**Team :*Jugaad Stack***

***Virtualized Server Tracking***

**Software Design Document**

**Names :**

1. **Deep S Nirmal ( Project Lead and Backend Developer )**
2. **Venkataramana Pai K ( Web master )**

**Date : 13th Feb 2016**

**Table of contents**

1. **Introduction 3**
   1. Purpose
   2. Scope
   3. Overview
   4. Definitions, Keywords and Acronyms
2. **System Overview 5**
3. **System Architecture 6**
   1. Decomposition Description
   2. System Architecture Flowchart
4. **Data Design 9**
   1. Data Description
   2. Data Flow Diagram
5. **Sequence Diagram 11**
6. **State Chart Diagram 12**
7. **Human Interface Design 13** 
   1. Overview of interface
   2. Use cases diagram
8. ***Introduction***
   1. **Purpose**

This software design document describes the architecture and system design of the project Virtualized Server Tracking.

* 1. **Scope**

It is an interface for user/organization and their assets which tells and governs the information and statistics about the usage of Virtual Machines or Infrastructure which they hold. User can easily turn OFF/ON their VM based on the usage. With this the user can save money as the VM won't be running all the time. Our goal is to achieve the easiness for the user/organization having their VM’s to control and access based on their needs and easy payment option with billing system on Real-time basis.

* 1. **Overview**

This project mainly deals with a dynamic website connected to the database. When any user logs in into the website, it will show all the attributes and characteristics from the database of the VM they own. User can switch ON/OFF their VM according to their use and check the cost generated which is based on time elapsed and per hour cost. The functionalities provided in the website is login or signup for new user, user profile which gives brief introduction of the user , VM configuration which the user holds, powering ON and OFF of the virtual machine , start timer and stop timer while powering ON/OFF of the VM respectively, the cost generated which is calculated by the elapsed time and lastly logging out from the session. In addition, the user will be alerted about his usage through SMS and E-mail.

* 1. **Definitions, Keywords and Acronyms**
* VM (Virtual Machine), CPU(Central Processing Unit), RAM(Random Access Memory)
* **Virtual machine** (**VM**) is an emulation of a particular computer system.

**Virtual machines** operate based on the computer architecture and functions of a real or hypothetical computer, and their implementations may involve specialized hardware, software, or a combination of both.

* A server-side **dynamic web** page is a **web**page whose construction is controlled by an application server processing server-side scripts. In server-side scripting, parameters determine how the assembly of every new **web**page proceeds, including the setting up of more client-side processing.
* **Tracking Server**provides the ability to monitor entities that are moving or changing in real time. Data from any real-time data source can be received by tracking server, which serves as a centralized hub for collection and dissemination of real-time data. Real-time processing and analysis can be performed at the point of data reception by applying actions. From Tracking Server, real-time data is sent to Web and desktop client applications

1. **System Overview**

* Website is coded using the language  HTML/CSS and PHP.

-         It will contain all the required parameters that user wants with

respect to the  VM which he/she selects.

* There will be a list of all VMs in the main page that are connected as Guest to that particular remote PC.

-         User selects any one VM and it should display all the respective

Real-Time operations.

-         There will be a Real Time Display and values will change dynamically according to the particular VM.

-         Good animation and designer graphics will be given to each entry to make it look attractive to the user.

* On clicking the VM  name , it will connect to the database which will have all the tables and data entries related to Virtual Machines and its operation.

-         Database will have 9 different tables for VM and its corresponding characteristics.

-         While making a connection to the database , it will search for that particular entry and it relates to other tables to fetch that particular entry for that VM using Primary key and Foreign Key concept.

* Now, the Database entries  will be operated differently using PHP script (this is totally different from main Website)

-         That script will include the code for fetching the VM track.

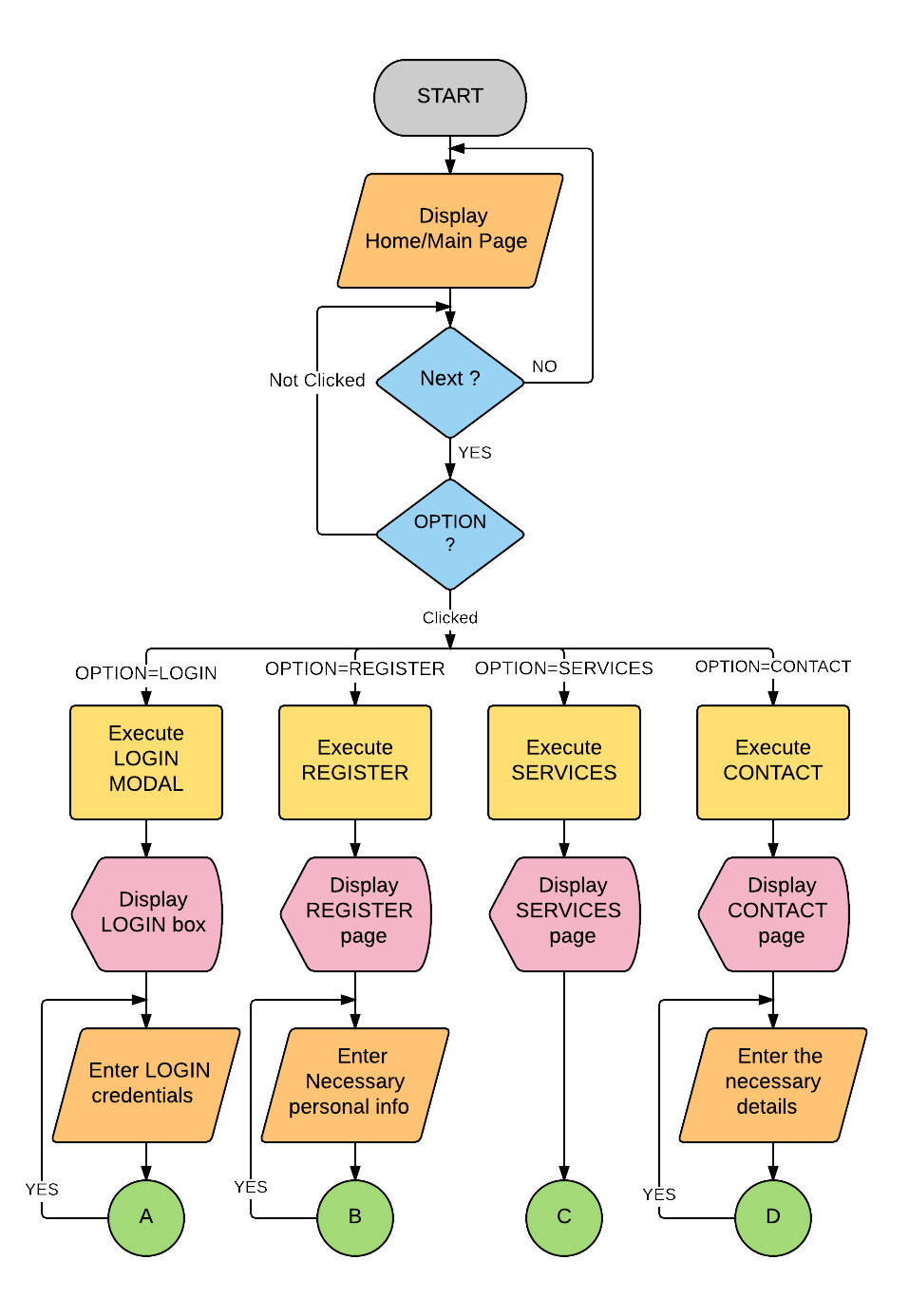
-        While entering into the database the script will be automatically loaded/refreshed and entries will be updated to the database dynamically.

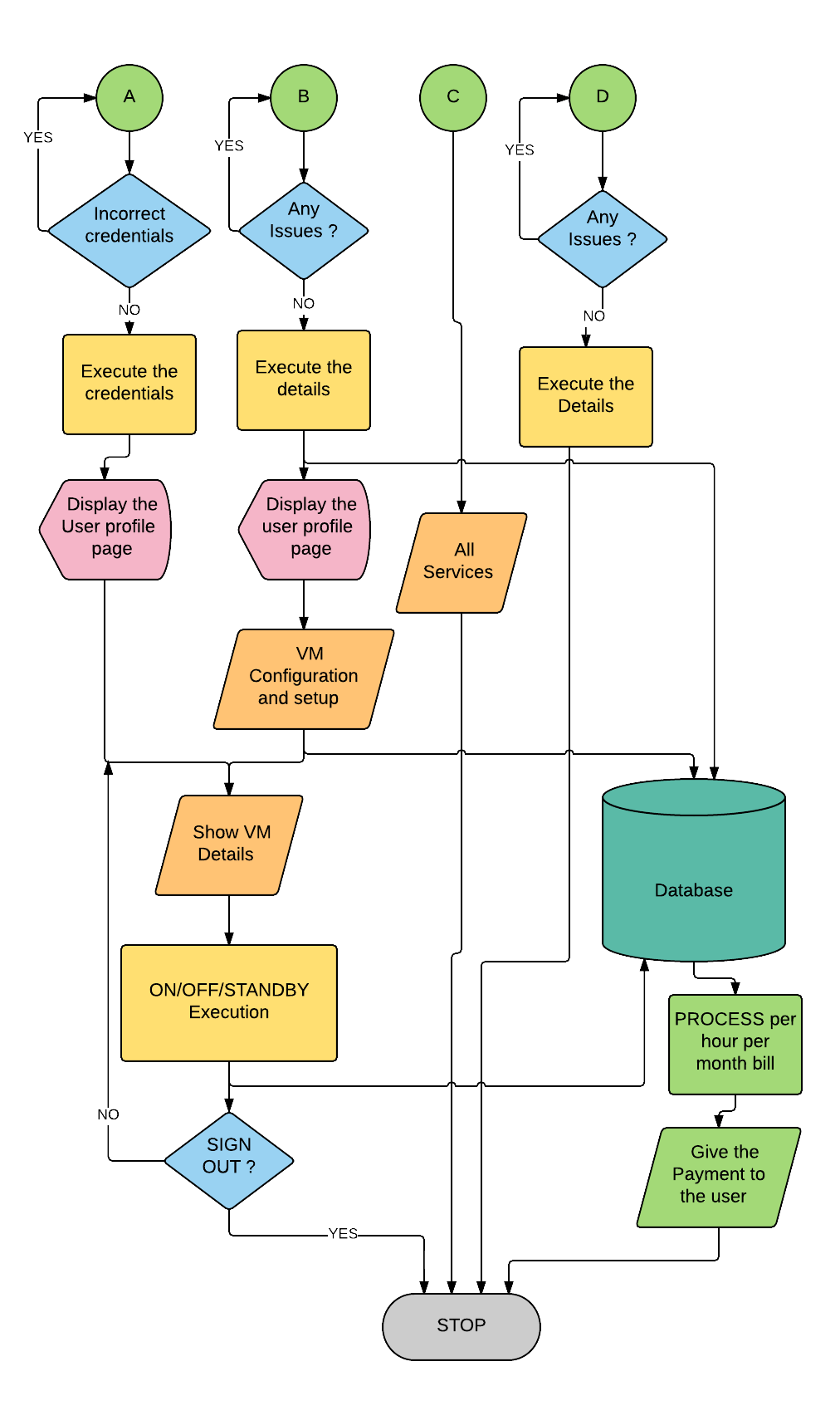
-        The script will return a complete entry for the particular VM.

-        So the complete entry will be sorted and cut in different rows and stored into the respective table of the database.

1. ***System Architect*** 
   1. **Decomposition Description**

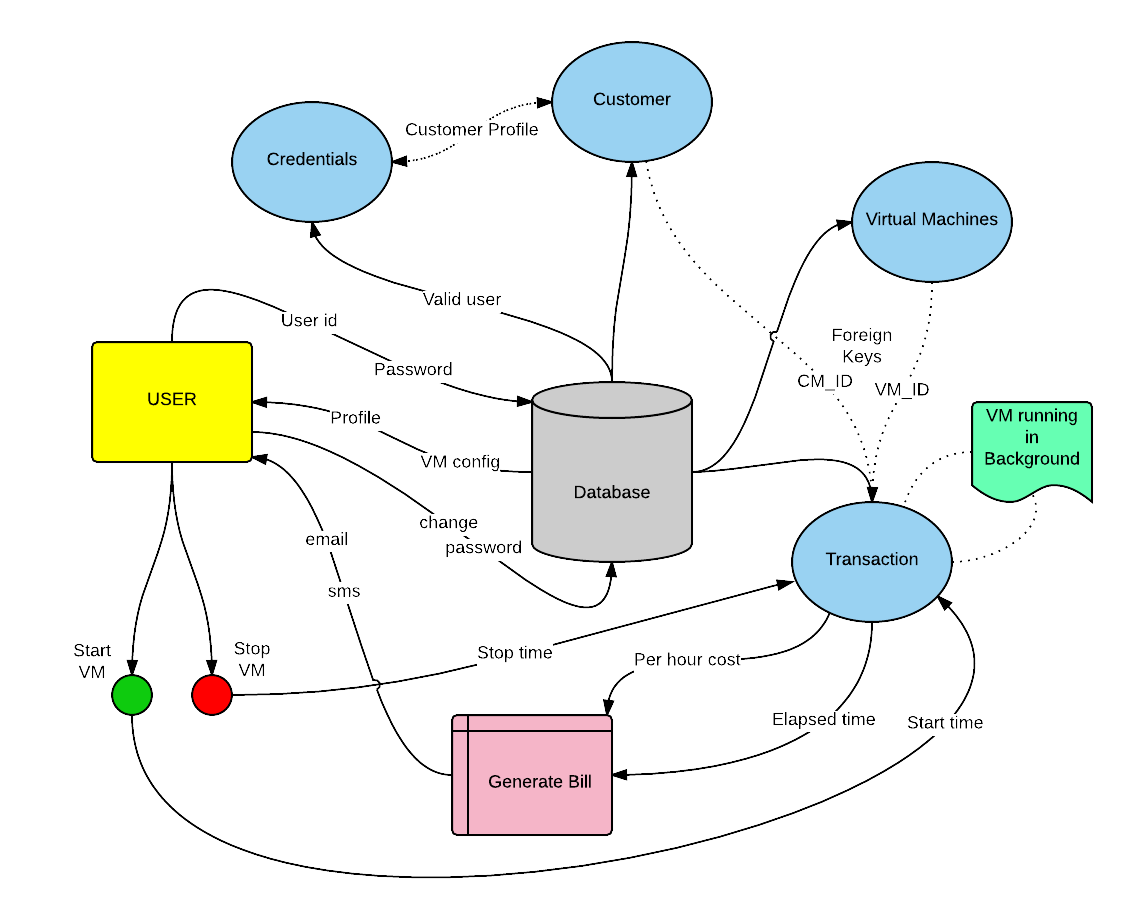
* The system starts by displaying the main page i.e Home page of the website. It consists of all the general information of our software.
* Menu has options as shown in the diagram below, user need to go for any one of the options for accessing it.
* LOGIN option gives the user to login into his/her profile by entering his credentials and get verified as a authorized user. After that user will be given his/her profile to update/modify. Necessary changes (on/off) will be noted and the database will be updated.
* REGISTER option gives the user to join this website for the use of this software for his/her use. Enter the details and it gets added to the database and user gets his profile. VM configuration to be done and changes will be updated to the database.
* SERVICES option gives the user to get aware of all the services given by our software team for efficient usage of this product.
* CONTACT option gives the user to give feedback/suggestions regarding the issues arose. The form submitted will be read by our software team and user will get support to clear his/her issues.
* Finally the database processes the bill as per hour per month usage and generated bill is sent to user using email and SMS.
  1. **System Architecture Diagram**



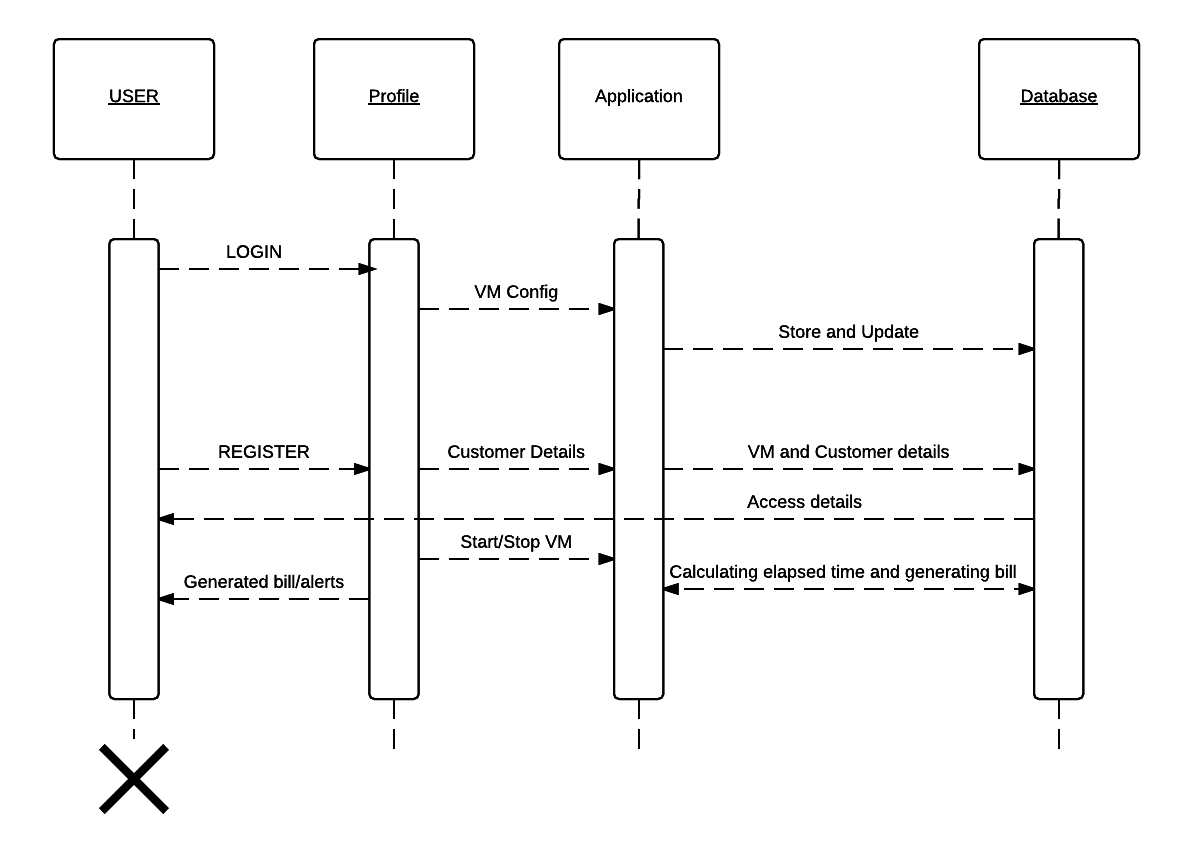


1. ***Data Design*** 
   1. **Data Description**

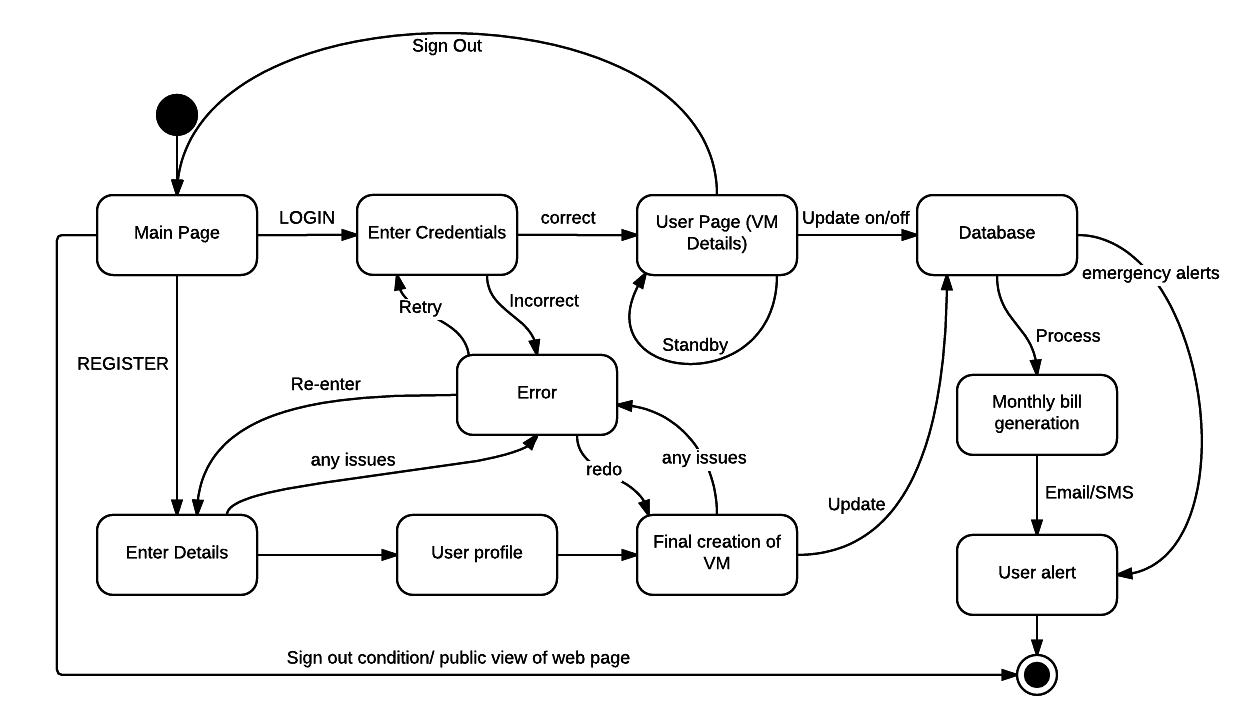
* User sends the credentials to the database and gets the profile page along with the customer details and VM configuration which the user holds.
* User can change the password easily , a new password will be updated in the database.
* There will be two buttons for switching on/off VM given to the user.
* When the user clicks the ON button it goes to another PHP page where it connects to the database.
* It also sets the default timezone to Asia/Kolkata.
* Current time is stored into a variable as startTIME  and showed it to the user as indication for starting the VM.
* It also creates an entry to the database carrying the current user and its associated VM along with its start time and stop will be null for that particular time.
* A shell script will run in background to start the VM.
* There will be a stop button in the same page, so when user clicks on that button , it jumps to another page and connects to the database.
* It will again set the default timezone to Asia/Kolkata.
* An elapsed time will be calculated from the Start time and Stop time and an amount will be generated based on the Cost per hour given to the user.
* Bill will be showed to the user on the next page after stopping the VM
* Generated bill will also be send to the user Email-id and a text message to the mobile number.
  1. **Data Design diagram**



1. ***Sequence Diagram***



1. ***State Chart Diagram***



1. ***Human Interface Design***
   1. **Overview of interface**

Four different users/handlers will be interacting with the website : new user, user, system and admin.

**User** will be having functionalities of login, profile, VM config, start & stop VM, online payment, logging out and billing services.

**New user** will be having register page as its functionality. Then later it will join in the User category with all the functionalities

**Admin** will be the one handling database with adding and removing customers, set VM config, transaction entries and sending bill via mail and SMS.

**System** will be the backend code will be calculating the elapsed time and generating the cost based on the usage of the Virtual machine associated with the User.

* 1. **Use cases diagram**

