ACE ASE Revised Project Proposal:

MyPlace

Group Members:
Michaella Schaszberger (mls2290)
Julia Sheth (jns2157)
Gabi Munoz (gmm2172)
Francesca Callejas (ffc2108)

Part 0: Language and platform

We will be using Java on Eclipse. We will import and use JavaFX to implement the GUI to go with our program.

Part 1: Proposed application

The software engineering project that we plan to work on is a command line tool that displays a pop-up display to save places that you have been to that you liked and places that you want to go to in the future. This application will allow for people to more easily save their travel recommendations and travel wishlist with a model more interactive and engaging than a traditional list.

When a user first opens this command line tool, she will have to create an account. When she is done using the application, she can log out of this account. The next time she uses the application she will have to sign back in, and all of her information will be saved. We will use a database (mySQL) to store users' inputted information with their account logins. The way the login will work is that there will be a command line prompt asking the user if she wants to sign in, and if she says yes then a pop-up display will open where the user can enter her credentials and view the rest of the application.

Once the user has an account and is logged on to the program, she will see two tabs in the pop-up display. One tab will be for all of the places she has been to and wants to save and the other tab will be for all the places she wants to visit. In each tab, there will be a plus button that will allow the user to add a place. She will type in the name of the place, and a list will come up of possible suggestions of real places that she could be referring to. We will implement this capability by using the Google Places API or the FourSquare API which provides rich entity information cards for each possible location. Then she will click "Add" and this place will be added to the tab that she's in. Additionally, when she adds a place to one of the tabs, she will be able to add a comment for the place so that when she goes back to look at it later, she will remember why she likes it or why she wants to visit. When she comes back to that tab later, she will see all the places that she added and her comments on those places.

Part 2: User Stories

User logging on: As a <user>, I want <to log in> so that <I can access my personal page>. My conditions of satisfaction are <a prompt that indicates whether I would like to login in and then a pop-up login page where I can enter my username and password, a sign up link to create an account, and a "forgot password?" link that will send an email containing a temporary password>.

User pinning previous places: As a <user>, I want <to be able to add locations that I have visited> so that <I can compile a map and list of places I've been>. My conditions of satisfaction are <a comprehensive list of all locations inputted by the user that are linked to a real location on the map (using a Maps API), which will produce an interactive map with all the places I've been and each location's relevant information>.

User commenting on places: As a <user>, I want <to be able to see my comments on previous places I visited> so that < I am able to remember what I liked/disliked about each place>. My conditions of satisfaction are <binding my comments to a distinct pinned location so that when I view it, I can easily view my review>.

User pinning future places: As a <user>, I want <to be able to add locations that I would like to visit > so that <I can compile a list of places I'd like to go to>. My conditions of satisfaction are <a comprehensive list of all locations inputted by the user that are linked to a real location on the map (using a Maps API)>.

User categorizing places: As a <user>, I want < to categorize the places I've been> so that <I can easily search for a place according to its purpose and experience>. My conditions of satisfaction are <a comprehensive list of all added locations that represent a specific category. For example, a simple command from the user, "coffee shop," should return a comprehensive list of all coffee shops visited and intended to visit. However, the user has the option of only viewing

coffee shops he or she had intended to visit, instead. The tool should be able to accurately and neatly display the data requested by the user >.

User sees photo of pinned places: As a <user>, I want <to be able to add photos to the locations I have specified on my visited list and wish list > so that <I can visualize these locations>. My conditions of satisfaction are <binding my pictures to a distinct pinned location so that when I view it, I can easily view this location with visuals of the location's food and environment>.

User logging out: As a <user>, I want <to log out> so that <I can safely exit my personal account in this software>. My conditions of satisfaction are <a logout pop-up screen that is easy to access by typing a simple command>.

Part 3: Acceptance Testing

Our MVP user stories are the user logging on, the user pinning previous places, the user pinning future places, and the user logging out.

For the user logging on user story, we would test to make sure that users can correctly create accounts and log into those accounts. For creating the account, the inputs we would test on would be:

1. Using a username that doesn't exist in the database and a password that does exist in the database → Output should be that an account is created
2. Using a username that doesn't exist in the database and a password that doesn't exist in the database → Output should be that an account is created
3. Using a username that does exist in the database and a password that does or doesn't exist in the database → Output should be that the account is not created and user is prompted to use a different username

For making sure that the user can log into her account, the inputs that we would test on would be:

1. Username and password that don't exist in database → Output should be failed login, try another password or username
2. Username that doesn't exist in database, but password does exist in the database → Output should failed login, try another password or username
3. Username in the database using an incorrect password → Output should be a failed login, try another password or username
4. Username and password that do exist in the database → Output should be that user is logged into the corresponding account

For pinning previous places and future places, the inputs we would test on would be:

1. Pinning a location that exists in the Places API → Output should be a rich entity card about the place that gets stored in the previous places list or future places list (depending on which the user was trying to add to)
2. Pinning a location that doesn't exist in the Places API → Output should be a Create Location Field that will allow user to manually input Name of place, Address of place, Hours of operation of place (optional field)

For the user logging out user story, the inputs that we would test would be:

1. Pressing the logout button in the app → Output should be a message that the user has been logged out and if the user tries to open the app again, she is prompted with the login page again
2. Pressing the cancel button of the pop-up or closing the pop-up window→ Output should be a command line prompt dictating "Logout successful."