Quizas

Flash Card Game

Progress Report 1

CS3216 Final Assignment
Team 4

Richard, Tack Kian, Yatish, Yuanda

Application Prototype

Figure 1. Start Page

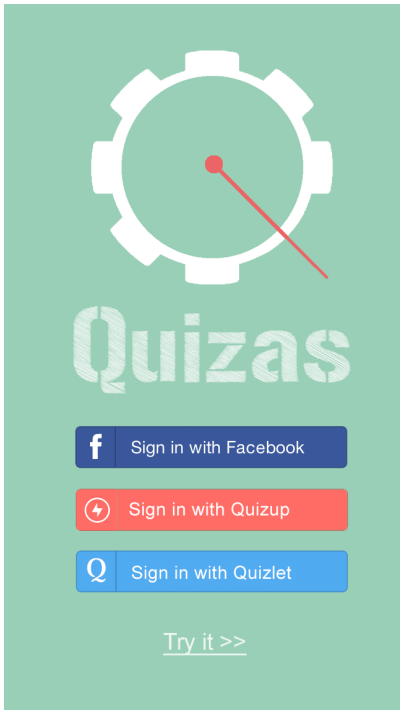


Figure 2. Main Page v1

Figure 3. Main Page v2

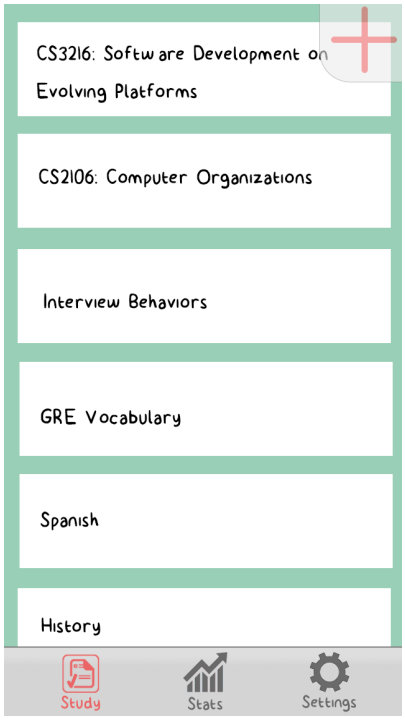


Figure 4. Game Page

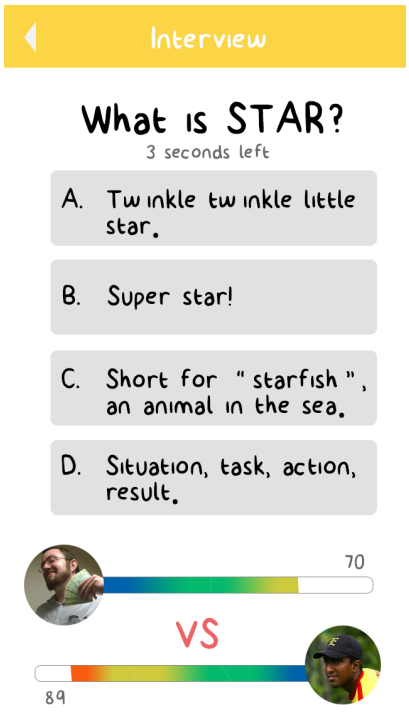
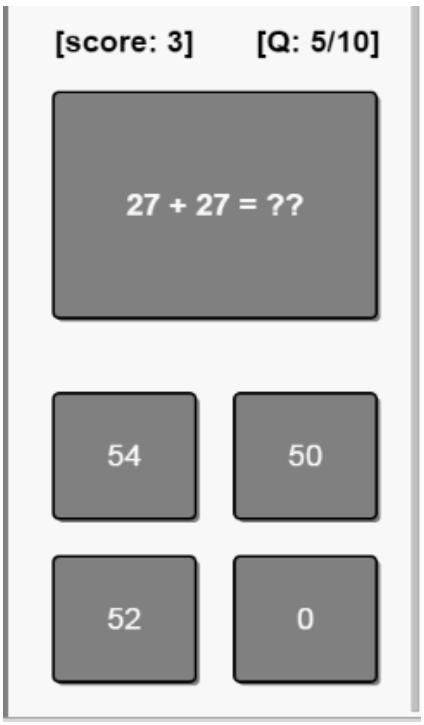


Figure 5. Study Page

More: http://bit.ly/QuizasMockup



Previous Project Schedule: Milestones & Timeline

Week 7	29 Sep - 05 Oct	- (Done) Discuss flow of application and UI. - (Done) Prepare wireframe mockups. - (Done) Understand Quizlet API.
Week 8	06 Oct - 12 Oct	- (Done) Refine UI design after UI/UX meeting. - (Done) Contact target users and integrate discuss product with them. - (In Progress) Begin application development in scrum cycles of 4-5 days.
Week 9	13 Oct - 19 Oct	- (Delayed) Have a working MVP by end of the week. - (Delayed) Rope in additional users/testers.
Week 10	20 Oct - 26 Oct	- Receive feedback on the working application by the contacted customers. - Improve application based on received feedback. - **21st October - Progress Report 1** - Start work (UI and backend) on additional features like rankings, multiplayer support. - Begin aggressive marketing of application.

State of Progress & Problems Encountered

For the development of the backend component of this project, we have decided to use a Python micro web-framework called Flask. Due to the lack of experience of the team using this framework, the work so far has been exploratory, investigating feasibility.

For the game we would like to implement web-sockets to support communication, this is so that we can facilitate “live” interaction between two users of the app, enabling them to compete at the same time with these flashcards, like a multiplayer game. Using web sockets, we have been able to simulate rooms which contain the two players. Data can be communicated between the players who are part of a room. However, the problem currently pertains to avoiding an intruder who isn’t part of this two-player room from communicating with the server. We also require each client to be identified with the server, for which a handshaking method needs to be implemented between the client and server.

While our front-end designers have experience using jQuery Mobile, due to the general rigidity of jQuery Mobile we have decided to use jQuery Mobile for the navigation and general UI, while using the Quintus Engine for the game itself for more creative flexibility. Thus there was also time invested into learning and understanding the Quintus API, which resulted in a slower-than-anticipated progress.

With regards to the implementation of the actual game, as mentioned earlier, the Quintus Engine is used to render the game, and at this time of writing a functional, solo-play quiz skeleton is built that is also ready to accept server data (but currently accepting local test data). The skeleton UI for the rest of the app (menu, navigation, other pages, etc.) has also been completed.

Some investigation was also done into Chef as a server provisioning tool. While Chef and such DevOps tools are quite cool, maybe work on the actual application could have been more of a priority.

Contingencies

In summary, the setbacks we’ve had have been due to new technologies, and perhaps not working hard enough to keep to the deadlines we set for ourselves. Part of this is failing to meet promises of doing work, and only working efficiently when we meet together. It’s good that we’re trying new things, but at this stage we’re through the woods about this. To ensure that further deadlines would be adhered to, we plan to meet more frequently for these team sprints.

New Insights

We haven't made changes to the application. Worth noting, though, is the new understanding that we're targeting this application at students, and that the UI plays a supremely important part in how "good" our app is to be. An alternative was brought up for discussion that we could alternatively try targeting a revision app for lecturers, providing a service of analytics and feedback. This idea was shelved due to a lack of time, as well as our decision to focus on a core focus for our app: which is that of a quiz game.

Updated Project Schedule: Milestones & Timeline

Week 10	20 Oct - 26 Oct	- Frontend coherent skeleton by 22 Oct 2359. - Rope in additional users/testers. - Iterate on UI/UX Design. - End of week, front-end and back-end efforts integrated with each other.
Week 11	27 Oct - 02 Nov	- Improve Automated Testing. - Continue implementation of additional features. - Begin aggressive marketing of application. - **29th October - Consultation**
Week 12	03 Nov - 09 Nov	- Continue implementation of features and keep improving application. - Bug fixes. - Continue selling the application to new users and maintain contact with target group and receive feedback.
Week 13	10 Nov - 16 Nov	- Fix bugs. - Iterate! Iterate! Iterate! - Do not stop marketing! - **3rd November - Progress Report 2 (Oral)**
Reading Week	17 Nov - 21 Nov	- Polish! Ensure the project is presentable!! - **19th November - STEPS** - **21st November - Final Report**