Chapter 1: Introduction

* 1. Over view of project

Gym management system used to minimize the work load of the customer and their staff. By developing the systematic record details through software. The user can create profile to access the system. After that able to use all features inside it. Can add user, user trainer, payment of customer, trainer for gym. User can search details of customer. Can see the total payment and number of customers at dashboard.

* 1. Justification of project
     1. background of project

The system is generally for the desktop-based application where user can record and add details as per software. User can have access after the login and from the dashboard the major functions are usable otherwise it is not possible. Customer can join and can have trainer on the gym. Payment of their details are also recorded.

For the development of application visual studio used and to store the data database server as MySQL 2014 used. I have followed the model, view, controller (MVC) pattern for this project.

* 1. Overview

The whole system completes with the all kinds of requirement which perform effectively. But some of the features listed in should have, could have are not able to do because of time. The system can perform well and accuracy is maintained.

The dashboard shows the basic total customer, total payment and some of other data. The program is small but perform the better performance with all requirement.

* 1. Aims and Objectives

Aims:

The main aims of this projects are:

* To implement the new system
* To reduce the redundancy of data
* Time consuming method to erase
* Better, faster and reliable record of data

Objectives:

* Reliable data store
* Accuracy and maintainable system
* Retrieve of data without any problem
* Reducing tedious from work

Chapter 2: Analysis

2.1 Introduction to analysis

Analysis is the process of breaking down of complex form into the smaller datum. In this project the task has been also decomposed into different categories in order to perform more accurate and easier. The analysis parts consist of the feasibility study, requirement analysis consist of functional, non-functional, Moscow and SRS, use case and class diagram.

The some of the activities that are involved in the analysis are described below:

Requirement determination:

In this stage requirements are gathered for the system that is going to be developed. But in this case requirements are with the most prioritized are gathered. They must be included in the system and parts of the system.

Requirement specification:

The system or the application on which requirements are listed in order to fulfill the requirement held in this process. Users specifications are gathered and collected as what actual need through discussion.

2.2 Feasibility study:

The feasibility study shows the requirement that actually meet the requirement with user needs. It is also as a cross checking for the product identifying the technical, operational, efficiency of the product.

Final statement:

This shows the actual achievements of the systems. What things have been achive and left to do. Shows the stages or level of achievements still required and fulfill by the team shows in the SRS formats. In this stage maximum requirements have been captured.

Hardware requirement:

In this stage the products that developed by the team goes to its size and requirement for the overall management are analysis. If system took low or high amount of resources depends on the size of the project. Hardware, data, response time, volume are measured in this level.

System design:

This is the main parts of the projects where real-life implementation of the requirements are captured and implementation. Building of database, logical design, testing, implementation is done in this stage.

Implementation of system:

Includes of coding to the system, testing, user testing, database and documentation are done in this stage.

Evaluation:

Evaluating the system from the technical and the user views. Meet the requirement and still have doubt. Can be improve or can be listed the problem for better improvement.

Modification:

The unapproved or default listed on evaluation are further used as modification of system. It generally helps to improve the system from the adding, updating, fixing the features.

The main reasons for performing the analysis are:

* Can easy to identify problems:

The major information is to identify the problem before the project begins. So, that the problem can be easily minimized before the begin.

* Task can be prioritized:

Major systems and their requirements can be prioritized before the beginning of the work. So, according to the requirement system can be developed according to the major prioritized.

* System can be breakdown:

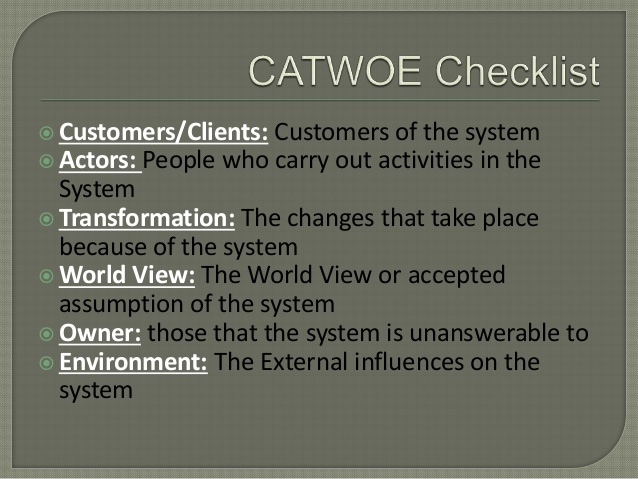
So, according to the needs as per the work the system can be breakdown in order to divide the task. So, brings up the piles of work into the routine wise schedule. Helps in the effective and efficiencies in the system.

Requirement analysis

Requirement can be easily identified if the analysis process can do early stages. Including economic, technical, resources and other requirements.

CATWOE analysis:

For this project CATWOE analysis is preferred as the main analysis as it confirm to the business and decision making. It is the technique for undertaking the customer or stockholder problems and follow the different processes in order to finish the problem trend. (Elmansy, 2015)



The CATWOE analysis includes of number of stages which are explain below:

Customer:

The team manages to solve or gather the problematic condition of user or customer. In this project the customer is gym management system where I am stockholder the requirement given for the project as a client.

Actor:

Refer to those who participate in the situation directly or indirectly might be stockholders or employee. This project belongs me as actor because of I am involved to carry out the system.

Transformation:

Determining the process of inputs and their outputs in this system. What kinds of input does it takes and through back the output? And all the processes between them are realized. (pham, 2014) for the gym management system

World View:

It simply clarifies the definition and the understanding of project with problematic and its solutions. It refers to the important framework of CATEOE analysis in this assessment.

Owner:

Represent the person who fully have control over the project and stand against the system to lead whether to implement or not. The owner has fully control over the system and lead to the overall project till the end with the highest decision.

Environmental constraint:

The environment constraint gives the meaning of internal and external matters that affects will be constraint through over the projects like resources, financial, laws, ethical factors. (Kukhnavets, 2017)

2.2 feasibility study

A feasibility study determines the wide range of area which includes all the causes that needs to be view to implement the project. The study shows technical, legal, social, environmental, economic, resource. The feasibility shows the major identify whether the project begin or does not meet the feasibility requirement. (mukund, 2018)

The feasibility study is classified into the different sub tasks are:

1. Economic feasibility:

In this assessment economic feasibility includes cost that need to required during the task. As like electricity, components, devices that need during the projects. In this assessment the economic feasibility is possible to manage for the project.

1. Technical feasibility:

Generally technical feasibility show available resources is capable of converting them into to productivity. This includes the hardware, software and other determines that how it will make different in working system. In this assessment the technically requirement is possible.

1. Schedule feasibility:

The project is divided into the different sub tasks in order to finish the product on schedule. The chance of pass and fail determined by the schedule time and achievement. In this assessment most important is scheduling and success. The time and schedule conducting throughout the project is possible.

1. Operational feasibility:

It shows the delivery product and their productivity against the complete product. Weather the product satisfied the requirement well or not. To satisfy the terms and the requirement in this assessment can be success to manage.

1. Legal feasibility:

The legal parts also need to be view while drawing this assessment because whether the legal conflict brings. The legal feasibility helps to clarify the legal site of project whether the boundary on favor. The project that I am going to developed is not against legal and should be legal free software to developed.

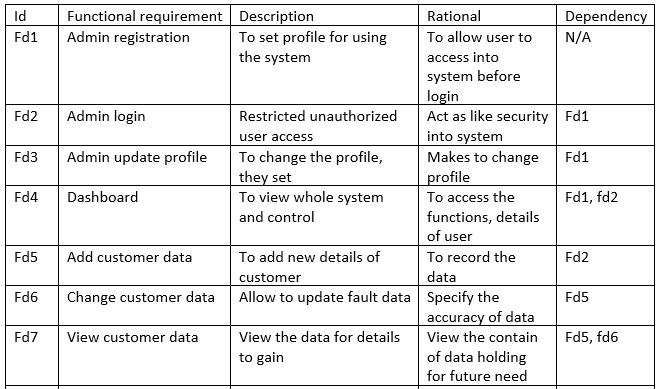
2.3.1. Requirement analysis

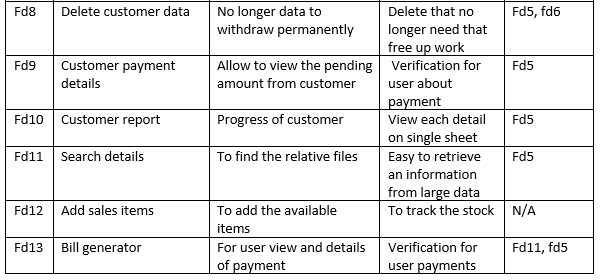
Requirement analysis is the process of identifying the major requirement of the customer in order to meet the quality, acceptance and upraise from the failure. It defines the user expectation in the application.

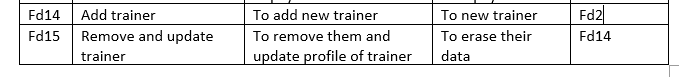
* + 1. Functional requirement:

Functional requirement is those behavior of the system that made the system more accurate and well balanced. Make the system more accurate, efficiency, acceptances, performance and well developed with user perspective. (eriksson, 2012)

* Admin registration
* Admin Login
* Admin update profile
* Dashboard
* Add customer data
* Change customer data
* View customer data
* Delete customer data
* Customer payment details
* Customer report
* Search details
* Add sales items
* Bill generation



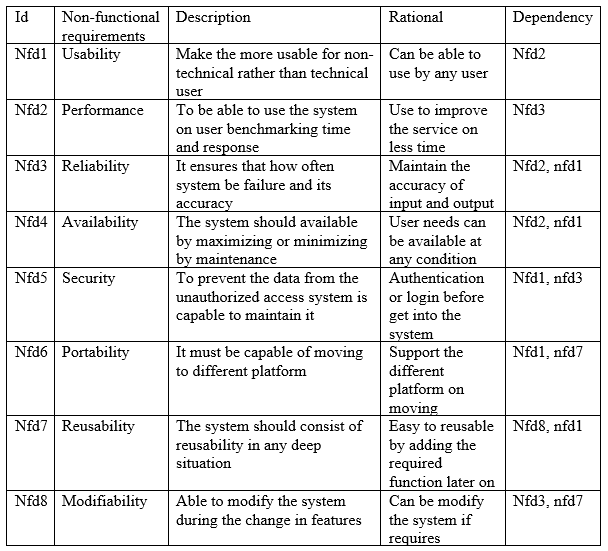




* + 1. Non-functional requirements:

Basically, the non-functional defines the how the system works. Most of the requirement are based on functional due to which large gap between functional and non-functional are common. (Eriksson, 2011)

* Usability
* Performance
* Reliability
* Availability
* Security
* Portability
* Reusability
* Modifiability



* + 1. MOSCOW prioritization:

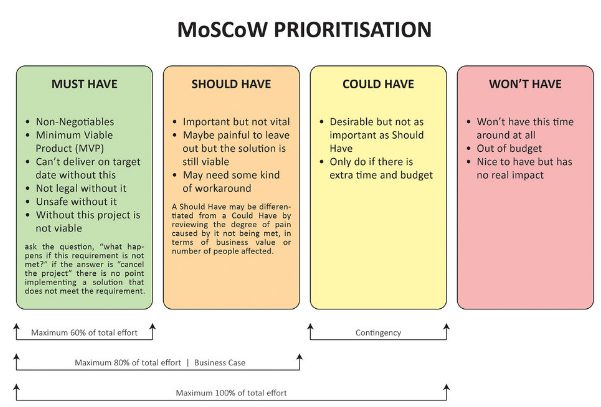
The MoSCoW prioritization shows the priority given to the stockholders and their needs. It implements the must have, should have, could have and won’t have the four principle.

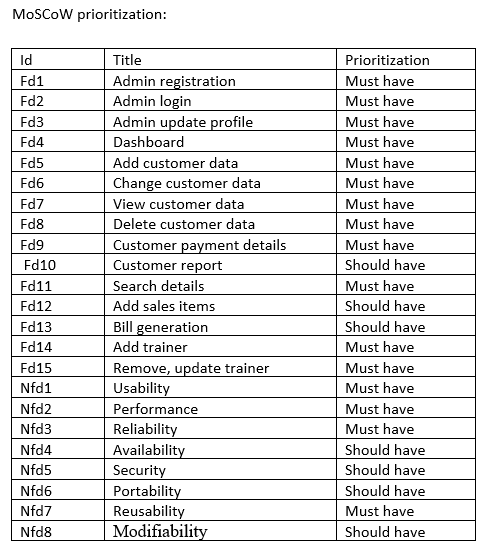
M stand for: must have which should be include

S stand for: should have which is necessary but not include

C stand for: could have which will include on later on if necessary

W stand for: won’t have ignored by stockholders



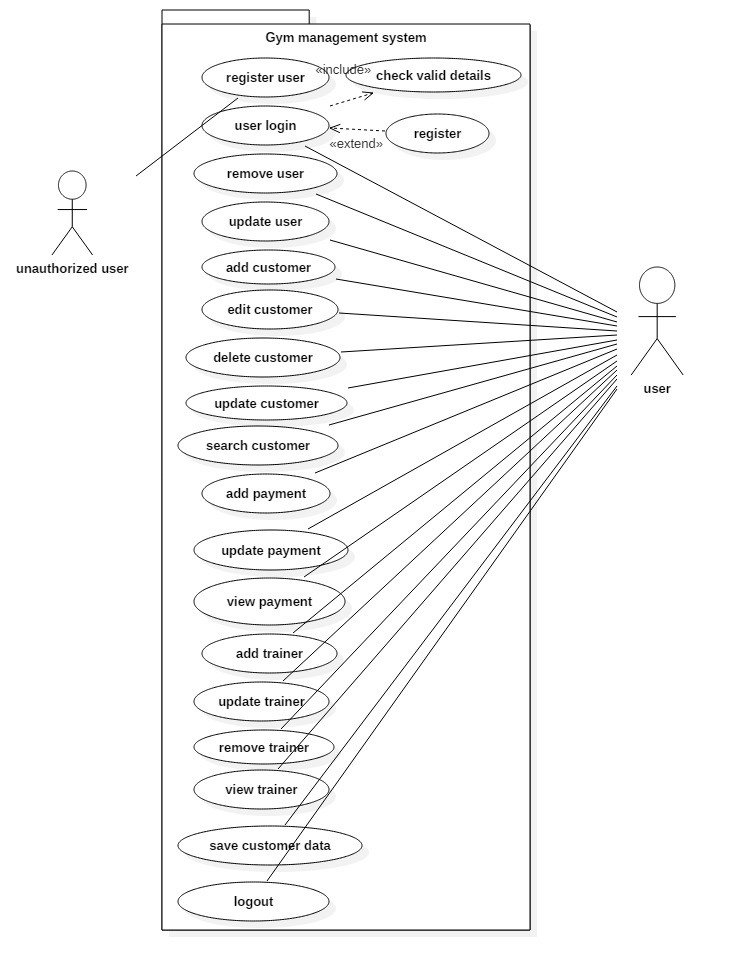


* + 1. System requirement specification (SRS)

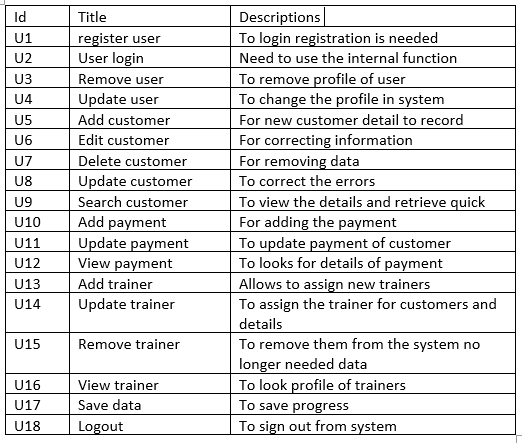
System requirement specification includes of all the functional and the non-functional and also a stockholder requirement. It helps to improve the project and prevent from failure. This also make it to reduce the redesign process. In this project the requirement for the project development are

|  |  |
| --- | --- |
| Operating system (OS) | Windows 7/8/9/10 |
| System type | 32-bit or 64-bit operating system |
| Processer | Intel(R) core M480 2.66GHz |
| Ram | 2GB/ higher |
| Database | Microsoft SQL server |
| Programming language | C# |

2.4 Use case diagram:



screen Use case diagram of gym management system



User story:

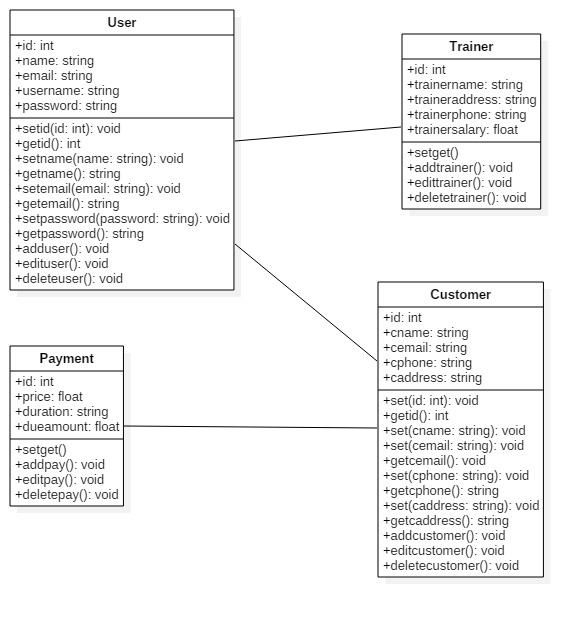
Gym management system is appliation that used to manage day to day activities to keep the record details of the customers. The system will keep the data recorded in the database. The system is handled by those who operates preffered as admin. The cuatomer can provide the details on the front desk and system use can insert the data into the system. The main system user have privillage of data insert, update, delete as well as view.

The customer can get their data to view if he/she prffered to view it. The system also tried to sales the items from the place but not as important. The customer payment details also recorded . For the billing process the management can have alternative possiblities. For the easyness the search for the details also be included. Their time period also recorded as trainer assged them.

The selected nouns, vebs and adjective are given below from the user story:

|  |  |  |
| --- | --- | --- |
| Candidate class lists (noun) | Attribute of candidate list (adjective) | Canddate list of (verb) |
| System, customer, user, trainer | System, details, record | Insert, update, delete, view, edit, remove |

2.5 Initial class diagram:



screen : Initial class diagram

Conclusion:

The gym management system analysis part is done with the user requirement. For the better and quality, the functional and non-functional parts are done through in this assessment. The term of requirement is shown in the form of MoSCoW prioritization. The use case diagram shows the actual movement of the system how it works and initial class diagram also added in the analysis part.

Chapter 3: Design

Introduction:

In gym management system the design of different perspectives plays the significant role to achieve the project. In this project the design parts involve of different pattern of sub division of the parts of system. This includes the structural design that consist of class and data flow diagram. The behavioral design includes activity and sequence diagrams whereas database includes data dictionary and ER-diagram. The architecture design has user interface and prototyping.

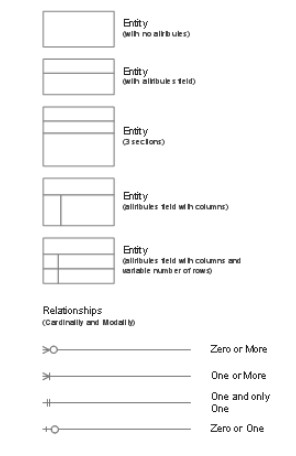
For this project the requirement of designs is sketch through the help of star UML. This star UML helps to create the diagram makes it easy to sketch. Unified modeling language (UML) is used to build software and relative of any software system.

3.1 Structural design:

The structural design pattern basically shows the relationship between entities. Shows the object and classes how they are interrelated to each other combination of large complexity into easy design. In this project this structural design makes the project more complex into simple easy understanding form to know how the system going to implement. (tomar, 2012)

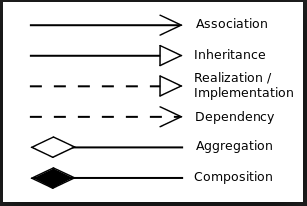
* + 1. Class diagram (final):

The class diagram shows the relationship between each classes and their class name,attribute and methods. In this project the relationsip are shown through the simple association that is solid line.



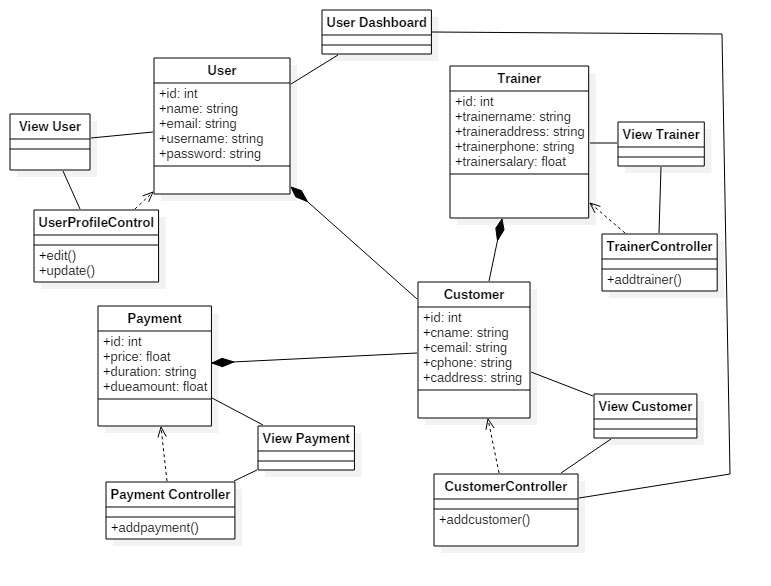
The some of the notation used in the class diagrams are shown in above screen. Some of them are described below:

* Entity: they can be used to identify the class which comes from the user story by filtering it.
* Entity attribute: they can be considered at attribute of classes
* Entity method: represent the process made by the attibute to draw the process



This notation can be used to join the classes with other classes.

* The association: it indicates that object in one class have relationship with object in another class
* The inheritance: the behaviour of parent class transfer to the child class
* Aggregation: it indicate that parent class can exist if there is missing the single part of child class
* Composition: it indicates that child class cannot exist if there does not exist parent class



screen : final class diagram

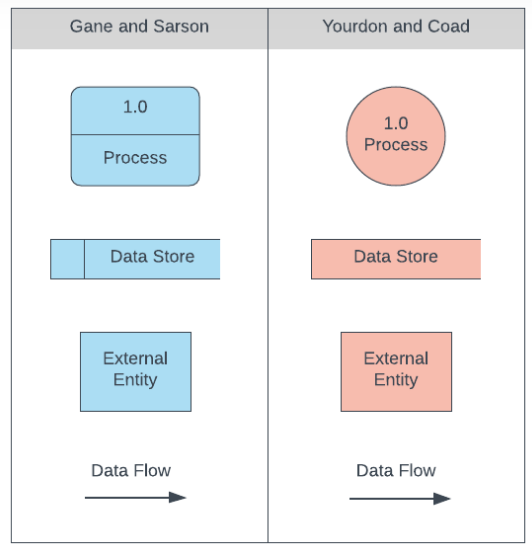
the class diagram shows the entity, method and the attribute into the system. Identify and display the role of class. Their attribute and the methods that implement by the class.

Important:

* It generally shows static diagram of any system.
* Better to develop for new or another member.
* Shows relationship between classes.
  + 1. Data flow diagram:

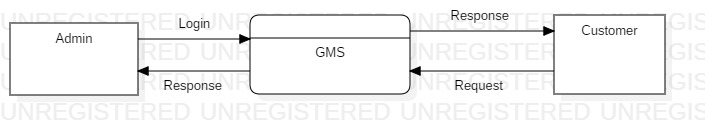
The data flow diagram shows the flow of information through the system and process. The different notation and symbols are used to identify the flow of data. They describe the entity and the relational flow. The flow makes the better understanding of system and improvement further on future development.

The data flow diagram are two types physical and logical. But the both diagrams shows same information flow.



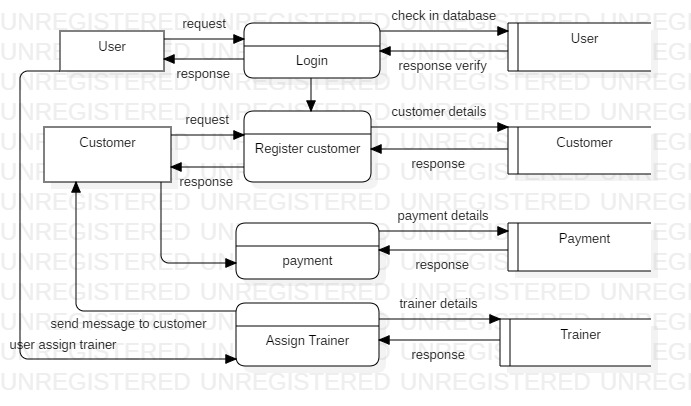
The above diagram in which the different notation has different meanings:

* Process: this known as process in which it known as input and output on DFD. Actually, transport incoming data to outgoing data.
* Data store: this stores the data that has been already processed. Input process are stored in the data and output is retrieved from database.
* External entity: they are the actor, entity of the system. They are initial beginning before the processed.
* Data flow: identify the flow from entity to the process and process to the database. They can be reserve back.



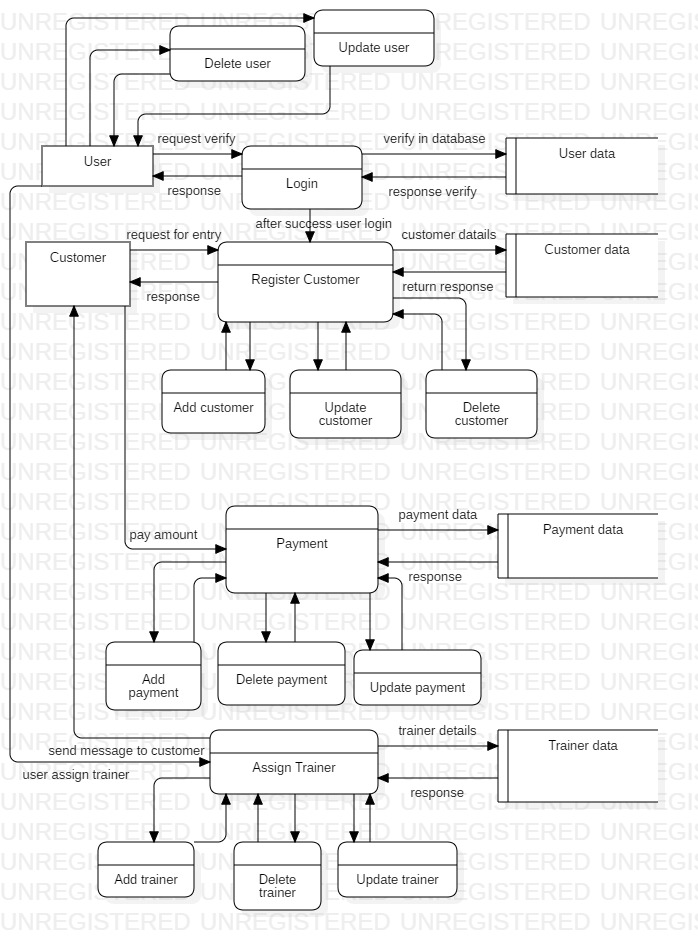
screen : DFD level 0

0 level DFD does not have more details but shows that the flow of information from external entity into the internal process.



screen : DFD level 1

level 1 DFD shows the flow of information from the entity to the process and from process to the internal database system. How they perform the function between them?



screen : DFD level 2

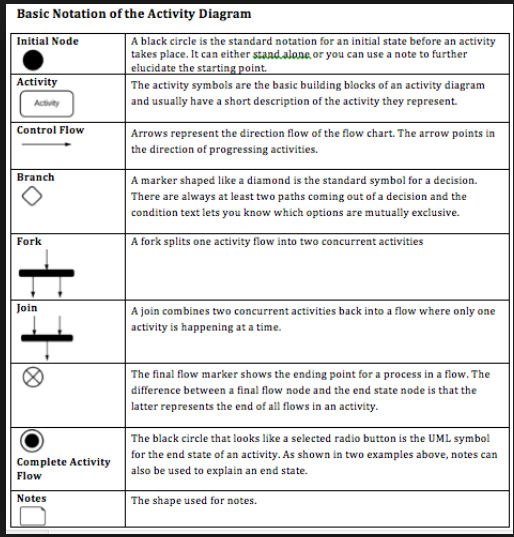
level 2 DFD shows the entire process of the entity and their major activities. From CRUD process to their impact into database and processes.

3.2 Behavioral design:

The behavioral design represents the activity and sequence diagrams. This diagram helps to identify the activities of user and admin whereas the sequence show the flow of order of data.

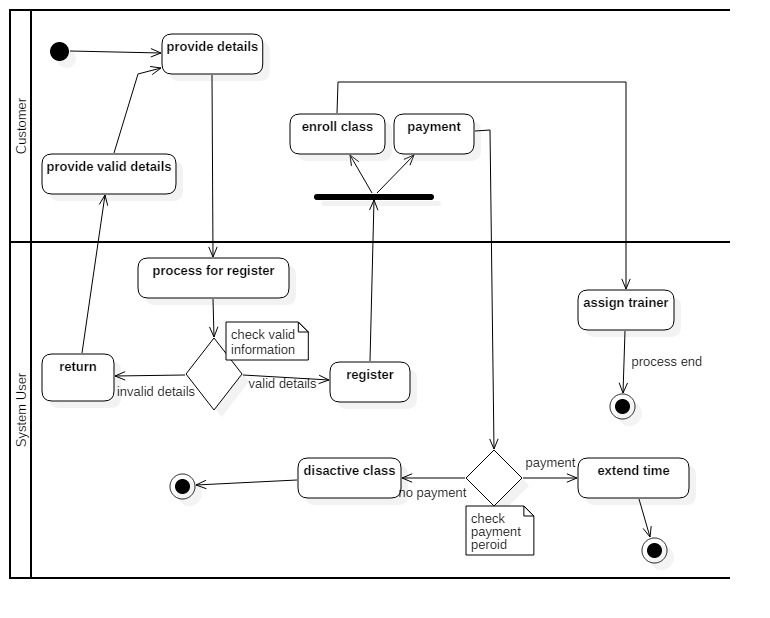
3.2.1 Activity diagram:

The activity diagram actually performs or shows the flow of activity from anther activity. This represent dynamic aspect of system. This diagram is not used for the visual capturing but also used for forward and reserve engineering. (Warren lynch, n.d.)



The some of the notation that are sown in above figure. This notation can have meaning with their used. This used combined formed the activity with their meaning. Some of them are described below:

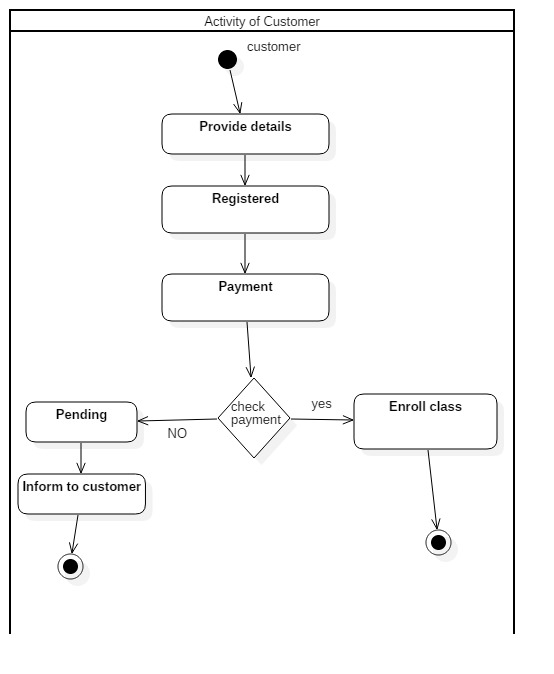
* Initial node: used for starting the system.
* Activity: The activity shows processes in a model process
* Control flow: used to show the flow of data and also preferred as direction flow or control line
* Branch/ decision symbol: identify the decision make during activity
* Fork: helps to split the activity into multiple process
* Join: makes the split of activity into single activity or synchronization
* The flow final symbol: show the ending process flows in a single flow in an activity
* The end symbol: represent the complete end of the process
* The note symbol: allows the diagram creator additional message that do not in the diagram itself



screen : Activity diagram

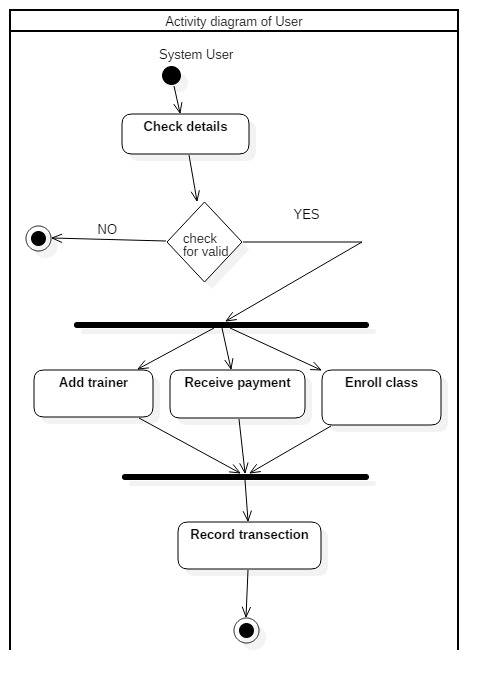
the main activity diagram of whole scenario where the interaction of user with customer are shown. The customer begins the details provided to the system user to register or enroll into the gym. Which flows can be clearly seen in above diagram.

The Swimland divides the activity between the system user and the customer. The notation can show activity of them and flows.



screen : Activity diagram of customer

this is activity diagram of customer shows the flow of interaction of customer with the system user. And activity only done by the customer during the process interacting with system user.



screen : Activity diagram of system user

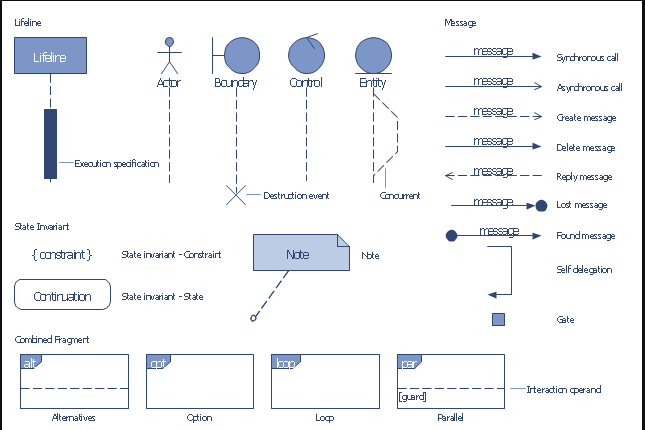
this is the activity single diagram of user. Visualize the activity done by the user considering on certain activities belongs to user during the performance on system. The main system user can make such activities during process.

Importance of activity diagrams:

* Detect order of flow of activity
* Modeling the work by activity
* Easy to model the business requirement
* Understanding of system from high functionality

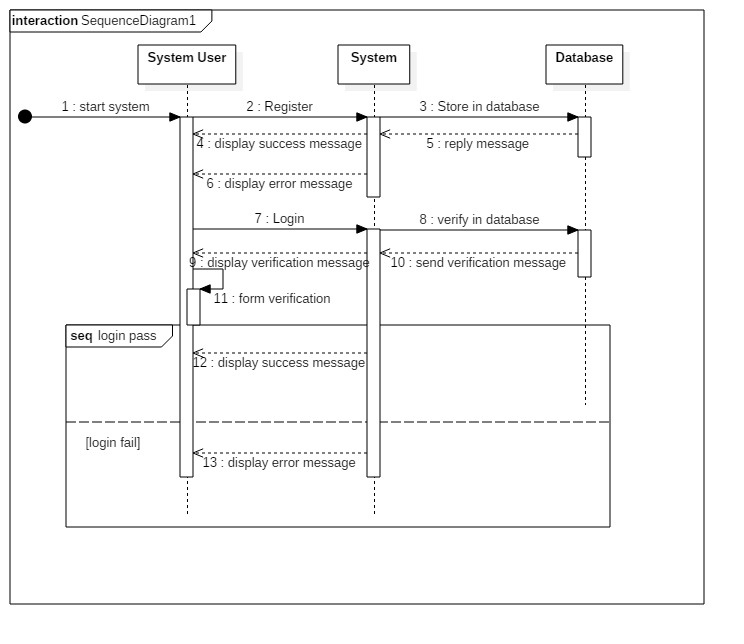
3.2.2 Sequence diagram:

The diagram helps to identify the each and everything details in this project. It describes the interaction of the different parts of the system. In this project the sequence diagram shows the actual interaction with different entity with each system.



The sequence notation is described below only some of them:

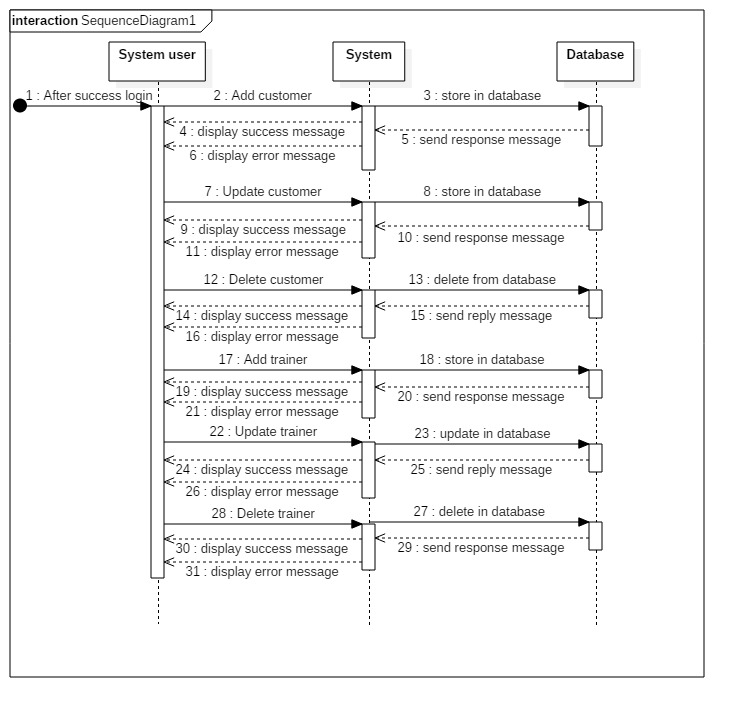
* Object lifeline: indicates that how long the entity life line exist during the whole process of sequence
* Activation: represent the active of the flow of message and the entity that involve on it
* Message: identify the flow of message between the object
* Message synchronous: the kind of message that need to wait response before continuity next
* Message asynchronous: this message does not need to wait for the response
* Return message: the message that come back after send to certain object
* Self-delegation: offer the self-message within the object
* Found message: initial beginning indicated by it
* Alternatives: provides the condition to apply on it
* Loop: consist of same work to continuity for the certain period of time



screen : Sequence diagram

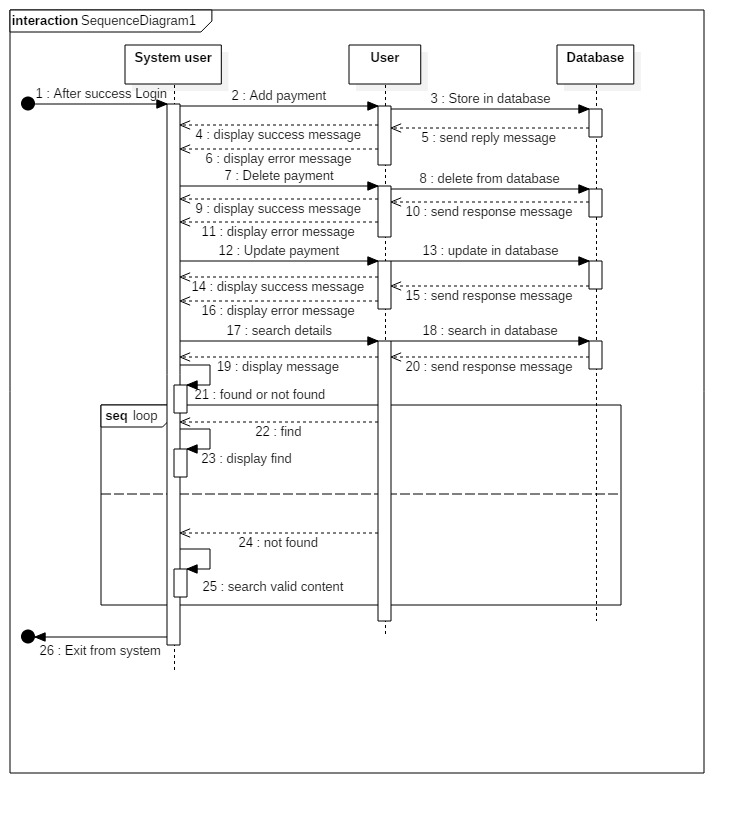
the initial part of sequence diagram that shows the register for login and the reaction between system user, system and database how actually perform the flow between them before beginning of system. From where the system begin and what function should need to perform to login into system.

The user first register into system before login into the system. And the flow of process shown in the above diagram.



screen : Sequence diagram

the second part of the sequence diagram where user perform the system after success login into the system. The different activities can be done after login success. The additional major features can be access to run the system. That includes all features of system.



screen : Sequence diagram

in this diagram the user can perform the continuity in the search case until the user cannot found the search item. So, loop process is used in search part of sequence diagram. And after the complete process the user can exist from the system.

Important of sequence diagram:

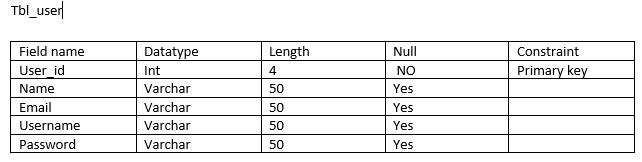
* Deletion and creation of object
* Identify the object actions
* Identify the lifeline of entity
* Makes the reverse engineering

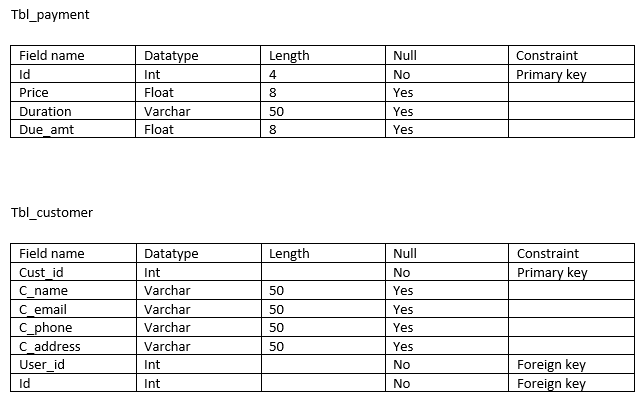
3.3 Database:

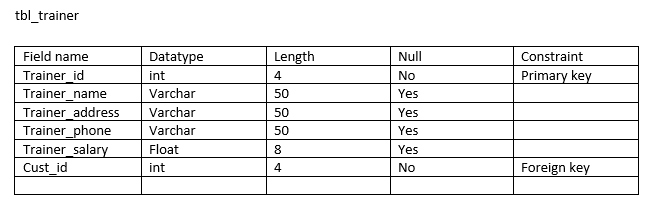
The database shows the data and their relationship to store the procedures. In this project the database includes data dictionary and entity relationship diagram (ER diagram). The ER- diagram is all about relationship between each entity or classes. Data dictionary includes metadata, datatype, data length and many others.

3.3.1 Data dictionary:

It is the details description of the data that are in the database. Each element in the database has its own metadata. In this data dictionary it identifies or shows the datatype, references, keys and more. Provides the details description of relationship, meaning, sources. (kononow, 2018)





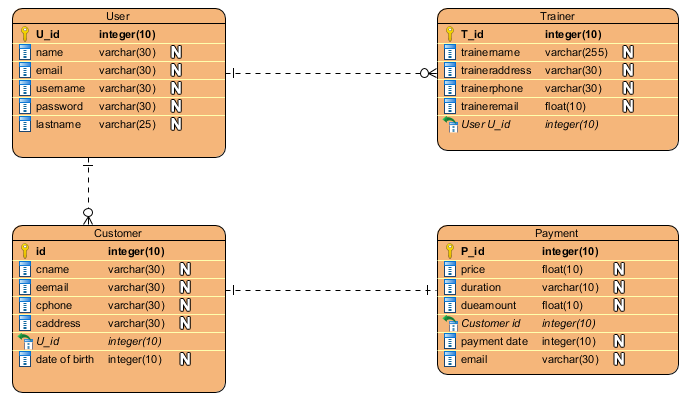


Important of data dictionary:

* Better during utilization of software requirement of data
* Easy to understand nature of data
* Provides the referential integrity
* More secure encryption

3.3.2 Entity relationship diagram (ER-diagram):

The ER-diagram provides the better development of database and their relationship. Provides the relationship between each entity in the system design of database.



screen : ER-diagram

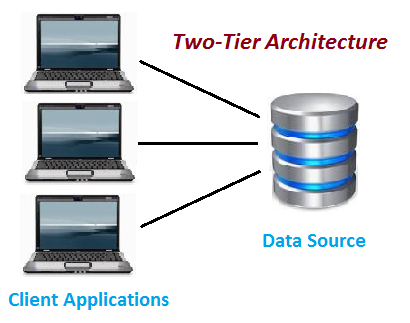
the ER-diagram basically shows the relationship between each entity. In above diagram shows the relationship between each entity involve in gym management system. Shows the clear views and better understanding of data through diagram.

Important of ER-diagram:

* Provides the relationship between each entity
* Easy to understand
* Visual representation
* Effective flow of communication

3.4 Architecture Design:

The outer look of the system before the implementation into product. It defines the total structure of the system exactly the same as in the architecture design. So, this design can be easily be understand by the implementation and knowledgeable.



Our desktop base application is based on 2 tier architecture because of the all the layers application and the server layer.

In this architecture the direct communication takes between the client and the server. Through the medium of application layer. In this system same server can be used by the multiple of user from multiple of application. (Evans, 2019)

The server is connected from the place into the application layer where it consist of all the code that directly connect to the server, where user can only face to the application interface to conduct the activities.

Important of architecture design:

* It is also considered as blue print of system
* Improve quality, function of system
* Visible only system structure but hide implementation process

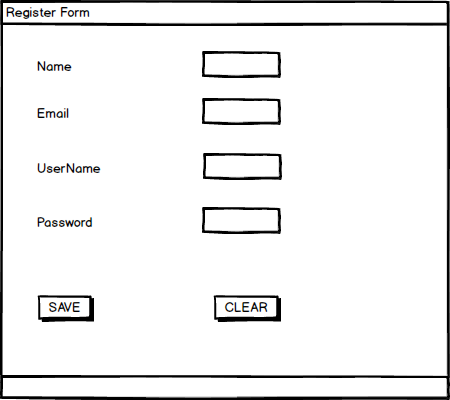
3.4.1 User interface design (UI Design):

The design is classified into two categories UX-design and UI-design. They are actually different from each other. They perform similar character although role is different. UX refers to user experience whereas UI refer to user interface design. The UX design is for customer satisfaction and improving the usability, use. Whereas UI design is the process of looking and feeling of the product what would be like. (LAMPRECHT, 2019)

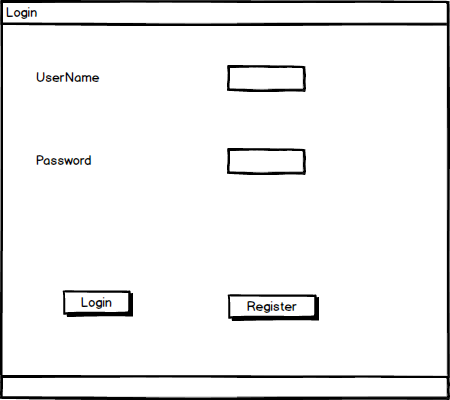
But in this assessment and project I preferred to use the UI-design for the beginning outlook of the program.

3.4.2 Prototype:

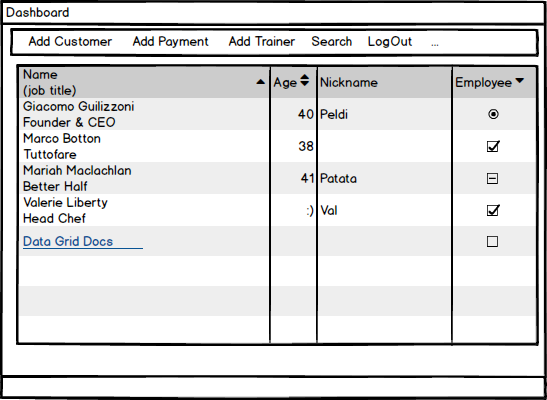
In this project the early version of prototype is made by using the Balsamiq mockups software to made prototype. Due to the easiness and easy to sketch.



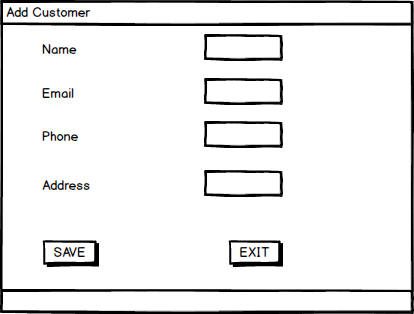
This provide the user to register into system before going to login. So, new user needs to register before to excess into the internal function of system. Prevent from the direct entry of unauthorized user to access the system features.



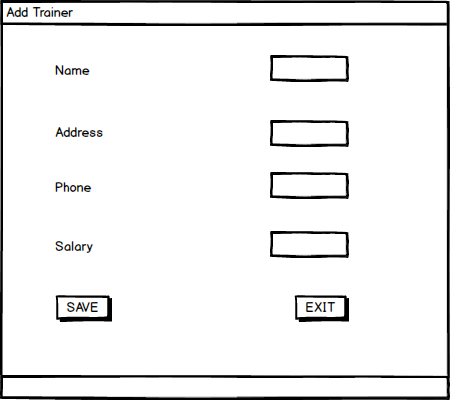
The second step of system is to login after the success register into the system. Each one the interact into internal mechanism need to login through this.



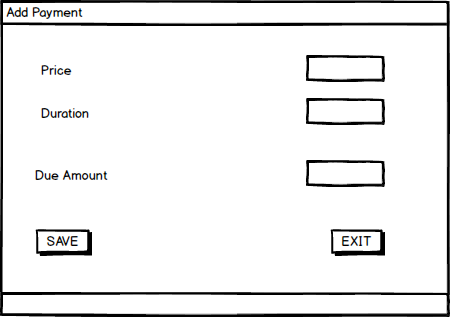
This is the main dashboard of the system. After login success this interface is open and allows user to access different additional features to operate. The user can add customer, payment, search, add trainer from this form.



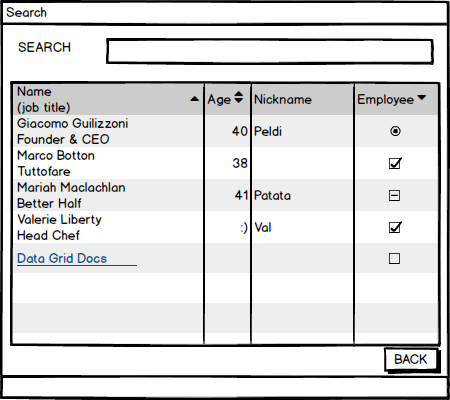
From this form the user can add new members into the system. This form can access after the success login and from the dashboard of the system.



This form considered as the data of trainer to add. This allows to add the details of the trainer in the gym management system.



This form can add the payment details of the customer that enroll in the gym management system. Keep the record of transection of payment.



This form allows to user to perform the task more easily. It allows the user to search the details of the customer exist in the system.

Importance of UI-design:

* Allows user to feel the early version of software
* Easy for the developer
* Reduce development cost
* More customer acceptance
* User friendly design

Conclusion

After the effort made on to design that includes all the format is completed. Although there are problems on making all the diagrams are made with the help of software references. Though the design phase of this project has been completed. Different parts of design are done through the required format and process with explanation.

Chapter 4: Implementation:

The part where the implementation of all the logical parts are carried out. This includes all the code that forms the parts of program and their basic needs of parts. It is the major parts of project where all the theoretical parts are gathered and developed the visible form.

4.1 programming language:

Language for the development are many in the real world. But I have familiar more in the C#, PHP programming language rather than other.

For this assignment gym management project have chosen the C# language to develop the project. This is easy to do because of object oriented programming language. The software I have used visual studio 2015 and for the database to store MySQL 2014 which as a server for the project.

4.2 Development environment

For the better result the development result must be better available for the people who involves in the project. This environment involves of different physical like software, quality, knowledge, testing, server. Such environment can make the quality of software n quality of environment. For my gym management system, the development environment provides by the help of MySQL, Visual Studio, references, internal and external sources, MVC framework.

Chapter 5: 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

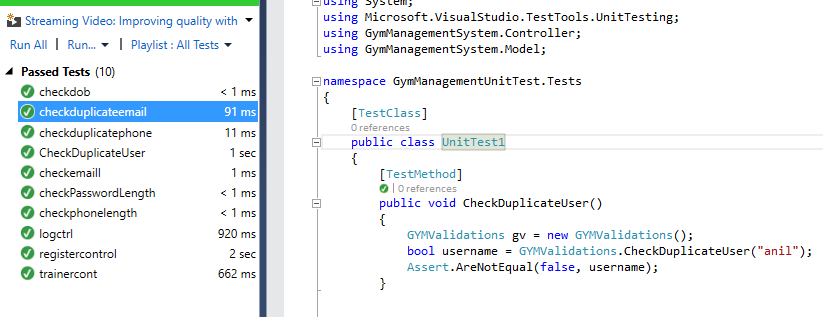
In this project black box and unit testing are considered as major testing.

5.1 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

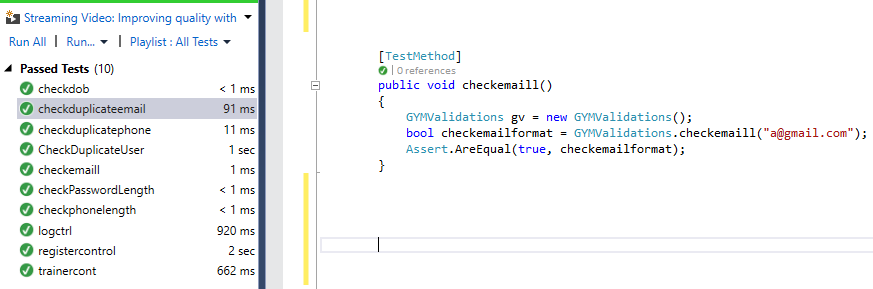
|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Test id | Test scenario | Test input data | Expected result | Actual result | Pass/fail |
| 1 | Check duplicate user | Username | Username should not be repeated | Duplicate data cannot be inserted with same name | Pass |



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

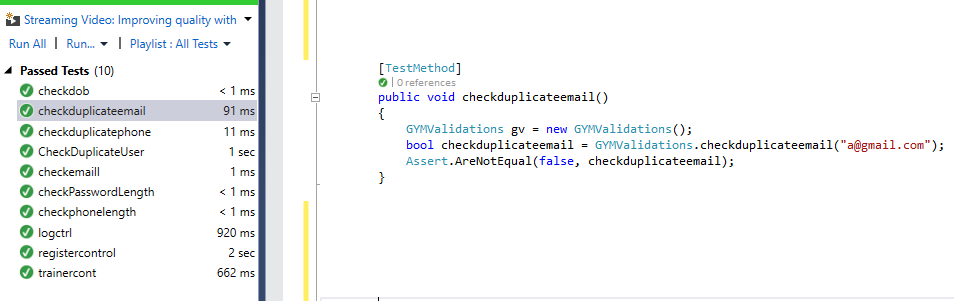
|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Test id | Test scenario | Test input data | Expected result | Actual result | Pass/fail |
| 3 | Check password length | Length must be greater than 5 | Must include more than five | Password must have greater than 5 length | Pass |



Testing figure checking password length

Test Name: checking duplicate email address

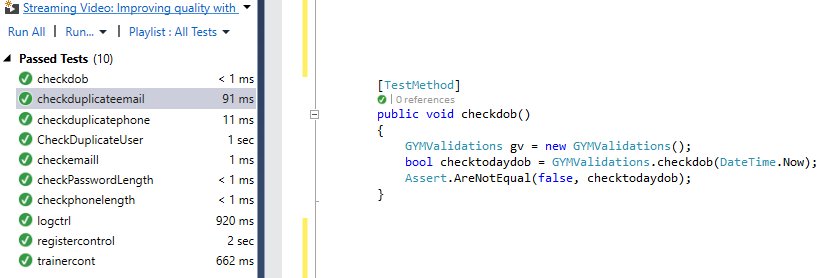
|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Test id | Test scenario | Test input data | Expected result | Actual result | Pass/fail |
| 4 | Check duplicate email address | Email address | Must not able to use same email address | Duplicate has been stope | Pass |



Testing figure checking duplicate email

Test Name: check date of birth

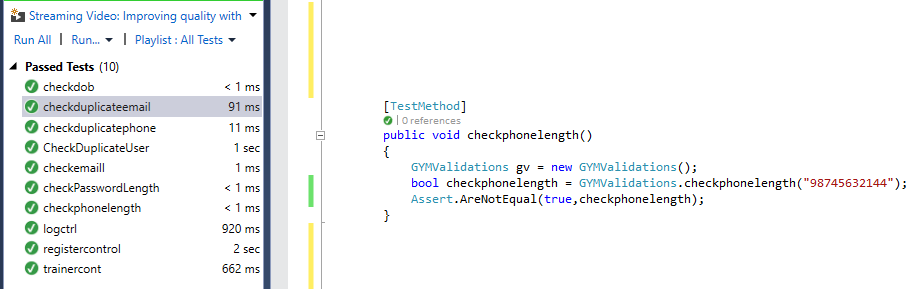
|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Test id | Test scenario | Test input data | Expected result | Actual result | Pass/fail |
| 5 | Check date of birth | Insert current date time | Should not ignore current date time | Must include before today date not coming date | Pass |



Testing figure testing date of birth

Test Name: check phone length

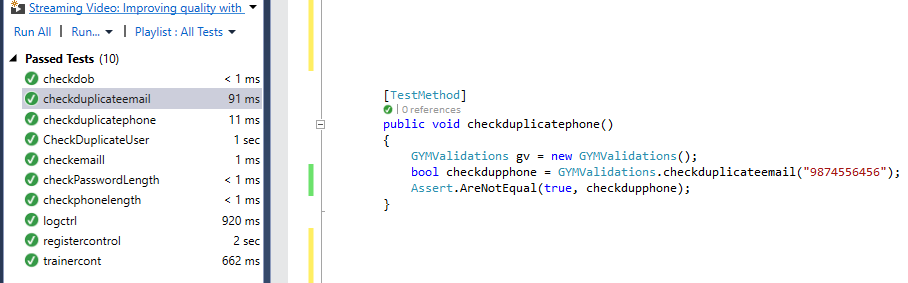
|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Test id | Test scenario | Test input data | Expected result | Actual result | Pass/fail |
| 6 | Check phone length | Length must be of 10 digits or less | Must ignore greater than 10 digits | Insert having less than 10 digits | Pass |



Testing figure testing phone length up to 10 digits

Test Name: checking duplicate phone number

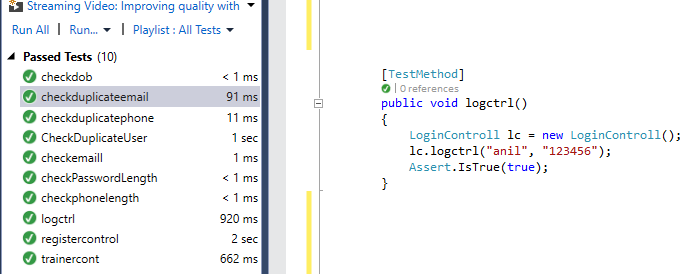
|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Test id | Test scenario | Test input data | Expected result | Actual result | Pass/fail |
| 7 | Check duplicate phone | Phone number | Ignore the same phone number twice from inserting | Must insert single unique phone not duplicate | Pass |



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

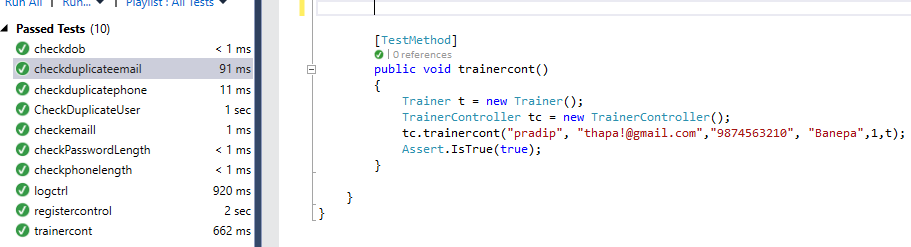
|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Test id | Test scenario | Test input data | Expected result | Actual result | Pass/fail |
| 9 | Check register form | Fill up all data | Must be register success | Success inserted into database | Pass |



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

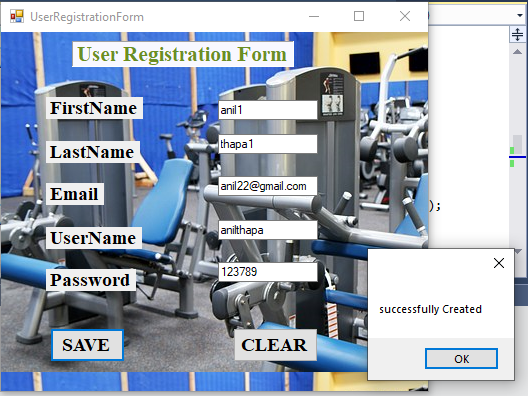
5.2. Black box testing:

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.

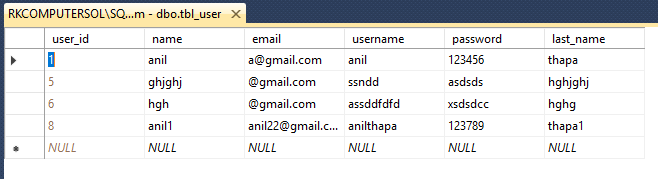
Test-name: registration test

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

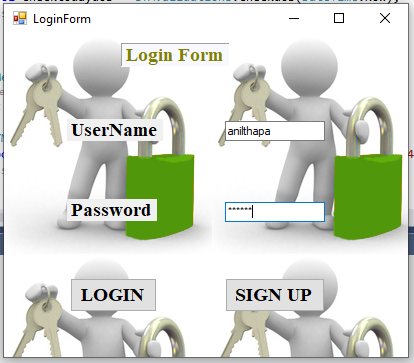


Testing figure view from database of success register data

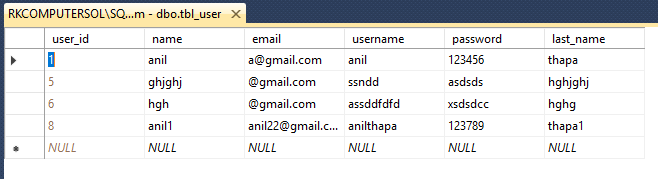
Test-name: login test

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

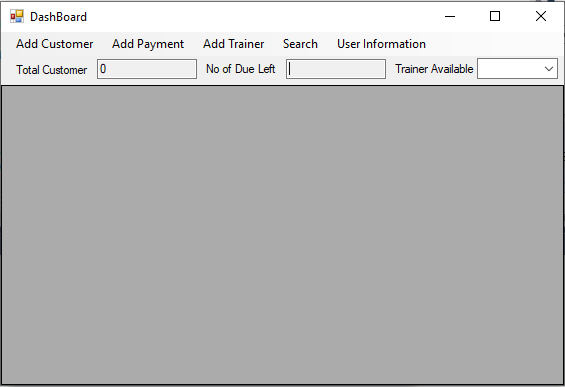


Testing figure user name and password of login data

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 |

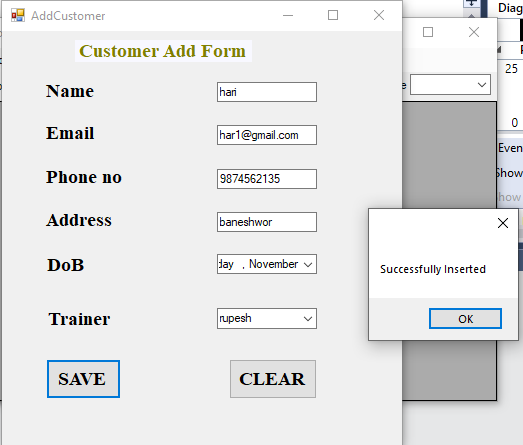


Testing figure after success login access to dashboard

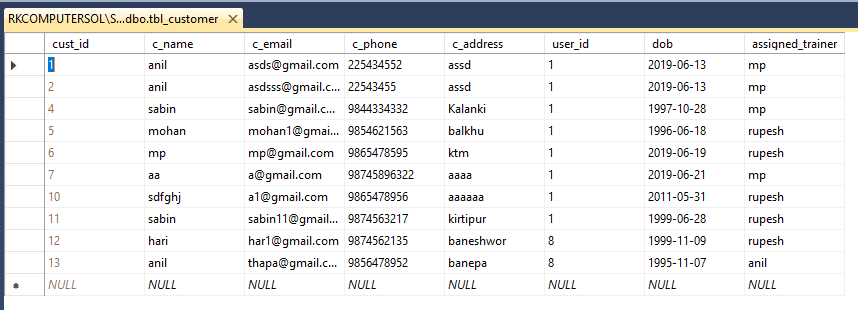
Test-name: dashboard test

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

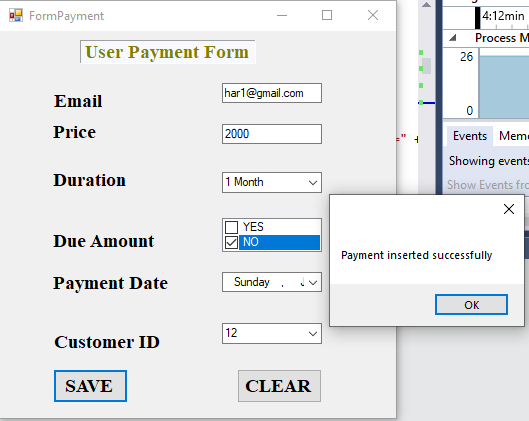


Testing figure data in database from interface

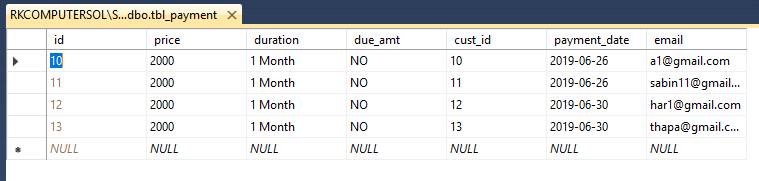
Test-name: user payment test

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

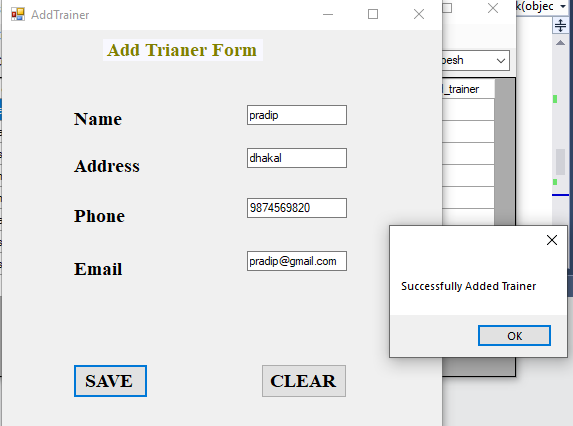


Testing figure data from database from form inserted

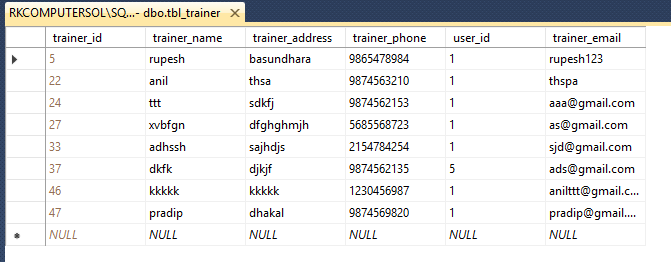
Test-name: user add trainer test

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

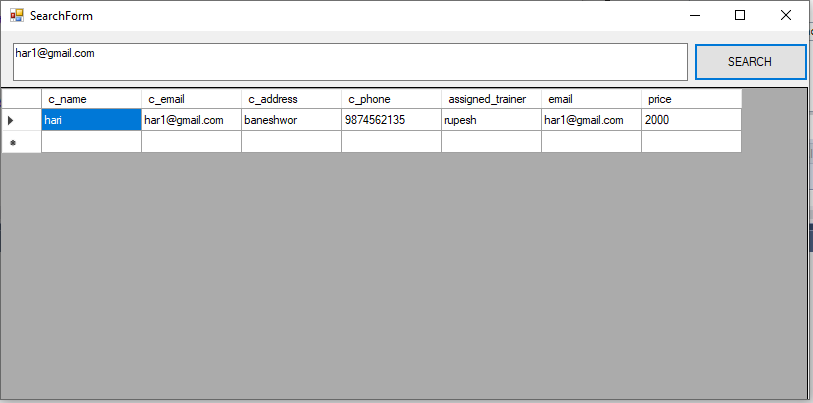


Testing figure data into the database from form

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 |

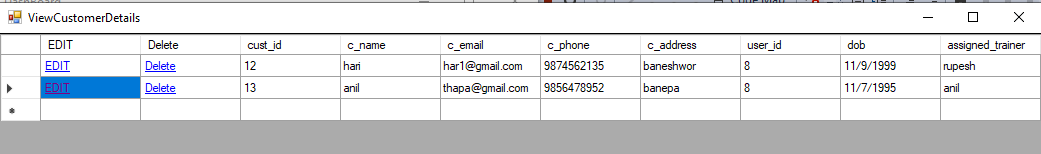


Testing figure searching user data through email

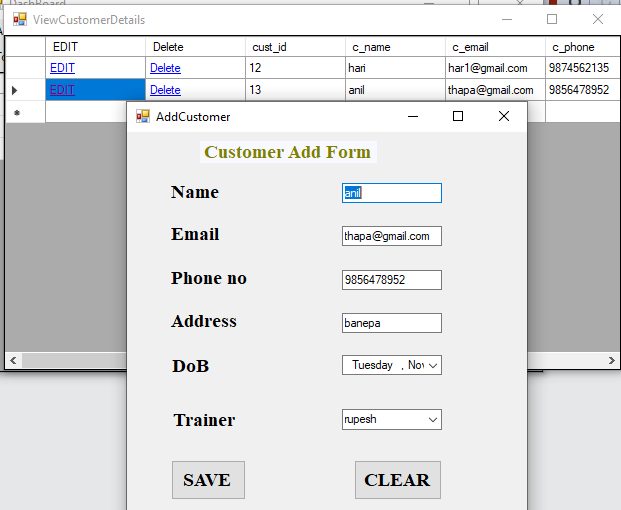
Test-name: update customer test

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

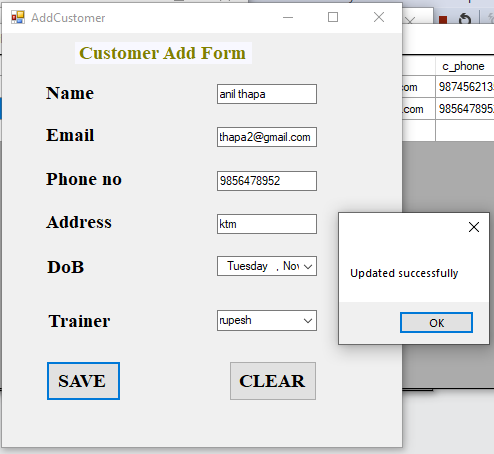
|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Test no | Test scenario | Test input data | Expected result | Actual result | Pass/fail |
| Bb8 | Update customer | Name: anil thapa  Address: ktm  Email:  [thapa2@gmail.com](mailto:thapa2@gmail.com)  Remaining field constraint as it was | Must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



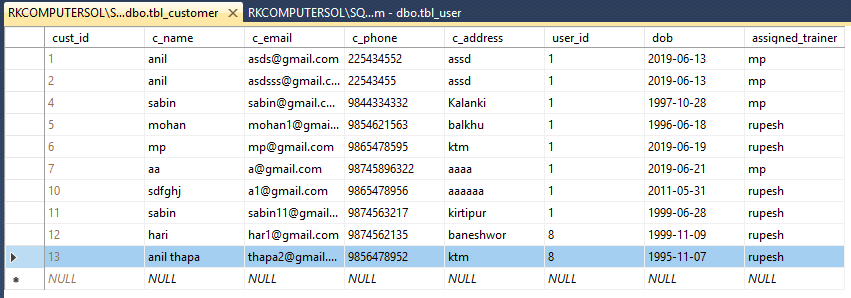
Testing figure data before updating



Testing figure change in data for updating



Testing figure changed date updated

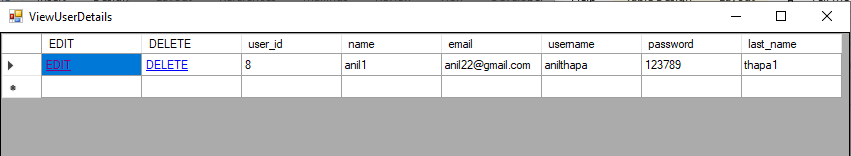


Testing figure changed data into database

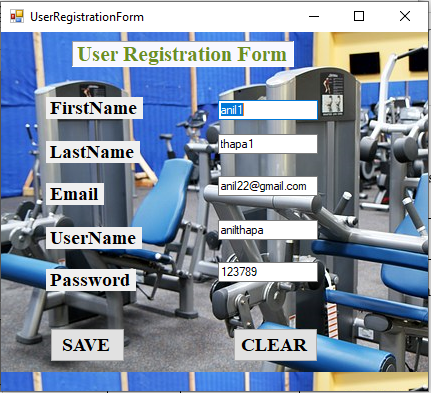
Test-name: update user data test

Pre-condition: after dashboard user can update their profile

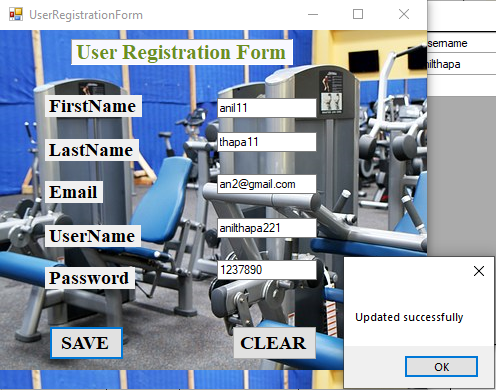
|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Test no | Test scenario | Test input data | Expected result | Actual result | Pass/fail |
| Bb9 | User can update their details | Email: [an2@gmail.com](mailto:an2@gmail.com)  First name: anil11  Last name: thapa11  Username: anilthapa221  Password: 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 |



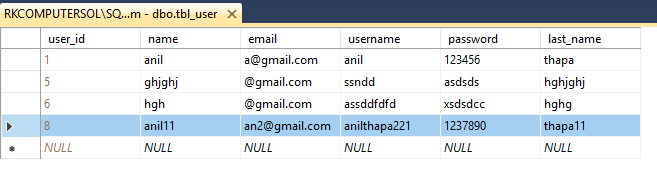
Testing figure user data before updating



Testing figure old data to changed



Testing figure new data to be changed

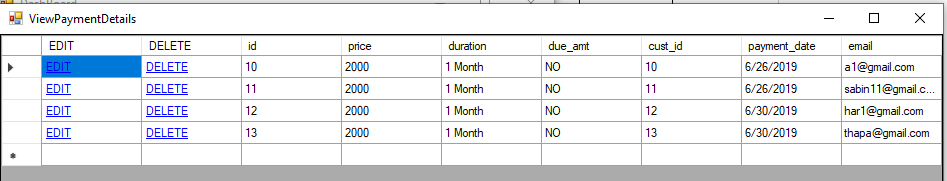


Testing figure change data in database

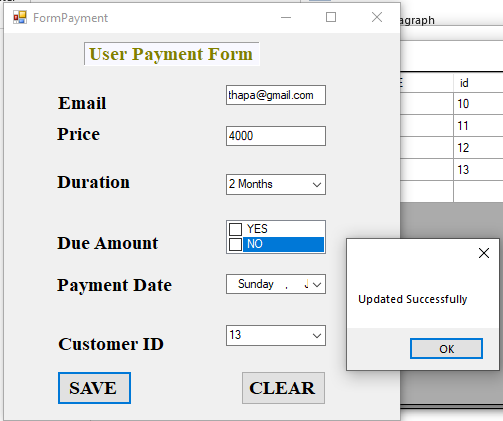
Test-name: update customer details test

Pre-condition: after payment that data can be updated

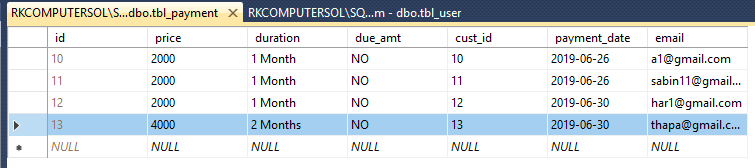
|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Test no | Test scenario | Test input data | Expected result | Actual result | Pass/fail |
| Bb10 | User can update their payment details | Email: [thapa@gmail.com](mailto:thapa@gmail.com)  Price: 4000  Due amt: NO  Payment date: 6/30/2019  Duration: 2month | Data in the database can change and user must see 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

6. Other project issues

6.1. Risk management:

Process of identifying the risk of consequences before the project begins. Forecasting the details risk on project costs, schedules, consequences, outputs, time in process. The visible risk is identified and minimized before project. They measured based on risk likelihood and consequences based.

Likelihood:

|  |  |
| --- | --- |
| Likelihood | Values |
| Low | 1 |
| Medium | 2 |
| High | 3 |

Consequence:

|  |  |
| --- | --- |
| Consequence | Values |
| Very low | 1 |
| Low | 2 |
| Medium | 3 |
| High | 4 |
| Very high | 5 |

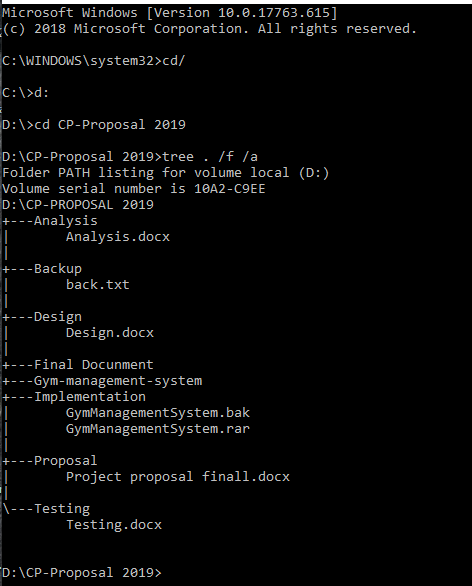
**Impact=Likelihood \* consequence**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Risk | Likelihood | Consequence | Impact | Action |
| Insufficient skill | 2 | 3 | 6 | Demo should be given |
| Disk failure | 2 | 3 | 6 | Backup of the data at regular base |
| Problem of schedule timing | 2 | 4 | 8 | Time must be managed within the schedule |
| Loss of data | 1 | 5 | 5 | Must be protected |
|  |  |  |  |  |

6.2 Configuration management

The process of handling the change, integrity in the system over a period of time. The major process is to identify any integrity in the system that can be identify on later on stages. It allows the version control made over the change occurs in the system. (Heidi, 2019)

Quick recovery, replaceable environment can be easy with the configuration management.



screen : Tree View of configuration management

6.3 project issues:

During the development of the system comes lots of problems to solve. The management of power supply comes first. Secondary part is the strong connection of the internet which is not much reliable. The failure of distribution of power supply makes the huge role to run the necessary items to conduct the project. Unreliable connection makes the project more complicated because needs to takes references from the outsources.

There are also the parts of implementation, testing, problems during the project. Needs to takes the different references from those who can solve it. Needs to consult with others. Managing the electricity, connection to internet, good reference for project, different methods for implementing, software to sketch the diagrams different diagrams.

6.4 limitation

* The system has not featured of adding profile picture.
* There is not printing of details of customer.
* There is not facility of selling items record.
* There is no biometric verification.
* No online payment system.

6.5 future work:

* In a future the system will be online accessible to users.
* Can be access from the internet access.
* There will be the more feature such as recording the trainer income details.
* Gym item can be sold and recorder into the system.
* Online payment system will be implemented.

Chapter 7: Conclusion:

The project gym management system has been conducted according to the task. In this step the project issues are collected and defines the various terms like risk management, configuration management, projects issues, limitation and future work carried out in this stage. All the parts of this project are breakdown into the six segments where all the segments are conducted as per schedules.

This system can make the day to day activities of user more effective and accuracy. The system can record the data of the customer, payment, trainer details. For this project MVC design pattern has been followed to complete the whole project.