GUJARAT TECHNOLOGICAL UNIVERSITY
Chandkheda, Ahmedabad
Affiliated

Birla Vishvakarma Mahavidyalaya
Eng. College, V.V. Nagar

A
Project Report
On

E-Paper

Under subject of
DESIGN ENGINEERING – II - B
B.E. III, Semester – VI
IT Branch

Submitted by:
Group:

Sr.	Name of Student	Enrollment No.
1	Dhwani Bhansali	130080116001
2	Bhavik Bhavsar	130080116002
3	Jaydeep Gabani	130080116013
4	Hit Kalariya	130080116016

Prof. Vishal A. Polara
Faculty Guide

Prof. Zankhana H. Shah
Head of Department

Academic Year
(2015-16)

Birla Vishvakarma Mahavidyalaya

Information Technology Department

CERTIFICATE

This is to certify that project entitled with **“E-Paper”**has been carried out by **Dhwani Bhansali, Bhavik Bhavsar, Jaydip Gabani** and **Hit Kalariya** under my guidance in fulfillment of 6th semester subject Design Engineering II-B of Bachelor of Engineering in Information Technology Department during the academic year 2015-16.

Date:

Place:

Project Guide: Prof. Vishal A. Polara

ABSTRACT

As we know the use of paper affect directly to the environment as paper is made from trees. E-paper is an approach to make these activities less cumbersome. Through this project we intend to make most of the places that need to reduce use of paper like in Shopping mall, Tickets, Bills etc. By this project people won't have to keep the tickets or bills, they'll be able to keep the records of their bills, tickets in their mobile phones or in E-mails.

People are always confounded about loss of the tickets and also often found helpless as they'll have to go to ticket booth and have to pay again for the ticket, so for that portal also provides to send tickets to their mails or mobile.

As in shopping mall for only one item we purchase they give us long paper containing all fields. Instead of that this system does message on customer's mobile phones and for confirmation, system gives OTP and ticket checker will match the code. One of the major features is that on the server all the data stored and can be retrieve whenever we want.

So the waste of paper is solved by this project. All these features of the portal make the environment safe by not cutting the trees for paper.

ACKNOWLEDGEMENT

We are using this opportunity to express our gratitude to everyone who supported us throughout the course of this Design Engineering project. We are thankful for their aspiring guidance, invaluably constructive criticism and friendly advice during the project work. We are sincerely grateful to them for sharing their truthful and illuminating views on a number of issues related to the project. We would like to thank our head of the department Prof. Zankhna Shah.

We would also like to thank our project guide Prof. Vishal Polara and all the people who provided us with the facilities being required and conductive conditions for our Design Engineering project.

TABLE OF CONTENTS

TITLE PAGE **I**

CERTIFICATE **II**

ABSTRACT **III**

ACKNOWLEDGEMENT **IV**

TABLE OF CONTENTS **V**

LIST OF FIGURES **VI**

LIST OF TABLES **VII**

Chapter 1 Introduction to PD canvas **1**

Chapter 2 AEIOU Framework for Observation **3**

2.1 Activity Record Sheet 3

2.2 Environment Record Sheet 6

2.3 Interaction Record Sheet 7

2.4 Object Record Sheet 8

2.5 User Record Sheet 9

Chapter 3 Review Report of Prior Art Search **10**

3.1 Factsheet- saving papers in schools 10

3.2 Development of online bus-ticket reservation system 11

3.3 Review of E-PAPER 12

Chapter 4 Learning Need Matrix **13**

Chapter 5 Snapshot of First Prototype **15**

Chapter 6 Data Dictionary **17**

Chapter 7 Diagrams **20**

7.1 Usecase Diagram 20

7.2 Sequence Diagram 25

7.3 Activity Diagram 32

7.4 E-R Diagram 43

7.5 DFD 46

Chapter 8 Implementation **51**

8.1 Home page 51

8.2 Service page 51

8.3 Service page 52

8.4 E-Mail 52

8.5 Database tables 53

Conclusion and Future Work **54**

LIST OF FIGURES

Sr. no.	Description	Page no.
1	Image of people standing in a queue	3
2	Image of People buying stuff	3
3	Image of Person making bill.	4
4	Image of Misplaced ticket at railway station	4
5	Image of Verification of tickets	4
6	Image of Activities Canvas(AEIOU FRAMEWORK)	5
7	Image of Environment Canvas (AEIOU framework)	6
8	Image of Interactions Canvas (AEIOU framework)	7
9	Image of Objects Canvas	8
10	Image of Users Canvas (AEIOU framework)	9
11	Image of Learning needs matrix Canvas	14
12	Snapshot of Billing Prototype	15
13	Snapshot of Tollbooth receipt Prototype	15
14	Snapshot of Ticket Prototype	16
15	Use Case Diagram for ticketing	23
16	Use Case Diagram for billing	24
17	Sequence diagram for ticketing	30
18	Sequence diagram for billing	31
19	Activity diagram for billing	38
20	Activity diagram for Ticketing	39
21	Activity diagram for message sending	40
22	Activity diagram for Payment	41
23	Activity diagram for code matching	42
24	E-R diagram	45
25	Level 0 Dataflow diagram	47
26	Level 1 Dataflow diagram	48
27	Level 2 Dataflow diagram	49
28	Level 2 Dataflow diagram	49
29	Level 2 Dataflow diagram	50
30	Home page	51

31	Service page	51
32	Service page	52
33	E-Mail	52
34	Table 1	53
35	Table 2	53
36	Table 3	53

LIST OF TABLES

Sr. no.	Description	Page no.
1	Components	2
2	Reject, Redesign and Retain	2
3	Elements, features and special notes of Activity Canvas	3
4	General impressions / Observations	7
5	Learning needs matrix	13
6	Customer dictionary table	17
7	Bill dictionary table	17
8	Toll-tax data dictionary table	18
9	Ticket dictionary table	18
10	Item dictionary table	19
11	Use Case diagram symbols table	20
12	Sequence diagram symbols table	25
13	Activity diagram symbols table	32
14	E-R diagram symbols table	43
15	Data flow diagram symbols table	46

Chapter 1: Introduction to PD canvas

The product development canvas mainly consists of 8 parts:

1. Purpose
2. People
3. Product experience
4. Product functions
5. Product features
6. Components
7. Customer revalidation
8. Reject, redesign and retain

1. Purpose:

- The main purpose of our project is to save paper by reducing use of it and make data persistent for longer period of time and increase customer satisfaction.

2. People:

- Students
- Housewives
- Travelers
- Conductor
- Cashier
- Senior citizens
- Differently able persons

3. Product Experience:

- This project (website) will be easy to use.
- Data can be stored and retrieved easily as all the things are stored on single server.
- No fear of losing tickets/bills.

4. Product Functions:

- Calculation- calculate total price and travelling charge.
- Check- can see the bill and tickets or receipt any time.
- Compatibilities: - useful in shopping mall, shops, toll booths, bus/railway stations and air ports, etc.
- Notify- Give detail bill/ticket by text message and email.
- Generate- used to generate new bill/ticket.
- Verify- To verify whether the bill/ticket is authentic or not.

5. Product Features:

- Multiple-language support
- Easy to keep record.

- No worries of losing ticket/receipt/bills.
- No need of paper/printer.

6. Components:

COMPONENT NAME	USE
Computers	To make bills.
Mobile	To receive bills/tickets/receipts along with code via text message/email.
Messaging Software	To Send bills/tickets/receipts
Code generator	To generate code
Code matcher	To match the generated code
Server and database	To Keep records
Internet	To run messaging Software online
Mobile Network	To get texts

Table – 1: Components

7. Customer Revalidation:

- By talking to other people about this product, we came to know that most of the people are satisfied with the E – PAPER as it saves a lot of paper and also it is more efficient and reliable.

8. Reject, Redesign and Retain

Reject	Redesign	Retain
Barcode	Instead of barcode a simple code can be used	We have kept calculations, view, verify as it was.
	Email notification is added with text messages	

Table – 2: Reject, Redesign and Retain

Chapter 2: AEIOU Framework for Observation

2.1 Activity Record Sheet

❖ General impressions / Observations

- People standing in queue
- People buying stuff
- Employees making bills
- Person misplaced his ticket
- Civilians paying toll – tax
- Checking of paid bills

❖ Elements, features and special notes

➢ Messaging Software	Sends bill/ticket	Easy to save message
➢ Code Matcher	Do verification	Reduce human effort

Table – 3: Elements, features and special notes

Figure – 1: Image of people standing in a queue

Figure – 2: People buying stuff

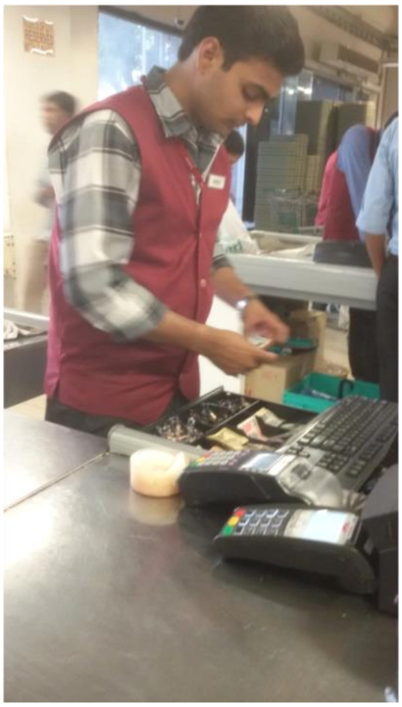
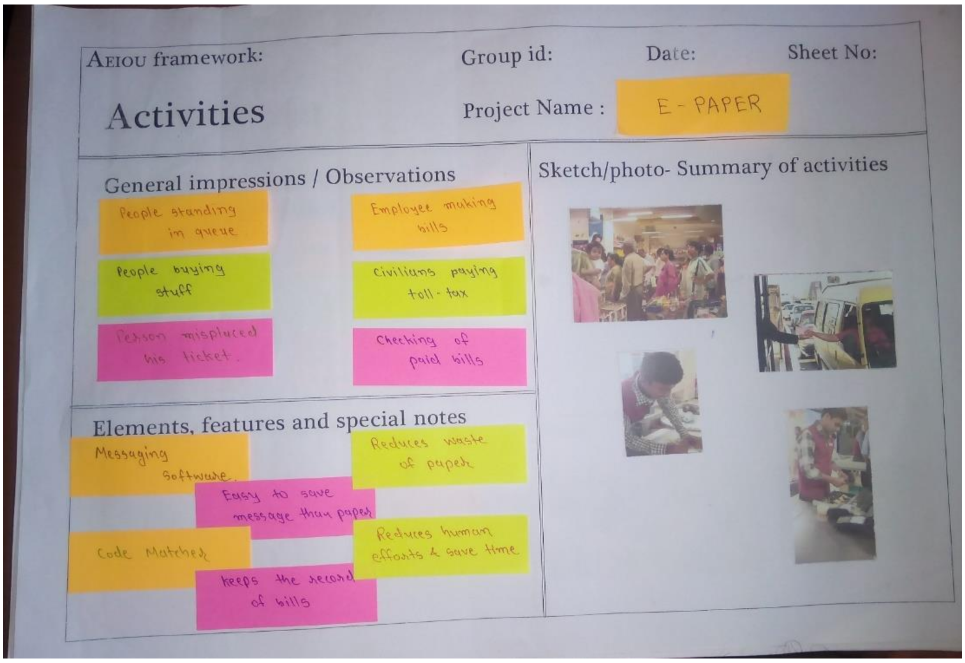


Figure – 3: Person making bill.

Figure – 4: Misplaced ticket at railway station

Figure – 5: Verification of tickets



AEIOU framework: Group id: Date: Sheet No:

Activities Project Name : E - PAPER

General impressions / Observations

People standing in queue

Employee making bills

People buying stuff

Civilians paying toll - tax

Person misplaced his ticket.

Checking of paid bills

Elements, features and special notes

Messaging Software.

Reduces waste of paper

Easy to save message than paper

Code Matcher

Reduces human efforts & save time

keeps the record of bills

Sketch/photo- Summary of activities

Figure – 6: Image of Activities Canvas

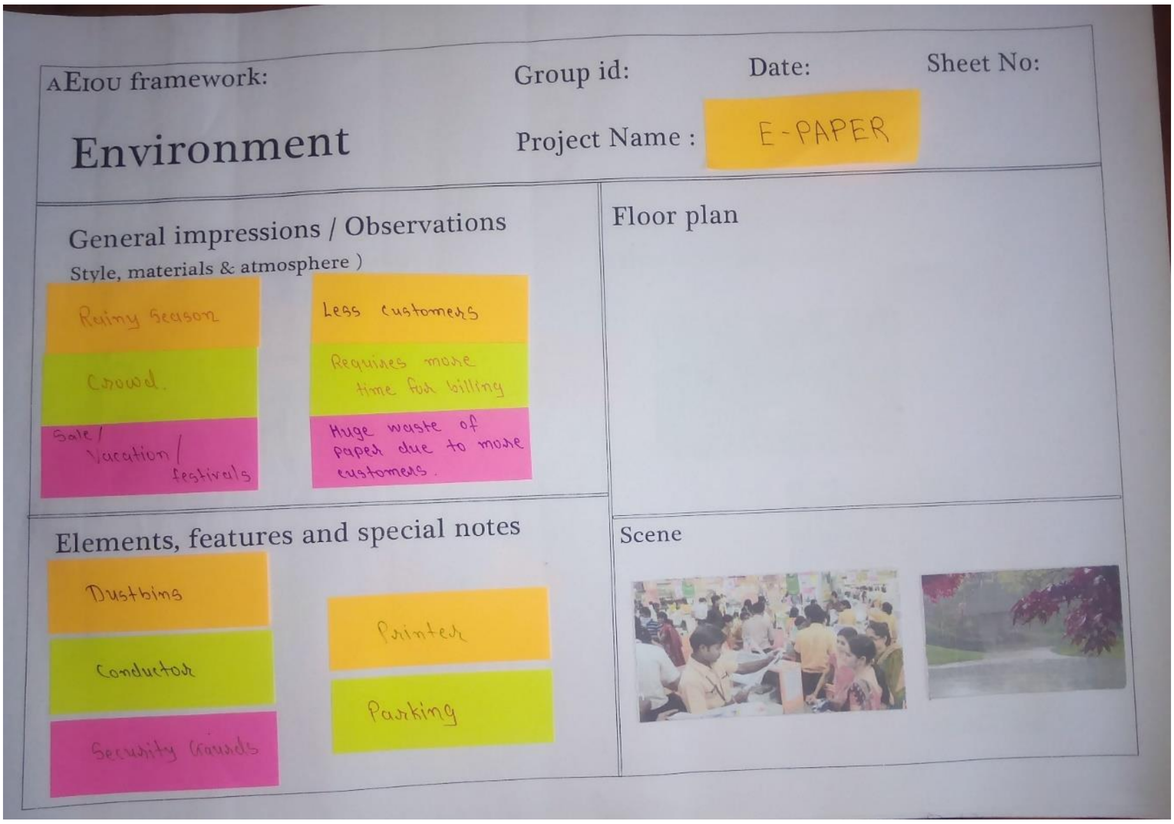
2.2Environment Record Sheet

❖ General impressions / Observations

- Rainy
- Crowd
- Less customers
- Sale/vacation/festival
- Huge to more customers
- Requires more time for billing

❖ Elements, features and special notes

- Dustbins
- Ticket checker
- Printer
- Security guards
- Parking



AEIOU framework: Group id: Date: Sheet No:

Environment

Project Name : E-PAPER

General impressions / Observations

Style, materials & atmosphere)

Rainy Season

Crowd.

Sale / Vacation / festivals

Less customers

Requires more time for billing

Huge waste of paper due to more customers.

Floor plan

Elements, features and special notes

Dustbins

Conductor

Security Guards

Printer

Parking

Scene

Figure – 7: Image of Environment Canvas

2.3 Interactions Record Sheet

❖ General impressions / Observations

➢ Customer	Customer/helper/cashier	Shopping, billing, payment
➢ Traveler	Traveler, guide, driver	travelling

Table – 4: General impressions / Observations

❖ Elements, features and special notes

- Shopping malls
- Bus stations
- Rail – way stations
- Toll booths

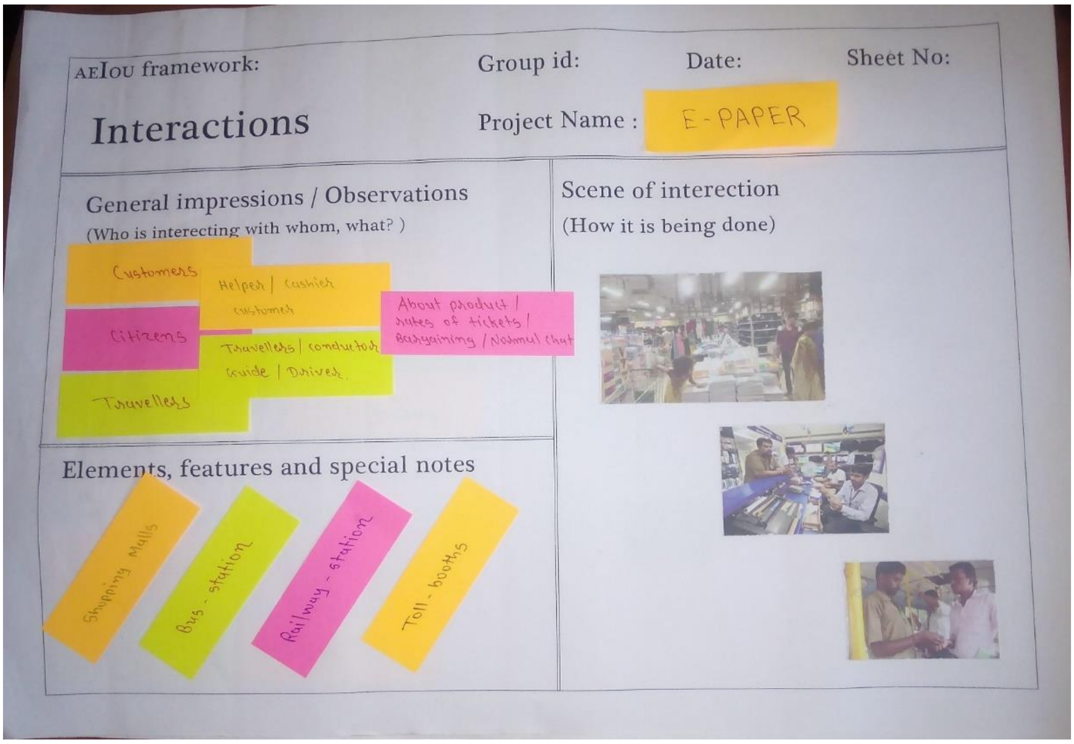

AEIOU framework:
Interactions
Group id:
Date:
Sheet No:
Project Name :
E-PAPER
General impressions / Observations
(Who is interecting with whom, what?)
Customers
Helper / cashier customer
Citizens
Travellers / conductor Guide / Driver
Travellers
About product / rates of tickets / Bargaining / Normal chat
Scene of interection
(How it is being done)
Elements, features and special notes
Shopping Malls
Bus - station
Railway - station
Toll - booths

Figure – 8: Image of Interactions Canvas

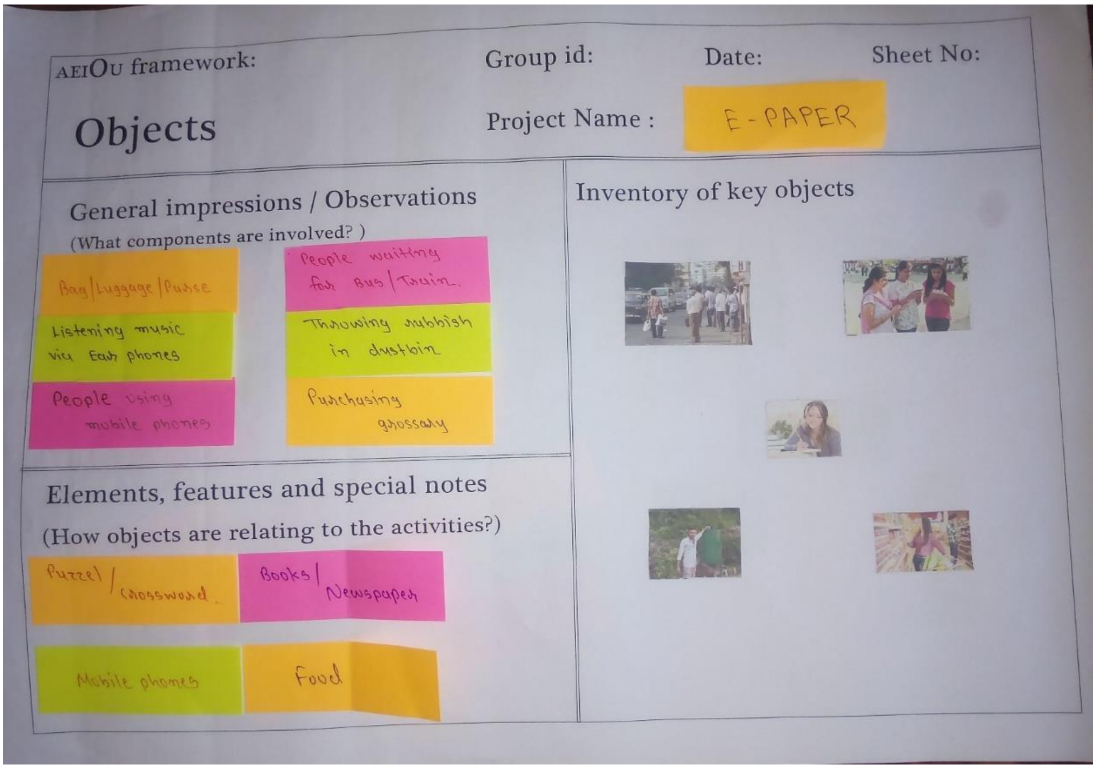
2.4 Object Record sheet

❖ General impressions / Observations

- Mobile
- Computer
- Internet connection
- Bags/luggage/purse
- Listening music
- People using mobiles
- People waiting for bus/train
- Throwing rubbish in dustbin
- Purchasing grocery

❖ Elements, features and special notes

- Puzzle/crossword
- Books/newspapers
- Mobile phones
- Food



AEIOu framework: Group id: Date: Sheet No:

Objects

Project Name : E - PAPER

General impressions / Observations

(What components are involved?)

Bag/Luggage/Purse

Listening music via Ear phones

People using mobile phones

People waiting for Bus/Train.

Throwing rubbish in dustbin

Purchasing grossary

Inventory of key objects

Elements, features and special notes

(How objects are relating to the activities?)

Puzzel/crossword.

Books/Newspaper

Mobile phones

Food

Figure – 9: Image of Objects Canvas

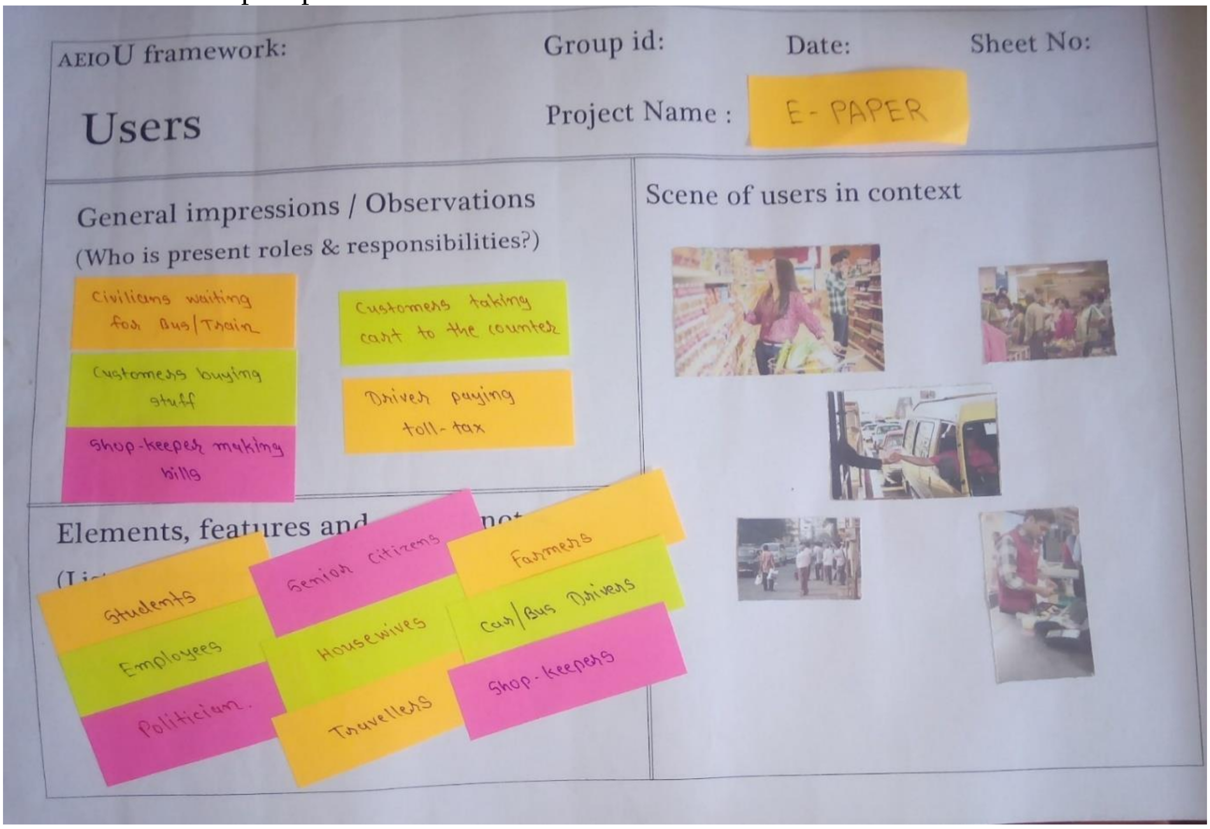
2.5 User Record sheet

❖ General impressions / Observations

- Civilians waiting for bus/train
- Customers buying stuff
- Shopkeeper making bills
- Customers taking cart to counter
- Driver paying toll - tax

❖ Inventory of people

- Students
- Politicians
- Businessman
- Employees
- Senior citizens
- House wives
- Travelers
- Farmer
- Car/bus drivers
- Shopkeepers



AEIOU framework: Group id: Date: Sheet No:

Users

Project Name : E-PAPER

General impressions / Observations
(Who is present roles & responsibilities?)

Civilians waiting for Bus/Train

Customers taking cart to the counter

Customers buying stuff

Driver paying toll-tax

Shop-keeper making bills

Scene of users in context

Elements, features and no…
(Li…

Students

Senior citizens

Farmers

Employees

Housewives

Car/Bus Drivers

Politician.

Travellers

Shop-keepers

Figure – 10: Image of Users Canvas

Chapter 3: Review Report of Prior Art Search

3.1 Fact sheet: Saving paper in schools

Why It Matters

Producing and disposing of paper pollutes land, air and water, threatens wildlife and affects human health.

A single sheet of paper may seem like an insignificant thing. But the use of thousands of sheets each day can have a huge impact on the environment. That's especially true in schools, where students and teachers and other staff all use lots of paper.

It all starts when trees are cut down, which hurts both forests and the animals that live in them. Cutting down forests even affects the earth's climate, since trees absorb carbon dioxide, one of the greenhouse gases that cause global warming. (The paper industry also creates lots of greenhouse gases -- it's the third largest source of global warming pollution in the world!)

Project Ideas:

- **Paper proposals**: Take some time to observe how paper is used in your school. You'll probably discover examples of waste. Make a list of five to10 ways your school can use paper more efficiently, then present it to your principal.

- **Scrap pile**: Choose a spot in your class for collecting paper that can be reused. Some of it might make good scribble paper. Or you could collect some bigger sheets and staple them together, perhaps on a cardboard backing, to make assignment pads. Colored paper is good for art projects -- try cutting it up to use for "mosaics."

- **Make your own recycled paper**: Making your own paper is a fun project. And you'll be doing your own recycling! There are lots of books and websites to help you. Try this guide from the Exploratorium.

- **Paper review**: Find out what kind of paper your school buys for its offices and classrooms. Check the packaging to see if it's recycled (it should have at least 30 percent post-consumer waste). Also, see if it has a PCF label to show it was made without chlorine bleaching. If your school uses paper whitened with chlorine bleach, talk to your principal.

3.2 Development of an Online Bus Ticket Reservation System for a Transportation Service in Nigeria

E-ticketing could be extended to major entertainment and touristic sites and thus facilitate access to major points of interest within cities, making e-ticketing also interesting for travelers. Urban tourism is the fastest growing tourism sector in the world (Paskaleva, 2014).

Public transport operators have been trying to replace paper-based tickets with electronic media, and many countries have implemented or are about to introduce e-ticketing systems. The main characteristic of e-ticketing is that tickets are sold and stored in electronic devices.

The system is very simple in design and to implement. The system requires very low system resources and the system will work in almost all configurations. It has got following features:

- It will ensure data accuracy.

- Records will be efficiently maintained by DBMS.

- Availability of seats can be enquired easily.

- Passengers can also cancel their tickets easily.

- Minimum time needed for the various processing.

- It will provide better Service.

3.3 E - Paper

Basic Features:

- User can get a copy of bill/ticket/receipt through mobile phone.
- It’s easy to keep record of customers.
- No worries of storing bill/ticket/receipt in paper format.

Key features:

- Saves paper that is saving nature itself.
- Saves time.
- Reduces the efforts of human beings.

Compared to similar project:

- Instead of recycling of paper we are just removing the paper by providing E-PAPER.
- The second project is as similar as ours but we are not providing online transactions.

Chapter 4: Learning needs matrix

Learning Needs Matrix	During BE II	During BE III	During BE IV
Tools/ Methods/ Theories/ Application process involved.	1. Analysis of system 2. Computer 3. Mobile	1. Disk for storing the database. 2. Internet. 3. Server	1. Code Matcher 2. Code Generator
Software/ Simulation/ Skill/ Mathematical Requirement.	1. C 2. C++. 3. My SQL	1. HTML 2. CSS 3. Java.	1. PHP. 2. .NET 3. Advanced Java.
Applicable standards and design specifications/ Principles and Experiments.	1. Easy to understand. 2. Accurate	1. Faster 2. Reduces Human efforts	1. Echo- friendly 2. Data records.
Component materials' strength criteria(exploration- varieties/ testing requirements)	1. Understanding the needs of people.	1. Bootstrap programming. 2. Security of database.	3. Testing

Table – 5: Learning needs matrix

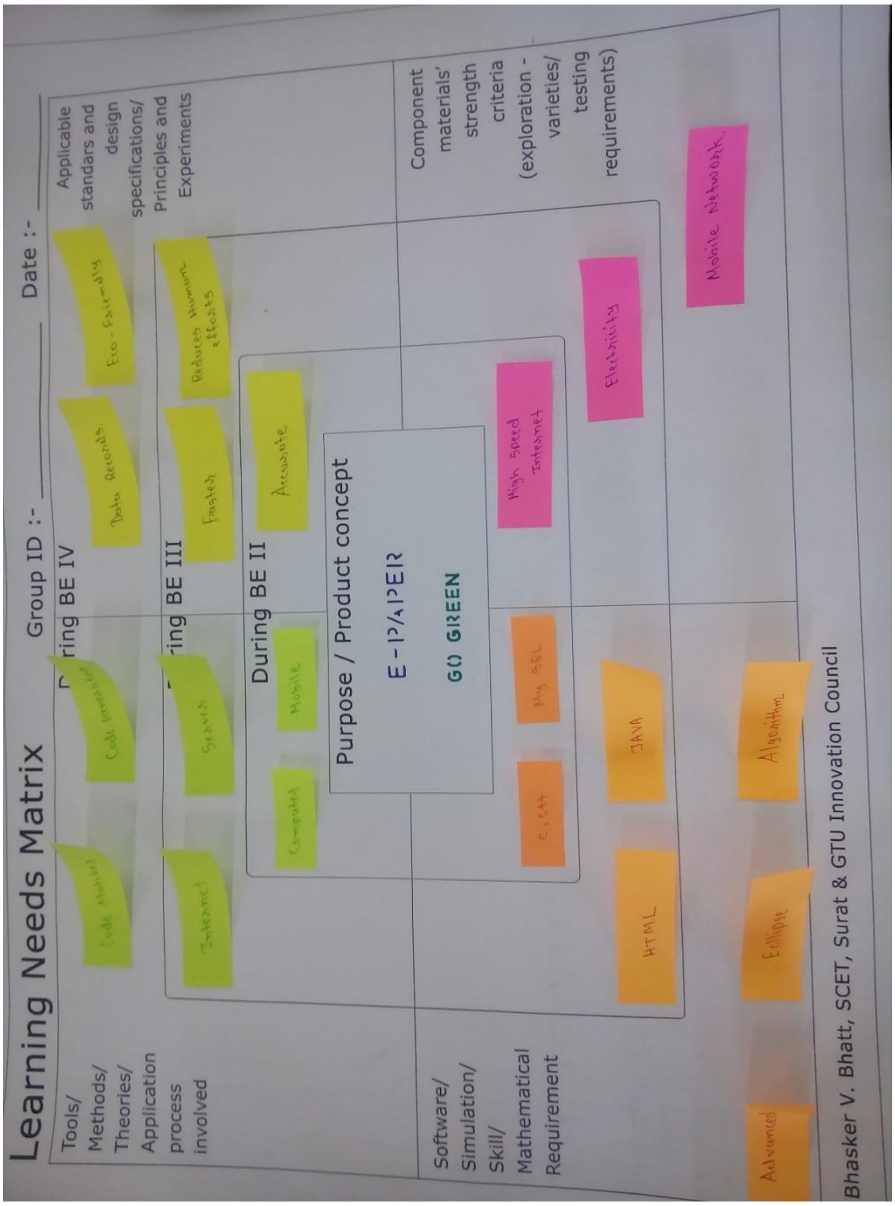

Learning Needs Matrix
Group ID :-
Date :-
Tools/ Methods/ Theories/ Application process involved
Applicable standars and design specifications/ Principles and Experiments
Software/ Simulation/ Skill/ Mathematical Requirement
Component materials' strength criteria (exploration - varieties/ testing requirements)
During BE IV
During BE III
During BE II
Purpose / Product concept
E-PAPER
GO GREEN
Code Module
Code Generator
Data Records
Eco-Friendly
Internet
Servers
Fastest
Reduces Human efforts
Computer
Mobile
Accurate
C, C++
My SQL
High Speed Internet
Electricity
Mobile Network
HTML
JAVA
Eclipse
Algorithm
Advanced
Bhasker V. Bhatt, SCET, Surat & GTU Innovation Council

Figure – 11: Image of Learning needs matrix Canvas

Chapter 5: Snapshot of Fast Prototype

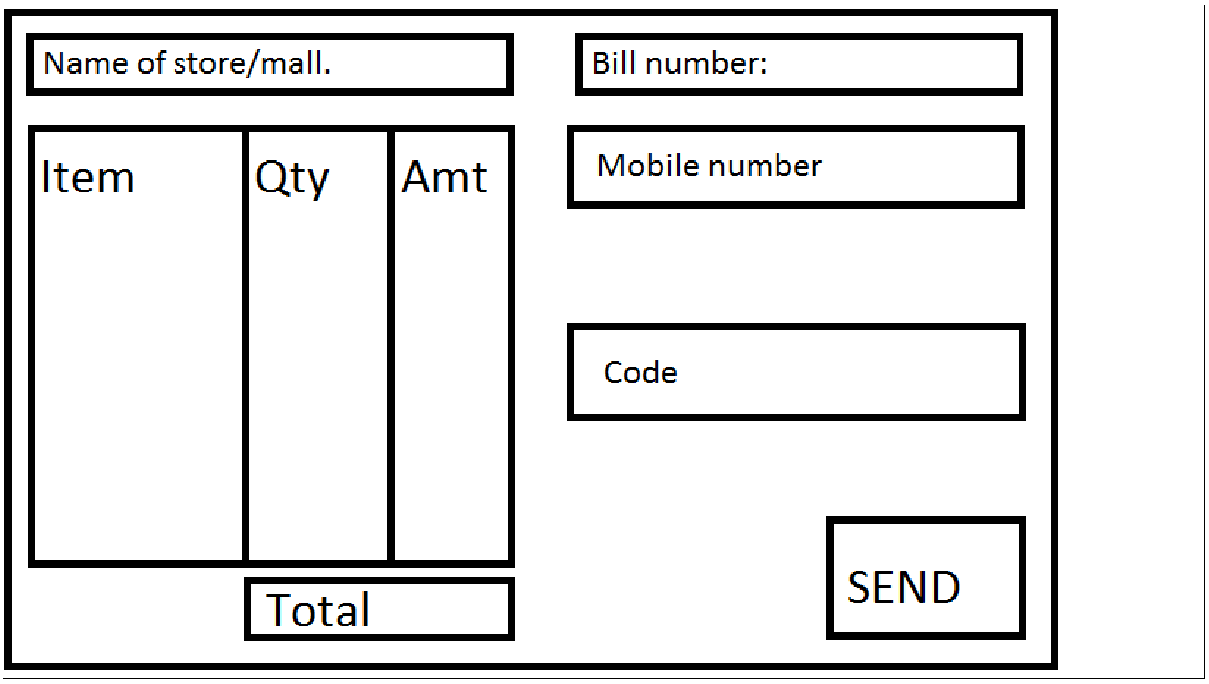


Figure – 12: Image of billing Prototype

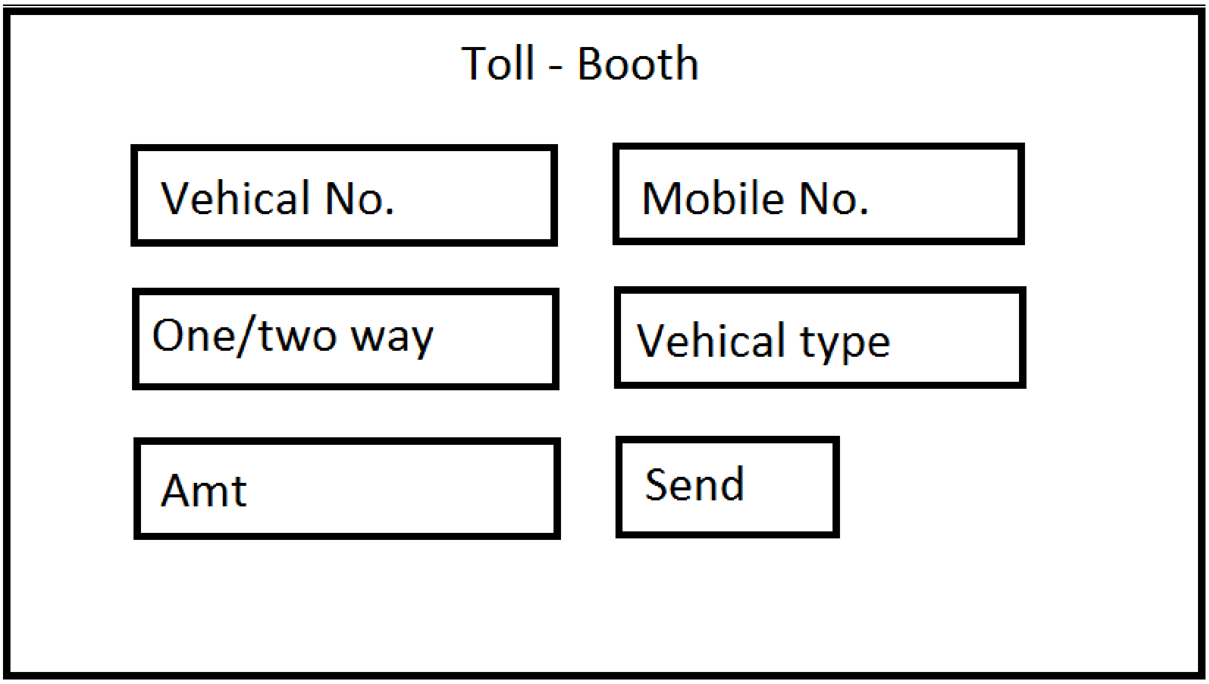


Figure – 13: Image of tollbooth receipt Prototype

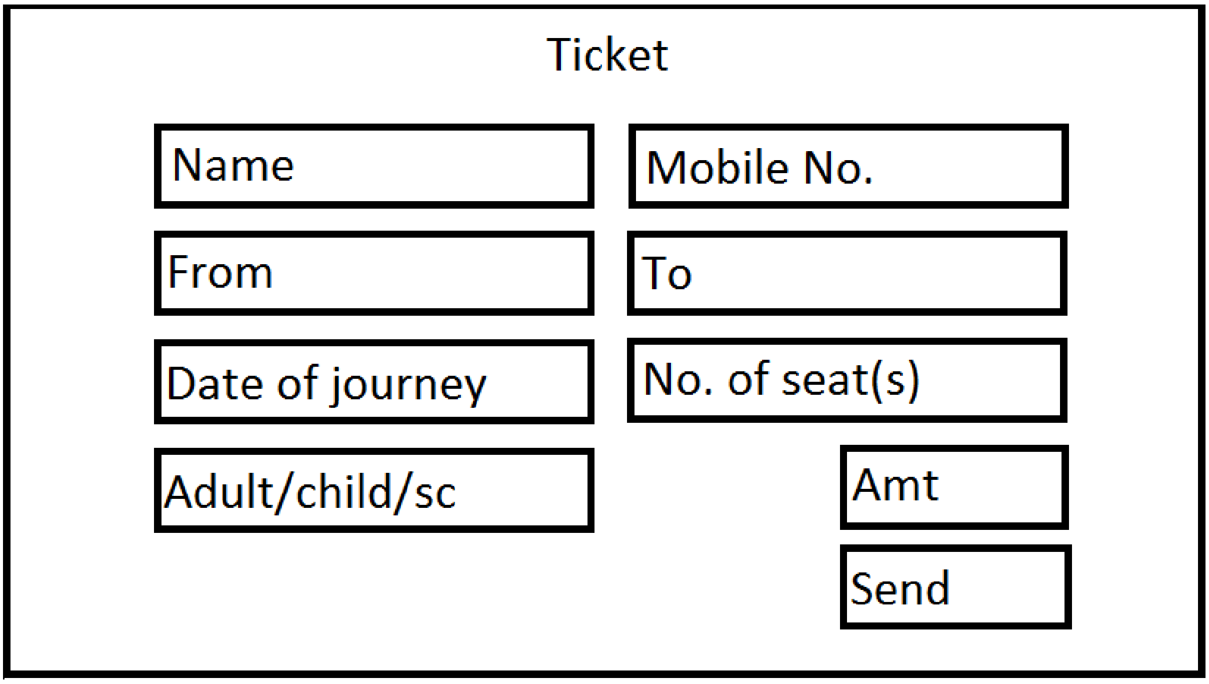
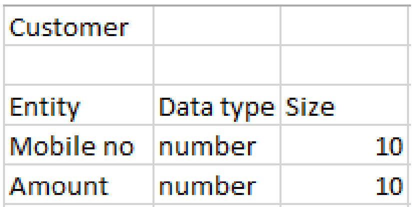


Figure – 14: Image of ticket Prototype

CHAPTER 6: DATA DICTIONARY

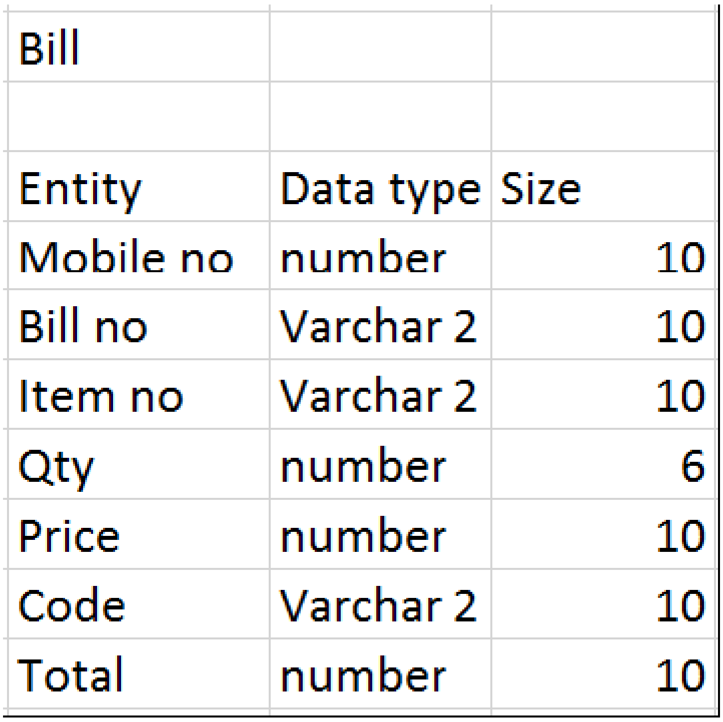
A data dictionary or metadata repository is a centralized repository of information about data such as meaning, relationships to other data, origin, usage, and format. It has a list of all files in the database, the number of records in each file, and the names and types of each field. Most database management systems keep the data dictionary hidden from users to prevent them from accidentally destroying its contents.

Data dictionaries do not contain any actual data from the database, only bookkeeping information for managing it. Without a data dictionary, however, a database management system cannot access data from the database.



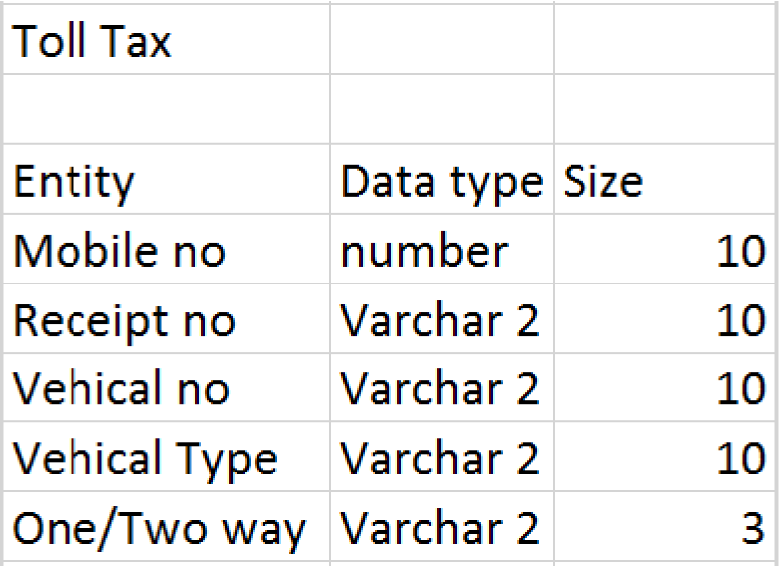
Customer		
Entity	Data type	Size
Mobile no	number	10
Amount	number	10

Table 6: Customer



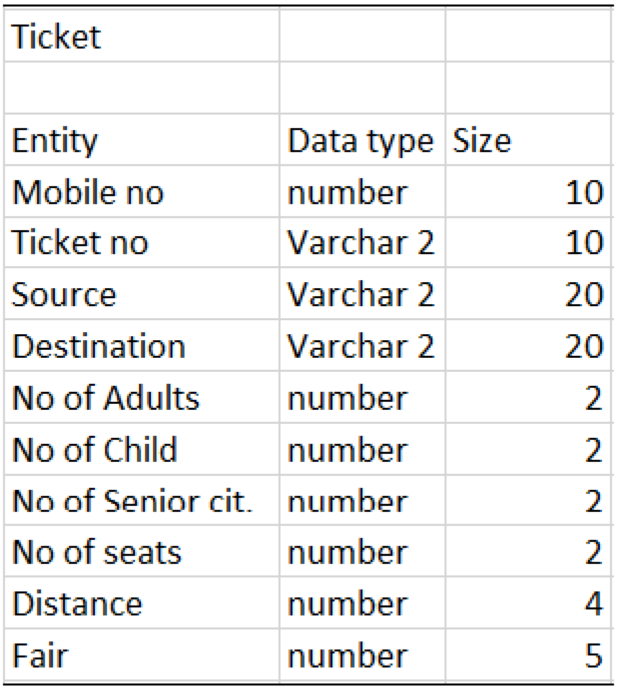
Bill		
Entity	Data type	Size
Mobile no	number	10
Bill no	Varchar 2	10
Item no	Varchar 2	10
Qty	number	6
Price	number	10
Code	Varchar 2	10
Total	number	10

Table 7: Bill



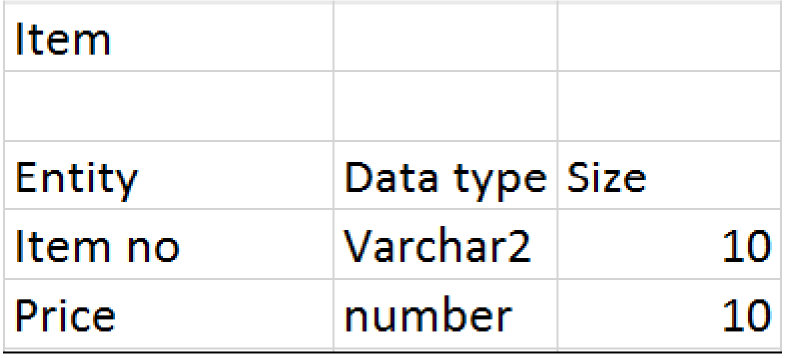
Toll Tax		
Entity	Data type	Size
Mobile no	number	10
Receipt no	Varchar 2	10
Vehical no	Varchar 2	10
Vehical Type	Varchar 2	10
One/Two way	Varchar 2	3

Table 8: Toll Tax



Ticket		
Entity	Data type	Size
Mobile no	number	10
Ticket no	Varchar 2	10
Source	Varchar 2	20
Destination	Varchar 2	20
No of Adults	number	2
No of Child	number	2
No of Senior cit.	number	2
No of seats	number	2
Distance	number	4
Fair	number	5

Table 9: Ticket



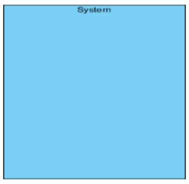
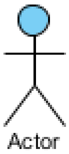
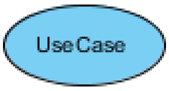
Item		
Entity	Data type	Size
Item no	Varchar2	10
Price	number	10

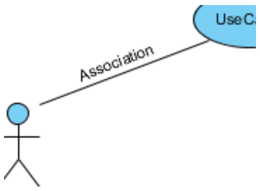
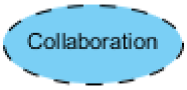
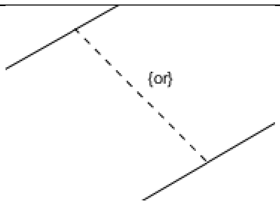
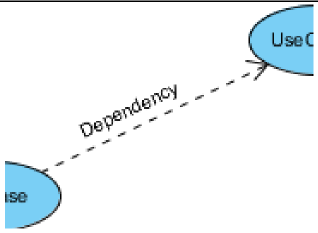
Table 10: Item

Chapter 7: Diagrams

7.1 Use Case Diagram

1. Use:
    - It is used to model the system/subsystem of an application.
    - It is used to get an outside view of a system.
    - It is used to gather requirements of a system.
    - It is used to identify external and internal factors influencing the system.
    - It is used to show the interacting among the requirements are actors.
    - Used to capture the dynamic aspect of a system.
2. Symbol:

Symbol	Description
	System: If a subject (or system boundary) is displayed, the use case ellipse is visually located inside the system boundary rectangle.
	Actor: An Actor models a type of role played by an entity that interacts with the subject (e.g., by exchanging signals and data), but which is external to the subject (i.e., in the sense that an instance of an actor is not a part of the instance of its corresponding subject). Actors may represent roles played by human users, external hardware, or other subjects. Note that an actor does not necessarily represent a specific physical entity.
	Use Case: A use case is the specification of a set of actions performed by a system, which yields an observable result that is, typically, of value for one or more actors or other stakeholders of the system.

	Association: An association specifies a semantic relationship that can occur between typed instances. It has at least two ends represented by properties, each of which is connected to the type of the end. More than one end of the association may have the same type. An end property of an association that is owned by an end class or that is a navigable owned end of the association indicates that the association is navigable from the opposite ends; otherwise, the association is not navigable from the opposite ends.
	Collaboration: A collaboration specifies a view (or projection) of a set of cooperating classifiers. It describes the required links between instances that play the roles of the collaboration, as well as the features required of the classifiers that specify the participating instances. Several collaborations may describe different projections of the same set of classifiers.
	Constraint: A condition or restriction expressed in natural language text or in a machine readable language for the purpose of declaring some of the semantics of an element.
	Dependency: A dependency is a relationship that signifies that a single or a set of model elements requires other model elements for their specification or implementation. This means that the complete semantics of the depending elements is either semantically or structurally dependent on the definition of the supplier element(s).

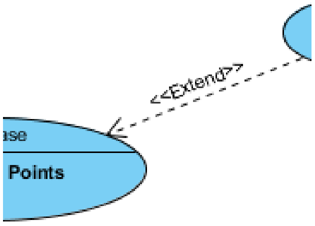
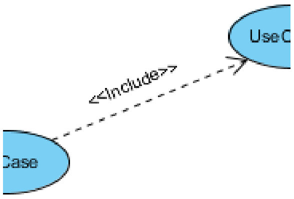
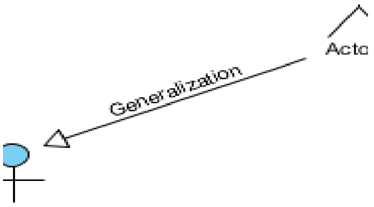
	Extend: This relationship specifies that the behaviour of a use case may be extended by the behaviour of another (usually supplementary) use case. The extension takes place at one or more specific extension points defined in the extended use case.The same extending use case can extend more than one use case. Furthermore, an extending use case may itself be extended.
	Include: Include is a Directed Relationship between two use cases, implying that the behaviour of the included use case is inserted into the behaviour of the including use case. The including use case may only depend on the result (value) of the included use case. This value is obtained as a result of the execution of the includeduse case.
	Generalization: A generalization is a taxonomic relationship between a more general classifier and a more specific classifier. Each instance of the specific classifier is also an indirect instance of the general classifier. Thus, the specific classifier inherits the features of the more general classifier.

Table – 11: Use case symbols

3. Diagram:

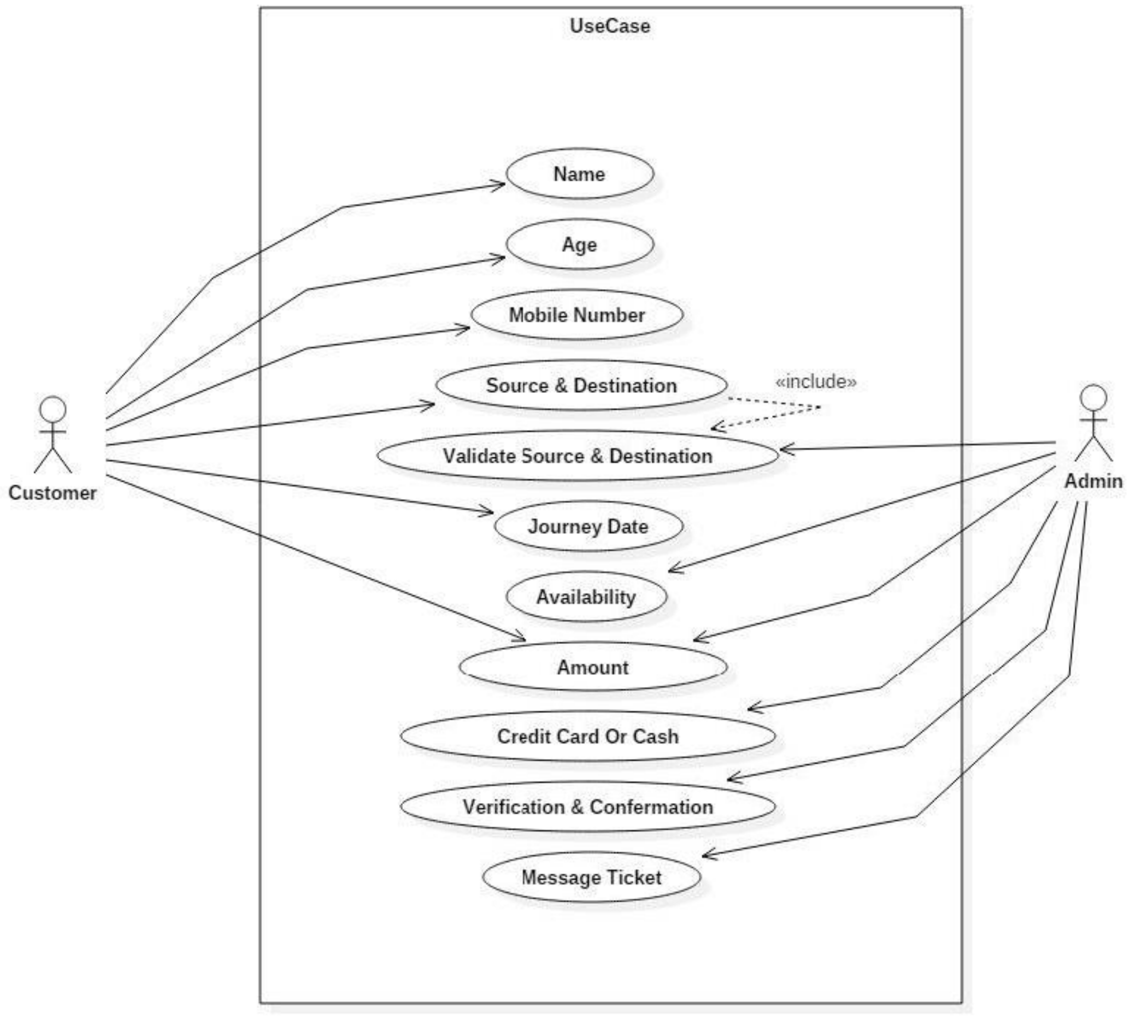


Figure – 15: Usecase diagram for ticking

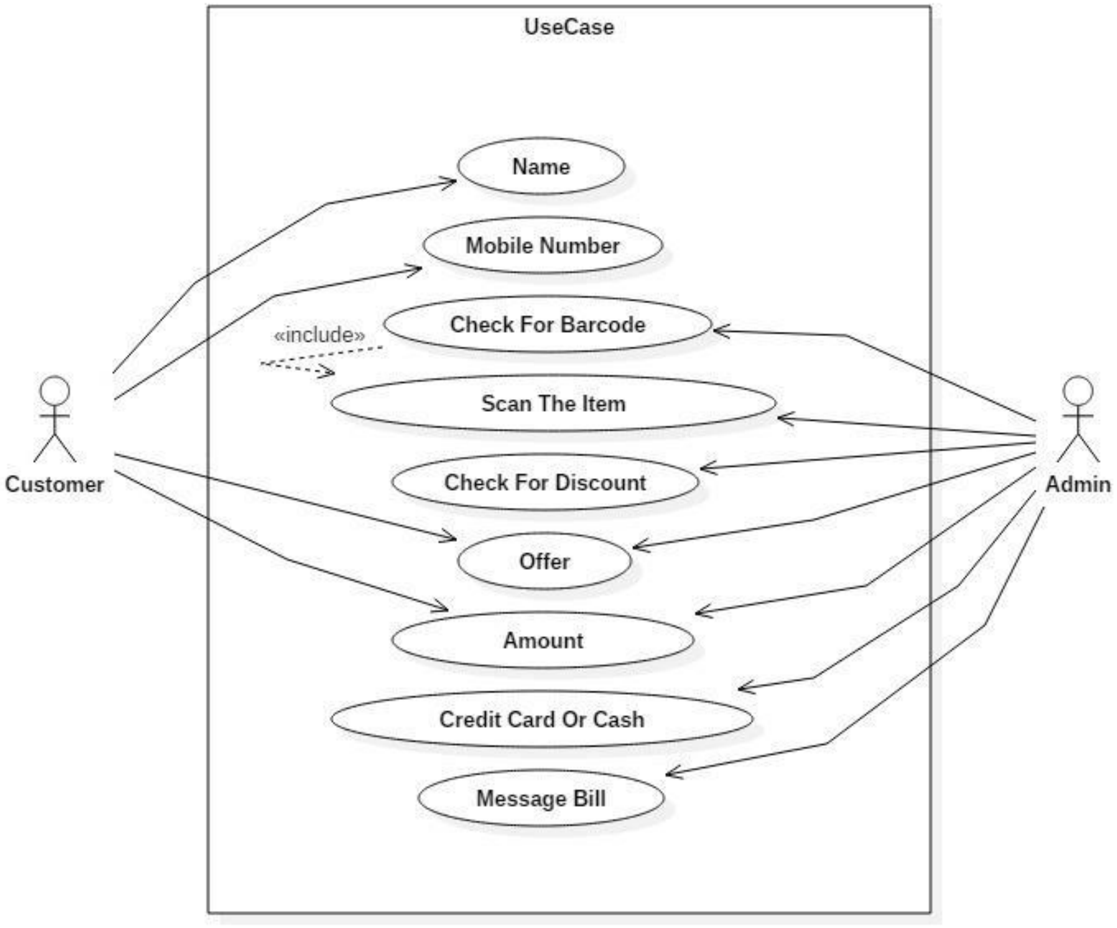
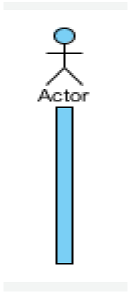
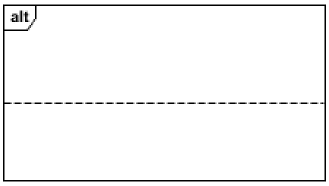
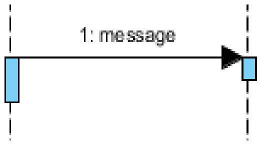


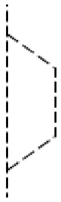
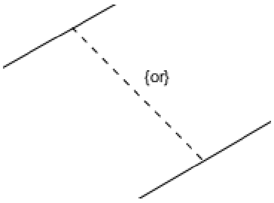
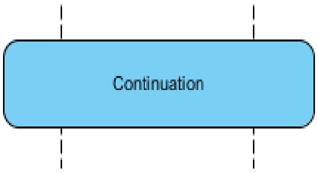
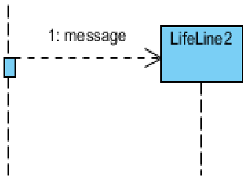
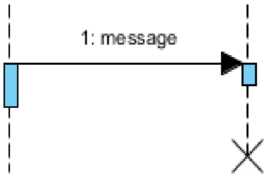
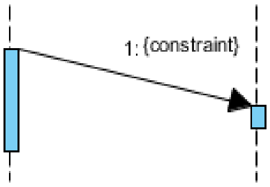
Figure – 16: Usecase diagram for billing

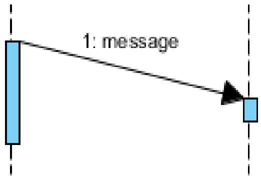
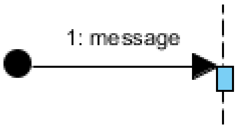
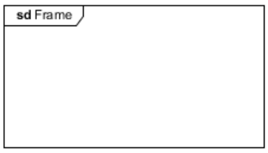
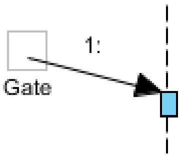
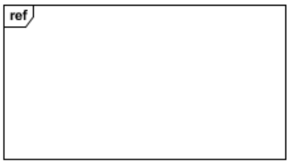
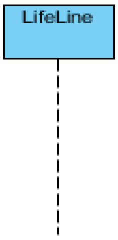
7.2 Sequence Diagram

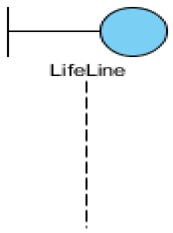
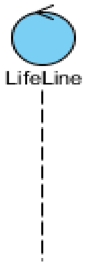
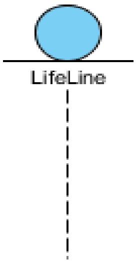
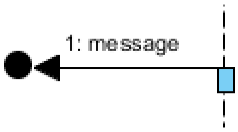
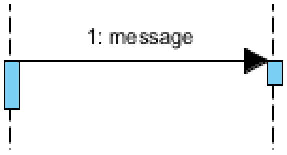
1. Use:
   - Used to show how the objects interact with others object in a particular scenario of a use case.
   - It is used to capture the orders of message flowing from one object to another.
   - It is used to capture dynamic nature but from a different angle.

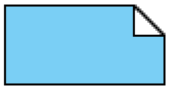
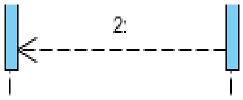
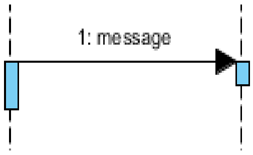
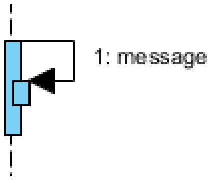
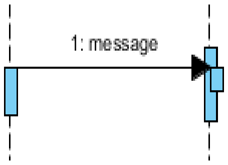
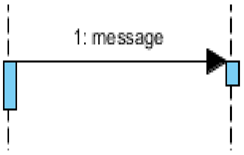
2. Symbol:

Symbol	Description
	Actor: An Actor models a type of role played by an entity that interacts with the subject (e.g., by exchanging signals and data), but which is external to the subject (i.e., in the sense that an instance of an actor is not a part of the instance of its corresponding subject). Actors may represent roles played by human users, external hardware, or other subjects. Note that an actor does not necessarily represent a specific physical entity.
	Alternative Combined Fragment: A combined fragment defines an expression of interaction fragments. A combined fragment is defined by an interaction operator and corresponding interaction operands. Through the use of Combined Fragments the user will be able to describe a number of traces in a compact and concise manner.
	Call Message: A message defines a particular communication between Lifelines of an Interaction. Call message is a kind of message that represents an invocation of operation of target lifeline.

	Concurrent: A concurrent represents a session of concurrent method invocation along an activation. It is placed on top of an activation.
	Constraint: A condition or restriction expressed in natural language text or in a machine readable language for the purpose of declaring some of the semantics of an element.
	Continuation: A Continuation is a syntactic way to define continuations of different branches of an Alternative Combined Fragment. Continuation is intuitively similar to labels representing intermediate points in a flow of control.
	Create Message: A message defines a particular communication between Lifelines of an Interaction. Create message is a kind of message that represents the instantiation of (target) lifeline.
	Destroy Message: A message defines a particular communication between Lifelines of an Interaction. Destroy message is a kind of message that represents the request of destroying the lifecycle of target lifeline.
	Duration Constraint: A Duration Constraint defines a Constraint that refers to a Duration Interval. A duration used to determine whether the constraint is satisfied.

	Duration Message: A message defines a particular communication between Lifelines of an Interaction. Duration message shows the distance between two time instants for a message invocation.
	Found Message: A found message is a message where the receiving event occurrence is known, but there is no (known) sending event occurrence. We interpret this to be because the origin of the message is outside the scope of the description. This may for example be noise or other activity that we do not want to describe in detail.
	Frame: A frame represents an interaction, which is a unit of behaviour that focuses on the observable exchange of information between Connectable Elements.
	Gate: A Gate is a connection point for relating a Message outside an Interaction Fragment with a Message inside the Interaction Fragment.
	Interaction Use: An Interaction Use refers to an Interaction. The Interaction Use is a shorthand for copying the contents of the referred Interaction where the Interaction Use is. To be accurate the copying must take into account substituting parameters with arguments and connect the formal gates with the actual ones.
	LifeLine: A lifeline represents an individual participant in the Interaction.

	LifeLine <<Boundary>>: A lifeline represents an individual participant in the Interaction.
	LifeLine <<Control>>: A lifeline represents an individual participant in the Interaction.
	LifeLine <<Entity>>: A lifeline represents an individual participant in the Interaction.
	Loop Combined Fragment: A combined fragment defines an expression of interaction fragments. A combined fragment is defined by an interaction operator and corresponding interaction operands. Through the use of Combined Fragments the user will be able to describe a number of traces in a compact and concise manner.
	Lost Message: A lost message is a message where the sending event occurrence is known, but there is no receiving event occurrence. We interpret this to be because the message never reached its destination.
	Message: A message defines a particular communication between Lifelines of an Interaction.

	Note: A note (comment) gives the ability to attach various remarks to elements. A comment carries no semantic force, but may contain information that is useful to a modeller.
	Return Message: A message defines a particular communication between Lifelines of an Interaction. Return message is a kind of message that represents the pass of information back to the caller of a corresponded former message.
	Send Message: A message defines a particular communication between Lifelines of an Interaction. Send message is a kind of message that represents the start of execution.
	Recursive Mess A message defines a particular communication between Lifelines of an Interaction. Recursive message is a kind of message that represents the invocation of message of the same lifeline. Its target points to an activation on top of the activation where the message was invoked from.
	Re-entrant Message: A message defines a particular communication between Lifelines of an Interaction. A re-entrant message points to an activation on top of another activation.
	Terminate Message: A message defines a particular communication between Lifelines of an Interaction. Terminate message is a kind of message that represents the termination of execution.

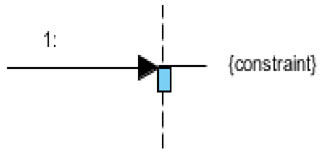
	Time Constraint: A Time Constraint defines a Constraint that refers to a Time Interval.

Table – 12: Sequence symbols

3. Diagram:

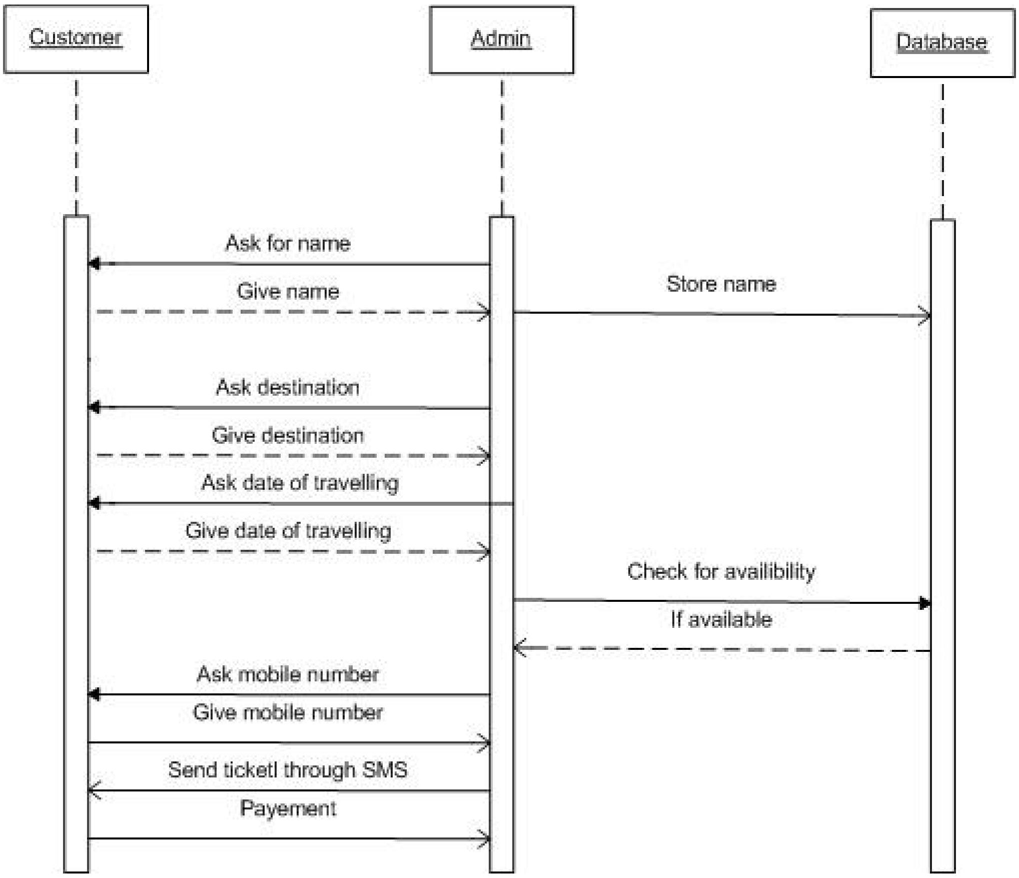


Figure – 17: Sequence diagram for ticketing.

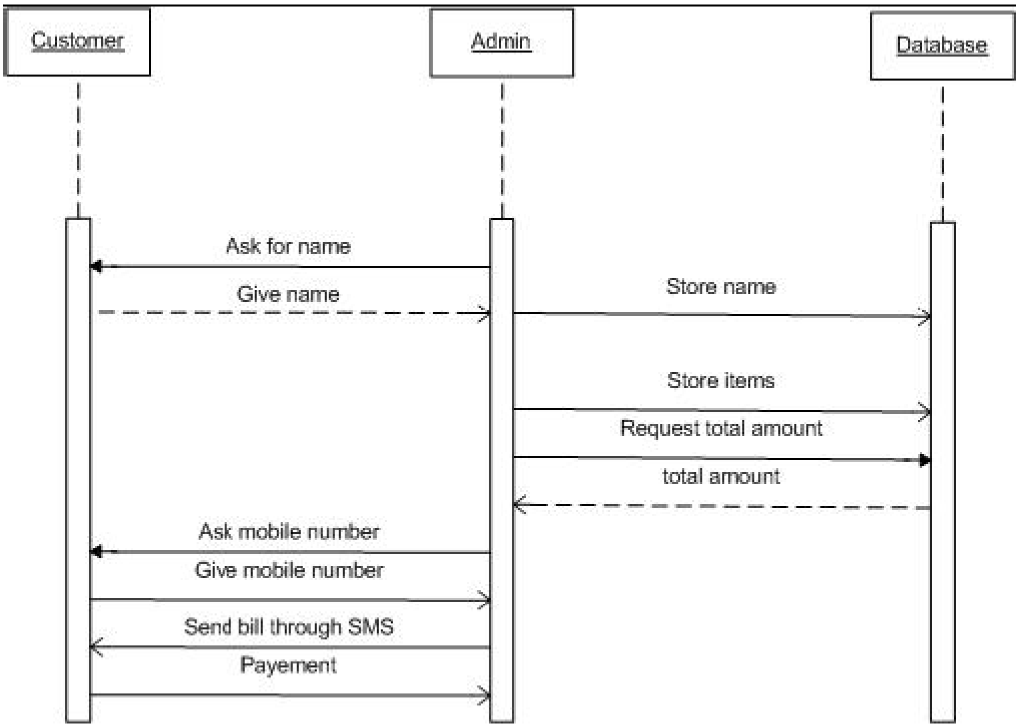


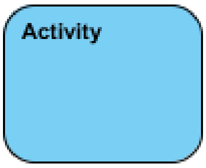
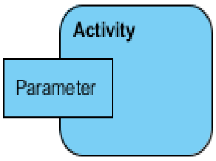
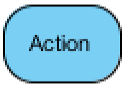
Figure – 18: Sequence diagram for billing.

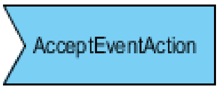
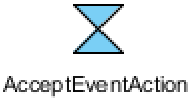
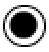
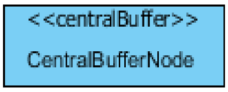
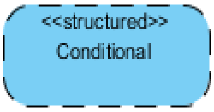
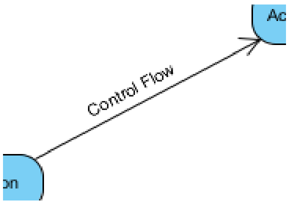
7.3 Activity Diagram

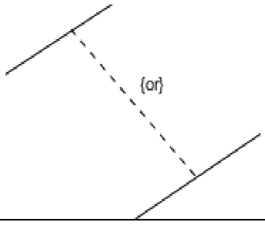
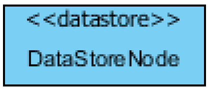
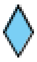
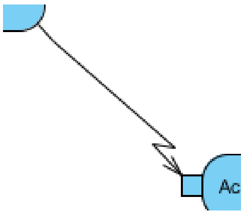
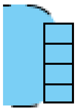
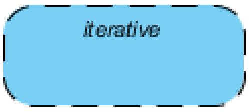
1. Use:

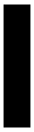
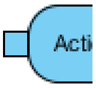
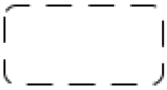
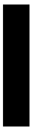
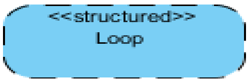
- Used to show message flow from one activity to another.
- It is used for visualizing dynamic nature of a system.
- It is used to construct the executable system by using forward and reverse engineering techniques.
- It is used to describe dynamic aspects of the system.

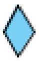
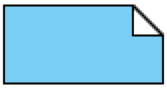
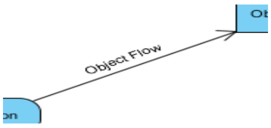
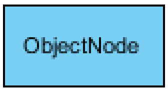
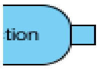
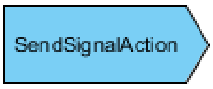
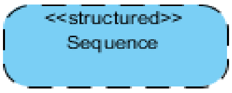
2. Symbol:

Symbol	Description
	Activity: Activities may describe procedural computation. In this context, they are the methods corresponding to operations on classes. Activities may be applied to organizational modelling for business process engineering and workflow. In this context, events often originate from inside the system, such as the finishing of a task, but also from outside the system, such as a customer call. Activities can also be used for information system modelling to specify system level processes.
	Activity Parameter Node: Activity parameter nodes are object nodes at the beginning and end of flows that provide a means to accept inputs to an activity and provide outputs from the activity, through the activity parameters. Activity parameters inherit support for streaming and exceptions from Parameter.
	Action: An action represents a single step within an activity, that is, one that is not further decomposed within the activity. An activity represents a behaviour that is composed of individual elements that are actions. An action may have sets of incoming and outgoing activity edges that specify control flow and data flow from and to other nodes. An action will not begin execution until all of its input conditions are satisfied. The completion of the execution of an action may enable the execution of a set of

	successor nodes and actions that take their inputs from the outputs of the action.
	Accept Event Action: Accept Event Action is an action that waits for the occurrence of an event meeting specified condition.
	Accept Time Event Action: If the occurrence is a time event occurrence, the result value contains the time at which the occurrence transpired. Such an action is informally called a wait time action.
	Activity Final Node: An activity may have more than one activity final node. The first one reached stops all flows in the activity.
	Central Buffer Node: A central buffer node accepts tokens from upstream object nodes and passes them along to downstream object nodes. They act as a buffer for multiple in flows and out flows from other object nodes. They do not connect directly to actions.
	Conditional Node Specification: A conditional node is a structured activity node that represents an exclusive choice among some number of alternatives.
	Control Flow: A control flow is an edge that starts an activity node after the previous one is finished.

	Constraint: A condition or restriction expressed in natural language text or in a machine readable language for the purpose of declaring some of the semantics of an element.
	Data Store Node:: Determines where the data store node appears within different Namespaces within the overall model, and its accessibility.
	Decision Node: A decision node accepts tokens on an incoming edge and presents them to multiple outgoing edges. Which of the edges is actually traversed depends on the evaluation of the guards on the outgoing edges.
	Exception Handler: An exception handler is an element that specifies a body to execute in case the specified exception occurs during the execution of the protected node.
	Expansion Node: An expansion node is an object node used to indicate a flow across the boundary of an expansion region. A flow into a region contains a collection that is broken into its individual elements inside the region, which is executed once per element. A flow out of a region combines individual elements into a collection for use outside the region.
	Expansion Region: An expansion region is a strictly nested region of an activity with explicit input and outputs (modelled as Expansion Nodes). Each input is a collection of values. If there are multiple inputs, each of them must hold the same kind of collection, although the types of the elements in the different collections may vary. The expansion region is executed once for each element (or

	position) in the input collection.
	Flow Final Node: A flow final destroys all tokens that arrive at it. It has no effect on other flows in the activity.
	Fork Node: A fork node is a control node that splits a flow into multiple concurrent flows. A fork node has one incoming edge and multiple outgoing edges.
	Initial Node: An initial node is a control node at which flow starts when the activity is invoked. An activity may have more than one initial node.
	Input Pin: Input pins are object nodes that receive values from other actions through object flows. See Pin, Action, and ObjectNode for more details.
	Join Node: An interruptible activity region is an activity group that supports termination of tokens flowing in the portions of an activity. An interruptible region contains activity nodes. When a token leaves an interruptible region via edges designated by the region as interrupting edges, all tokens and behaviour in the region are terminated.
	Join Node: A join node is a control node that synchronizes multiple flows. A join node has multiple incoming edges and one outgoing edge.
	Loop Node: A loop node is a structured activity node that represents a loop with setup, test, and body sections.

	Merge Node: A merge node is a control node that brings together multiple alternate flows. It is not used to synchronize concurrent flows but to accept one among several alternate flows. A merge node has multiple incoming edges and a single outgoing edge.
	Note: A note (comment) gives the ability to attach various remarks to elements. A comment carries no semantic force, but may contain information that is useful to a modeller.
	Object Flow: An object flow is an activity edge that can have objects or data passing along it.
	Object Node: An object node is an activity node that indicates an instance of a particular classifier, possibly in a particular state, may be available at a particular point in the activity. Object nodes can be used in a variety of ways, depending on where objects are flowing from and to, as described in the semantics sub clause.
	Output Pin: Output pins are object nodes that deliver values to other actions through object flows.
	Send Signal Action: Send Signal Action is an action that creates a signal instance from its inputs, and transmits it to the target object, where it may cause the firing of a state machine transition or the execution of an activity.
	Sequence Node: A sequence node is a structured activity node that executes its actions in order.

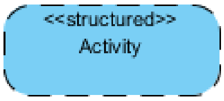
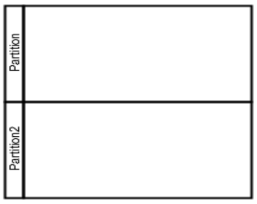
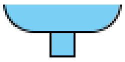
	Structured Activity Node: A sequence node is a structured activity node that executes its actions in order.
	Swim lane: Swim lane is used for partitioning the children in an activity diagram.
	Value Pin: A value pin is an input pin that provides a value to an action that does not come from an incoming object flow edge.

Table – 13: Activity symbols

3. Diagram:

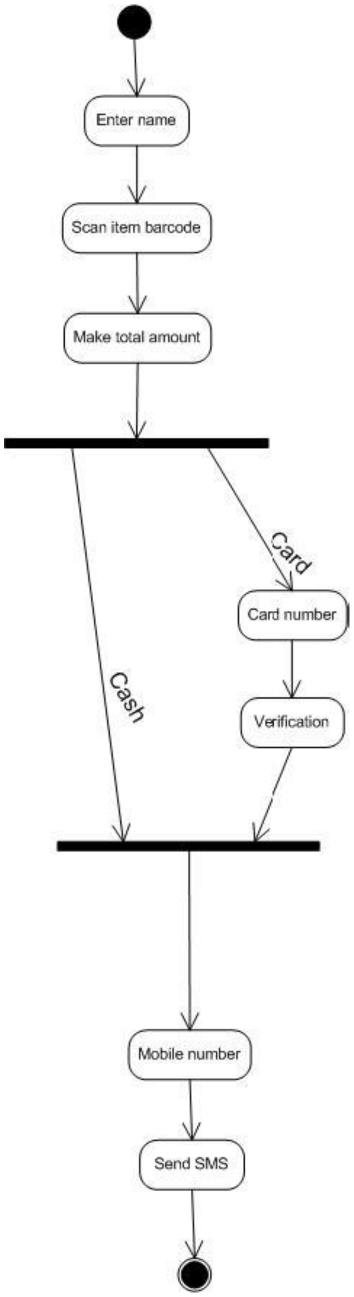


Figure – 19: Activity diagram for billing

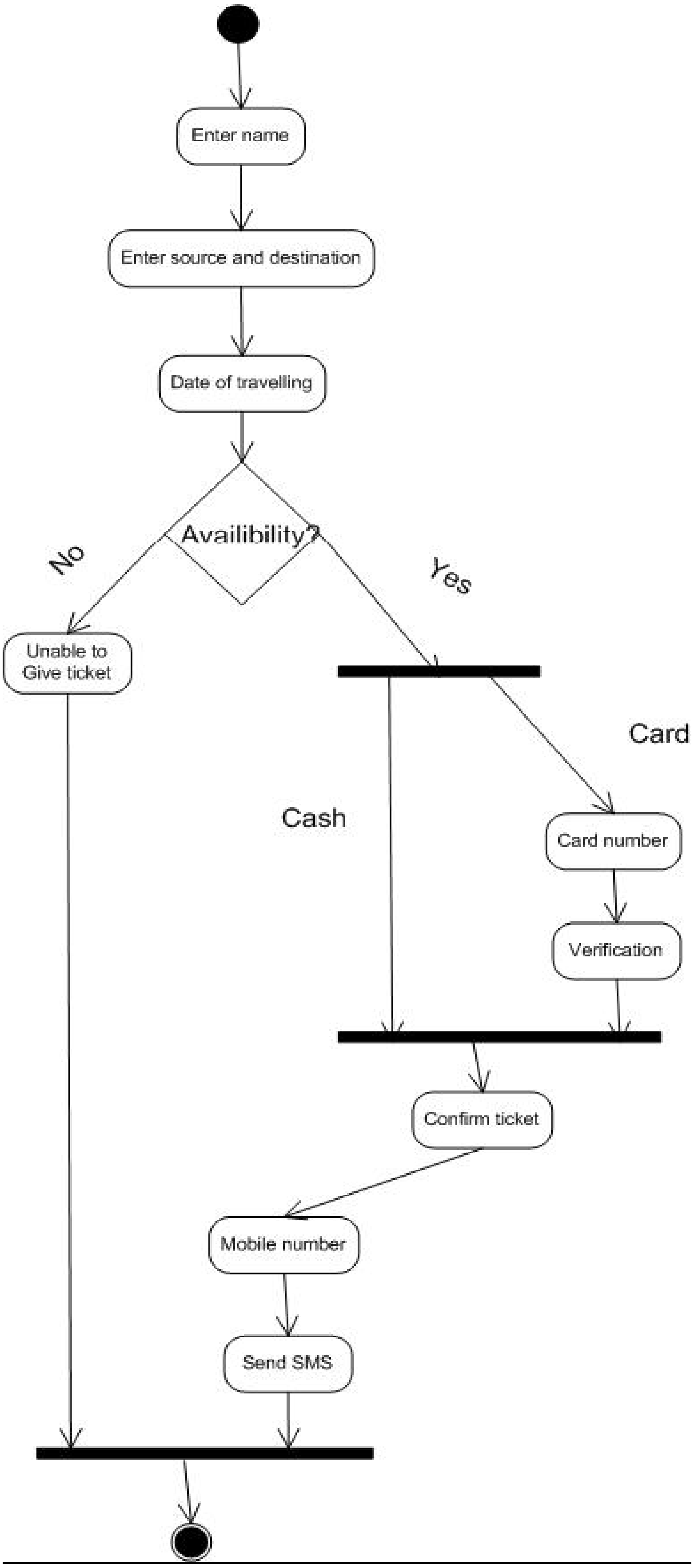


Figure – 20: Activity diagram for ticketing

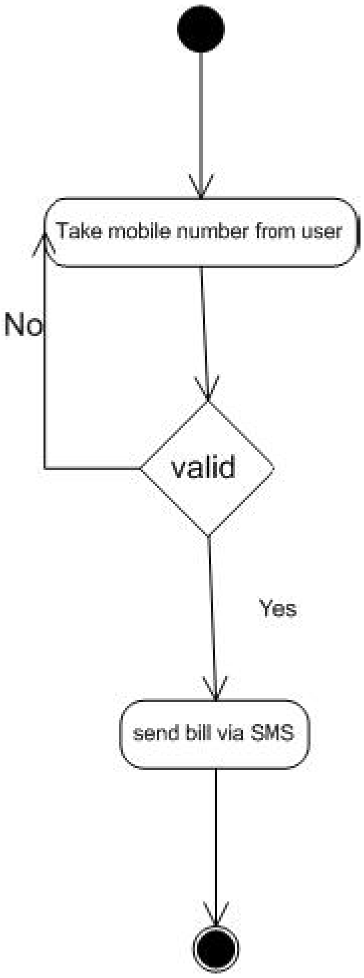


Figure – 21: Activity diagram for Message Sending

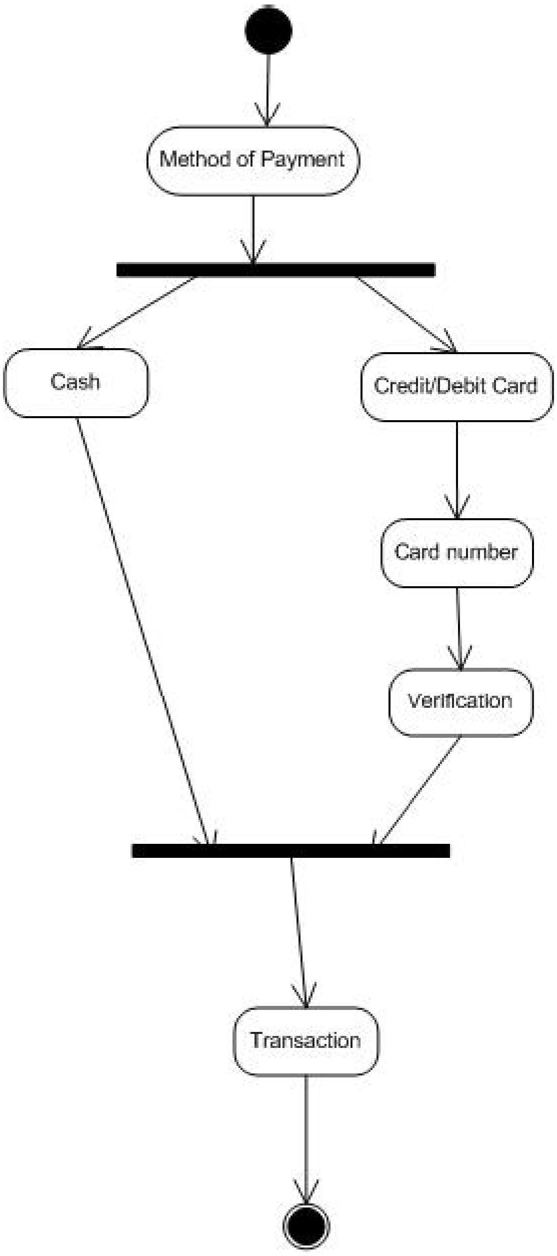


Figure – 22: Activity diagram for payment

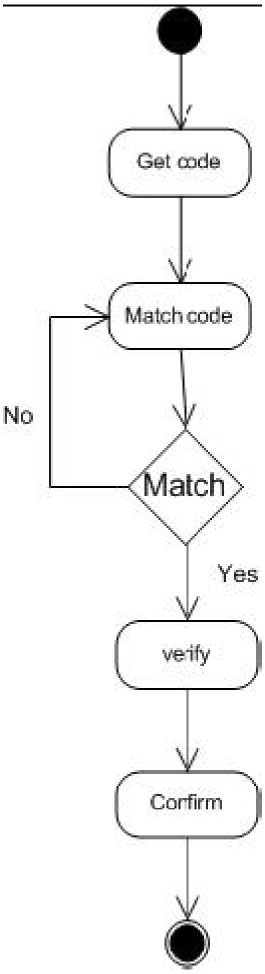


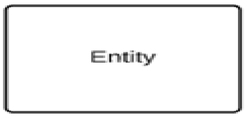
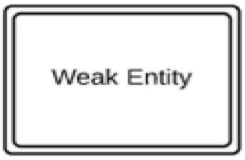
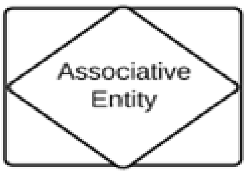
Figure – 23: Activity diagram for code matching

7.4 E-R Diagram

1. Use:

- It is used for graphical representation of entities and their relationship to each other.
- Typically used in computing in regard to the organization of data within in databases or information systems.
- It help us to visualize how data is connected in general way and are particularly useful for constructing a relational database.

2. Symbol:

Symbol	**Description**
	Entity: Entities are objects or concepts that represent important data. They are typically nouns, e.g. *Customer*, *supervisor*, *location*, or *promotion*. • Strong entities exist independently from other entity types. They always possess one or more attributes that uniquely distinguish each occurrence of the entity.
	• Weak entity: • Weak entities depend on some other entity type. They don't possess unique attributes (also known as a primary key) and have no meaning in the diagram without depending on another entity. This other entity is known as the owner.
	• Associative entity: • Associative entities are entities that associate the instances of one or more entity types. They also contain attributes that are unique to the relationship between those entity instances.

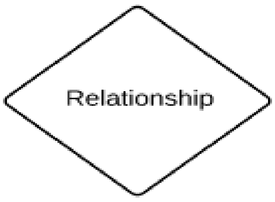
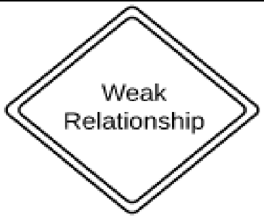
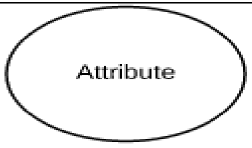
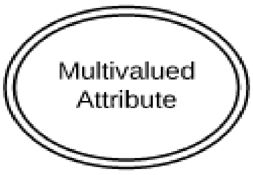
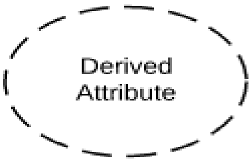
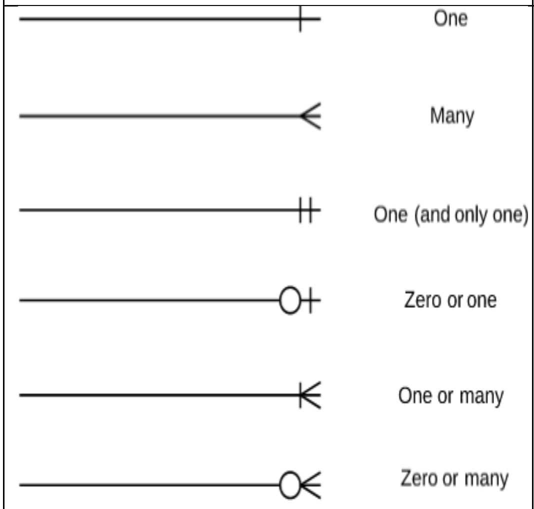
	Relationships: Relationships are meaningful associations between or among entities. They are usually verbs, e.g. *Assign*, *associate*, or *track*. A relationship provides useful information that could not be discerned with just the entity types.
	• Weak relationship: • Weak relationships, or identifying relationships, are connections that exist between a weak entity type and its owner.
	Attributes: Attributes are characteristics of either an entity, a many-to-many relationship, or a one-to-one relationship.
	Multivalued attribute: Multivalued attributes are those that are capable of taking on more than one value.
	1. Derived attributes : 2. Derived attributes are attributes whose value can be calculated from related attribute values.
 One Many One (and only one) Zero or one One or many Zero or many	• Cardinality: • Cardinality and ordinarily, respectively, refer to the maximum number of times an instance in one entity can be associated with instances in the related entity, and the minimum number of times an instance in one entity can be associated with an instance in the related entity. Cardinality and ordinarily are represented by the styling of a line and its endpoint, as denoted by the chosen notation style.

Table - 14: E-R symbols

3. Diagram:

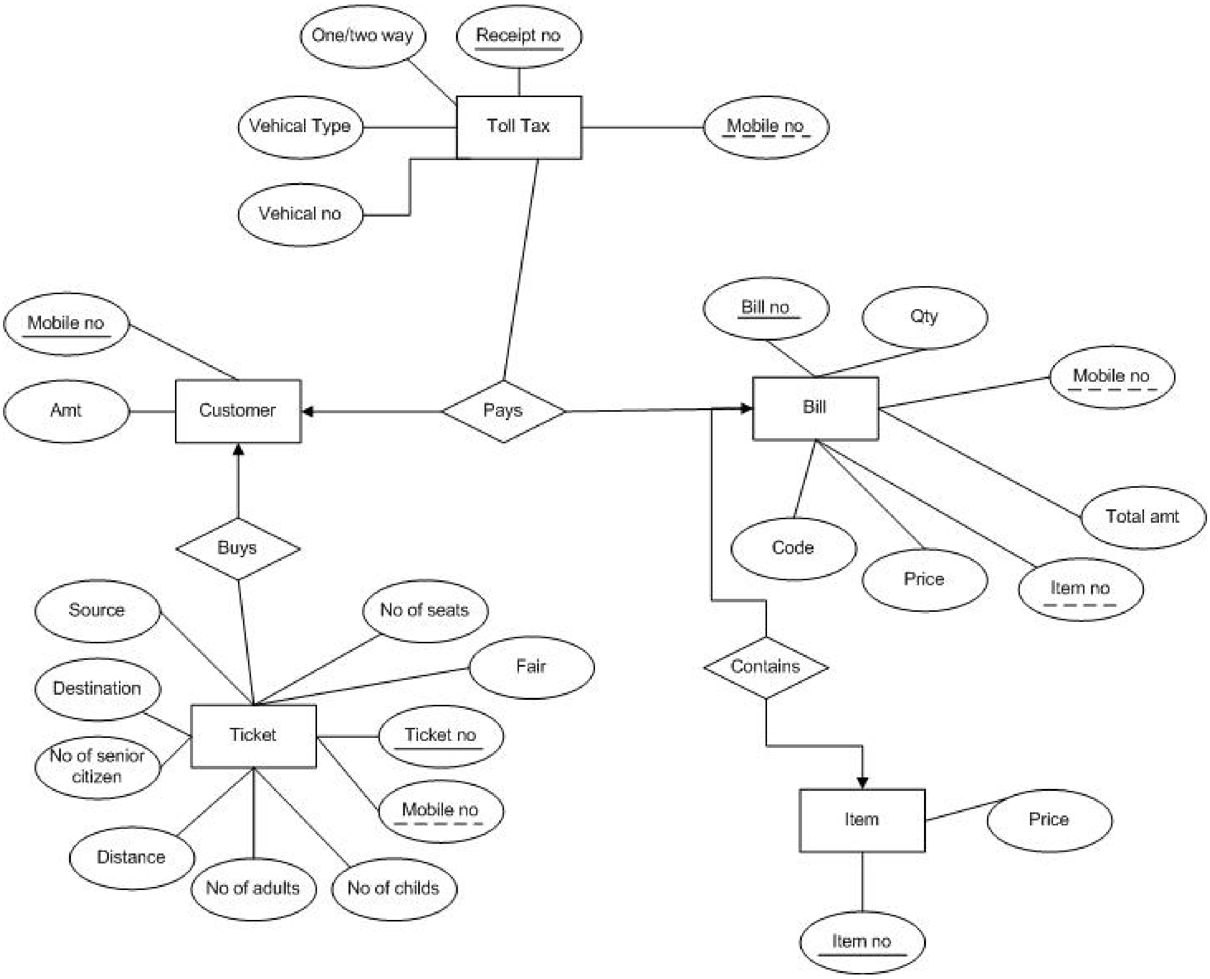


Figure - 24: E-R diagram

7.5 Data Flow Diagram

1. Use:

- It is used to identify the flow of information, where data comes from, where it goes and how it gets stored.
- Used to capture the dynamic aspect of a system.

2. Symbols:

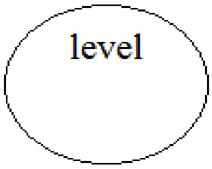
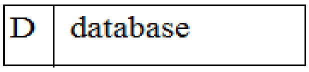
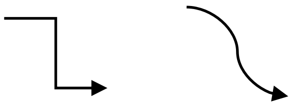
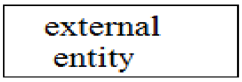
Symbols	Description
	Circle: A process transforms incoming Data flow into outgoing data flow.
	Datastore/database: Datastore/database are repositories of data in the system. They are sometimes also referred to us files.
	Dataflow: Dataflows are pipelines through which packets of information flow. Label the arrows with the name of the data that moves through it.
	External Entity: External entities are objects outside the system, with which the system communicates .External entities are sources and destinations of the system's inputs and outputs.

Table - 15: Dataflow Symbols

3. Diagram:

(1) Level 0 Dataflow Diagram:

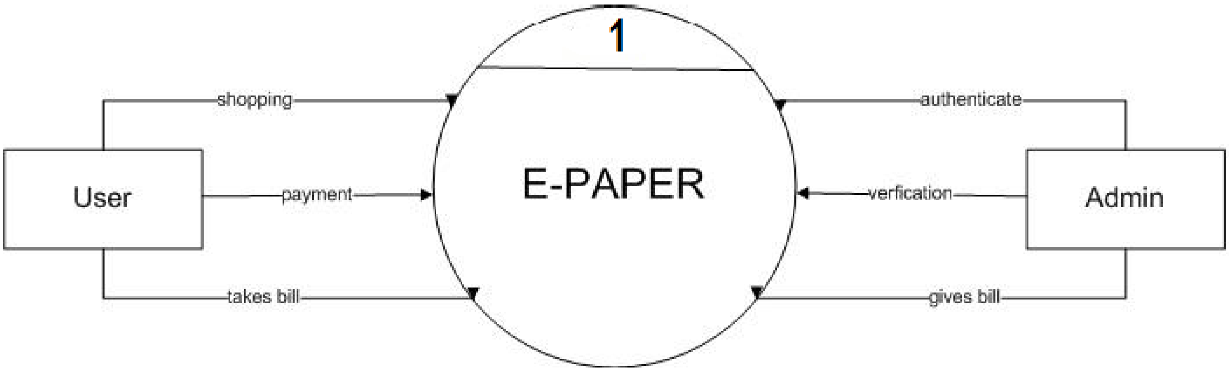


Figure - 25: Level 0 Dataflow diagram

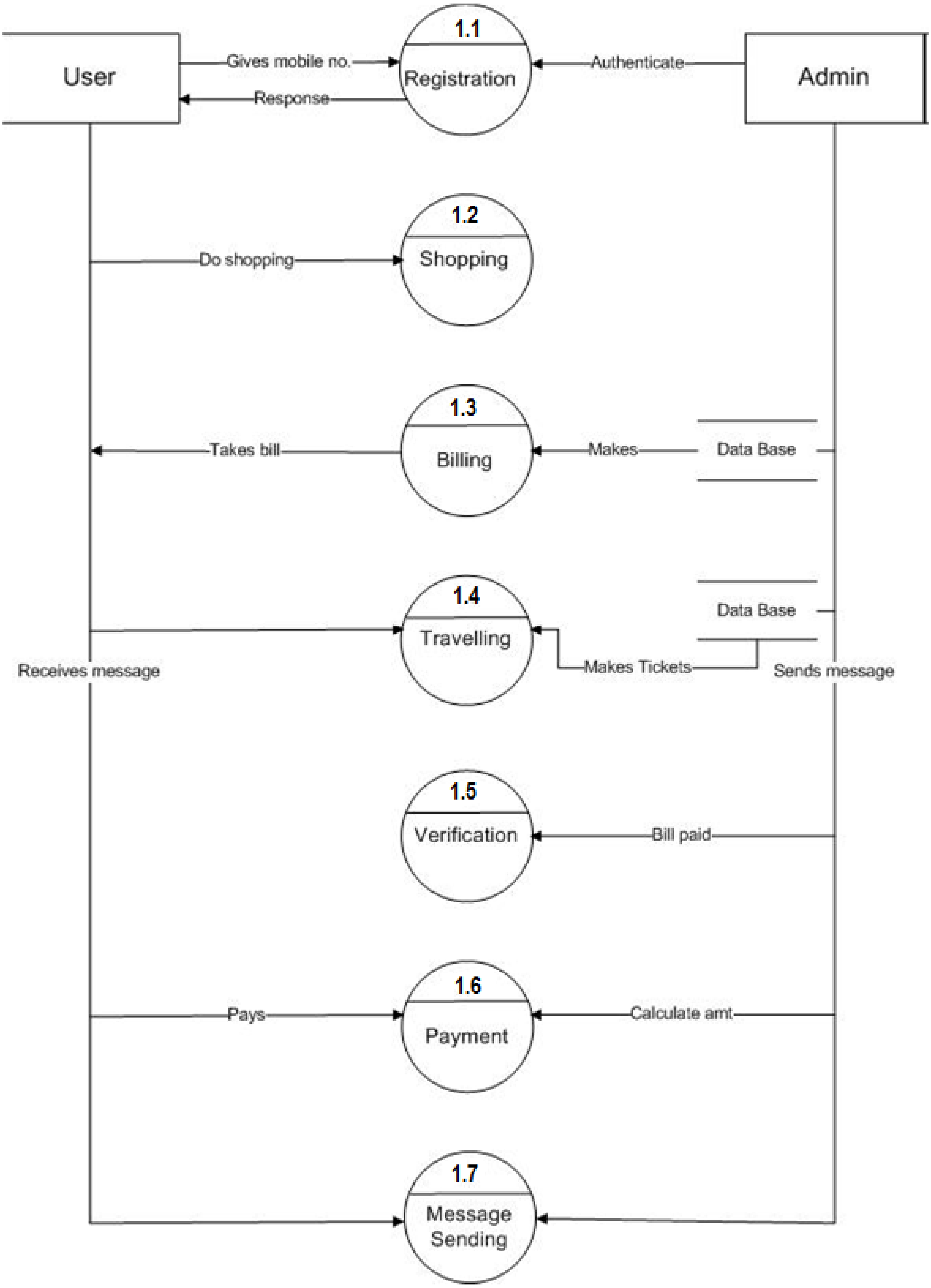


Figure - 26: Level 1 Dataflow diagram

(2) Level 2 DFD

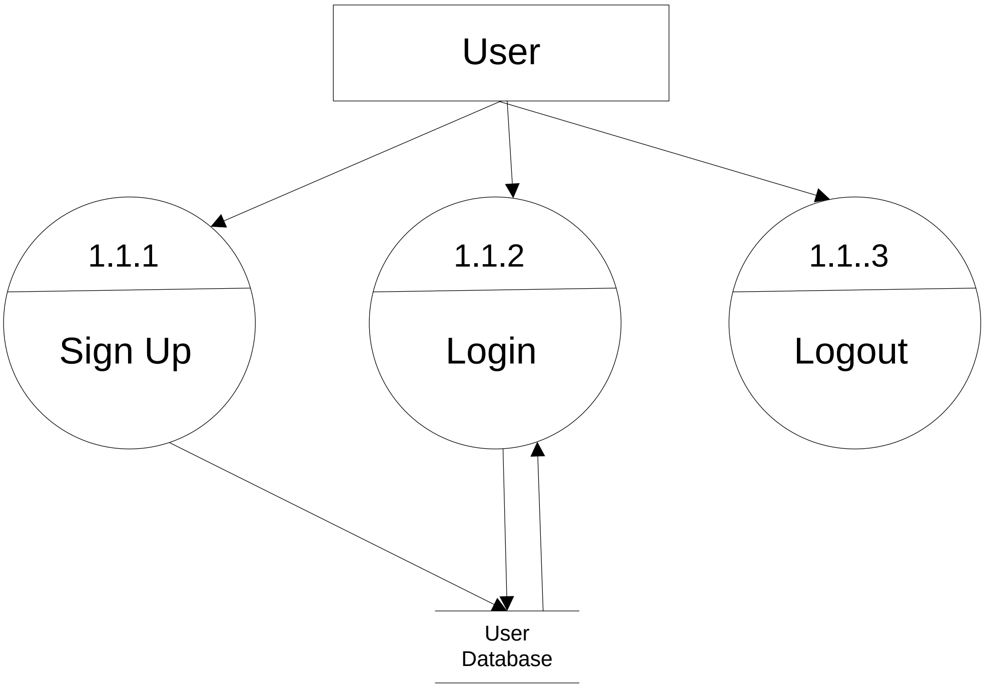


Figure - 27: Level 2 Dataflow diagram for registration

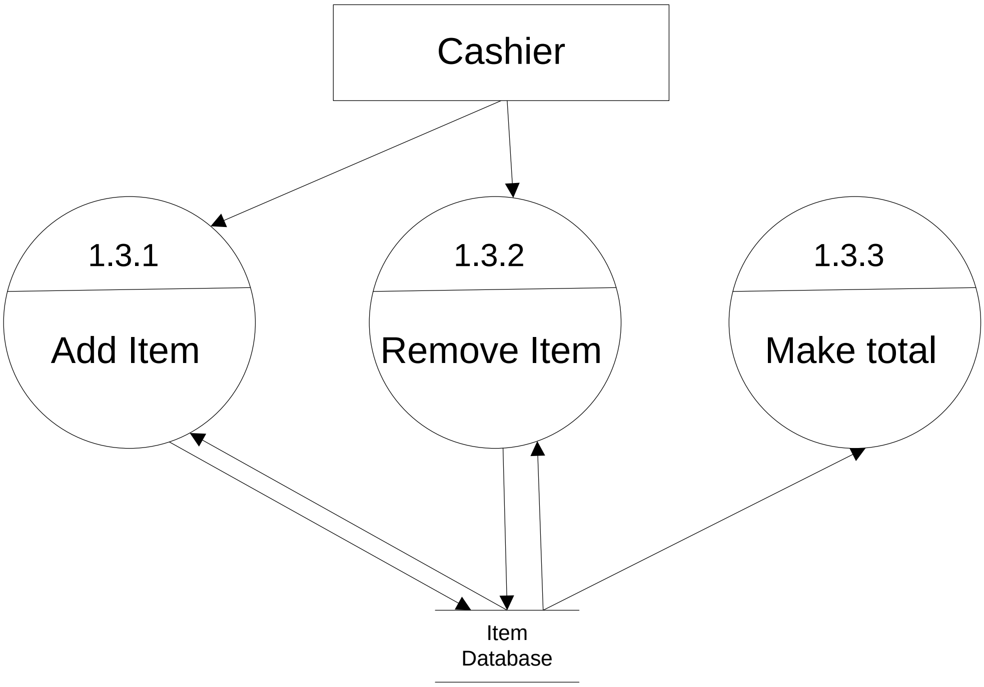


Figure - 28: Level 2 Dataflow diagram for billing

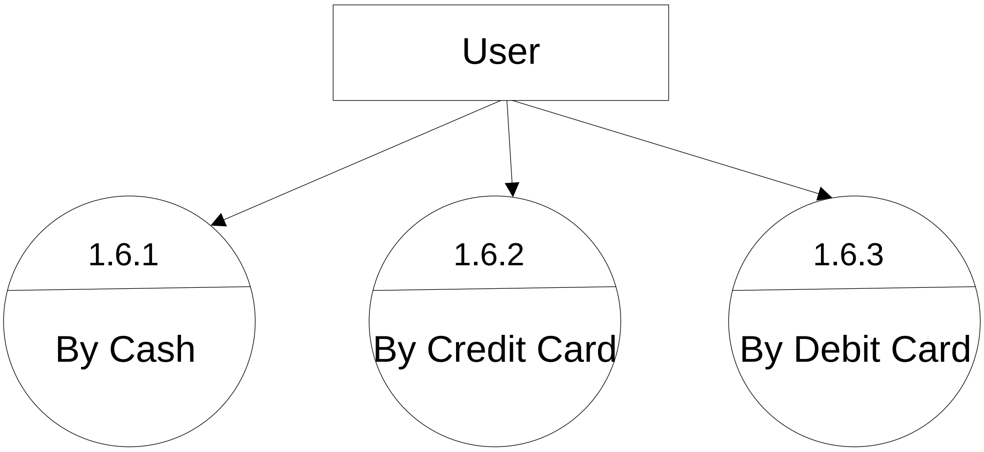


Figure - 29: Level 2 Dataflow diagram for payment

Chapter 8: Implementation

8.1 Home page: We can go to service page from the home using SERVICE tab.

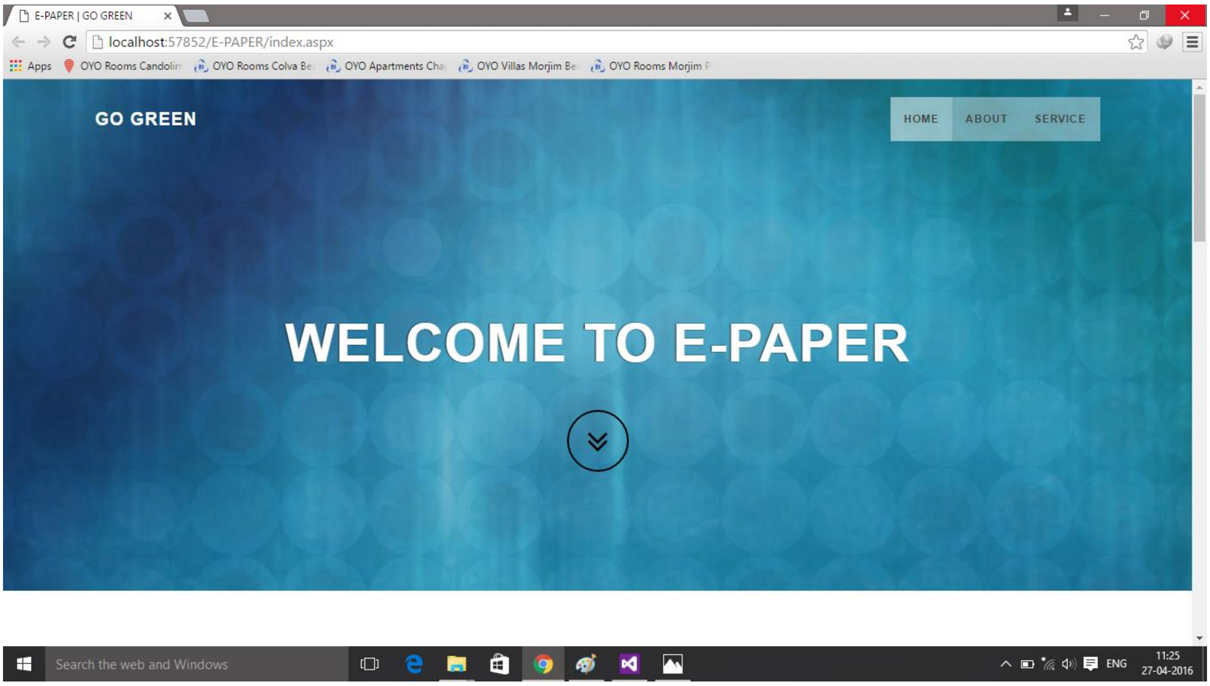
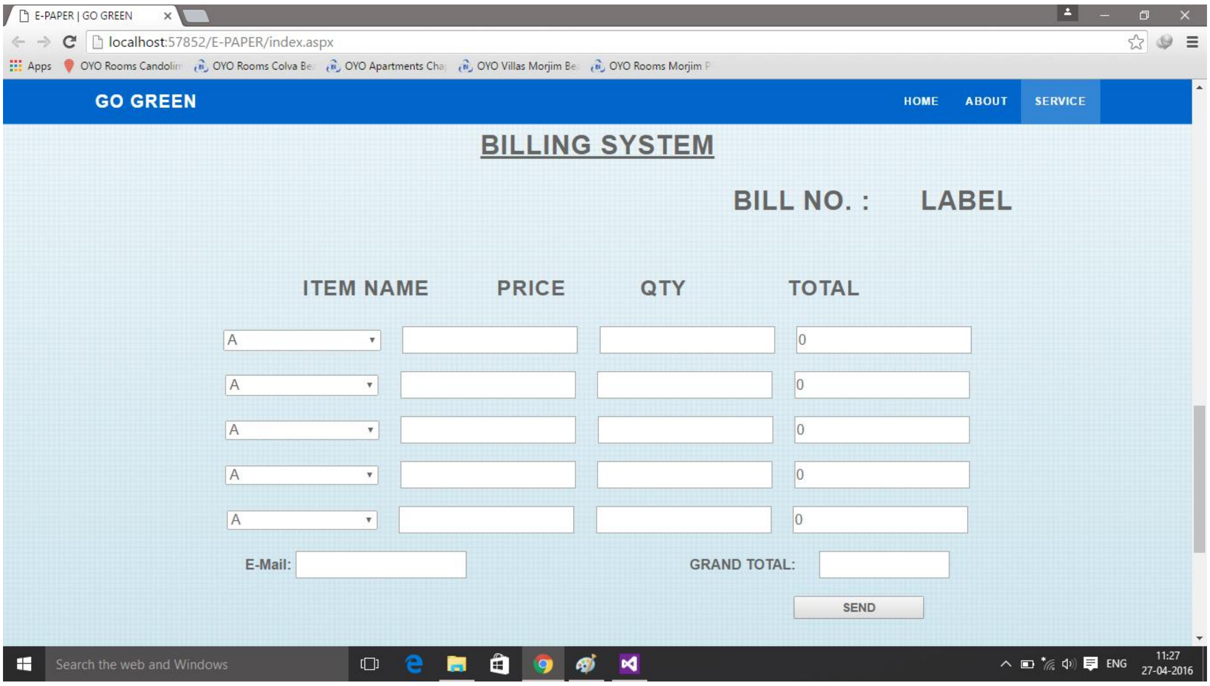


Figure - 30: HOME page

8.2 Service page: We can fill details of bill using given dropdown lists and textbox and can generate bill and also send it to the customer.



BILLING SYSTEM

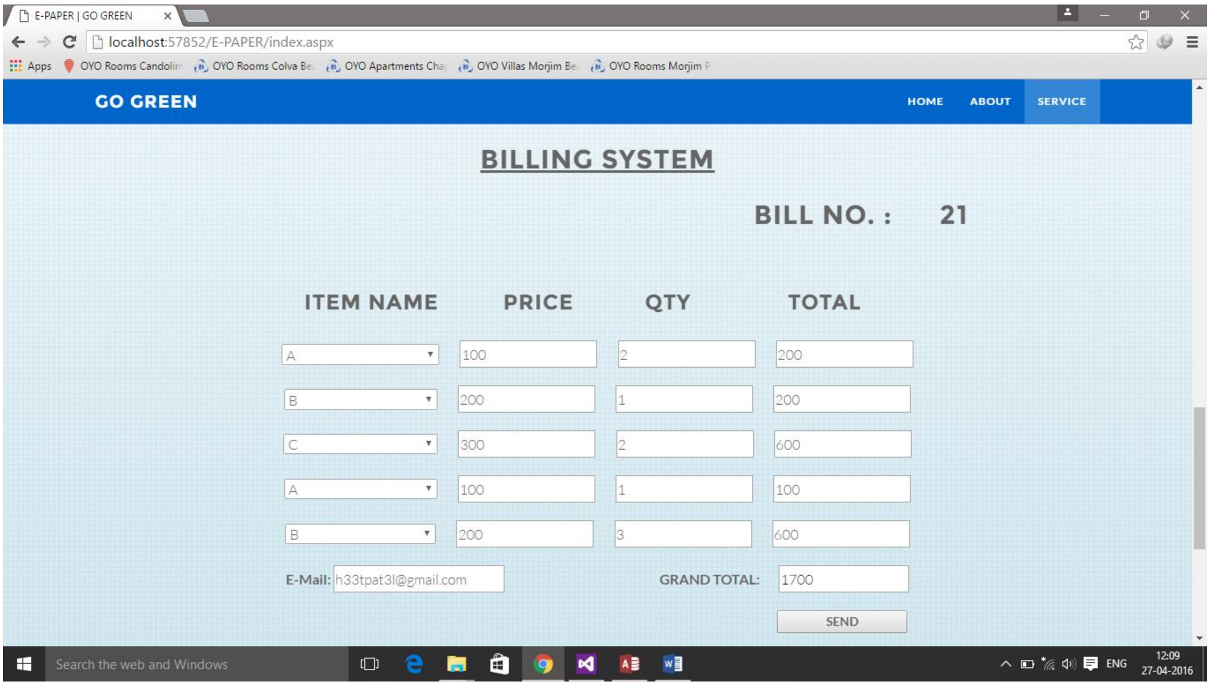
BILL NO. : LABEL

ITEM NAME	PRICE	QTY	TOTAL
A			0
A			0
A			0
A			0
A			0

E-Mail: GRAND TOTAL:

Figure - 31: SERVICE page

8.3 Service page : This is example of filling the details.



BILLING SYSTEM

BILL NO. : 21

ITEM NAME	PRICE	QTY	TOTAL
A	100	2	200
B	200	1	200
C	300	2	600
A	100	1	100
B	200	3	600

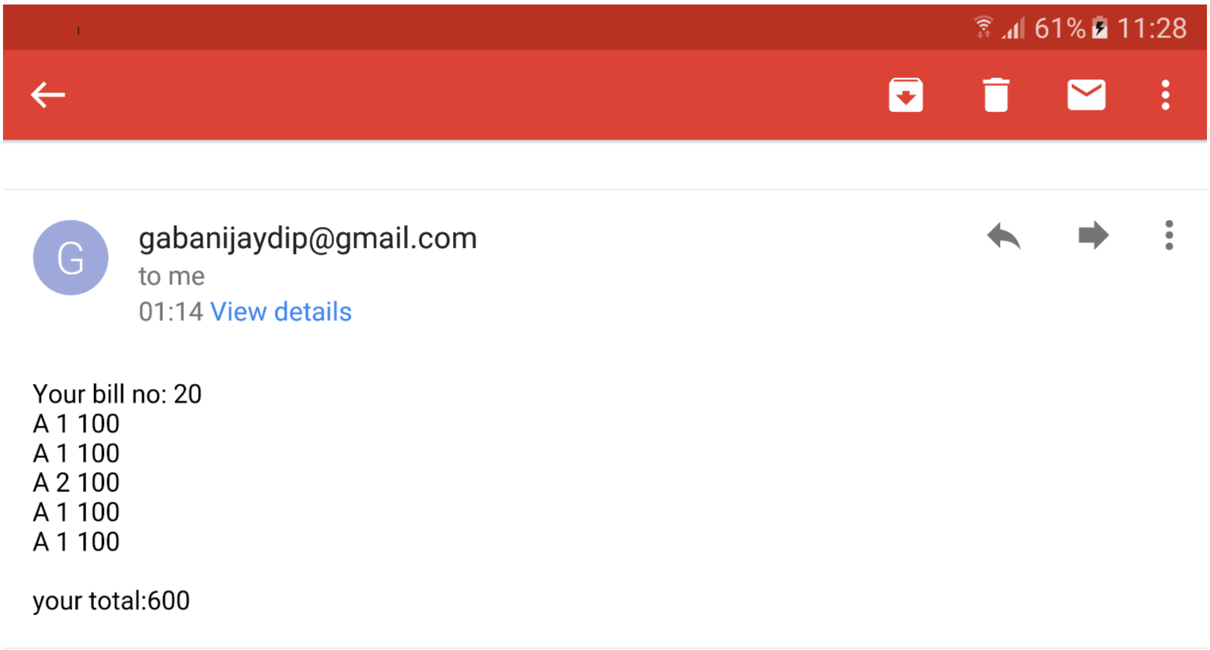
E-Mail: h33tpat3l@gmail.com

GRAND TOTAL: 1700

SEND

Figure - 32: SERVICE page

8.4 Mail



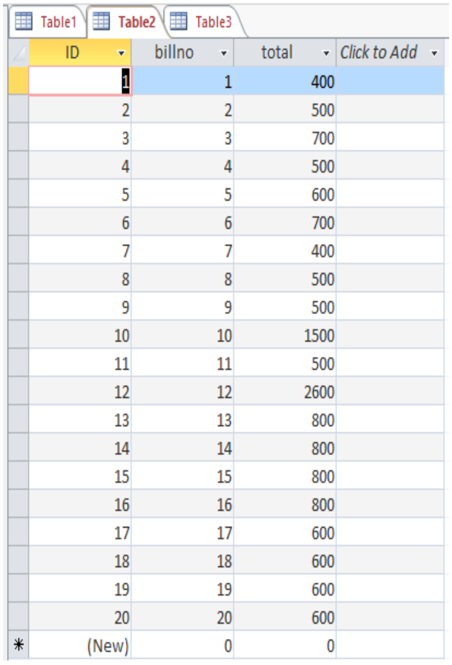
gabanijaydip@gmail.com
to me
01:14 View details

Your bill no: 20
A 1 100
A 1 100
A 2 100
A 1 100
A 1 100

your total:600

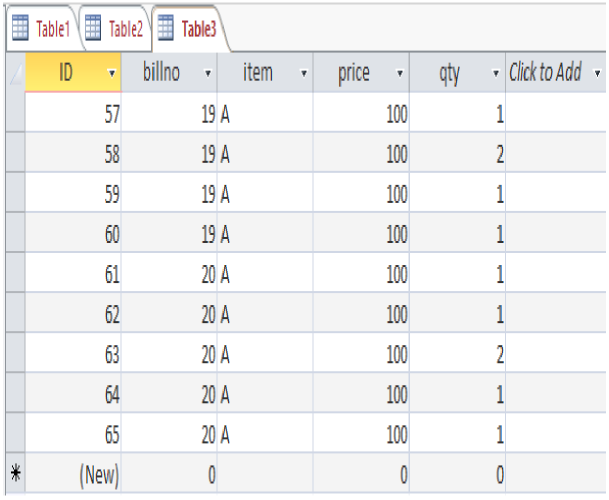
Figure - 33: E-MAIL

8.5 Database



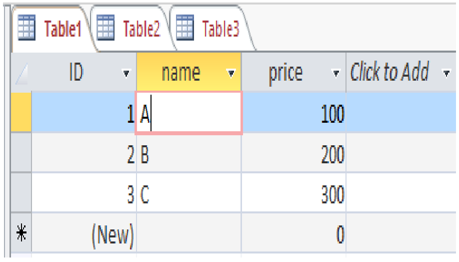
	ID	billno	total	Click to Add
	1	1	400	
	2	2	500	
	3	3	700	
	4	4	500	
	5	5	600	
	6	6	700	
	7	7	400	
	8	8	500	
	9	9	500	
	10	10	1500	
	11	11	500	
	12	12	2600	
	13	13	800	
	14	14	800	
	15	15	800	
	16	16	800	
	17	17	600	
	18	18	600	
	19	19	600	
	20	20	600	
*	(New)	0	0	

Figure - 34: Table 1



	ID	billno	item	price	qty	Click to Add
	57	19	A	100	1	
	58	19	A	100	2	
	59	19	A	100	1	
	60	19	A	100	1	
	61	20	A	100	1	
	62	20	A	100	1	
	63	20	A	100	2	
	64	20	A	100	1	
	65	20	A	100	1	
*	(New)	0		0	0	

Figure - 35: Table 2



	ID	name	price	Click to Add
	1	A	100	
	2	B	200	
	3	C	300	
*	(New)		0	

Figure - 36: Table 3

Conclusion and Future Work:

In this semester we’ve have made website page that can generate the bill for any shop. We can send e-mail containing that bill using this software. We have stored this bill for future reference of admin or customer.

We would like to add this same functions for ticket booking and receipts and any area that contains use of paper.