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 we will functional at the end of the first iteration.

First 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 create a bare bones version of our Android App, but we will only implement the sign in and sign up functionality on the Android App.

Task: Creating the Database
Responsible: Jessica Yeh

- Create tables for Users, ParkingLocations, Ratings, FavoriteGarages, CommuteTimes, and Requests.
- Add referential integrity to the tables.
    - Ratings references ParkingLocations and Users
    - FavoriteGarages references ParkingLocations and Users
    - CommuteTimes references Users
    - Requests references Users
- The Users table should have unique keys on the Username and Email.
- ExternalType and ExternalID together are also a unique key on the Users table.

Task: Signin Feature (Regular)
Responsible for GUI: James Lomeo

- Ensure that the signin dashboard is on every page.
- A user will type in his or her username and password.
- Ensure that if the user doesn't enter a username/password, authentication fails.
- When a user is signed in, the dashboard should read his or her username.
- Allow user to sign out by clicking "sign out" on the User Dashboard.

Responsible for Database: James Lomeo

- Ensure that when a user successfully signs in, the signin section is replaced with the "user dashboard" on every working page (will require session tokens).
- Ensure that the user can successfully sign out by deleting the session.
- Ensure that when a user enters an incorrect username/password combination, authentication fails.
- Ensure that if a user is signed in, he or she cannot access the signup page.

Task: Signup Feature (Regular)
Responsible for GUI: James Lomeo

- A user will enter a First and Last Name.
- Ensure that the password is at least eight characters long.
- Ensure that the email address is valid format for an email address.

- Ensure that the phone number is only 10 numbers.
- All the form validation will be done using HTML 5.

Responsible for Database: James Lomeo
- Ensure that the signin page is only accessible to users who are not signed in.
- Upon signing up, the user account should be created in the database.
- Ensure that the password is hashed and salted.
- The created user account will be verified and a sign in token will be created so the user does not have to "re-sign in" during the current session.

Task: View Garages Feature (List View)
Responsible for GUI: Evan Kohn and Justin Trantham
- Display all of the garages stored within the main database in a row based format containing:
  - The most recent rating.
  - The name of the garage.
  - The address of the garage.
  - A button to Rate the current capacity which takes the user to the rating page for that garage.

Responsible for Database: Jordan Kayse
- Use a query to get the list of garages by joining the ParkingLocations and Ratings tables and to get the name, location, and the most recent rating of every garage.
- For the first iteration, we will use a the most recent rating for the rating display, but a future iteration will include the average rating of each garage.
- The results will be listed alphabetically by their name.
- Return the result of this query as a JSON object.

Task: View Garages Feature (Map View)
Responsible for GUI: Evan Kohn and Justin Trantham
- The default map view will be focused on the SMU campus.
- Each garage will be shown as a pin, and clicking on a pin will bring up a dialog box allowing the user to see the following:
  - The name of the garage in the top left.
  - The address of the garage below the name.
  - The most recent rating at the bottom left.
  - A button to Rate the current capacity which takes the user to the rating page for that garage.
- Map should be zoomable and pannable.
- The dialog box should disappear when the user clicks on another garage or somewhere else on the map.

Responsible for Database: Story Zanetti

- Use a query for a selected garage to join the ParkingLocations and Ratings tables and get the name, location, and the most recent rating of that garage.
- For the first iteration, we will use a the most recent rating for the rating display, but a future iteration will include the average rating of each garage.
- Return the result of this query as a JSON object.

Task: Rate Garages Feature
Responsible for GUI: Evan Kohn

- Display the garage's name, address, and picture if available.
- Create a form for rating the garage's current capacity on a Full-Empty scale.
- Verify that exactly one level and rating has been selected upon submission.
    - If the parking location does not have levels, the level will be "none."
- Upon submission, redirect to the home screen.

Responsible for Database: Jessica Yeh

- Use the ParkingID from the page the user rated.
- Accept user input that specifies a rating for that garage.
- Associate the rating with a particular level of that garage.
    - If the "none" level is selected, the level is null in the database.
- Use the current time as the timestamp for the rating.
- Insert all the aforementioned data as a row in the Ratings table, and associate it with the currently signed in user's UserID.
- Don't allow people who aren't signed in to be able to rate.

Task: Creating the Android App with Appropriate Features
Responsible: Justin Trantham
For the first iteration, the main goal is to develop a base structure and platform to later build upon. We narrowed down the tasks to the bare bone basics of allowing the users to sign up/sign in and browse blank pages. The functionality to sign up/sign in is the same as that of the website.

Subtask: Signin Screen

- Ensure that a user can successfully sign in, notify user of success or failure.
- Ensure that the signin button is displayed to user on the action bar for every page if not already signed in.
- Ensure that when a user selects "sign out" that they are signed out.
- Ensure that the username and password text gathered from the signin text fields is successfully encapsulated in a JSON object to be verified by remove server verification logic.
- Ensure that both fields have the correct username and password to successfully sign in, otherwise show that authentication failed.
- Ensure that if a user is signed in, the signup page is not shown.
- Ensure that if a user is not signed in, he or she is not allowed to rate garages.

Subtask: Signup Screen

- Ensure that this screen is only displayed if not signed in.
- Ensure that the signup option is given to the user on the homepage if not already signed in.
- Ensure that Name, Email, and Password are displayed as text fields on the signup screen.
- Ensure that all three fields are required to successfully sign up, otherwise display an error next to the text field that is missing or incorrect.
- Ensure that if all fields are properly filled, a message stating signup was successful is displayed.
- Ensure that the signup option is displayed on the action bar if the user is not already signed in.

Subtask: Blank Pages

Action Bar (Navigation bar below logo towards top)

- Ensure that the bar has options to allow the user to navigate to other pages.
- Ensure that the bar displays the option to sign in or sign up.

For all the following pages:

- Ensure that the portrait orientation is the only orientation allowed.
- Ensure that the action bar with action items is shown below the logo horizontally.
- Ensure that the logo is properly positioned on the top left of page.

Home Page Activity

- Ensure that signup/signin is shown.
- Ensure that there is a space to represent the interactive map.

Report Availability Activity

- Ensure that placeholders are shown to represent the ability to rate parking.

ListView Activity

- Ensure that there is a row based list of placeholders where the ratings would be.

My Account Activity

- Ensure that there is a scrollable list of blank placeholders which represent the user’s account settings and preferences.