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 where as database includes data dictionary and ER-diagram. The architecture design has user interface and prototyping.

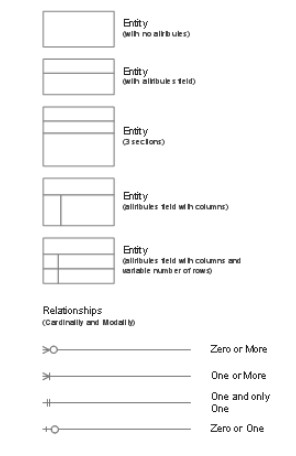
For this project the requirement od design are sketch through the help of star UML. This star UML helps to create the diagram makes it easy to sketch. Unified modeling language (UML) is used to build software and relative of any software system.

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)

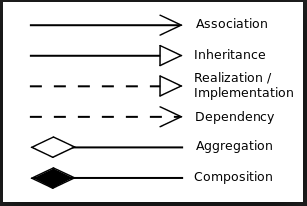
1. Class diagram (final):

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.



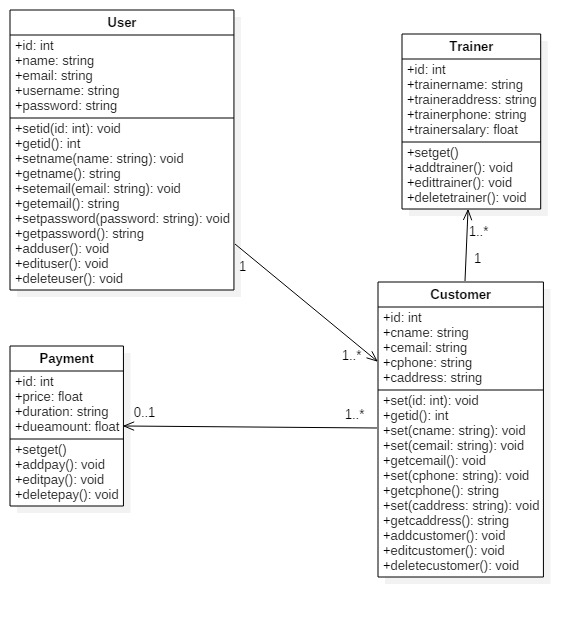
The some of the notation used in the class diagrams are shown in above screen. Some of them are described below:

* 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

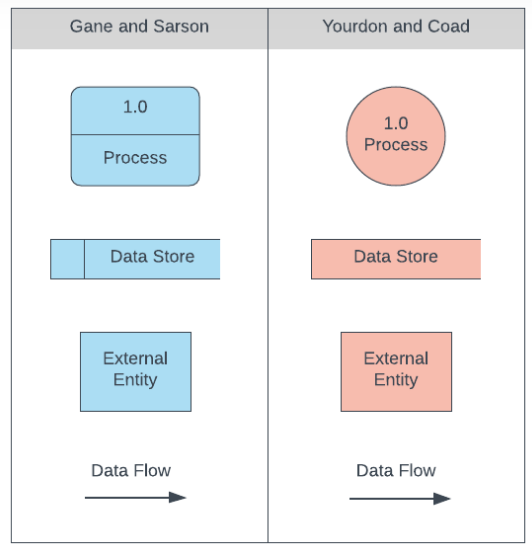
Important:

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

1. Data flow diagram:

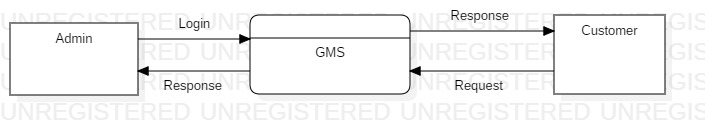
The data flow diagram shows the flow of information through the system and process. The different notation and symbols are used to identify the flow of data. They describe the entity and the relational flow. The flow makes the better understanding of system and improvement further on future development.

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

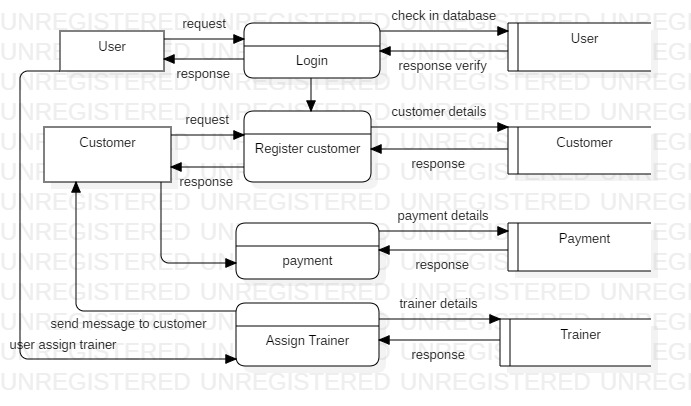


The above diagram in which the different notation has different meanings:

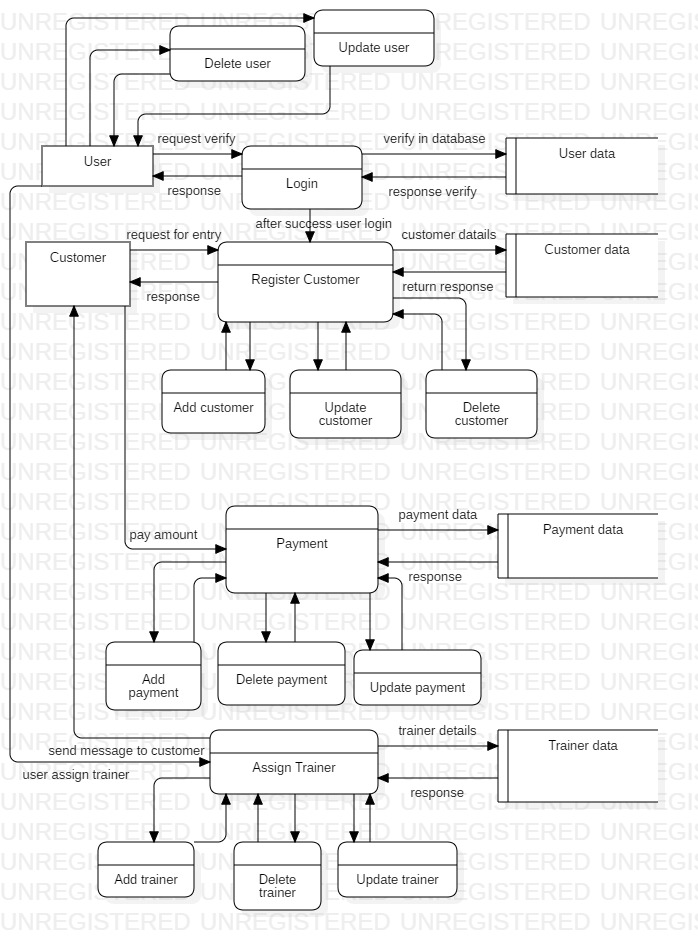
* Process: this known as process in which it known as input and output on DFD. Actually, transport incoming data to outgoing data.
* Data store: this stores the data that has been already processed. Input process are stored in the data and output is retrieved from database.
* External entity: they are 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



screen : DFD level 1



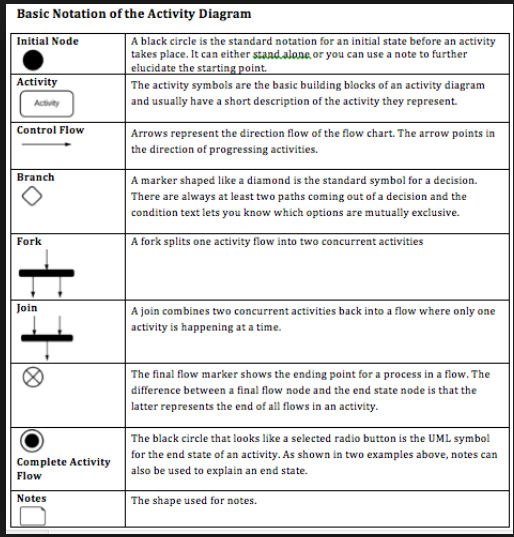
screen : DFD level 2

Behavioral design:

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

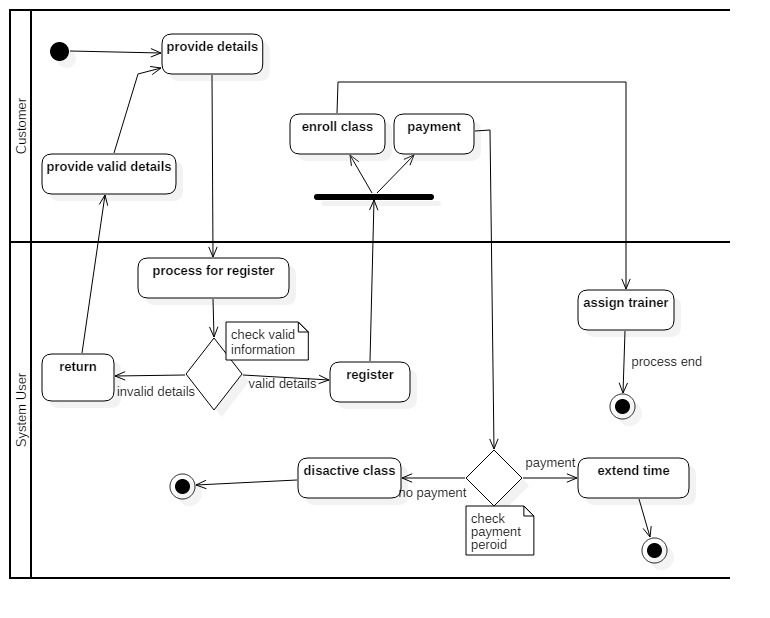
Activity diagram:

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.)

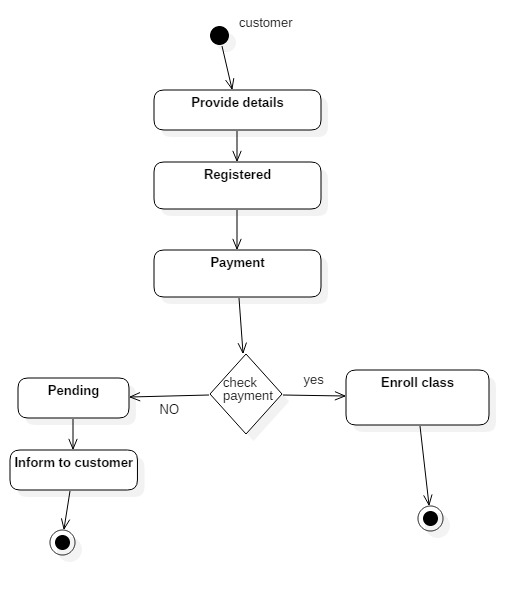


The some of the notation that are sown in above figure. This notation can have meaning with their used. This used combined formed the activity with their meaning. Some of them are described below:

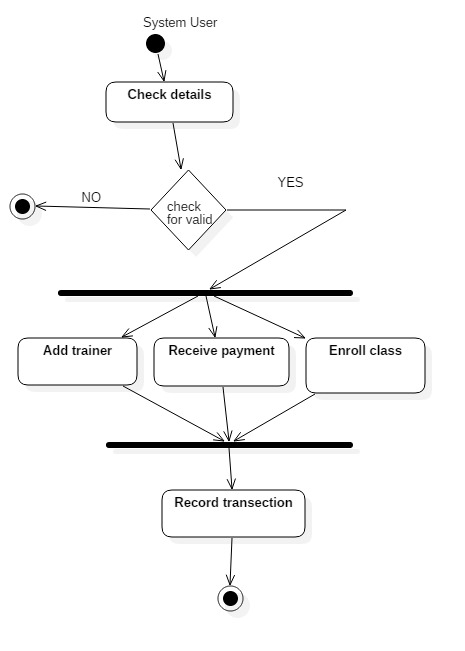
* Initial node: used for starting the system.
* Activity: The activity shows processes in a model process
* Control flow: used to show the flow of data and also preferred as direction flow or control line
* Branch/ decision symbol: identify the decision make during activity
* Fork: helps to split the 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



screen : Activity diagram of customer



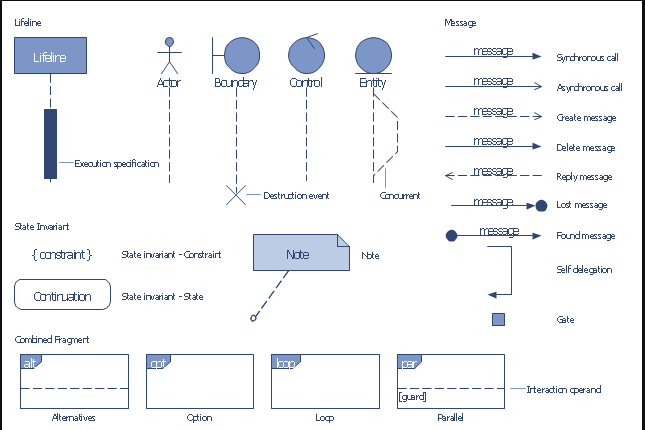
screen : Activity diagram of system user

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

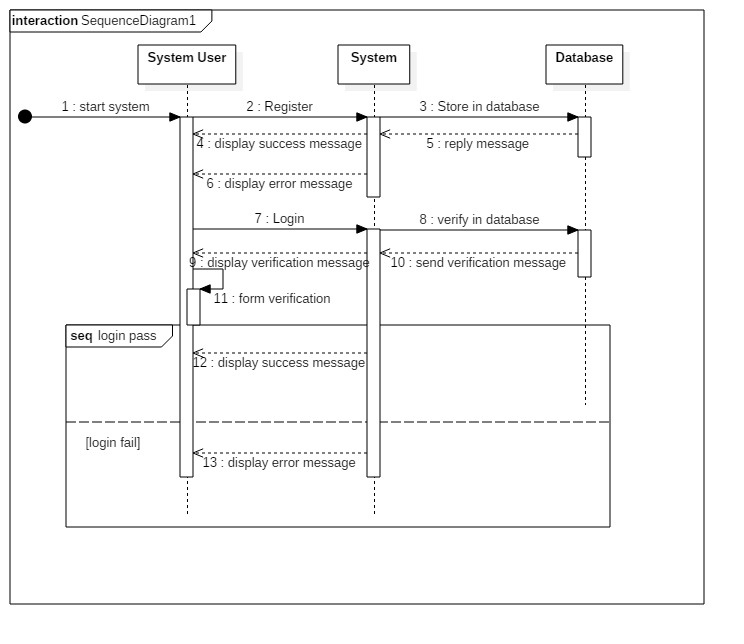
Sequence diagram:

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.

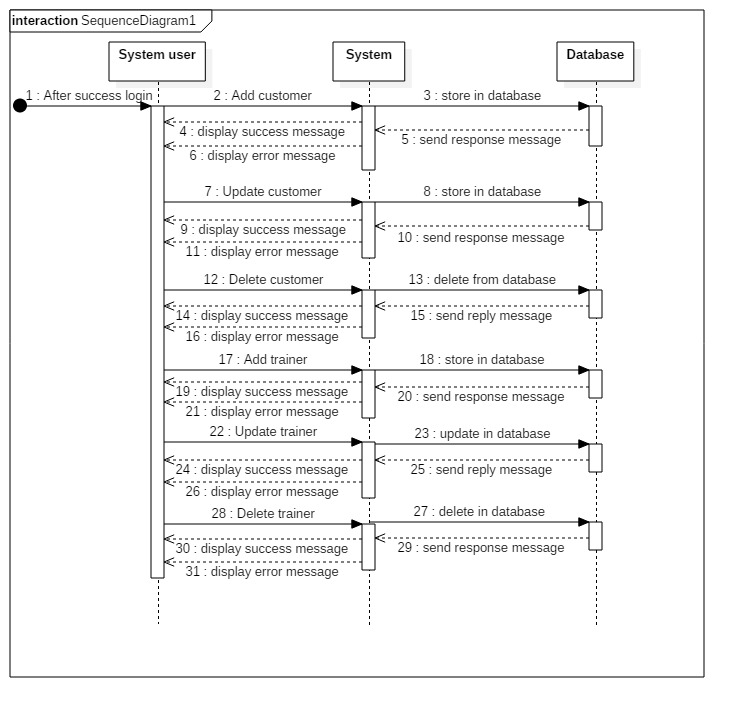


The sequence notation is described below only some of them:

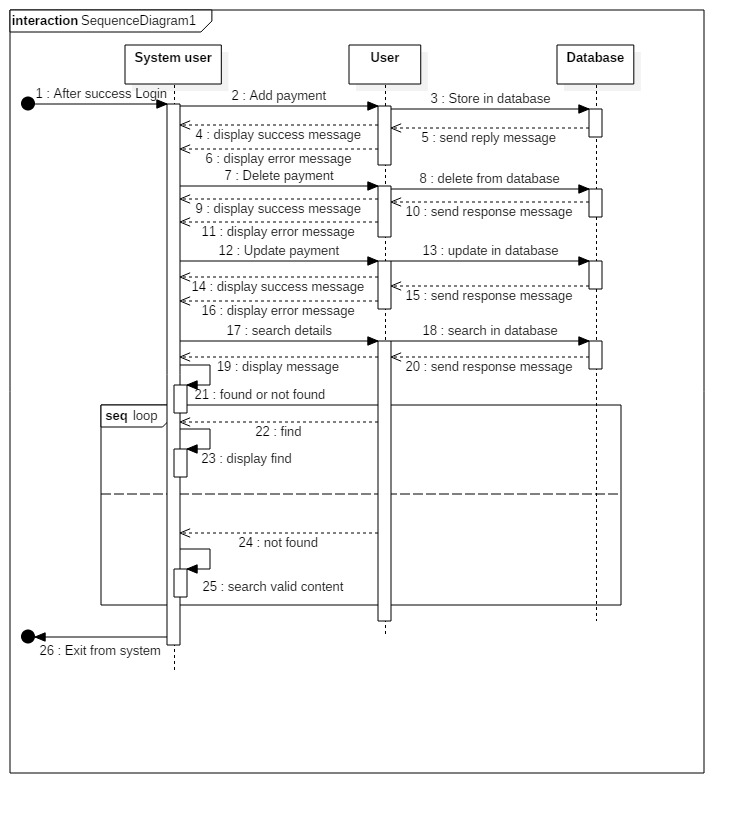
* Object lifeline: indicates that how long the entity life line exist during the whole process of sequence
* Activation: represent the active of the flow of message and the entity that involve on it
* Message: identify the flow of message between the object
* Message synchronous: the kind of message 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



screen : Sequence diagram



screen : Sequence diagram

Important of sequence diagram:

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

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.