Project Ångström

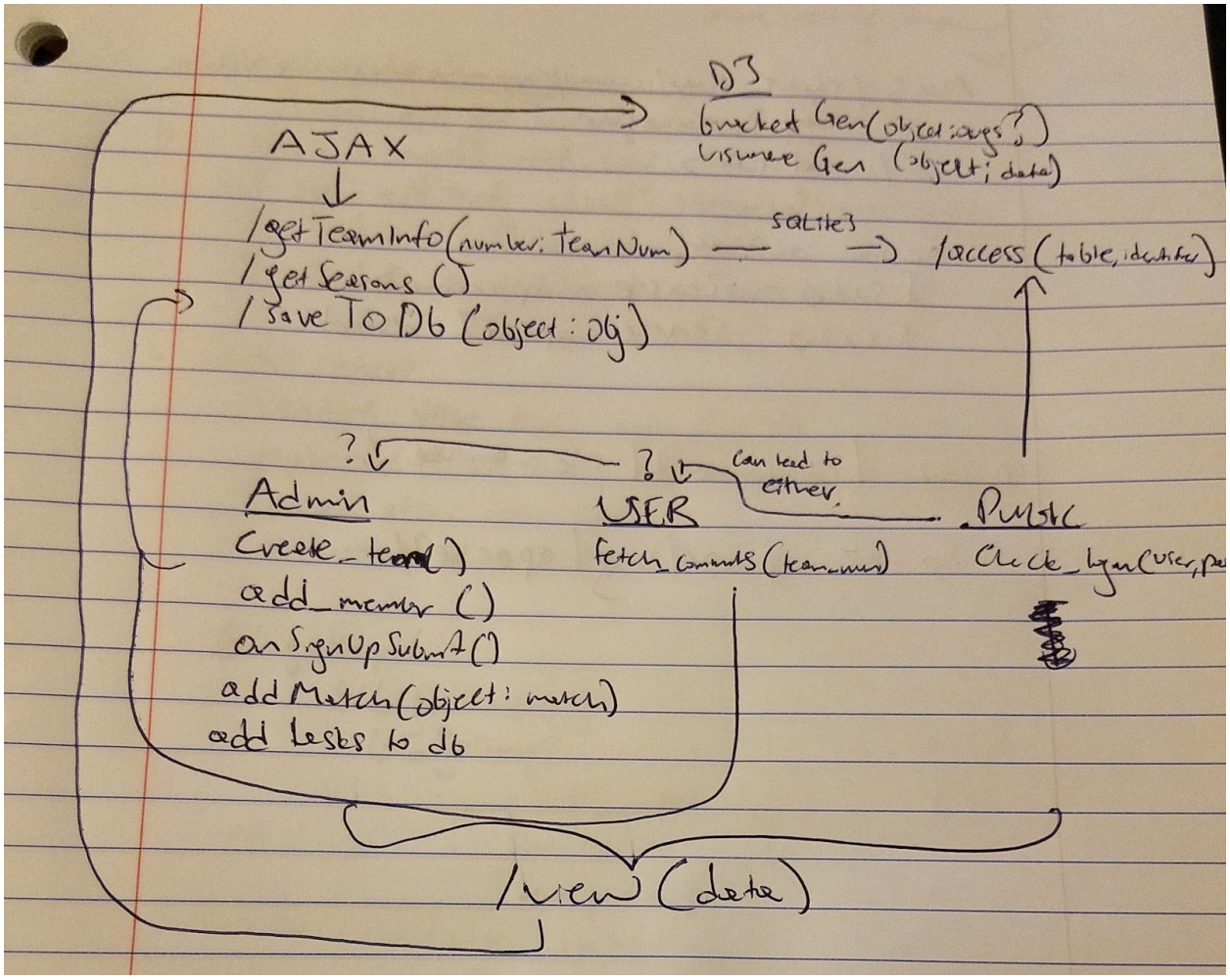
FIRST Tech Challenge Scouting Website

Atomic Theory - Khinshan Khan (project manager), Stanley Lin, Ish Mahdi, Ryan Siu
AtomicTheory_khanK_linS_mahdiI_siuR

Summary

The aim of this website is to provide greater insight on teams participating in FIRST Tech Challenge competitions and to organize the data collected on these teams. Users will input match data while watching a match (scouting), and this data will be stored in a database. We will perform data analysis on match data to determine the best teams at the competition, in different game-specific categories, and this analysis will be displayed in the form of charts, graphs, tables. Each team at the competition will also have a profile page, with basic identifying information (picture of robot, location, year last reached Worlds, etc.).

Component Map



Components

- Python (Flask) side
  - ADMIN
    - create_team()
      - Pulls data from API and generates data (pics, name)
      - Adds other fields, like robot picture, team size based on user input
    - add_member()
      - Generate 0-permission accounts for scouters to add match data
  - USER
    - fetch_comments(team_num)
      - Gets comments made on a particular team
  - PUBLIC
    - check_login(user, pass)
      - Validates user login
  - Server
    - add_tasks_to_db(dict)
      - Parses and adds the dict received from addTasks() call to the db
    - add_prescout_to_db(dict)
      - Parses and adds the dict received from addPreScout() call to the db
- JS side
  - Forms
    - onSignUpSubmit()
      - Obtains form data from team sign up, such as robot picture
    - addMatch(object: match)
      - A json object as follows
        -
```
{
        teamNum: <number>,
        matchNum: <number>,
        tasks: [
              {
                    taskId: <number>,
                    count: <number>
              },
              …
        ],
        note: <string>
}
```
      - Send the object to the flask server
    - addPreScout(object: data)

- A json object as follows
  - {

```
teamNum: <number>,
auton: <number>,
teleop: <number>,
endgame: <number>,
note: <string>
```

}

- Send the object to the flask server
- D3
  - bracketGen(object: args?)
    - Generates a competition bracket
  - visualizationGen(object: data)
    - Uses the provided object to generate a visualization of the data
- AJAX
  - getTeamInfo(number: teamNum)
    - Uses the `/apiv2/team/<teamNum>` endpoint in the TOA api
  - getSeasons()
    - Returns all season ids, uses the `/apiv2/seasons` endpoint
  - saveToDb(object: obj)
    - Sends the json formatted object `obj` to the flask server

- SQLite
  - Database that we will be using to store information
  - Tables on:
    - Team (Basic Info)
    - Match performances on each team
    - Scouting table
    - User information table
- Front-end framework: Materialize
  - We are using Materialize instead of Bootstrap/Foundation because it looks better and less time is needed to set it up.
  - Documentation: https://materializecss.com/about.html

Site Map

blue: public, green: user (permission 0), red: admin (permission 1)

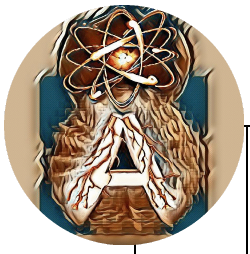
DB Schema

Team

Number (unique ID)	Name	Location	Robot picture	# of Members	Last reached Worlds
310	“Stuy Fission”	“New York”	“abc.jpg”	12	2000

Match Performance

Team #	Match #	User ID	Score 1	Score 2	Score 3	Score 4	Score 5	Score 6	Score 7	Notes



310	4	3	68	9	62	83	9	71	22	“yay”

Pre-scout

Team #	Auton Prediction	Teleop Prediction	Endgame Prediction	Notes
310	48	20	68	“not very good”

User

ID (unique)	Name	Password (hashed)	Permission
3	Ryan	abc123	1

Permission: 0 (user that can input data), 1 (admin--manually added into the database)

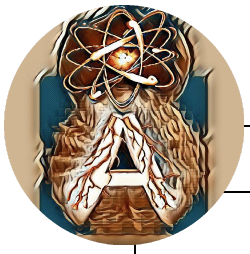
APIs

The Orange Alliance (https://theorangealliance.org/apidocs)

- Has identifying information about teams such as location, name, team number
- Has information about teams’ performances at previous competitions

Tasks

Task	**Assigned To**	**Deadline**
Project setup	Khinshan	5/18
Base template	Ryan, Khinshan	5/18
Database creation + methods	Stanley	5/18
Homepage + login views	Khinshan, Ish	5/20
Data input + visualization views (including generating stats)	Ryan	5/22
Search + team profile view	Khinshan	5/23
Admin views	Stanley, Ish	5/24



Pre-scout views	Stanley, Ish	5/25
Playoff bracket	Stanley	5/30
Improve data visualizations (more/better statistics, graphs, charts, etc)	Ryan	5/30
About Us view	Khinshan, Ish	5/30
Stretch	N/A	...

- User groups
    - Standard team member
    - Team lead
        - Manage competitions (specify the teams at a comp, specify number of matches)
        - Import data on teams from Orange Alliance
        - Add accounts for people
    - Tournament Organizer ??
- Team information entry form
    - Team size, robot picture
    - Enter notes on other teams
    - Pre-scouting information
- Match data entry form
    - Input the number of times a team accomplishes something in a match
- Data visualization (D3)
    - Bracket generation
- Number crunching (generate statistics)
    - Factors in match data + pre-scouting data + orange alliance data (if there is)
    - Projected wins
    - Probability of completing certain actions
- Website design

Stretch:

- Automatic prize distribution (for tournament organizers)

Timeline

- More utility focused (individual, local team usage)
- More broad, multiple teams can use the site, adding in tourney organizer
- Additional functionality for team organizers