

OPPORTUNITY

SUPPLIER DELIVERABLE 1



Jake Runzer	Management plan
	UI Walkthrough
Zev Isert	Minimal System
	Proof
Claire Champernowne	Executive Summary
	Editing
Dylan Golden	Functional Specifications

Glossary

Term	Definition
REST	Representational State Transfer, an architecture style for designing networked applications, that relies on a stateless, client-server, cacheable communications protocol.
API	Application Program Interface, specifies how software components should interact.
Opp	Event created by the Opportunity application referring to the meeting of all preconditions for a given Opportunity notification.
SSD	Solid State Drive, a persistent data storage device using integrated circuits and designed without moving parts.
UI	User Interface, the set of design choices for an information device of which a human being may interact with.
iOS	iPhone Operating System, a popular mobile operating system designed by Apple Inc.

Executive Summary

This report describes the mobile application, Opportunity. Specifically, the interface, use interaction, functional specifications, and plans for implementation and management will be covered.

Opportunity can be used for various functions. Users will utilize the app to help them:

- Remember or plan activities that they want/need to do.
- Fill stretches of free time with appropriate activities
- View and select appropriate local events

Users interact with Opportunity in a variety of ways. In Opportunity, a user can:

- create an account using email, Twitter, or Google
- create custom Opps
- Delete or disable previously created Opps

Though the user will only be interacting with the Opportunity app, the system as a whole is comprised of several parts:

- IOS mobile application
 - designed using Sketch
 - protoyped using Invision
- backend server
- database
 - SQLite
- website

To be considered minimally acceptable, the following triggers must be available in Opportunity:

- Time and date
 - user can specify time in hours, or use generalizations such as sunrise or sunset
- Weather
 - ability to query a weather API
 - restricted to current weather at present location of device
- Availability
 - ability to query user's calendar
- Location
 - in minimal implementation, only addresses or pin based location radiuses will be provided

Opportunity aims to be a flexible service capable of improving the user's productivity. S1 summarizes its implementation and management, as well as detailing user interaction and interface.

Functional Specifications

Opportunity will be used by people who have difficulty remembering or planning activities that they want or need to do. It allows users who have identify stretches of free time and fill it with appropriate activities. Opportunity will be able to show local events to the user letting them decide what they want to do with their time. Users can do all the things they want to do when the timing is perfect, it also allows spontaneously meet up and do activities with their friends.

For the minimal implementation, the phone will be used as hardware and data will be stored on the phone itself. We will be using the built in GPS, accurate to within 10 meters of location. The application is meant to be dependable, so that business and power users can rely on it. For the minimal implementation security is not a high requirement since everything is stored on the phone and none of the information is given out or stored on a server. Opps in the minimal implementation will include conditions based on time/date, calendar, specific locations and weather based settings. For all implementations minimal battery usage will be a top priority, as the application will be running in the background all the time. Depending on the Opps set up, the application will determine what data to collect.

The goal implementation for Opportunity involves a mobile application developed for iOS. The server and database will be deployed onto a Digitalocean Ubuntu server 14.04 virtual machine which has 1GB memory, 1 CPU core, and 30GB SSD storage. The server can dynamically scale to accommodate a large flux in demand. The response time to the server should be less than 2 seconds. For the goal implementation, security will be a requirement since the application will have users sharing data and their locations (if the user wants to share that data) as well as the information on the servers will have to be secure. For the stretch goals contacts (users meeting up to do things), hours of stores and general locations will be added to the Opps settings.

The stretch goal will have the application using stores' inventories, sales, nearby events, and ski resort snow levels being added as conditions for the Opps.

Contextually and Situationally Aware

Opportunity is aware of its surroundings and its location in the world. It uses GPS along with data to determine if an Opp's parameters are met or not. It also monitors the weather; observing the conditions. The application also is aware of the locations of other users and Opps that they choose to share. Opportunity will also be aware of local events, store inventories, and ski resort snow levels.

User Interaction

The following tables provide some sample use cases that a typical user will likely experience while using the app.

Use Case 1 | New user wants to create a new account after downloading Opportunity

- 1 User opens Opportunity, and selects 'sign up'
- 2A User chooses to sign up with email and enters their name, email, and desired password
- 2B User chooses to sign up with their Google account and selects 'Google'
- 2C User chooses to sign up with their Twitter account and selects 'Twitter'
- 3A After entering their information, user selects 'create' to make their new Opportunity account

Use Case 2 | User wants to be reminded when it's sunny

- 1 User opens Opportunity, and selects (+) allowing them to create a new Opp
- 2 Names their new Opp "Sunny"
- 3 Selects 'weather' from the addable conditions
- 4 Selects 'sun' from various weather options
- 5A It is raining, and the Opp is not triggered
- 5B It is not raining, and the Opp is triggered

Use Case 3 | Opp is triggered

- 1 Opp preconditions met, user alerted
- 2A user accepted Opp, is dismissed
- 2B user snoozed Opp

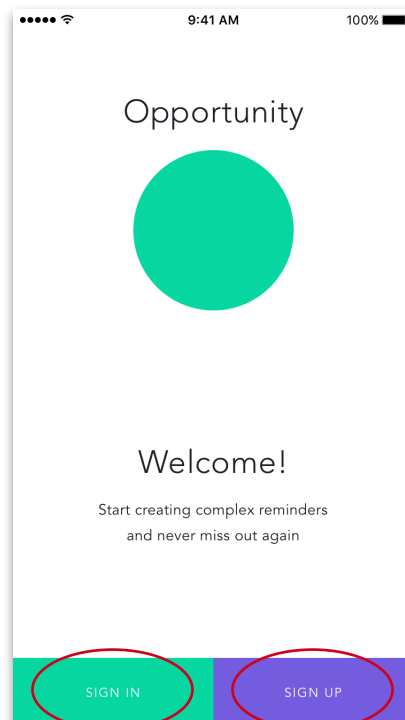
Use Case 4 | User wants to delete an Opp they have previously created

- 1 User selects the Opp they want to delete
- 2A User chooses to select 'disable', allowing them to reactivate the Opp at a later time
- 2B User chooses to select 'delete', destroying the Opp

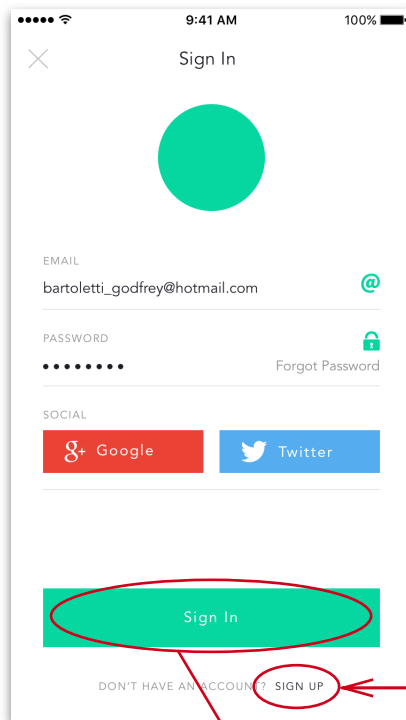
Opportunity

app walkthrough

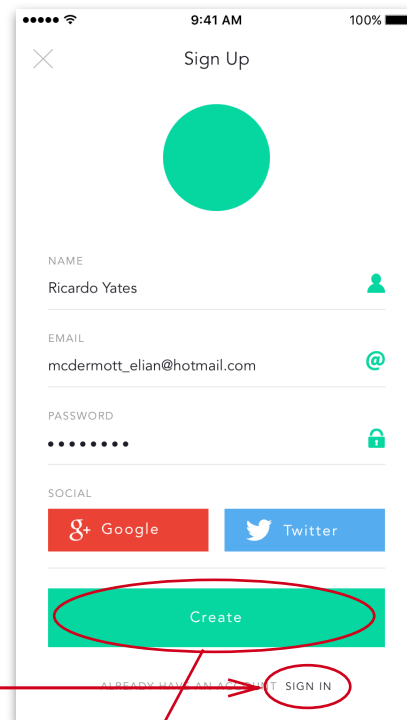
The first page to load when the app is installed or the user is not logged in



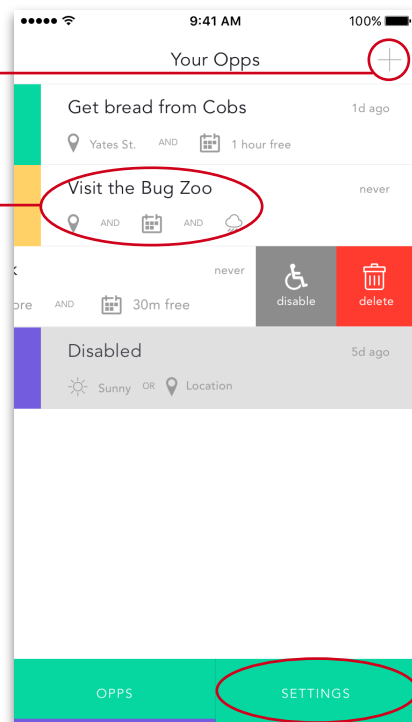
The user can sign in with email and password or Google or Twitter



The user can sign up with email and password or with other social platforms

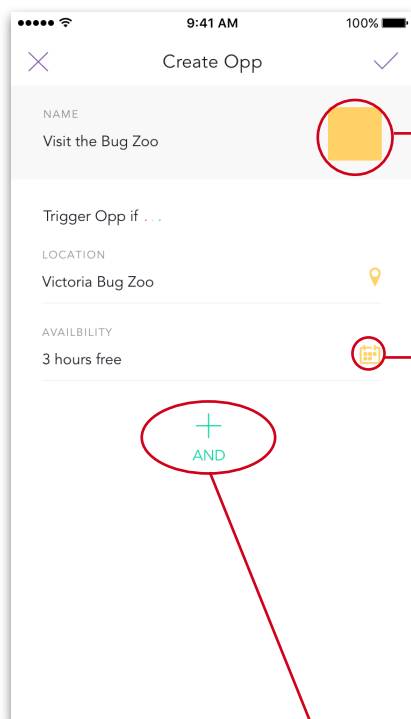


The first page to load when the user is logged in. A list of all the users Opps are shown here

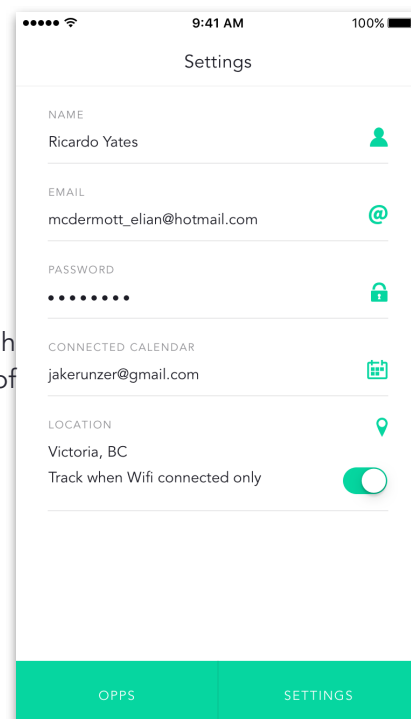


A disabled Opp will not be triggered

The edit and create Opp page. Here the user can add conditions to the reminder which will determine when the Opp is triggered

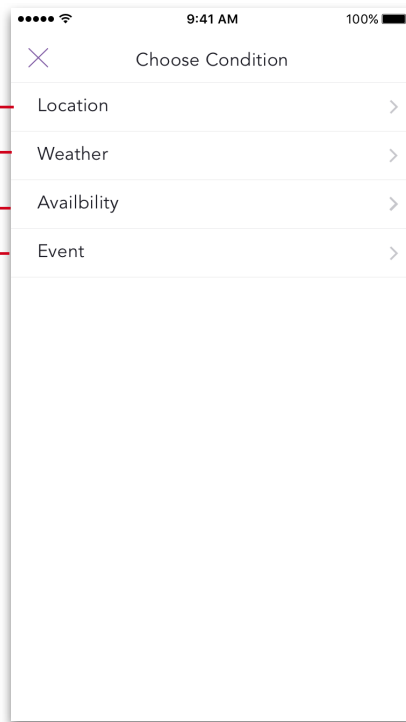


The user can change the colour of the Opp. This colour will follow through the creation of the Opp changing the icon colours

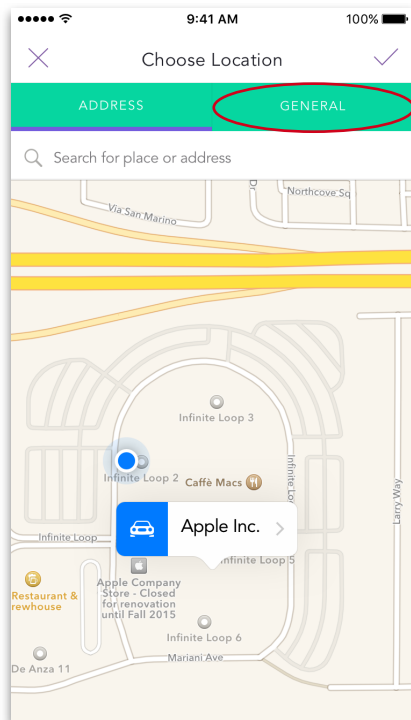


The settings page where the user can

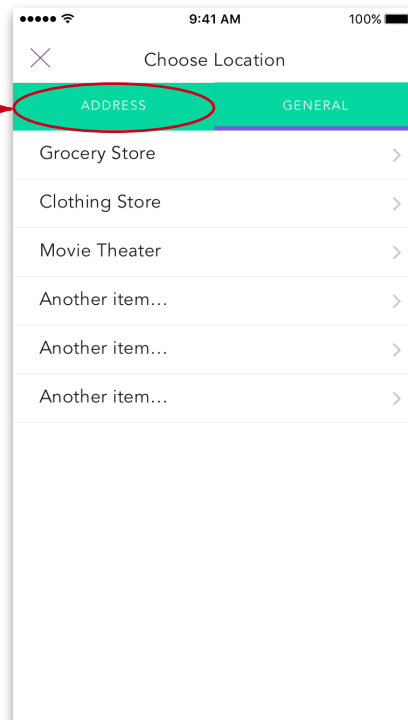
- edit account information
- connect a calendar
- change location
- set to track while connected to Wifi only



Here the user can choose what condition to add to the Opp



The user can enter an address. The Opp will be triggered when the user is near this location

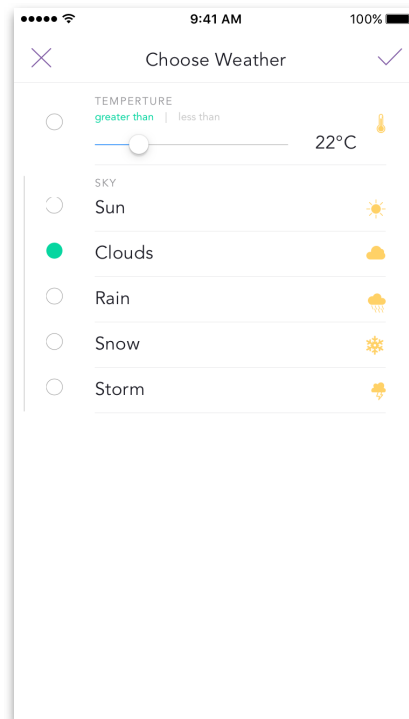
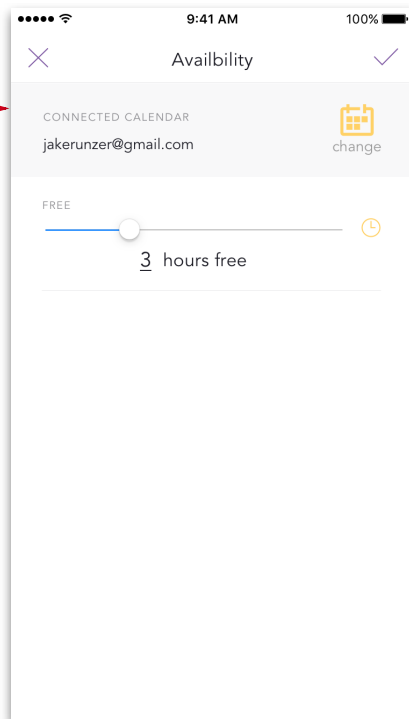


Here the user can choose a general location to trigger the app

Here the user selects how much free time on their calendar will trigger an Opp

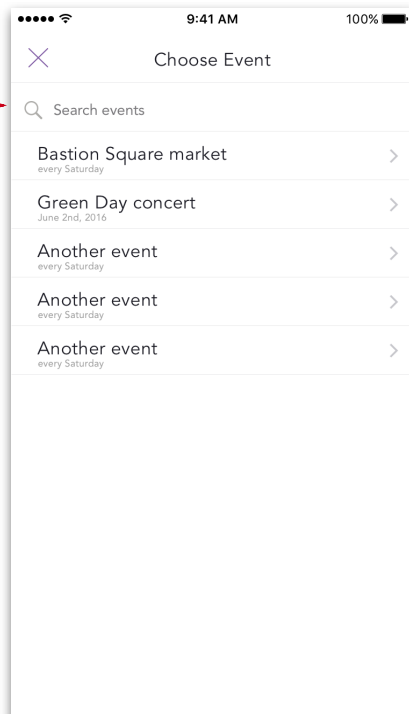
The connected calendar can be setup or changed here aswell

The user can search for and select events that will trigger the Opp



The user can set to trigger an alarm based on temperature (greater or less than) or by the status of the sky (sunny, cloudy, etc.)

The sky section is a radio group, where only 1 option can be selected



Management Plan

There are four main components required for the goal implementation of the Opportunity system. These components are:

- mobile application
- backend server
- database
- website

The user will operate Opportunity through the mobile application. The app is where the user will create and login to an account, create Opps, and be notified of triggered Opps. The application will contact the server through a REST API to authenticate users and create and delete Opps. The server will store user and Opp information in a database. Details on these components are detailed below.

Mobile Application

The application will be designed using Sketch, an interface design tool, and prototyped using Invision, a prototyping platform. Using these tools before actually coding the app will allow us to quickly iterate between designs to efficiently decide on the design with the best user experience.

The mobile application will be developed for the IOS platform. The currently designed app interface can be seen in the UI walkthrough. An Android application may also be made if time permits. The IOS app will be developed using the Swift programming language on the XCode development IDE. The interface will be created using XCode storyboards and coded using view controllers for each view. User info and Opps will be stored locally on the device using a core data database. The app will use background app refresh to wake the app up at specific times, to check the weather, determine the location, and check availability on the user's calendar. Each time background app refresh wakes the app the device's battery will be drained. Therefore, it is important to only wake the app when absolutely necessary. The app will communicate with the server via a REST API. Using GET and POST requests to specific endpoints, the app will upload user and Opp data to the server.

User Permissions

The application will require the following permissions to be accepted by the user.

Location services

In order to provide the user with location based Opportunity notifications, the application should be able to access the device location in the background.

System notifications

Notifications are the fundamental service used to communicate the availability of an Opportunity to the user when the Opportunity app is not in the foreground.

Opps

Opps are created and triggered using the mobile application. This section will detail how each condition in the Opp will be technically implemented.

Many of the conditions require waking the app in the background periodically to check data. These checks will be squished into one check for all Opps to save device battery.

Location

There are two components to location in Opportunity. The first is a specific address the user wishes to be notified at. When the user has an Opp with a location condition, the app will use the device functions to be woken when the device approaches a specific location. Device battery will drain when the location is checked with GPS, so the location checks for all Opps that use location will be combined into one.

The second component of location is to trigger an Opp when the user is near a general location, such as grocery store, clothing store, or gas station. This is not a minimum requirement as a more complex use of Google or an external API is needed to find a list and location of stores and businesses.

Weather

We are using an external API to query for weather data for the user's location. The API we are using is the OpenWeatherMap API (<http://openweathermap.org/api>). The free plan allows for 60 request per second and 50 000 requests per day. This is within our expected demand. When the user creates an Opp that includes a location condition, the app will periodically check the weather at the user's location to see if they meet the Opps conditions.

Availability

In the app the user will be app to connect a Google calendar. Authentication with Google will be done using secure OAuth2 using pre-existing libraries. The app will refresh the calendar once every hour to pull in new events. Within the app, the calendar events will be analyzed against the user's Opps to determine if

Events

We will be using the Eventful API (<https://api.eventful.com/>) to get event information. This API will provide the app with information about concerts, festivals, sports, etc. The user will be able to search for upcoming events and the application will alert the user near the time of the event.

Server

The server will be implemented if time permits. It is not an essential part of the system as the mobile application can function on its own. The server will be implemented using the Python programming language using either the Django or Flask framework. The server should be lightweight as it solely serves a REST API.

Authentication

Server authentication is one of our stretch goals. To implement this a backend server is required to make API calls to. The user will either create an account with social platform Google or Twitter, or with email and password. The server will generate the client with a token when valid credentials are provided. This token will be used in the header of all requests that change user Opps or account information. Passwords on the server will be encrypted with bcrypt, an irreversible encryption algorithm.

Opp Sync

The data for authenticated users will be stored on the server in a database. This data will be synced to the user's device to make sure they have the most up to date data. Storing user data on the server will allow multiple devices to be used for the same account. Also, if the user's device is lost or broken, they will easily be able to restore all data to a new device.

Database

The database will be SQLite. SQLite is a lightweight relational database that will allow us to quickly setup and configure it. We are not expecting to be storing large amounts of data so a more advanced database (PostgreSQL, MySQL, etc...) is not needed. The database will store user and Opp information which will be served by the server. The database will be hosted on the same machine as the backend server.

Website

The website is a small component of the system. It will describe what the app is and have a link to the app store where you can download the app. There will be screenshots of the interface and short descriptions of all the functions. This will be the main place to market the Opportunity system.

Minimal System by EOT

Should time restraints affect the progress of the management plan outlined above, a minimal implementation is forecasted herein for delivery by the end of the development term. FixCode assures that a minimally viable system composed of a self-supported application - that is, without dependence on any internal system servers - is produced by the development deadline. Such a system will consist of the ability to create Opps with triggers dependent on weather, user availability, and user location. As mentioned, the minimal implementation does not require the use of any servers internal to Opportunity to provide the described functionality. Furthermore, a minimally complete system will have a smaller set of accessible states within the application user interface.

Short list for triggered Opps

Below are trigger types considered as required in a minimally complete system.

Time and Date

All Opps should have the ability to be triggered at a certain time. The user may specify the time in hours, or by general times, such as sunrise or sunset.

Weather

Any Opp should have the ability to query a weather API and react to the resulting data as one of the specified triggers. To refrain from over-extending the minimally viable system, weather based Opp triggers are restricted to the current weather at the present location of the device. The user should be able to specify triggers for weather events including active precipitation, and current temperature above or below a set value.

Availability

All Opps should have the ability to query the user's calendar. Minimal state triggers are if the triggered time doesn't conflict with an existing calendar event, or when a calendar event begins or ends.

Location

Any Opp should also be able to use device location as a trigger. In the minimal implementation only address or pin based locations will be provided, and will trigger the Opp when the device location is within a specified radius of the trigger location.

States accessible in minimal user interface

Due to the possibility that not all of the projected features make it into the delivered product, the outline below lists the portions of the app that the user will be able to navigate to in any implementation.

Opp list

Considered to be the main screen of the application, this screen shows the list of active Opps that are awaiting their trigger conditions to be satisfied.

Opp configuration

Opp creation and editing can be done using the same interface. This interface will likely consist of a set of a screens for viewing and configuring a selected Opp.

Team Structure

Jake – CEO of FixCode

Jake has been working on the design and UI of Opportunity. He created all of the flow charts delivered in S1, as well as writing some of sections. He will help design and develop the IOS application.

Zev – Tool Smith

Zev created the outline of FixCode's S1 deliverable. He also has been working on setting up the conceptual architecture used within the app, server, and their communications.

Dylan – Analyst

Dylan wrote much of S1. He will be working on analyzing the product and requirement.

Claire – Document Writer

Claire wrote the Summaries and edited some of S1. She will be working mostly as a document writer and editor.