Test Cases:

As a customer, I would like to have a modern and easy login experience.

Johnson the customer decides to register for an account in the Autoclaim service. In order to do this, he clicks on the button that will take him to the "registration" window. In the Autoclaim register page, the user then inputs their information into the required text fields. In the password field, he enters "password123". He expects for the characters in his password to be replaced with dots so that no one else will see it. He also inputs his email address, js@gmail.com, as well as his username,

Once Johnson has completed his registration, he must sign in. On the sign in page, Johnson enters his email into the email text field, and then enters his password. The email he enters has a typo, and is missing the 's', and the password he enters is missing a '2'. This results in an "incorrect email and/or password" error.

Finally, Johnson corrects his email and password by deleting them from their respective fields, and inputs the correct information into them. After this, he clicks the 'sign in' button, and is able to move on to the next screen. Johnson is able to look at the customer view since he selected "customer from the list".

As a customer, in order to quickly file a claim at any time without having to visit an office or speak with an agent, I would like to do the whole process to be smooth.

Once Johnson has signed in, he will be greeted to the customer GUI, which will grant him a view of a side menu with three buttons, Home, My Insurance, and My Claims. Each button on the left should change the page on the right to include options corresponding to what was chosen.

Once the button "My Claims" has been chosen, Johnson should be able to get a view of the claims that he has open so far, as well as previous closed claims, which will be greyed out. Johnson is also able to scroll through the list of claims. Next to this list is the option to open up a new claim. Once he clicks on this button, he will be able to move to the "Input accident information" page.

In the "Input accident information" page, Johnson will be able to enter all claim information regarding the accident itself. Johnson notes down the details of the accident, the time it occurred, and the information of his vehicle, such as the make and model, and the licence plate number. After he creates the claim, a claim ID should be assigned to it.

Code Smells:

Repeating Code:

LoginController.java

Line 81 contains if/else statements which reuse the same block of code multiple times. This can be changed by altering the code so that this block of code only has to be used once, perhaps by creating a new method with the reused expressions.

Long Method:

LoginController.java

Further down, the same alerts are repeated multiple times. In combination with the other expressions that have been duplicated, this method becomes much longer. Perhaps if this method is split into smaller methods so code does not have to be duplicated, the length of it will be shortened.

Overused Comments:

These smells can be fixed by making names of methods more clear, and by using comments to explain why certain code is used.

LoginController.java

Currently contains comments of different iterations of the controller. Once the login controller is fully completed, these extra comments should be removed.

Claims.java

Contains comments on what every method is but does not contain the 'why' on why they are used. These comments should be replaced with one comment explaining why these methods are used.

Long Parameter List:

Claims.java

Lists the parameters for a multitude of the methods that are used, and the getters and setters that are used. Perhaps some of the parameters used could be split up into different methods.