Software Requirements Specification

for

CalendR  
Shared Calendar /   
Event Coordination App

Version 1.0

Prepared by Group 11  
  
Bryce Shaw  
Matthew Vanderwey  
Stephanie Walsh  
Sean Watters

ENSF 613 – Fall 2019 – Project II

**December 5, 2019**

# **Table of Contents**

[1. Introduction 1](#_Toc26358403)

[1.1 Purpose 1](#_Toc26358404)

[1.2 Overview of Document 1](#_Toc26358405)

[1.2.1 Document Conventions 1](#_Toc26358406)

[1.2.2 Readers and Audience 1](#_Toc26358407)

[1.3 Product Scope 2](#_Toc26358408)

[1.4 References 4](#_Toc26358409)

[2. Overall Description 5](#_Toc26358410)

[2.1 Product Perspective 5](#_Toc26358411)

[2.2 User Classes and Characteristics 5](#_Toc26358412)

[2.3 Operating Environment 5](#_Toc26358413)

[2.4 System Constraints 6](#_Toc26358414)

[2.5 Assumptions and Dependencies 6](#_Toc26358415)

[3. System Features and Use Cases 7](#_Toc26358416)

[3.1 Login / Register for App & Integrate User Calendar (UC-1) 7](#_Toc26358417)

[3.2 Post Provisional Event (UC-2) 8](#_Toc26358418)

[3.3 Post Concrete Event (UC-3) 9](#_Toc26358419)

[3.4 Promote Event (UC-4) 10](#_Toc26358420)

[3.5 Join Event (UC-5) 11](#_Toc26358421)

[3.6 Administrate System (UC-6) 12](#_Toc26358422)

[3.7 Search Event (UC-7) 13](#_Toc26358423)

[3.8 Share Event (UC-8) 14](#_Toc26358424)

[3.9 Advertise to App Users (UC-9) 14](#_Toc26358425)

[3.9.1 Description of Use Case 14](#_Toc26358426)

[4. Summary of Functional Requirement 15](#_Toc26358427)

[5. Product Conceptual Model 16](#_Toc26358428)

[6. External Interface Requirements 17](#_Toc26358429)

[6.1 User Interfaces 17](#_Toc26358430)

[6.2 Hardware Interface 17](#_Toc26358431)

[6.3 Software Interfaces 18](#_Toc26358432)

[6.4 Communications Interfaces 18](#_Toc26358433)

[7. Non-Functional Requirements/ Quality Requirement 19](#_Toc26358434)

[7.1 Performance Requirements 19](#_Toc26358435)

[7.2 Security Requirements 19](#_Toc26358436)

[7.3 Portability Requirements 19](#_Toc26358437)

[8. Prioritization and Release Plan 20](#_Toc26358438)

**List of Figures**

[Figure 1‑1 Use-Case Diagram 3](#_Toc26274679)

[Figure 2‑1: Operating Environment 5](#_Toc26274680)

[Figure 3‑1: Use Case 1 – Login / Register For App (Create Account & Import Contacts) 7](#_Toc26274681)

[Figure 3‑2: Use Case 2 – Posting a Provisional Event 8](#_Toc26274682)

[Figure 3‑3: Use-Case 3 – Posting a Concrete Event 9](#_Toc26274683)

[Figure 3‑4: Use-Case 4 – Promote Event 10](#_Toc26274684)

[Figure 3‑5: Use-Case 5 – Join Event 11](#_Toc26274685)

[Figure 3‑6: Use-Case 6 – Administrate System 12](#_Toc26274686)

[Figure 3‑7: Use-Case 7 – Search Event 13](#_Toc26274687)

[Figure 3‑8: User-Case 8 – Share Event 14](#_Toc26274688)

[Figure 5‑1: App Conceptual Model 16](#_Toc26274689)

[Figure 6‑1: Login Screen Prototype 17](#_Toc26274690)

[Figure 6‑2: Search Screen Prototype 17](#_Toc26274691)

[Figure 6‑3: Hardware Interface 18](#_Toc26274692)

**Glossary:**

|  |  |
| --- | --- |
| **Term** | **Description** |
| **API** | Application Programming Interface:  A set of functions and procedures allowing the creation of applications that access the features or data of an operating system, application, or other service. |
| **Concrete Event** | A formally organized event with a set schedule and location. |
| **Domain** | Internet Domain: the address of a computer network connection and contains information about the type of the entity that owns the address and could include the country or origin. |
| **Domain**  **Model** | A conceptual model of the domain that incorporates both behaviour and data. |
| **Provisional Event** | A block of time defined by the app User in which they are available to join events with attributes that they predefine, e.g. their interests, location, other users that they would like to accompany, etc. |
| **Sequence**  **Diagram** | Depicts interaction between objects in a sequential order, i.e. the order in which these interactions take place. |
| **SDK** | Software Development Kit:  A group of tools that enable the programming of mobile applications. |
| **Use Case** | A list of actions or event steps typically defining the interactions between a role (known in the Unified Modeling Language (UML) as an actor) and a system to achieve a goal. |

# Introduction

## Purpose

The unique value proposition of this app is how it will enable people to share with a curated audience, when they have free time, along with their location, and some identifiers on how they would like to spend that time. From there, the system would allow Users to see other people in their community who have similar free time and interests, facilitating the easy coordination of an offline interaction.

While calendar applications (Google calendar, Facebook events) already exist, these are poorly suited for connecting people who have free time but no set plans. As well, there are already existing platforms for connecting people with shared interests (online forums / message boards / Facebook groups) but these are only suited for facilitating group events that are predetermined. This app proposes to take these existing social media tools to the next level, by not only helping its Users coordinate planned (Concrete) events, but by also helping them turn their otherwise nebulous plans (Provisional Events) into organized events shared with others on the basis of common interests and objectives.

The vision for this app is that it would work in concert with these existing tools, which its potential Users are likely already vested in. The app would enable a custom activity tag to be used by members of a Facebook group to connect them on this new platform. As well, the User will be able to generate a calendar which they can use to display results from a query to the system (e.g. generate a Google Calendar populated with all the offerings (events and free time) tagged with “running” for a specified location and time of week.

## Overview of Document

### Document Conventions

This document follows the standards for technical documents outlined in the charter for Group 11, including the established “Group 11” styles for Microsoft Word (Office 365) documents.

### Readers and Audience

This document is intended to provide project managers, business analysists, vendors, system administrators, marketing staff, developers, and testers with a clear and concise understanding of the proposed system’s features and modes of operation.

Section 2 of this document provides the context for the proposed system, including its major components and their external interfaces. This section also outlines the primary features of the system, anticipated User classes, operating environment, system constraints, User documentation, external dependencies and assumptions made in preparing this specification.

Section 3 presents detailed descriptions of the proposed system features and use cases. This includes supplemental Sequence Diagrams, where applicable.

Section 4 provides a summary of the system’s functional requirements based on the use cases presented in Section 3.

Section 5 outlines the Conceptual Model of the proposed system, showing the relationships between the various system components.

Section 6 describes the logical characteristics of each interface between the proposed application and its Users. This includes a prototype of the GUI interface and descriptions of the anticipated hardware, software and communications interfaces.

Section 7 reviews the non-functional requirements of the system, including performance, security and portability requirements.

Section 8 describes the methodology used for prioritizing the functional requirements of the proposed system. Rankings for the system’s functional requirements in order of priority, along with potential risks and proposed release dates, are also provided in this section.

Here are the suggested readings by audience:

|  |  |  |
| --- | --- | --- |
| **Audience** | **Reading Suggestions** | **Sign-Off Required** |
| Project Manager /  Business Analyst | Sections 2 & 4 & 6 | Yes |
| Vendor | Sections 2 to 4 | No |
| System Admin | Sections 2 to 4 | Yes |
| Marketing Staff | Sections 2 & 6 | No |
| Developers | Sections 2 to 6 | Yes |
| Testers | Sections 2 to 5 | No |

## Product Scope

The primary objective is to bring current friends closer together while also helping Users to expand their social circles. The secondary objective of the app is to help connect people with others in their local area who share interests and enjoy similar activities.

The proposed app would allow Users to search for events in their area, as well as publish their interests and availability, both in terms of schedule and location. Users would be able to control who can see their interests and availability, and premium Users would be able to integrate the app with their third-party calendar services, such as Outlook, iCal and Google Calendar.

Sharing your interests and availability (creating a provisional event) on the app will automatically give Users the option to connect with other Users who have similar interest and availability. Also, it will provide them with an array of events that have already been organized (concrete events) matching the similar criteria. As new events are organised, by either other app Users or Vendors/Promoters, Users will be notified of these new events that they may be interested in joining. Once a User has joined an event, they can chat (direct/instant message) with other Users who have joined the same event and invite other Users to join as well.

For a fee, Vendors/Promoters can post and promote events, directly targeting Users who may be interested in their event based on their profiles and offering them special deals for attending their event. As an example, a concert promoter would pay to promote their music event to Users who have expressed an interest in the performing artists (or similar artists) and offer them discounted tickets to the event for connecting with them through the app. Thus, providing the app Users with a deal on an event that they are likely interested in and the Promoter with an engaged audience for marketing future events to. The app would be made available for free, with advertisements powered by services like the Google Ads API. For a fee, premium users would be able to suppress these adds.

A Use-Case Diagram of the proposed App is shown in Figure 1-1 below:

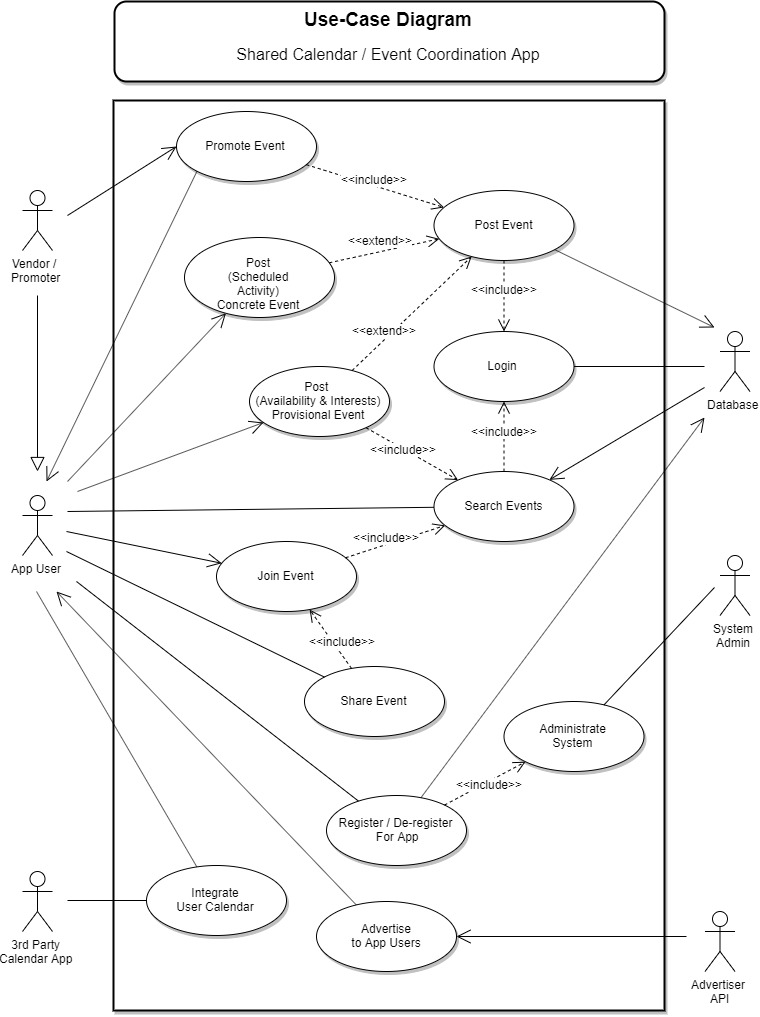


Figure ‑ Use-Case Diagram

## References

The following references were used in developing this document:

1. B. Shaw, *JoinR: Vision & Scope*, Business Case, 2019. [thoughts spoken out loud] Original concept for this app/platform.
2. The App Solutions Inc., “How to Build a Social Media App 2019”, 2019. [Online]. Available: <https://theappsolutions.com/blog/development/how-to-develop-social-media-app/> [Accessed December 2, 2019]
3. Canva, “Design Tutorials”, 2019. [Online]. Available: <https://designschool.canva.com/> [Accessed December 1, 2019]
4. T. Smale, “How to Make Money with Social Media”, 2016. [Online]. Available: <https://www.entrepreneur.com/article/274687> [Accessed November 25, 2019]

# Overall Description

This section provides background information about the app and its platform. While this section will not describe the components in detail, it will provide high level details about the system components.

## Product Perspective

This product is a new, shared-calendar app to help Users plan, share, and book events. While this application would include integration with other third-party platforms, e.g. Google Calendar and Facebook, it is intended to be a self-contained product. The whole system would be used both online and on a mobile phone. The front end would enable Users to search for events, post events (both Concrete and Provisional) and join events. There would also be a feature which enables Vendors/Promoters to post events for Users to join. Advertisers could also promote their events, activities and products targeted to Users based on their interests through the app as well.

## User Classes and Characteristics

On the front end, the primary User of this app will be an individual looking to fill their free time with events they enjoy doing. Their use will be limited to searching, posting, and joining events. This is the most important class of User to satisfy, as this is who the advertisers, venders and promoters will be trying to reach, and the people who will be joining the events.

On the back end, and one of the means that the app would be monetized, a key group of Actors would be those who sign up as Advertisers. They would not have direct access to individual User contact information; however, they would have groups of Users (categorized by their interests) to choose from for their advertising through the app.

In addition, the platform would also have a Vendor/Promoter class of User which would include businesses, artists, community activists, etc. who would post and actively promote events for Users to attend through the app.

## Operating Environment

This will be an app-based environment where Users will access the site from their desktop or laptop, or the mobile app through their mobile device. The system will be hosted on a server which will be the responsibility of the system administrator to manage directing or coordinate with the management of the hosting provider.

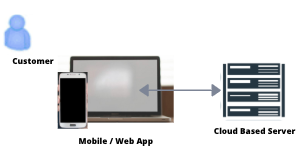


Figure ‑: Operating Environment

## System Constraints

Because the final product would be web-based, it must be fast enough on both desktop and mobile devices to keep the User engaged. The system must work on all types of web browsers and have similar access speeds on all platforms. This may limit the options available for final design purposes.

The system must interface with the various popular calendar apps in the marketplace today including Google Calendar, iCal, Outlook, and others.

Each User will be able to choose how much personal information they share with the app and other Users. Individual User data will be protected from advertisers and other Users, and only used on an aggregated level.

The web-based interface with the platform must have a security certification and be secure with a https:// URL.

## Assumptions and Dependencies

For the purpose of this system, the app should use existing APIs for modern calendar apps, the Google Advertisement API, as well as the iOS and Android SDKs. The main database for the platform should be hosted on a secure SQL server, accessible from anywhere on the internet. The mobile app would be developed in Apple XCode for iOS and Kotlin for Android.

This system would not be dependent on any third-party commercial components and is not intended to substantially reuse any existing software components developed for other applications.

Further details on the foreseen dependencies that this app may have on external systems are outlined in the Vision and Scope document provided separately.

# System Features and Use Cases

## Login / Register for App & Integrate User Calendar (UC-1)

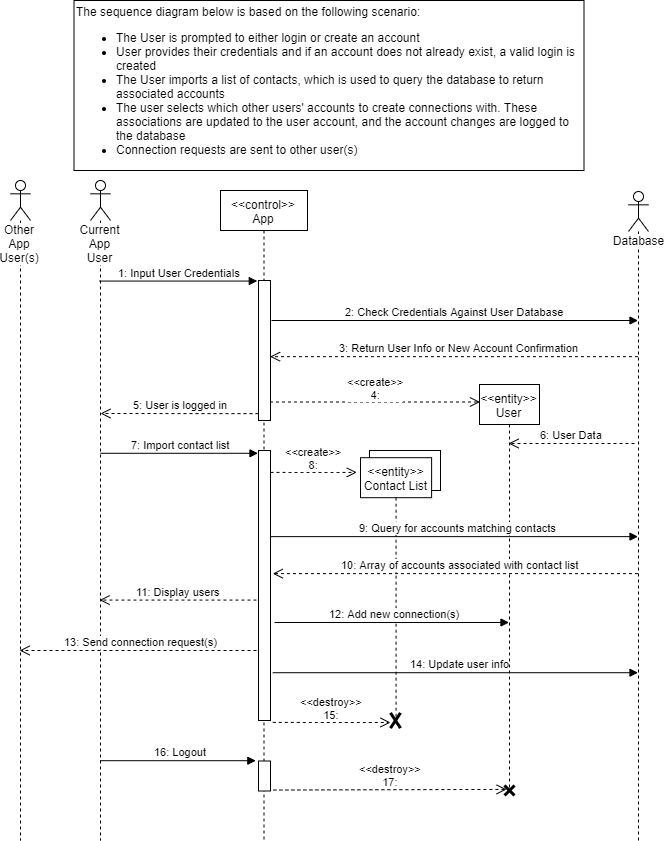


Figure ‑: Use Case 1 – Login / Register For App (Create Account & Import Contacts)

## Post Provisional Event (UC-2)

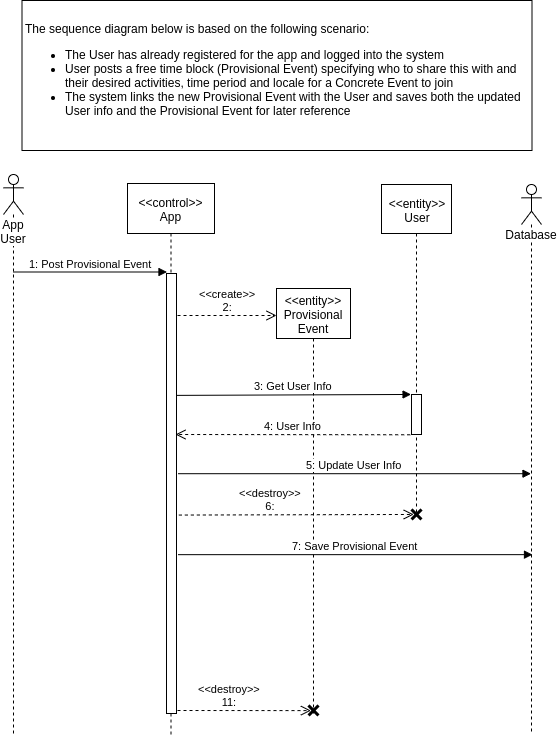


Figure ‑: Use Case 2 – Posting a Provisional Event

## Post Concrete Event (UC-3)

## 

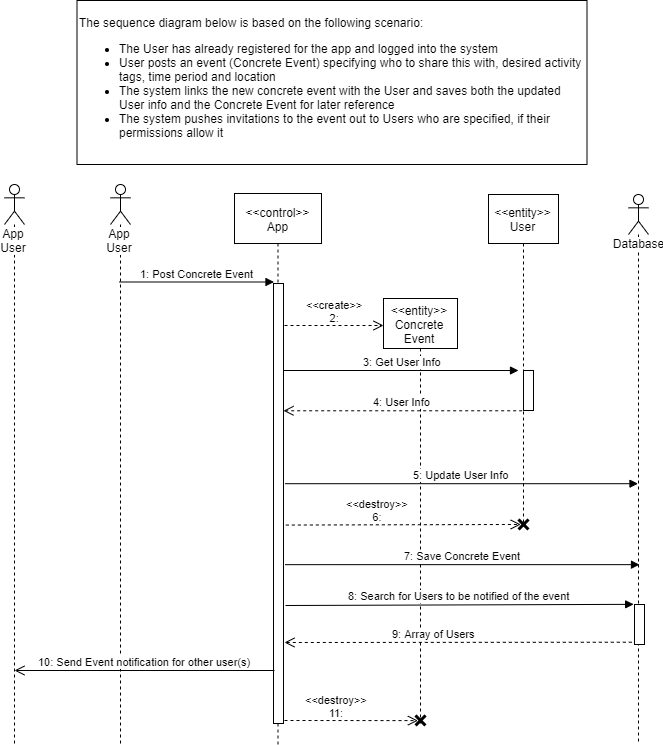


Figure ‑: Use-Case 3 – Posting a Concrete Event

## Promote Event (UC-4)

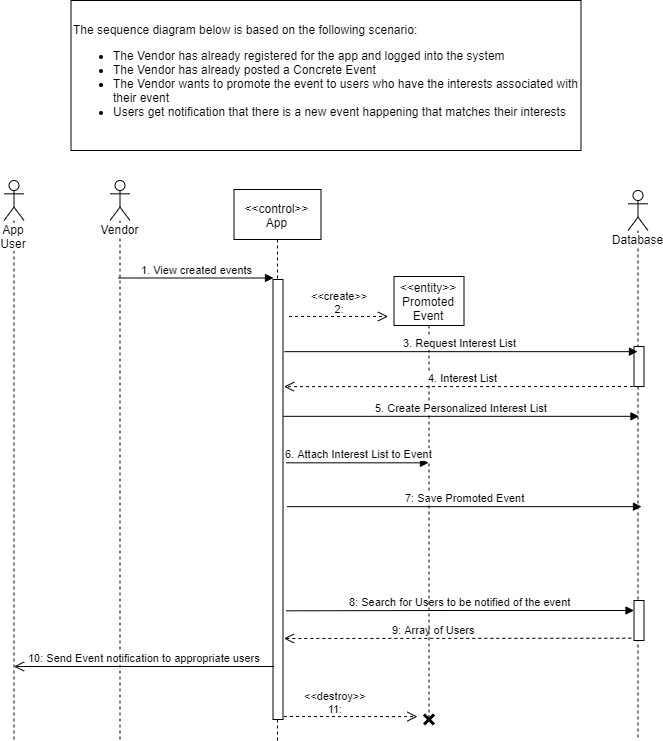


Figure ‑: Use-Case 4 – Promote Event

## Join Event (UC-5)

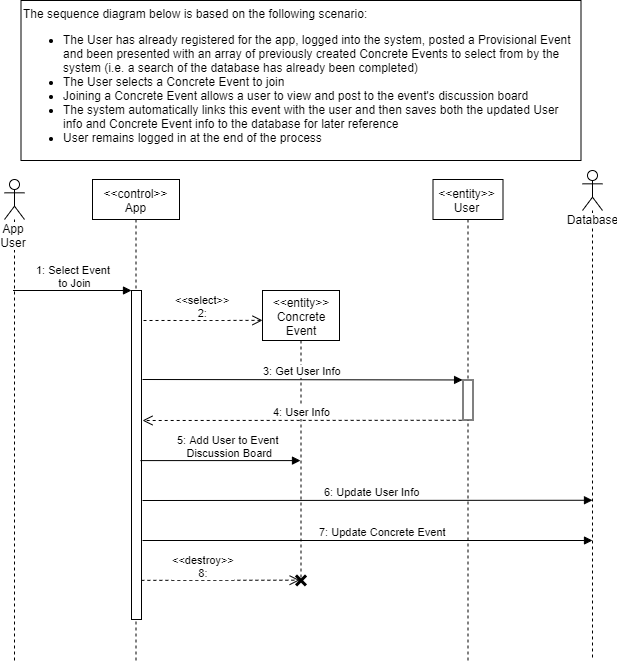


Figure ‑: Use-Case 5 – Join Event

## Administrate System (UC-6)

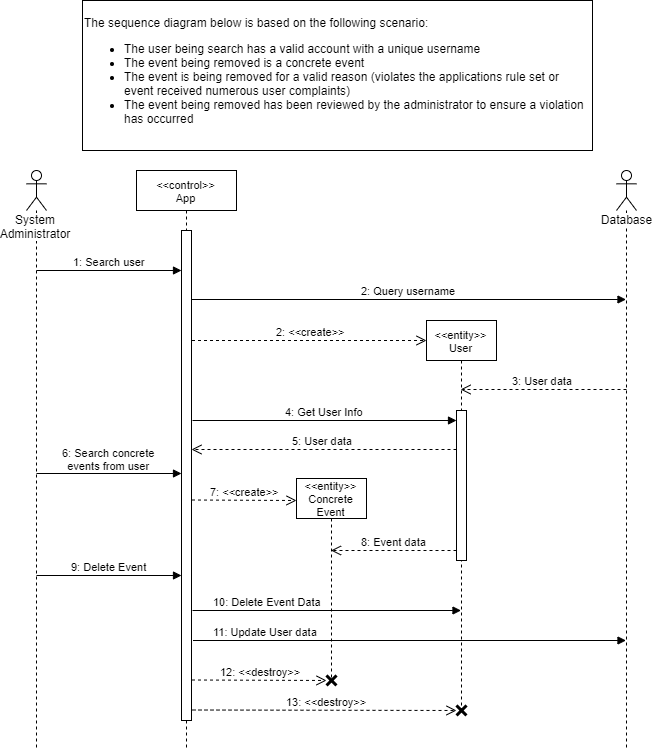


Figure ‑: Use-Case 6 – Administrate System

## Search Event (UC-7)

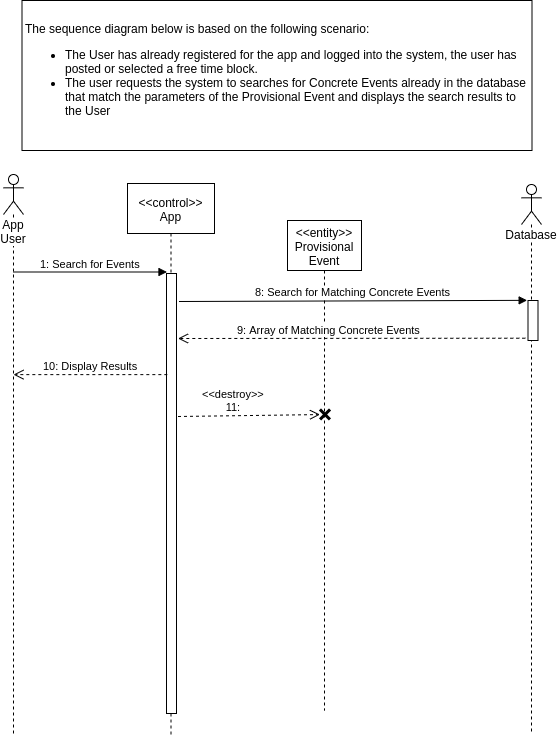


Figure ‑: Use-Case 7 – Search Event

## Share Event (UC-8)

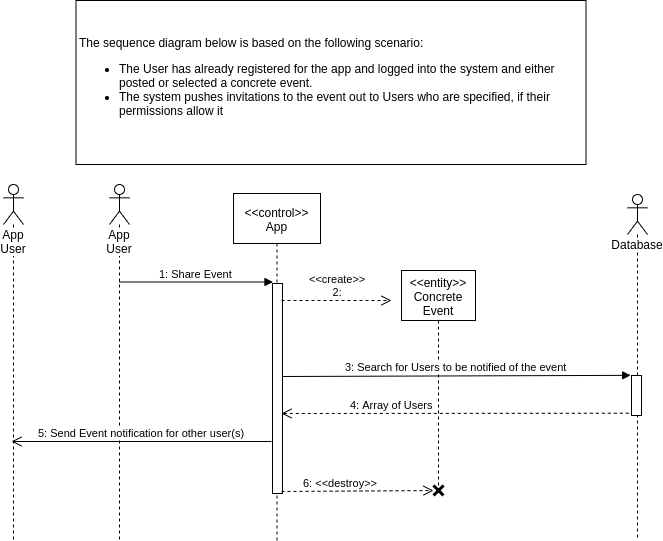


Figure ‑: User-Case 8 – Share Event

## Advertise to App Users (UC-9)

### Description of Use Case

Our app will use the Google Display Network to advertise to Users. The way Google advertising works is that the advertiser chooses certain interests and then Users are shown ads based on their Google accounts / cookies. Our app will simply connect with the Google API and nothing else on the app side is required.

# Summary of Functional Requirement

|  |  |
| --- | --- |
| **RE#** | **Description** |
| **RE 1** | App Users should be able to login to the system using their Username and password. |
| **RE 2** | Users should be able to search the app for events matching their criteria. |
| **RE 3** | Users should be able to post provisional (free time) events with their interests. |
| **RE 4** | Users should be able to join concrete events that exist in the database. |
| **RE 5** | Users should be able to share the events that they have joined. |
| **RE 6** | Vendors should be able to promote events that they have posted. |
| **RE 7** | The System Admin should be able to de-register Users if they have been reported by multiple other Users (or for any other violations of the terms of use). |
| **RE 8** | Advertisers should be able to use provided User data (general interests) to push ads for their events/activities to the app. |

# Product Conceptual Model

Figure 5-1 below is a conceptual model of the app. Please note that this is not a design class diagram, and further refinement (including potentially inserting additional classes) will be required at the time of system design.

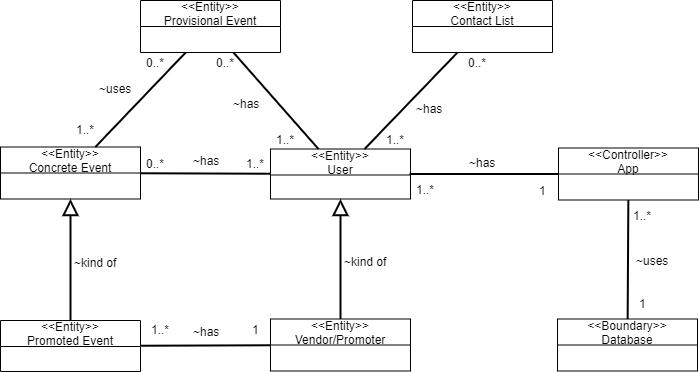


Figure ‑: App Conceptual Model

# External Interface Requirements

## User Interfaces

The front-end, client facing User Interface (UI) elements for this product will be written in Kotlin (for Android) and Apple XCode for iOS. A focus will be made to ensure that functionality and user experience is maintained across Apple and Android devices.

A/B testing (or split testing) should be completed as part of the product development to test UI elements (layouts, colours, interface structures, etc.). User data generated while browsing the site (e.g. referral source, search criteria, time spent on the site, when the user left the site, etc.) will be analysed to generate new UI configurations, and A/B tested. Machine learning should be applied to generate more insight into the potential app tweaks.

The exact requirements for the user interface will be outlined in a separate document. However, as a reference, the UI should follow the format shown in Figures 6-1 and 6-2 below.

|  |  |
| --- | --- |
| Figure ‑: Login Screen Prototype | A picture containing clock  Description automatically generated  Figure ‑: Search Screen Prototype |

## Hardware Interface

Servers / hardware will be implemented to:

* Implement client facing frontend elements (Web App Servers)
* Store site usage data, client booking and payment details, client account details. (Data Warehouses)
* Servers to support backend functions and implement vendor facing API elements (Job Servers)
* Implement backend elements

Each of these elements will be at least one server instance, which will physically be located on an Amazon Web Service (AWS) or Google server array.

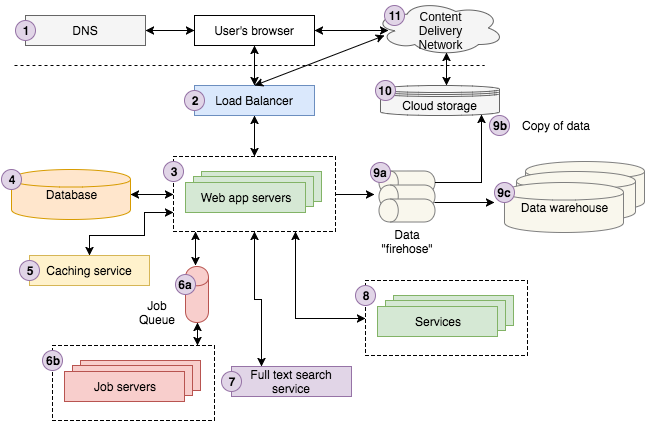


Figure ‑: Hardware Interface

## Software Interfaces

The client facing software will run on either iOS or Android, using a react native frontend. The client-side app will interface with a Google Firebase backend, which we will use for authentication and implementing a database. Interfaces with the advertiser APIs will be configured based on their standards.

## Communications Interfaces

The app itself will be the primary user interface, with premium users being able to integrate the app into their third-party calendars. We will interface with a firebase no SQL database.

# Non-Functional Requirements/ Quality Requirement

## Performance Requirements

The system should be fast for the User to be able to search, create, and join events. The system should be visibly appealing so that the Users should want to use it to manage their time. The site should be able to maintain speed to support multiple Users using the app at one time.

The site should contain a Google Analytics tag provided by the client so that they can maintain data on the site’s Users and be able to better advertise and remarket to their Users. The backend of the site should be User friendly for the client to make copy changes easily.

## Security Requirements

For User data, our site must be secure and block any types of unauthorized use of a User’s account.

## Portability Requirements

The app should work on mobile devices, tablets, and desktop computers. It should also work on all type of browsers including Chrome, Firefox, Safari and Edge.

# Prioritization and Release Plan

**9.1 Product Development Milestones**

Below are the project milestones. By releasing as early as possible we will be able to generate data to provide feedback into the products functionality and create improvements wherever needed.

**Jan 2, 2019 – Front End Prototype I**

Proof of concept: Basic UI functionality, focusing event creation, sharing and communicating. Datalogging should be included to being generating a data set to inform design choices going forward.

**Feb 2, 2019 – Front End Prototype II**

Second iteration of the front-end prototype, with enough functionality to fully simulate the user experience.

**Feb 20, 2019 – Full Stack Prototype I**

A full stack prototype that includes all functionality to incorporate profile data, vendor promotion and administrative capabilities

**Mar 1, 2019 Soft public release**

All functionality should be in the product. The initial release will occur in a controlled geographical location, such as Calgary, to test market acceptance. Users can utilize all site functionality, but restrictions may occur depending on the geographical location of the events being created. User base may be small enough that technical problems involving event creation and sharing can be resolved on a case-by-case basis, if the resources exist for this.

**Mar 30, 2019 – Hard public release**

All functionality is implemented, services are live on a global scale. User metrics are recorded and tracked closely. Features and UI changes are A/B tested for impact on user experience.

**9.2 Systematic Decision Grid**

The potential features to be incorporated into the initial launch of the application weight through the use of the systematic decision grid below. The top 4 scored features were selected.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Feature** | **Performance** | **Security** | **User Satisfaction** | **Portability** | **Total** |
| Communication System between users | 1 | 2 | 3 | 3 | **9** |
| Mobile Platform | 1 | 1 | 2 | 3 | **7** |
| Integrated Third Party Calendars | 1 | 1 | 3 | 3 | **8** |
| Administrative System | 3 | 3 | 2 | 0 | **8** |
| Ticket Purchasing for promoted events | 1 | 1 | 3 | 2 | **7** |
| Interaction between user events | 2 | 2 | 3 | 2 | **9** |

**9.3 Feature Release List**

1. Event creation, interaction, event sharing
2. Basic User interface
3. Event promotion
4. Integrate 3rd Party Calendars
5. Improved User Interface
6. Administrative
7. Advertisement Implementation

|  |  |  |  |
| --- | --- | --- | --- |
| **Feature ID** | **Priority Level** | **Release Date** | **Remarks** |
| FR 1  FR 2 | 1 | Jan 2, 2019 | Front End UI proof of concept. Allows users to create, search, and share events with other users |
| FR 3 | 1 | Feb 2, 2019 | Interface with local business and venues, highlight the event promotion features incorporated in the UI |
| FR 6 - A | 2 | Feb 2, 2019 | Basic data logging to generate site usage data for the frontend prototype |
| FR 6 - B | 1 | Feb 20, 2019 | More advanced site usage tracking |
| FR 7 | 2 | Mar 1, 2019 | Basic Advertising layout concept |
| FR 4 | 2 | Mar 30, 2019 | Integrate functionality to connect application with 3rd party calendar apps such as Google Calendar |
| FR 5 | 1 | Mar 30, 2019 | UI changes are tested against user metrics |