# **Chapter 3: Design**

# **Introduction**

Design is the process of defining the software methods, functions objects and overall structure of the system along with the interaction which satisfies the user requirements. It is done after finalizing the requirements specification where we figure out the process to implement the working information system.

# **Structural Design/Model**

Structural design or modelling is based on the mathematical model, computer algorithm and statistical model which fits network of constructs to data. It is used to define the system itself so that the programmers in the implementation phase get a better understanding of the system and implement in the most efficient way. These diagrams reflect the static relationships of structure, like Class or Package diagrams.

## **Final Class Diagram**

Class Diagram is static diagram that represents the static view of an application. Class Diagram is not only used for visualizing, describing and documenting different aspects of a system but also for constructing executable code of the software application.

# **Behavioral Model**

Behavioral Model mean using the available and relevant customer and business spending data to estimate future behavior.

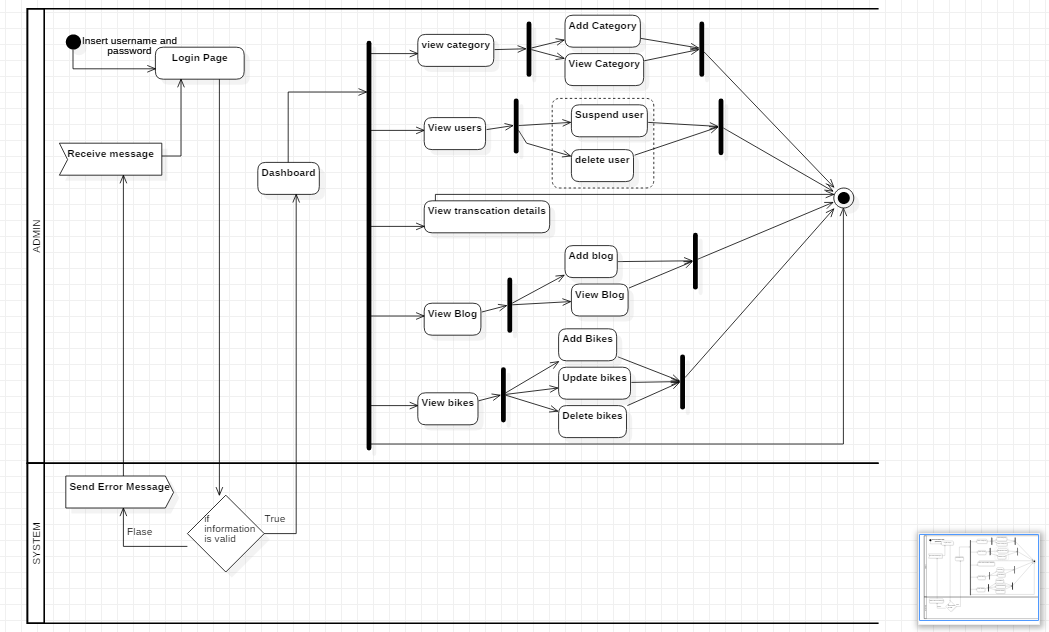
## **Activity Diagram**

Activity diagram is an important diagram which describe the dynamic aspect of the system. It is basically a flowchart to represent the flow from one activity to another activity.

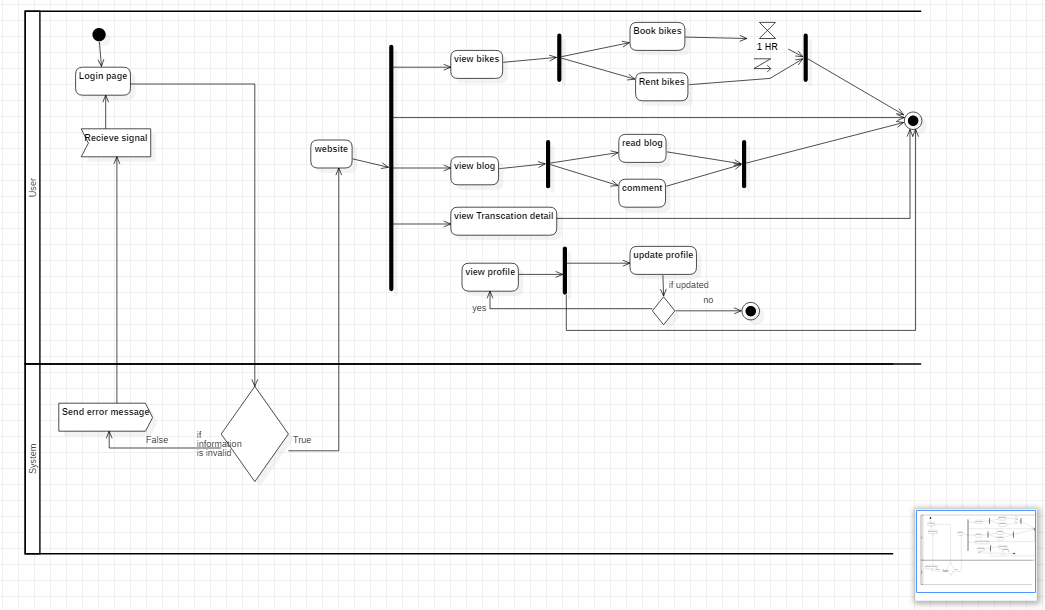
This approach is taken to describe how the work flow in the system. This approach tells us what the users are capable of accomplishing in the system.

|  |  |  |
| --- | --- | --- |
| Notation used | Notation | Description |
| Action |  | Shows what  activity is done |
| Initial |  | Denotes start of the activity |
| Final |  | Denotes end of the activity |
| Fork |  | split a single incoming flow into multiple concurrent flows |
| Join |  | Joins two action into one |
| Decision |  | Decides the condition |
| Control Flow |  | Show the flow of activity |

|  |  |  |
| --- | --- | --- |
| Send Signal |  | Send message through the system |
| Accept Signal |  | Receive the message from send signal |
| Swimlane |  | group related activities into one column |
| Activity Interrupt |  | interrupts the flow denoted with a lightning bolt. |
| Accept Time Event |  | event that stops the flow for a time |
| Interruptible  Activity  Region |  | Activity terminated if interruption occurs |



The following activity diagram show how the activity flow of the admin. Admin login the system providing the username and password which is validated in the system. If true then admin is sent to dashboard else back to login page. In the dashboard there are many options like above figure and the system will process about how the admin panel will work.



The following activity diagram is based on the user where user rents, book a bike as well as how the user’s website will work.

## **Sequence Diagram**

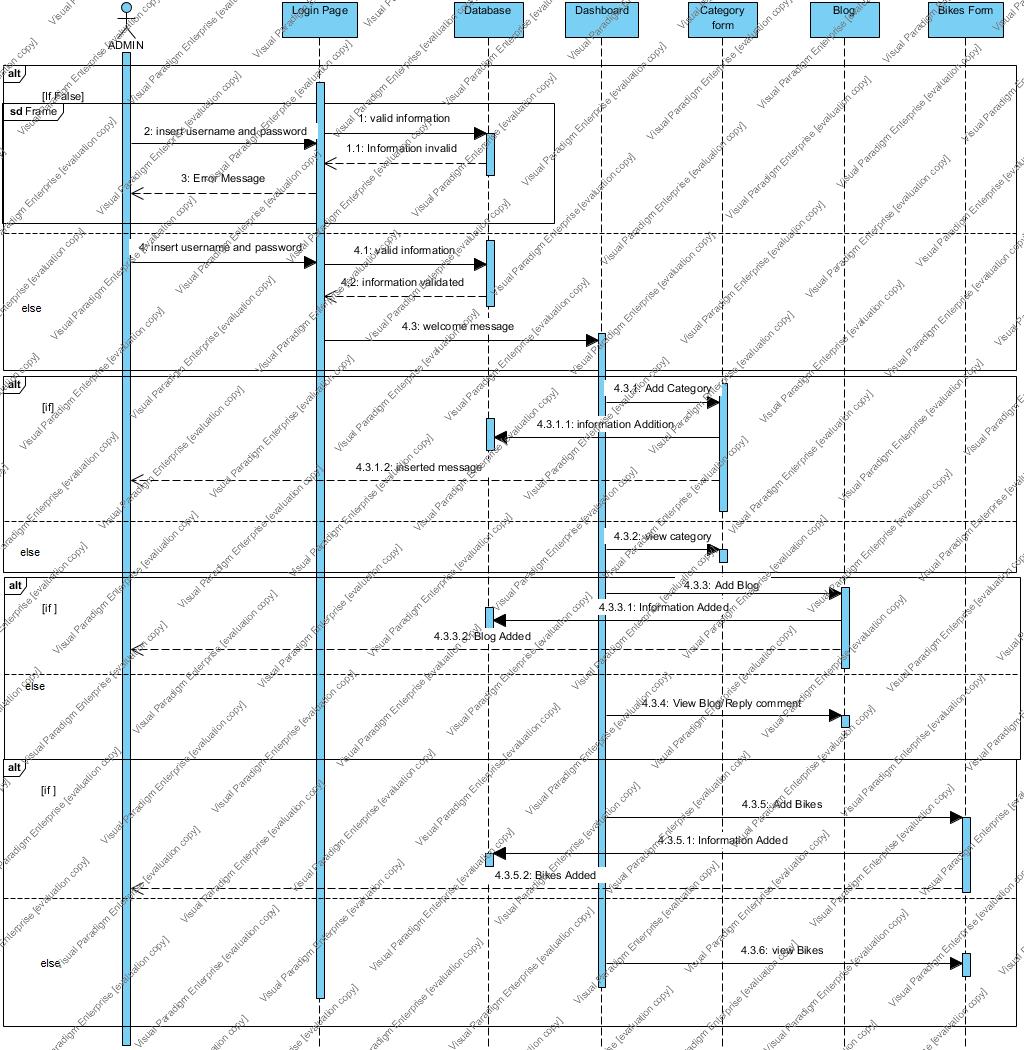
Sequence diagram shows the interactions of the object arranged in time sequence. It is also called event diagrams or event scenarios.

It depicts the objects and classes involved in the scenario and the sequence of messages exchanged between the objects needed to carry out the functionality of the scenario.

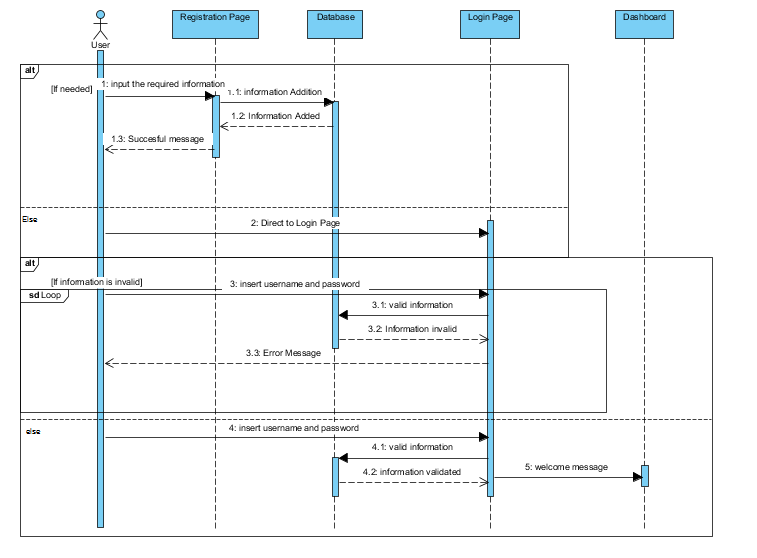
This approach is taken to show at what time what interaction occurs in the system. This allows to know after what process what will occur.

|  |  |  |
| --- | --- | --- |
| Notation Used | Diagram | Description |
| Actor |  | the particular sequence diagram is owned by a use case. |

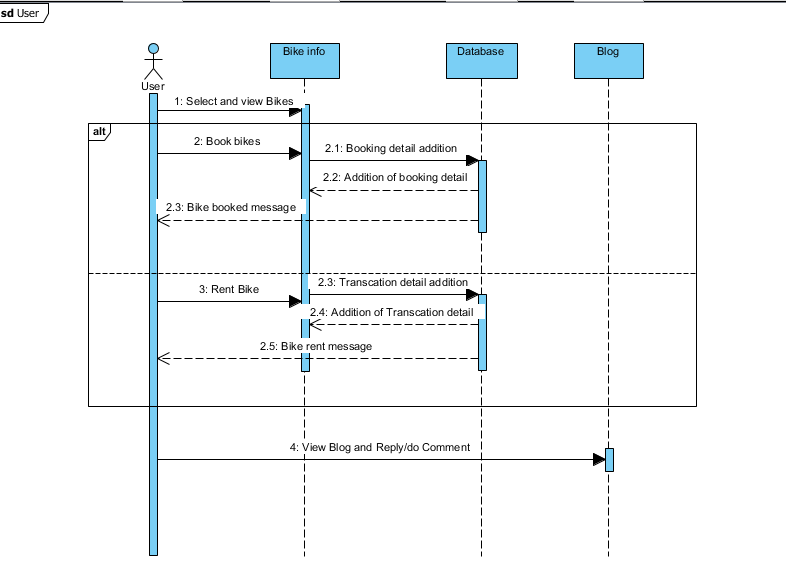
|  |  |  |
| --- | --- | --- |
| Lifeline |  | interact with each other in the system during the sequence. |
| Message Arrow |  | Describes flow of message |
| Alternate frame |  | It models ‘if…else’ logic |



The following sequence show how the admin login and how the admin panel will works in the system.



The following sequence shows how the user will register and login on the system.



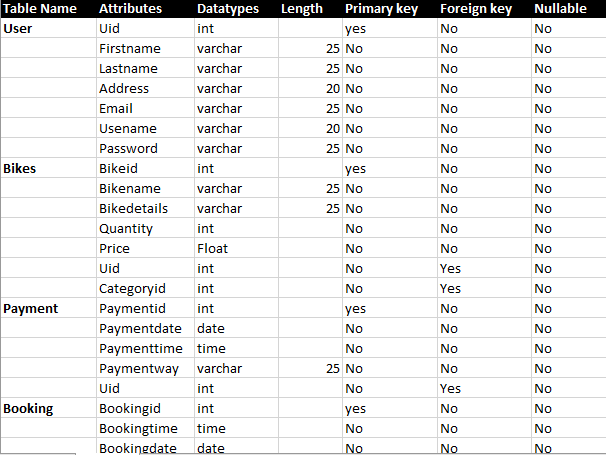
The following sequence shows how the user will rent and book a bikes for their purpose and as well how the user will see the blog.

# **Database Model**

Database Model refers to the logical structure, representation or layout of a database and how the data will be stored, managed and processed within it. The designer determines what data must be stored and how the data elements interrelate. With this information, they can begin to fit the data to the database model. Database design involves classifying data and identifying interrelationships.

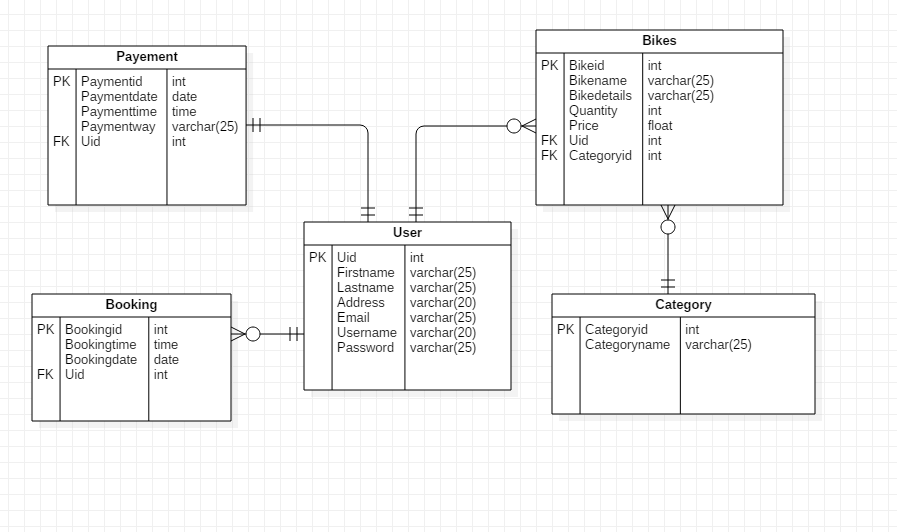
## **Data Dictionary**

Data Dictionary provides metadata for the creation of entities or tables in database. The metadata i.e. data dictionary for the given table is shown below:-



## **ER Diagram**

An entity–relationship model describes interrelated things of interest in a specific domain of knowledge. A basic ER model is composed of entity types and specifies relationships that can exist between entities.



# **Architectural Model**

Architectural model is a scale model that is design to meet the structure and design of the system. My architectural model is based on MVC pattern and it follows 3-tier architecture where the system is divided into:

1. Model
2. View
3. Controller

This approach is taken to describe the structure of the system currently being made.

# **UI Modelling**

User interface modeling is a development technique used by computer application programmers.

## **Prototyping**

It is a graphical representation of how the system would look after it is finished it helps up to understand the system better. The prototype created base line for the UI design.

