Iteration Feature Set Plan 1

Ethan Busbee, Anthony Cloudy, Cameron Keith, Justin Konersmann, Will Spurgin

Mobile Application

Login View (March 20th)
➔ Create buttons for submit of login and create user account
➔ Create text fields to input username and password
➔ Create title text for the view
➔ Store data to NSUserDefaults to allow persistent storage of login information

Create Account View (March 21st)
➔ Create button to submit account creation
➔ Create text fields to allow for a user to input all required data to create an account
➔ Store user login data to NSUserDefaults to allow for persistent storage of user login
➔ Create title text for the view

Set Alarm View (March 25th)
➔ Create picker view that allows for the choosing of when a user wants their alarm set
➔ Create button to start alarm and begin sleeping
➔ Create title text for the view

Alarm Running View (March 29th)
➔ Display the alarm time in a label
➔ Display the time until alarm in a label
➔ Create title text for the view
➔ Upon timer reaching the wake up time, display a Snooze and Wake-Up button
➔ If the Snooze button is pressed, add 5 minutes to the alarm
➔ If the Wake-Up button is pressed, return to the Set Alarm view

Web Application

Game:

Framework Design (March 20th)
➔ Render sprites to the canvas (March 18th)
➔ Create the foundation for the three logical views (March 20th)
   ◆ World map view

- ◆ Area map view
- ◆ Encounter view

Combat Implementation (March 28th)
- ➔ Make a working menu for player options in combat (March 23rd)
  - ◆ Attack menu and the attacks within
  - ◆ Spell menu and the spells within
- ➔ Implement the basis for battling (March 28th)
  - ◆ Create Player Objects
  - ◆ Create Monster Objects
  - ◆ Implement attacking
  - ◆ Implement "death" for the combatants
  - ◆ Implement winning and losing

Middleware: (March 26)
- ➔ Handling Routing (Most custom routing is done in the API but just in case) March 18th
- ➔ Backend Session handling (March 20)
  - ◆ User Session handling (permissions)
  - ◆ User login
  - ◆ User logout

REST API:
Stage 1 (March 24)
- ➔ Get user (by username) /users/:username
- ➔ Post (create) user /users
  - ◆ Create a character
  - ◆ JSON request packet should include new user : username, password, and email
- ➔ Get character (by username) /characters/:username
- ➔ Get inventory (by username) /characters/:username/inventory
- ➔ Post (add new) item for characters inventory /characters/:username/inventory
  - ◆ JSON request packet should include the item_id to be added
- ➔ Get areas (all) /areas
- ➔ Get areas monsters (by area) /areas/:id/monsters
- ➔ Get active World Bosses /bosses
- ➔ Get active World Fights (by character) /fights/:username
- ➔ Post new World Fight /fights
  - ◆ JSON packet should include boss_id, and username
- ➔ Get Leaderboards /leaderboard

Stage 2 (March 26)
- ➔ Put (update) character energy /characters/:username/energy/:energy
  - ◆ The ":energy" variable can either be additional energy to be added, or new total energy (room for debate there)

➔ Put (update) character experience /characters/:username/experience/:experience
  ◆ The ":experience" variable can either be additional experience to be added, or new total experience (room for debate there)

Database:
Creation (March 21)
➔ Add these tables to Night Knights Database:
  ◆ Users
  ◆ Characters
  ◆ Inventories
  ◆ Items
  ◆ World Bosses
  ◆ World Fights
  ◆ Monsters
  ◆ Area Monsters
  ◆ Areas

Setup (March 23)
➔ By the 23th all local machines will have this initial database up and ready to query.
➔ SQL dump of database on all team members machines capable of doing current working queries

Setup Continued (March 26)
➔ Create the database on the server
  ◆ SQL dump of database on server capable of doing current working queries
  ◆ Actual monster, world boss, and additional data added to database