Testing:

The process in which identifying the performance of software through the helps of functional and non-functional testing. The major process is to identify the weakness and their improvement in the system. The testing is classified into the two-category functional and non-functional.

Importance of testing:

* To ensure that the functional requirement has been developed
* To make the developed produce has no errors
* To ensures the quality of the developed produce. (thakur, 2017)
* For effective performance of product
* For ensuring that software perform actual task not opposite because it might be expensive later on future improvement

The sum of list of functional and non-functional are listed below: The functional and non-functional testing based on the black box testing:

Functional testing:

* Unit testing
* Integration testing
* Interface testing
* Regression testing
* System testing
* Beta testing

Non-functional testing:

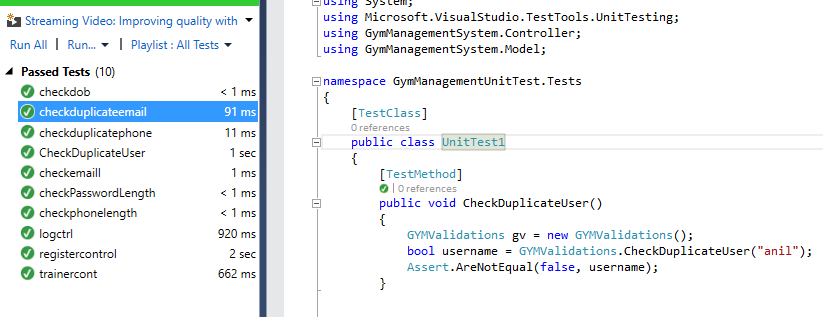
* Usability testing
* Reliability testing
* Installation testing
* Security testing
* Recovery testing

Unit testing:

Unit testing is the part of the application of testing in which smallest part of code is used to test that the parts of code is performing the actual function or not.

Test Name: check duplicate user

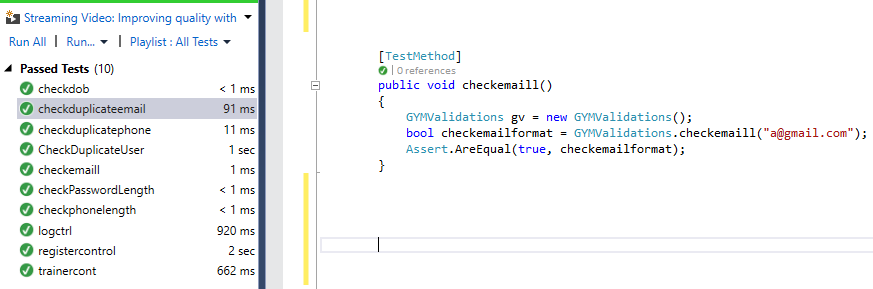
|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Test id | Test scenario | Test input data | Expected result | Actual result | Pass/fail |
| 1 | Check duplicate user | Username | Username should not be repeated | Duplicate data cannot be inserted with same name | Pass |



Testing figure testing duplicate user

Test Name: check email format

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Test id | Test scenario | Test input data | Expected result | Actual result | Pass/fail |
| 2 | Check email format | Email in @gmail.com format | Email must contain @gmail.com | Email must have @gmail.com to insert | Pass |



Testing figure testing email contains @gmail.com

Test Name: checking password length

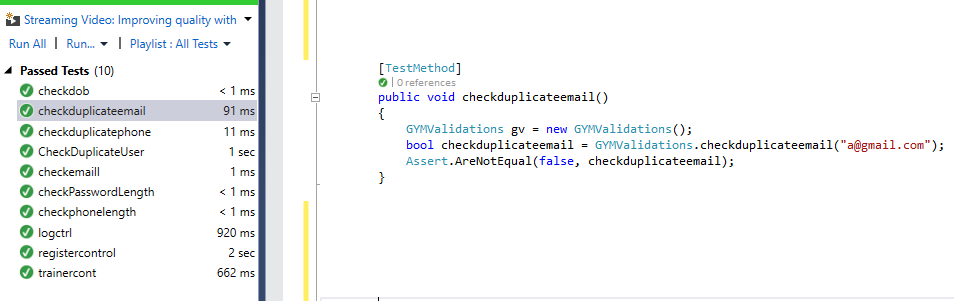
|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Test id | Test scenario | Test input data | Expected result | Actual result | Pass/fail |
| 3 | Check password length | Length must be greater than 5 | Must include more than five | Password must have greater than 5 length | Pass |



Testing figure checking password length

Test Name: checking duplicate email address

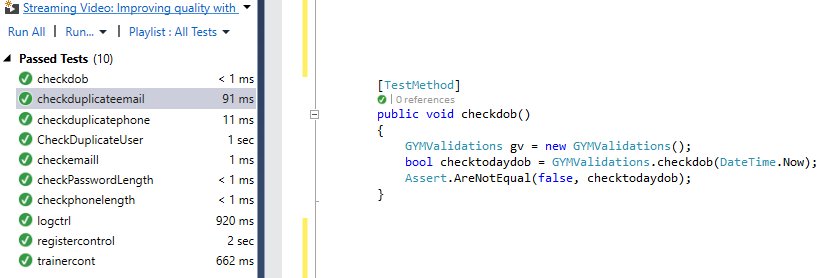
|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Test id | Test scenario | Test input data | Expected result | Actual result | Pass/fail |
| 4 | Check duplicate email address | Email address | Must not able to use same email address | Duplicate has been stope | Pass |



Testing figure checking duplicate email

Test Name: check date of birth

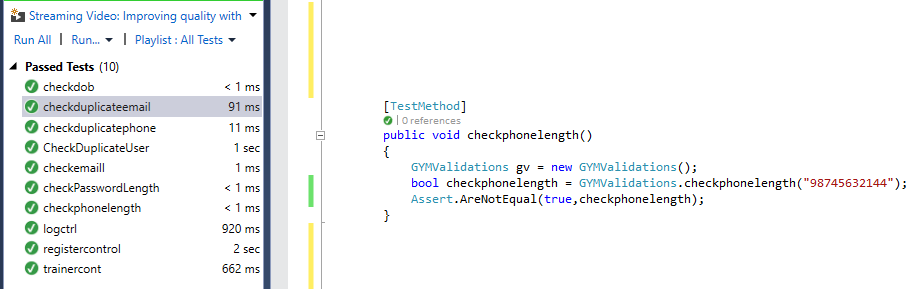
|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Test id | Test scenario | Test input data | Expected result | Actual result | Pass/fail |
| 5 | Check date of birth | Insert current date time | Should not ignore current date time | Must include before today date not coming date | Pass |



Testing figure testing date of birth

Test Name: check phone length

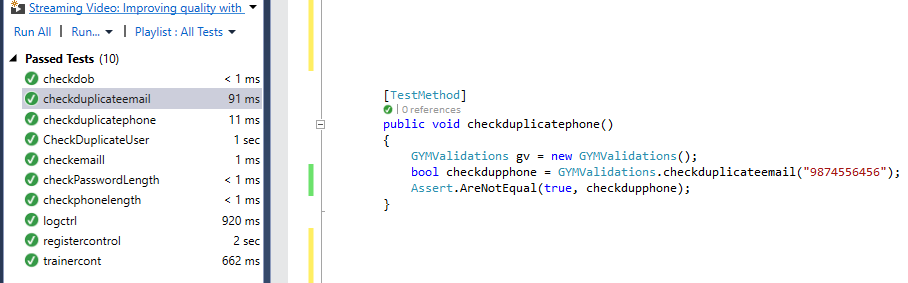
|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Test id | Test scenario | Test input data | Expected result | Actual result | Pass/fail |
| 6 | Check phone length | Length must be of 10 digits or less | Must ignore greater than 10 digits | Insert having less than 10 digits | Pass |



Testing figure testing phone length up to 10 digits

Test Name: checking duplicate phone number

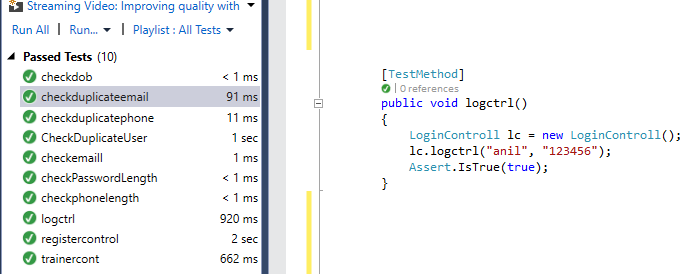
|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Test id | Test scenario | Test input data | Expected result | Actual result | Pass/fail |
| 7 | Check duplicate phone | Phone number | Ignore the same phone number twice from inserting | Must insert single unique phone not duplicate | Pass |



Testing figure testing to not allowed duplicate phone number

Test Name: Check login form

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Test id | Test scenario | Test input data | Expected result | Actual result | Pass/fail |
| 8 | Check login form data | Username and password | Login successful | Successfully pass the username and password | Pass |



Testing figure checking login form

Test Name: insert registering form

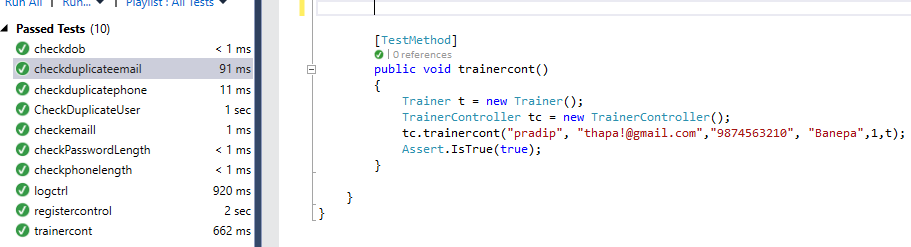
|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Test id | Test scenario | Test input data | Expected result | Actual result | Pass/fail |
| 9 | Check register form | Fill up all data | Must be register success | Success inserted into database | Pass |



Testing figure testing register form

Test Name: insert trainer

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Test id | Test scenario | Test input data | Expected result | Actual result | Pass/fail |
| 10 | Check trainer inserted | All data of trainer form | Must insert without errors | Inserted without errors | Pass |



Testing figure testing trainer form

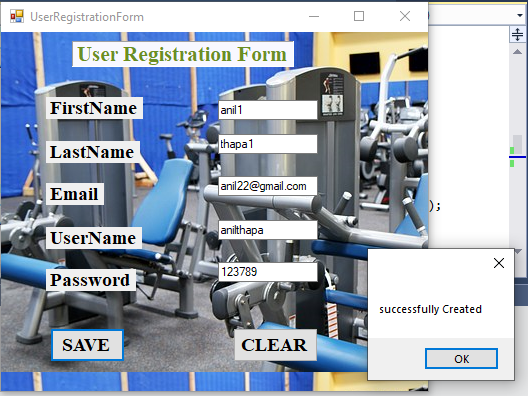
Black box testing:

It is the process in which testing of user interface of data no the internal parts of the system. In this process the internal function does not to test. Only from the internal interface data inserted and tested as it made. In this project choosing this testing is simply to test the need and actual performance of the program.

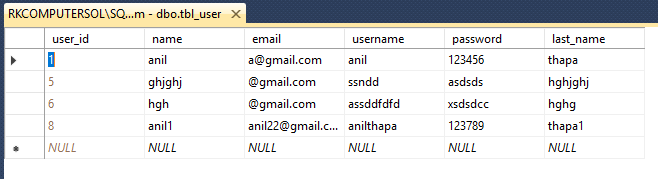
Test-name: registration test

Pre-condition: must login before login

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Test no | Test scenario | Test input data | Expected result | Actual result | Pass/fail |
| Bb1 | Check and insert registering data | First name: anil1  Last name: thapa1  Email: anil22@gmail.com  Username: anithapa  Password; 123789 | The data must be insert without errors and before login into system | Successfully inserted data and ready for the login | Pass |



Testing figure registering user details

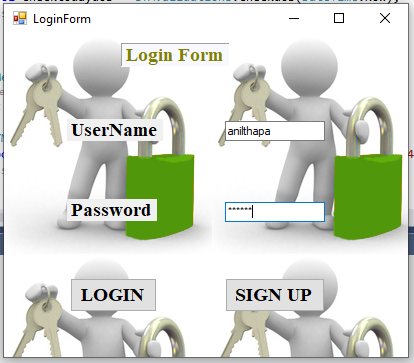


Testing figure view from database of success register data

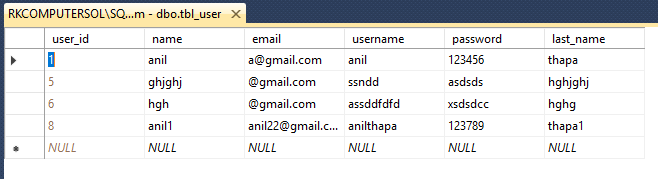
Test-name: login test

Pre-condition: must login before inserting into dashboard

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Test no | Test scenario | Test input data | Expected result | Actual result | Pass/fail |
| Bb2 | Login with the registered data | Username: anithapa  Password; 123789 | The data must be login without errors and redirect into dashboard | Successfully login and into dashboard | Pass |



Testing figure login with registered data

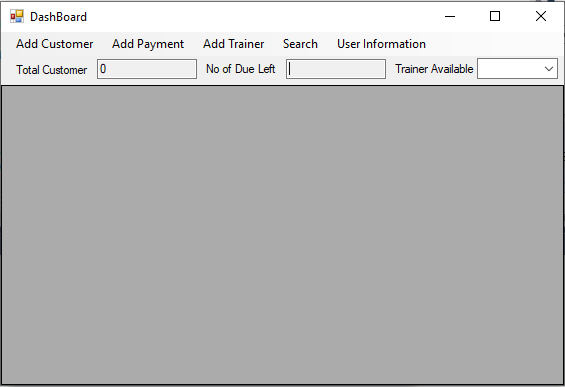


Testing figure user name and password of login data

Test-name: dashboard test

Pre-condition: must login before dashboard

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Test no | Test scenario | Test input data | Expected result | Actual result | Pass/fail |
| Bb3 | Login with the registered data and redirected to dashboard | All required function from dashboard | Allow to see all features | Successfully login and into dashboard, allowed to use all functions | Pass |

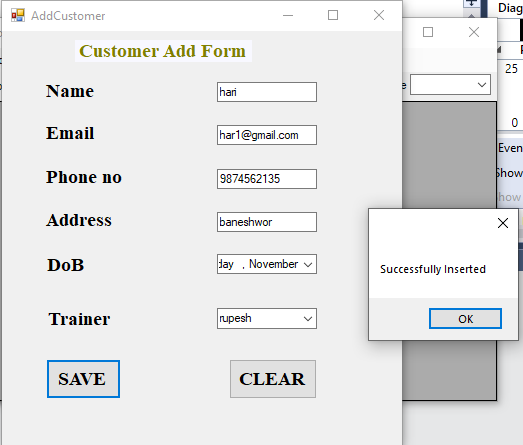


Testing figure after success login access to dashbord

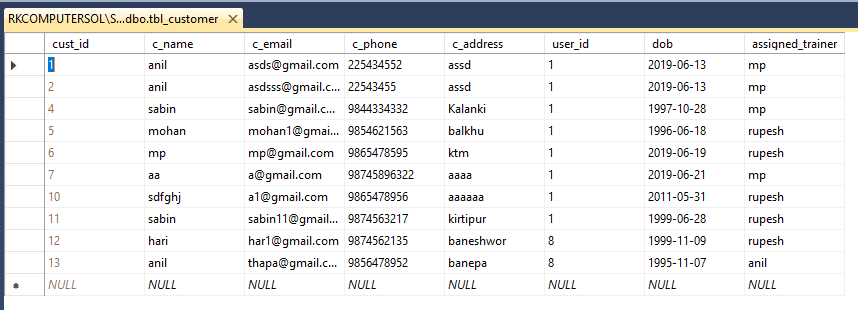
Test-name: dashboard test

Pre-condition: must login before dashboard

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Test no | Test scenario | Test input data | Expected result | Actual result | Pass/fail |
| Bb4 | Login with the registered data and redirected to dashboard | All required function from dashboard | Allow to see all features | Successfully login and into dashboard, allowed to use all functions | Pass |



Testing figure add customer data

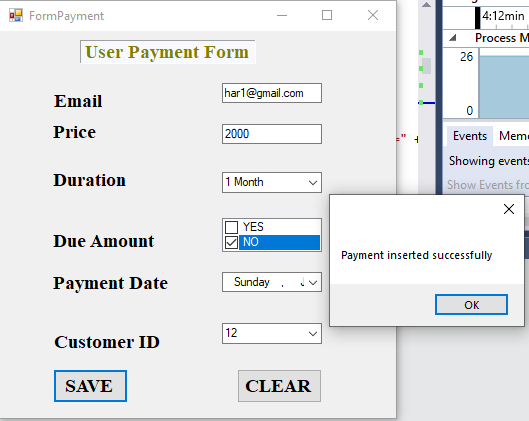


Testing figure data in database from interface

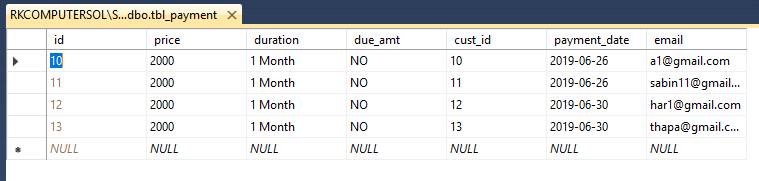
Test-name: user payment test

Pre-condition: after dashboard user payment details

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Test no | Test scenario | Test input data | Expected result | Actual result | Pass/fail |
| Bb5 | User payment add details after access to database | Email of customer: [har1@gmail.com](mailto:har1@gmail.com)  Price: 2000  Duration: 1month  Due amount: No  Payment date: current date  Customer id= 12 | Must be insert without any errors | Successfully inserted payment of particular added customer | Pass |



Testing figure user payment data

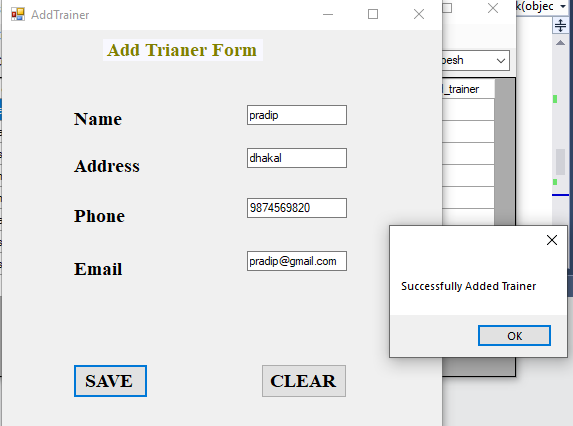


Testing figure data from database from form inserted

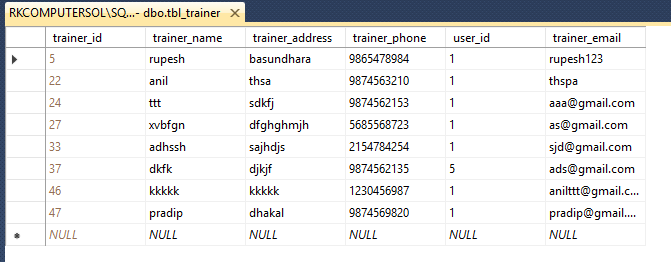
Test-name: user add trainer test

Pre-condition: after dashboard user can access to add trainer

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Test no | Test scenario | Test input data | Expected result | Actual result | Pass/fail |
| Bb6 | User added details of trainer | Name: Pradip  Address dhakal  Phone: 9874569820  Email: pradip@gmail.com | Must be insert without any errors | Successfully added trainer of particular users | Pass |



Testing figure adding trainer from form

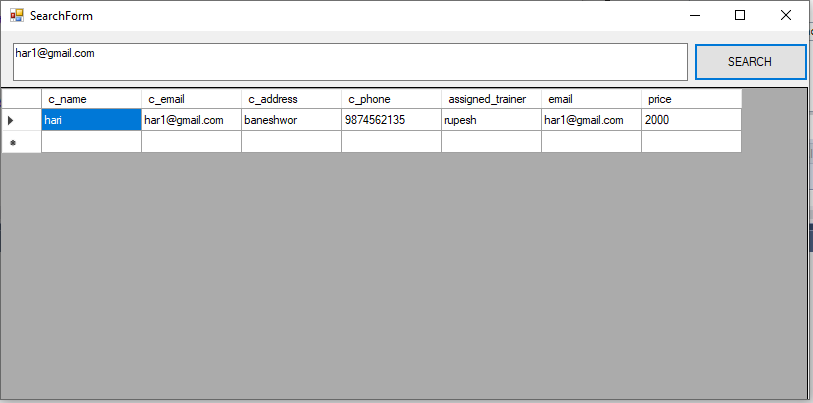


Testing figure data into the database from form

Test-name: user add trainer test

Pre-condition: after dashboard user can access to add trainer

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Test no | Test scenario | Test input data | Expected result | Actual result | Pass/fail |
| Bb7 | Search customer details | Email: har1@gmail.com | Must show the details of customer if exists | Successfully display the search data of customer without errors | Pass |

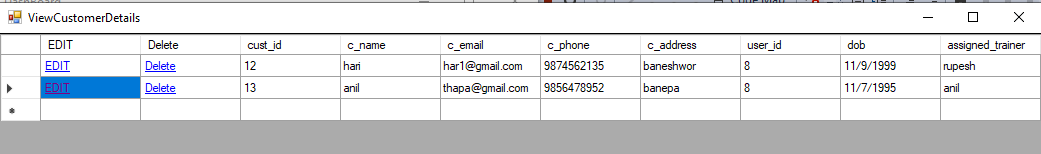


Testing figure searching user data through email

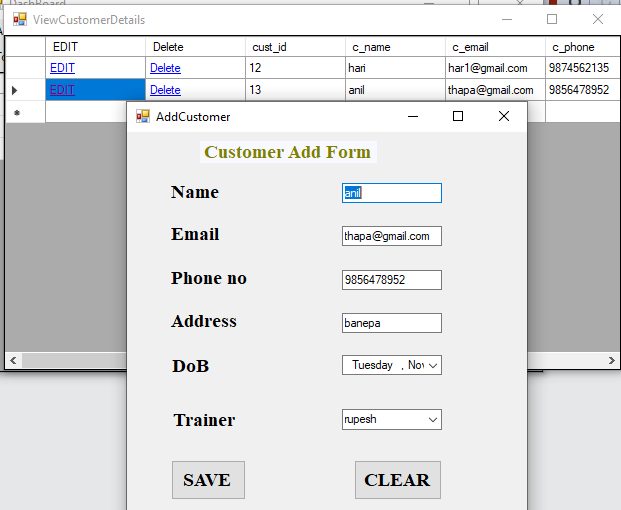
Test-name: update customer test

Pre-condition: after dashboard user can access to update profile

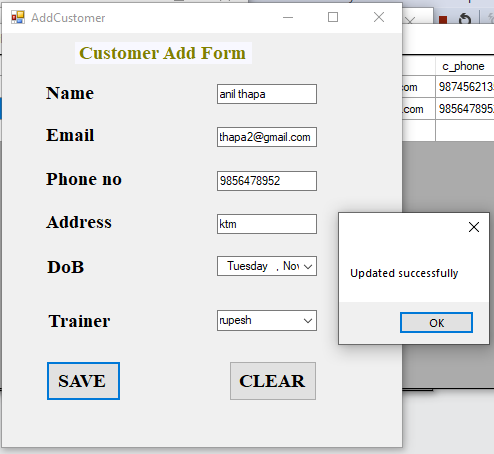
|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Test no | Test scenario | Test input data | Expected result | Actual result | Pass/fail |
| Bb8 | Update customer | Name: anil thapa  Address: ktm  Email:  [thapa2@gmail.com](mailto:thapa2@gmail.com)  Remaining field constraint as it was | Must change the data from previous one | Successfully update the data in database | Pass |



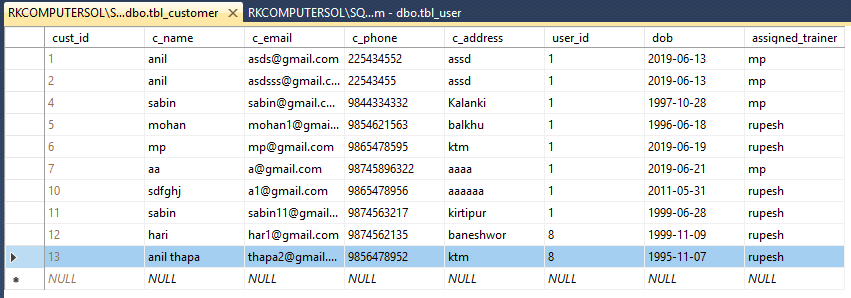
Testing figure data before updating



Testing figure change in data for updating



Testing figure changed date updated

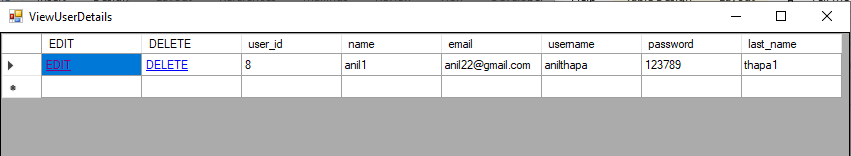


Testing figure changed data into database

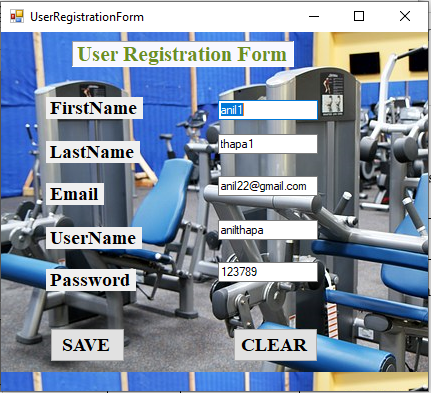
Test-name: update user data test

Pre-condition: after dashboard user can update their profile

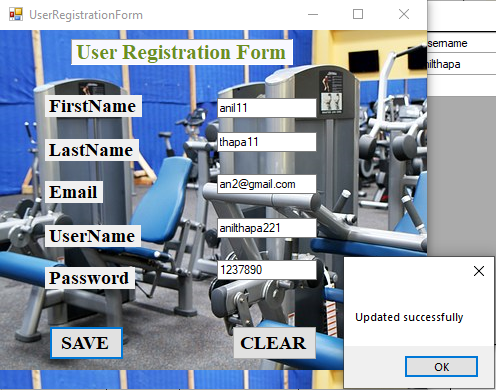
|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Test no | Test scenario | Test input data | Expected result | Actual result | Pass/fail |
| Bb9 | User can update their details | Email: [an2@gmail.com](mailto:an2@gmail.com)  First name: anil11  Last name: thapa11  Username: anilthapa221  Password: 1237890 | Data in the database can change and user must input the updated date in-order to login next time | Successfully updated their data without errors | Pass |



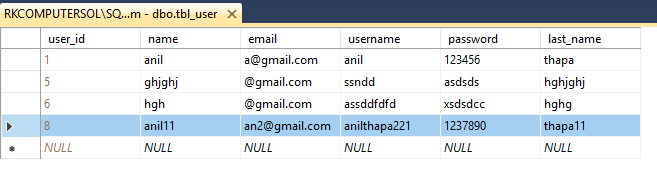
Testing figure user data before updating



Testing figure old data to changed



Testing figure new data to be changed

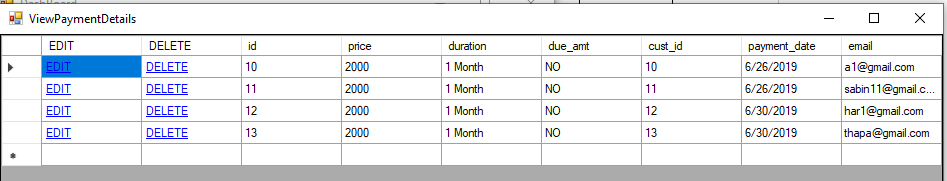


Testing figure change data in database

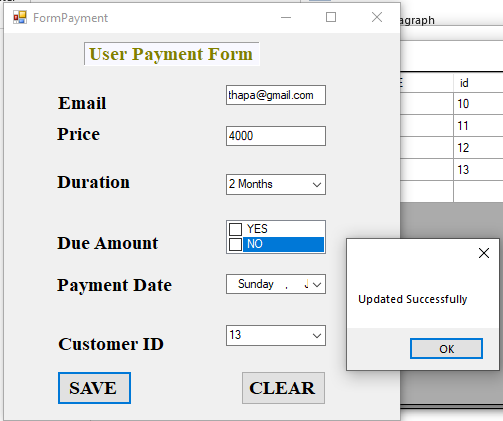
Test-name: update customer details test

Pre-condition: after payment that data can be updated

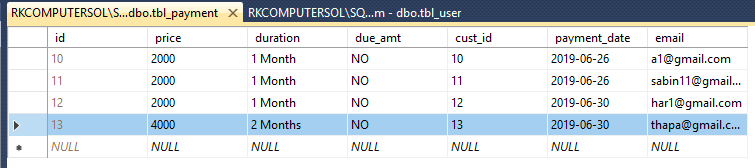
|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Test no | Test scenario | Test input data | Expected result | Actual result | Pass/fail |
| Bb10 | User can update their payment details | Email: [thapa@gmail.com](mailto:thapa@gmail.com)  Price: 4000  Due amt: NO  Payment date: 6/30/2019  Duration: 2month | Data in the database can change and user must seen the updated data | Successfully updated their data without errors | Pass |



Testing figure data before updating of payment



Testing figure new data added to be updated



Testing figure updated data from database