Quack

Team 3
Justin Hartman, Mason Wesolek, Theo Burkhart, Todd Griffin, Tyler Rex

Project Backlog

Problem Statement

With the current iClicker system that is used in classes across Purdue's campus, it is very limited in its functionality. Students have no way to see their results on their handheld devices, professors have no easy way of scheduling quizzes and managing multiple classes at a time, and students can only answer multiple choice questions with up to 5 options. The iClicker system also requires additional proprietary hardware that the students and the professor must purchase.

Background Information

Since everyone has some kind of laptop or smartphone with them during class, it does not make sense to have students buy a seperate device for taking quizzes. Instead, we plan on developing a system, Quack, which will be an improvement on the in-class quizz experience. Quack can be used by students not just at Purdue University, but any college or high school around the world.

There are current products such as HotSeat or Kahoot that take a similar approach to utilizing modern technology in the classroom. However, these systems are limited in their capabilities. HotSeat only allows a student to enter up to 100 characters as a response and Kahoot is less professional and marketed towards younger students. We believe that students would benefit from having the past results of all their quizzes to use as a study guide and being able to see how their performance compares to other students in the course. We also think that professors could save time setting up quizzes with an effortless solution to schedule and make quizzes with Quack.

Functional Requirements

1. 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.
2. As a user, I would like to be able to view past quizzes so that I can keep track of my progress in the class.
3. As a user, I would like to be able to view the quiz on my device so I do not have to look at the front of the class.
4. As a user, I would like to be able to take quizzes in class.
5. As a user, I would like to be able to answer poll questions during class.
6. 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)

7. As a user, I would like to be able to register all my classes within the app so that it is easily organized.
8. As a user, I would like to take quizzes without the app so that I can use my laptop or phone's browser.
9. As a user, I would like an easy-to-use interface.
10. As a user, I would like a consistent interface between different platforms.
11. As an instructor, I would like to create classes.
12. 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.
13. As an instructor, I would like to input and track all my students for each class.
14. As an instructor, I would like to be able to create quizzes.
15. As an instructor, I would like to see live results of my quizzes.
16. As an instructor, I would like see a list of students in attendance.
17. As an instructor, I would like to keep record of students' attendance.
18. As an instructor, I would like to show a class quiz results live.
19. As an instructor, I would like to administer multiple choice and free response quizzes.
20. As an instructor, I would like to see analytics about my quiz results.
21. As an instructor, I would like to be able to visualize data about my class results through graphs or charts.
22. As an instructor, I would like to manage multiple classes.
23. As an instructor, I would like to export results into a spreadsheet.
24. As an instructor, I would like an easy-to-use interface.
25. As an instructor, I would like to be able to access the system through a web browser.
26. As an instructor, I would like to prevent my students from cheating or committing other types of academic dishonesty.
27. As a developer, I would like a method of developing test cases for React Native code.
28. As a developer, I would like a consistent method of checking out branches and making pull requests to the master branch.
29. 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.
30. As a developer, I would like to gain user feedback from an anonymous feedback feature that users and professors can use.
31. As a developer, I would like to know when our servers are reaching capacity.
32. As a developer, I would like to create a system that is scalable to handle fluctuating demand.

Non-Functional Requirements

Architecture and Performance
We would like to be able to handle requests from an entire class simultaneously so we can show live results for every phone and for the professor. The largest lecture halls at Purdue seat around 500 students. We will need a system that can scale quickly to handle high traffic during quizzes. Using either MongoDB or AWS with autoscaling features setup, we can make it easy for servers to ramp up capacity and drop off as needed.

Security
We will also need to use some kind of authentication service to keep user data secure and prevent professors' quizzes from being hacked. Both Google and Amazon provide authentication tools that take the heavy lifting out of holding sensitive user data. We will also give professors the ability to set permissions on anyone in their classes to ensure that no one has access to something they should not.

Usability
For this app to be successful, it must be easy for not only students to setup and access their accounts but also easy to join a quiz and take within a small amount of time. It will be useless if students can not easily login and take the quiz within the time frame that is allowed in class. Also, it must be easy and efficient for professors to setup their quizzes and not waste time figuring out the UI of our application.

Hosting/Development
Since we are building our frontend in React Native, it will be easily to deploy updates simultaneously on both iOS and Android as well as our web application. We plan on either using AWS and/or MongoDB to host our website and database. For our backend, we plan to use Node.js that can be maintained independently from the frontend.