



Software Requirements Specifications

Will Harrison
Cody Henderson
Nath Tumlin
Thomas Willingham

March 2, 2017

Change Log

Version	Summary	Date
0.1	Document Created	27 January 2017
0.2	Initial Draft	20 February 2017
0.3	Added introduction, Description, Requirements, Diagrams	01 March 2017
1.0	Added System Architecture, User Interface	02 March 2017

Table of Contents

Change Log	2
Table of Contents	3
1: Introduction	4
1.1: Purpose	4
1.2: Scope	4
1.3: Glossary	4
2: Project Description	6
3: Requirements	7
3.1: Functional Requirements	7
3.2: Non-functional Requirements	7
4: Diagrams	8
4.1: Use Case Diagram	8
4.2: Activity Diagrams	9
4.3: High-level Class Diagram	12
5: System Architecture	13
6. User Interface	15
6.1: Home Screen	15
6.2: User Account Screen	16
6.3: Game Screen	16
6.4: Tournament Screen	16

1: Introduction

1.1: Purpose

The purpose of this document is to define the requirements for Skorch. This document describes the problem Skorch attempts to solve, describes functional and nonfunctional requirements for Skorch, and provides use case diagrams, activity diagrams, and a high level class diagram.

1.2: Scope

Skorch attempts to provide a comprehensive solution for users looking to manage tournaments and keep track of scores of individual games that may or may not be part of a tournament. The status of tournaments and games will update in real time for users following them, and results of games will be saved for historical purposes.

1.3: Glossary

Skorch: the service that allows users to create games and keep track of scores from a variety of devices

Game Center: the place where a user can manage and create games and manage user account settings

Game: a session that allows a user to keep track of scores based on the game type

Game Type: settings imposed on a game that describe the number of players, duration, scoring rules, and winning conditions

Tournament: a set of similar game types set up in a bracket style progression

Game Code: a three- or four-word phrase used to identify a unique session

Public Game Code: the game code that lets any user view a game

Private Game Code: the game code that lets registered users view and control a game

Public Tournament Code: the game code that lets any user view a tournament and all games within it

Private Tournament Code: the game code that lets registered users view and control a tournament and all games within it

GUI Interface: the interface accessible from a desktop or mobile browser that allows users to create, join, query and update a game's status

Voice Interface: the set of voice commands that use the Skorch API to join, query and update a game's status

MeteorJS: a complete platform for building web and mobile apps in pure JavaScript

ReactJS: a declarative, efficient, and flexible JavaScript library for building user interfaces

2: Project Description

Skorch will be a management platform for games and tournaments. Users should be able to easily create a game or tournament, with or without logging in. Games can be quickly created by defining its type, how it is scored, the win conditions, and the players. They should then be able to update the status of the game as it is played. A link should be made available to share a live-updating, view-only page for the individual game. Users will also be able to group multiple games into tournaments with similar management and sharing functionalities. Additionally users will be able to associate games with their accounts for easy access to managing and viewing previous results of games.

Skorch will be built to support multiple platforms, including the web and Amazon Alexa, in an attempt to reach as many users as possible. Potential users extend beyond people looking to manage literal games and tournaments, and extends to any sort of tracking of “points”, such as a teacher keeping track of students’ good behavior or participation.

3: Requirements

3.1: Functional Requirements

- The user will be able to create a single-user game
- The user will be able to create a multi-player game
- The user will be able to save a game's score.
- The user will be able to view a game's score.
- The user will be able to update a game's score.
- The user will be able to join a game.
- The user will be able to create a tournament bracket.

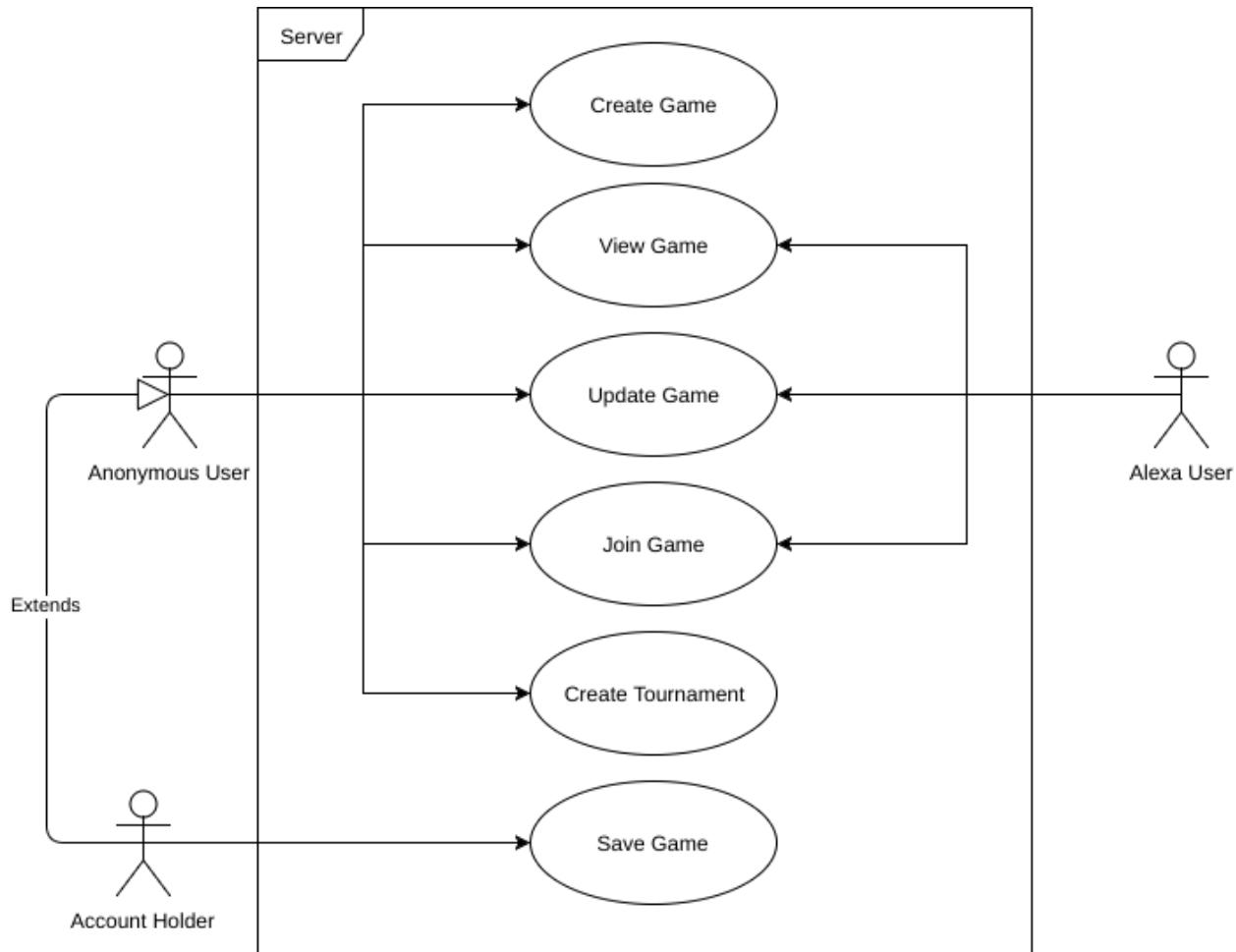
3.2: Non-functional Requirements

- The Alexa must respond to requests within a few seconds.
- The game creation process should be easy, with a user being able to load the website, create a simple game based on provided defaults, and obtain link to share within about 45 seconds.
- Game codes should be easy to memorize and share.
- Skorch should be able to support at least 50 concurrent users on an AWS t2.micro EC2 instance.

4: Diagrams

The following diagrams will aid in the understanding of the Skorch system.

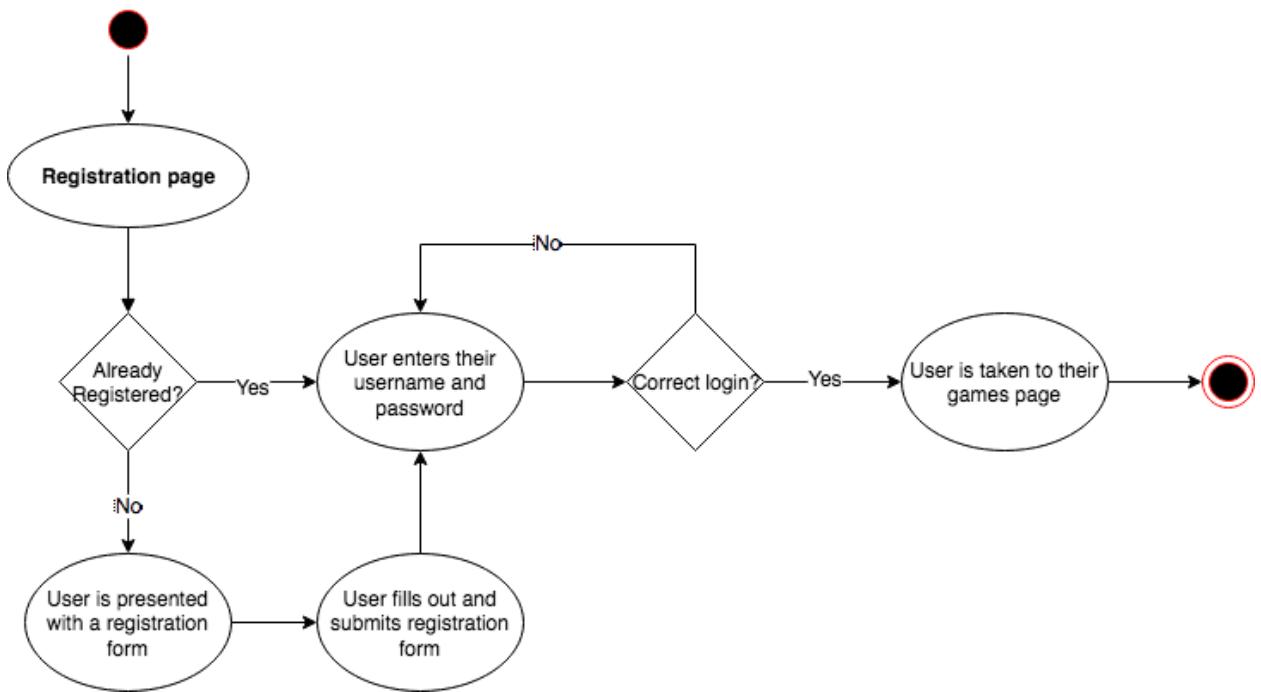
4.1: Use Case Diagram



All users, anonymous and registered, are allowed to create games and tournaments. They are also allowed to join, view and update games based on the settings of the individual games. Registered users, or account holders also have the ability to save games to their account to be referenced again in the future if necessary. Alexa users have a smaller subset of actions; these voice-interface users will be able to join, view and update games based on the game settings, but we do not expect them to be able to create games through the voice interface.

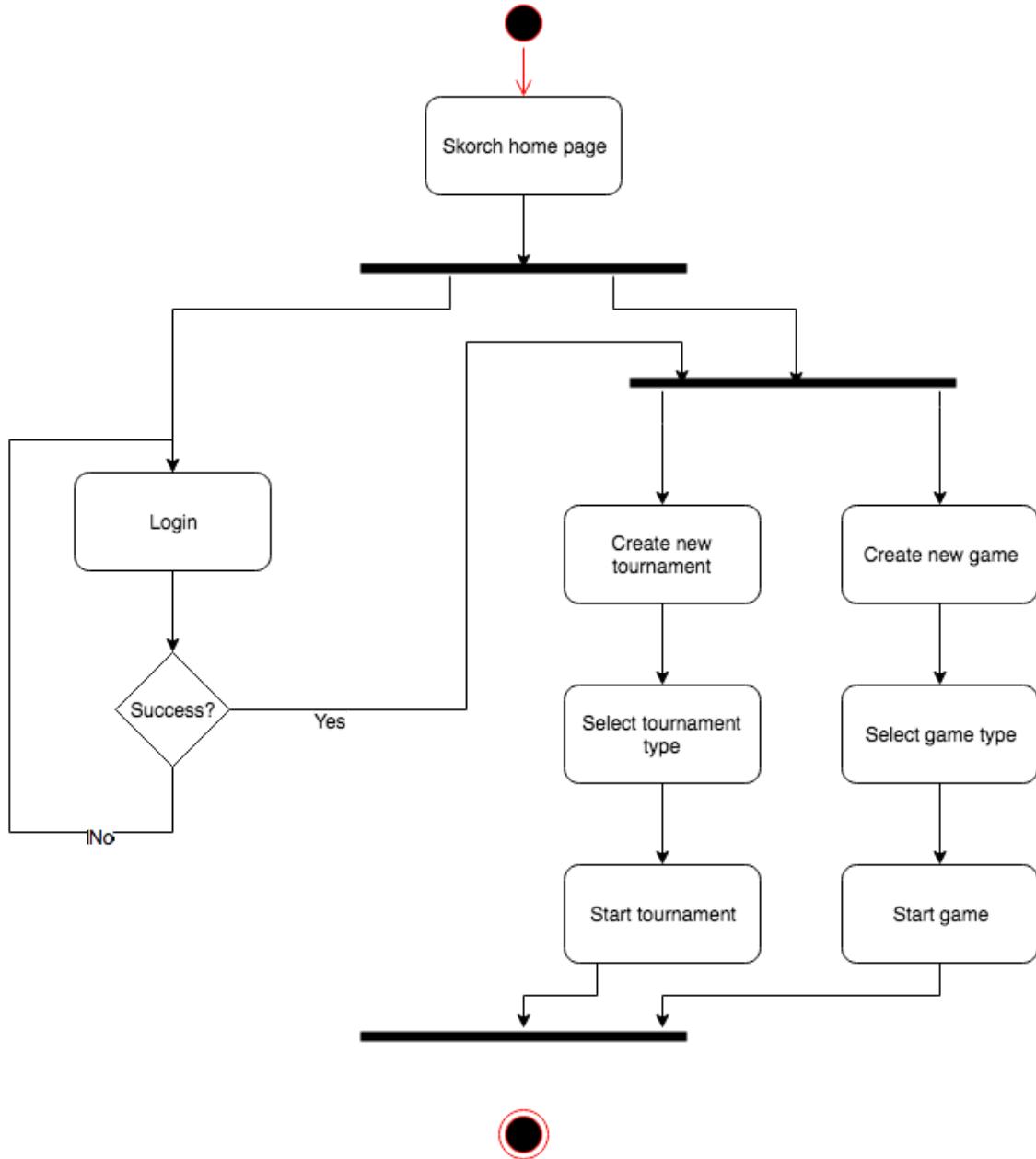
4.2: Activity Diagrams

Skorch: Registration and Login



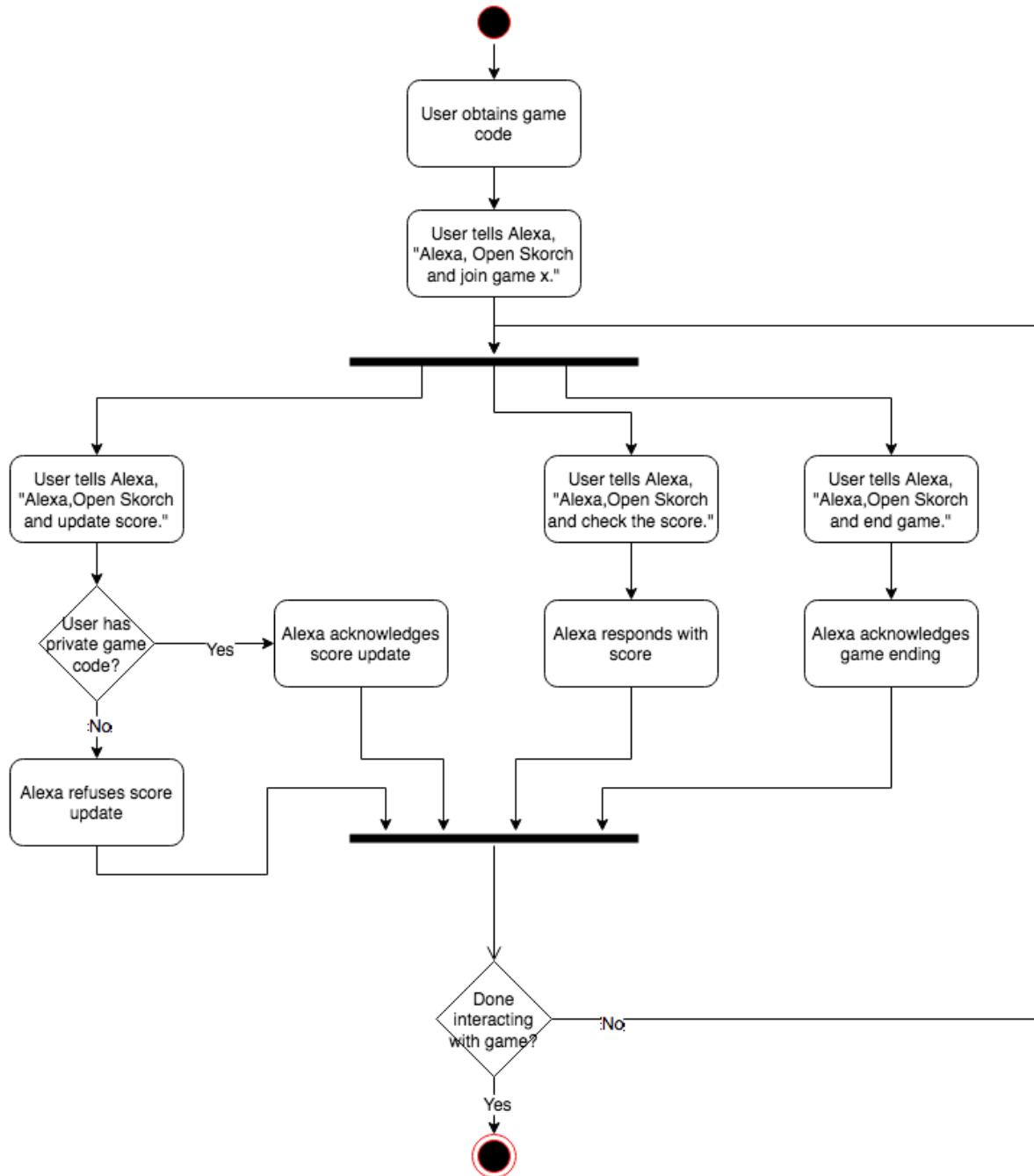
When logging in on a GUI interface, a form allows users to log in if they already have an account. If the user is not already registered, the interface will allow the user to register through a registration form. Once registered, the user can enter their username and password. If the combination matches a user in the system, the user is taken to the game center page.

Skorch: Create New Game or Tournament



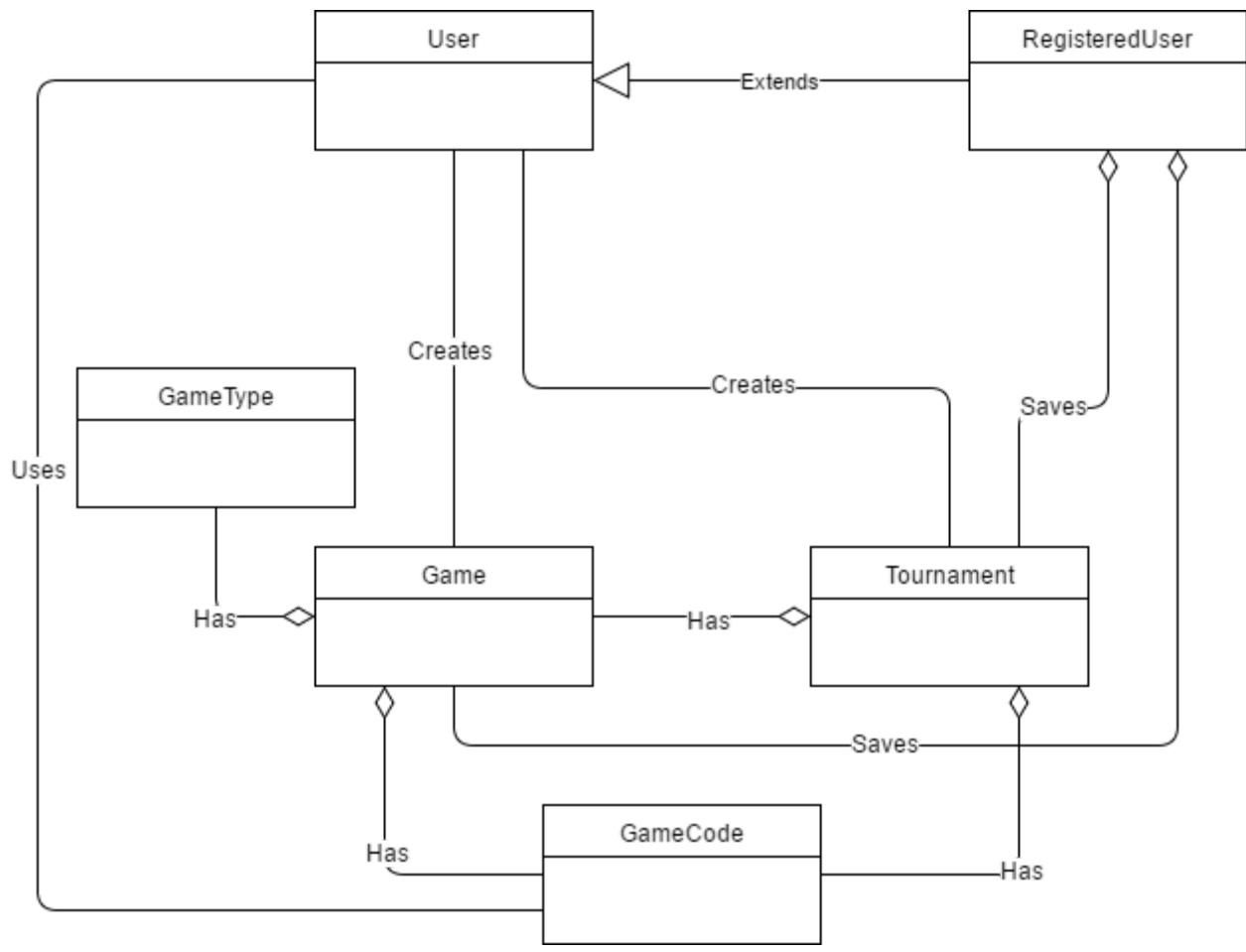
Users can use the GUI interface to create new game sessions. Users have the option of logging in to save game sessions. Any user can create a new tournament from the home page, set the tournament type, and start the tournament. Users can also create a new game, select the type, and start the session.

Skorch: User Connects Amazon Alexa to game



An Alexa user can interact with Skorch using voice commands. Using the public or private game code, the user can tell Alexa to join a game session. Once joined, the user has a few options. The user can update the game score if the game was joined with a private game code. The user can check on the score of the game, and the user can end the game when complete.

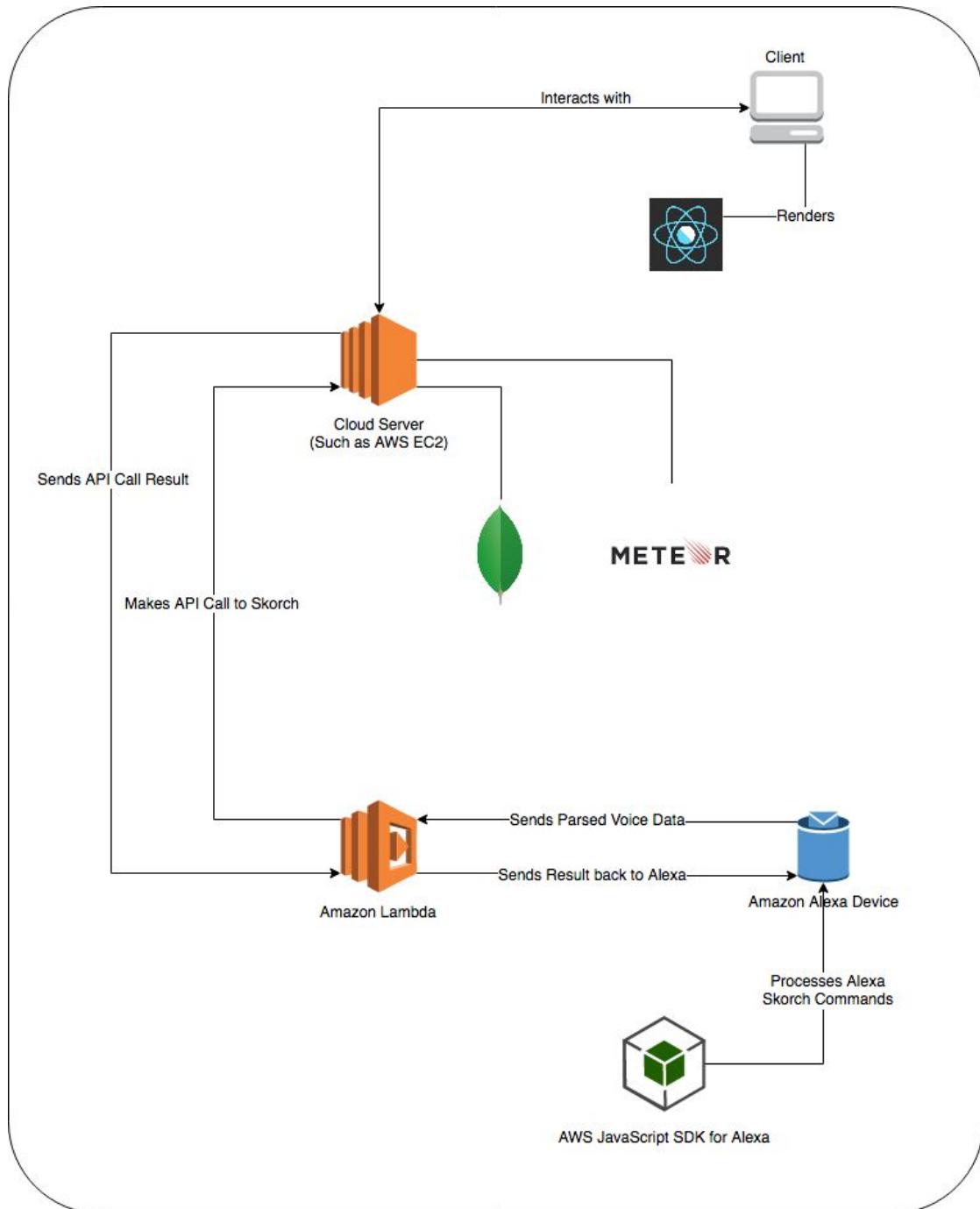
4.3: High-level Class Diagram



High-level Class Diagram

In our system, a User will have the most control in the system. A User may create a Game or Tournament, and a User may use a GameCode to find a Game or Tournament. A RegisteredUser has the same functionality as a User, but can also save Games or Tournaments. A Tournament is made up of one or more Games. A Game has a GameType that stores the settings for the game session. Games and Tournaments also have a GameCode with is used to identify the Game.

5: System Architecture



Skorch will employ a number of different technologies and elements in its system architecture. The primary server will run on an Amazon Web Services (AWS) EC2 t2.Micro

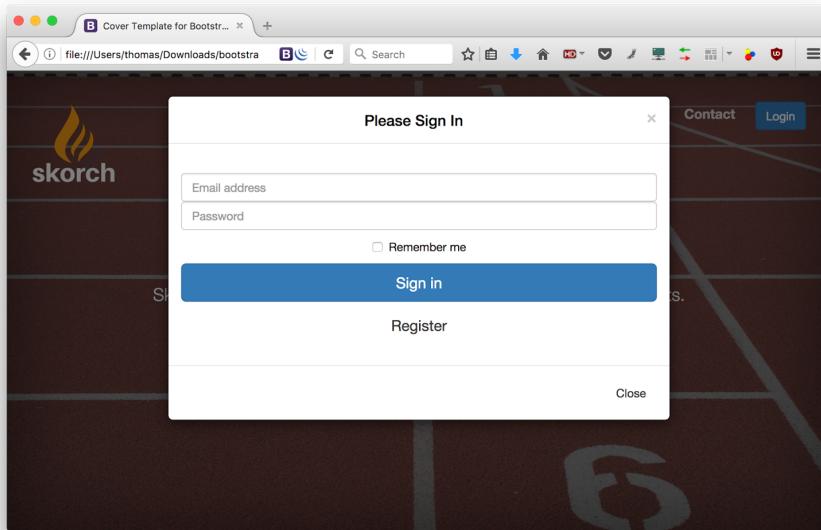
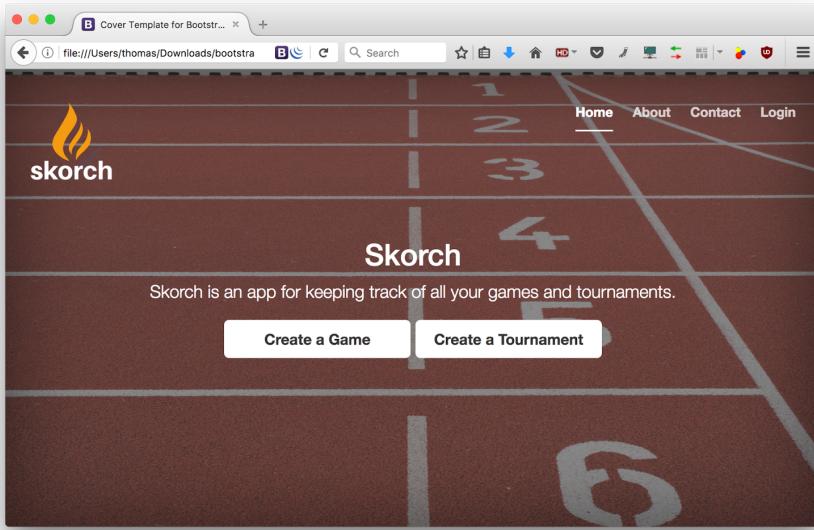
instance. This server will be a linux server that runs NodeJS, MeteorJS, and MongoDB. All of the data storage and core business logic will be performed on this server. All web and mobile clients will render HTML using ReactJS, and they will connect to the primary server directly.

The Amazon Alexa requires a few more pieces to integrate with Skorch. The Amazon Alexa will connect to an Amazon Lambda instance that will host Skorch's AWS Alexa SDK code. All of the parsed voice commands from the Skorch Alexa skill will be sent to the Lambda server, which will in turn determine the proper API calls that need to be made to Skorch. The Lambda server will then make those API calls to Skorch and send the response data back to the Alexa.

6. User Interface

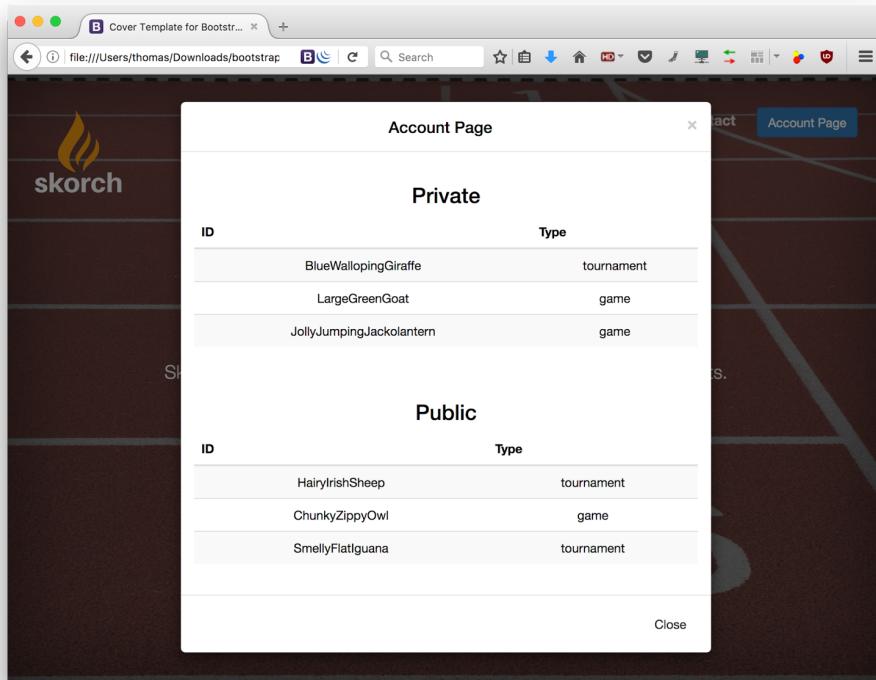
The Skorch application UI will consist of four main components: the home screen, the user account screen, the game screen, and the tournament screen.

6.1: Home Screen



On the home screen, the user will have the option to create a game or tournament, enter game or tournament code, or log in.

6.2: User Account Screen



Once logged in, a user can go to their account page to view a list of their saved games. The user will be presented with the scores of in progress games they have saved, and saved completed games will show the game's winner.

6.3: Game Screen

The game screen will show the scores of the players of the game. If the screen was accessed with a private game code, the user will also have the ability to update the scores of each player and edit the game's configuration such as scoring method, the number of players, and player names.

6.4: Tournament Screen

The tournament screen will show the status of a tournament, including scores and results from its individual games and the progression of players or teams through a bracket. Users will be able to navigate to the game screens of individual games in the tournament. If the tournament screen was accessed with a private tournament code, the user will also have the ability to edit the tournament's configuration such as the number of games played, participating players and teams, elimination style, and seeding.