Gabriela Johnson
PresPrice - Assignment 5
11/10/17
Github Repository: https://github.com/gabrielajohnson/PresPrice

Use Case Descriptions

Create An Account	
Primary Actor	User buying medicine
Goal in Context	To create an account on the site so they can find the cheapest price for their medicine.
Preconditions	User must navigate to web page
Trigger	User clicks on register account
Scenario	1. User enters webpage 2. User clicks sign up link 3. User fills out form 4. User submits form 5. System ensures all needed inputs are validated 6. User is brought to the search page
Exceptions	1. User inputs are invalid: see Validate user inputs
Priority	High Priority; to be implemented as a basic function
Open Issues	none

Log into Account	
Primary Actor	User buying medicine
Goal in Context	To log into a preexisting account
Preconditions	User must have created an account
Trigger	User clicks on log in
Scenario	1. User enters email and password 2. User submits email and password 3. System validates that email and password are valid 4. User is brought to search page

Exceptions	1. User ID and password are invalid: see Validate ID and password
Priority	High Priority; to be implemented as a basic function
Open Issues	none

Validate User Inputs	
Primary Actor	User buying medicine
Goal in Context	Validate the user inputs into the registration fields
Preconditions	Must be attempting to create an account
Trigger	User inputs invalid information into a registration field
Scenario	1. User clicks sign up 2. User enters invalid input 3. System checks if input is correct 4. A red box is placed around the input field 5. User is prompted to re-enter a correct value
Exceptions	none
Priority	High Priority; to be implemented as a basic function
Open Issues	none

Validate ID and password	
Primary Actor	User buying medicine
Goal in Context	Validate the user ID and password
Preconditions	Must be trying to log into an existing account
Trigger	User inputs incorrect information into ID and password
Scenario	1. User attempts to log in 2. User enters incorrect input 3. System checks if input is correct 4. A red box is placed around the input fields 5. Forgot password link appears
Exceptions	none

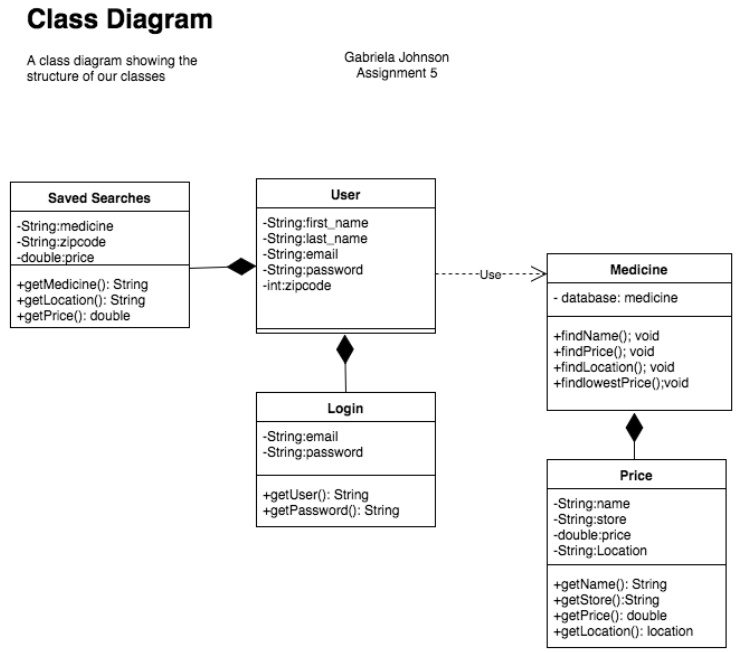
Priority	High Priority; to be implemented as a basic function
Open Issues	How a forgotten password will be handled

Search for Medicine with Account	
Primary Actor	User buying medicine
Goal in Context	To search for medicine with a pre-existing account
Preconditions	User must navigate to web page
Trigger	User enters homepage
Scenario	1. User enters webpage 2. Log in screen user logs in; *See Log in to Account* 3. User searches medicine name 4. Search Results for cheapest prices come up
Exceptions	User inputs are invalid: see Validate user inputs
Priority	High Priority; to be implemented as a basic function
Open Issues	none

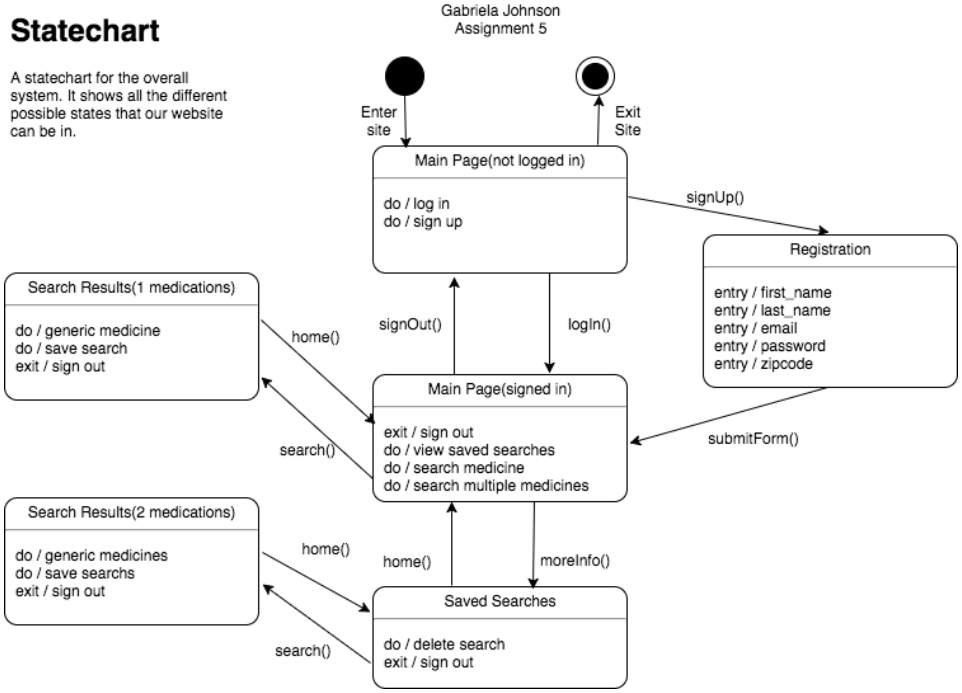
Search for Two Medications with Account	
Primary Actor	User buying medicine
Goal in Context	To search for two medications with a pre-existing account
Preconditions	User must navigate to web page
Trigger	User clicks Log In
Scenario	1. User enters webpage 2. Log in screen user logs in; *See Log in to Account* 3. User searches medicine name 4. Search Results for cheapest prices come up
Exceptions	User inputs are invalid: see Validate user inputs
Priority	High Priority; to be implemented as basic function
Open Issues	none

Save Searches	
Primary Actor	User buying medicine
Goal in Context	To save a searched medication
Preconditions	User must navigate to web page
Trigger	User searches medication
Scenario	5. User enters webpage 6. User search medicine 7. Log in screen user logs in; *See Log in to Account* 8. User searches for medication 9. Search Results displayed 10. User decides to save search which will save lowest price
Exceptions	User inputs are invalid: see Validate user inputs
Priority	High Priority; to be implemented as basic function
Open Issues	none

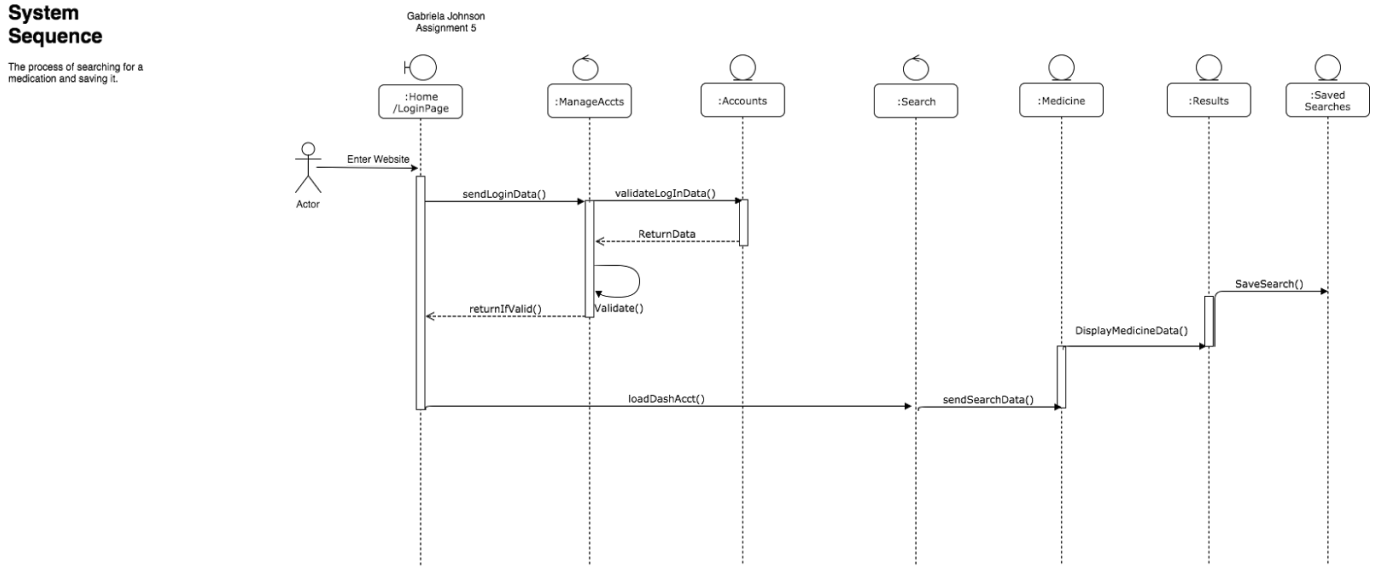
Class Diagram



Statechart



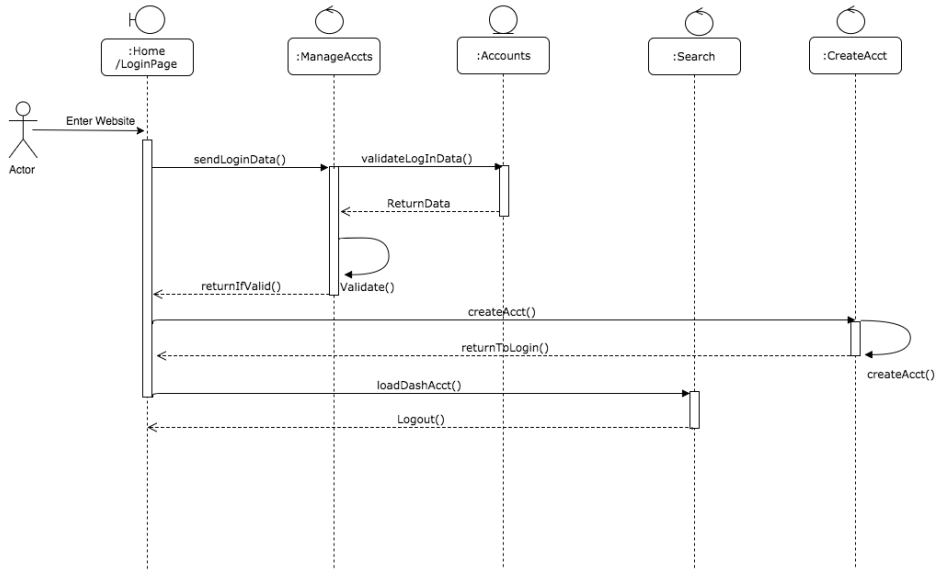
System Sequence Diagrams



System Sequence

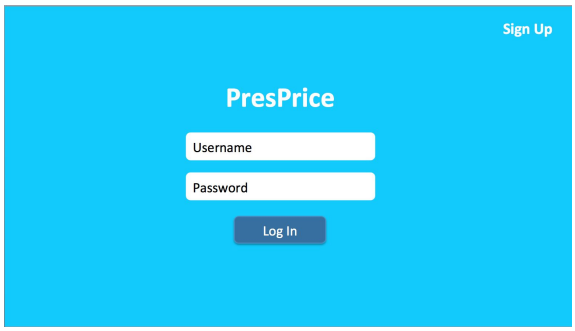
The process of Logging In and Creating an Account

Gabriela Johnson
Assignment 5

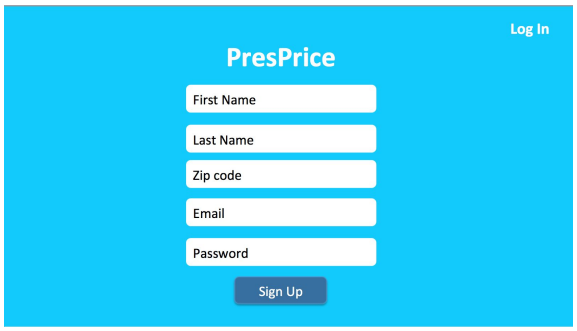


UI Design

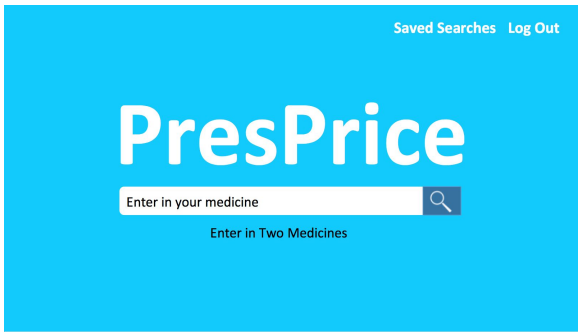
Home LogIn Page



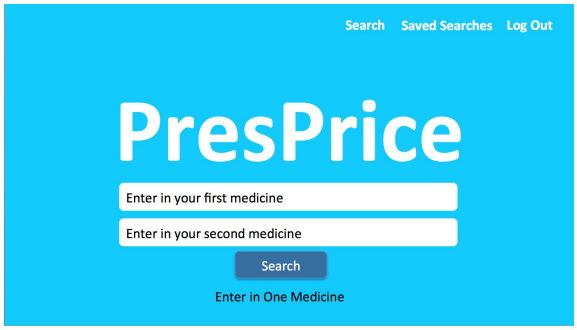
Sign Up Page



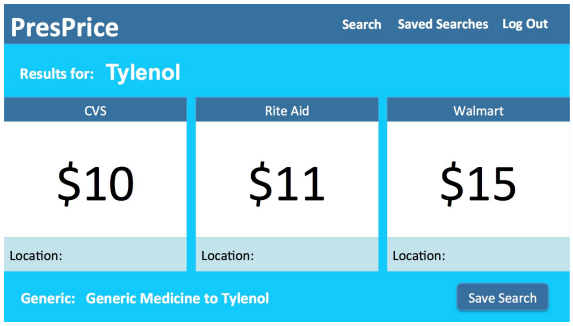
Search One Medicine



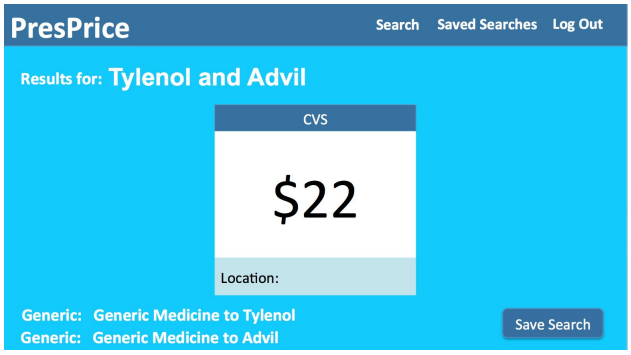
Search Two Medicines



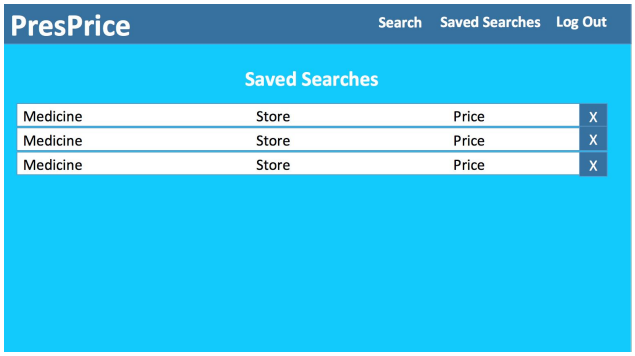
Search Results for One Medicine



Search Results for Two Medicines



Saved Searches



Eight Golden Rules:

1. Strive for consistency

The navigation menu at the top will always have the same options, unless the user is already on the page of one of the options. This menu will remain relatively identical throughout the site. The app will also have a consistent blue color palette and simple and direct interface where the data displayed is visible and easy to understand.

2. Enable frequent users to use shortcuts

There are multiple shortcuts throughout the site. Once the user is logged in they always have the ability to log out, search, or view saved searches available on every page. They also have the generic equivalent of the searched medicine available at the bottom, which can be clicked by the user and will take them to the lowest prices of those generic medications.

3. Offer informative feedback

The buttons on the site will change color when hovered know so the user knows when they are interacting with the site. The user is also able to save searches and will be prompted with a "Saved Search" Overlay Box to let them know that their search was saved to their account.

4. Design dialog to yield closure

After filling out the create account form, the user is able to submit it and is taken to the search page so they know that their account has been accepted and validated. The user is also able to save searches and will be prompted with a "Saved Search" Box to let them know their search was saved to their account.

5. Offer simple error handling

If the user enters incorrect login information, they will be prompted with a red box that states that the information was incorrect and to try again. The Create an Account form will also require all fields to be filled in, if not the user will be prompted to do so once again and will out the invalid boxes.

6. Permit easy reversal of actions

For the Search function, if the user searches something that doesn't exist in the database, they will be told there were no results and they can try to search again. The user can also delete saved searches in the saved searches page.

7. Support internal locus of control

The user can search different medications and save them on their own so they have complete control over what medicine they want to look back on. This is helpful to them so they will always know the location to find their cheapest medicine.

8. Reduce short-term memory load

The user will never have to remember a search they had made because they have the ability to save it to their account to view later. Also, if they click to view the generic alternative to their brand name medicine, they can always click the back button to view the medicine they originally looked for.

Analyzing Requirements

To ensure high cohesion and low coupling between modules, I hope to keep independence between modules low by keeping every function such as login, create an account, search and saved searches in different controllers so they are easier to edit and have less of a chance of affecting each other. For information hiding I will have the passwords hashed.

For my combined low price algorithm, I will be taking the prices from the API and add them together to find the lowest combination within the same store. Only two medicines from the same store will be added together, and the sums will be saved to an array and then iterated through to find the lowest one. This is the one displayed to the user. I will be saving the initial prices from each medicine from each store into an array to ensure efficiency.

I will be using a SQL database to save user information and saved searches. Every time a user will try to save a search, a new column for medicine name, store and price will be saved in their own separate column. The column name for each one will have a numbered ID after it such as name_1, store_1 and price_1 so that each search will remain grouped together and can be

retrieved by the user. As stated before, the passwords will also be hashed when saved into the SQL database for security.

Testing

For unit testing I will first make sure the most basic components of the application work such as account creation and login. After these work correctly I will incrementally add more basic features and test them to make sure those work with the overall app. For integration testing I will be using the Bottom-Up method since this coincides with my method for unit testing. I rather that the login and account functions are completed so I can focus all of my energy into my combined low price algorithm and incrementally integrate that into the app. For system testing I plan on utilizing Usability Testing and Regression testing to ensure my app is functional. I will test the app myself using my previously written use cases as well as test it with others so I can assess the easiness of access and understanding of the app. Regression testing I believe will coincide with me making incremental changes to the app as it progresses, as I will have to test all features of the app after I make a change to the source code.

The testing tools discussed during class and on the wiki page are for languages and systems not related to Ruby except for TestUnit. If I will be using any testing tool it will be this one. I do not plan on using any debugging tools and hope that testing functionality of my code in small increments will help avoid broad errors in my code.

Test Case Design

Functionality	**Inputs**	**Expected Output**	**Actual Output**
Create Account	- Go to site - Click Sign up - Enter information into provided fields - Click submit	Redirected to Search page	
Log in with account	- Go to site - Click Log In - User enters ID and password -Service validates that ID and	After login, user is directed to search page	

	password are valid		
Log In with invalid inputs	-User attempts to login -User enters invalid input -System checks if input is correct -A red box is placed around the input fields	Redirected to Login to re-enter information/forgot password link appears	
Creating account with invalid inputs	-User clicks sign up -User enters invalid input -System checks if input is correct -A red box is placed around the input field -User is prompted to re-enter a correct value	Redirected to Sign Up form to re-enter information	
Search for medication with account	- Go to site - Enter medicine into search bar - Click search	User is redirected to results for medication	
Search for Two Medications with Account	- Go to site -Log In -Click “Enter in two medicines” - Enter each medicine into separate search bar - Click search	User is redirected to results for two medications	
Click on generic medication	-See “Search for medication with account” -Click generic medication	Lowest Price results for Generic Medication will be displayed.	
Click on generic medication out of two medications	-See “Search for medication with account” -Click generic medication	-Lowest Price for generic medication will be combined with other brand name medication to display lowest price at same store	
Click on generic medication but want to return to previous medication	-See “Search for medication with account” -Click generic medication -Click provided back button	-Will be returned to previous medication that was searched for	

Search returns no result	- Go to site - Enter medicine into search bar - Click search	User will be told that there were no results and asked to search again	
Save Searches	-Go to site -Enter medicine into search bar -Click Search -Click Save Search	Search Result is saved to saved searches for user	
Delete Searches	-Go to site -Log In -Click on Saved Searches -Click the “X” next to the saved search to be deleted	-search should be deleted	