CPSC 310 Assignment 7: Demo Script

Team Tau: Norman Sue, Brandon Chang, Jeffrey Gamage

http://www.foodsorce.appspot.com

Sprint 1 User Stories

As a user I want to sign in to the app using my Google account so I can use the app according to my personal preferences.

Priority: 1 Acceptance Criteria:

- Verify that when the `Sign In` button is pressed, the user is redirected to a Google Accounts sign in page.
- Verify that when the user signs in correctly, the page displayed has additional buttons for accessing their personal account settings.
- Verify that when the user successfully signs in, that the browser is redirected to the main app page.

Story Points: 5

Task Breakdown:

- Create the login RPC service which includes the `LoginInfo`, `LoginService`, and `LoginServiceAsync` classes. (Issue #23)
- Add all user preference fields to the `LoginInfo` class. (Issue #24)
- Add a `Sign In` button widget and associated client-side mouse handler code that invokes the remote login RPC service. (Issue #25)
- Create a method which redirects the logged in user to the main web app page after successfully logging in. (Issue #26)

As a logged in user, I want to be able to have a profile page I can view and other users can see so that I can let other users know who I am and vice versa.

Priority: 2 Acceptance Criteria:

- Verify that the user profile page displays the user's photo, display name, gender, headline, favourite type of food, hometown, and website URL.
- Verify that when a new user account is created, the fields that can be edited in the `Edit Settings` page are left at their default values.
- Verify that the user profile page reflects changes made in the `Edit Settings` page.

Story Points: 3 Task Breakdown:

- Implement UI panel code that displays the user's photo, display name, gender, headline, favourite type of food, hometown, and website URL that are saved in fields of the `LoginInfo` class. (Issue #27)
- Add a UI button widget labelled `Edit Settings` that redirects the user to the edit settings page. (Issue #28)

As an admin, I want to import the data set from the Vancouver data site so our app can use it.

Priority: 1 Acceptance Criteria:

- Verify that when you are logged on as an admin, there is an `Import Data` button that can be used to access the data.
- Verify that when the `Import Data` button is pressed, the application receives the latest version of the data set from the website.
- Verify that the `Import Data` button does not cause any other data (such as account data and review data) to be modified.
- Verify that only admin users can import data.

Story Points: 5 Task Breakdown:

- Add a button widget labelled `Import Data` that is only displayed for admin users. (Issue #29)
- Implement mouse handler code that invokes the `.xls` retrieval method. (Issue #30)
- Implement a `.xls` retrieval method that makes a cross-site request to retrieve the `.xls` file from the Vancouver data site. (Issue #31)
- Implement exception handling code in the case that data cannot be retrieved from Vancouver data site. (Issue #32)
- Add a UI component for alerting the admin user about a thrown exception if data cannot be retrieved from the Vancouver data site. (Issue #33)

As an admin, I want the data to be automatically parsed and cleaned into objects that I can keep stored in a database.

Priority: 1 Acceptance Criteria:

- Verify that unnecessary data is discarded when the data is imported.
- Verify that the parsing and cleaning methods do not discard necessary data.
- Verify that the data are parsed into appropriate fields in the vendor object class.
- Verify that the data is parsed and cleaned into `Serializable` classes.
- Verify that missing fields (such as the name of the food vendor) are filled with appropriate default values.

Story Points: 5 Task Breakdown:

- Create parser class using the Apache POI-HSSF API to parse the retrieved `.xls` file. (Issue #34)
- Implement a method that populates parsed empty cells with appropriate default values. (Issue #35)
- Implement a method that validates parsed GPS coordinates. (Issue #36)
- Implement a method that discards unnecessary text from the `.xls` file (location column and pending vendors). (Issue #37)
- Write JUnit tests for various input cases based on the `.xls` file. (Issue #38)

As an admin, I want the most recently updated data set to persist until a new one is uploaded.

Priority: 1 Acceptance Criteria:

- Verify that the data set is still accessible after logging out and logging back in.

- Verify that all users can access the stored data set.
- Verify that the data set is completely overwritten if a new data set is imported.
- Verify that no other data such as ratings or accounts is overwritten when a new data set is imported.

Story Points: 3 Task Breakdown:

- Create a method that iterates over each row of the parsed `.xls` file and instantiates `Vendor` objects for each row of the parsed and cleaned `.xls` file. (Issue #39)
- Create a `Vendor` class that implements `Serializable` to store each vendor object and ensure that it uses JDO annotations to specify persistent fields and uses a latitude/longitude library. (Issue #40)
- Create a server-side class that `extends RemoteServiceServlet` and makes calls to the JDO API for persisting vendor data. (Issue #41)
- Write JUnit tests that ensure that data are correctly stored in the `Vendor` class. (Issue #42)

As a user, I want to plot my location on a map so I can find my way around to the food vendor.

Priority: 2 Acceptance Criteria:

- Verify that a marker representing the user's location is displayed on a map.
- Verify that the user can either enter their current location into a text field or that their location is determined from their browser.
- Verify that the application correctly translates the location into GPS coordinates.
- Verify that the user's map marker changes when the user enters a new location into the text field or the location reported by their browser changes.
- Verify that when a user enters a location outside of Vancouver, that they are alerted that their location is outside of Vancouver and must re-enter a location within Vancouver.
- Verify that when no user location is found, the plot has a default location of downtown Vancouver.

Story Points: 8 Task Breakdown:

- Using the Google Maps Library for GWT to create a basic Google Maps plot on the main web app page. (Issue #43)
- Add Google Maps API code that attempts to get the user's location from the browser using HTML5 geolocation. (Issue #44)
- Add a text box to the Google Maps API code that uses the geocoding service to convert the text input into the user's address. (Issue #45)
- Add a map marker that to the map at the user's location. (Issue #46)
- Add code that refreshes the map when the user's location changes. (Issue #47)
- Add a method that checks whether the user's address is within Vancouver, and alerts the user to re-enter a location if it is outside of Vancouver. (Issue #48)
- Add code that centers the map and plots a user map marker in downtown Vancouver by default when no user location is found. (Issue #49)

As a logged in user, I want to search and filter food vendors near my location so I can go and eat somewhere nearby.

Priority: 2 Acceptance Criteria:

- Verify that a logged in user is presented with search radius options of 2 km/5 km/10 km.

- Verify that when the logged in user selects a particular search radius, that only food vendors within that distance of the logged in user's current location are plotted on the map.
- Verify that when search results are displayed on the map, that they are also displayed in a table alongside the map, with a summary of the food vendor's information, including: name, location, description, and average rating based on user reviews.

Story Points: 8 Task Breakdown:

- Add a `RadioButton` widget with the search radius options of 2 km/5 km/10 km to the UI code of the main app page. (Issue #50)
- Implement a method that filters out `Vendor` objects that are not within the radius selected by the user. (Issue #51)
- Add the Google Maps API code that plots markers on the map plot for the filtered `Vendor` objects. (Issue #52)
- Add a `VerticalPanel` to the UI of the main app page to display the filtered `Vendor` objects. (Issue #53)
- Add code that displays information for each `Vendor` object (name, location, description, and average rating based on user reviews). (Issue #54)
- Add a `Button` widget labelled `Search` that refreshes the page and produces a new Google Maps plot and a populated `VerticalPanel` based on the search radius. (Issue #55)

Sprint 2 User Stories

User stories and demo sequence that Norman will present

As a user I want to be able to set and change my user profile settings so I can personalize my account.

Priority: 2 Acceptance Criteria:

- Verify that when the `Edit Settings` button on the user profile page is pressed, that the browser is redirected to a page where the following fields are text input fields that can be edited: display name, gender, headline, favourite type of food, hometown, and website URL.
- Verify that the `Edit Settings` input page also shows a `Save Changes` and a `Cancel` button at the bottom.
- Verify that the `Edit Settings` page populates the text fields with the previously saved settings.
- Verify that when text fields are changed, and the `Cancel` button is pressed, that no settings are actually changed.
- Verify that when text fields are changed, and the `Save Changes` button is pressed, that the settings are saved and displayed in the user's profile page.
- Verify that when either of the `Save Changes` or `Cancel` buttons are pressed, the browser is redirected to the user profile page.
- Verify when the user logs out and logs back in, that the above settings are persisted.

Story Points: 5 Task Breakdown:

- Modify the `Profile` object to contain a `HashMap` of settings.
- Implement a `FlexTable` to store the settings.
- Add an `Anchor` to change the `FlexTable` to show `TextBox`s for editing and submitting settings values.
- Add a `Submit` button that makes an RPC with to the `setProfile` method for persisting the settings.

Demo Tasks:

1. Click on the `Profile` navigation link
2. Click on the `Edit Profile` link
3. Add some text to various `TextBoxes` that aren't `Search Distance` or `Search Text`
4. Press `Submit`
5. Log out.
6. Log back in.
7. Click on the `Profile` navigation link to show that the changes persist.

As a logged in user, I want to store my search settings so that I can use them again in when I log back into the app in the future.

Priority: 2 Acceptance Criteria:

- Verify that when a user signs up for a new account, that the default search filter preferences are selected. Default values: `searchDistance = "all"` and `searchText = ""`.
- Verify when the user logs out and logs back in, that the previous search filter preferences are restored.

Story Points: 3 Task Breakdown:

- Add fields to `Profile` for storing search settings. (Issue #62)
- Add method to give search settings to the `VendorListPanel` and the `MapSearchPanel` after the user logs in. (Issue #63)

Demo Tasks:

1. Click on the `Admin` navigation link.
2. Click on the `Delete User` link for current User.
3. Log out.
4. Log back in to create new profile.
5. Click on the `Profile` navigation link to show Default values: `searchDistance = "all"` and `searchText = ""`.
6. Click on any radio button and input any string into serachText.
7. Log out.
8. Log back in.
9. Click on the `Profile` navigation link to show that the changes persist.

As a logged in user, I want to be able to search vendors by the kind of food they sell so that I can go and purchase food I like eating.

Priority: 2 Acceptance Criteria:

- Verify that only vendors whose type of food contains the search text appear in the `vendorTable`.
- Verify that the `MapSearchPanel` updates when the `Search` button is pressed to only display markers that (a) meet the distance radio button selected (b) meet the search text.

Story Points: 2 Task Breakdown:

- Update the `VendorListPanel` and `MapSearchPanel` classes to contain fields `allVendors`, `nearbyVendors`, and `matchingVendors` and appropriate coordinating method calls. (`nearbyVendors` match only the distance setting; `matchingVendors` match the distance setting AND the search text.) (Issue #67)
- Create a `TextBox` and a `Button` at the top of the `VendorListPanel` with a `ClickHandler` that filters the `nearbyVendors` to produce an `ArrayList<Vendor> matchingVendors`. (Issue #68)
- Add a method call that sends the `matchingVendors` from the `VendorListPanel` to the `MapSearchPanel` after the `Search` button is pressed. (Issue #69)

Demo Tasks:

1. Click on the `Find Food` navigation link.
2. Input "Hot Dog" in VendorSearch Field.
3. Press Search
4. Look at VendorPanel
5. Press 5km radio button.
6. Look at VendorPanel

As a logged in user, I want to be able to see all of the information and reviews for a food vendor after I have selected it so I can find out more about it.

Priority: 2 Acceptance Criteria:

- Verify that when the title of a food vendor in the vendor list is clicked, that the browser is redirected to the food vendor's information page.
- Verify that the food vendor's information page contains it's name, location, description, average quality, average cost, and a list of all reviews at the bottom (with newest reviews at the top of that list).

Story Points: 3 Task Breakdown:

- Create a `VendorInfoPanel` that displays the fields from the `Vendor` object. (Issue #58)
- Add a `ClickHandler` to each vendor title cell in the `VendorListTable` that switches the display to the `VendorInfoPanel`. (Issue #59)
- Add code that displays a `ViewReviewsPanel` at the bottom of the `VendorInfoPanel`. (Issue #60)
- Add a `Add Review` button at the bottom of the `VendorInfoPanel`. (Issue #61)

Demo Tasks:

1. Click on the `Find Food` navigation link.
2. Click on any Vendor name.
3. Look at VendorInfoPage.

User stories and demo sequence that Brandon will present

As a user, I want to view vendor info from Google Map markers that I click on so that I can learn more about vendors based on location.

Priority: 3 Acceptance Criteria:

- Verify that when a map `Marker` for a particular food vendor is clicked, that the center `Widget` of the `DockLayoutPanel` changes to the `VendorInfoPanel` for the clicked `Marker`.

- Verify that when a map `Marker` is clicked, a dialog pops up for that vendor showing the vendor name (that is a link to display the `VendorInfoPanel` for that vendor) type of food, and location text. If there is at least 1 review, the popup dialog should also display the number of reviews, the average quality rating, and the average cost rating.
- Verify that when a map `Marker` has been previously clicked, and different map `Marker` is clicked, the dialog that was previously open for the first `Marker` is closed and a dialog for the second `Marker` is opened.
- Verify that when a map `Marker` has been previously clicked, and the same map `Marker` is clicked, the open dialog closes.

Story Points: 5 Task Breakdown:

- Create a click handler method for all the markers which changes the centre panel to the `VendorInfoPanel` when a marker is clicked. (Issue #78)
- Set markers options to include and grab number of reviews, the average quality rating, and the average cost rating and display in marker description. (Issue #79)
- Set marker options to properly set description to false when another click handler method is called on a marker. (Issue #80)
- Create a method which hides the marker description when the current marker is clicked on. (Issue #81)

Demo Tasks:

1. Click on the `Find Food` navigation link.
2. Click on any Vendor Marker on Map.
3. Look at VendorInfoPage.
4. Click on same Vendor Marker.
5. Click again on any Vendor Marker on Map.
6. Click on another Vendor Marker.

As a logged in user, I want to be able to save a list of my favourite food vendors so I can remember which food vendors I want to try out.

Priority: 3 Acceptance Criteria:

- Verify that the `VendorInfoPanel` shows a popup indicating whether a vendor is already favorited when clicking `addFavourites` .
- Verify that when the user is not logged in, the `VendorInfoPanel` does not show add favourites button.
- Verify that when the user logs out and logs back in, that the favourites persist.
- Verify that when a new user account is created, there are no favourites displayed in the user's profile page.

Story Points: 3 Task Breakdown:

- Add a persistent field `favourites` to the `Vendor` class that is an `ArrayList<UserEmail>` containing user emails of all who favorited the `Vendor`. (Issue #73)
- Modify the `VendorInfoPanel` to show buttons to add vendors. (Issue #74)
- Implement a `ProfileFavouritesPanel` (that will be displayed within the `ProfilePanel`) that displays favourites. (Issue #75)
- Implement a `setVendor` method in `VendorService` that updates an existing `Vendor` object in the datastore with the newly modified `usersWhoFaved`. (Issue #76)

Demo Tasks:

1. Click on the `Admin` navigation link.
2. Click on the `Delete User` link for current User.
3. Log out.
4. Click on the `Find Food` navigation link.
5. Click on any vendor.
6. Look at VendorInfo Page.
7. Log back in to create new profile.
8. Click on the `Profile` navigation link.
9. Look at Profile Page Favourited Vendors
10. Click on the `Find Food` navigation link.
11. Click on any vendor.
12. Click on `Add to Favourites`
13. Click on the `Profile` navigation link.

As an admin, I want to be able to delete normal user accounts so that I can remove problem users from the app.

Priority: 3 Acceptance Criteria:

- Verify that the `AdminPanel` displays a table with all `Profile` objects (including other administrator's `Profile` objects) and has a `Delete User` button in each row associated with each user.
- Verify that when the `Delete User` button is pressed by an admin, that the normal user's `Profile` object is removed from the datastore.

Story Points: 2 Task Breakdown:

- Add a `getAllProfiles` method to the `ProfileService`, `ProfileServiceAsync`, and `ProfileServiceImpl` classes that returns all `Profile` objects from the datastore to the `AdminPanel`. (Issue #70)
- Create a `FlexTable` in the `AdminPanel` that displays all the retrieved `Profile` objects along with `Delete User` buttons. (Issue #71)
- Create private helper methods in `AdminPanel` that deletes the `Profile` object of the user whose `Delete User` button is pressed in the `FlexTable`. (Issue #72)

Demo Tasks:

1. Click on the `Admin` navigation link.
2. Look at AdminPage.
3. Click on the `Delete User` link for current User.
4. Log out.
5. Log back in.
6. Click on the `Admin` navigation link.
7. Look at AdminPage.

User stories and demo sequence that Jeffrey will present

As a user, I want to be able to reuse my Facebook profile photo in this app so I can maintain a consistent online identity.

Priority: 3 Acceptance Criteria:

- Verify that in the user profile a page, an `Import Facebook photo` link is provided so that the user can log in with Facebook to automatically reuse their Facebook profile photo.
- Verify that after the user has logged into Facebook, that their profile photo URL is fetched and set as their new profile photo.
- Verify that when a new user account is created, the user is given a default profile photo.
- Verify when the user logs out and logs back in, that the photo URL is stored and their photo is displayed.
- Verify that the app only stores the URL of the photo hosted on akamaihd.net servers, and does not save a copy of the photo in the app's data storage.

Story Points: 5 Task Breakdown:

- Add Facebook login button to profile page. (Issue #82)
- Fetch Facebook photo from URL. (Issue #84)
- Get Facebook photo URL on login. (Issue #83)
- Store Facebook Photo URL in `Profile` object. (Issue #85)

Demo Tasks:

1. Click on the `Profile` navigation link.
2. Look at Profile page.
3. Click on the `Import Facebook Photo` button.
4. Input Facebook username.
5. Click on the `Submit` button.
6. Log out.
7. Log back in.
8. Click on the `Admin` navigation link.
9. Look at AdminPage.
10. Click on the `Delete User` link for current User.
11. Log out.
12. Log back in.
13. Click on the `Profile` navigation link.
14. Look at Profile page.

As a Facebook user, I want to Like this app so I can share with my friends.

Priority: 3 Acceptance Criteria:

- Verify that when a user presses Like on the app, and they are not signed in on Facebook, that they are redirected to a Facebook login page.
- Verify that when the user presses Like on the app, and they are signed in on Facebook, that the app shows up as a Like on their Facebook profile.

Story Points: 2 Task Breakdown:

- Add Facebook like button code using an iframe to the `SocialMediaPanel`. (Issue #91)

Demo Tasks:

1. Login to Facebook.
2. Click on Facebook Like button.
3. Logout of Facebook.
4. Click on Facebook Like button.

As a Twitter user, I want to follow the page for this app so I can show all of my Twitter followers that I'm using this app.

Priority: 3 Acceptance Criteria:

- Verify that when the Twitter follow button is pressed, the Twitter page for this app is now listed as followed under the user's Twitter profile.
- Verify that if the user is not logged into Twitter, and the Twitter follow button is pressed, that the user is prompted to log into their Twitter account.

Story Points: 2 Task Breakdown:

- Add Twitter follow button code using an iframe to the `SocialMediaPanel`. (Issue #90)

Demo Tasks:

1. Login to Twitter.
2. Click on Twitter Follow button.
3. Logout of Twitter.
4. Click on Twitter Follow button.

As a logged in user, I want to review a vendor and see what past reviewers have said about a vendor.

Priority: 2 Acceptance Criteria:

- Verify that all previous vendor reviews for a particular vendor are shown in the vendor info page in for that particular vendor in ascending order by review date (oldest at top of the reviews section). (Also verify that the review date is displayed for each review).
- Verify that when the `Add Review` button (located at the bottom of the vendor info page) is clicked and the user is not logged in, the page changes to the `LoginPanel`.
- Verify that when the `Add Review` button (located at the bottom of the vendor info page) is clicked and the user has not previously submitted a review, the display changes to a form for creating a review for that particular vendor.
- Verify that when the `Add Review` button does not appear when a review was already submitted.
- Verify that when the user creates a new review they can select a between 1 and 5 stars for quality and between 1 and 5 dollar signs for cost.
- Verify that when the user creates a new review, a multi-line text field is displayed to allow the user to submit text.
- Verify that the same user cannot submit multiple reviews of the same vendor.

Story Points: 5 Task Breakdown:

- Implement a `ViewReviewsPanel` class (that is to be displayed in within the `VendorInfoPanel` for a particular vendor) for viewing all existing `Review` objects for a particular vendor. (Issue #64)
- Implement a `CreateReviewPanel` class for creating a new `Review`. (Issue #65)
- Implement a `setVendor` method in `VendorService`, `VendorServiceAsync`, and `VendorServiceImpl` that updates an existing `Vendor` in the datastore by adding a newly created `Review`. (Issue #66)

Demo Tasks:

1. Click on the `Find Food` navigation link.
2. Click on any vendor.
3. Look at VendorInfo Page.
4. Click on `Add to Review` button.
5. Select values for quality, dollar sign and review text.
6. Click on `Submit Review` button.
7. Click on the `Find Food` navigation link.
8. Look at VendorList quality and cost field.
9. Click on vendor where review was submitted.
10. Log out.
11. Log back in.
12. Click on the `Find Food` navigation link.
13. Click on vendor where review was submitted.
14. Log out
15. Click on the `Find Food` navigation link.
16. Click on any vendor.
17. Look at VendorInfo Page.
18. Click on `Add to Review` button.