COMS 309, Fall 2017

Team Info:
RB_B_5:
Michielu Menning - SE(Junior)
Walter Seymour - SE(Junior)
Victor Amupitan - SE(Junior)
Lucas Kern - SE(Junior)

READYSET BALL

Project Description:
This web app was designed to help connect athletes of all levels to play different sports anywhere.

Actors:
Hosts: Hosts games for people in the vicinity to join. Can update, control game features.
Game Members: Can join games.

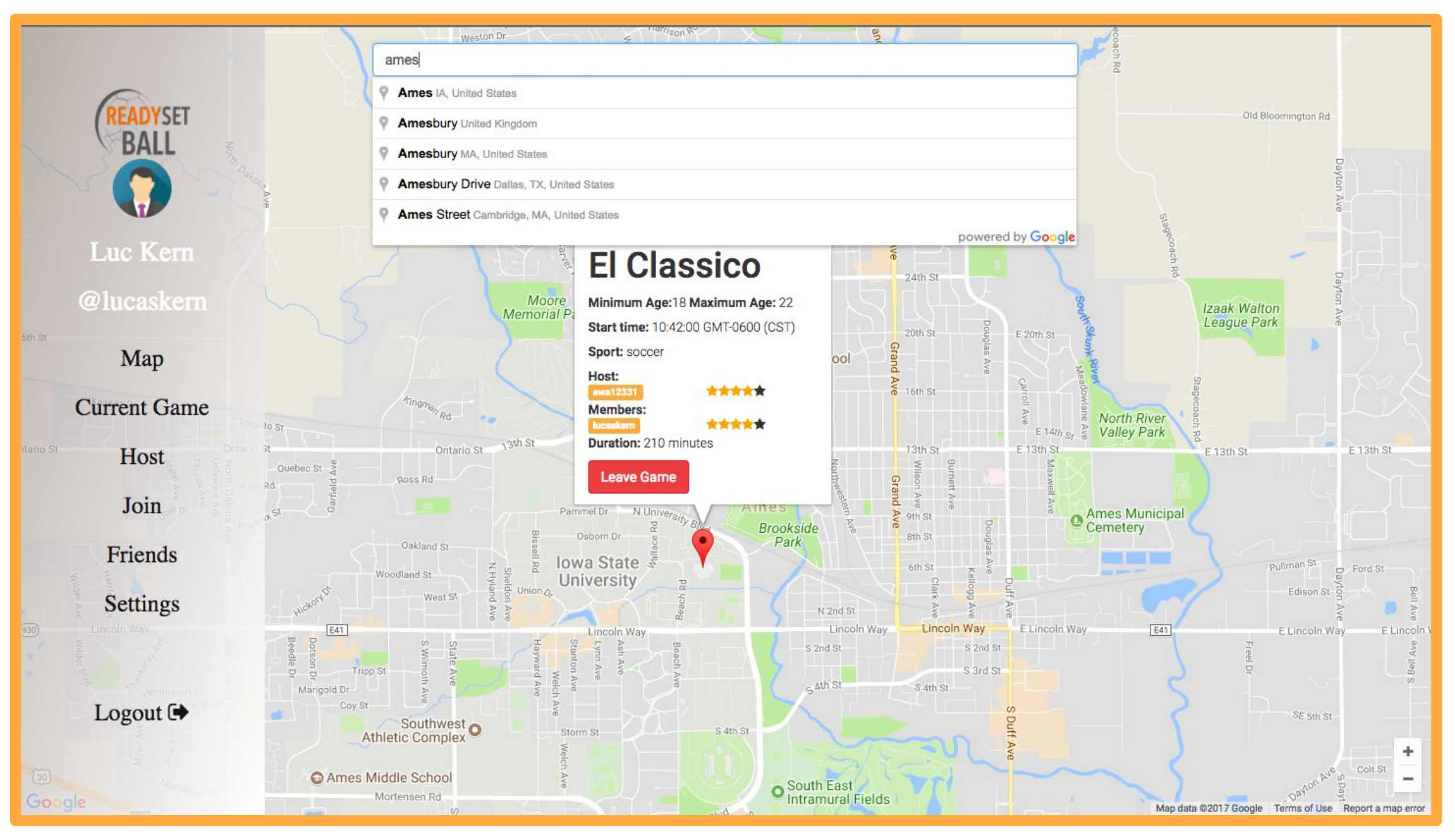
What went wrong
- Using a separate library with Google Maps React integration
- Not enough time to make it mobile
- Some pages rendered slower than desired

What went right
- Server concurrency
- Using websockets to make the app real-time
- Our own implementation of sessions
- Interactive among users
- Worked well as a diverse team

Lessons Learned:
- Modularity
- Programming in Golang
- Dependency Injection
- Asynchronous programming in JavaScript
- Good Test Cases

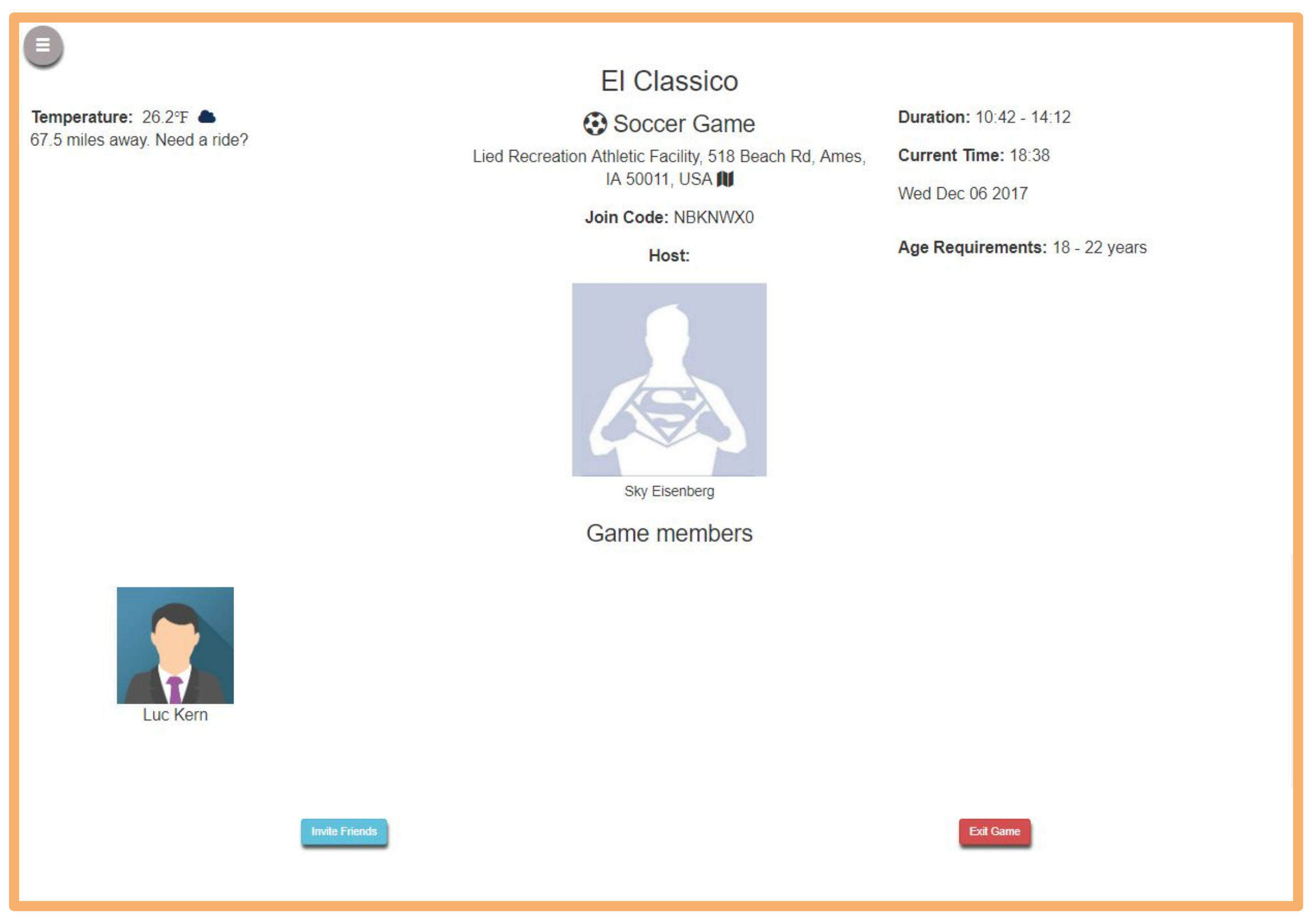
Module Interfaces:
CreateEntity() : handles the creation of client-facing backend models e.g. game, account
Edit() : handles modifying of models based on user-specification
Establish(): handles creation and monitoring of a socket connection
ExitGame() : handles requests of the user to leave their current game
GetGame() : returns the current game of the user or the specified game
GetGamesByLocation(): returns a set of games that are within the specified geographic coordinates
JoinGame() : handles joining a game
GetUser() : handles getting all or some properties of a user
Invites() : handles inviting and reviewing user and game invites
RateGame() : handles rating a game by players who participated
Remove() : handles deletion of friends of a user
UploadAvatar() : handles uploading and management of the user's avatar

Find Games



El Classico
Minimum Age:18 **Maximum Age:** 22
Start time: 10:42:00 GMT-0600 (CST)
Sport: soccer
Host: awa12331
Members: lucaskern
Duration: 210 minutes

Join Games

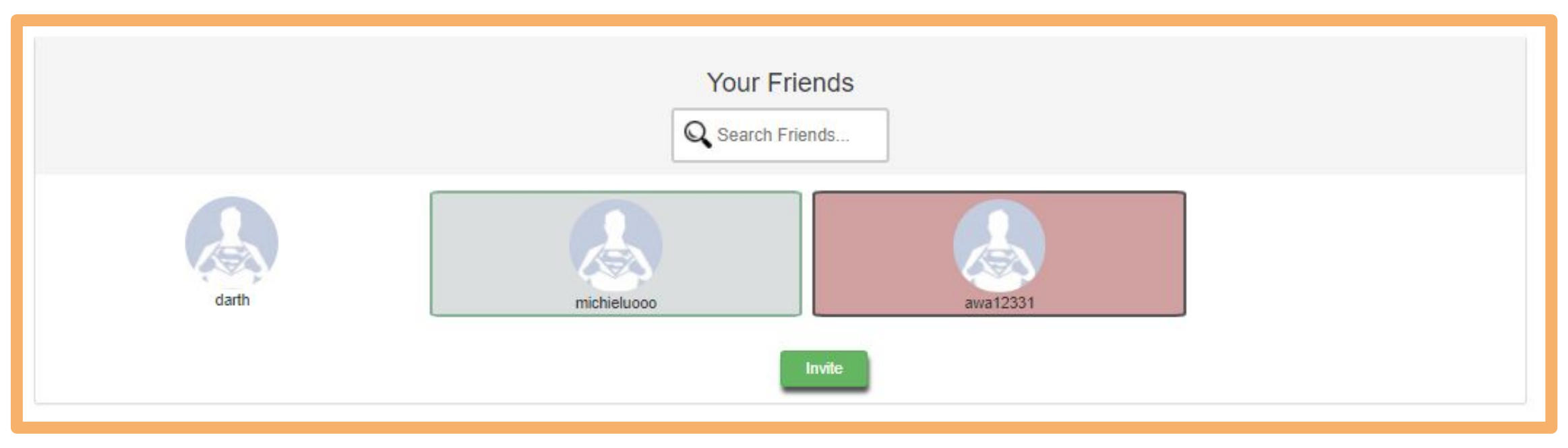


El Classico
Soccer Game
Lied Recreation Athletic Facility, 518 Beach Rd, Ames, IA 50011, USA
Join Code: NBKNWX0
Host:
Sky Eisenberg
Game members
Luc Kern

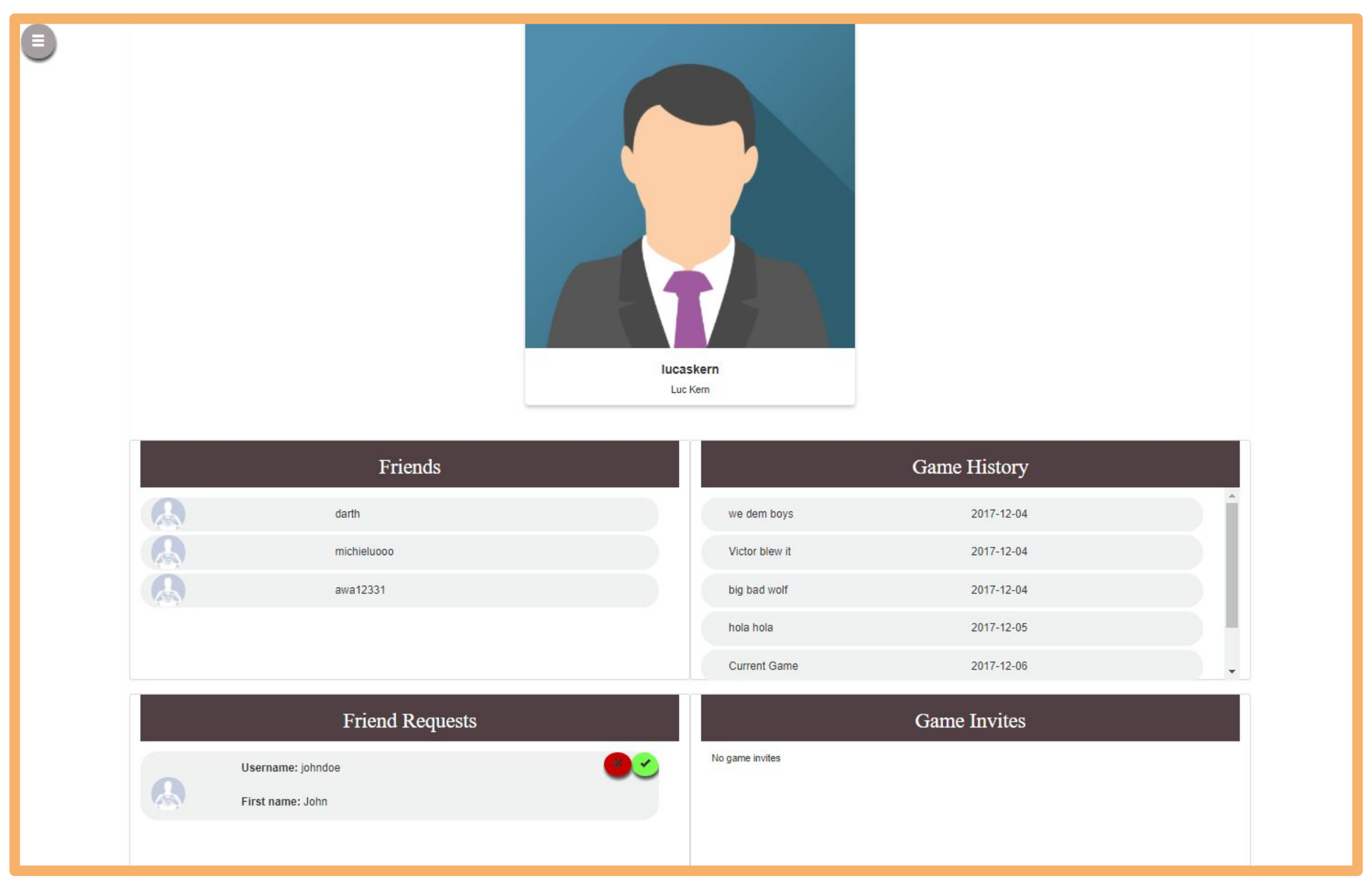
Temperature: 26.2℉
67.5 miles away. Need a ride?

Duration: 10:42 - 14:12
Current Time: 18:38
Wed Dec 06 2017
Age Requirements: 18 - 22 years

Invite Friends



View Profile



lucaskern
Luc Kern

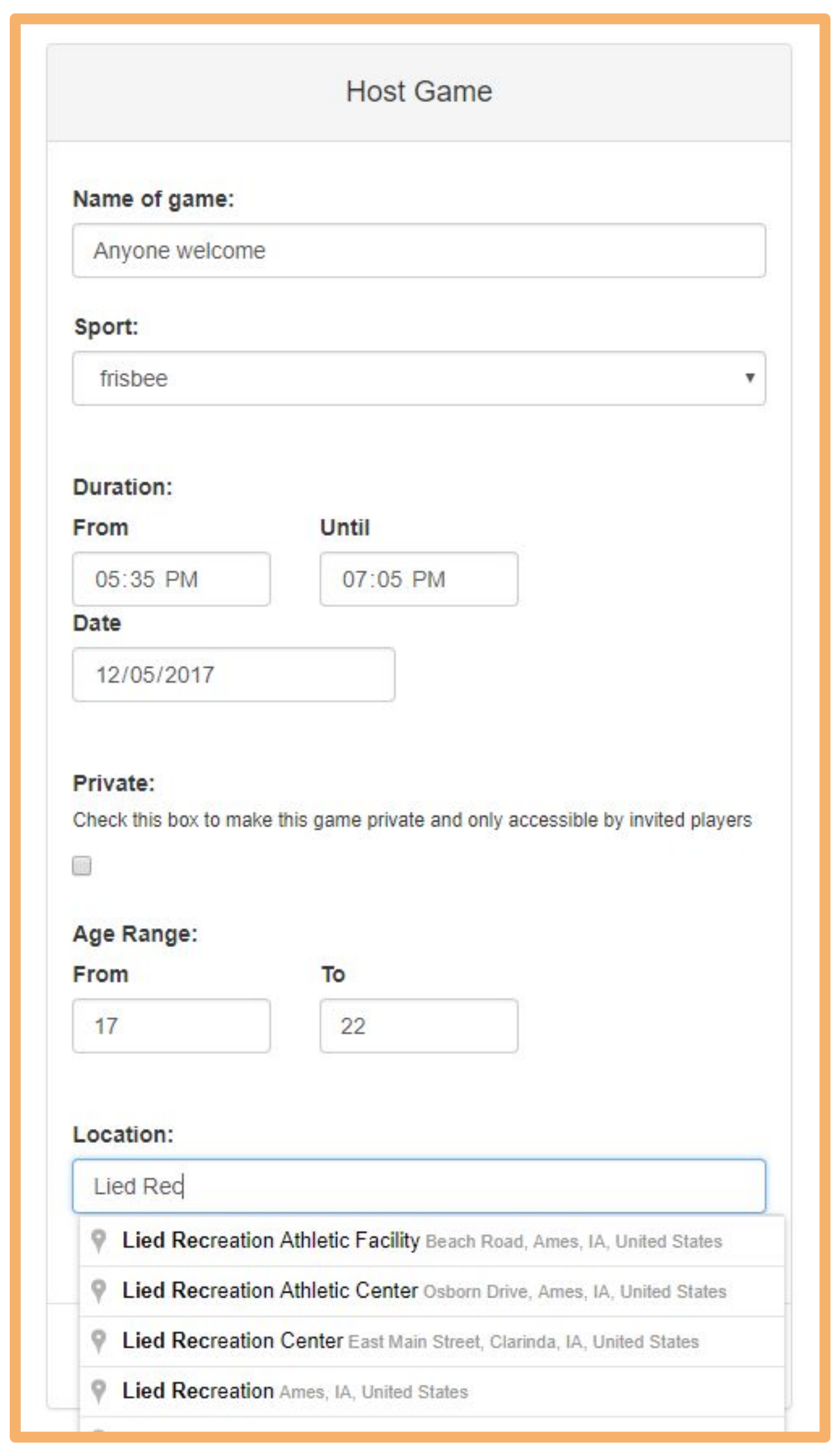
Friends
darth
michieluooo
awa12331

Game History	
we dem boys	2017-12-04
Victor blew it	2017-12-04
big bad wolf	2017-12-04
hola hola	2017-12-05
Current Game	2017-12-06

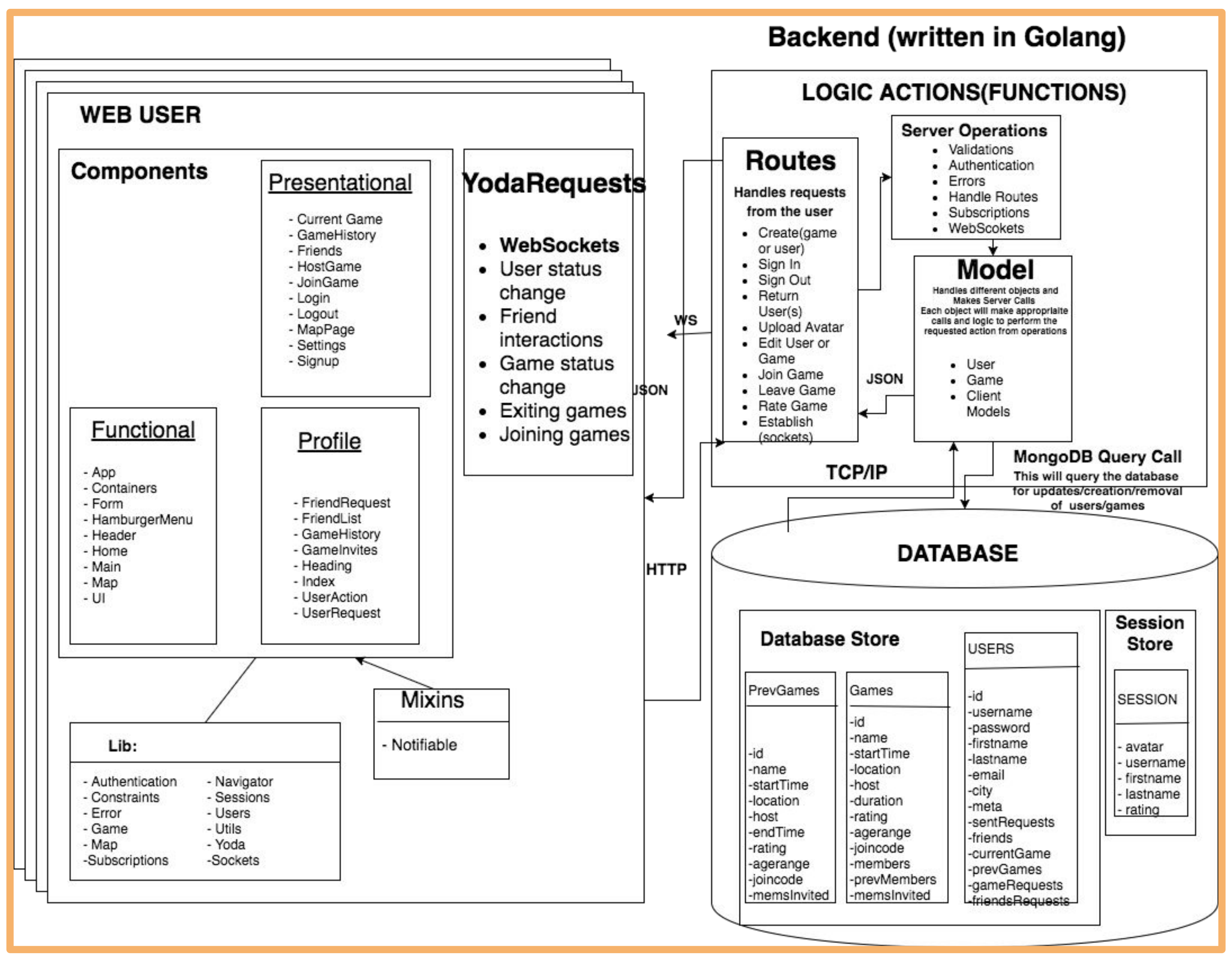
Friend Requests
Username: johndoe
First name: John

Game Invites
No game invites

User Interfaces:
Map Page
Way to view games using Google Maps
Current Game Viewer
Can view location, temperature, distance, and other members of the game. Displays option to invite or leave the game.
Profile Page
Handles game invites, game history, friend requests, and friends.
Design Decision:
- Concurrent system in the backend and asynchronous system on the front end
- Used MongoDB to enable immense scalability in the future and avoid expensive joins. It was difficult because there seemed to be an underlying expectations to use tables, and MongoDB is document-based.
- Using react on the frontend enabled us to have very efficient DOM rendering by only updating new changes in real-time with the server
- Interactive and in real-time



Host Game
Name of game: Anyone welcome
Sport: frisbee
Duration:
From: 05:35 PM
Until: 07:05 PM
Date: 12/05/2017
Private:
Check this box to make this game private and only accessible by invited players
☐
Age Range:
From: 17
To: 22
Location: Lied Rec
Lied Recreation Athletic Facility Beach Road, Ames, IA, United States
Lied Recreation Athletic Center Osborn Drive, Ames, IA, United States
Lied Recreation Center East Main Street, Clarinda, IA, United States
Lied Recreation Ames, IA, United States


Backend (written in Golang)
LOGIC ACTIONS(FUNCTIONS)
WEB USER
Components
Presentational
- Current Game
- GameHistory
- Friends
- HostGame
- JoinGame
- Login
- Logout
- MapPage
- Settings
- Signup
Functional
- App
- Containers
- Form
- HamburgerMenu
- Header
- Home
- Main
- Map
- UI
Profile
- FriendRequest
- FriendList
- GameHistory
- GameInvites
- Heading
- Index
- UserAction
- UserRequest
Lib:
- Authentication
- Constraints
- Error
- Game
- Map
-Subscriptions
- Navigator
- Sessions
- Users
- Utils
- Yoda
-Sockets
Mixins
- Notifiable
YodaRequests
WebSockets
User status change
Friend interactions
Game status change
Exiting games
Joining games
WS
JSON
HTTP
Routes
Handles requests from the user
Create(game or user)
Sign In
Sign Out
Return User(s)
Upload Avatar
Edit User or Game
Join Game
Leave Game
Rate Game
Establish (sockets)
Server Operations
Validations
Authentication
Errors
Handle Routes
Subscriptions
WebScokets
Model
Handles different objects and Makes Server Calls
Each object will make appropriate calls and logic to perform the requested action from operations
User
Game
Client
Models
JSON
TCP/IP
MongoDB Query Call
This will query the database for updates/creation/removal of users/games
DATABASE
Database Store
PrevGames
-id
-name
-startTime
-location
-host
-endTime
-rating
-agerange
-joincode
-memsInvited
Games
-id
-name
-startTime
-location
-host
-duration
-rating
-agerange
-joincode
-members
-prevMembers
-memsInvited
USERS
-id
-username
-password
-firstname
-lastname
-email
-city
-meta
-sentRequests
-friends
-currentGame
-prevGames
-gameRequests
-friendsRequests
Session Store
SESSION
- avatar
- username
- firstname
- lastname
- rating