| Test operations for adding/deleting/displaying Bike Rack comments | | | |
| --- | --- | --- | --- |
| Test tables are visible | | | |
| # | Test Steps | Expected Behavior | P/F |
|  | Using a browser of your choice, go to this URL:  <http://1-dot-sanghyeokpark124.appspot.com/>  Log in using Gmail accounts. | On successful login, to the right half of the screen, there is a tab-like widget that contains two tabs:  The first tab contains the table that displays all visible bike rack locations on the map.  The second tab contains a table that will display all favorite bike racks locations marked by the user. |  |
| Test table data is updated in the table | | | |
|  | Log in as admin accounts.  Ensure that there is no loaded data. | The table displays no bike rack data. When “load data” button is pressed, all the parsed bike rack data should be loaded in to the table. |  |
|  | Log in as admin accounts.  Ensure that there is loaded data. | When “remove data” button is pressed, table will be cleared displaying no bike rack data in the table. |  |
|  | Ensure that there is a bike rack locations displayed as markers on the map.  Ensure that there is loaded data. | Number of bike rack data in the table matches the number of markers on the map. |  |
|  | Ensure that there is loaded data  .  Search for bike racks available around for specific point of interest. | If there is no markers on the map after the search, table displays an empty table.  If new markers are displayed after the search, table is updated with the bike rack data of those markers.  Table is updated every time markers on the map are updated. |  |
|  | Ensure that bike rack table is displayed and is filled with bike rack data. | Bike rack data is displayed in a table with appropriate columns. |  |
| Test Comment DialogBox | | | |
|  | Ensured that table is displayed with the bike rack data loaded, along with the view comments button. | On load of table, each bike rack should also have a “Comments” column, with a button named “View Comments” |  |
|  | Ensure that the Comment Dialogbox appears. | On click on view comments, a DialogBox should open containing the Comment Table. |  |
| Test Adding/Viewing Comments | | | |
|  | Add comment using the “Post” button. Check that the comment has been added to datastore.  Type desired comment into the textbox, then press post. | On click the text should appear in the table.  To check that the comment has been added to the datastore, close the DialogBox then reopen it so it loads the data back. The comment should appear.  \*NOTE\*: Adding entry to datastore takes a while, so the changes may not be immediately reflected. Keep reopening the DialogBox and the changes should appear eventually. |  |
| 10. | Type desired comment into the textbox, and press enter to post the comment. | Same result as #9. |  |
| 11. | Log out and sign in as a different user. Go to the same bike rack that the other user left comments on. | The comments left by the other user should be visible, along with the username and the delete buttons. |  |
| 12. | Add two comments with new user. One with the “Post” button and one with the “Enter” key. | The comments should now be visible along with the comments from the other user. |  |
| Test Delete Comment | | | |
| 11. | Attempt to delete a comment made by another user with the “Delete” button. | On click an alert screen should appear saying that it’s not your comment, and nothing is deleted. |  |
| 12. | Attempt to delete a comment made by the current user. | On click the comment should disappear from the table. To see if the change is reflected in the datastore, close and reopen the Comment DialogBox, and the comment that was deleted should not appear.  \*NOTE\*: Similar to adding comments, deleting an entry from the datastore takes a while, and the changes may not be immediately reflected. Reopen the DialogBox and changes should eventually appear. |  |