

---

# **Software Requirements Specification**

for

## **Event Bay**

Gabriel Castro  
Brayden Cloud  
Casey Kinnamon  
Joe Niu

// TODO

<b>1. Introduction</b>	<b>3</b>
1.1 Purpose	3
1.2 Scope	3
1.3 Definitions, Acronyms, and Abbreviations	3
1.4 Overview	3
<b>2. Description</b>	<b>4</b>
<b>3. Specific Requirements</b>	<b>4</b>
3.1 External Interface Requirements	4
3.2 System Features	5
3.3 Nonfunctional Requirements	6
<b>4. Index</b>	<b>7</b>

# 1. Introduction

## 1.1 Purpose

The purpose of this document is to build a Progressive Web App which aims to organize impromptu meetups without the hassle of creating Facebook Events.

## 1.2 Scope

The name of this Progressive Web App is “Event Bay” or “Ebay” for short. The purpose of our Progressive Web App is to coordinate informal meetups without the hassle of creating Facebook Events or having to. Our Progressive Web App is established on a non-relational database. Our non-relational database server will initially be intended to support the hundreds of million of people living in the United States. Our ultimate objective is to empower the connection of people by helping improve communication between one another.

## 1.3 Definitions, Acronyms, and Abbreviations

This document uses the ensuing formality.

Amazon EC2 - Amazon Elastic Compute Cloud

API - Application Programming Interface

AWS - Amazon Web Service

GDPR - General Data Protection Regulation

HTTPS - Hypertext Transfer Protocol Secure

PCI - Payment Card Industry

PWA - Progressive Web App

SDK - Software Development Kit

## 1.4 Overview

This document contains the Software Requirement Specifications for our Progressive Web App. After the introduction, we will discuss the overall description of our app. In our overall description, we will discuss a high-level overview of our PWA. Next we will discuss the specific requirements of our PWA. In our specific requirements, we will discuss the software requirements of our PWA in detail. Lastly, we have the index of this document. In the index, we will list alphabetically names and subjects with references to where they will be located in our document.

## 2. Description

The idea for EventBay came from a need for a quick and easy event scheduling application. EventBay is meant to be an alternative to other event organization apps such as Facebook events, and will allow users to create and RSVP for events through an account tied to their phone number. While it is expected that users will primarily use the application on their mobile device, EventBay will be platform agnostic. EventBay is also intended to be quick and easy to use, and as such will not require users to download an application. For these reasons, EventBay will be a Progressive Web Application allowing any user to easily use the app through a browser, or in the case of a mobile user, pin the app to their home screen.

The core functionality of EventBay will allow a user to create an account tied to their phone number or use OAuth to tie their account to a preexisting Google or Facebook user. Users will then be able to create events, specifying the date, time, location, description, and the name of the event. The creator of an event will also have the ability to kick people from the event and modify or cancel the event entirely. Other users will be able to navigate to a specific event page through an event link, which the creator of the event can save and share to specific people or group chats which will provide an easy way to invite groups of people. Once at an event page, users will be able to see the event details and RSVP for the event

EventBay's anticipated users are groups who want to find a quick, platform unspecific way of organizing small events. We are assuming that users will use EventBay to organize events for groups of people in group messages. Because of this assumption, we have made the design choice of using event links as the primary way of sharing an event. This allows users to share events easily with single or multiple group chats, regardless of the chat platform used. When a user creates an event, the user will be the admin of that specific event. There is no inherent difference in user types, just the context that the user has in relation to specific events.

## 3. Specific Requirements

### 3.1 External Interface Requirements

#### **User Interfaces**

General users will interact with our application through the use of a Progressive Web App (PWA). The user will be able to create an account using an email and password or OAuth through a “Create Account” button on the home page, and login to our app using their newly created account. Once logged in they can view user profiles, event details, and create events using the app’s drawer.

Additionally, an administrative option will be visible on all event and user detail pages to users created as an administrator. Administrator accounts can be created by the developer only.

#### **Software Interfaces**

Ebay will interface with various services such as our REST API and Auth0. The REST API will be used to satisfy the majority of features required by our application. All interactions with event data will be done through the API, while Auth0 will handle interactions with user data and authentication. Additionally, links to events will be short and easy to remember, enabling links to be easily shared among friends in group chats.

#### **Communication Interfaces**

All communications with the API and Auth0 will occur over HTTPS. Communication with the MongoDB database will be done using the MongoDB SDK, which will use HTTPS.

### 3.2 System Features

This section will outline the various functional requirements of our application. Our application will consider two types of users: User and Administrator.

#### **Create Account**

Any User can create an account using their phone number and a password. Text verification will be used to validate the phone number. An alternative to create an account will be using OAuth methods such as Google or Facebook. All account creation will be handled by Auth0.

#### **Login**

Any User who has created an account or Administrator can login using the phone number and password they provided. Auth0 will validate the combination and log the user in, returning their profile data along with a temporary access token.

**Delete Account**

Any User who has created an account or Administrator can delete their account. Auth0 will handle the database side of deleting an account, while the API will handle any cleanup of data that is necessary. For example, deleting the removed user's events.

**View Profile**

As a User or Administrator logged in, you can view another user's profile details including names and previous events that user has attended.

**Search Users**

As a User or Administrator logged in, you can search for other users on the app by typing their phone number or name into a search bar.

**Create Event**

Any User who is logged in can create an event. Creating an event will entail the user entering event details such as date, time, location, event name, and short description. In this step, the user can also provide a list of users or phone numbers to invite immediately.

**Invite Users To Event**

Any User who is logged in and viewing their own event can send invites to any other user. Users should be found by their name or phone number.

**Delete Event**

Any User who is logged in and who has created an event can delete their event. Deleting an event will remove the event from all RSVPs calendars. Additionally, any Administrator will also have the ability to delete the event.

**Share Event Link**

Any User who is logged in should be able to share an event via link. Anyone who is viewing the link can see event details and sign up to RSVP.

### 3.3 Nonfunctional Requirements

**Portability**

Event Bay must be accessible via mobile phone or desktop browser.

**Scalability**

As our application increases in use, the app must stay fast in response times and data access.

**PCI Compliance**

Our application must be PCI compliant. This means all personally identifiable information stored on a user must be encrypted at rest.

**GDPR Compliance**

EBay must be GDPR compliant. This means all user data stored in the database must be accessible to the user. Users and Administrators must be able to download their data as well as delete it.

**Security Requirements**

Users should be able to make requests only on behalf of their own account. This includes creating and deleting events as well as RSVPing to events and deleting a user's account.

## 4. Index

**Amazon EC2** - Amazon Elastic Compute Cloud has resizable, secure cloud computing services.(page 5)

**Auth0** - is an extensive third party authentication service which will allow passwordless authentication via text verification as well as OAuth integration through Google and other platforms(page 2,3,4)

**API** - a set of functions and procedures that can communicate with a database to create, remove, update, delete.(Page 2,4)

**AWS** - Amazon Web services offers cloud computing services.Can be used to host websites or databases(page 2)

**GDPR** - General Data Protection Regulation are guidelines for collection and processing personal information.(page 6)

**HTTPS** - Hypertext Transfer Protocol Secure is the protocol that is used to transfer data on the web.(page 2,4)

**MongoDB** - MongoDB is a document database, which means it stores data in JSON-like documents(page 2,5,6)

**PCI** - payment card industry data security standard is an information security standard.(page 6)

**Progressive Web App** - A type of software application that is delivered via the web and works on relatively any platform.(page 2,4,5)

**REST API** - Is a way for two code bases to communicate in our project its used to create, remove, update and delete information from the database.(page 2,3)

**SDK** - Software Development Kit is a set of tools used for developing applications. Gives the tools to work with and communicate with MongoDB. (page 4)