

# Team Dragon

## Development Iteration 1 Plan

### Individual Plans

Each team member plan begins on a new page. Sort them by last name, then first name.

## Baniya, Tara (TB)

**Tasks to be attempted this iteration.** Express at high enough level so that there are no more than 3. Tasks must be verifiable (e.g., demoable and executable). Everyone must code.

- **Task:** create new user
- **Task:** login/logout

**Minimum acceptable verifiable outcomes.** List the minimum acceptable outcomes for this teammate to receive a satisfactory productivity evaluation. If there are non-functional requirements (e.g., follow UI style, follow code standards), state them.

- **Product:**
  - create model class and migration scripts (email, password, and confirm password )
  - view for create new user
  - controller method to manage creating new user
  - form validation
  - encrypt password
  - authenticate user with email and hashed password
  - after successful log-in, user should be redirected to user home page
  - manage user session till user is logged in the system
  - clear user session and redirect to application root page after user successfully logs out

**Approved by:** TB, NP, SK,NM , RY, CT

## **Khatun, Saleha (SK)**

**Tasks to be attempted this iteration.** Express at high enough level so that there are no more than 3. Tasks must be verifiable (e.g., demoable and executable). Everyone must code.

- **Task:** view game details
- **Product:**
  - view page to view status, history, players and other details of a game.
  - controller method for viewing game details
  - a link to go back to the user specific homepage.
- **Task:** edit game
- **Product:**
  - view to edit game details e.g. game history, map etc.(according to the game rule).
  - controller method for editing game
  - cancel button to cancel the changes.If user presses the cancel button, then it will redirect to the game details view page.

**Approved by:** SK, NP , RY,NM,TB, CT

## **Maharjan, Nabin (NM)**

**Tasks to be attempted this iteration.** Express at high enough level so that there are no more than 3. Tasks must be verifiable (e.g., demoable and executable). Everyone must code.

- **Task:** Create model, view and controller classes for creating a new game
- **Task:** View list of active games
- **Task:** Allow Dungeon Master to invite user(s) to join a game

**Minimum acceptable verifiable outcomes.** List the minimum acceptable outcomes for this teammate to receive a satisfactory productivity evaluation. If there are non-functional requirements (e.g., follow UI style, follow code standards), state them.

- **Product:**
  - create a model class and related scripts for Game
  - create a view for creating a new game
  - create a controller and required methods for creating a new game
  - user creating a new game should be assigned a role of Dungeon Master for that game
  - create a view for listing active games (games in which user is either DM or player)
  - view for Dungeon Master to invite user(s) to join his/her game.
  - enforce form validation while creating a new game or sending invitation to user(s) to join a game
  - view and controller method to list pending game invitations

**Approved by:** NM, RY, NP, SK, TB, CT

## **Nurmohammed Patwary (NP)**

**Tasks to be attempted this iteration.** Express at high enough level so that there are no more than 3. Tasks must be verifiable (e.g., demoable and executable). Everyone must code.

- **Tasks:**

1. Join game:

An action button or link to “join game form” for each game in a list of pending game invitations. User joins the game using the password sent by the Dungeon master and selecting a player character for the game.

2. Leave game:

an action button that will allow user to leave a game. A pop up window should appear to confirm that user is certain to leave the game. When a user leave the game his player character should be destroyed from that game.

3. Delete game.

An action button which will allow the user to delete the game for which he is the Dungeon Master.

- **Products:**

1. a view and controller method for joining game.

2. form validation when user joins a game (user should provide valid game password and same player should not allowed to join multiple games simultaneously.)

3. controller method for leaving game and “leave game” button with confirmation pop-up notification that removes a player character association from a game (The player character must not be deleted.).

4. controller method for deleting game and “delete game” button with pop-up notification that will delete a game model from the game list.

**Approved by:** NP, SK, RY,NM,TB, CT

## Taylor, Chris (CT)

**Tasks to be attempted this iteration.** Express at high enough level so that there are no more than 3. Tasks must be verifiable (e.g., demoable and executable). Everyone must code.

- **Task:** Creating model view controller for player character details. (Build character sheet)
- **Task:** Design Print Character sheet view
- **Task:** Collaborate with team members on building site theme/layout

**Minimum acceptable verifiable outcomes.** List the minimum acceptable outcomes for this teammate to receive a satisfactory productivity evaluation. If there are non-functional requirements (e.g., follow UI style, follow code standards), state them.

### **Product:**

- a form for the user to build a character sheet (character details)
- create Player Character model and scripts
- create Player controller (should append: and related method to create/build new Player)
- create view that allows user to view and edit character details (the plan is to build a modular view that can also be used by a dungeon master or a user in the party in a game)
- form validation
- from home page, provide a link to a list of user's player characters, where they can see character details and print.
- iteration 1 should have a site layout that Andrew (the customer) approves.

**Approved by:** CT,TB, NP, SK,NM, RY

## **Yasmin, Rojoba (RY)**

**Tasks to be attempted this iteration.** Express at high enough level so that there are no more than 3. Tasks must be verifiable (e.g., demoable and executable). Everyone must code.

- **Tasks:**

1. Buy item for player character
2. Sell item

- **Products:**

1. Generate view page containing list of Items available for buy Examples: Swords etc.
2. Generate view page containing list of items available for sell.
3. Generate controller methods to buy and sell items.
4. Generate view forms for buying and selling items.
5. Generate Model class for items.
6. Form validation.
7. A link to go back to player details page.

**Approved by:** RY,NM, NP, SK,TB, CT