**Assignment 3 Project Reports**

## **Group 1: Lyrist Pineda, Rodrigo Saavedra, David Ramirez, Gerardo Salcedo, Garrett Hubach**

Deployment Document:

Application Summary: The purpose of the application is to identify any restrooms on campus with open toilets. We start by logging into the toilet application with our username and password. Once we log into the application, it allows us to see where the toilets are located within the campus via a map that shows the entire campus. On this map dashboard we can find several buttons that represent available toilets. Initially no toilet is selected but when selected we can see information based on the site selected at the bottom portion of the screen. We can also log out if needed.

When the review button is pressed, you will be taken to another page that will let you make reviews and view them as well depending on which toilet is selected on the dashboard. You can leave a rating or comment and then submit. Only a select most recent will be viewable. You can leave the review section and return back to the main dashboard at any point.

Deployment Details:

Apache Tomcat 10.1 was used alongside Eclipse IDE for web development. Once the tomcat is ready we can then configure the server to run our program locally using any used ports available to our machine. Since this is a web based application, front end languages such as Html and Javascript will be used GUI and to facilitate client side features, while Java is used for backend functionalities such as server side dealing and in general logic features. On further note, Html is responsible for handling the static elements of the page, CSS for stylized appearance and layout, and Javascript for handling the dynamic aspect of the page. When running the program, other web based features need to be accounted for.

Servlets function to help take requests from the frontend to the backend. The use of servlets in this case will be to help fulfill specific query requests made by the user and give off a response accordingly. The three main queries to be used are searching for a specific user within the database and then logging them in or not if the user was found, to receive input from a user from one of the many languages.

The type of database that is to be used will be an SQL. The SQL holds information in the form of attributes pertaining to certain important objects in our code. The three main objects that need these settings will be toilets, users, and review. In order to use this data, what is needed is to connect the backend Java with a server hosting the database. One way to make connectivity easier with Java and SQLs is through the use of the Java Database Connectivity (JDBC) API. The JDBC provides necessary methods to connect the languages with each other.

Networking Requirements:

A method of accomplishing the networking requirements could be to utilize web sockets if the web project has been pushed further into deployment. A web socket should enable the user to more accurately interact with the application in a server client relationship. One use for this in our application was planned to use it to allow for real time status changes such as to notify and tally if a user is occupying a restroom stall or not.

Test Cases:

Black Box Testing:

1. The Login Page
   1. Enter a username and password then press the login button: Logic behind the login button does not work, servlet is not properly mapped to the login page. Will however deny the user and throw out a notification noting that they do not have the appropriate username and password set. Also will allow the user to proceed if they set their username and password as “u” and “p”
   2. Enter a username and password then press the register button: Logic behind the register button does not work. Users will just be pushed to the next page.
   3. Press the guest button: Logic behind the Guest button lets a user move to next page
2. The Map Dashboard
   1. Using the logout button: Can return back to login using the logout button, cannot remove user since there is no status as to whether the user
   2. Using the review button: Redirect to a review page, review page is the same for each site.
   3. Using the selection button: Does nothing. Both front end and back portion not implemented
   4. Using one of the 10 provide toilet buttons:
      1. Returns one notification “Toilet 1 Selected”, does not update information panel accordingly
      2. Returns one notification “Toilet 2 Selected”, does not update information panel accordingly
      3. Returns one notification “Toilet 3 Selected”, does not update information panel accordingly
      4. Returns one notification “Toilet 4 Selected”, does not update information panel accordingly
      5. Returns one notification “Toilet 5 Selected”, does not update information panel accordingly
      6. Returns one notification “Toilet 6 Selected”, does not update information panel accordingly
      7. Returns one notification “Toilet 7 Selected”, does not update information panel accordingly
      8. Returns one notification “Toilet 8 Selected”, does not update information panel accordingly
      9. Returns one notification “Toilet 9 Selected”, does not update information panel accordingly
      10. Returns one notification “Toilet 10 Selected”, does not update information panel accordingly
3. The Review Section
   1. Pressing a number button for the review: returns notification stating which button has been pressed
   2. Textfield: text can be written within, but this cannot be used for anything is not affected by another buttons
   3. Submit button: regardless of any previous input, there will be no change within

White Box Testing:

Very little black box testing was performed since classes necessary for networking, multithreading, and for handling the SQLs were incomplete. Some testing was done with CSV data collected and then later meant to be used in the creation of the SQLs. Information provide may also

1. Login logic using a list array of users created from a csv of 10,000 users: no issues occurred when attempting this, screens transitioned seamlessly without much trouble. Search was based on username and password, took roughly the same amount of