Quack

Team 3 - Sprint 3 Planning Document

Justin Hartman
Todd Griffin
Theo Burkhart
Tyler Rex
Mason Wesolek

Sprint Overview

For Sprint 3, we aim to complete all the app's features, fix bugs, refine the navigation, and connect the backend, native app, and web app together. A majority of the core features are now complete for the web and native app, but require thorough testing. We would like to complete the apps as quickly as possible in order to leave ample time to test and work out any bugs that might come from larger scale testing.

SCRUM Master: Tyler Rex

Meeting plan: We plan to meet on Mondays and Wednesdays in a reserved room at WALC at 11:30 am.

Risks & Challenges:

The main challenges we face in this sprint is the uncertainty that comes from connecting our backend to frontend. We have yet to create a system to handle real-time quizzes. We also have not thoroughly tested sending and receiving data from the backend through the native or web app.

Sprint Detail

User Story #1

As an instructor, I would like to be able to set a password for a quiz so that it is harder for students to cheat on attendance.

#	Description	Estimated Time	Owner
1.	Create option to administer any quiz with a password	1 hr	Theo
2.	Generate unique token on backend to be used with quiz	3 hrs	Justin
3.	Manual Testing	30 min	Theo

Acceptance Criteria

- Given the first task is completed successfully, an instructor should see a pop-up when launching a quiz allowing them to launch with or without a password
- Given that the second task is completed successfully, an instructor should see a uniquely generated token when a quiz is started
- Given that the second task is completed successfully and an instructor enables it, a student will have to enter a passcode before taking a quiz.

User Story #2

As an instructor, I would like a form to create a new quiz in a course.

#	Description	Estimated Time	Owner
1.	Create a “Create quiz” button	1 hr	Theo
2.	Create UI to set quiz options (type, password protected, etc.)	2 hrs	Theo
3.	Create UI to add questions to a quiz	8 hrs	Theo
4.	Create UI to add images to quizzes	2 hrs	Theo
5.	Create UI to save quiz to course	1 hr	Theo

6.	Connect quiz creation to backend (save quizzes, upload images, & set options)	5 hrs	Justin
7.	Manual Testing	30 min	Theo

Acceptance Criteria

- Given the quiz options UI is completed successfully, an instructor should be able to set options when creating a new quiz.
- Given the UI to add questions to a quiz is completed successfully, an instructor should be able to add questions of all types to a quiz and input the correct answer.
- Given adding images to quiz questions is implemented successfully, an instructor should be able to add an image via a file on their computer or a URL.

User Story #3

As an instructor, I would like to be able to start a quiz I’ve already created.

#	Description	Estimated Time	Owner
1.	Create “Start quiz” button	1 hr	Theo
2.	Add GraphQL resolver to start a quiz	2 hrs	Justin

Acceptance Criteria

- Given the “Start quiz” button is created successfully, next to each unadministered quiz will be the button to start it.
- Given that the second task is completed successfully, a user should see an option to take a quiz when it is started.
- Given the “Start quiz” button is created successfully, a quiz will be available for students of the course to take.

User Story #4

As an instructor, I would like quiz results to update live in the database.

#	Description	Estimated Time	Owner
1.	Research/create GraphQL Subscriptions api to the Quack API	20 hrs	Todd

2.	Create GraphQL Subscription about updated Quiz results	10 hrs	Todd
3.	Add test cases for ‘updated quiz’ subscription	30 mins	Todd

Acceptance Criteria

- Given the server receives the correct request, a client can subscribe to a certain event.
- Given a client subscribes to an event, it gets their updates.
- Given a subscription is successfully implemented, all the test cases will pass.

User Story #5

As an instructor, I would like to be able to visualize data about my quiz results through graphs or charts.

#	Description	Estimated Time	Owner
1.	Create UI to view results of past quizzes	3 hrs	Theo
2.	Implement API to produce graphs/charts based on quiz data	2 hrs	Theo
3.	Create resolvers that gather data for different quiz IDs	3 hrs	Justin
4.	Manual Testing	30 min	Theo

Acceptance Criteria

- Given the UI to view past quiz results is created successfully, an instructor will be able to select a quiz and view information such as responses and attendance.
- Given the graphs/charts API is implemented successfully, non-free response quizzes should have accompanying graphs and charts to visualize student responses.
- Give

User Story #6

As an instructor I would like to login to my account through a browser.

#	Description	Estimated Time	Owner

1.	Finalize instructor login UI	1 hrs	Theo
2.	Add roles to users when registering through the web vs. mobile app	3 hrs	Justin
3.	Add test cases for Roles class	2.5 hrs	Todd
4.	Manual testing	30 min	Theo

Acceptance Criteria

- Given the instructor's login UI is completed successfully, an instructor will be able to log into the service via a web browser with their email and password.
- Given the instructor’s login UI is completed successfully, an instructor will be able to register for the service via a web browser.
- Given roles are implemented successfully, signing up as an instructor will give that account instructor permissions.
- Given the Roles class test cases are ran, they will all pass.

User Story #7

As a user, I would like to join a course.

#	Description	Estimated Time	Owner
1.	Create a search bar on the homescreen	.5 hr	Tyler
2.	Allow users to search for and add classes with the search bar	2 hrs	Tyler
3.	Create an add button that will add the class	1 hr	Tyler
4.	Send a request to the backend to add the course to the users list of courses	2 hr	Tyler
5.	Create unique 5-digit tokens for course ID	2 hrs	Justin
6.	Manual Testing	.5 hrs	Tyler

Acceptance Criteria

- Given that you are on the home screen of the mobile app, when you click in the search bar and enter text, you should see classes that match your search.
- Given that you press on the plus sign next to a course, that course should be added to your list.

- Given that you try to add a course you are already in, you should receive an error message.

User Story #8

As an instructor, I would like to see live results of my quizzes.

#	Description	Estimated Time	Owner
1.	Create UI to view quiz results in real-time while quiz is running	5 hrs	Theo
2.	Create GraphQL subscription to listen for ‘newly created answers’ to a specified quiz	10 hrs	Todd
3.	Add test cases for ‘newly created answers’ subscription	30 mins	Todd
4.	Manual testing	30 mins	Theo

Acceptance Criteria

- Given quiz results can be viewed in real-time, an instructor should be able to see responses from their quiz for non-free response questions (true/false & multiple choice).
- Given quiz results can be viewed in real-time, a user should only be able to see their own answer and not any others for that quiz.
- Given quiz results can be viewed in real-time, a user should only be able to see the class results once the instructor has made it available to view.

User Story #9

As a user, I would like to be able to see results of my quiz and others that have taken the quiz.

#	Description	Estimated Time	Owner
1.	Design the UI for quiz results screen	1 hrs	Mason
2.	Implement a graph that shows data from other people that have taken the quiz	4 hrs	Mason

3.	Connect the UI to the backend	4 hrs	Mason
4.	Implement labels on the graph	2 hrs	Mason
5.	Manual Testing	30 min	Mason

Acceptance Criteria

- Given that multiple users have taken a quiz, the graph will show their results if it was a multiple choice quiz.
- Given that the UI has been implemented correctly, it will show the results on the screen from the backend.
- Given that the UI is implemented correctly, there will be no overlap on the screen with items.

User Story #10

As a developer, I would like to gain user feedback from an anonymous feedback feature that users and professors can use.

#	Description	Estimated Time	Owner
1.	Make GraphQL type for Feedback class.	1 hrs	Todd
2.	Make GraphQL resolvers for Feedback class.	4 hrs	Justin
3.	Make test cases for Feedback GraphQL resolvers	2.5 hrs	Todd
4.	Create UI for a student to submit feedback on native app	4 hrs	Mason
5.	Create UI for a professor to submit feedback on web app	2 hrs	Theo

Acceptance Criteria

- Given the server receives the appropriate request, a Feedback can be created in the database.
- Given the server receives the appropriate request, a Feedback can be received from the database.
- Given the tests for the Feedback are ran, they all pass.

User Story #11

As a developer, I would like to make sure that both iOS and the Android versions of quack run the same.

#	Description	Estimated Time	Owner
1.	Test for inconsistencies between iOS and the Android versions of the app	2 hr	Mason
2.	Fix errors with Android app that may cause it to crash	3 hrs	Mason
3.	Verify the backend is working the same between both versions	1 hr	Mason
4.	Manual testing	30 minutes	Mason

Acceptance Criteria

- Given that the app is optimized correctly, views will look the same between the iOS and Android versions
- Given that the app is implemented correctly, both versions will connect and submit data to the backend.
- Given that the app is implemented correctly, the keyboard views will be consistent between versions.

User Story #12

As a developer, I would like to know the status of our server.

#	Description	Estimated Time	Owner
1.	Create GraphQL Subscription to listen for ‘feedback_created’ events.	2 hrs	Todd
2.	Create test cases ‘feedback_created’ GraphQL Subscription	30 mins	Todd

Acceptance Criteria

- Given the appropriate query is created, a socket.io socket can subscribe to the ‘feedback_created’ event.

- Given a successful subscription, a socket listening for the ‘feedback_created’ event on our server will receive all feedbacks created.
- Given the appropriate request, a socket.io socket can unsubscribe to the ‘feedback_created’ event.

User Story #13

As a user, I would like to be able to navigate between all of the screens in the mobile app.

#	Description	Estimated Time	Owner
1.	Connect the quiz screens made in the last sprint to the homepage.	2 hrs	Tyler
2.	Allow a user to navigate to a live quiz from the homepage or from a course details screen	1 hr	Tyler
3.	Ensure that a user always has a way to return to the homepage and that the animation goes backwards	2 hrs.	Tyler
4.	Manual Testing	.5 hrs	Tyler

Acceptance Criteria

- Given that a class that I am in has a quiz starting, I should be able to see it on the course details screen and navigate to it.
- Given that you are on any screen in the mobile app, you should be able to return to the Homepage.
- Given that you have finished a quiz, you should be automatically returned to the course details screen for the course that you took the quiz for.

Remaining Backlog

Functional Requirements

1. ~~As a user, I would like to input an answer to a multiple choice quiz.~~
2. ~~As a user, I would like to input an to answer to yes or no / true or false questions in a quiz.~~
3. ~~As a user, I would like to input an to answer to open-ended questions in a quiz.~~
4. ~~As a user, I would like to input an to answer to fill-in-the-blank questions in a quiz.~~
5. ~~As a user, I would like my answers to quizzes to be recorded by the database.~~
6. ~~As a user, I would like to join a course after being given a course ID from the instructor.~~
7. ~~As a user, I would like to be able to sign up with just an email and school ID number so that it is easy to set up.~~
8. ~~As a user, I would like to be able to answer questions in a poll.~~
9. ~~As an instructor, I would like to create courses.~~
10. ~~As a user, I would like to be able to register all my classes within the app so that it is easily organized.~~
11. ~~As a user, I would like to be able to view past quizzes and see the grades.~~
12. ~~As an instructor, I would like to be able to set a password for a quiz so that it is harder for students to cheat on attendance.~~
13. ~~As an instructor, I would like a form to create a new quiz in a course.~~
14. ~~As an instructor, I would like my quizzes to be saved to the database.~~
15. ~~As an instructor, I would like to be able to start a quiz I've already created.~~
16. ~~As an instructor, I would like quiz results to update live in the database.~~
17. ~~As an instructor, I would like to see live results of my quizzes.~~
18. ~~As an instructor, I would like see a list of students in attendance.~~
19. ~~As an instructor, I would like to be able to visualize data about my quiz results through graphs or charts.~~
20. ~~As an instructor I would like to login to my account through a browser.~~
21. ~~As an instructor, I would like to export quiz results into a spreadsheet.~~
22. ~~As an instructor, I would like to be able to login through a web app.~~
23. ~~As an instructor, I would like an interface to switch between courses.~~
24. ~~As an instructor, I would like to view course details for each course.~~
25. ~~As an instructor, I would like to view course quizzes for each course.~~
26. ~~As an instructor, I would like to view the course roster for each course.~~
27. ~~As an instructor, I would like to view the course roles for each course.~~

28. ~~As a user, I would like to be able to view the quiz on my device.~~
29. ~~As a developer, I would like to gain user feedback from an anonymous feedback feature that users and professors can use.~~
30. ~~As a developer, I would like to implement JEST for testing the UI in our project.~~
31. ~~As a developer, I would like to know when our servers are reaching capacity.~~
32. ~~As a developer, I would like a consistent method of checking out branches and making pull requests to the master branch.~~
33. ~~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.~~
34. ~~As a developer, I would like a method of developing test cases for React Native code.~~