Introduction for PonyPark

Our product allows users to rate how much parking is available and provides that data to others users. This document specifies the features that will be functional at the end of the second iteration.

Second Iteration Requirements

The below features will be developed on the website version of our product, which we have designated the full version. We will also implement the list view, the user's ability to view and add favorites, and the user's ability to submit a rating of a garage on our Android App.

Task: Improving Aesthetic Quality of Website
Responsible: Evan Kohn
We will alter the color scheme, box sizes and types, and layout of the page to make it more aesthetically pleasing.

Task: Official Rating
We will improve the rating display system from the first iteration. Instead of showing just the most recent rating, we will get averages.

Responsible for GUI: Evan Kohn

- On the home page, for each garage the website will display the average/last rating for the whole garage.
- On an individual garage's page, the website will display the average/last rating for each level, as well as the average/last rating for the whole garage.
- Display the appropriate message depending on whether there were any ratings during the past 2 hours.
    - Example if there were ratings:
        - Average capacity in past 2 hours: Full
    - Example if there were no ratings:
        - Most recent rating: Some

Responsible for Database: Jordan Kayse, Jessica Yeh, and Story Zanetti

- The query to obtain ratings will be modified to:
    - If there are ratings for the garage from the past 2 hours, then average the ratings from the past 2 hours and return the average.
    - If there are no ratings for the garage from the past 2 hours, then take only the most recent rating and return it.

Task: Add Pictures of Garages
Responsible for GUI: Evan Kohn

- On each garage's page, a picture of that garage will be displayed if there is one currently stored in the database.

Responsible for Database: Jessica Yeh

- Ensure that the picture file names will contain the IDs of the garages so that they can be associated with each other.

Task: Google+ Signup Feature
Responsible for GUI: Evan Kohn

- Add a "Sign in with Google" button to the Login box at the top of the website.
- Add a "Sign in with Google" button to the Signup page.
- When signed in, the user's Google+ name will be displayed at the top of the screen in the "Welcome back to PonyPark" box.

Responsible for Database: Jordan Kayse, Jessica Yeh, and Story Zanetti

- Utilize the Google API to authenticate the user.
- If authentication is successful and there is not already a row in the Users table that corresponds with the Google ID, then create one and populate it with as much information as possible (first name, last name, email, phone number). The ExternalType in the Users table will be set to Google.
- If authentication is successful, set session variables and redirect the user to the home page.
- If authentication is unsuccessful set the login state equal to false.

Task: Facebook Signup Feature
Responsible for GUI: Evan Kohn

- Add a "Login with Facebook" button to the Login box at the top of the website.
- Add a "Login with Facebook" button to the Signup page.
- When logged in, the user's Facebook name will be displayed at the top of the screen in the "Welcome back to PonyPark" box.

Responsible for Database: Jordan Kayse, Jessica Yeh, and Story Zanetti

- Utilize the Facebook API to authenticate the user.
- If authentication is successful and there is not already a row in the Users table that corresponds with the Facebook ID, then create one and populate it with as much information as possible (first name, last name, email, phone number). The ExternalType in the Users table will be set to Facebook.
- If authentication is successful, set session variables and redirect the user to the home page.
- If authentication is unsuccessful set the login state equal to false.

Task: Add Commute Times Feature
Responsible for GUI: James Lomeo

- Ensure that the user will be able to add a time for a day of the week as a commute time. The user will check the days of the week they want that specific commute time. A "time" consists of an hour, minute, and time of day (am/pm).

Responsible for Database: Jordan Kayse, Jessica Yeh, and Story Zanetti

- One or more rows will be inserted into the CommuteTimes table in the database depending on the user inputted data with each day of the week being a unique row. The current user's ID, received from the session, will be used for the UserID.

Task: Delete Commute Times Feature
Responsible for GUI: James Lomeo

- Ensure that the user will be able to remove a commute time he or she has created.

Responsible for Database: Jordan Kayse, Jessica Yeh, and Story Zanetti

- The row corresponding with the CommuteID from the user's selection will be deleted from the CommuteTimes table.

Task: Edit Commute Times Feature
Responsible for GUI: James Lomeo

- The user will be shown every commute time for each day of the week.
- The user will be able to edit the commute time.
- Upon hitting "Update," the user will be shown a page confirming his or her changes.

Responsible for Database: James Lomeo

- Ensure that the database will send all the current commute times the user has created and to which day of the week those times correspond.
- Ensure that all changes will be updated in the database.

Task: List Commute Times Feature
Responsible for GUI: James Lomeo

- The user will be shown every commute time for each day of the week.

Responsible for Database: Jordan Kayse, Jessica Yeh, and Story Zanetti

- The database will return a list of all the commute times for the current user. This list will contain the CommuteID, Time, Day, and WarningTime for each commute time in the CommuteTimes table corresponding with the user's ID.

Task: Add Favorites Feature
Responsible for GUI: James Lomeo

- A list of the garages will be given to the user. If the user does not have a particular garage in his or her favorites, a link "Add to Favorites" will appear underneath that garage's address.
- Clicking on this link will add the garage to the user's favorites with this priority, display a success message, and keep the user on the Edit Favorites page.

Responsible for Database: Jordan Kayse, Jessica Yeh, and Story Zanetti

- A new row will be inserted into the FavoriteGarages table using the ParkingID from the user selected garage and the current user's ID. The Priority will be incremented from the previously most recent favorite.

Task: Delete Favorites Feature
Responsible for GUI: James

- A list of garages will be given to the user. If the user has a particular garage in his or her favorites, a link "Delete From Favorites" will appear underneath that garage's address.
- Clicking on this link will delete the garage from the user's favorites, display a success message, and keep the user on the Edit Favorites page.

Responsible for Database: Jordan Kayse, Jessica Yeh, and Story Zanetti

- The row corresponding to the user's ID, retrieved from the session, and the user selected garage's ParkingID will be deleted from the FavoriteGarages table.

Task: List Favorites Feature
Responsible for GUI: James

- A page will be created that lists the user's favorite garages and their current rating and address.
- From this page the user will be able to click "Edit Favorites," which will allow him or her to add or delete garages to or from the user's favorites.
- If no favorites are set there will be an instruction message to click "Edit Favorites" in order to add favorites.

Responsible for Database: Jordan Kayse, Jessica Yeh, and Story Zanetti

- The database will join the FavoriteGarages table with the ParkingLocations table and will return a list of the information for each garage in ParkingLocations whose ParkingID corresponds with the ParkingID in FavoriteGarages for every entry in FavoriteGarages where the current user's ID corresponds with the UserID in FavoriteGarages.
- The favorite garages are ordered by priority (most recently added is on the bottom).

Task: Creating the Android App with Appropriate Features
Responsible: Justin Trantham

Subtask: Favorites Screen

- Activity which will display the current user's saved favorite garages in a list format.
- From this activity, the user will have the ability to edit or delete his or her favorite's list. Depending upon if the user has not already added the garage to his or her list then an option to "Add to Favorites" will be given to the user on the selected garage page.
- Editing individual favorites can be done after selecting the desired favorite in the list. To add or edit favorites the user must be logged in.

Subtask: List View

- One of the three main tabs which will display to the user all the garages with latest status update.
- Each row of the list will have the name, status, and address of garage.
- Each item in the list will be clickable, allowing the user to be directed to another page which will display information (option to rate, garage info, add to favorites) about the selected garage.

Subtask: Submit Rating

- Once the user has selected the desired garage from the list view, he or she will be shown the garage information page. From this page, the user will have the option to rate the availability of the garage by clicking the button that says "Rate."