Chapter 4
SYSTEM DESIGN

Systems design is the process of defining the architecture, components, modules, interfaces, and data for a system to satisfy specified requirements. Systems design could be seen as the application of systems theory to product development.

4.1 GENERAL

Design is the creation of a plan or convention for the construction of an object or a system (as in architectural blueprints, engineering drawings, business processes, circuit diagrams and sewing patterns). Design has different connotations in different fields. In some cases the direct construction of an object (as in pottery, engineering, management, cowboy coding and graphic design) is also considered to be design.

Designing often necessitates considering the aesthetic, functional, economic and sociopolitical dimensions of both the design object and design process. It may involve considerable research, thought, modeling, interactive adjustment, and re-design. Meanwhile, diverse kinds of objects may be designed, including clothing, graphical user interfaces, skyscrapers, corporate identities, business processes and even methods of designing.[4]

The automated online recharge system consists of two main parts that are mentioned below:

a. Login credentials
b. Setting the threshold

Automated Online Recharge Portal is a webpage designed using Hypertext Markup Language and Cascading Style Sheets. Bootstrap is used to make the page interactive. Laravel – The PHP Framework is used for the server side programming. It is uploaded online using a web hosting site. Online recharge portal information page consists of the login and sign up page. For sign up, user needs to provide his/her Name, Mobile Number, Email id, Password. For login, user needs to provide his/her phone number and password. The user needs to connect his/her digital wallet with bank account using UPI(Unified Payment Interface) providing their registered mobile number. User also needs to manually set his/her threshold for mobile recharge. Threshold can be changed at any time by logging in to their account.[6]

Threshold is of 2 types:

- Volume threshold
- Time threshold

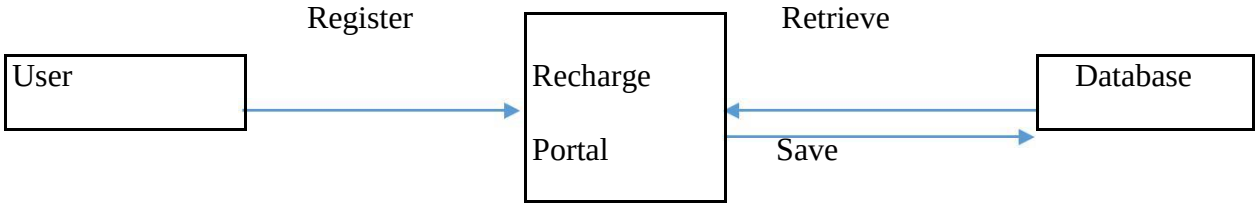

Fig 4.1 General Diagram

4.2 SYSTEM ARCHITECTURE

A system architecture or systems architecture is the conceptual model that defines the structure, behavior, and more views of a system. An architecture description is a formal description and representation of a system, organized in a way that supports reasoning about the structures and behaviors of the system.

A system architecture can comprise system components, the externally visible properties of those components, the relationships (e.g. the behavior) between them. It can provide a plan from which products can be procured, and systems developed, that will work together to implement the overall system. There have been efforts to formalize languages to describe system architecture, collectively these are called architecture description languages (ADLs).

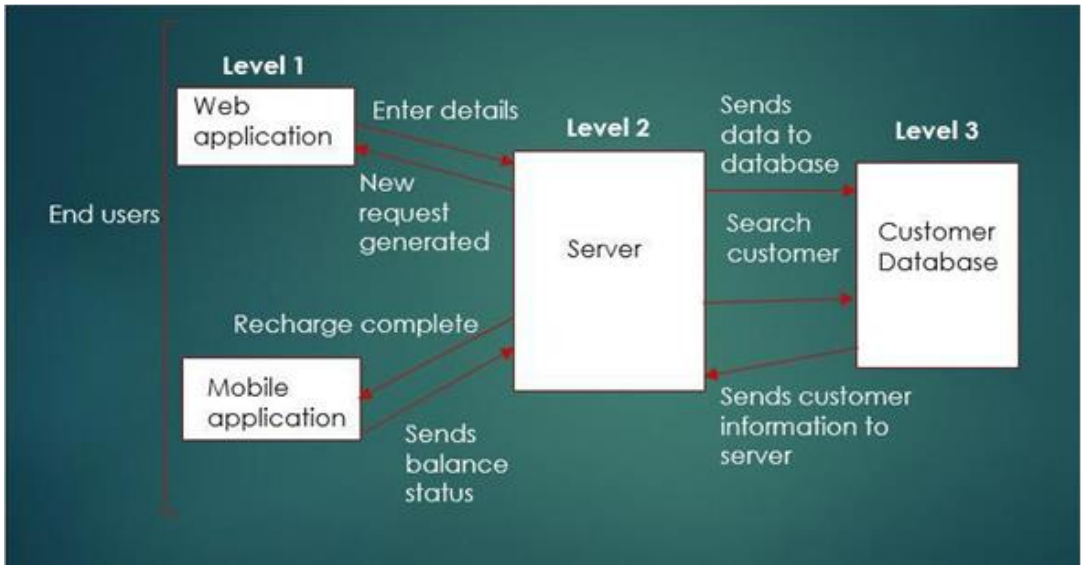

Fig 4.2 SYSTEM ARCHITECTURE

The automated online recharge system architecture consists of three levels:

a. Level 1 consists of the end user part

b. Level 2 consists of the application server part.

c. Level 3 consists of the database server part.

a.. End user part :

Consists of the web application part where the users enter their information and registers on the online recharge portal.

b. Application server part :

This part consists of the application server which checks the information entered by user. It notifies the user of any duplicated values being entered in the database.

c. Database server part :

This part consists of the server which consists of the database that contains the user information such as his/her threshold limit.

4.3 DATA FLOW DIAGRAMS

A data flow diagram (DFD) is a graphical representation of the "flow" of data through an information system, modelling its *process* aspects. A DFD is often used as a preliminary step to create an overview of the system, which can later be elaborated. DFDs can also be used for the visualization of data processing (structured design).[2]

A DFD shows what kind of information will be input to and output from the system, where the data will come from and go to, and where the data will be stored. It does not show information about the timing of process or information about whether processes will operate in sequence or in parallel.

I. LEVEL 1

This level represents the basic flow of data connecting the user, recharge system, server and customer database.

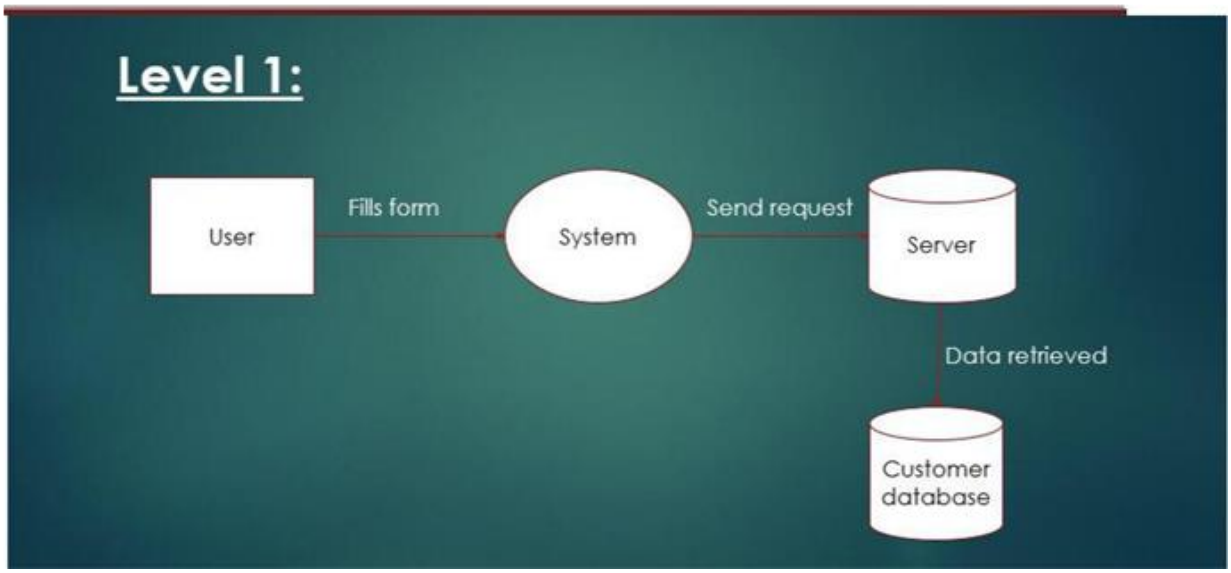

Fig 4.3 DFD LEVEL 1 DIAGRAM

II. LEVEL 2

This provides a detailed level consisting of registration and login pages.

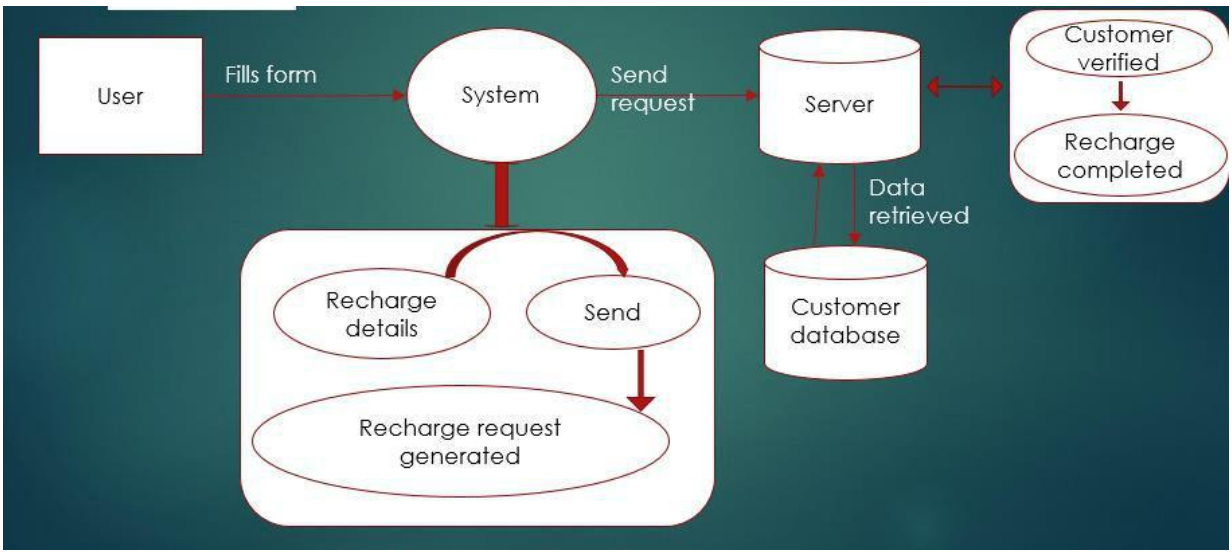

Fig 4.4 DFD LEVEL 2 DIAGRAM

III. LEVEL 3

Detailed structure of the automated online recharge system is given in level 3.

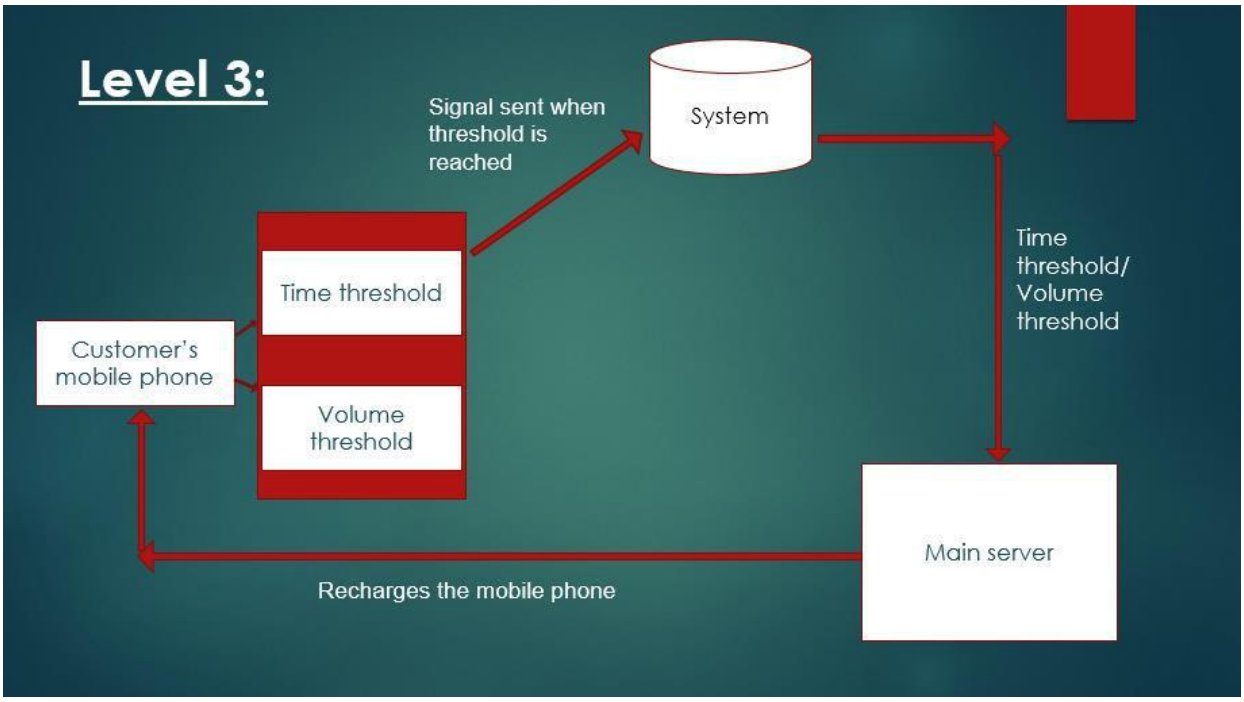

Fig 4.5 DFD LEVEL 3 DIAGRAM

4.4 UML DIAGRAMS

The Unified Modeling Language (UML) is a general-purpose, developmental, modeling language in the field of software engineering that is intended to provide a standard way to visualize the design of a system.

UML was originally motivated by the desire to standardize the disparate notational systems and approaches to software design developed by Grady Booch, Ivar Jacobson and James Rumbaugh at Rational Software in 1994–95, with further development led by them through 1996.

In 1997 UML was adopted as a standard by the Object Management Group (OMG), and has been managed by this organization ever since. In 2005 the Unified Modeling Language was

also published by the International Organization for Standardization (ISO) as an approved ISO standard. Since then it has been periodically revised to cover the latest revision of UML.

Though well-known and widely used in education and academic papers, as of 2013 UML is little-used in industry, and most such use is informal and ad hoc.

4.4.1 USE CASE DIAGRAMS

A use case diagram at its simplest is a representation of a user's interaction with the system that shows the relationship between the user and the different use cases in which the user is involved. A use case diagram can identify the different types of users of a system and the different use cases and will often be accompanied by other types of diagrams as well.

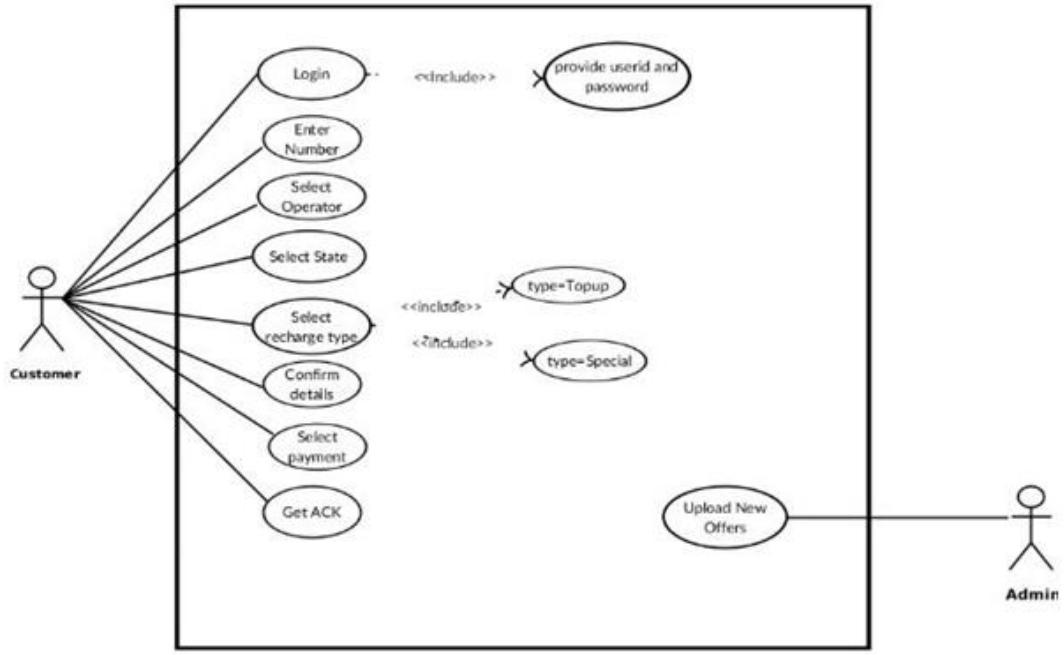

Fig 4.4.1 GENERAL USE CASE DIAGRAM FOR ONLINE RECHARGE SYSTEM

The online recharge system consists of two actors:

I. User

II. Admin

User enters details in the registration form.

The admin controls the online recharge system.

The given system represented inside the rectangular box as online recharge system.
The use cases are as follows:

1. Login
2. Enter number
3. Select operator
4. Select state
5. Select recharge type
6. Confirm details
7. Select payment
8. Get ACK

Registration contains entering details of user's details.
The admin can upload new offers periodically.

4.4.2 SEQUENCE DIAGRAM

A Sequence diagram is an interaction diagram that shows how processes operate with one another and in what order. It is a construct of a Message Sequence Chart. A sequence diagram shows object interactions arranged in time sequence. 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. Sequence diagrams are typically associated with use case realizations in the Logical View of the system under development. Sequence diagrams are sometimes called event diagrams or event scenarios.[2]

A sequence diagram shows, as parallel vertical lines (*lifelines*), different processes or objects that live simultaneously, and, as horizontal arrows, the messages exchanged between them, in the order in which they occur. This allows the specification of simple runtime scenarios in a graphical manner.

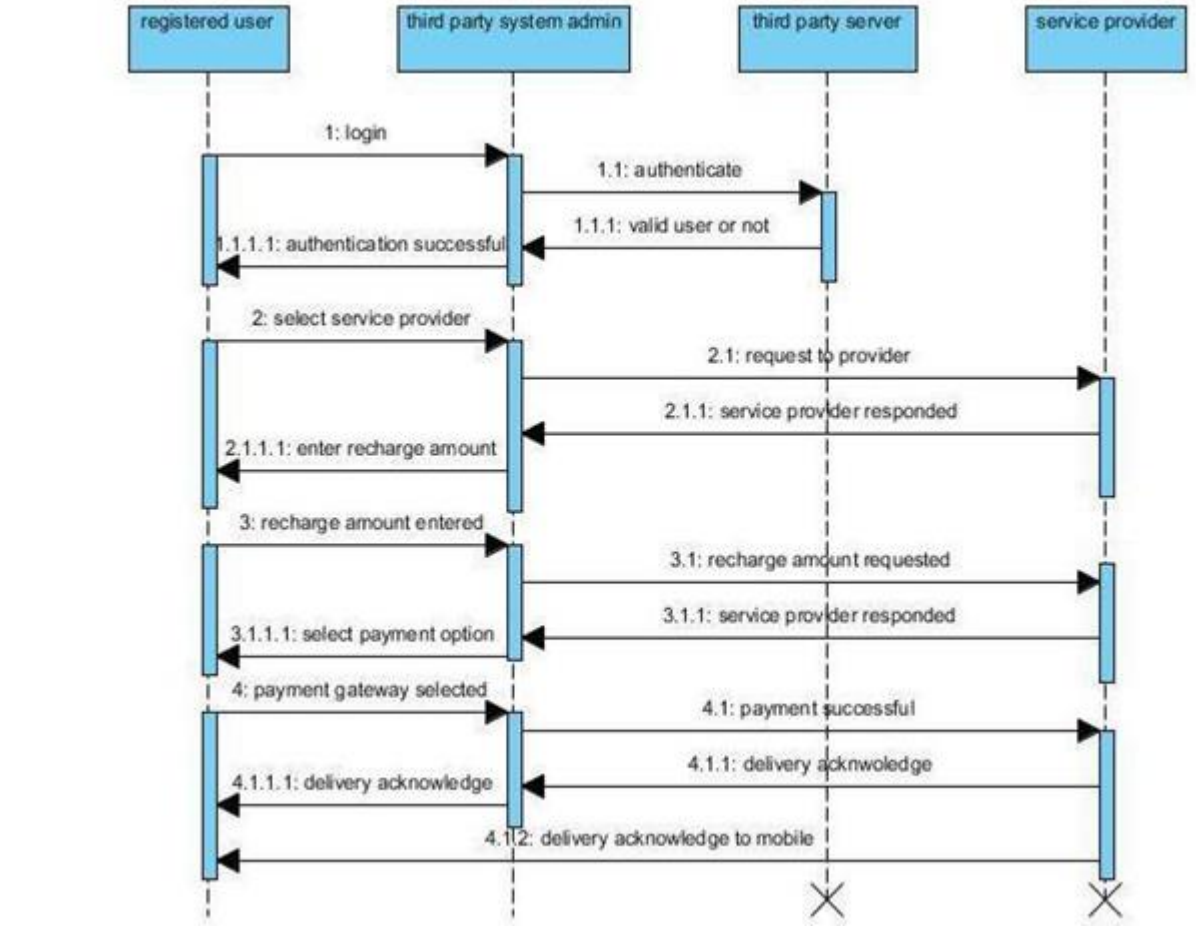

Fig 4.4.2 SEQUENCE DIAGRAM FOR ONLINE RECHARGE SYSTEM

Sequence diagram consists of four objects:

I. Registered User
II. System admin
III. Server
IV. Service Provider

4.4.3 ACTIVITY DIAGRAMS

Activity diagrams are graphical representations of workflows of stepwise activities and actions with support for choice, iteration and concurrency. In the Unified Modeling Language, activity diagrams are intended to model both computational and organizational processes (i.e. workflows). Activity diagrams show the overall flow of control.

Activity diagrams are constructed from a limited number of shapes, connected with arrows. The most important shape types:

- *rounded rectangles* represent *actions*;
- *diamonds* represent *decisions*;
- *bars* represent the start (*split*) or end (*join*) of concurrent activities;
- a *black circle* represents the start (*initial state*) of the workflow; An *encircled black circle* represents the end (*final state*).

Arrows run from the start towards the end and represent the order in which activities happen.

Activity diagrams may be regarded as a form of flowchart. Typical flowchart techniques lack constructs for expressing concurrency. However, the join and split symbols in activity diagrams only resolve this for simple cases; the meaning of the model is not clear when they are arbitrarily combined with decisions or loops.

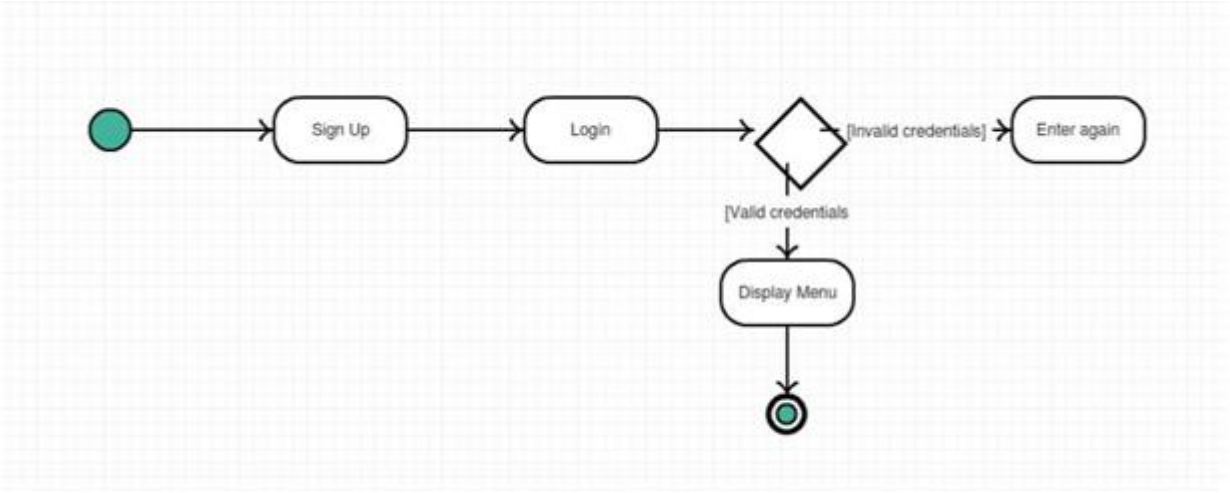

Fig 4.4.3 ACTIVITY DIAGRAM FOR ONLINE RECHARGE SYSTEM

SUMMARY

This chapter shows the general design and system architecture of the automated online recharge system. There are also the UML diagrams that include use case diagram to depict actors and use cases, sequence diagram to show the life line of various activities and finally the activity diagram that shows the stepwise flow of activities that take place during the recharge process.