Contents

[Artefact 1 – New Client Features 2](#_Toc465434782)

[Artefact 2 – GUI Designs 3](#_Toc465434783)

[Artefact 3 – Storyboard 4](#_Toc465434784)

[Artefact 4 – SQL to Reports 6](#_Toc465434785)

[Summary of the finances for every event. 6](#_Toc465434786)

[Finances of a specific event 6](#_Toc465434787)

[List all the volunteers volunteering to an event 7](#_Toc465434788)

[List all the members attending an event 8](#_Toc465434789)

[Artefact 5 – Test Data 9](#_Toc465434790)

[Planners table 9](#_Toc465434791)

[Planners event 9](#_Toc465434792)

[Event table 10](#_Toc465434793)

[Artefact 6 - Automated Testing 11](#_Toc465434794)

[Registers new members 11](#_Toc465434795)

[Login Validation 12](#_Toc465434796)

# Note

The full artefacts are available in the folders above this document.

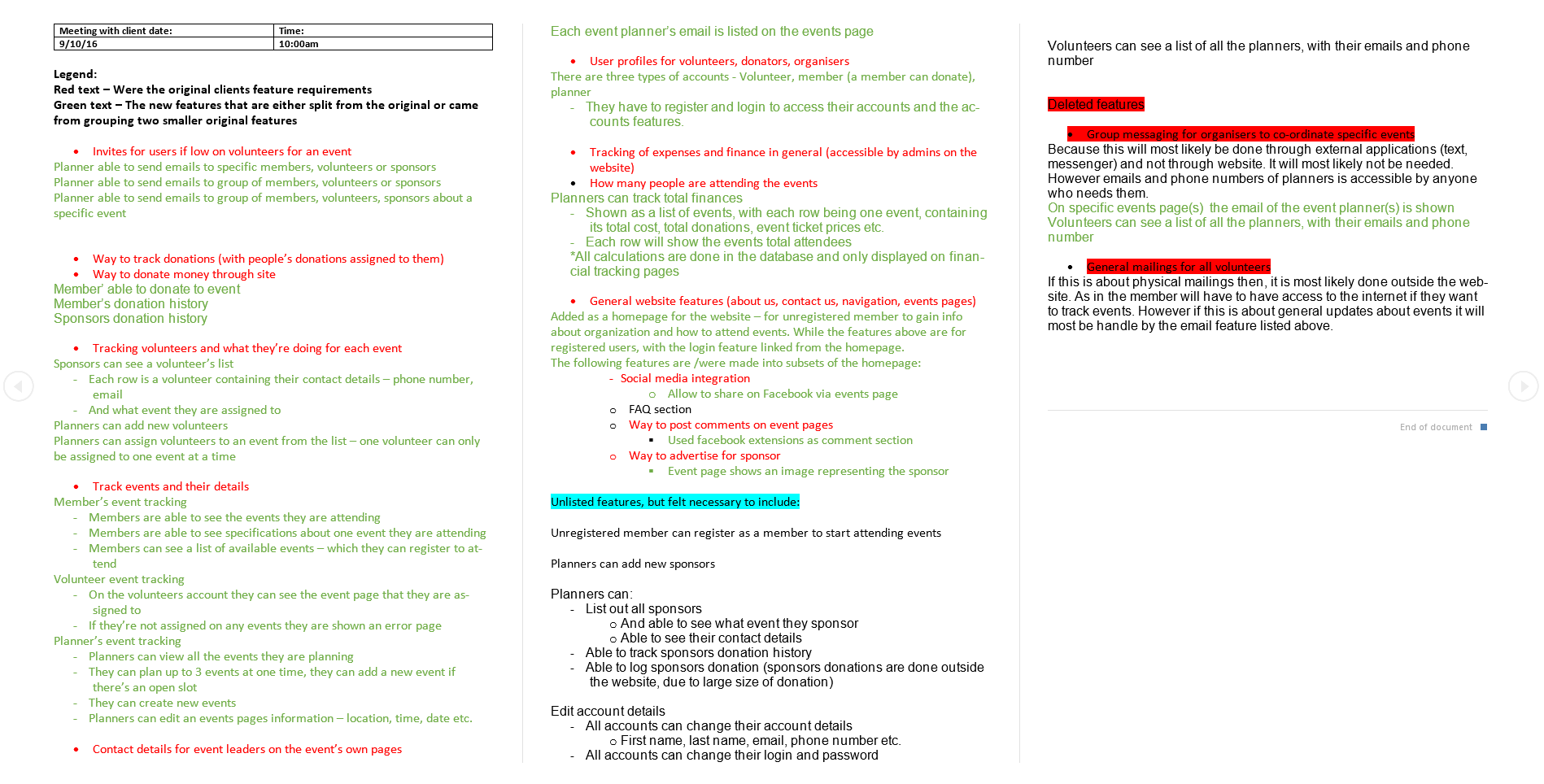
# Artefact 1 – New Client Features

After what I felt like was a poor release one. I changed most of the already existing feature specifications from the client. So that I could redo the user stories. Mostly I just split up the existing features, into more specific features. So that the new user stories based off of them were smaller and more specific. I also based each user stories to follow the GUI of the account type they belong to. With each user story belonging to one page. So that during development they could be more modular. I also redid the acceptance criteria, so that they could actually be tested.

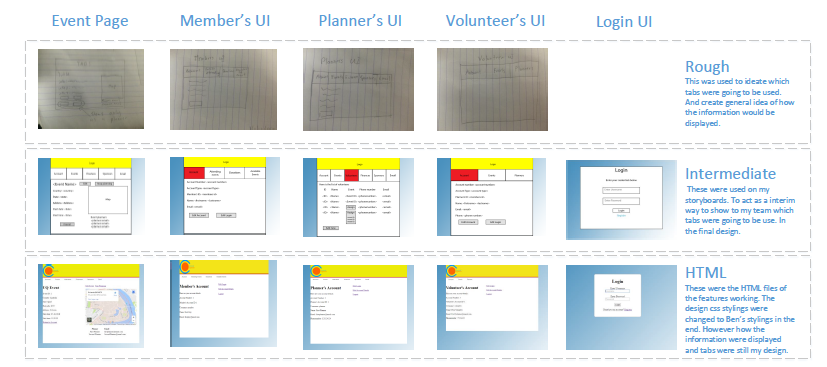
I finished these features a couple of days after the release one presentation. However due to public holidays and mid semester I couldn’t have had the meeting with the client team until the sprint three presentation. During this meeting I explained to the clients how I changed the features they wanted, with the client team being fine with the changes.

I also got rid of a couple features that I felt were a bit out of the scope from the client specifications.

The full document, with more specific explanations of the changes I made.



# Artefact 2 – GUI Designs

To convey to my team how the new features were going to be connected to each other I created GUI designs of each feature. I wanted my team to focus on the tabs the user would see after they log in depending on the account type. Here is a summary of the GUIs.

The rough GUIs were mainly me ideating on what the user would see after they login.

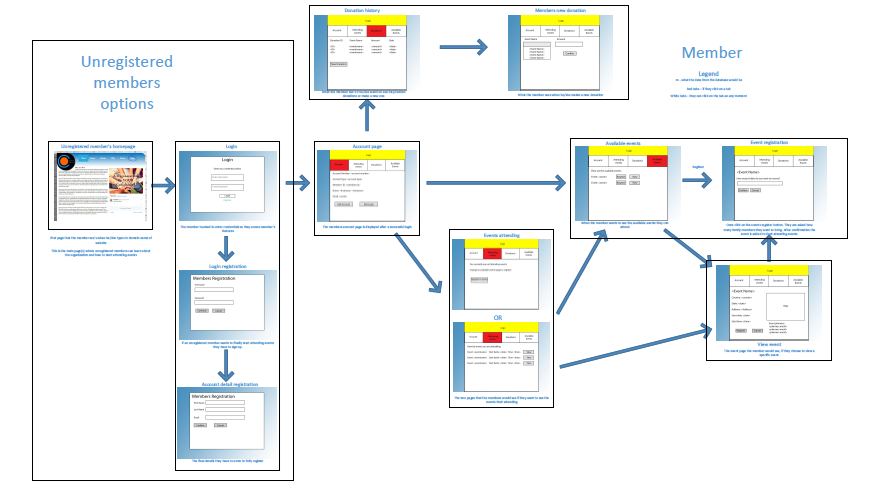
The intermediate designs were the GUIs I made to be used in the Storyboard. These were the GUI’s that I used to convey to my team how I wanted the new features to look. The summary only shows some of the GUI’s in the artefact folder contains all the GUIs I made to be used in the storyboard.

The HTML GUIs were the semi-final designs of the features finally developed and working. Although in the final design we ended going with were Ben’s css stylings. What is shown is the features fulling working and doing what they were designed to do.

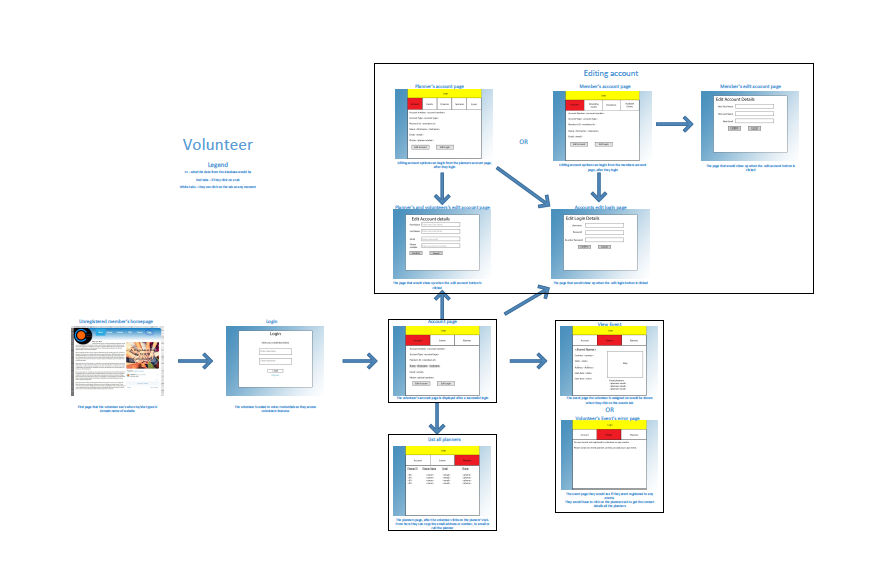
# Artefact 3 – Storyboard

I created three storyboards to show to my team what the user would see when using our website, depending on their account type. Although the created storyboard isn’t laid out like a typical storyboard it still functions the same way. Each path available is all the options available to the user. So each path taken acts as an individual storyboard.

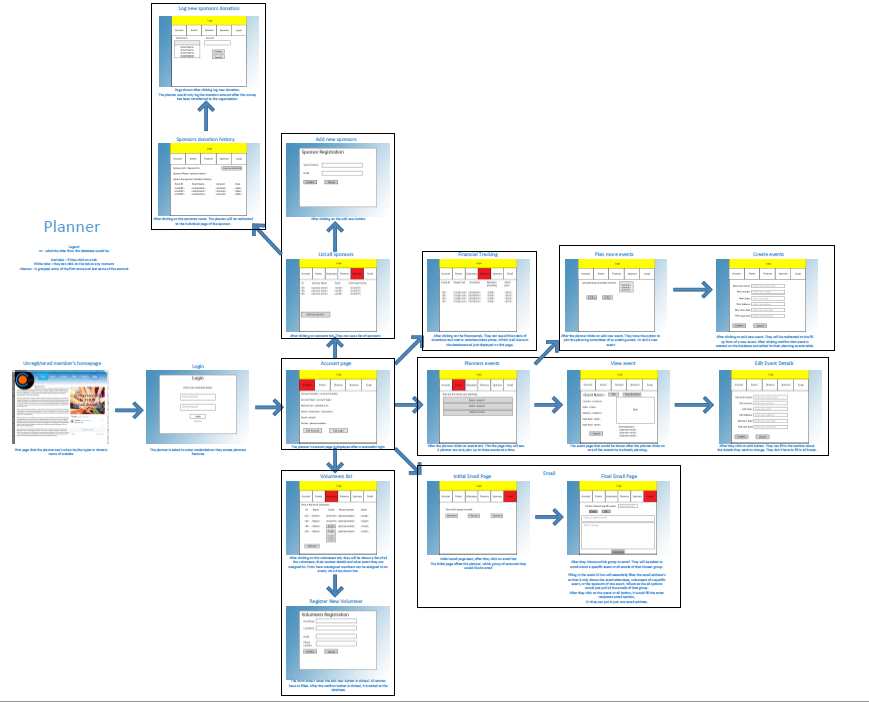
Member’s storyboard – showing all the options of a user who logged in as a member.



Volunteer’s Storyboard – all of the options of a user who logged in as volunteer



Planner’s Storyboard – all options of a user who logged in as a planner



# Artefact 4 – SQL to Reports

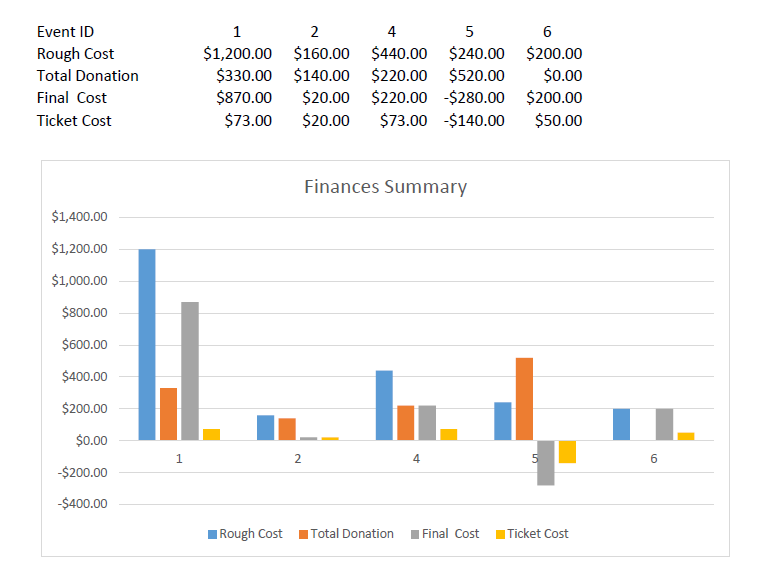
After developing the features planned. I wanted to see some data from the database, to make sure they’ve been working properly. I based each query and report on what the planner would see.

## Summary of the finances for every event.

SQL –

*SELECT `Event\_ID`,`Rough\_cost`,`Total\_donation`,`Final\_cost`,`Ticket\_cost` FROM `finances`*

After I ran this query in PHPmyAdmin and exported as an excel file. I then formatted it in excel to display the graph.



## Finances of a specific event

SQL –

*Select the finances of event id 1*

*SELECT \* FROM `finances` WHERE `Event\_ID` = 1;*

*Select all members donations for evend id 1*

*SELECT `member\_ID`,`Amount`,`Date` FROM `members\_donation` WHERE Event\_ID = 1*

*Select all the sponsors donations for event id 1*

*SELECT `sponsor\_ID`,`Amount`,`date\_donated` FROM `sponsors\_donation` WHERE `event\_ID` = 1*

After I exported the file from PHPmyAdmin, I then formatted the csv dump to show the report. With the totals being generated using commands in excel.

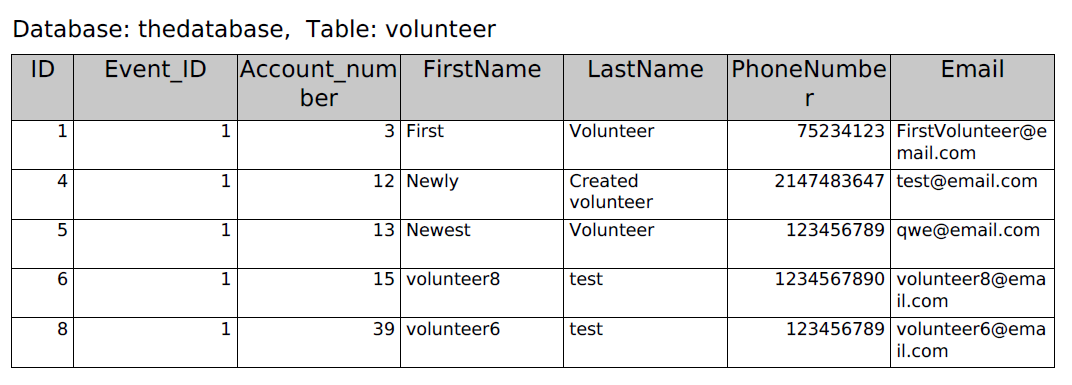


## List all the volunteers volunteering to an event

SQL –

*SELECT \* FROM volunteer where Event\_ID = 1*

After running this query in PHPmyAdmin, I exported it as a pdf. With the styling and formatting all being done by PHPmyAdmin.

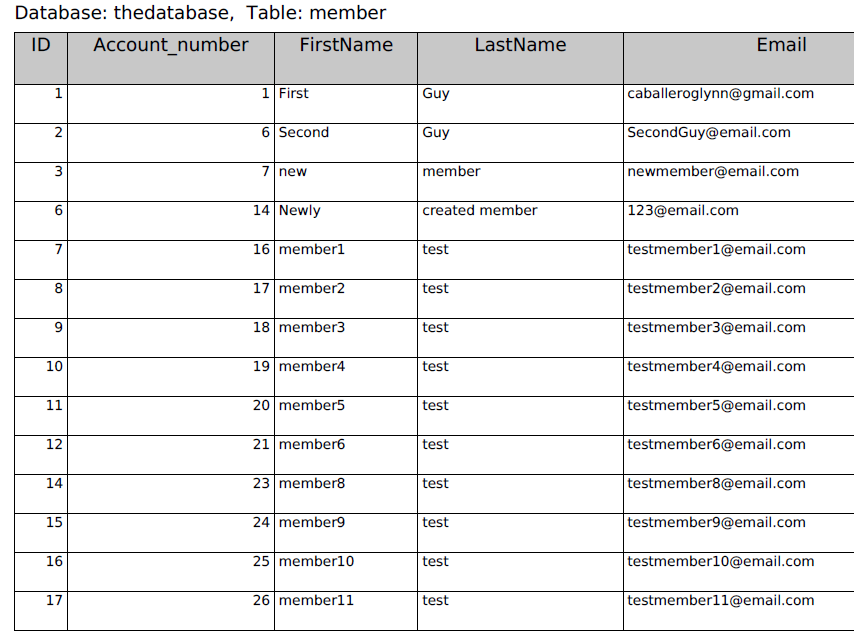


## List all the members attending an event

SQL –

*SELECT \* FROM member where ID IN (SELECT ID\_number from attending\_events where Event\_ID = 1)*

After running this query in PHPmyAdmin, I exported it as a pdf. With the styling and formatting all being done by PHPmyAdmin.



# Artefact 5 – Test Data

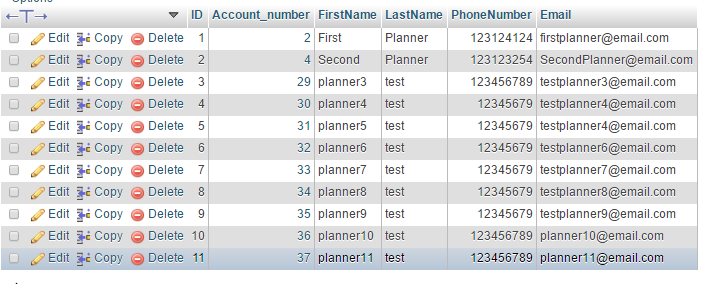
I created multiple test data to make sure the features being developed were working properly.

I used two methods to insert the test data into the table, they were: the insert into tab from PHPmyAdmin and running sql queries. Specific inputs of these are available in the test data folder.

From these I created these sample test data (there are more in the test data folder):

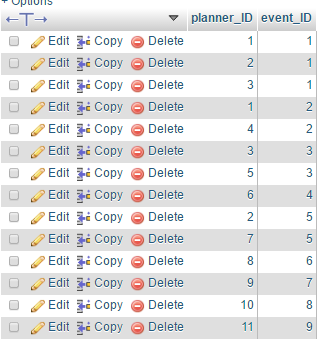
## Planners table

Table containing the information of every planner.



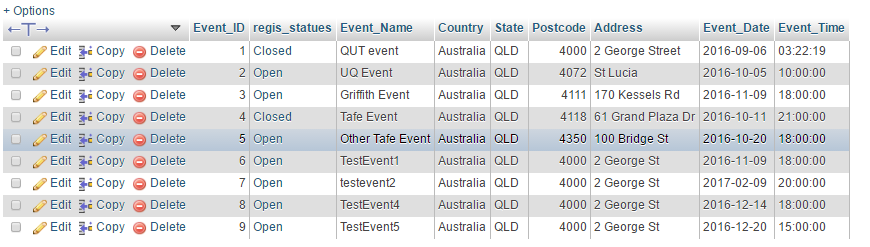
## Planners event

The table used to link the planner’s account id to the event they are planning.



## Event table

The event table containing all the information of an event



# Artefact 6 - Automated Testing

To test the features were working properly I created testing scripts for selenium.

## Registers new members

Sample of script (full script is on the automated testing folder) –

|  |  |
| --- | --- |
| <tr>  <td>open</td>  <td>/Login/Login\_form.php</td>  <td></td>  </tr>  <tr>  <td>clickAndWait</td>  <td>link=Register</td>  <td></td>  </tr>  <tr>  <td>type</td>  <td>id=username</td>  <td>member1</td>  </tr>  <tr>  <td>type</td>  <td>id=password</td>  <td>qwe</td>  </tr>  <tr>  <td>type</td>  <td>id=con\_password</td>  <td>qwe</td>  </tr>  <tr>  <td>clickAndWait</td>  <td>id=save</td>  <td></td>  </tr> | <tr>  <td>type</td>  <td>id=f\_name</td>  <td>member1</td>  </tr>  <tr>  <td>type</td>  <td>id=l\_name</td>  <td>test</td>  </tr>  <tr>  <td>type</td>  <td>id=email</td>  <td>testmember1@email.com</td>  </tr>  <tr>  <td>clickAndWait</td>  <td>id=save</td>  <td></td>  </tr>  <tr>  <td>clickAndWait</td>  <td>id=confirm</td>  <td></td>  </tr> |

Running this script the first time was a success. While running it a second time caused it to fail. Which was expected as the first time, the database table has been inserted the username and password of the new member. Meaning running it a second time the username has been taken, as the account already exists in the database.

## Login Validation

Sample script (full script in the automated testing folder) –

|  |  |  |
| --- | --- | --- |
| Valid Login | Invalid Username | Invalid Password |
| <tr>  <td>open</td>  <td>/Login/Login\_form.php</td>  <td></td>  </tr>  <tr>  <td>type</td>  <td>id=username</td>  <td>member</td>  </tr>  <tr>  <td>type</td>  <td>id=password</td>  <td>qwe</td>  </tr>  <tr>  <td>clickAndWait</td>  <td>id=login</td>  <td></td>  </tr>  <tr>  <td>clickAndWait</td>  <td>link=Logout</td>  <td></td>  </tr> | <tr>  <td>open</td>  <td>/Login/Login\_form.php</td>  <td></td>  </tr>  <tr>  <td>type</td>  <td>id=username</td>  <td>membre</td>  </tr>  <tr>  <td>type</td>  <td>id=password</td>  <td>qwe</td>  </tr>  <tr>  <td>clickAndWait</td>  <td>id=login</td>  <td></td>  </tr>  <tr>  <td>clickAndWait</td>  <td>link=Logout</td>  <td></td>  </tr> | <tr>  <td>open</td>  <td>/Login/Login\_form.php</td>  <td></td>  </tr>  <tr>  <td>type</td>  <td>id=username</td>  <td>member</td>  </tr>  <tr>  <td>type</td>  <td>id=password</td>  <td>ewq</td>  </tr>  <tr>  <td>clickAndWait</td>  <td>id=login</td>  <td></td>  </tr>  <tr>  <td>clickAndWait</td>  <td>link=Logout</td>  <td></td>  </tr> |

As expected only 1 test case passed. Which was the valid login script while the other two failed.