**App Name:** ???

**Concept:** Score keeping Web application that can be used by the score keeper at junior / minor league hockey games to record game / player statistics.

**Scope of work**

**Schema:**

player: player id, first name, last name, jersey number, team id

teams: team id, team name

games: game id, home team id, visiting team id, game date, game time, address

goals: goal id, player id, game id

penalties: penalty id, player id, game id

**Seeds:**

Will need someone to populate mass data.

Use NPM “Faker”.

**Relationships:**

1 player belongs to 1 team

1 team can have many players

1 scoresheet per 1 game

**Minimum Viable Product (MVP) / Scope**

**Input:**

A simple scorekeeper app: Update scoresheet, timer (count down for game, start, stop, modify the time)

Same jersey can exist on home team and visitor side.

Build 2 sides;

Left side of the screen red button for home team with caption “GOAL”.

Then jerseys show up see goal button below.

Right side of the screen green button visitor with caption “GOAL”.

Then jerseys show up see goal button below.

Left side of the screen red button for home team with caption “PENALTY”.

Then jerseys show up see goal button below.

Right side of the screen green button visitor with caption “PENALTY”.

Then jerseys show up see goal button below.

Middle side of the screen green button visitor with caption “PENALTY”.

Then jerseys show up see goal button below.

Stop / pause button:

Goal button: input jersey for goal, jerseys for first assist (optional), jerseys for second assist (optional), Does not stop the clock. Clock is independent.

Penalty button: input jersey, reason penalty (optional), length of penalty.

Multiple penalty timers - Neil

Team management: Add players, Add teams, Add games

**Output / Routes required:**

Scoreboard per game (mandatory)

Player stats by player (optional)

Player stats per game (optional)

Teams information (optional)

Games information (optional)

**Process:**

Authentication, keep username and password, generate token / string, append to post (optional)

**Preference for team work: (**To be discussed on Wednesday, January 30th)

API routes, HTML routes, env (hiding password), coding standards

Models: db

Views: Front end

Controller: Back end js / routes

Gurneet: Front end, Schema

Chris: Back end, Sequelize ORM,

Neil: Back end, Sequelize ORM

Action items:

Linting - Pending

Branches - Done