CIS 4911 – SENIOR PROJECT

Picture Marketing’s Social Wall Ver.2

Software Requirement Document

Members: Steve Noel

Mentor: Carlos Ocampo

Instructor: Masoud Sadjadi

April 30, 15

**Abstract**

The Social Wall is a an IOS mobile application that allows registered users to show their images as a slideshow on any display using a Chrome cast device. The purpose of this document is to outline in detail the Social Wall system, in particular the necessary functionalities the system should have.

**COPYRIGHTS AND TRADEMARK NOTICES**

This is the work of Steve Noel, unless specified otherwise. Carlos Ocampo and Louis Zuckerman of Picture Marketing provided aid and guidance.

**Table of Contents**

|  |  |
| --- | --- |
| **Title** | **Page Number(s)** |
| 1. Introduction | 4 |
| 1.1. Problem Definition | 4 |
| 1.2. Scope of the System | 4 |
| 1.3. Terminology | 4-5 |
| 1.4. Overview of document | 5-6 |
| 2. Current System | 6 |
| 3. Project Plan | 7-9 |
| 3.1. Project Organization | 7 |
| 3.2. Work Breakdown | 8 |
| 3.3. Cost Estimate | 9 |
| 4. Proposed System Requirements | 9-21 |
| 4.1. Functional Requirements | 10-13 |
| 4.2. Analysis of System Requirements | 14-23 |
| 4.2.1. Scenarios | 14-16 |
| 4.2.2. Use case model | 17 |
| 4.2.3.Static model | 18-19 |
| 4.2.4. Dynamic model | 20-23 |
| 5. Glossary | 24 |
| 6. Appendix | 23-46 |
| 6.1. Appendix A - Complete Use Cases | 23-32 |
| 6.2. Appendix B - Use Case Diagram | 32 |
| 6.3. Appendix C - Static UML Diagrams | 32-34 |
| 6.4. Appendix D - Dynamic UML Diagrams | 35-39 |
| 6.5. Appendix E – User Interface Designs | 40-41 |
| 6.6. Appendix F - Diary of Meetings and Tasks | 42-46 |
| 7. References | 47 |

**1. Introduction**

This chapter introduces the project and defines the purpose, scope, terms, and acronyms on the system. It also deals with describing how the rest of this document will unfold using the chapter and section structure of the document.

**1.1 Problem Definition**

The Social Wall system aims to solve the inability of our users to easily and efficiently display their online photo albums in a slideshow format on any large screen they desire without the hassle of a complicated setup process. The system should also allow users to seamlessly customize their slideshows on the fly without hindering user experience. With todays highly accessible software and hardware, the Social Wall project combines the latest innovations in mobile and web technologies to help users free and view their media wherever they want.

**1.2 Scope of System**

For this project, an IOS mobile application will be created. Also, a Chrome cast receiver web application will be created that will run on a Chrome cast device. The mobile application will be able to connect to the Chrome cast device and launch the receiver application on the device. The system will be limited to IOS devices running IOS 6.0 and above and will be able to login to the FotoZap system and display a users images on the screen connected to the Chrome cast device.

**1.3 Terminology - Definitions, Acronyms, and Abbreviations**

PM - Picture Marketing

SW - Social Wall

Cordova - Native mobile framework

h/w - Hardware

s/w - Software

OS - Operating System

App - application

CC - Chrome Cast

AngularJS- Front –end MVC javascript framework

Javascript – web programming language

CSS - Cascading Style Sheets language

ConnectSdk – Framework for connecting to Chrome cast device.

Sencha Touch – A Mobile Web App Framework

HDTV – High Definition Television

MVC – Model-View-Controller architectural pattern

JSON – Javascript Object Notation (data serialization format)

FotoZap – A Picture Marketing application for capturing and storing branded images.

**1.4 Overview of document**

The document is divided into 7 chapters with many having inner sections. In Chapter two, the current system and its limitations and shortcomings are discussed. In the third chapter the project plan of the system is outlined, in particular, the breakdown of the work for the project , the individual roles in the project and finally the projects estimated cost of development. In chapter four the system requirements are laid out and analyzed in detail using static and dynamic models. Chapter five is the glossary of terms used throughout the document. Chapter six incorporates the UML diagrams describing the system, the screenshots of the system and the meeting diaries for the project. Finally, chapter seven shows the references used for the project.

**2. Current System (Limitations and Problems)**

Currently, there are two major ways of displaying an image slideshow on a large screen: the local solution, which uses applications such as Microsoft PowerPoint to display a image slideshow from a computer and the web solution which uses a web application such as Tintup to show the slideshow using a web service. The local approach requires the user to download the images on their computer and connect to the screen or connect a flash drive device to the screen, which requires some initial setup and limits the control of the slideshow. This approach forces companies to dedicate time and resources for the creation of an appealing final product.

The second option using web applications such as Tintup or Postano which offer similar functionalities as the Social Wall System however, these, services while useful, can be rather costly, and can cost upwards of $1,000 a month (or more). This option may be viable for some companies, however it may not be a viable solution for smaller companies and some individuals.

Moreover, there are mobile applications available that can accomplish slideshow streaming to screens such as AllCast or Photocast. These are open source solutions that offer similar features however they are limited in the sources that can be used for the slideshows, they do not offer FotoZap users to view their pictures and their limited in the actions that can be performed on the image slideshow to dynamically customize it. Finally, the Social Wall system version one used a web application instead of a mobile application, which offers very similar functionalities that the mobile system provides but the mobile system is more intuitive and easier to use since most people have mobile devices and are used to using mobile applications.

**3. Project Plan**

This chapter describes the plan of action for the software requirements document. Here, the member roles, work breakdown and cost estimation for the project are outlined. The work schedule is presented in a tabular format and the cost estimation for the project has taxes and other fees included.

**3.1. Project Organization**

Team Member Roles:

Steve Noel – Scrum Master, Document Editor, UML Diagram Creator, Interface designer, and lead Developer.

Carlos Ocampo- Mentor

Hardware:

* Mac Computer (OS X or higher)
* Chrome Cast device
* Television with HDMI port

Software:

* StarUML (version 2.0.0)
* Mingle (Project Management Tool)
* Google Chrome Cast IOS Framework
* Google Drive
* GitHub (version 2.0 for Mac)
* Apache Server
* Xcode IOS Development Environment
* Cordova Native App Framework
* Sencha Touch Mobile App Framework
* Sublime Text 2 (Text Editor)
* AngularJS MVC Javascript Front-End framework

**3.2. Work Breakdown**

|  |  |
| --- | --- |
| **Deliverables** | **Due Date** |
| Feasibility Study and Project Plan | 4/30/2015 |
| Software Requirements Document | 4/30/2015 |
| Design Document | 4/30/2015 |
| Final Document | 4/30/2015 |
| Installation Guide | 4/30/2015 |
| User Manual | 4/30/2015 |
| Project Videos | 4/30/2015 |

|  |  |
| --- | --- |
| **Milestone** | **Due Date** |
| Create Hybrid IOS Application | 2/7/2015 |
| Connect/Disconnect from Chrome cast device | 2/26/2015 |
| Launch CC Receiver App on Chrome cast | 2/26/2015 |
| Send Messages to CC receiver App | 3/10/2015 |
| Develop Slideshow Engine | 3/28/2015 |
| Rejoin Web App | 4/17/2015 |

**3.3. Cost Estimate**

|  |  |
| --- | --- |
| Resource | Price (USD) |
| Mac Laptop running OS X or better, 4 GB RAM, Intel i5 processor or better | $1300 |
| HDTV with HDMI port | $107.50 |
| Chrome Cast device | $34.99 |
| iPhone 5s | $600 |
| Total (taxes and fees included): | $3449.99 |

**4. Proposed System Requirements**

This chapter discusses the functionalities the system should include. In addition to the system requirements this chapter discusses the non-functional requirements of the system or the constraints the system should abide by in order to be a viable solution for the Picture Marketing the Product Owner. For a further in depth analysis of the system requirements the UML diagrams of the use cases are shown in this chapter along with the scenarios that a typical user might encounter using the system.

**4.1 Functional Requirements**

1) The system shall allow the user to login to the system.

Non-Functional Requirements:

Usability- No previous training time needed.

Readability

* 1. Mean time to Failure – 5% failures for every twenty-four hours of operation is acceptable.
  2. Availability – Down time for Login Back-up 15 minutes in a 24-hour period.

Performance

1. Request should be sent and received within 5 secs.
2. System should be able to handle 100 request in 1 minute.

Supportability-

a) System should be supported on IOS mobile devices.

2) The system shall allow the user to logout from the system.

Non-Functional Requirements:

Usability- No previous Training time needed.

Readability

* 1. Mean time to Failure – 3% failures for every twenty-four hours of operation is acceptable.
  2. Availability – Down time for 10 minutes in a 24-hour period.

Performance

1. Request should be sent within 5 secs.
2. System should be able to handle 1000 request in 1 minute.

Supportability-

a) System should be supported on IOS mobile device.

3) The system shall allow the user to connect to a Chrome cast device.

Non-Functional Requirements:

Usability- No previous Training time needed.

Readability

* 1. Mean time to Failure – 10% failures for every twenty-four hours of operation is acceptable.
  2. Availability – Down time for 15 minutes in a 24-hour period.

Performance

1. Request should be sent within 5 secs.
2. System should be able to handle 100 requests in 1 minute.

Supportability-

a) System should be supported on IOS mobile device.

4) The system shall allow the user to disconnect from a Chrome cast device.

Non-Functional Requirements:

Usability- No previous Training time needed.

Readability

* 1. Mean time to Failure – 10% failures for every twenty-four hours of operation is acceptable.
  2. Availability – Down time for 15 minutes in a 24-hour period.

Performance

1. Request should be sent within 5 secs.
2. System should be able to handle 100 requests in 1 minute.

Supportability-

a) System should be supported on IOS mobile device.

5) The system shall allow the user to select a campaign.

Non-Functional Requirements:

Usability- No previous Training time needed.

Readability

* 1. Mean time to Failure – 5% failures for every twenty four hours of operation is acceptable.
  2. Availability – Down time for 15 minutes in a 24 hour period.

Performance

1. Request should be sent and received within 5 secs.
2. System should be able to handle 100 requests in 1 minute.

Supportability-

a) System should be supported on IOS mobile devices.

6) The system shall allow the user to view the Campaign on the Chrome cast.

Non-Functional Requirements:

Usability- No previous Training time needed.

Readability

* 1. Mean time to Failure – 5% failures for every twenty four hours of operation is acceptable.
  2. Availability – Down time for 10 minutes in a 24 hour period.

Performance

1. Request should be sent and received within 5 secs.
2. System should be able to handle 100 requests in 1 minute.

Supportability-

a) System should be supported on IOS mobile devices.

.

**4.2 Analysis of System Requirements**

**4.2.1. Scenarios**

Scenario 1) Login

Purpose: Scenario that describes the use of the login system of IOS app by a user.

Individual: User registered on the system.

Equipment: IOS mobile device running IOS 6.0 and above

Scenario:

1) User opens application on mobile device.

2) System displays login form.

3) User enters their username and password into the form.

4)User presses login button

5) System authenticates user and displays the users campaigns selecting page.

6) As an alternative the user can submit incorrect credentials then the system notifies the user that the creadentials submitted where incorrenct and allows the user to try again.

Scenario 2) Display Campaign Image Slideshow on Screen

Purpose: Scenario that describes the use of the display images system of IOS app by a user.

Individual: User registered on the system.

Equipment: IOS mobile device running IOS 6.0 and above

Scenario:

1) User opens application on mobile device.

2) System displays login form.

3) User enters his username and password into the form.

4) User presses login button

5) System authenticates user and displays the users campaign selecting page.

5)User selects the connect to Chrome cast button

6)System connects to Chrome cast device

7) The user selectst a campaign to view.

8) The user selects the show on Chrome cast button

9) System shows the seleced campaign on the Chrome cast screen.

Scenario 3) Logout

Purpose: Scenario that describes the use of the logout system of IOS app by a user.

Individual: User registered on the system.

Equipment: IOS mobile device running IOS 6.0 and above

Scenario:

1) User opens application on mobile device.

2) System displays login form.

3) User enters his username and password into the form.

4) User presses login button.

5) System authenticates user and displays the users campaign selecting page.

6) The user selects the logout button.

7) System destroys current view and redirects user to login screen.

Scenario 4) Connect t to Chrome cast device

Purpose: Scenario that describes the use of the connect to Chrome cast feature of the mobile app.

Individual: User registered on the system.

Equipment: IOS mobile device running IOS 6.0 and above

Scenario:

1) User opens application on mobile device.

2) System displays login form.

3) User enters his username and password into the form and presses login button.

4) System authenticates user and displays the users campaign selecting page.

5) The user selects the connect to Chrome cast button

6) System returns a confirmation whether the system was connected to the Chromecast.

Scenario 5) Disconnect t to Chrome cast device

Purpose: Scenario that describes the use of the Disconnect to Chrome cast feature of the mobile app.

Individual: User registered on the system.

Equipment: IOS mobile device running IOS 6.0 and above

Scenario:

1) User opens application on mobile device.

2) System displays login form.

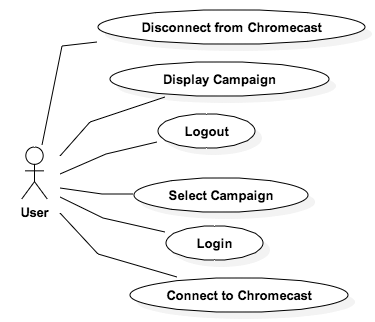
3) User enters his username and password into the form.

4) System authenticates user and displays the users campaign selecting page.

5) The user selects the disconnect to Chrome cast button

6) System returns a confirmation whether the system was disconnected from the Chromecast.

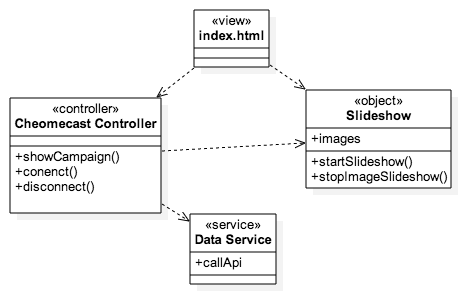
**4.2.2. Use case model**

****

**4.2.3. Static model**



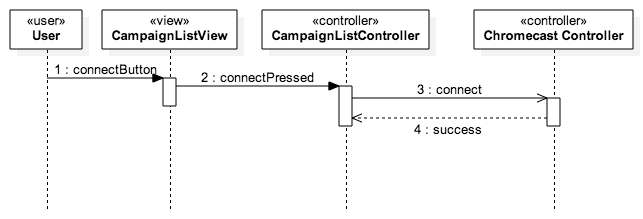
Sender Application Class Diagram



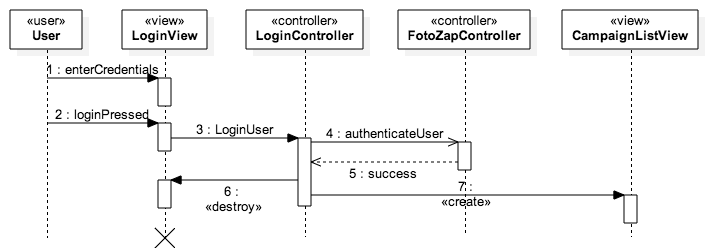
Custom Chrome cast receiver application Class Diagram.

**4.2.4. Dynamic model**

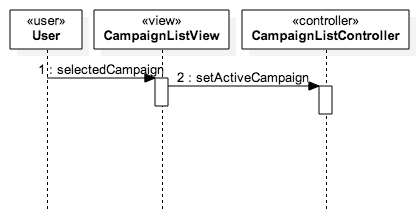
**Connect to Chrome cast Sequence Diagram**



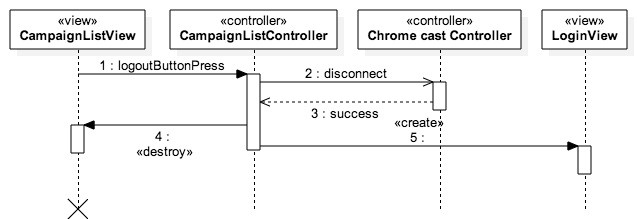
**Login Sequence Diagram**



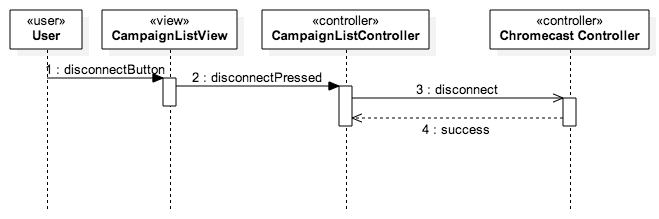
**Select Campaign Sequence Diagram**



**Logout Sequence Diagram**



**Disconnect from Chrome cast Sequence Diagram**



**Show Campaign on Chrome cast**



**5. Glossary**

|  |  |
| --- | --- |
| Term | Definition |
| Slideshow or Carousel | An collection of images that are displayed one at a time. |
| Dynamic | Capable of action or change |
| Static | Stationary, fixed |
| Functional Requirements | Features that are integral to the desired output of the system |
| Non-functional Requirements | Constraints on the system that determine the quality of said system. |

**6. Appendix**

**6.1. Appendix A - Complete use cases**

**Use Case ID:** S-WALL/2-001/Login

**Use Case Level:** High Level

**Details:**

* **Actor:** User
* **Preconditions:**

1. User must have opened the application on their mobile device.

* **Description:**

1. The use case begins when the user enters his credentials, username and password for example johndoe and password321.

2. The user then presses the “Login” button.

3. The system shall send the credentials to the server for authentication.

4. The use case ends when the server responds with a success json message and the system destroys the current view and changes to the campaign list view.

**Post conditions:**

1. The user is in the Campaign List View of the system.

**Alternative Courses of Action**: N/A

**Exceptions:**

* The system is unable to send the credentials to the server.
* There is no Internet connection on the device.

**Related Use Cases:**

**Logout**

**Decision Support:**

* **Frequency:** Will be used every time the software is used.
* **Criticality:** High. Core functionality of software.
* **Risk:** Low. Dependent upon external entities.

**Constraints:**

**Modification History:**

* **Owner: Steve Noel**
* **Initiation Date:** February 6, 2015
* **Last Modified:** April 9, 2015

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Use Case ID:** S-WALL/2-002/Logout

**Use Case Level:** High Level

**Details:**

* **Actor:** User
* **Preconditions:**

1. User must have opened the application on their mobile device.

2. User must be logged into the System.

* **Description:**

1. The use case begins when the user presses the “Logout” button.

2. The system prompts the user whether he wants to continue or not.

3. The user then presses the okay button.

4. The use case ends when the system resets the state variables, closes the current view and navigates to the main or login screen.

**Post conditions:**

1. The user is in the login screen of the system.

**Alternative Courses of Action**:

1.User can press the “NO” button in step 2 and cancel the logout.

**Exceptions:**

**Related Use Cases:**

**Login**

**Decision Support:**

* **Frequency:** Will be used on average 2 times the software is used.
* **Criticality:** High. Core functionality of software.
* **Risk:** Low. Fundamental functionality.

**Constraints:**

**Modification History:**

* **Owner: Steve Noel**
* **Initiation Date:** February 26, 2015
* **Last Modified:** February 26, 2015

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Use Case ID:** S-WALL/2-003/Connect to Chrome cast

**Use Case Level:** High Level

**Details:**

* **Actor:** User
* **Preconditions:**

1. User must have opened the application on their mobile device.

2. User must be logged into the system.

3. There is a Chrome cast device on the same network as the mobile device.

* **Description:**

1. The use case begins when the user presses the “Connect to Chrome cast“ button.

2. The system then shows a list of Chrome cast devices on the network.

3. The user then selects a device from the list of devices.

4. The system then hides the list view and establishes a connection with the selected device.

5. The use case ends when the system launches the custom receiver application on the Chrome cast device and establishes a web App Session with the running application.

**Post conditions:**

1. The user is connected to the Chrome cast device.

2. The user has an active web App Session with the custom receiver application.

3. The custom receiver application is running on the Chrome cast device.

**Alternative Courses of Action**:

1. The user can select the “Cancel” button on step 2 to not connect to a device.

**Exceptions:**

**Related Use Cases:**

**Disconnect from Chrome cast**

**Decision Support:**

* **Frequency:** Will be used every time the software is used.
* **Criticality:** High. Core functionality of software.
* **Risk:** Low. Dependent on external entities.

**Constraints:**

**Modification History:**

* **Owner: Steve Noel**
* **Initiation Date:** February 5, 2015
* **Last Modified:** February 19, 2015

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Use Case ID:** S-WALL/2-004/Disconnect from Chrome cast

**Use Case Level:** High Level

**Details:**

* **Actor:** User
* **Preconditions:**

1. User must have opened the application on their mobile device.

2. User must be logged into the system.

3. User must be connected to a Chrome cast device.

**Description:**

1. The use case begins when the user presses the “Connect to Chrome cast“ button.

2. The system then displays the disconnect Chrome cast view.

3. The user then presses the “Disconnect” button.

4. The use case ends when the system hides the disconnect Chrome cast view, disconnects from the Chrome cast device and sets some state variables.

**Post conditions:**

1. The user is disconnected from the Chrome cast device.

**Alternative Courses of Action**:

1. The user can press outside the view at step 2 to cancel the use case.

**Exceptions:**

**Related Use Cases:**

**Decision Support:**

* **Frequency:** Will be used on average 3 times the software is used.
* **Criticality:** High. Core functionality of software.
* **Risk:** Low. Dependent on external entities.

**Constraints:**

**Modification History:**

* **Owner: Steve Noel**
* **Initiation Date:** March 17, 2015
* **Last Modified:** April 8, 2015

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Use Case ID:** S-WALL/2-005/Select Campaign

**Use Case Level:** High Level

**Details:**

* **Actor:** User
* **Preconditions:**

1. User must have opened the application on their mobile device.

2. User must be logged into the system.

3. User must be connected to a Chrome cast device and have an active web app Session.

* **Description:**

1. The use case begins when the user selects a campaign from the campaign list.

2. The system then sets the campaign as the active campaign.

3. The system then sends a JSON message to the custom receiver application.

4. The system then receives the message and parses the message.

5. The system identifies the message and fetches the necessary data.

6. The use case ends when the system creates and displays a image slideshow object with the fetched data.

**Post conditions:**

1. The system stores the selected Campaign.

2. The system displays a image slideshow based on the campaign selected.

**Alternative Courses of Action**: N/A

**Exceptions:**

**Related Use Cases:**

**Decision Support:**

* **Frequency:** Will be used every time the software is used.
* **Criticality:** High. Core functionality of software.
* **Risk:** Low. Dependent on external entities.

**Constraints:**

**Modification History:**

* **Owner: Steve Noel**
* **Initiation Date:** March 20, 2015
* **Last Modified:** March 20, 2015

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Use Case ID:** S-WALL/2-006/Switch Campaign

**Use Case Level:** High Level

**Details:**

* **Actor:** User
* **Preconditions:**

1. User must have opened the application on their mobile device.

2. User must be logged into the system.

3. User must be connected to a Chrome cast device and have an active web app Session.

* **Description:**

1. The use case begins when the user selects a campaign from the campaign list.

2. The system then sets the campaign as the active campaign.

3. The system then sends a JSON message to the custom receiver application.

4. The system then receives the message and parses the message.

5. The system identifies the message and fetches the necessary data.

6. The use case ends when the system modifies the image slideshow object with the new fetched data.

**Post conditions:**

1. The system stores the selected Campaign.

2. The system displays a image slideshow based on the campaign selected.

**Alternative Courses of Action**: N/A

**Exceptions:**

**Related Use Cases:**

**Decision Support:**

* **Frequency:** Will be used every time the software is used.
* **Criticality:** High. Core functionality of software.
* **Risk:** Low. Dependent on external entities.

**Constraints:**

**Modification History:**

* **Owner: Steve Noel**
* **Initiation Date:** March 22, 2015
* **Last Modified:** March 22, 2015

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Use Case ID:** S-WALL/2-007/ Play Pause Slideshow

**Use Case Level:** High Level

**Details:**

* **Actor:** User
* **Preconditions:**

1. User must have opened the application on their mobile device.

2. User must be logged into the system.

3. User must be connected to a Chrome cast device and have an active web app Session.

* **Description:**

1. The use case begins when the user selects a campaign from the campaign list.

2. The system then sets the campaign as the active campaign.

3. The system then sends a JSON message to the custom receiver application.

4. The system then receives the message and parses the message.

5. The system then identifies the message.

6. The use case ends when the Image Slideshow is either paused or resumes depending on the message sent.

**Post conditions:**

1. The system Image slideshow on the Chrome cast receiver application is either paused or resumed.

**Alternative Courses of Action**: N/A

**Exceptions:**

**Related Use Cases:**

**Decision Support:**

* **Frequency:** Will be used every time the software is used.
* **Criticality:** High. Core functionality of software.
* **Risk:** Low. Dependent on external entities.

**Constraints:**

**Modification History:**

* **Owner: Steve Noel**
* **Initiation Date:** March 24, 2015
* **Last Modified:** March 24, 2015

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Use Case ID:** S-WALL/2-008/Rejoin Web App

**Use Case Level:** High Level

**Details:**

* **Actor:** User
* **Preconditions:**

1. User must have opened the application on their mobile device.

2. User must have logged into the system.

3. System must be connected to a Chrome cast device.

4. Custom receiver app must be running on the Chrome cast device.

* **Description:**

1. The use case begins when the user presses the “Join Web App” button.

2. The system shall send the custom Chrome cast application a rejoin signal.

3. The use case ends when the system establishes a web session with the Chrome cast receiver application.

**Post conditions:**

1. The user has an active web app session with the receiver application.

**Alternative Courses of Action**: N/A

**Exceptions:**

* The system is unable to establish a web app session.

**Related Use Cases:**

**Decision Support:**

* **Frequency:** Will be used on average 3 times the software is used.
* **Criticality:** High. Core functionality of software.
* **Risk:** Low. Dependent upon external entities.

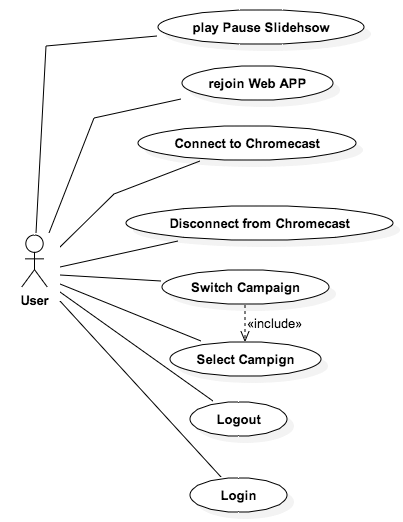
**Constraints:**

**Modification History:**

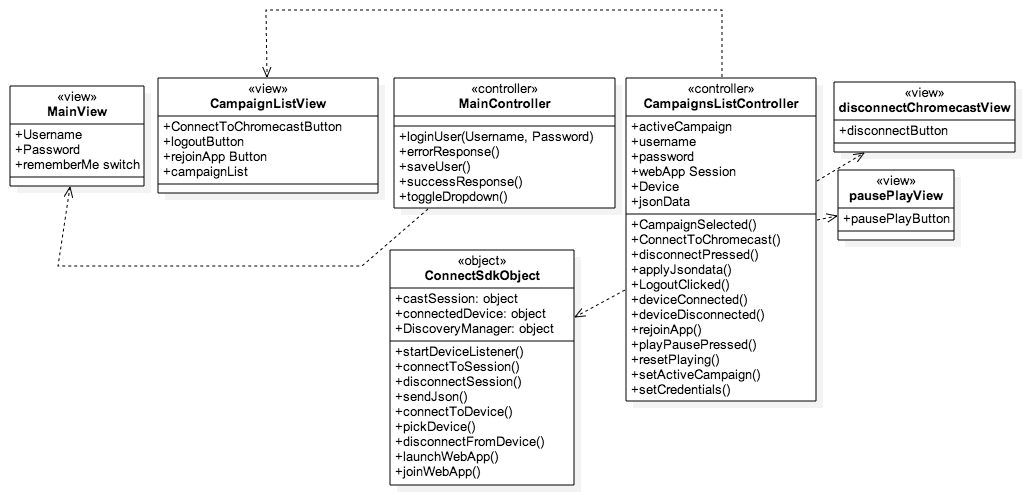
* **Owner: Steve Noel**
* **Initiation Date:** April 9, 2015
* **Last Modified:** April 9, 2015

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**6.2. Appendix B - Use case diagram using UML**

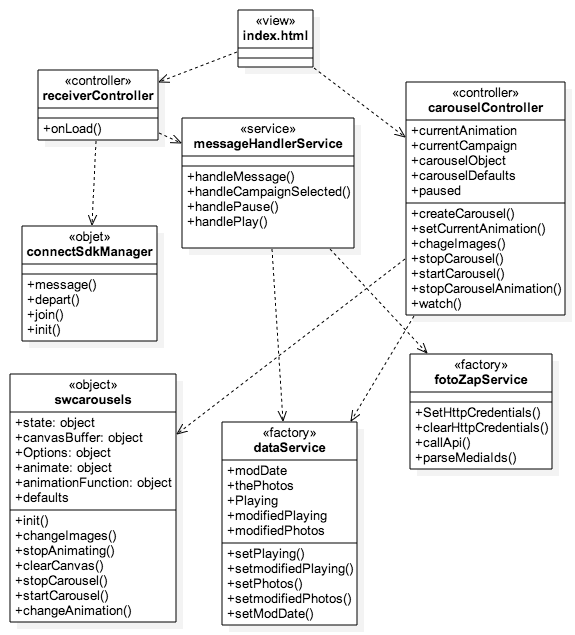


**6.3. Appendix C - Static UML Diagram**

****

**Sender Application Class Diagram**

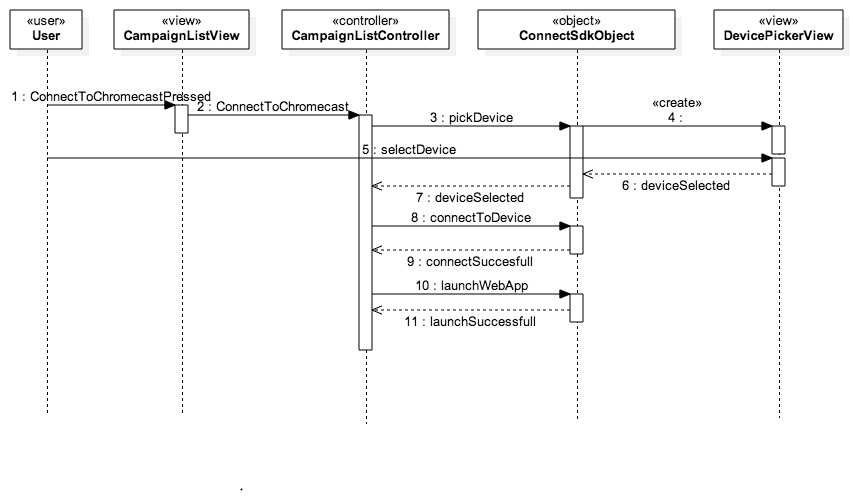
The Sender Application is made up of 7 different entities. It has two important views the Main view or login view and the Campaign List View which is the view that has the list of campaigns from the user and all other important buttons which are also views themselves. Then it has two main controllers the Main or login Controller and the Campaign List Controller. Basically the Campaign Controller handles the majority of the sender applications logic and user inputs like selecting campaign, pause and playing the slideshow and connecting to the Chrome cast device. The login controller handles the login view and the logic for logging a user in. Finally the



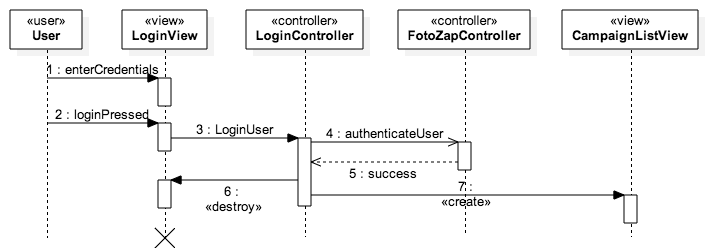
**Custom Chrome cast receiver Application Class Diagram**

The chrome cast receiver application is made up of 8 different objects. The Carousel controller which controls the carousel object,the carousel object which handles the image slideshow, the dataService which acts as storage for fetched data, the connectSdk object which receives the senders messages, the messageHandler service which handles the message and the fotoZaoService which fetches dat from photoZap Servers.

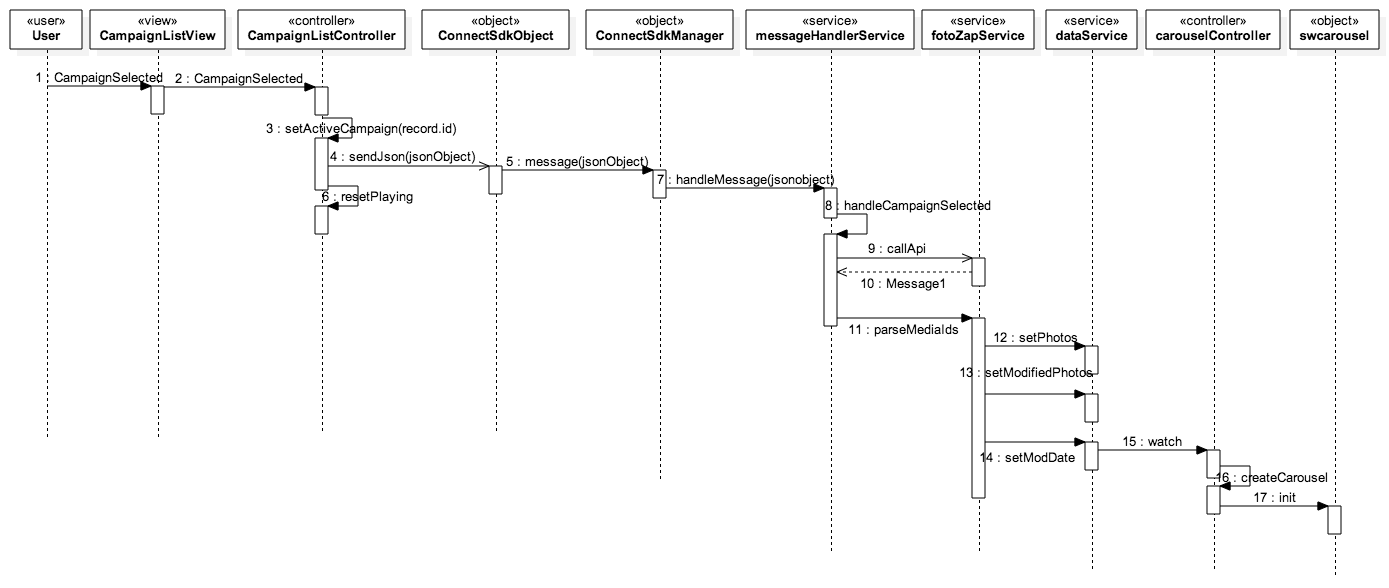
**6.2. Appendix D – Dynamic UML Diagrams**



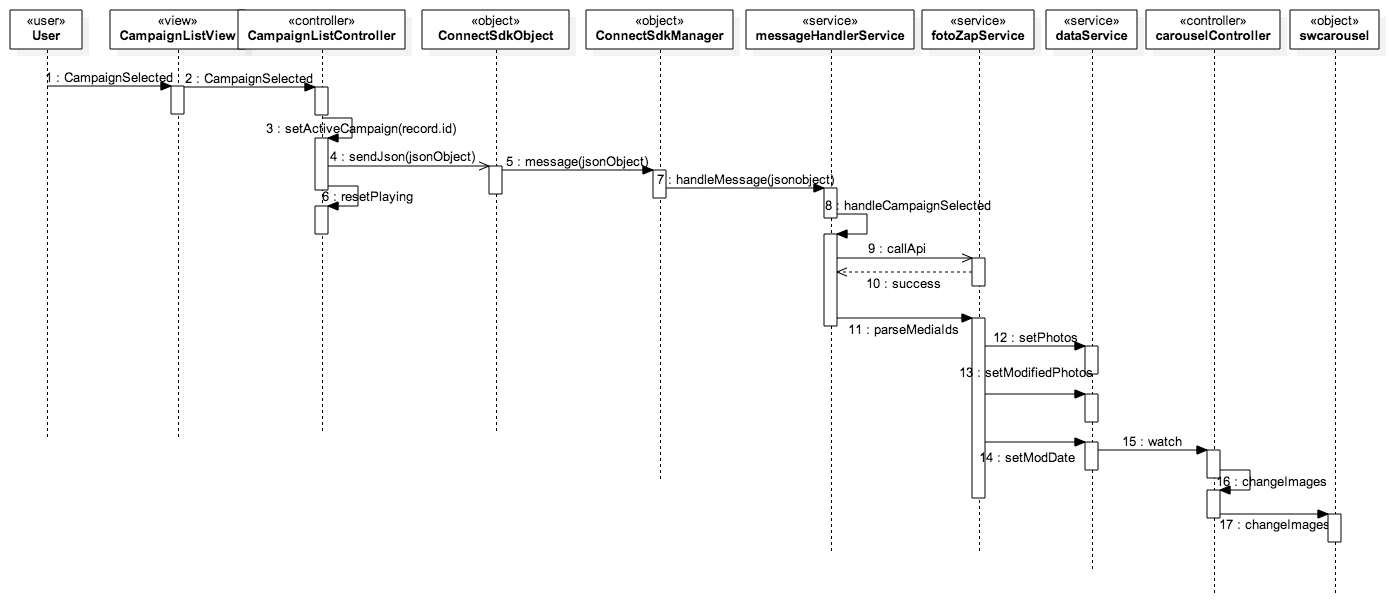
**Sequence Diagram of S-WALL/2-003/Connect to Chrome cast**



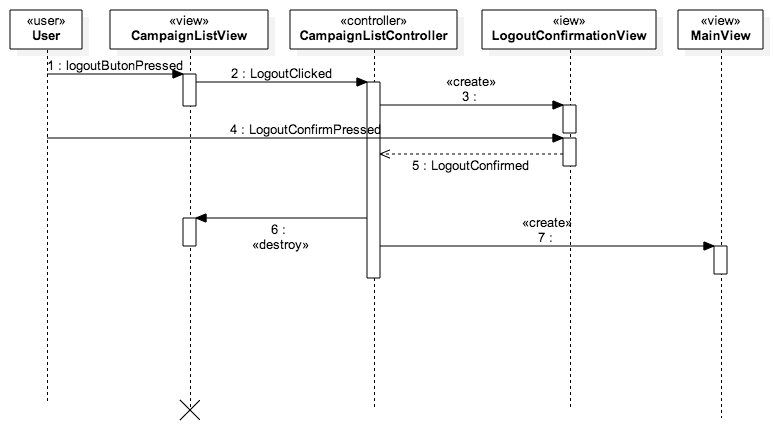
**Sequence Diagram of S-WALL/2-001/Login**



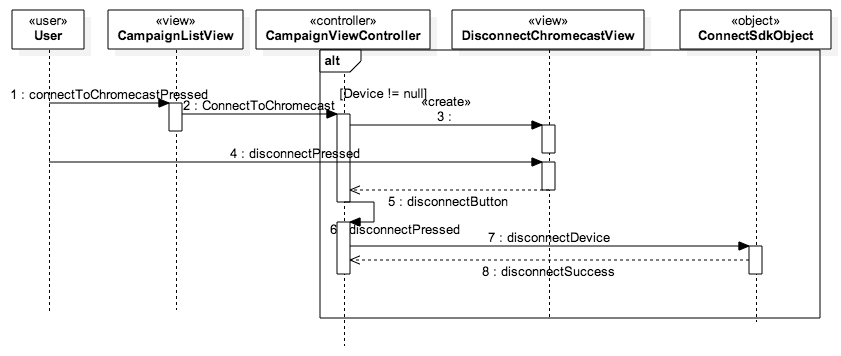
**Sequence Diagram of S-WALL/2-005/Select Campaign**



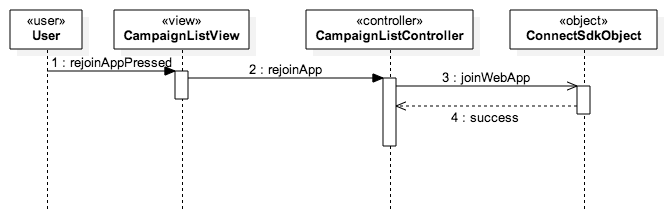
**Sequence Diagram of S-WALL/2-006/Switch Campaign**



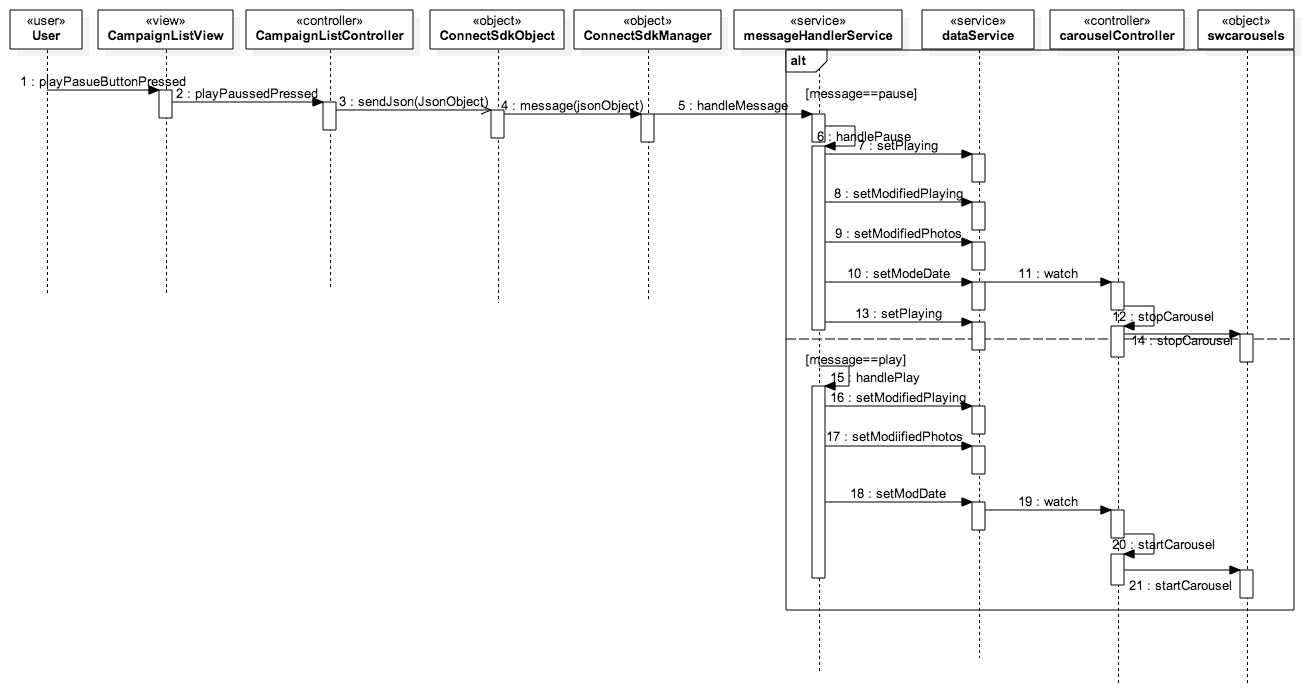
**Sequence Diagram of S-WALL/2-002/Logout**



**Sequence Diagram of S-WALL/2-004/Disconnect from Chrome cast**



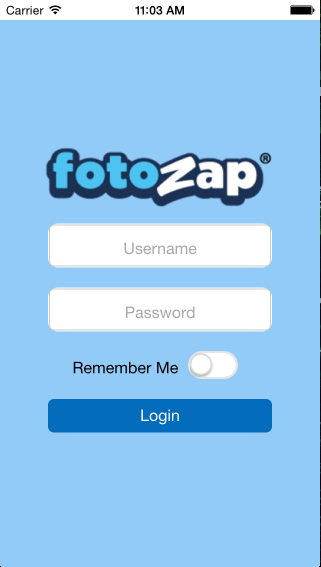
**Sequence Diagram of S-WALL/2-008/Rejoin Web App**

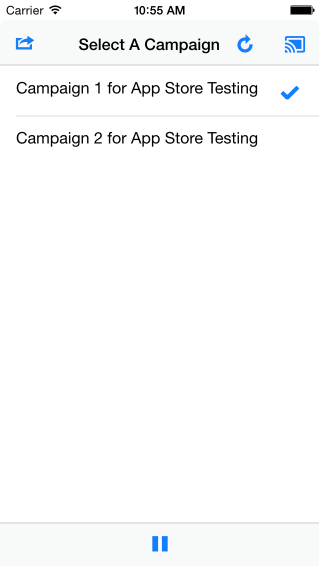


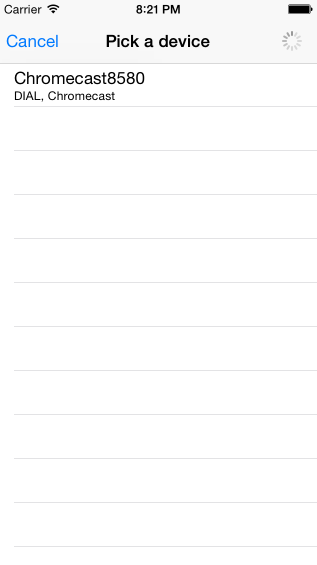
**Sequence Diagram of S-WALL/2-007/ Play Pause Slideshow**

**6.2. Appendix E – User Interface Designs**

**User Interface for Sender Application**

****

****

****

**User Interface for Chrome cast Receiver Application**



**6.6. Appendix F - Diary of meetings and tasks.**

Diary Entry 1:

Date: February 8, 2015

Location: Picture Marketing Offices

Start time: 12:00 pm

End time: 1:30 pm

In Attendance: Carlos Ocampo , Louis Zuckerman

Late: N/A

Agenda:

* Introduction and initial meeting
* Learn about Picture Marketing.
* Explanation and details of the project
* Go over requirements of the system.
* Talk about expectations of the project.

Summary of Discussion:

Introduced to the Picture Marketing Developers and Management. Discussed the Social Wall project, what is needed, what tools to use and what is expected.

Diary Entry 2:

Date: February 13, 2015

Location: Picture Marketing Office

Start time: 12:00 pm

End time: 1:30 pm

In Attendance: Carlos Ocampo , Louis Zuckerman

Late: N/A

Agenda:

* Talk about implementation so far.
* Discussed use cases.
* Discussed requirements and constraints

Summary of Discussion:

Discussed the implemented user stories thus far and the issues that have been discovered during development.

Diary Entry 3:

Date: February 19, 2015

Location: Picture Marketing Offices

Start time: 12:30 pm

End time: 1:30 pm

In Attendance: Carlos Ocampo, Louis Zuckerman

Late: N/A

Agenda:

* Discuss Login Bug.
* Demo of current system.
* Discuss application control flow and current

Summary of Discussion:

A demo of the current system was shown. Discussed how to fix the login use case problem. Discussed the performance of the slideshow engine. Decided to change the slideshow to a canvas implementation.

Diary Entry 4:

Date: February 27, 2015

Location: Picture Marketing Offices

Start time: 12:00 pm

End time: 1:30 pm

In Attendance: Carlos Ocampo , Louis Zuckerman

Late: N/A

Agenda:

* Fixed Logout Use Case
* Discuss current user stories implemented.

Summary of Discussion:

Demoed the current version of the system. Discussed the implementation of the fotozapService and how the images are loaded from the server. Decided the development of the image slideshow most critical user story.

Diary Entry 5:

Date: March 6, 2015

Location: Picture Marketing Offices

Start time: 12:00 pm

End time: 1:00 pm

In Attendance: Carlos Ocampo , Louis Zuckerman

Late: N/A

Agenda:

* Slideshow engine

Summary of Discussion:

Discussed different canvas frameworks that can be used for the project especially the react-canvas framework for a crisp looking hardware- accelerated slideshow.

Diary Entry 6:

Date: March 20, 2015

Location: FIU ECS Computer lab

Start time: 12:00 pm

End time: 2:00 pm

In Attendance: Carlos Ocampo , Louis Zuckerman

Late: N/A

Agenda:

* Discuss progress
* Slideshow engine and Disconnect Chrome cast demo

Summary of Discussion:

Talked about the new slideshow engine and the Disconnect Chrome cast features. Demoed the current system. Talked about the efficiency of using a sliding animation on the Chrome cast device and decided to change the animation used to a hard cut animation.

Diary Entry 7:

Date: April 1, 2015

Location: Picture Marketing Offices

Start time: 12:00 pm

End time: 2:00 pm

In Attendance: Carlos Ocampo , Louis Zuckerman

Zuckerman

Late: N/A

Agenda:

* Demo current version of system
* Discuss bugs

Summary of Discussion:

Demoed the current system, added some functionalities that the system should have, discussed the aspect ratio bug and also discussed the joining the web app bug.

Diary Entry 8:

Date: April 9, 2015

Location: FIU SCS Computer Lab

Start time: 12:00 pm

End time: 1:30 pm

In Attendance: Carlos Ocampo , Louis Zuckerman

Late: N/A

Agenda:

* Discussed Refactoring Code
* Discussed future developer setup instructions
* Discussed UI fixes

Summary of Discussion:

Made sure developer setup instructions were clear so future developers can follow it. Also, made sure code was readable and self-documenting.

Diary Entry 9:

Date: April 17, 2015

Location: Picture Marketing

Start time: 12:00 pm

End time: 2:00 pm

In Attendance: Carlos Ocampo , Louis Zuckerman

Late: N/A

Agenda:

* Demo of Final Product
* Discussed submitting App to store

Summary of Discussion:

Demoed the final version of the system. Also discussed the submittal of the app store but where not able to finally submit to store. Explained the setup Instructions to Picture Marketing Developers.

**7. References**

1. <http://angularjs.org/> - Javascript MVC Framework.
2. <http://docs.sencha.com/touch/2.2.0/> - Sencha Touch MVC Framework for developing HTML mobile applications.
3. <http://connectsdk.com/>- Connects to Chrome cast device.
4. <http://cordova.apache.org/> - Porting web app to IOS
5. <http://nodejs.org/> - Porting web app to IOS
6. <https://www.npmjs.com/> - Package manager for receiver web application
7. <http://www.techsmith.com/camtasia.html/> - Screen Recorder for videos
8. <http://newegg.com/> - Prices for hardware and software requirements
9. <http://photoshop.com/> - Photoshop CC
10. <http://www.bryntum.com/docs/siesta/> - Unit and Integration testing for Sencha Applications
11. <http://jasmine.github.io/> - Unit Testing Framework for Web Applications.
12. <http://angular.github.io/protractor/#/> - Integration testing Framework for Angularjs web Applications