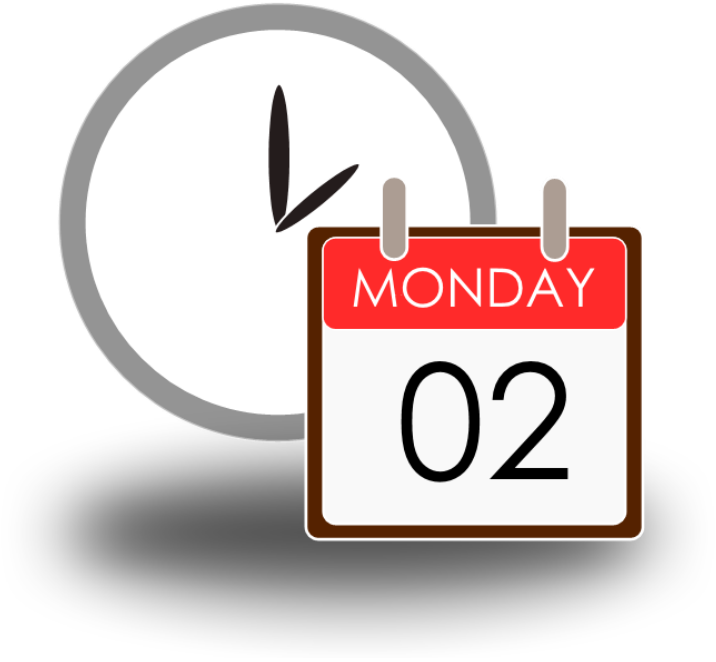
Software Requirements Specifications

CIS 454 -- Syracuse University, Spring 2015

Preston R. Cannedy, Kenton Guan, Anders Kroyer, Sebastian Karstensen



AUTOBOOK

Table of Contents

1. Preface **3**

1.1 Document Title *3*

1.2 System Type *3*

1.3 Current Version *3*

1.4 Target Audience *3*

1.5 Official Launch Date *3*

1.6 Document Description *3*

1.7 Revision History *3*

2. Introduction **4**

2.1 Purpose *4*

2.2 Overview *4*

2.3 Scope *4*

2.4 Assumptions & Dependencies *4*

3. Glossary **5**

4. User Requirements **5**

4.1 Functional User Requirements *5*

4.2 Non-Functional User Requirements *5*

5. System Architecture **6**

5.1 Application Type *6*

5.2 General System Architecture *6*

5.3 Architecture Type *6*

5.4 Application Objects *6*

5.5 Process Flow *7*

6. System Requirements **8**

6.1 Functional System Requirements *8*

6.2 Non-Functional System Requirements *10*

7. System Models **10**

7.1 Class Diagram *10*

7.2 Use Case Diagram *11*

7.3 Activity Diagram *12*

7.4 Sequence Diagram *12*

8. System Evolution **12**

1. Preface

1.1 Document Title: Software Requirements Specification for “Autobook” Mobile Application for Android

1.2 System Type: Mobile Application- Android

1.3 Current Version: Version 2.0.0

1.4 Target Audience: Android Mobile Device Users

1.5 Official Launch Date: April 18, 2015 - *Open House, Syracuse University*

1.6 Document Description:

- To understand the scope of this project
- To specify the requirements to assure high quality deliverables
- To map the software development process for implementation
- To provide thorough understanding of the application and the relationships between its parts, for maintenance

1.7 Revision History:

Version	Date	Description
1.0.0	10/19/2014	Initial SRS, awaiting approval
1.0.1	11/01/2014	Approved SRS
1.1.0	11/29/14	Grammar Changes throughout
1.2.0	12/01/14	Revisions to State Diagram
1.2.1	12/01/14	Sequence Diagram Created
1.2.2	12/05/14	Revisions to Activity Diagram
1.2.3	12/05/14	Revisions to UseCase Diagram
1.2.4	04/26/15	Revision of entire document

2. Introduction

2.1. Purpose

The AutoBook application is designed to make it easier for users to remember to send text messages and post to their social media outlets at any specific date and time, seamlessly. This application will send personalized messages to designated recipients for any and all significant events in the future, automatically.

2.2. Overview

Over time, technology has expanded to where we are no longer sending each other the same amount of real cards in the mail, or calling each other on occasions such as birthdays, graduation days, anniversaries, and similar holidays. It has become tedious to remember to send a text message or other forms of interaction on social media for every friend and family member throughout the year.

The Autobook application will allow the user to manually input special days into Autobook, and associating these days with members of their contact lists on their mobile device. Autobook will remember to create and send messages via text, Facebook, or Twitter which will be automatically sent to their contacts at any time of the user's choosing. This will allow the user to easily manage and maintain the relationships they have with friends and family members, by acknowledging significant occasions throughout the year, without having to take time out of their busy schedules to remember.

Users will be able to maintain and edit any and all interaction Autobook has with members of their contact list. Personalized messages will be able to be sent to the users contacts, or they may also select default greetings associated with popular occasions in the United States.

2.3. Scope

The Autobook project planning and system design will take place during the Fall, 2014 semester and will be implemented during Spring, 2015 semester. The development and implementation stages will be finished by the end of April, 2015.

2.4. Assumptions & Dependences

Only users who possess an Android-enabled mobile device and have access to the Google Play store will be authorized to download and run the Autobook mobile application.

3. Glossary

Event — A significant occasion which is associated with a particular, noteworthy/congratulatory date

Notification — An automated message intended to be sent to a specific person, at a particular point in time

4. User Requirements

4.1 Functional User Requirements

4.1.1 The application shall allow the user to create an event for which to send an automated message.

4.1.2 The application shall allow the user to delete an existing event.

4.1.3 The application shall allow the user to edit an existing event.

4.1.4 The application shall allow the user to create a notification to associate with an event.

4.1.5 The application shall allow the user to delete a notification associated with an event.

4.1.6 The application shall allow the user to edit a notification associated with an event.

4.1.7 The application shall allow the user to import existing contacts from the systems default contacts application.

4.2 Non - Functional User Requirements

4.2.1 The application shall be used on a device with the Android operating system installed.

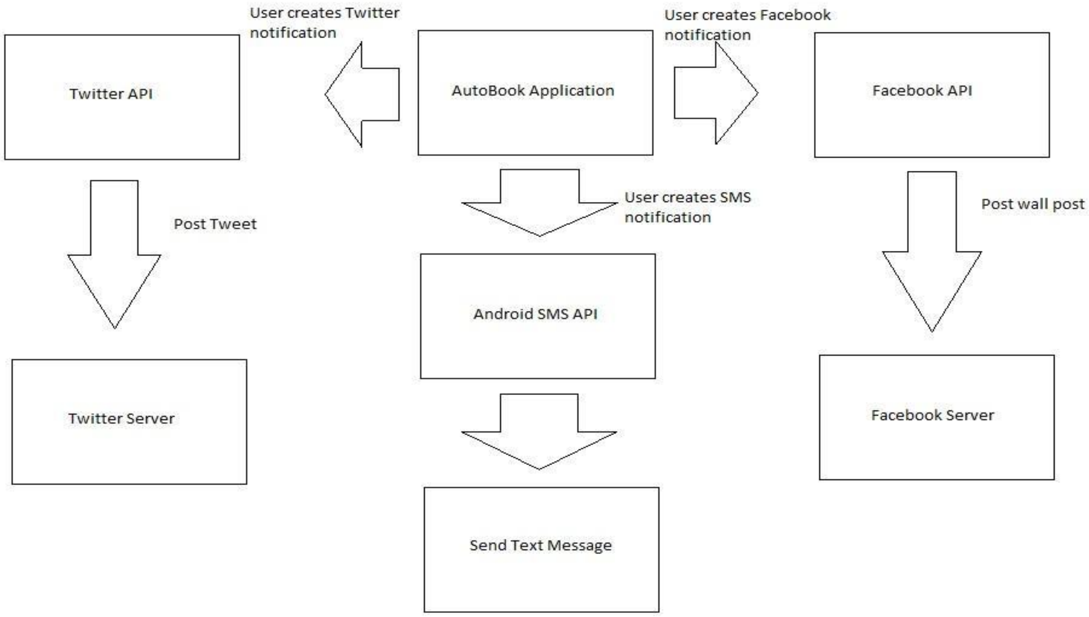
4.2.2 The application shall be able to send Twitter/Facebook notifications when the user has the appropriate account for the service required.

5. System Architecture

5.1. Application Type: System of Systems

This application is a "system of systems" type of application because it is composed of multiple systems. The different systems that we are integrating are Facebook, Twitter and SMS messaging. The application is a central hub that will allow the user to manage sending automatic messages from any of these 3 systems.

5.2. General System Architecture



5.3. Architecture Type: Client/Server

This application will be built using a client/server architecture, where the application is the client, which will connect to the outside servers using the available APIs from each respective service, in order to post messages. All of the processing/data storage will be done within the application itself, without the need of another server/database.

5.4. Application Objects

5.4.1. Event Object - This object will contain an "event", which will have a date, name and event type with it. This object is created when the user creates a new event in the application.

5.4.2. Facebook Notification Object - This object will contain a "Facebook Notification", which will have a Message and Event object associated with it. This object is created when the user creates a new Facebook notification within the application. The application will post the Message to the user's wall at the time specified in the Event object.

5.4.3. Twitter Notification Object - This object will contain a "Twitter Notification", which will have a Tweet and Event object that is associated with it. This object is created when the user creates a new Twitter Notification within the application. The application will post the Tweet associated with the object at the time specified in the Event object.

5.4.4. SMS Notification Object - This object will contain an Event and a Message that is associated with it. The object is created when the user creates a new SMS notification within the application. The application will send the Message at the time specified in the Event object associated with it.

5.5. Process Flow

The user will open the application which will allow them to create a new event and/or notification associated with it. Creating these events/notifications will create the associated objects within the application. The application will have a scheduler that will check to see if the specified date in the event has been met. If so, it will deconstruct the Notification objects associated with that event. Then the application will use the information taken from the deconstructed Notification objects and send it to the respective service using the appropriate API calls. Upon successful posting of the message(s), the user's phone will vibrate if their notification was successfully posted to the proper services.

6. System Requirements

6.1 Functional System Requirements

6.1.1 Create an Event

The user will launch the application and the application home screen will load. On the home screen, there will be a button that the user can tap one of three buttons (Facebook, Twitter, SMS) to create a new event. Upon choosing any of these options, the user will be able to create a new event for that type of notification. During the creation of the event, the user will be able to specify the date of the event, the receivers for this event, and associate a notification with the event.

6.1.2 Edit an Event

The user will launch the application and navigate to events. Tapping on any of the upcoming events will trigger editing options for that event.

6.1.3 Delete an Event

The user will launch the application and the application home screen will load, where the user can navigate to events. Pressing and holding on these events will trigger deletion options.

6.1.4 Create a Notification for an Event

On the event screen, there will be different buttons that the user can tap to create a new SMS notification, Twitter notification or a Facebook notification.

6.1.5 Edit a Notification for an Event

On the event screen, the user will be able to tap on the associated notifications to edit them.

6.1.6 Delete a Notification for an Event

On the event screen, the user will be able to tap on the associated notifications to delete them.

6.1.7 Import existing Contacts

The user will launch the application and the application home screen will load. From here, the user can manage their contacts by pressing the “Contacts”-button. Upon creation of a new user, there will be an option to import existing contact information from the devices system contact app.

6.2 Non - Functional System Requirements

6.2.1 Android Device Constraints

This application will be written in Java and require the user to have a smartphone running the Android operating system. This is because Java is not compatible with iOS, which runs objective-C and Swift. Also, since the Android operating system is less locked down than the iOS operating system, it should be easier to implement the ideas for this application within an Android application rather than an iOS application.

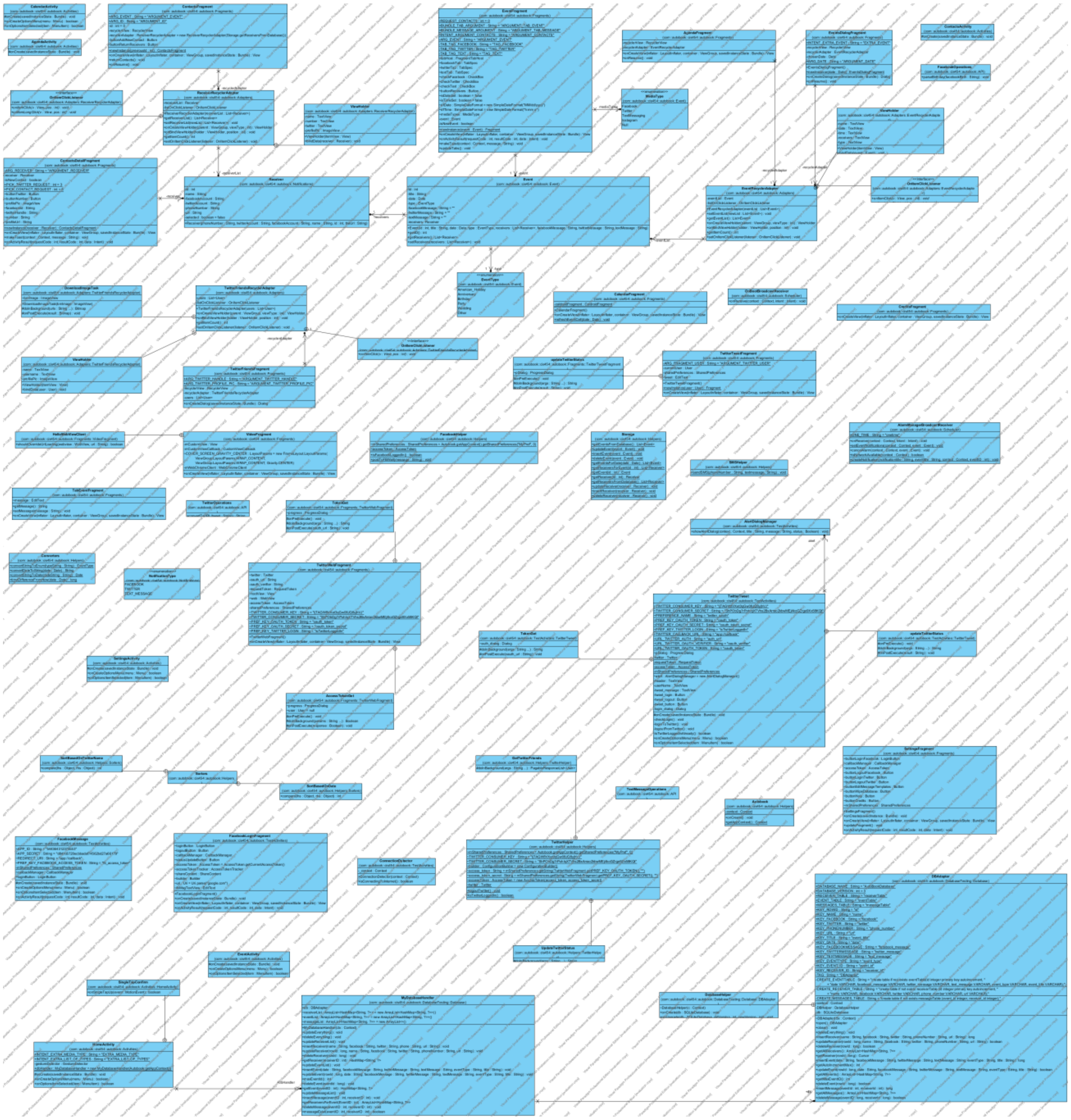
6.2.2 User has required Service accounts

The use of the Facebook and Twitter portions of the application is not possible unless the user has the required service accounts. In order to access the user’s friends list and send messages on the respective sites, the user needs to log into the services within the application and grant permission to the application to post messages and access their friends list.

7. System Models

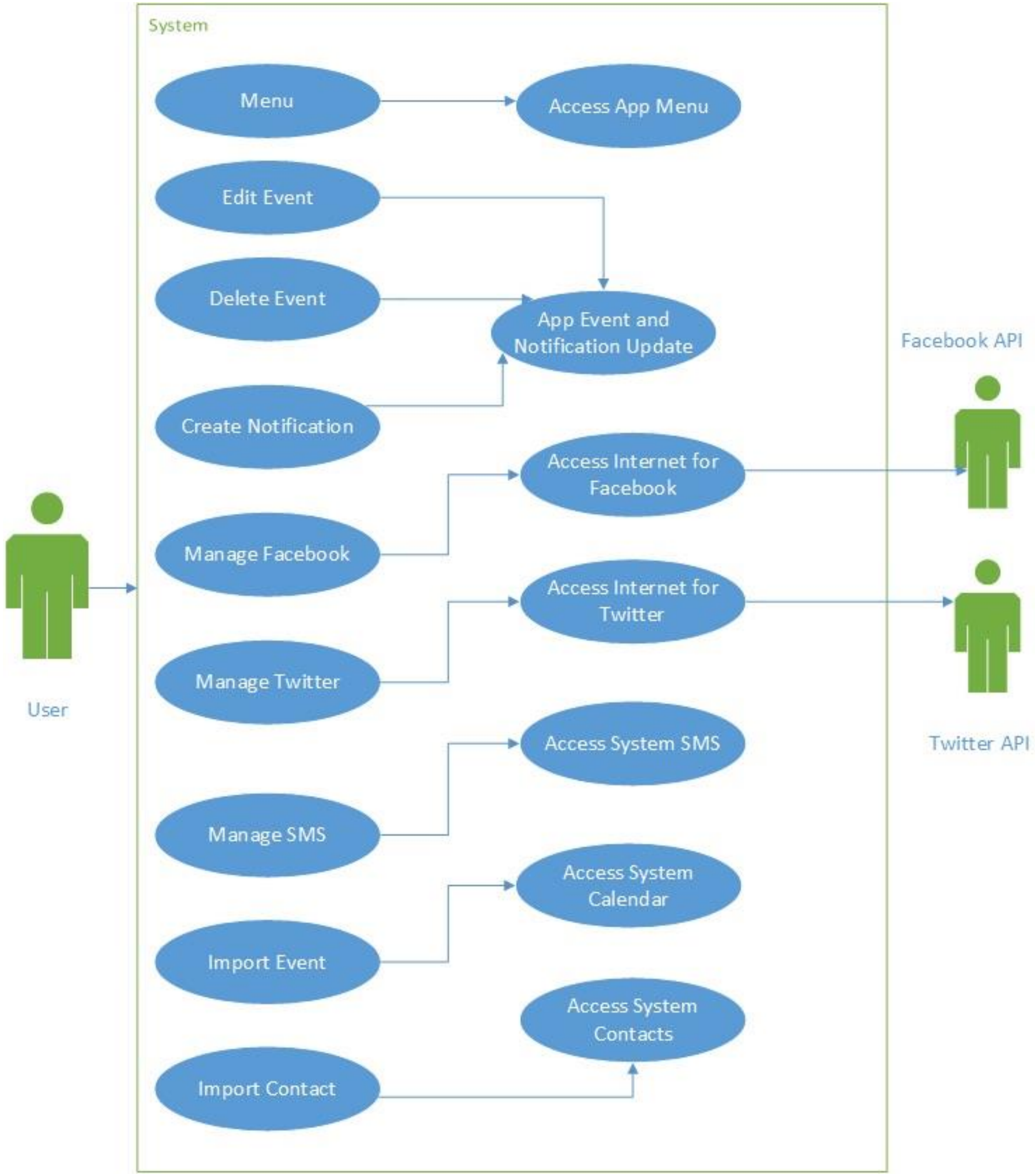
7.1. Class Diagram

The diagram below is an overview of the main classes to be used in our system.



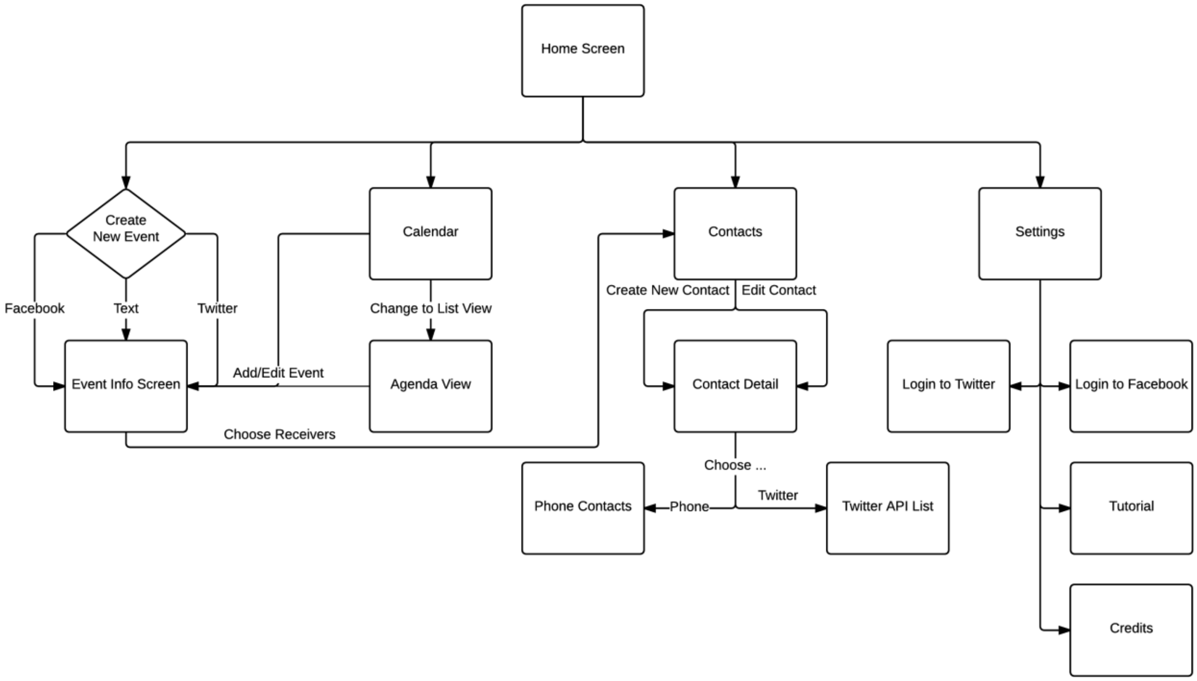
7.2. Use Case Diagram

The diagram below shows the interaction of a user with the system depending on the task to be completed.



7.3. Activity Diagram

The diagram below represents the workflow of the different steps to be taken when utilizing the application.



8. System Evolution

There are several features we would like to add to our design in the future in order to accommodate the projected growth of our product.

- Email Integration
  - Give users the option to also be able to send email notifications to anybody in their contact list at a predetermined time
- Port to IOS
  - Increase our targeted audience by creating an IOS app so that apple product user could also enjoy using our application
- Web Integration
  - Extending users' way of using our product by giving them the option of using it from their computer through a browser; this will also make our product cross-platform