

TourneyTrack

6.170 Final Teamwork Plan Document

Team Members: Jamar Brooks, Yachi Chang, Chris Rogers, Katharine Xiao

Stakeholders

- Users
 - MIT Affiliates
 - Intramural Leagues
 - Dorm Tournaments
 - Informal tournaments (ping pong, video games, etc.) amongst friends
- Advertisers* (Not in MVP or Final, but could be a stakeholder if app proceeds past class)
 - Sports Apparel Vendors
 - Sports Television Channels

Resources

- MIT People API - we are relying on this external resource to provide us with a list of all current and valid MIT Kerberos usernames so that we can use email validation to confirm the identity (and MIT status) of each user.

Tasks (Mohs' Hardness Scale 1-10)

- User Authentication - **3** (Judy)
 - Kerberos validation
 - Email confirmation
- Creating Mongoose Schemas - **3**
 - translate Data Model (everyone)
- Implementing API methods to update model (API Post/Put/Delete Methods) - **8**
 - User (Judy)
 - Match (Chris)
 - Bracket (Judy + Jamar)
 - Tournament + Team (Katharine)
- Implementing Model Access API methods (API Get Methods) - **6**
 - User (Judy)
 - Match (Chris)
 - Bracket (Judy + Jamar)
 - Tournament + Team (Katharine)
- Testing for API methods - **4** (everyone)
 - each person will test their own methods
- Tournament/Team/Player-Specific Interfaces (User Interfaces) to display data and respond to user interactions - **8**
 - Tournament Home Page (w/ Bracket display) (Jamar)
 - Tournament Administration Page (Jamar + Judy)

- **Tournament Modal Window** (for result approval) (Jamar)
- **All Tournaments Page** (Katharine)
- **New Tournament** (Katharine)
- **Match View** (Chris)
- **Match Modal Window** (Chris)
- **Sign Up/Login** (Judy)
- **User Profile Page** (Judy)
- **Team Profile Page** (Judy + Katharine)
- **Team Administration Page** (Judy + Katharine)

Finish by 11/10

Finish by 11/13

Finish by 11/16

Risks

- **Design Risks**
 - Update latency depends on how attentive admins are
 - **Reminder system that notifies admins when new information is submitted**
 - Players must be trusted to submit accurate match outcome data
 - **Require admin approval**
- **Implementation Risks**
 - Handling multiple outcome submissions from different sources
 - **Provide a list of all submissions, which admins can look through and choose from**
- **Development Risks**
 - Aggregating Tournament and Team Data based on which Players have participated may be difficult to perform using MongoDB
 - **Use AngularJS \$http service to make multiple queries to the API and have UI be filled in as data is returned with AngularJS data-binding**

Minimum Viable Product

- Concepts included: User, Team, Match, Outcome, Bracket, Tournament
- Users should be able to create a persistent account on the system
- Users should be able to create tournaments, create a team which joins a tournament, and join a team
- Users should be able to report the outcome of a match, have it verified by the administrator of the tournament and see who their next match is against
- Users should be able to see a list of all the teams they are apart of and see that team's page, as well as any upcoming matches they are participating in
- Given a list of participating teams and a Tournament type of either Round Robin or Single Elimination, the app should generate a Tournament as a series of Matches that depend on each other
- Team profiles should display the team's next match and the team's current record

- Tournament pages should display general information about the teams in that tournament and the current brackets, and allow users to create teams to join the tournament
- **Tasks postponed:** Team/User statistics, Additional types of tournaments, Separation of current and past tournaments