Software Requirements Specification

Table of Contents

Introduction

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. The name of this app is “Event Bay” or “Ebay” for short. The Event Bay frontend will be created using the Angular Framework. The Angular Framework is a structural framework for dynamic web apps. Our backend will be a REST API created using the Express Node library. A REST API is an application program interface that handles HTTP requests to interact with data. Our backend will be hosted using an AWS EC2 instance. An AWS EC2 is a web service that provides a protected, reusable compute capacity in the cloud. Our Progressive Web App or PWA for short will be hosted using Google’s Firebase Hosting. Firebase Hosting provides quick and protected hosting for web apps. For our database, we will be using MongoDB which is a NoSQL database which Node has great support for. For authentication we’ll be using Auth0. Auth0 is an extensive authentication service which will allow passwordless authentication via text verification as well as OAuth integration through Google and other platforms.

This document contains the Software Requirement Specifications for our Progessive 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.

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.

Specific Requirements

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 and login to our app, view user profiles, event details, and create events.

In addition to general users, there will be an administration view in our application which will allow an administrator to view all events, profiles and perform destructive actions on them.

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.

System Features

This section will outline the various functional requirements of our application, including all user and administrator functionalities.

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

Search Users

As a user 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, an admin 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.

Nonfunctional Requirements

Platform

In order to reach the most users, we will be developing a PWA for the web. Creating a web application enables to reach every modern platform with a single codebase. The PWA aspect allows users to save our application to their home screen or desktop for quicker access.

Scalability

As our application increases in use, the app must stay fast in response times and data access. To accomplish this, we are using MongoDB which allows us to scale horizontally. This means as our user base grows, MongoDB will handle the increase in traffic. Amazon EC2 has a similar feature. As our traffic begins to increase, Amazon will auto-scale our microservices to account for the increase in traffic. Our EC2 instances will sit behind a load-balancer which will allow requests to be handled efficiently.

PCI Compliance

In order to comply with the PCI standards, we are taking advantage of MongoDB’s encryption-at-rest method of storing data. This means that any data dumped from our MongoDB database will be encrypted.

GDPR Compliance

In order to comply with the GDPR standards, we must enable a way for users to view and delete all data kept on them. To accomplish this, we will delete all references of a user upon deletion of their account. Additionally, we will provide a link to download the user object we have on a user stored in MongoDB.

Security Requirements

In order to protect the privacy of our users, we will ensure all communications occur via HTTPS and will be following security best practices. This includes hashing passwords, keeping sensitive data out of local memory, etc..

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)

AWS - Amazon Web services offers cloud computing services.(page 2)

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

HTTPS - Hypertext Transfer Protocol Secure is a way of communication.(page 2,4)

MongoDB - MongoDB is a NoSQL database where we will store all information. (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 (page 4)