**Pharmacy Quiz Game Proposal**

February 3, 2017

CS1980 Capstone

Mike Bowen

Ariella Hanna

Francis Mangornong

Adam Mitro

Kenny Morse

|  |  |
| --- | --- |
| **User Stories** | **Rank** |
| As a stakeholder  I want to host PharmGenius on Pitt’s servers  So that user traffic is less limited | 1 |
| As a developer  I want the data for the application stored on a custom MySQL database  So that I have greater control over the data used for the application | 2 |
| As an administrator, I want validation on Category additions, So that less cleanup of duplicate entries will be required | 3 |
| As an administrator, I want questions to be able to be included in multiple quizzes/categories, So that duplicate questions can be prevented. | 4 |
| As an administrator  I want to be able to control user accounts  So that others that I choose can perform administrative tasks in my stead | 5 |
| As a stakeholder  I want the application to be portable  So that it may be hosted on other servers in the future if it should be required | 6 |
| As an administrator  I want data from the prior version of the application migrate to the new database  So that potential data loss will be kept to a minimum | 7 |
| As a user,  I want to be able to sign in with a non-Gmail e-mail,  So that I do not have to create a new e-mail if I don’t already have a Gmail account | 8 |
| As a quiz taker, I want to be able to add an explanation to a question that does not have one, So that future quiz takers will have more detail about the questions they answer | 9 |
| As a quiz creator, I want to be able to search the available questions, So that if my question already exists I can include it in the quiz | 10 |
| As a quiz taker,  I want to see percentages of how other people answered questions,  So that I can compare myself to other students and see how difficult questions are | 11 |
| As a quiz taker, I want to be able to undo my vote, So that if I change my mind I can leave neutral feedback on a question. | 12 |
| As a stakeholder,  I want PharmGenius to work on Internet Explorer,  So that the application can reach as broad an audience as possible. | 13 |
| As a user, I want to be able to sort the leaderboard, So that I can see leader statistics based on criteria I specify. | 14 |
| As a quiz taker,  I want to be able to take quizzes with multiple categories at once,  So that I can do all of my studying at one time | 15 |
| As an administrator, I want users to be able to have a friend list, So that a sense of community can be established around the application | 16 |
| As a user,  I would like to be able to view a user’s profile by clicking on their name on the leaderboard,  So that I can quickly look at their stats or add them as a friend | 17 |
| As a quiz taker, I want to be able to earn achievements/badges, So that I have another incentive to keep taking quizzes | 18 |
| As a user,  I want to be able to sign out of PharmGenius without signing out of Google,  So that I don’t have to log back in to all of my accounts when I am done playing | 19 |
| As a developer,  I want a default user icon to display if the user has not uploaded a photo,  So that the website looks more professional | 20 |
| As a user,  I would like to sign in with Pitt Passport,  So that I do not have to create a new account | 21 |

**Overview**

Our group will be working on PharmGenius, an instructional pharmacy quiz game made primarily for students. Students can choose quizzes with questions catered to specific exams in their courses. PharmGenius is currently hosted on Google Cloud Platform, but our goal is to move it to University of Pittsburgh’s unix-based servers. We also want to add a new quiz section to prepare students for the Pharmacy Law exam, make some changes to the UI, and add more administrator options.

**Communication**

Our team will communicate mostly using GroupMe, but will send e-mail for more formal conversations. We will also be using Github for version control. We will be meeting bi-weekly with our stakeholders, Dr. Ravi Patel and Jonathan Velez, in person and sending email updates in between. After each meeting, we will send a summary which details what we discussed, what we plan to do before the next meeting, and what we need from our stakeholders. In addition to these summaries, we will be sending more in-depth updates at the end of each sprint, which are two weeks long.

**Languages and Technology**

PharmGenius is an existing application, so we will be using the same languages and frameworks that the original creators used. This means that will be using Bootstrap for our php framework. For the backend, the previous teams used Python, so we will be as well. Since we will be migrating from Google, we will have to create our own database using MySQL. The big change will be the server that is actually hosting the website, since it is Unix-based and will have its own databases and whatnot.

**Current State of the Application**

The current version of the PharmGenius web application is setup with a Model-View-Controller architecture that uses HTML and Javascript for the frontend, Google App Engine (cloud-based) for the backend, and Python as the go between. The application is currently functioning; however, the stakeholder has become dissatisfied with the reliance on Google App Engine. With the current version of the application, a user must possess a Google account and that account must be used to login to the system. There are issues that stem from this processes, including the fact that logging out of the PharmGenius application logs the user completely out of Google. Furthermore, because all of the data is stored on the Google cloud, administrative users and the stakeholders do not have the ability to oversee and maintain the data in way that is conductive to allow application to thrive. Case in point: as of this writing, our group does not yet have access to the project on the Google App Engine, as the owner of the project is someone from one of the prior groups. This has led to some difficulties in determining the outstanding architecture and what can be done to improve it.

The current flow of the application, in a broad sense, proceeds as follows. When the user first accesses the application via its URL, or when the user navigates to a different page on the site, the request is intercepted by the Python-based control code. This code maps all of the individual pages on the site to certain functions within the code. These functions are responsible for retrieving the appropriate data from the Google App Engine and building appropriate data objects that the pages can use. Once the data objects are built, they are supplied to the page and the page is rendered. To perform these functions, the control code relies upon a number of Google and Python APIs that abstract away a significant portion of the backend. This may have allowed for faster development, but has sacrificed too much flexibility.

**Project Plan**

The application needs to be migrated off of Google App Engine. This will allow the stakeholder greater control over the project, as well as allowing the application to be deployed wherever the stakeholder may choose. Some examples of that control would be to allow for the administrators to control user accounts, review and add/delete/correct questions, and fix any possible issues that may arise. Some of these features are implemented, but the fact that administrative accounts need to be flagged through the Google App Engine is unsatisfactory. Also, the use of Google App Engine opens the application to anyone with a Google account, which may expose security and data volume concerns. Making these changes will require a significant amount of work on both the frontend and backend.

For starters, a custom standalone database will need to be created. For this we will rely on MySQL and phpMyAdmin, as these are proven technologies that are just about as portable as one can expect. Most webhosting services come with these technologies standard, and deploying on a Pitt-based Linux server should also be fairly simple. One task that we hope to be able to do is to migrate the existing data off of Google App Engine and onto the new database. It is yet to be seen if this will be possible, however.

We will need to rebuild to the control layer away from Python. There may be options for keeping a Python-based structure, but it would rely on outside APIs and be less portable. To do this, we will basically need to rewrite the entire control code in PHP. This will be the appropriate choice as using PHP to interact with a MySQL based database is fairly simple. Rewriting the controller may not be the most difficult thing ever, but it will require a lot of time.

To support the controller, a number of database stored procedures and triggers will need to be created. These will facilitate simple PHP calls. For instance, instead of calculating a user’s total score using PHP, we will simply be able to call a stored procedure on the database and supply the username. This should help cut down on the amount of work that needs to be done in the middle layer of the application.

The front end will also need to be adjusted appropriately. With the above changes, the structures of some of the data objects may change slightly, so the frontend will need to be able to handle those changes. In addition, there are bug fixes and feature additions that we will hope to add to the application, and those will require additional frontend work.

**Testing Plan**

We expect a lot of things to break after we move to the new server, so testing will be very important. Creating tests for everything that works before the move will make debugging after the move significantly easier. We will also play with the quiz game as it is right now to see if there any bugs before we start anything. Once we fix everything that might break from the move, we will use test-driven development to implement new features. Since the website has active users, we can continuously get feedback if any bugs are found as well.

**Division of Labor**

For this project, we are essentially divided into working on either the front end or back end. Mike, Adam, and Ariella will be working on the back end and Francis and Kenny will be working on the front end. We put more team members on the back end because moving to a new server will be back end heavy, at least initially. After completing the move, we will reevaluate and potentially move some team members over to the front end.

**Potential Issues**

Moving from the Google app engine to Pitt’s servers is likely going to cause a lot of complications. We will be replacing the Python code with PHP, a language that is new to many of us, so that will require a lot of learning on the job. Hosting on Google abstracted a lot of the server side from the developers, so we will have to be implementing a lot of that on our own. If we are unable to get the Google development access credentials, then we will have to be doing a lot from scratch. Another issue to work around is that current user accounts are Google accounts, so another type of account will have to be made and data may not be able to transfer over.