

## Team Working Agreement

### Basic format:

- Wrap all React components
- If possible, put functions in separate files
- Comment all relevant functions

### Criteria for work being done:

- Relevant task is completed
- Passes all backend tests in tests files
- All UI elements react properly
- Server does not crash. Database does not crash
- Code adheres to style standards
- Code can be merged to github
- Code can be pushed to heroku

### Database Format:

- Username: PickupCS115
- Password: PickUp115CamelCase
- Database name = pickup
- Mlab url=mongodb://pickup:[cs115@ds251819.mlab.com](https://cs115@ds251819.mlab.com):51819/pickup
- Collections:{games, users}
- users:
  - username: "string"
  - email: "string"
  - games: [array]
  - password : "string"
  - bio: "string"
  - pic: "string"
  - friends: [array]
  - feed: [array]
- Games:
  - sport: "string"
  - name: "string"
  - location: "string"
  - isprivate: boolean
  - coords: { type: "Point", coordinates: [lng(Number), lat(Number)] }
  - id: number
  - owner: "string"
  - startTime: #
  - endTime: # // using Date as timer
  - players: [ "strings" ] // usernames of players
- teams:
  - name: "string"
  - captain: "string" // must be a valid username

- sport: "string"
- members: [ "strings" ] // usernames of players
- city: "string"
- games: [ { teams: [ "strings" ], winner: "strings" } ] // change teams to opponent?
- teamgames:
  - sport: "string"
  - name: "string"
  - location: "string"
  - isprivate: boolean
  - coords: { lat: number, lng: number }
  - id: number
  - captain:"string"
  - teams: [{captain,name}] // usernames of players

### **Coding Style:**

#### Functions

1. Functions are commented
2. Everything in the functions is bracketed and tabbed
3. Use array functions and anonymous functions when able

#### Bracketing

1. All control statements and wrapped structures and in brackets and indent with tab

#### Indentation

1. All indentation is a 1 space tab
2. Use indentation on bracketing

#### Endpoints

1. All endpoint names have to reference the function of the post/get/patch/delete request
2. Use the proper post/get/patch/delete request for each endpoint

#### Naming

1. All names are camelCase
2. Names must properly reflect the referenced object or variable

#### Modeling

1. All relevant database objects must be encapsulated by a mongoose model
2. All mongoose models must be delivered with ES6 bracket importing

#### Whitespace

1. Vertical white spaces between functions
2. Horizontal white spaces only between words

#### CSS

1. Don't add CSS to tags
2. Have unique non conflicting CSS class names

#### Members:

Khachatur Mirijanyan: Product Owner

Rithu Simha: Dev/S1 Scrum Master

Shini Zhang: Dev/S2 Scrum Master

Alex Blumer: Dev/S3 Scrum Master  
Rogelio Molina: Dev/S4 Scrum Master