Texas A&M Disc Golf Club Website

**Team Royal Flush**

Andrew Cramer

Alok Chandrawal

Chandler Ochs

Prathiksha Prasad

Snehasish Sen

Vignesh Babu Manjunath Gandudi

**Team Roles**

Scrum Master: Prathiksha Prasad

Product Owner: Chandler Ochs

**Github Repo**

<https://github.com/acramer/tamu-disc-golf>

**Pivotal Tracker**

<https://www.pivotaltracker.com/n/projects/2467338>

**Heroku Deployment**

<http://tamudiscgolf.herokuapp.com>

**User Stories Implemented**

1. Make 'Lost & Found' section dynamic

As a club organizer

I want the competitive team information to be dynamic

So that I can easily add and modify that information

1. Add Lost and Found Item Adding to Lost and Found Page

As a club member

I want to be able to add items that I have found to the Lost and Found Page

So that I can inform other members about their lost items

1. Make 'Upcoming Events' section dynamic

As a club organizer

I want the upcoming events information to be dynamic

So that I can easily add and modify that information

1. Make 'Competitive Teams' section dynamic

Why

As a club organizer

I want the competitive teams section to be dynamic

So that I can easily add and modify that information

1. Make 'Officers' Section Dynamic

Why

As a club organizer

I want the officers section to be dynamic

So that I can easily add and modify that information

1. DB Migration Setup - *Work in progress. This will be deployed in the next iteration.*

As a technical lead

I want automatic DB migrations to be set up along with development databases on local

So that DB changes made by developers are managed systematically without conflicts

1. Login permissions - *This story depends on DB Migration Setup story.*

As a technical lead

I want appropriate permissions to be set up for different modules

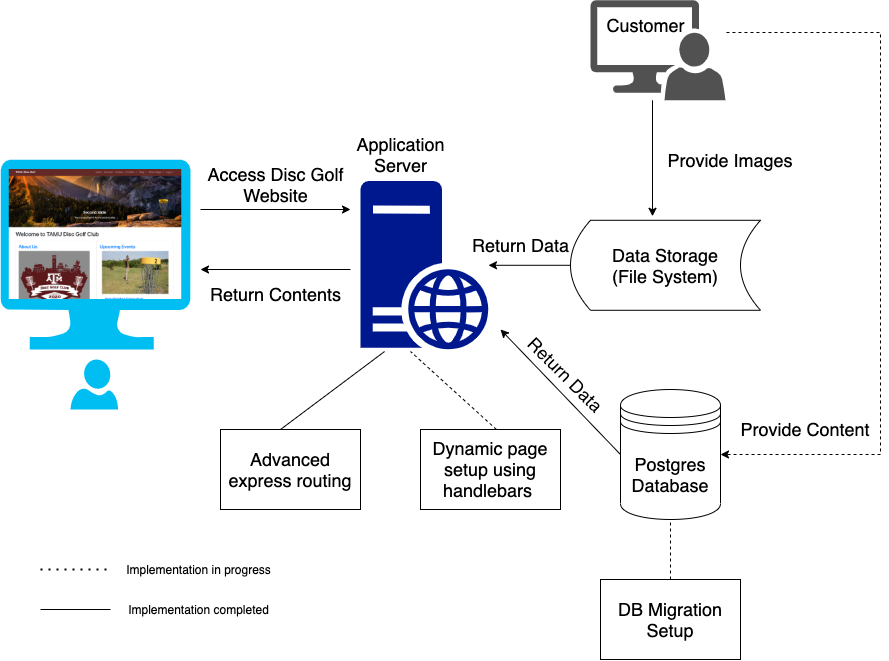
So that only authorized members of the club can update the contents of different modules

**User Stories in Progress**

1. **DB Migration Setup**: As a part of this task we have enabled database migrations to incrementally update the database in a consistent manner. We have used the node sequelize package to achieve this. It also enables us to populate some initial data to the db.  
   We have also included the express-handlebars package which will allow us to write html code in a hierarchical fashion. This enables us to easily build html elements with for loops and if conditions. This also helps in populating entries from the database dynamically. The number of elements(cards, divs, etc.) now will dynamically change according to the number of entries available in the database.  
   The only missing piece here is auto-setup of the database on heroku. We are currently working on achieving a smooth deployment on heroku every-time some db changes are done or new data needs to be pushed to the database.
2. **Login permissions**: On the DB Migration branch, we have created a ‘users’ table that stores the name, email, password and internal id. The id is automatically generated by postgres, and both id and email are unique values in the database. The password that is stored is first encrypted, and the encryption, decryption and verification are handled by bcrypt. To display that the login system works, we have created a login, registration and profile page. The login and registration page work as expected, and the registration page takes on the additional restrictions in password length and password confirmation. We have also set up sessions with express-session and created authorization functions that redirect the user to the profile page when a logged in user navigates to the login page, and vice versa for users that have not yet logged in. For our next steps we plan to implement (1) error notifications such as “password” and “confirm password” forms not matching (2) password reset (3) member roles (4) custom role authorization for competitive team creation.

**Design Diagram**

Given below is the design diagram based on the elements implemented in this iteration. Several components were made dynamic in this iteration by including database connections and queries. Some components are in progress and are indicated by dotted lines. The components that have been fully implemented are indicated in solid lines.



**Unit Testing and Code Coverage (need to update this with latest build)**

* Test cases: 39 passing (1s)
* Code coverage: % Stmts = 83.33, % Branch = 70, % Funcs = 66.67, % Lines = 83.1
* Unit testing using mocha and javascript.
* Enabled continuous integration on the GitHub repository.
* To view the latest unit testing results and code coverage:
  1. Navigate to <https://github.com/acramer/tamu-disc-golf/actions>
  2. Click on latest push to master
  3. Click build, then expand the line ‘npm test’
* To run the tests on your local:
  1. Clone the repository
  2. cd into tamu-disc-golf
  3. Run the command: $ npm test

**Customer Feedback**

We deployed the changes made in this iteration to heroku and requested the customer for feedback. The customer was excited to see several elements working dynamically with database connections. The customer appreciated all the efforts put by the team to add new features to the website.

We are still waiting for more information to be provided by the customer about Officers and Competitive events. The details will be enhanced in the future iterations when we get an update from the customer.