Calendar

Software Requirements Specifications

Benjamin Romano

CJ Guttormsson

Bryan Anderson

2016-03-7

Calendar	Version: 1.0
Software Requirements Specifications	Date: 2016-03-7
Calendar_SRS.docx	

Revision History			
Date	**Version**	**Description**	**Author**
2016-02-15	0.1	Initial Draft	
2016-03-7	1.0	Initial	

Calendar	Version: 1.0
Software Requirements Specifications	Date: 2016-03-7
Calendar_SRS.docx	

Table of Contents

Calendar	Version: 1.0
Software Requirements Specifications	Date: 2016-03-7
Calendar_SRS.docx	

Calendar	Version: 1.0
Software Requirements Specifications	Date: 2016-03-7
Calendar_SRS.docx	

1. Introduction

Calendar is an Android application that simplifies the process of finding common times with friends and discovering new events in your area. The app will have the ability to sync with existing calendar platforms starting with Gmail. Another feature we have is the ability to add friends and view their calendars. A friend can also set which events are visible to others.

2. Glossary

Friend: Someone has given you access to their calendar.

Event Visiblity: Setting for visibilty of an event. Hidden - The event is not shown to other people. To others it will appear as if the user is free during that time period. Partial - Time blocked off by event is shown to friend, but not details. Friends - Friends can see all the details of an event. Location - People in the nearby area can see all details about an event.

Nearby Area: 20 mile radius around the **current** location of the user. This means that the nearby area is not necessarily static.

OAuth Token: A token provided from a serivce provider (in this case Google+) to allow third parties to access a user's data. With the token, we can view and change a user's calendar and events. An oauth token is recieved after the user accepts our application's request to get access to their data.

3. User Requirements Definition

3.1 Functional Requirements

1. A user is able to login using Google+ and give our app permission to view calendar.
2. A user must have a Google+ account.
3. The system will only handle one of the user's calendars.
4. A user is able to find and add friends using email address.
5. A user is able to find upcoming events.
6. To use location features, the user must have GPS enabled on their phone.
7. A user is able to create events.
8. A calendar event can be set to different visibility levels: Location, Friend, Partial, Hidden.
9. The system given a gmail oauth token can scrape calendar events.

3.2 Non-Functional Requirements

1. The system must be able to sync calendars within 10 minutes of a change.
2. The system must be able to run on devices with at least Lollipop.
3. The system must use strive to drain the battery as little as possible.

4. System Architecture

The system will be composed of five main activities, two popups, a navbar and a database. The five activities include Login, Calendar, Create/Edit Event, Common Time Finder, Friend List. The two popups are the add friend popup and Calendar Settings popup. The navbar shows the different activities and lets users switch between the activities quickly.

4.1 Login Activity

The login screen is the first screen a user sees if they have not already signed into an account. On this screen a user is presented with a google+ button that takes them through the Google+ login activity. Once completed, the user is shown to the Calendar activity, which is our main activity.

4.2 Calendar Activity

The calendar activity is the main activity for the application. On this activity, the user is able to view their calendar and see upcoming events. On the top of the activity, there will be a gear icon which will start up the Calendar Options Popup. On this popup, a user is able to customize what is being displayed in the calendar activity. In addition to the Calendar Options Popup, the calendar activity will have a floating plus button in the bottom right which will allow a user to quickly open up the Create/Edit Event activity or the Add Friend popup. If a user clicks on an event the user will be directed to the Create/Edit Event Activity with the event info filled in.

4.3 Create/Edit Event Activity

The Create/Edit Event popup will allow a user to either a create an event or edit an existing event. If a user clicked an event to get to this activity, the event's info will be filled in the form. The fields of the form will include time, location, visibility, name, color.

4.4 Common Time Finder Activity

The Common Time Finder Activity contains a form and a submit button. The form includes fields for specifying which users to add, how long the event will run and time constraints. A user can be selected from the friends list dropdown or an email can be typed manually. An error will occur if the

email supplied is not a user of Calendar. Once submit is clicked, the query will be executed and the user will be returned to the calendar activity with the relevant information being presented.

4.5 Friend List Activity

The friend list activity lists all friends of a user by their name (pulled from gmail info) or their email if no name is present. The user is able to delete a friend on this activity. The floating plus button on the bottom right will be displayed on this activity. The two options Add Friend and Add Event will be displayed.

4.6 Add friend popup

The add friend popup will display a field to enter an email address. If the email entered is associated with a user who is registered to our service, the friend will be added to a user's friend list. Otherwise an error message is displayed.

4.7 Calendar options popup

The calendar options popup will display whether or not to show upcoming events from either friends or people nearby. In the calendar options, a user can also set which calendar to display. The user could choose between showing only their own calendar or adding in other friend's calendars. Another feature of the calendar options popup is to set how to display the calendar. The user can choose between a month, week and day view.

The calendar options popup will allow users to configure the current calendar activity. The first setting is checkbox which lets a user specify whether or not to display events in the nearby area. The next is a dropdown with checkboxes at each element which lets users select which friend's calendars to show. Each element in the dropdown shows the friend's real name or email if a name is not available. Next there is a dropdown to select how to format the calendar view. The settings for this are month, day, and week.

4.8 Navbar

The navbar will be used to switch between the multiple activities. The navbar will include button to the five activities. The navbar will also display user info in the top.

4.9 Server

The server is a REST API which can be used by the app to access the database.

4.10 Database

The database will cache event information and store user info. The dataService class will interact with this database.

4.11 Hardware Requirements

1. GPS – location data will be used to recommend local events
2. Data connectivity – used to sync with gmail and fetch info from other user's calendars.

Calendar	Version: 1.0
Software Requirements Specifications	Date: 2016-03-7
Calendar_SRS.docx	

5. System Requirements Specification

The project will involve several activities, and these activities will be implemented by a number of classes. These classes include User, Calendar, Event, CommonTimeOptions, and DataService.

5.1 User Class

The User class contains information on an individual user, including their personal information and OAuth token. The user also has a list of friends (other instances of the User class) and a calendar (an instance of the Calendar class).

5.2 Calendar Class

The Calendar class represents the calendar that belongs to some user (an instance of the User class). The calendar contains a list of events (instances of the Event class). It also contains a list of calendar options, including which users the calendar is visible to, which view to display (month, week, day), and whether or not to show other users who are available during the times being displayed.

5.3 Event Class

The Event class represents an event that users may place on their calendars. It contains a beginning and end time as well as identifying information (a name, a location, an image, and a description). It also contains a list of calendars (instances of the Calendar class) that have chosen to list the event, along with that calendar's chosen privacy level (whether its participation is visible to all, visible only to friends, visible only as an unmarked block of busy time, or visible to no one).

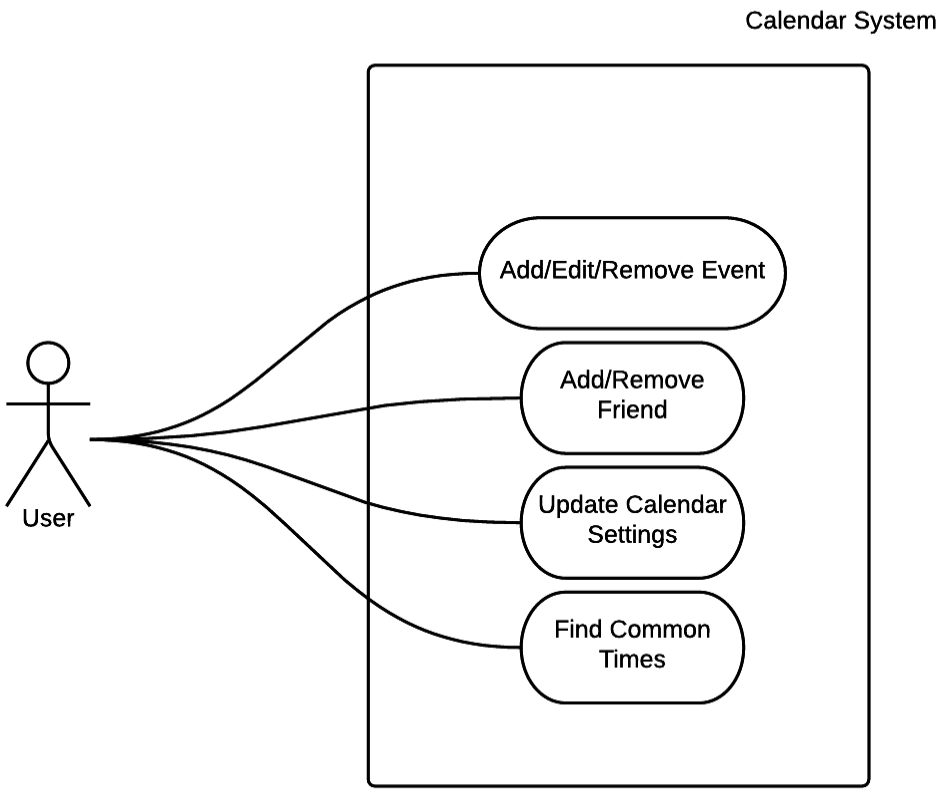
5.4 CommonTimeOptions Class

The CommonTimeOptions class contains a list of options to search for a common time to schedule an event. It includes a list of calendars (instances of the Calendar class) to query for available times, as well as a range of beginning and end times to consider. There may be multiple time ranges based on the user's preference (e.g. "either 7-10 PM on the first Tuesday of the month or 6-9 PM on the third or fourth Wednesday").

5.5 DataService Class

The DataService class is called by the other classes to interact with a remote database. It takes in a query about a user, calendar, or event and returns the requested information frfom the database.

6. System Models



Calendar	Version: 1.0
Software Requirements Specifications	Date: 2016-03-7
Calendar_SRS.docx