Software Requirements Specification & Design

for

Player Needed

Version 1.0 approved

Prepared for Nasseef Abukamail

Prepared by the 2g2q team

2git2quit

9/24/21

Table of Contents

Revision History

Name	Date	Reason For Changes	Version
2g2q	09/24/21	Initial edit	1.0a
2g2q	10/01/21	Revision	1.0b
2q2q	11/01/21	Revision	1.0c
2g2q	01/26/22	Revision	1.0d
2g2q	04/21/22	Revision	1.0e

1. Introduction

1.1 Purpose

To create an application that allows its users to schedule, sign up for, and keep track of “pick-up” games with other users. The application will help manage a large variety of events from traditional and nontraditional sports to fundraisers, parties, and generic events. Additionally users will be able to join and interact with groups and its members.

1.2 Document Conventions

When identifying items/groups in series or list, the topic will be bolded and each word capitalized. Additionally the text associated with each item must be a completed sentence and have appropriate punctuation. An example below (note: items need not be indented):

> **First Item:** First item text.
> **Second Item:** Second item text.
> **Third Item:** Third item text.

Heading titles will be in ‘Heading2’, Times new Roman, 14pt.
Body paragraphs will be in Normal, Arial, 11pt.

1.3 Intended Audience and Reading Suggestions

This document is intended to give an outline of the product goals, requirements and scope and thus could be useful to project developers and designers for initial implementation and future maintenance. Additionally this document may be used as reference for documentation written at a later date.

1.4 Product Scope

The product is an application that allows a user to create, join, and manage private or public groups and events with a primary focus on sporting and other small group events. Group and event members should be able to communicate with one another within the context of the event or group. Additionally group creators should have exclusive rights to edit, and manage a group and its members.

1.5 References

There are currently no documents/resources referenced in this document. References will be added in the future when qualifying modifications are made to this document.

2. Overall Description

2.1 Product Perspective

The application being specified in this document is a new product originally conceived as a project for Ohio University's CS4560 Software Development I & II class. The concept for this application originated from the class's instructor, and our team's client, Nasseef Abukamail. The project is split into a backend and frontend that communicate with one another in order to coordinate a variety of actions and data.

2.2 Product Functions

A user will be able to create an account, delete an account, join a public event or group, request to join a private event or group, create events or groups, manage events or groups, update/customize a profile, view public groups they are not part of, chat with group members, view a facade of private groups they are not part of.

2.3 User Classes and Characteristics

Sports Teams/Athletes: This class of users is expected to make up the bulk of the applications overall user base. This group of users is expected to interact with the application regularly to host private sporting events for team members/friends as well as public sporting events such as "pick-up games."
Individual Private Users: This user class is expected to make up a large portion of the application's user base and interact semi-regularly with the application, primarily using it to create, join and manage smaller, informal private events.
Educational Institutions/Instructors: This user class is expected to be made of predominantly individual teachers and schools. This is expected to be a fairly small user class that regularly uses the application for educational coordination such as field-trips, extracurriculars, and attendance.
Students: This user class is expected to use the application regularly for educational-related events hosted by fellow students or instructors. Students may use this application to coordinate study sessions, meetings and educational events.
Charities: This is expected to be the smallest user class. This class of users is expected to interact with the application infrequently, mostly using it for small private events.

2.4 Operating Environment

The product will run primarily as a native application on iOS and Android devices. Additionally the application will be able to run on any web environment supporting the latest version of React. The web version of this application should be used on mobile devices only, as the application is not designed to support laptop, desktop, or TV screen resolutions causing rendering issues.

2.5 Design and Implementation Constraints

Limiting factors to developers include time, small development team, other school work, initial learning curve, and operating environments such as Android and iOS. We are limited to the tools and applications that have been given to us by the school or that we can obtain ourselves at no cost. Data pertaining to an individual user should only be viewable by that individual user or developers/admin.

2.6 User Documentation

Setup: Gives a step by step guide to setting up the environment for, and installing/running the back-end and front-end. This will primarily be used by developers in the future and is subject to change.
User Manual: Gives end-users a plain english description of the applications capabilities and descriptions on how to perform key actions within the application.
Deployment: Gives the developers a guide to bundling the applications and shipping it for production on the App and Google Play Store (iOS and Android Builds)

2.7 Assumptions and Dependencies

The application specified in this document will be dependent on several libraries and frameworks. Any of these libraries or frameworks could develop a security issue, become out of date, or change their implementation in a way that disrupts the functionality of the specified application. In addition to these dependencies, the speed of the application will depend a lot on our server speed and ability to process requests.

React: version 16.13.1
npm: version 7.0.14
Expo: version 42.0.1
Firebase: version 0.2.3

3. External Interface Requirements

3.1 User Interfaces

The user interface will follow standard modern conventions. The UI will be intuitive, drawing design concepts from other modern, widely-used applications. We stuck with a slick dark theme aesthetic to both relieve stress on users' eyes, and save battery on OLED style phones. Button elements have a primary and secondary color and are reactive when interacted with. On every screen besides login and signup, there is a navigation bar at the bottom of the screen allowing the user to navigate between key screens on the application. The screens include a home page, profile page, groups page, create event/group page, and settings.

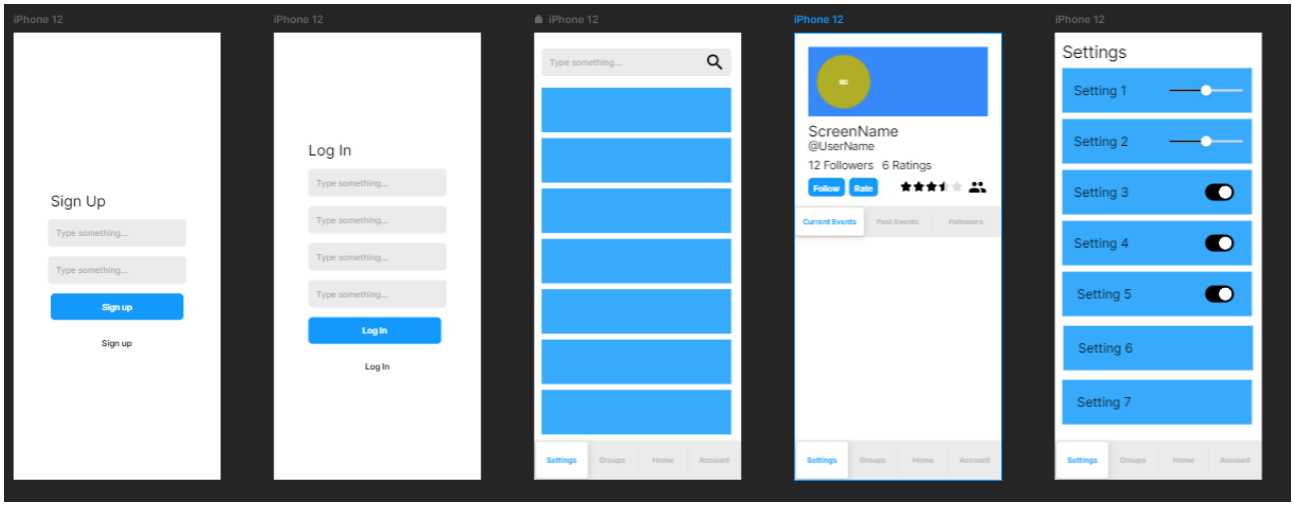


Diagram 3.1.1(Mock Up)

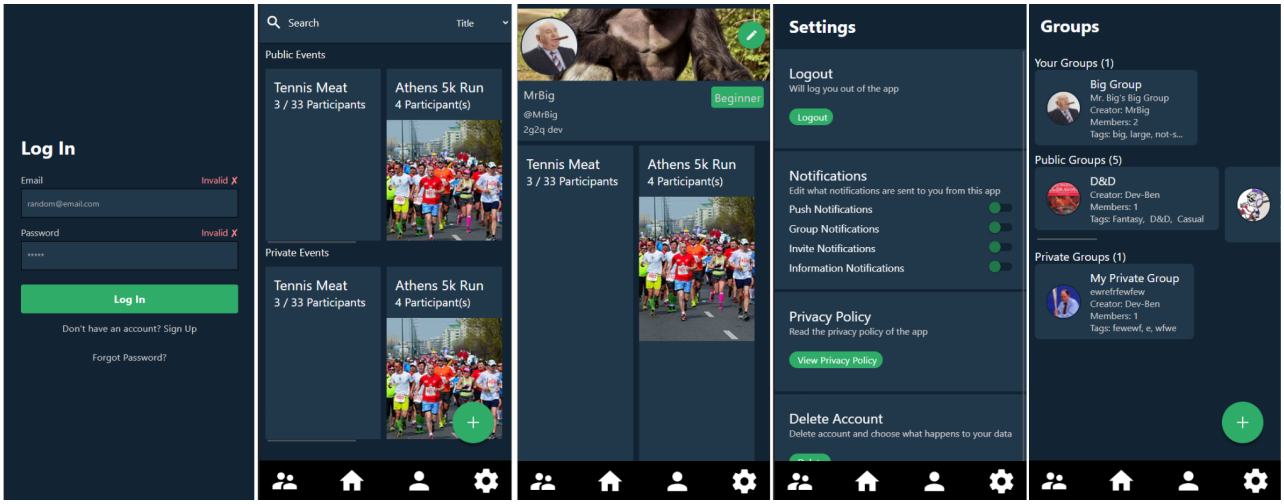


Diagram 3.1.2(Final)

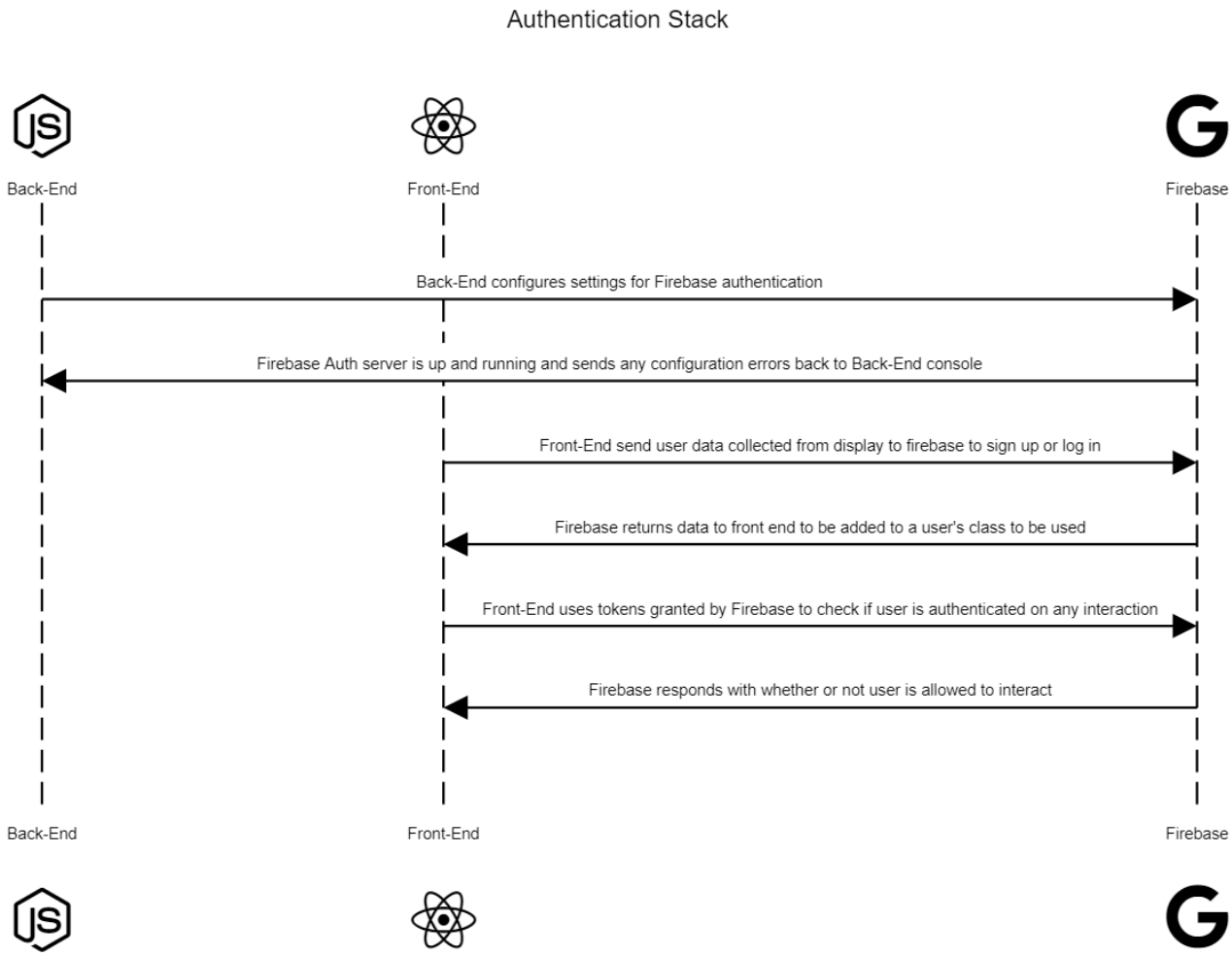
3.2 Hardware Interfaces

The application works for computers and android. The Expo platform allows us to simultaneously make Native apps for Android, and the web with React Native. React Native allows us to develop for both platforms simultaneously using JavaScript. We are utilizing a “write once run anywhere” approach in order to provide quicker development speeds and easier maintenance. The application supports native gesture interactions and looks visually appealing on all specified platforms. In addition to gestures, physical button interactions must function expectedly and uniformly on each device.

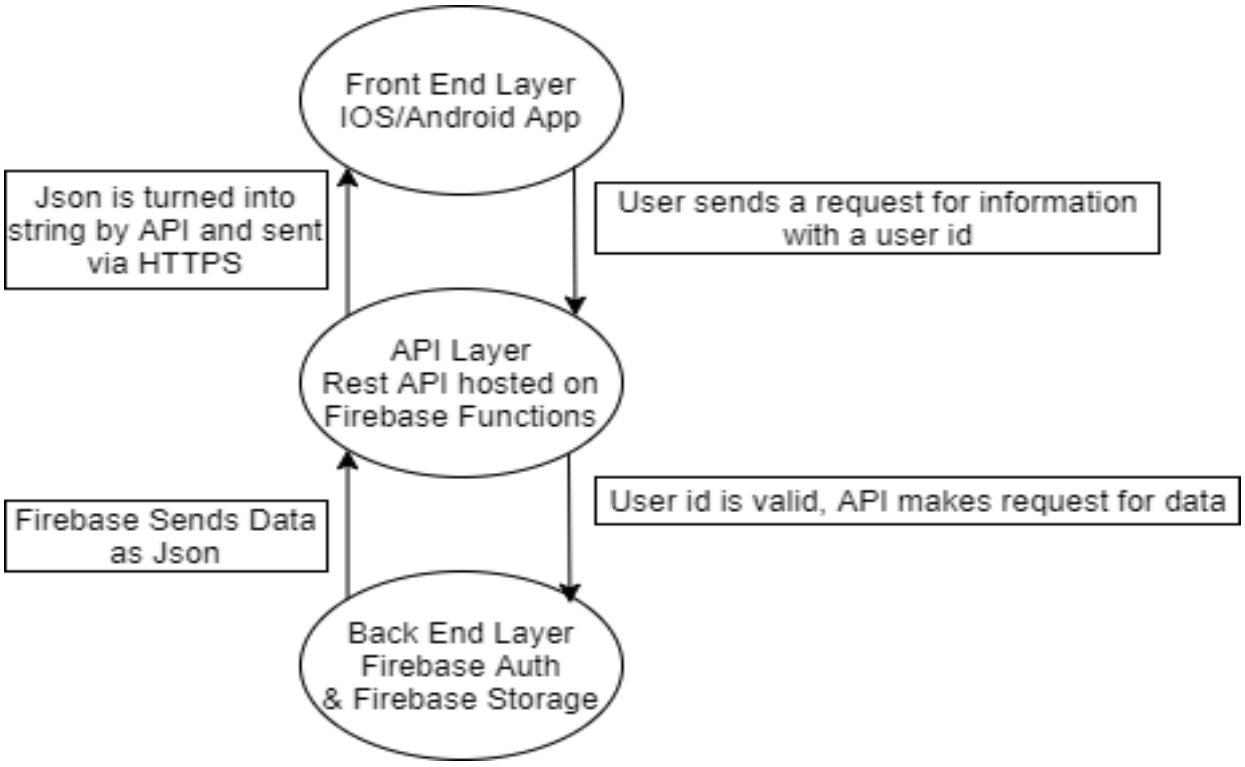
3.3 Software Interfaces

Connections will be made from the mobile client front-end, made with React Native to a firebase back-end. The back-end will contain firebase auth for authentication, and firestore for any other data.
Diagram 3.2

Diagram 3.3


Authentication Stack
Back-End
Front-End
Firebase
Back-End configures settings for Firebase authentication
Firebase Auth server is up and running and sends any configuration errors back to Back-End console
Front-End send user data collected from display to firebase to sign up or log in
Firebase returns data to front end to be added to a user's class to be used
Front-End uses tokens granted by Firebase to check if user is authenticated on any interaction
Firebase responds with whether or not user is allowed to interact
Back-End
Front-End
Firebase

3.4 Communications Interfaces



The requirements for connection to the platform are as follows. The user will need to sign up for the service with an email, username and password, Phone number verification might be utilized if more stringent methods are required down the line, but is not currently implemented. An internet connection and a mobile phone supporting react native will be required. Most messages are sent using the HTTPS protocol, And thus an SSL (Secure Sockets Layer) Certificate will be required. Diagram 3.4

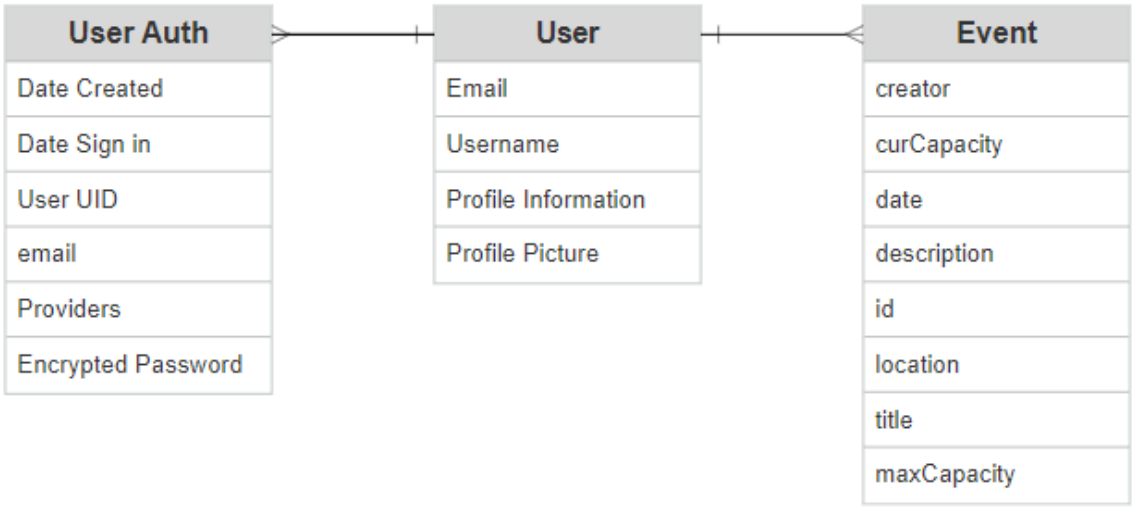


Diagram 3.5

4. System Features

4.1 Timeline

4.1.1 Description and Priority

A module to view current events. Each event will be filtered by a location set by the user, as well as other filters based on the user's preference, eg;(sport, skill, casual/serious, age group, amount of players, etc.)

4.1.2 Stimulus/Response Sequences

Users will be able to access this feature by simply clicking on the tab at the bottom of the app. users will also be able to filter and select events for preview, scroll up and down, swipe to add to watch list, see distance from current location, countdown clock till the event time.

4.1.3 Functional Requirements

REQ-1: Fast created list of events that can be filtered by event attributes
REQ-2: Swipe to add event to watch list
REQ-3: Be able to see a TL of events you have joined
REQ-4: TL for groups, friends

4.2 Groups

4.3.1 Description and Priority

Users should be able to create groups with other users that allow users to see who in the group is doing a certain activity at a specified time and allow them to join the activity easily. The group leader should be able to edit the group and remove users from the group. They

should also be able to create reports on members about things such as who has been participating in the group recently. This should be one of the highest priorities in the project as it is a big feature of the app.

4.3.2 Stimulus/Response Sequences

The groups page will accessed through the account page and it will take the user to a page with all the groups the user has currently joined

4.3.3 Functional Requirements

REQ-1: Users need to be able to create, join, invite, delete (owner only) and leave groups at any time.

REQ-2: Group events should take priority on the main page for all the members apart of the group so that they appear first.

REQ-3: Group members need to be able to chat with each other

REQ-4: Group owners need to be able to create group events that members will have the option of whether to participate in or not.

4.3 Chat

4.3.1 Description and Priority

Each group will have an associated chat room that allows them to talk with other group members in real time, this chat will also allow users to look at other users profile, by pressing on their pfp.

4.3.2 Stimulus/Response Sequences

The chat will be accessible through the chat button in each respective groups front page

4.3.3 Functional Requirements

REQ-1: Users need a text box to enter text

REQ-2: User needs to have a button that allows them to send messages

REQ-3: Messages must be sync to an off device database, that include uploading and download

REQ-4: Users must be able to press another users pfp to access their profile

5. Other Nonfunctional Requirements

5.1 Performance Requirements

The application is able to execute efficiently in an Android mobile environment. Additionally the application runs efficiently and looks visually appealing in a web environment. Searching, sorting and filtering must be efficient. Navigation between screens must be smooth and take little time to load cached and non-cached content. Additionally requests made to the backend must be handled quickly and accurately.

5.2 Safety Requirements

The biggest safety concern is the privacy of the users. The only personal information included should be the username and additionally whatever the user makes public in the form of a bio or public messages. In addition to user privacy, physical safety is a concern. We understand that people use mobile applications for things other than their intended purpose. We aim to limit the ability of one user to stalk, coerce or scam another user by limiting public information on individual users.

5.3 Security Requirements

The users' email, password, and any personal information should never be shared with others on the application. Users' names should only appear for the people that they are in groups with to keep from other unwanted people from obtaining that information. Additionally, user passwords are hashed by Firebase Auth's proprietary modified version of scrypt. This is handled automatically in the firebase backend. In the event databases are leaked, no passwords will be directly exposed due to the encryption.

5.4 Software Quality Attributes

Modularity and expandability are two key components of the software. Our code allows for quick additions and modifications without major reworking of the foundational code. Components have been designed to be used flexibly for multiple purposes. This standard should be used for any future development to maintain consistency and expected behavior across large parts of the app. Components should be designed without this standard iff use of the component in multiple areas of the application would jeopardize the integrity of the applications functionality or any user's private data.

5.5 Business Rules

Developers and administrators reserve the right to remove accounts, events, groups, chat messages, profile content, event content and any reason including but not limited to illegal content, offensive content, targeted abuse, violence, bot accounts.

6. Other Requirements

Additional requirements will be added later in the application's development as they arise.

Appendix A: Glossary

2g2q: the team name, too git to quit, devised by marketing genius and self proclaimed intellectual Ben Marschall
React: Software package used in the development of front-end for apps and websites.
TL: Timeline

Firebase Auth: Authentication software used to safely and securely connect users to their accounts
Firestore: Name of the online hosted database in use for events, groups, and non-authentication related user data

Appendix B: Analysis Models

Diagrams and visual graphics should be added continuously as the application is developed and more information becomes available.

Appendix C: To Be Determined List

1. Introduction
2. Overall Description
3. External Interface Requirements
4. System Features
5. Other Nonfunctional Requirements
6. Other Requirements