorporates a date time stamp for processing historical records and trends. The following considerations led to select MongoDB among the NOSQL databases:

1. A document-based data model which is a rich data structure capable of holding arrays and other documents equivalent to JSON, Python dictionaries, etc.

2. Deep query-ability. MongoDB supports dynamic queries on documents using a document-based query language that's nearly as powerful as SQL.

3. No schema migrations. Since MongoDB is schema-free, the code defines the schema.

4. A clear path to horizontal scalability.

Sprints Plan

**SPRINT-1**

**User Story #713: Preparation for the project:**

Setup and preparation for the project

**Acceptance Criteria:**

1. meetings with the product owner

2. Knowing the project previous work

3. Installing the software’s required

**User Story #714: Preparation for the project:**

Setup and preparation for the project

**Acceptance Criteria:**

1. Meetings with the product owner

2. Knowing the project previous work

3. Installing the software’s required

**User Story #715: Installing python and Django:**

As a developer I need to install python and Django

**Acceptance Criteria:**

1. Installation of python 3.5.2

2. Installation of Django

**User Story #716: Installing python and Django:**

As a developer I need to install python and Django

**Acceptance Criteria:**

1. Installation of python 3.5.2

2. Installation of Django

## SPRINT-2

**User Story #717: Develop a sample Django application**

Develop a sample Django application in which we can upload static files

**Acceptance Criteria:**

1. Creation of sample Django application.

2. Able to upload static files

**User Story #718: Develop a sample Django application**

Develop a sample Django application in which we can upload static files

**Acceptance Criteria:**

1. Creation of sample Django application.

2. Able to upload static files

**SPRINT-3**

**User Story #719: Installing Django on Ubuntu server**

As a developer I need to install a Django server on FIU Ubuntu server and run sample Django app that we have created from the Django server

**Acceptance Criteria:**

1. Login to the Django server.

2. Install Python pip and Django on the server.

3. Run sample Django App that we have created from the Django server

**User Story #720: Installing Django on Ubuntu server**

As a developer I need to install a Django server on FIU Ubuntu server and run sample Django app that we have created from the Django server

**Acceptance Criteria:**

1. Login to the Django server.

2. Install Python pip and Django on the server.

3. Run sample Django App that we have created from the Django server

**User Story #721: Installing Django on Ubuntu server**

As a developer I need to install a Django server on FIU Ubuntu server and run sample Django app that we have created from the Django server

**Acceptance Criteria:**

1. Login to the Django server.

2. Install Python pip and Django on the server.

3. Run sample Django App that we have created from the Django server

**User Story #722: Learn HTML, CSS, JavaScript and AngularJS.**

As a developer I need to learn HTML, CSS, JavaScript and AngularJS

**Acceptance Criteria:**

1. Learn HTML and CSS.

2. Learn JavaScript

3. Learn AngularJS.

**User Story #723: Learn HTML, CSS, JavaScript and AngularJS.**

As a developer I need to learn HTML, CSS, JavaScript and AngularJS

**Acceptance Criteria:**

1. Learn HTML and CSS.

2. Learn JavaScript

3. Learn AngularJS.

**User Story #725: Setting up Ionic Environment.**

As a Developer I need to setup Ionic Environment.

**Acceptance Criteria:**

1. Install Node.js

2. Install cordova and ionic command line tools.

3. Install webstorm IDE.

**User Story #726: Setting up Ionic Environment.**

As a Developer I need to setup Ionic Environment.

**Acceptance Criteria:**

1. Install Node.js

2. Install cordova and ionic command line tools.

3. Install webstorm IDE.

**SPRINT-4**

**User Story #731: Upload a picture to the server and set as wallpaper.**

As a user I need to upload a picture to the server, the server generates a configuration file to set the file, changing wallpaper of the device.

**Acceptance Criteria:**

1. Upload an image to the server.
2. Server should generate a configuration file to set the picture as wallpaper of the device.
3. The device consumes the file, changing the wallpaper of the device.

### Related Tasks:

​1. Research on .mobileconfig files.

**User Story #732: Upload a picture to the server and set as wallpaper.**

As a user I need to upload a picture to the server, the server generates a configuration file to set the file, changing wallpaper of the device.

**Acceptance Criteria:**

1. Upload an image to the server.
2. Server should generate a configuration file to set the picture as wallpaper of the device.
3. The device consumes the file, changing the wallpaper of the device.

**SPRINT-5**

**User Story #733: Fetch contacts from device contacts and add contact.**

Develop an app that fetch and display the contacts from the mobile device contacts and add contact from the app.

### Acceptance Criteria:

1. Fetch and display the contacts.
2. Add contact to the mobile device.

### Related Tasks:

1. Install android studio.
2. Create an android virtual device.
3. Research on Ngcordova contacts plugin.

**User Story #734: Fetch contacts from device contacts and add contact.**

Develop an app that fetch and display the contacts from the mobile device contacts and add contact from the app.

### Acceptance Criteria:

1. Fetch and display the contacts.
2. Add contact to the mobile device.

### Related Tasks:

1. Install android studio.
2. Create an android virtual device.
3. Research on Ngcordova contacts plugin.

**SPRINT-6**

## User Story #735: Integrate Google Maps into ionic app.

Develop a hybrid app that integrates google maps into the app and which works on android and ios platforms.

### Acceptance Criteria:

1. Integrate google maps into the app.
2. App should run on android and IOS platforms.

### Related Tasks:

1. Install android studio.
2. Create an android virtual device.
3. Research on cordova google maps plugin.

## User Story #736: Integrate Google Maps into ionic app.

Develop a hybrid app that integrates google maps into the app and which works on android and ios platforms.

### Acceptance Criteria:

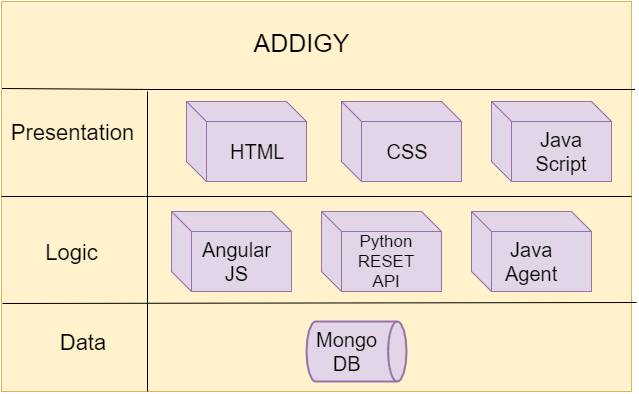
1. Integrate google maps into the app.
2. App should run on android and IOS platforms.

### Related Tasks:

1. Install android studio.
2. Create an android virtual device.
3. Research on cordova google maps plugin.

# System Design

This section contains information on the design decisions that went into this project. The architecture patterns are outlined and explained. The entire system is shown in a package diagram and the subsystems are explained. Finally, the design patterns used in the project are discussed.



# System Validation:

**User Story #731: Upload a picture to the server and set as wallpaper.**

Test Case 1(Sunny Day)

* Purpose: User should be able to set an image from the gallery as the wallpaper.
* Precondition: User should go to the settings page and select ‘change background

option’.

* Input: User should select an image to set as background image.
* Expected Result: The image should be set as the background image.
* Actual Result: The image set as the background image.
* Status: SUCCESS

Test Case 2(Sunny Day)

* Purpose: User should be able to change background image.
* Precondition: User should go to the settings page and select ‘change background option’.
* Input: User should select an image to set as background image.
* Expected Result: The image should be set as the background image.
* Actual Result: The image set as the background image.
* Status: SUCCESS

**User Story #732: Upload a picture to the server and set as wallpaper.**

Test Case 1(Sunny Day)

* Purpose: User should be able to set an image from the gallery as the wallpaper.
* Precondition: User should go to the settings page and select ‘change background

option’.

* Input: User should select an image to set as background image.
* Expected Result: The image should be set as the background image.
* Actual Result: The image set as the background image.
* Status: SUCCESS

**Test Case 2(Sunny Day)**

* Purpose: User should be able to change background image.
* Precondition: User should go to the settings page and select ‘change background option’.
* Input: User should select an image to set as background image.
* Expected Result: The image should be set as the background image.
* Actual Result: The image set as the background image.
* Status: SUCCESS

**User Story #733: Fetch contacts from device contacts and add contact.**

**Test Case 1(Sunny Day)**

* Purpose: User pushes the load button and app fetches the contacts from the device phonebook.
* Precondition: User should open the Contacts app.
* Input: User pushes the load button.
* Expected Result: The app should load contacts from the device phonebook.
* Actual Result: App loads the contacts
* Status: SUCCESS

**Test Case 2(Sunny Day)**

* Purpose: User pushes the new button and app adds the contact to the device phonebook.
* Precondition: User should open the Contacts app.
* Input: User pushes the new button.
* Expected Result: App should add new contact to the device phonebook.
* Actual Result: Contact is added to the device phonebook.
* Status: SUCCESS

**User Story #734: Fetch contacts from device contacts and add contact.**

**Test Case 1(Sunny Day)**

* Purpose: User pushes the load button and app fetches the contacts from the device phonebook.
* Precondition: User should open the Contacts app.
* Input: User pushes the load button.
* Expected Result: The app should load contacts from the device phonebook.
* Actual Result: App loads the contacts
* Status: SUCCESS

**Test Case 2(Sunny Day)**

* Purpose: User pushes the new button and app adds the contact to the device phonebook.
* Precondition: User should open the Contacts app.
* Input: User pushes the new button.
* Expected Result: App should add new contact to the device phonebook.
* Actual Result: Contact is added to the device phonebook.
* Status: SUCCESS

## User Story #735: Integrate Google Maps into ionic app.

**Test Case 1(Sunny Day)**

* Purpose: User Opens the Ionic maps app and app displays the maps from the google maps API.
* Precondition: User should open the Ionic maps app.
* Input: User opens the app.
* Expected Result: The app should display maps.
* Actual Result: App displays the maps.
* Status: SUCCESS

**Test Case 2(Sunny Day)**

* Purpose: User pushes the satellite button and the app displays the satellite view of the maps.
* Precondition: User should open the Ionic maps app.
* Input: User pushes the satellite button.
* Expected Result: App should display the satellite view.
* Actual Result: Satellite view is displayed.
* Status: SUCCESS

## User Story #736: Integrate Google Maps into ionic app.

**Test Case 1(Sunny Day)**

* Purpose: User Opens the Ionic maps app and app displays the maps from the google maps API.
* Precondition: User should open the Ionic maps app.
* Input: User opens the app.
* Expected Result: The app should display maps.
* Actual Result: App displays the maps.
* Status: SUCCESS

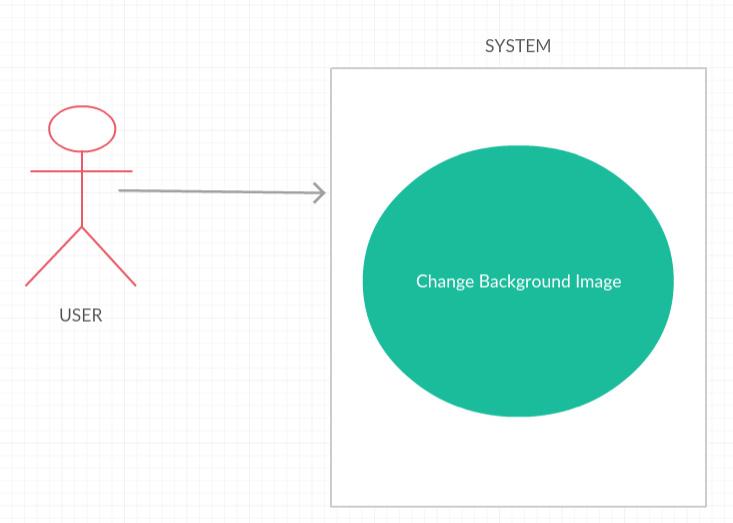
**Test Case 2(Sunny Day)**

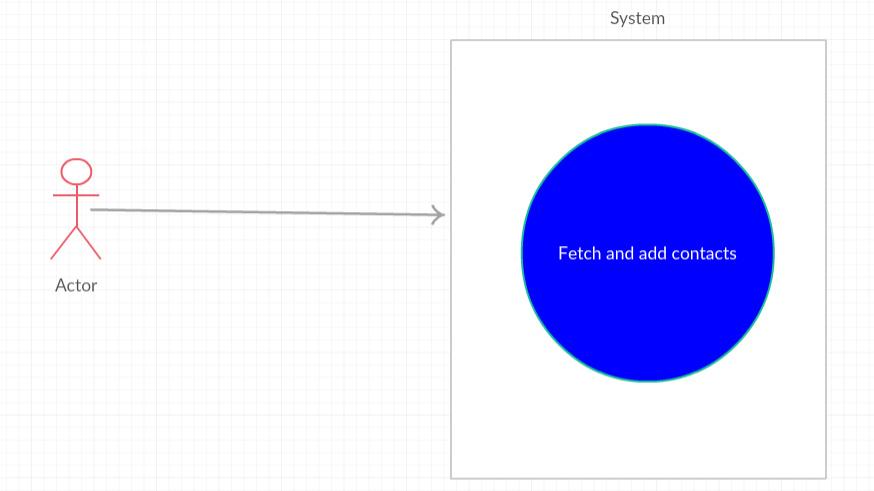
* Purpose: User pushes the satellite button and the app displays the satellite view of the maps.
* Precondition: User should open the Ionic maps app.
* Input: User pushes the satellite button.
* Expected Result: App should display the satellite view.
* Actual Result: Satellite view is displayed.
* Status: SUCCESS

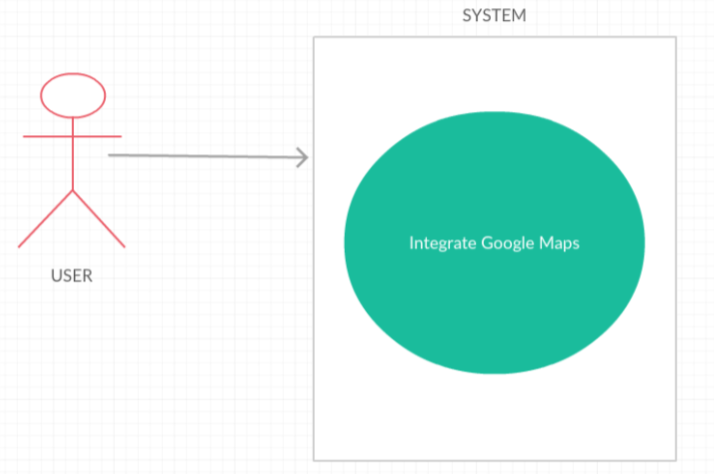
# 

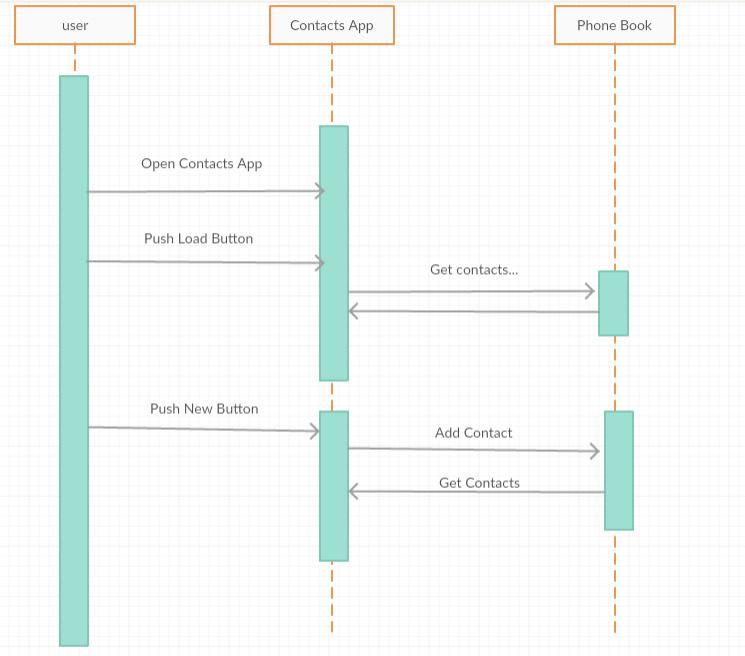
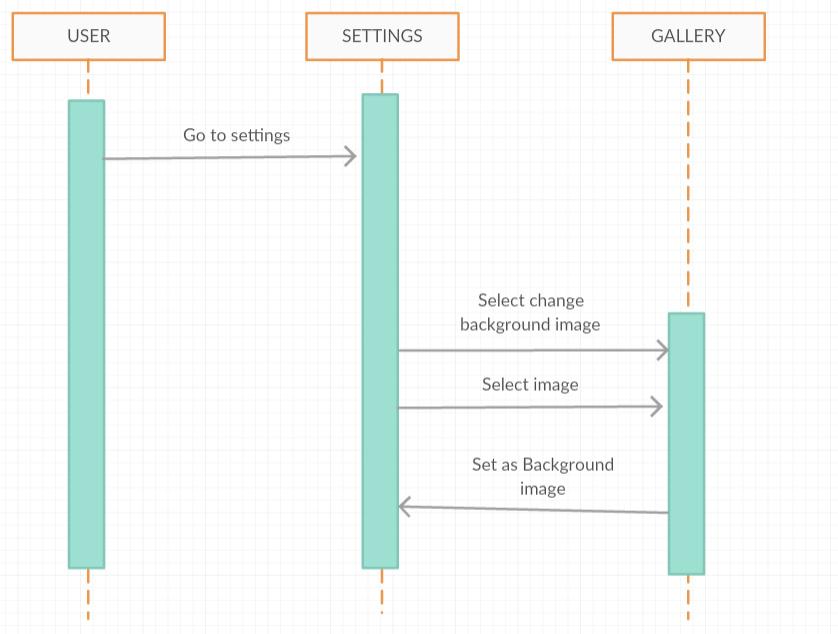
# Appendix

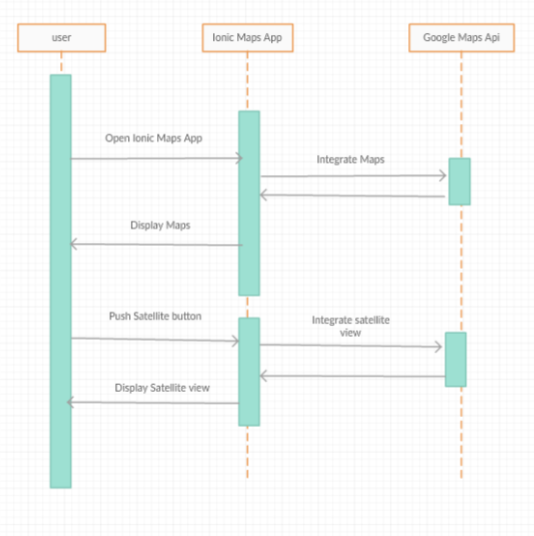
## Appendix A - UML Diagrams

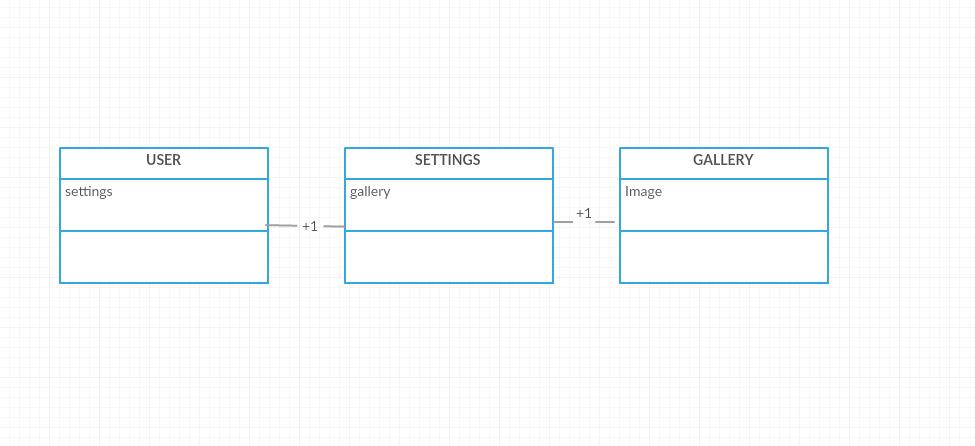


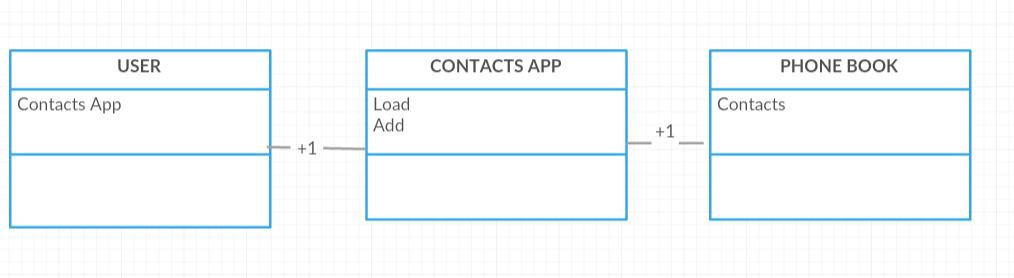


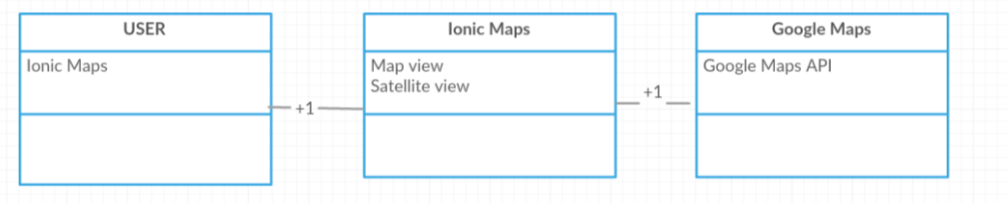






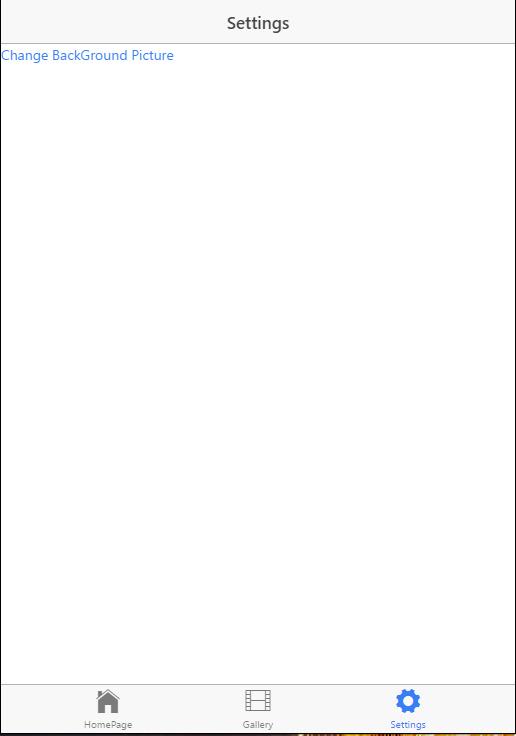
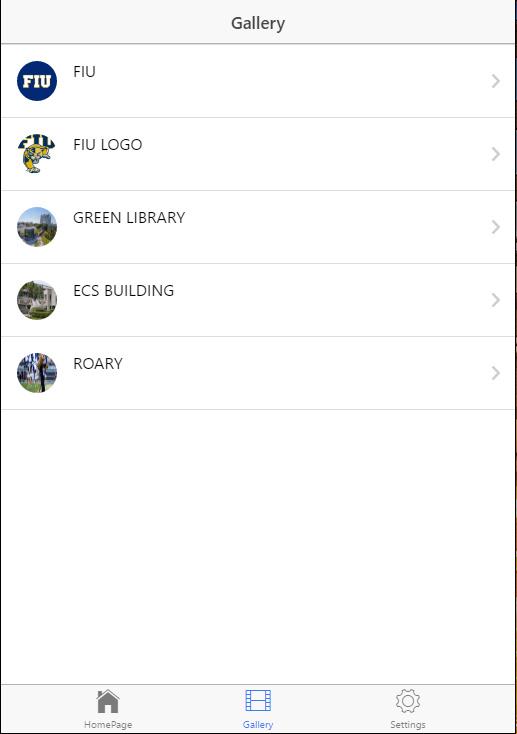




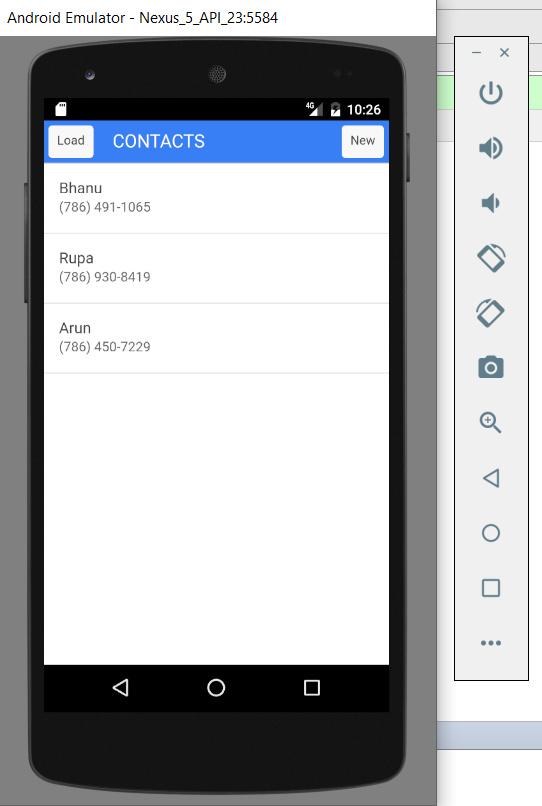
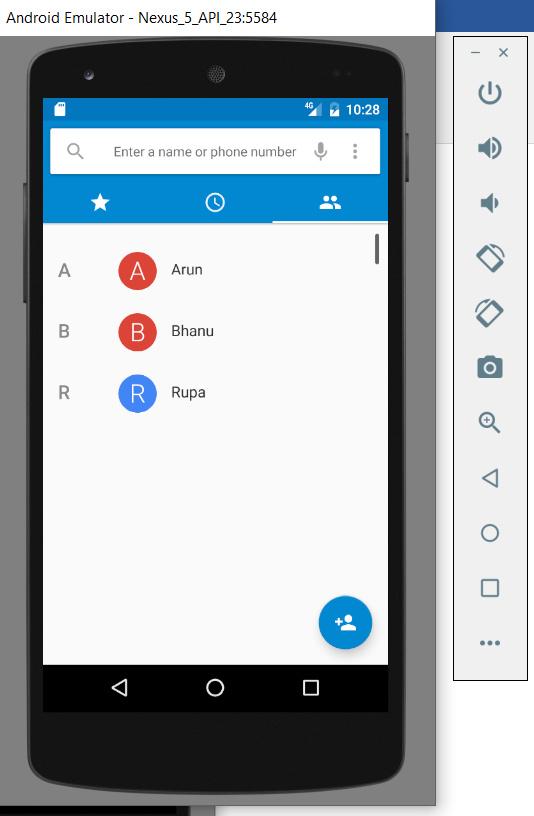


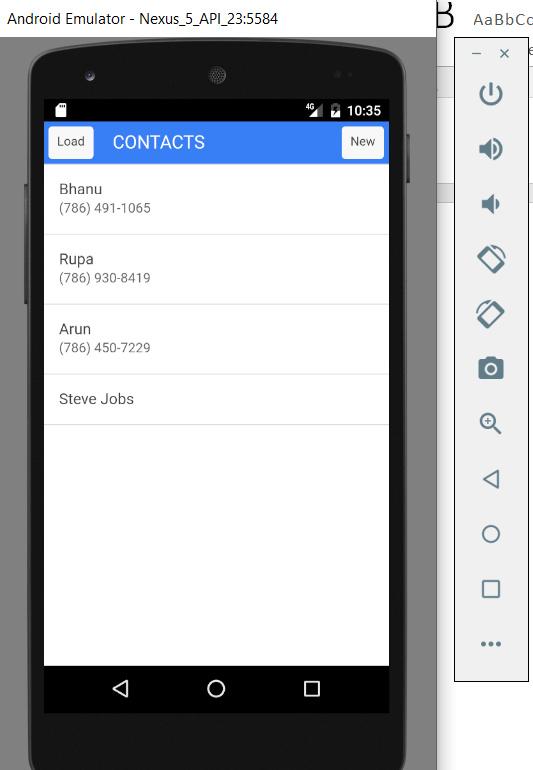
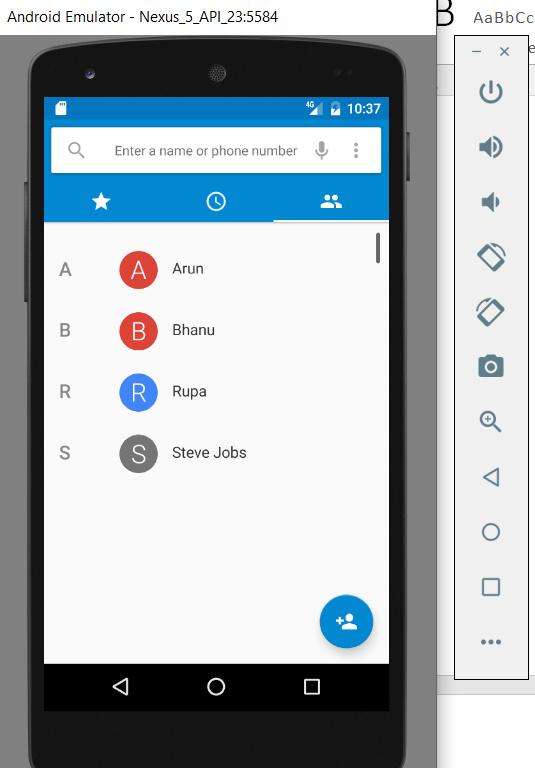
## 

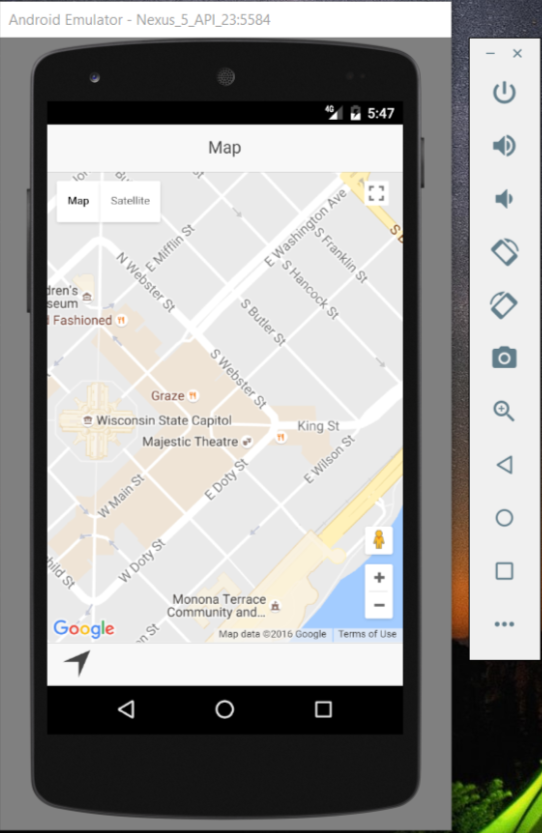
## Appendix B - User Interface Design

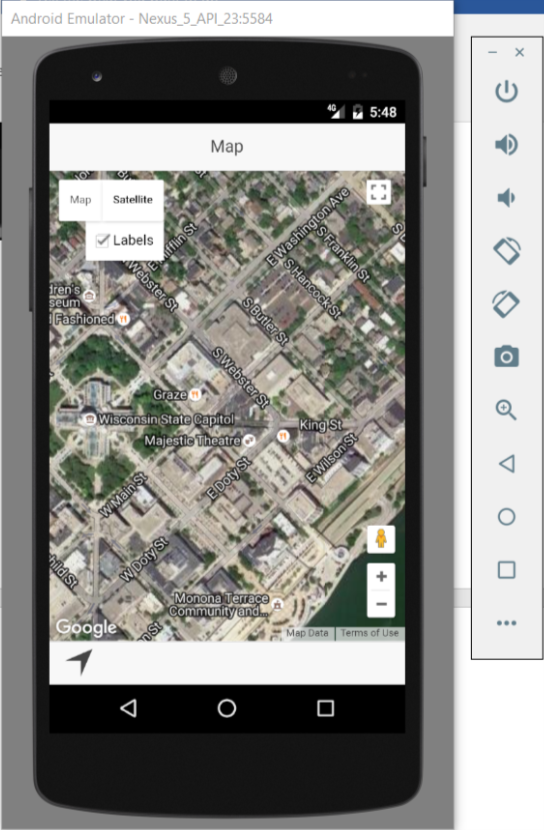
 







## 

## Appendix C - Sprint Review Reports

**SPRINT 2 REVIEW:**

Attendees: <Product Owner(Jason), Javier,  Bhanu, Rupa>

Start time: 20:30

End time: 21:00

After a show and tell presentation, the implementation of the following user stories were accepted by the product owners: All.

* Creation of a sample Django application  #720
* Creation of a Sample Django application #721

**SPRINT 3 REVIEW:**

Attendees: <Product Owner(Jason), Javier,  Bhanu, Rupa>

Start time: 20:30

End time: 21:00

After a show and tell presentation, the implementation of the following user stories were accepted by the product owners: All.

* Installing Django on Ubuntu Server #722
* Installing Django on Ubuntu Server #723
* Learn HTML, CSS, Javascript and Angularjs #724
* Learn HTML, CSS, Javascript and Angularjs #725
* Setting up ionic Environment #726
* Setting up ionic Environment #727
* Develop an ionic App #728
* Develop an ionic App#729

**SPRINT 4 REVIEW:**

Attendees: <Product Owner(Jason), Javier,  Bhanu, Rupa>

Start time: 17:30

End time: 18:00

After a show and tell presentation, the implementation of the following user stories were accepted by the product owners:

* Upload a picture to the server and set as wallpaper#730
* Upload a picture to the server and set as wallpaper#731

**SPRINT 5 REVIEW:**

Attendees: <Product Owner(Jason), Javier,  Bhanu, Rupa>

Start time: 17:30

End time: 18:00

After a show and tell presentation, the implementation of the following user stories were accepted by the product owners:

* Fetch contacts from device contacts and add contact.#732
* Fetch contacts from device contacts and add contact.#733

**SPRINT 6 REVIEW:**

Attendees: <Product Owner(Jason), Javier,  Bhanu, Rupa>

Start time: 17:30

End time: 18:00

After a show and tell presentation, the implementation of the following user stories were accepted by the product owners:

* Integrate Google Maps into ionic app.#733
* Integrate Google Maps into ionic app..#734

## 

## Appendix D - Sprint Retrospective Reports

**SPRINT-2  RETROSPECTIVE:**

Attendees: <Product Owner(Jason), Javier,  Bhanu, Rupa>

Start time: 20:30

End time: 21:00

What went wrong?

* Did we do a good job estimating our team's velocity?
  + Mostly
* Did we do a good job estimating the points (time required) for each user story?
  + Mostly
* Did each team member work as scheduled?
  + Mostly

What went right?

* Have learnt how the Django architecture works.

How to address the issues in the next sprint?

* How to improve the process?
  + We are trying very hard to understand the environment and Django framework and try to start the actual project soon.

**SPRINT-3  RETROSPECTIVE:**

Attendees: <Product Owner(Jason), Javier,  Bhanu, Rupa>

Start time: 20:30

End time: 21:00

What went wrong?

* Did we do a good job estimating our team's velocity?
  + Mostly
* Did we do a good job estimating the points (time required) for each user story?
  + Mostly
* Did each team member work as scheduled?
  + Mostly

What went right?

* We have learned HTML, CSS, JavaScript and AngularJS and we have build an app using ionic framework.

How to address the issues in the next sprint?

* How to improve the process?
  + We have been trying very hard learning HTML, CSS, JavaScript and AngularJS . and understanding the ionic framework and we will try to start the actual project soon.

**SPRINT-4  RETROSPECTIVE:**

Attendees: <Product Owner(Jason), Javier,  Bhanu, Rupa>

Start time: 20:30

End time: 21:00

What went wrong?

* Did we do a good job estimating our team's velocity?
  + Mostly
* Did we do a good job estimating the points (time required) for each user story?
  + Mostly
* Did each team member work as scheduled?
  + Mostly

What went right?

* We are able to change the background image of the device from the gallery.

How to address the issues in the next sprint?

* How to improve the process?

We need to work on uploading pictures to the server and the device detects the images and displays in the gallery and then setting those images to as the background image.

**SPRINT-5  RETROSPECTIVE:**

Attendees: <Product Owner(Jason), Javier,  Bhanu, Rupa>

Start time: 20:30

End time: 21:00

What went wrong?

* Did we do a good job estimating our team's velocity?
  + Mostly
* Did we do a good job estimating the points (time required) for each user story?
  + Mostly
* Did each team member work as scheduled?
  + Mostly

What went right?

* We are able to fetch and add contacts to the mobile device phone book.

How to address the issues in the next sprint?

* How to improve the process?

We need to create a form in which we can submit the contact name and phone number.

**SPRINT-6  RETROSPECTIVE:**

Attendees: <Product Owner(Jason), Javier,  Bhanu, Rupa>

Start time: 20:30

End time: 21:00

What went wrong?

* Did we do a good job estimating our team's velocity?
  + Mostly
* Did we do a good job estimating the points (time required) for each user story?
  + Mostly
* Did each team member work as scheduled?
  + Mostly

What went right?

* We are able to integrate google maps into the ionic app.

How to address the issues in the next sprint?

* How to improve the process?

We need to integrate the present location.

# References