DnDeity
Team 22 - Sprint 1 Retrospective

Benjamin Crumbacher, Nicholas Fordyce, Christian McKain, Riley Turnbull

What went well?

In general, we completed all of our user stories for this sprint. We are able to run the database, server, and frontend locally and have them interact together. Users are able to login, create an account, change their password, create a DnD character, and manage their existing characters.

1. As a user, I would like to be able to register for a DnDeity account

Number	Description	Time	Owner
1	Set up React App	30min	Nicholas
2	Set up Bootstrap	30min	Nicholas
3	Set up Flask Server	2hr	Riley
4	Set up REST API	5hr	Riley
5	Set up MySQL Database	2hr	Ben
6	Design frontend create user page	5hr	Christian
7	Backend endpoint for creating new user	4hr	Riley
8	Create database tables for users	2hr	Ben
9	Write insert query for new users	2hr	Ben
10	Testing: Display error if username/password/email are empty or if a username is taken, unit test for acceptable character input	2hr	Nicholas

We completed all of the tasks for this user story. We were able to set up all of the technologies we needed for this project. A user was able to navigate to a create user page, where they could create an account for our web application.

2. As a user, I would like to be able to login to DnDeity

Number	Description	Time	Owner
1	Design frontend login page	2hr	Christian
2	Backend endpoint for logging in	2hr	Riley
3	Use Flask-Login to handle user sessions	3hr	Riley
4	Write query for getting user info	1hr	Ben
5	Set up RDS hosting	2hr	Ben
6	Set up S3 buckets	2hr	Ben
7	Set up Elastic Beanstalk	2hr	Ben
8	Set up CloudFront	2hr	Ben
9	Testing: Display error if username/password are empty or if username/password are invalid	1hr	Nicholas

We completed all of the tasks for this user story. Current users are able to enter their account info to login into their account. Furthermore once logged in all relevant account info is remembered and displayed.

3. As a user, I would like to be able to reset my password

Number	Description	Time	Owner
1	Design reset password page	2hr	Nicholas
2	Backend endpoint for sending reset password email	1hr	Riley
3	Set up SMTP for sending email	4hr	Riley
4	Testing: Able to reset password from email, new password correctly updates.	2hr	Ben

We completed all of the tasks for this user story, and this feature is fully functional. Users can request an email to change their password, and then follow a link to a page where they can change their password. The only enhancement to this user story would be to store our passwords and user ids more securely by hashing them or encrypting them.

4. As a player, I would like to be able to create a character sheet and set details such as a character's name, ability modifiers, level, class abilities, and description.

Number	Description	Time	Owner
1	Design character creator race section	2hr	Nicholas
2	Design character creator class section	2hr	Nicholas
3	Design character creator ability score section	2hr	Nicholas
~~4~~	~~Design character creator background section~~	~~2hr~~	~~Nicholas~~
5	Design character equipment modal	3hr	Nicholas
6	Design spell modal	7hr	Christian
7	Create tables for DnD API data	2hr	Ben
8	DnD API data integration	7hr	Ben
9	Design character description section	2hr	Christian
10	Design character creator navigation menu	2hr	Nicholas
11	Backend endpoint for getting DnD data from the database	2hr	Riley
12	Backend endpoint for saving characters	2hr	Riley
13	Database insert query for new characters	2hr	Ben
14	Create character instance object	3hr	Riley
15	Build and handle server requests/responses on frontend	4hr	Nicholas
16	Testing: Check database to see if the character saved contains all of the information entered on the character creation screen	2hr	Nicholas

We completed all but one task for this user story, but we still completed this user story. Users are able to navigate to a character creation page where they can create a character and modify their race, class, ability scores, equipment, spells, and description. Some future enhancements we can make to the character creator are to add a background section and description text for items/spells/etc.

5. As a player, I would like to update and delete my existing character sheets.

Number	Description	Time	Owner
1	Design DnDeity Home Page	5hr	Christian
2	Design Home Page Navbar	2hr	Nicholas
3	Set up url routing on frontend using React Router	3hr	Nicholas
4	Design character sheet selection page	7hr	Christian
~~5~~	~~Design confirmation prompt for deleting characters~~	~~2hr~~	~~Christian~~
6	Backend endpoint for getting characters	1hr	Riley
7	Backend endpoint for deleting characters	1hr	Riley
8	Testing: Make multiple characters to see if they appear, delete characters to test if they disappear	2hr	Ben

We didn't finish one of the tasks for this user story, but users are still able to navigate to a page where they can see all of their existing characters, and update/delete them.

What did not go well?

In general, we did not thoroughly and programmatically test our application, we only tested manually by going through the features on the website. This means we might have unresolved edge cases that we have yet to find. We also did not restrict much as far as user input, especially in the area of usernames and passwords. We also did not thoroughly examine security for the application such as hashing passwords.

Even though we completed our user stories, we had some unfinished tasks, which are listed below.

User Story 4:

4	Design character creator background section	2hr	Nicholas

Not Completed

We did not get to the character backgrounds section of the character creator because the DnD API did not provide that information. In order to save time for this sprint, we decided to push it to the next sprint. We will have to manually enter the character background data into the database.

User Story 5:

5	Design confirmation prompt for deleting characters	2hr	Christian

<u>Not Completed</u>
We also forgot to include a confirmation prompt for when a user deletes a character. This task was not critical for the character deletion feature to function, but it would improve the user experience so that users can’t accidentally delete characters.

How should we improve?

Our main issue that we encountered this sprint is that we procrastinated too much. We saved a majority of our work for the last week, which meant that we had to pour in a huge amount of hours into a small amount of time. This also didn’t give us time to deal with any unexpected roadblocks, which could have been avoided if we started earlier and spread out our work.

If code is dependent on other code, then we will finish the necessary code first. In the first sprint we needed the backend endpoints to be done before we used them in the frontend, but some of these endpoints weren’t finished until the last few days. For the next sprint we will finish the backend endpoints early on so no one has to wait for code to be written to move forward.

One small improvement that we are going to make is to never push directly to our master branch. Most of the time we created branches and made pull requests into master, but towards the end of the sprint we started slacking and pushed directly to master a lot. This will make it easier to keep track of our changes.

Finally, in order to procrastinate less, we add artificial milestones to pace ourselves within the sprint. For instance, we will do a third of our user stories the first week, then the next third, and so on. This will help us not leave everything until the end.