Project Proposal
For
Dragon Quiz

Instructor:	Prof. David Augenblick
Team Members:	Brandin Bulicki, Briana Schuetz, Dennis Lougharn, Isaiah Noble
Session:	003
Date Submitted:	8/12/18

Dragon Quiz

1. **Project Idea:**

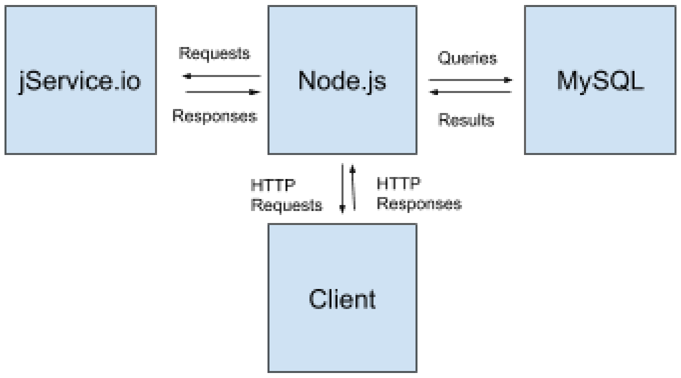
   Dragon Quiz is a trivia-based game in which users will answer questions from a variety of categories and compete with friends to gain the most points. Users can test their general knowledge on a variety of categories and accumulate points from correctly answering questions. The top earners will be displayed on a leaderboard.

2. **Functional Requirements:**

   Questions, categories (or themes), and difficulty levels will be pulled from the jService.io API. The game will store user data (score, name, and avatar) within a MySQL database, from which the leaderboard will be able to take the data to rank them. Node.js will handle the server-side execution processes.

3. **Scope:**
   - **Questions & Answers:** users will answer themed questions with assigned point values
     i. accessible from main menu
     ii. users will be able to choose from a list of available categories
     iii. questions will be displayed with their theme and the available point value questions (the point value will be determined by difficulty)
     iv. the user will select a point value and a question with possible answers will be displayed
     v. questions have four possible choices with only one answer
     vi. answering correctly adds points to the user's total score
     vii. answering incorrectly subtracts points from the user's total score
     viii. user's scores will be recorded in the database to be displayed in the user's profile and leaderboard
   - **Profile:** users will have profiles that display the user's name, a photo, and the amount of points they've accumulated
     i. accessible from the "Home" page and main menu
     ii. by default, the user's own profile is displayed
     iii. user can search other profiles in a search bar based on name or username
   - **Leaderboard:** users can access the leaderboard to see who has the highest scores
     i. accessible from the main menu
     ii. users are displayed from highest score to lowest score
     iii. user info will show only name and score, along with positional rank on the leaderboard

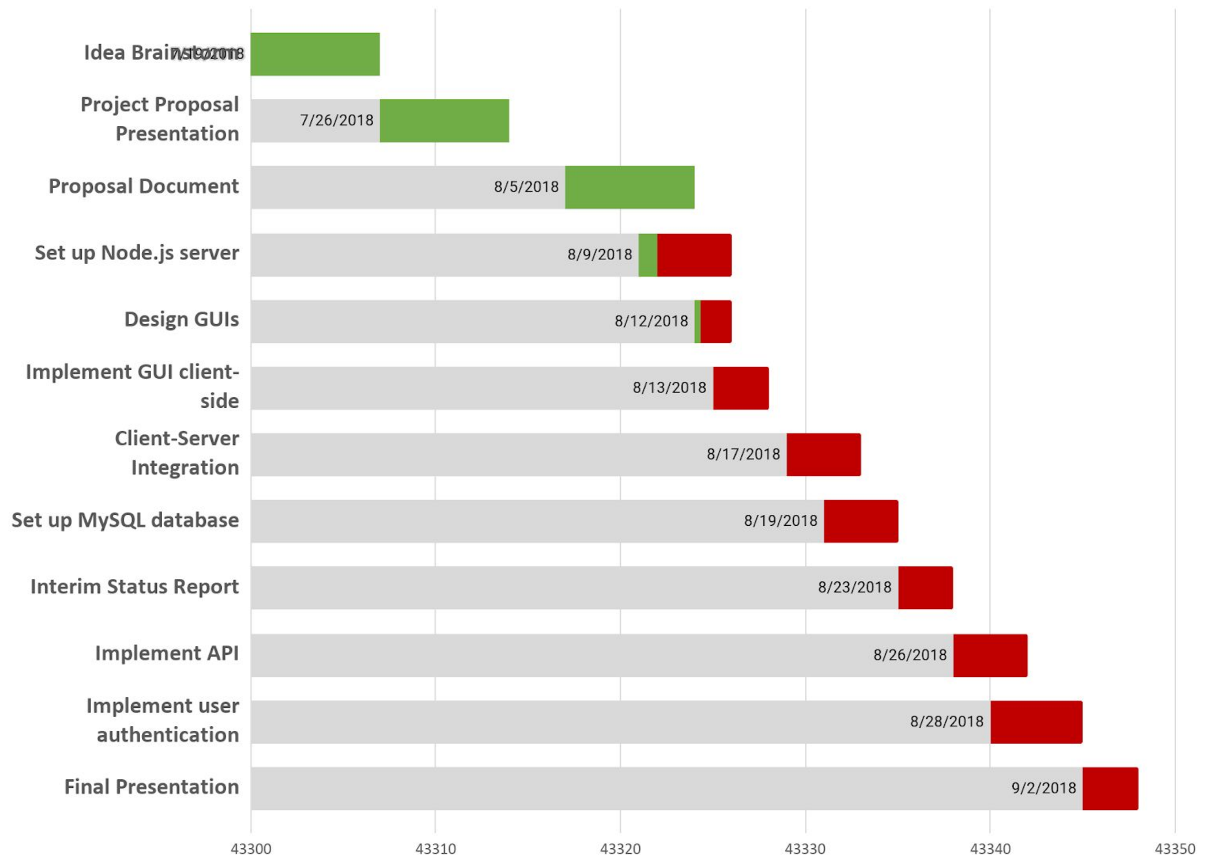
4. **Architectural Design:**
   - **Technologies:**
     i. Server: Node.js will act as the server, processing requests and relaying information from the API and database to the client.
     ii. Database: MySQL will store all user data needed for the application.
     iii. API: jService.io is a free external API that contains several categories of questions and answers which will be used by the application.
     iv. Framework: Bootstrap will be used to help design and format the user-facing client.
   - **Information Flow Diagram:**
     i. The Node.js server will be at the center. It will field requests from the client and return appropriate responses. It will relay requests to the jService.io API, receive responses, and send appropriate responses to the client. The server will also send queries to the MySQL database, receive results, and relay information back to the client.
     ii. Diagram:



5. **Gantt Chart:**
   - The Gantt Chart below has a rough outline of the rest of the term. Moving forward, we will add tasks as needed and update it regularly.
   - **Main roles:**
     i. Isaiah will be responsible for setting up the initial Node.js server during the beginning of Week 8. He will update the server as needed in the upcoming weeks.
     ii. Brandin will be responsible for designing and implementing the GUI during Week 8. If time allows, he will use Bootstrap.
     iii. Briana will be responsible for setting up the MySQL database also during Week 8. She will ensure that the server will be able to send queries to the database and it will respond.

iv. Once the server, client, and database are working together, Dennis will be responsible for implementing the API during Week 9.
v. Together, we will complete the Interim Status Report at the end of Week 9 and the Final Presentation at the end of Week 10.
vi. Throughout the upcoming weeks, we will work together to ensure that all the parts fit together.

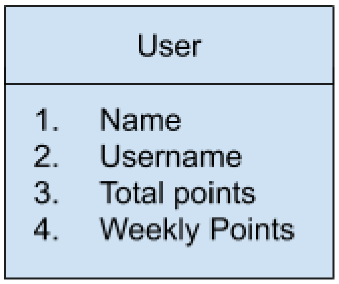
- **Chart:**



6. SQL Database Schema:

- The database will be relatively simple. There is not a lot of information that will need to be stored. It will only be storing basic user information such as name and points.
- The fields will likely be stored as follows:
  i. name VARCHAR(20),
  ii. username VARCHAR(20) key,
  iii. total_points INT,
  iv. weekly_points INT

- Schema:


User
1. Name
2. Username
3. Total points
4. Weekly Points