Quack

Team 3 - Sprint 1 Planning Document

Justin Hartman
Todd Griffin
Theo Burkhart
Tyler Rex
Mason Wesolek

Sprint Overview

In sprint 1, we would like to design the main UI of our application and implement the main features. Our goal is to allow instructors to create courses, sections, register students, and create quizzes. For students we plan on adding functionality to sign up for Quack, register for courses, view past quiz grades, and implement the first version of taking a quiz. We also want to design the API our system will use to communicate between the clients and server and create test cases to begin testing our React Native applications.

SCRUM Master: Tyler Rex

Meeting plan: We plan to meet on Wednesdays in a reserved room at WALC at 7pm.

Risks & Challenges: Luckily we all have experience designing backends and mobile/web applications. We are confident in our ability to move at the pace we need to once we get off the ground. However, our significant challenge will be starting from scratch with React. Only one of our front end designers has experience with React. We will need to quickly learn React and React Native to being working at a steady pace. Thankfully, Facebook (creators of React) provide lots of resources as well as quick prototyping tools to go from Javascript to a native application quickly.

Sprint Detail

User Story #1

As a user, I would like to be able to sign up with just an email and school ID.

#	Description	Estimated Time	Owner
1	Set up backend server	1 hour	Todd
2	Set up authentication with Relay	2 hours	Justin
3	Create register page, with email/password fields	2 hours	Theo
4	Add register function to server	4 hours	Todd

Acceptance Criteria

- Given the backend to register users is successful, when a request to register an account is made then the user's account will be added to the database given that their information for their account is correct.
- Given the user has been added to the database correctly, they will be able to login next time they try to open the app.
- Given the user can sign up, they will be able to register as an instructor and be able to have access to instructor permissions.

User Story #2

As a user, I would like to be able to register all my classes within the app so that it is easily organized.

#	Description	Estimated Time	Owner
1	Create screen for users to add a course	4 hours	Theo
2	Create 'add class' function to server	4 hours	Justin
3	Create screen to organize courses	5 hours	Theo
4	Add 'get courses' functionality to GraphQL schema	6 hours	Todd

Acceptance Criteria

- Given that a user is able to login successfully, a user will be able to add a course to their schedule and see it when they open the app.
- Given that a user is able to add themselves to a course successfully, a user will be able to see all of their courses on their main page and see the details of those classes, such as the lecture time and professor.
- Given that a user is able to see their courses and add them to their account successfully, when a user adds a new course, the database will update their information and it will update on the user's course page as well.

User Story #3

As an instructor, I would like to create courses.

#	Description	Estimated Time	Owner
1	Build client button(s) to run 'create class' API request	2 hours	Tyler
2	Write GraphQL schema to handle create class request.	3 hours	Todd
3	Add 'create class' function to server	3 hours	Justin
4	Handle permissions for creating class	5 hours	Justin

Acceptance Criteria

- Given that the user is using instructor interface, when an instructor hits the create class button they are able to create a class with a pop-up.
- Given that a pop-up appears, the class will appear to the client in the list of classes.
- Given that if you are not an instructor, you will not see the create class button

User Story #4

As an instructor, I would like see a list of students in attendance.

#	Description	Estimated Time	Owner
1	Setup database to keep track of quizzes/attendance	4 hours	Justin
2	Write GraphQL schema to handle student attendance counts	5 hours	Todd

3	Create screen that lists students in a given course	6 hours	Theo
4	Add server functionality to get student info from a course	4 hours	Justin

Acceptance Criteria:

- Given that an instructor is able to create a course, there will be a roster screen associated with each class.
- Given that an instructor can view the details of their class, the instructor will be able to view the attendance of each student.
- Given that an instructor has users in their class, they can view each students grades.

User Story #5

As a developer, I would like a consistent method of checking out branches and making pull requests to the master branch.

#	Description	Estimated Time	Owner
1	Research common practices for git collaboration	1 hour	Theo
2	Discuss and finalize a system for organizing our git repository	3 hours	Theo

Acceptance Criteria:

- Given a method of checking out branches and making pull requests is created, team members should be able to work concurrently without merge conflicts.
- Given a method of checking out branches and making pull requests is created, each branch should correspond to a feature or bug fix being worked on.
- Given a method of checking out branches and making pull requests is created, all team members should have permission to create branches off the master branch.
- Given a system for organizing our git repository is created, our master branch should remain finalized and bug free while a development branch is kept separate for features in progress.

User Story #6

As a developer, I would like to develop an API that can be used across all applications so that it is easy to set a standard for all platforms.

#	Description	Estimated Time	Owner
1	Set up GraphQL schema	2 hours	Todd
2	Set up server	2 hours	Todd

Acceptance Criteria:

- Given that a developer is developing on the mobile application, the API that is used to communicate to the server will be the same on every platform.
- Given that a developer is developing on the mobile application, the API will send the same results no matter the platform.
- Given that a developer has successfully authenticated the client, the API should be given the same access on each platform.

User Story #7

As a developer, I would like a method of developing test cases for React Native code.

#	Description	Estimated Time	Owner
1	Implement Jest into project	3 hours	Justin
2	Maintain tests throughout sprint	7 hours	Justin

Acceptance Criteria:

- Given that a developer is working with React Native on the project, they will implement tests as they make complete tasks on the project backlog.
- Given that a developer has a finished an item on the project backlog, they will be able to test their code with unit tests that are developed by the team.
- Given that a developer has successfully tested their code, they will modify and update tests as needed as the project continues to progress.

User Story #8

As an instructor, I would like to keep record of students' attendance.

#	Description	Estimated Time	Owner
1	Create a “roster” tab to the course menu	1 hour	Theo
2	Create a option to switch between and add new sections to a course	2 hours	Theo
3	Create a scrollable list of all students currently enrolled in a course	2 hours	Theo
4	Create an option to register students to current selected course	5 hours	Justin
5	Display attendance information for each student	3 hours	Tyler

Acceptance Criteria:

- Given the roster page is created successfully, when an instructor arrives on the roster page, they should see a list of their sections and students registered to each section.
- Given the sections menu is successfully implemented, when an instructor selects edit, they should be able to add or delete sections and assign different users to manage each section.
- Given the roster view is successfully implemented, when an instructor clicks on a student, they should be able to see when a student was absent.

User Story #9

As an instructor, I would like to manage multiple classes.

#	Description	Estimated Time	Owner
1	Create a “add course” button	2 hours	Theo
2	Create a sidebar menu to switch between courses	5 hours	Theo

Acceptance Criteria:

- Given the main dashboard is implemented successfully, when an instructor opens the software, all their current courses should be displayed on the left sidebar.
- Given the sidebar menu is implemented successfully, when an instructor clicks on a course, the main view of the application should switch to the selected course.
- Given the sidebar menu is implemented successfully, when an instructor clicks on “add a course” a pop-up should appear to fill out details for a new course.

User Story #10

As an instructor, I would like to input and track all my students for each class.

#	Description	Estimated Time	Owner
1	Create a roster page associated with all the students for each class	3 hours	Tyler
2	Create a student view that gives data about a specific student	3 hours	Tyler
3	Implement the navigation for access to the roster page and student views	2 hours	Mason

Acceptance Criteria:

- Given that the roster page and student view is implemented correctly, a professor will be able to see and select any student that is registered for his/her class.
- Given that the student has participated in the class, the professor will be able to see all relevant information for a selected student.

User Story #11

As a user, I would like to be able to view the quiz on my device

#	Description	Estimated Time	Owner
1	Implement functionality and navigation to only view a quiz when it is opened by the creator	6 hours	Mason
2	Create first version of quiz screen	2 hours	Mason
3	Implement the first version of quiz screen within the app	1 day	Mason
4	Implement functionality for the quiz screen to adapt to different types of quizzes	6 hours	Mason
5	Implement functionality for the quiz screen to have images and media that the creator supplies	6 hours	Mason

Acceptance Criteria:

- Given that the quiz screen has been implemented correctly, users will be able to take a quiz of any supported type when a professor opens it.
- Given that the quiz screen has been implemented correctly, users will be able to see all media that a quiz creator has included.
- Given that the navigation to get to the quiz screen is correct, users will be able to select a answer.

User Story #12

As a user, I would like to be able to view past quizzes and see the grades.

#	Description	Estimated Time	Owner
1	Create a scrollable list of grades for the current user	2 hours	Mason
2	Create a screen where the user can view grades and from a previous quiz	3 hours	Mason
3	Implement functionality to allow students to contact a TA or specified group with questions	5 hours	Mason
4	Implement the grade list and viewing screens in the app	8 hours	Mason
5	Implement functionality to see the exact quiz with questions and images on the grade screen	1 day	Mason

Acceptance Criteria:

- Given that the grade and past quiz screen has been implemented correctly, when a user wants to see the grade from a previous quiz they will have a list of all past quizzes with their grades.
- Given that a grade is available, the user will be able to see the exact quiz and questions that are attached to that grade.
- Given that the user has a question with their grade, there will be an option to contact the professor or TA with their question.

Remaining Backlog

Functional Requirements

1. As a user, I would like to be able to take quizzes in class.
2. As a user, I would like to be able to answer poll questions during class.
3. As a user, I would like to answer a multitude of quiz types (multiple choice, yes/no - true/false, open-ended (free response), fill-in-the-blank, matching)
4. As a user, I would like to take quizzes without the app so that I can use my laptop or phone's browser.
5. As a user, I would like an easy-to-use interface.
6. As a user, I would like a consistent interface between different platforms.
7. As an instructor, I would like to be able to set a class password for quizzes so that it is harder for students to cheat on attendance.
8. As an instructor, I would like to be able to create quizzes.
9. As an instructor, I would like to see live results of my quizzes.
10. As an instructor, I would like to show a class quiz results live.
11. As an instructor, I would like to administer multiple choice and free response quizzes.
12. As an instructor, I would like to see analytics about my quiz results.
13. As an instructor, I would like to be able to visualize data about my class results through graphs or charts.
14. As an instructor, I would like to export results into a spreadsheet.
15. As an instructor, I would like an easy-to-use interface.
16. As an instructor, I would like to be able to access the system through a web browser.
17. As an instructor, I would like to prevent my students from cheating or committing other types of academic dishonesty.
18. As a developer, I would like to gain user feedback from an anonymous feedback feature that users and professors can use.
19. As a developer, I would like to know when our servers are reaching capacity.
20. As a developer, I would like to create a system that is scalable to handle fluctuating demand.