



FPT ACADEMY INTERNATIONAL

FPT – APTECH COMPUTER EDUCATION

Centre Name: ACE-HCMC-2-FPT

Address: 590 Cach Mang Thang 8, District 3, Ho Chi Minh City, Viet Nam.

# TARS DELIVERY

## eProject Report Semester 3

<b>Supervisor</b>	Mr. Tran Phuoc Sinh	
<b>Batch</b>	T1.1910.M3	
<b>Group</b>	3	
<b>Serial No.</b>	<b>Student Name</b>	<b>Enrollment Number</b>
1.	Pham Quang Huy	Student1225636
2.	Pham Xuan Hoai	Student1225564
3.	Le Hoang Anh Tu	Student1225638
4.	Ngo Trung Kien	Student1228722
5.	Vo Cuong Nhut	Student1225635

**JANUARY-2021**

## Contents

<b>REVIEW 1 .....</b>	<b>5</b>
<b>INTRODUCTION.....</b>	<b>1</b>
<b>For TARS Delivery's users: .....</b>	<b>1</b>
<b>Existing scenario .....</b>	<b>2</b>
<b>PROBLEM DEFINITION.....</b>	<b>3</b>
<b>From the view point of Users:.....</b>	<b>3</b>
<b>From the view point of Registered Users:.....</b>	<b>3</b>
<b>From the view point of Admin:.....</b>	<b>3</b>
<b>From the view point of Employee:.....</b>	<b>3</b>
<b>    Functional Requirement Specifications .....</b>	<b>4</b>
<b>    Function of User:.....</b>	<b>4</b>
<b>    Function of Admin:.....</b>	<b>4</b>
<b>    Non-Functional Requirement Specifications: .....</b>	<b>5</b>
<b>        System requirement specifications .....</b>	<b>6</b>
<b>    Server requirements: .....</b>	<b>6</b>
<b>    Client Requirement.....</b>	<b>7</b>
<b>REVIEW 2 .....</b>	<b>1</b>
<b>I.       ARCHITECTURE AND DESIGN OF THE PROJECT .....</b>	<b>1</b>
<b>II.      ALGORITHM AND DATA FLOWCHART.....</b>	<b>1</b>
<b>2.      Main data flowchart for user .....</b>	<b>2</b>
<b>3.      Main data flowchart for admin.....</b>	<b>3</b>
<b>4.      Login process (For admins and users) .....</b>	<b>4</b>
<b>5.      Logout process (For admins and users) .....</b>	<b>4</b>
<b>6.      Insert User (For User Only) .....</b>	<b>4</b>
<b>7.      Update user's information (For User only).....</b>	<b>5</b>
<b>8.      Insert Comment (for user only) .....</b>	<b>5</b>
<b>9.      Delete User(For Admin Only).....</b>	<b>5</b>

<b>10.</b>	<b>Reset Password Admin(for Admin only) .....</b>	<b>6</b>
<b>11.</b>	<b>Chang Password User(for User only) .....</b>	<b>6</b>
<b>12.</b>	<b>Insert, Update, Delete Branch(for Admin only).....</b>	<b>6</b>
<b>13.</b>	<b>Insert, Update, Delete Employees(for Admin only) .....</b>	<b>6</b>
<b>14.</b>	<b>Update, Delete Comment(for Admin only) .....</b>	<b>7</b>
<b>III. Data Flow Diagram(DFD) .....</b>		<b>8</b>
<b>1.</b>	<b>Context Level Diagram .....</b>	<b>8</b>
<b>2.</b>	<b>Context DFD.....</b>	<b>9</b>
<b>1.</b>	<b><i>Login.....</i></b>	<b>9</b>
<b>2.</b>	<b><i>Customer And Accounts information.....</i></b>	<b>10</b>
<b>3.</b>	<b><i>Admin manage Admin .....</i></b>	<b>11</b>
<b>4.</b>	<b><i>Admin manage Branch.....</i></b>	<b>11</b>
<b>5.</b>	<b><i>Admin manage Employee .....</i></b>	<b>11</b>
<b>6.</b>	<b><i>Admin manage Bill .....</i></b>	<b>12</b>
<b>7.</b>	<b><i>Admin manage Comment .....</i></b>	<b>13</b>
<b>8.</b>	<b><i>User manage User.....</i></b>	<b>13</b>
<b>9.</b>	<b><i>User manage Feedback.....</i></b>	<b>14</b>
<b>IV. Use Case Diagram .....</b>		<b>14</b>
<b>1.</b>	<b>Admin use case.....</b>	<b>15</b>
<b>2.</b>	<b>User use diagram.....</b>	<b>15</b>
<b>3.</b>	<b>Guest user diagram .....</b>	<b>16</b>
<b>V. Entity – Relationship Diagrams.....</b>		<b>17</b>
<b>1.</b>	<b>Entities and Properties.....</b>	<b>17</b>
<b>Table Design.....</b>		<b>20</b>
<b>Relation Diagram.....</b>		<b>23</b>
<b>TASK SHEET REVIEW 2.....</b>		<b>24</b>
<b>REVIEW 3 .....</b>		<b>27</b>
<b>User: .....</b>		<b>28</b>
<b>1.Index Page .....</b>		<b>28</b>

<b>2.Services</b> .....	32
<b>3. Team</b> .....	35
<b>4.Contact Us</b> .....	36
<b>5.Login</b> .....	38
<b>6. Forgot password :</b> .....	40
<b>7. Sign up :</b> .....	43
<b>8.Bill History:</b> .....	46
<b>Admin :</b> .....	47
<b>1.Login</b> .....	47
<b>2.Dashboard:</b> .....	50
<b>3.Account</b> .....	52
<b>4.Details Account:</b> .....	54
<b>5.Create Account:</b> .....	56
<b>6.Bill:</b> .....	58
<b>7.Details Bill:</b> .....	60
<b>8.Update Bill:</b> .....	62
<b>9.Create Bill:</b> .....	64
<b>10.Branch:</b> .....	65
<b>11.Edit Branch:</b> .....	67
<b>12.PriceList:</b> .....	69
<b>13.Edit PriceList:</b> .....	71
<b>14.Create PriceList:</b> .....	73
<b>15.Comment:</b> .....	74
● <a href="#">SiteMap</a> .....	75
● <a href="#">API Swagger:</a> .....	77
<a href="#">Task Sheet REVIEW 3</a> .....	80

# REVIEW 1

Design Plan:TARS Delivery	Document Name:Review 1	SWD/Form No.1
Effective Date:22/01/2021	Version:1.0	Page No:1

## INTRODUCTION

TARS Delivery System is a postal organization with different branches at different cities delivering the mails through various options like courier, speed post, Normal post, VPP etc... and money orders. They used to charge for the delivery of the mails based on.

### For TARS Delivery's users:

- Provides a simple, friendly interface on the website platform to help users easily access it.
- Users can see pricelist and all services.
- Users can create accounts, log in and update account.
- Users can search status of their bill by bill code.
- Users can see all their bills by history order.
- Users can leave their comments and contact with admin.

	Prepared by (Student)	Approved by (Faculty)
	Project Group No:1	
Signature		
Date		

Design Plan:TARS Delivery	Document Name:Review 1	SWD/Form No.1
Effective Date:22/01/2021	Version:1.0	Page No:2

### Existing scenario

Currently they are using a stand alone application that holds the details of the mails like the delivery number, date of posting, the weight, type of delivery chosen, Destination address, the address of the person who has posted it, contact details, etc... As soon as the packet or the letter is received, the date of receipt at the destination place office and the date of delivery to the destination address are captured. Some of them (companies, etc...) send multiple envelopes, or couriers, etc....

Now they want an online application to be incorporated in the system so that the customer can also check from his end when the packet or the letter is delivered, as most of the packets or letters that are posted are mainly for the business purpose like tender application forms, etc..... Also the department people instead of maintaining separate applications each at a location they can use the same application (online) that performs all the tasks of the application that they are using.

The application which they are using right now, also calculates the charges for the delivery based on the weight, distance, and the type of the service chosen by the customer. The application that they are currently using also holds the information of the charges, Pin code details, and the contact details include based on the location. The Organization head alone has the right to change the charges for the delivery.

Even these concepts are needed to be incorporated in the online application.

	Prepared by (Student)	Approved by (Faculty)
	Project Group No:1	
Signature		
Date		

Design Plan:TARS Delivery	Document Name:Review 1	SWD/Form No.1
Effective Date:22/01/2021	Version:1.0	Page No:3

## PROBLEM DEFINITION

After selecting the topic for the group, we analyzed the structure and problems presented by the project. The issues that need to be resolved in the user interface and admin after being aggregated include:

### **From the view point of Users:**

- For Register page: The customer can register with the site provided he/she makes the payment . Customers can see all history order.
- Can see free recipe available, about TARS Delivery, Price list , contact and feedback to shop.
- User can provide their own new recipe if it is selected then the user will get the prize money decided by the admin as well as the participation certificate from the shop. The name of the candidate will be enlisted in the Home Page of the Website in one section under the top recipe column.

### **From the view point of Registered Users:**

- Link will be enabled with all the previous function mentioned.
- Customers can see status of bill by bill code.
- Users get registered he/she must be allowed to login with their email and password.

### **From the view point of Admin:**

- Admin can login logout admin pages and access the details of the deliverable where he can enter, update the details and as well the details of service charges, contact address based on the location, and the pin codes of the locations.
- Admin can create account for employees
- Manage dashboard, branch, account, comment.

### **From the view point of Employee:**

- Employees be able to manage bill where they are working.

	Prepared by (Student)	Approved by (Faculty)
	Project Group No:1	
Signature		
Date		

Design Plan:TARS Delivery	Document Name:Review 1	SWD/Form No.1
Effective Date:22/01/2021	Version:1.0	Page No:4

## Functional Requirement Specifications

### **Function of User:**

#### **Un-Registered Users:**

- REQ-01. Register new account.
- REQ-02. Use service send VPP service.
- REQ-03. See status of bill by bill code.
- REQ-04. View about TARS Delivery.
- REQ-05:View about pricelist of services.
- REQ-06. Send feedback.

#### **Registered Users:**

- REQ-07. All function like Un-Registered users.
- REQ-08. Login, logout in the website.
- REQ-09. Users can see your history order.
- REQ-10. Change account detail.

### **Function of Admin:**

- REQ-11. Login, logout, change password.
- REQ-12. View dashboard and export report.
- REQ-13. Manage account: see, create new, update, delete.
- REQ-14. Manage branch: see, create new, update, delete.
- REQ-15. Manage bill: see, create new, update, delete.
- REQ-18. Manage comment: see, delete.

	Prepared by (Student)	Approved by (Faculty)
	Project Group No:1	
Signature		
Date		

Design Plan:TARS Delivery	Document Name:Review 1	SWD/Form No.1
Effective Date:22/01/2021	Version:1.0	Page No:5

### **Non-Functional Requirement Specifications:**

- REQ-21. Good User Interface, User Experience.
- REQ-22. Clean code, project.
- REQ-23. Try hard improve the performance.
- REQ-24. Try this website run on all browsers.
- REQ-25. Responsive this website on mobile, tablet, desktop.
- REQ-26. Try limit bugs on this website.

	Prepared by (Student)	Approved by (Faculty)
	Project Group No:1	
Signature		
Date		

Design Plan:TARS Delivery	Document Name:Review 1	SWD/Form No.1
Effective Date:22/01/2021	Version:1.0	Page No:6

## System requirement specifications

### Server requirements:

#### HardWare:

Component	Requirement
<b>CPU</b>	Processor type: Pentium IV-compatible processor or faster Processor speed: Recommended: 2.0 GHz or faster
<b>OS</b>	Microsoft Windows 7 or higher with IIS
<b>Memory (RAM)</b>	RAM: Minimum: 2 GB Recommended: 4 GB or more
<b>Hard Drive</b>	Free space: Minimum: 500 MB Recommended: 50 GB or more Maximum: Operating system maximum

#### SoftWare:

Component	Requirement
<b>ASP.Net Core</b>	Version .Net Core 3.1
<b>RDBMS</b>	SQL Server 2008 or higher

	Prepared by (Student)	Approved by (Faculty)
	Project Group No:1	
<b>Signature</b>		
<b>Date</b>		

Design Plan:TARS Delivery	Document Name:Review 1	SWD/Form No.1
Effective Date:22/01/2021	Version:1.0	Page No:7

## Client Requirement

### HardWare:

Component	Requirement
CPU	Processor type: Pentium III-compatible processor or faster Processor speed: Recommended: 2.0 GHz or faster
OS	All OS(Window ,Linux ,Android ,Mac OS ...)
Memory (RAM)	RAM: Minimum: 2 GB Recommended: 4 GB or more Maximum: Operating system maximum
Hard Drive	Free space: Minimum: 1 GB

### SoftWare:

Component	Requirement
Web Browser	Firefox, Chrome or Higher.....
ASP.Net Core	Version .Net Core 3.1

### Development Software:

- SQL Server 2008 or higher.
- Microsoft Visual Studio 2019.

### Technology:

- ASP.Net core 3.1
- Entity Framework Core 3.1.10

	Prepared by (Student)  Project Group No:1	Approved by (Faculty)
Signature		
Date		

# REVIEW 2

Design Plan:TARS Delivery	Document Name:Review 2	SWD/Form No.1
Effective Date:22/01/2021	Version:1.0	Page No:1

## I. ARCHITECTURE AND DESIGN OF THE PROJECT

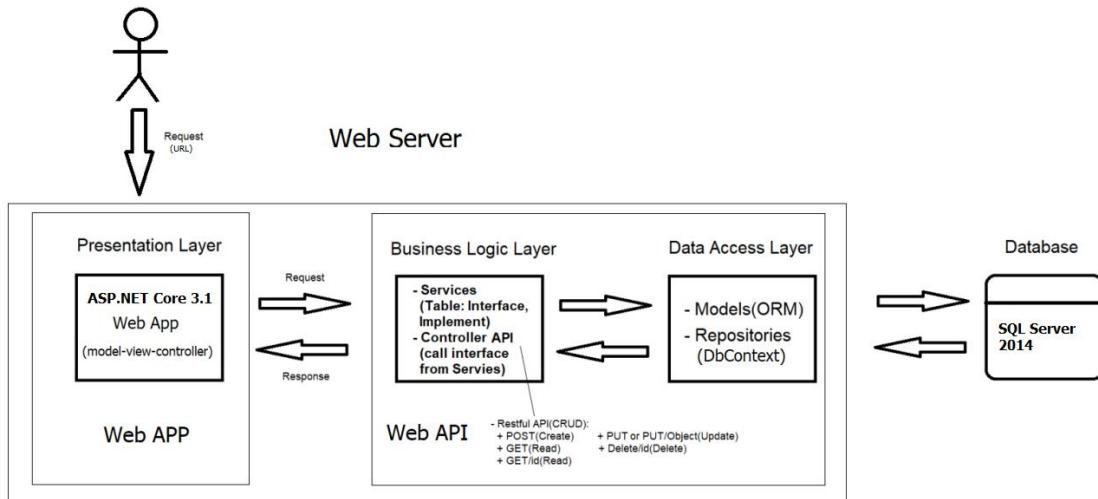


Figure 1: Architecture and Design of the Project

## II. ALGORITHM AND DATA FLOWCHART

### 1. Symbol generates:

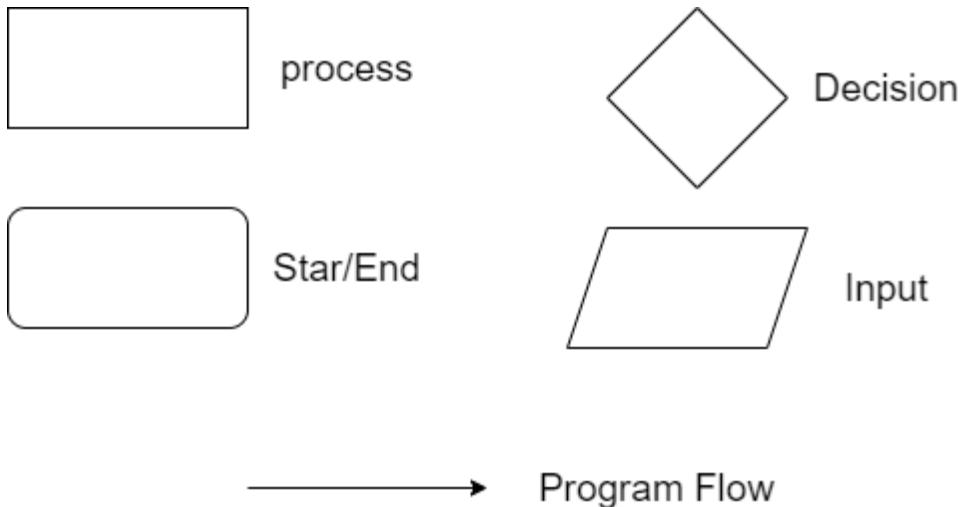
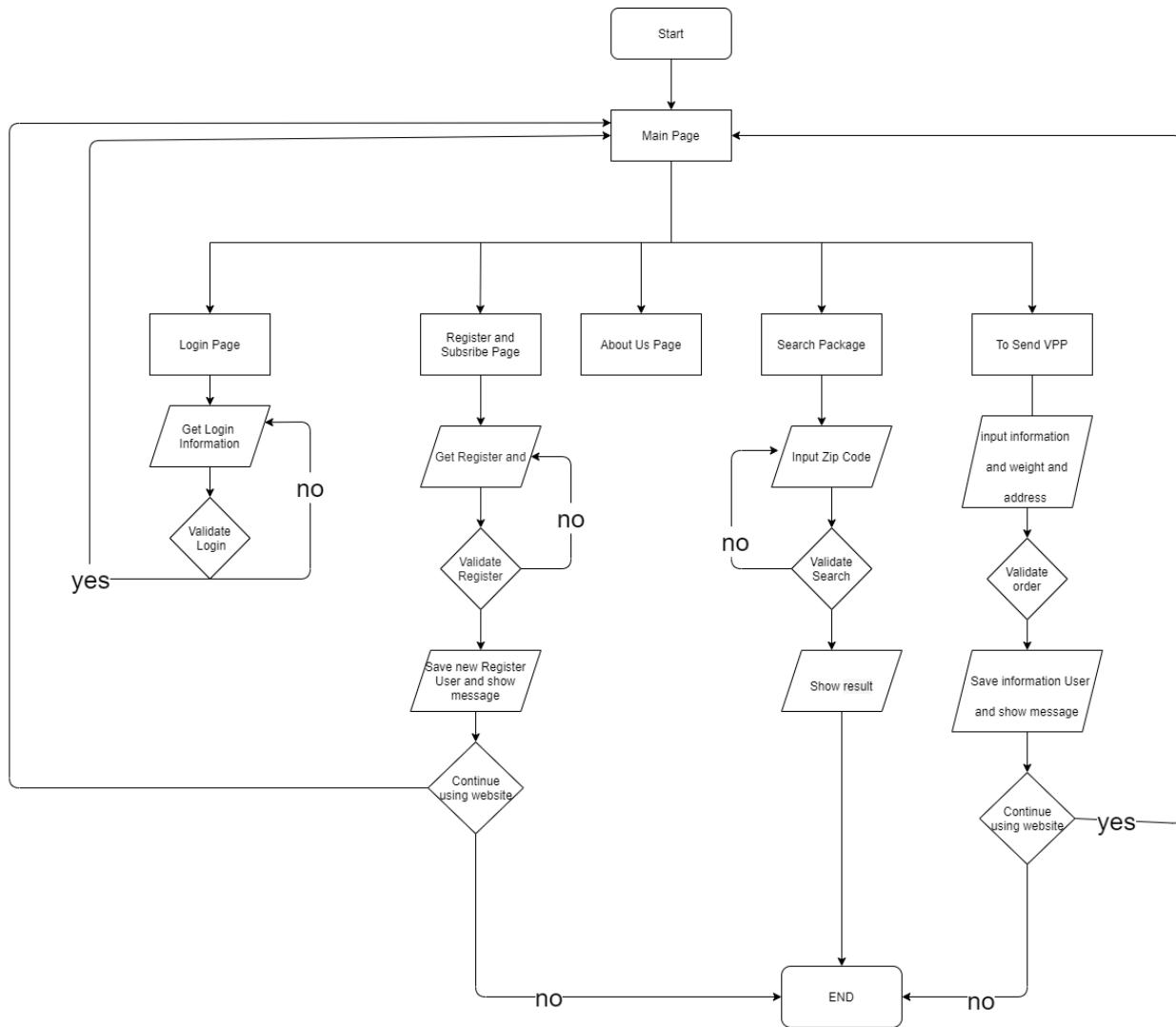


Figure 2: Symbol used in flowchart.

Prepared by (Student)	Approved by (Faculty)
Project Group No:3	
Signature	
Date	

Design Plan:TARS Delivery	Document Name:Review 2	SWD/Form No.1
Effective Date:22/01/2021	Version:1.0	Page No:2

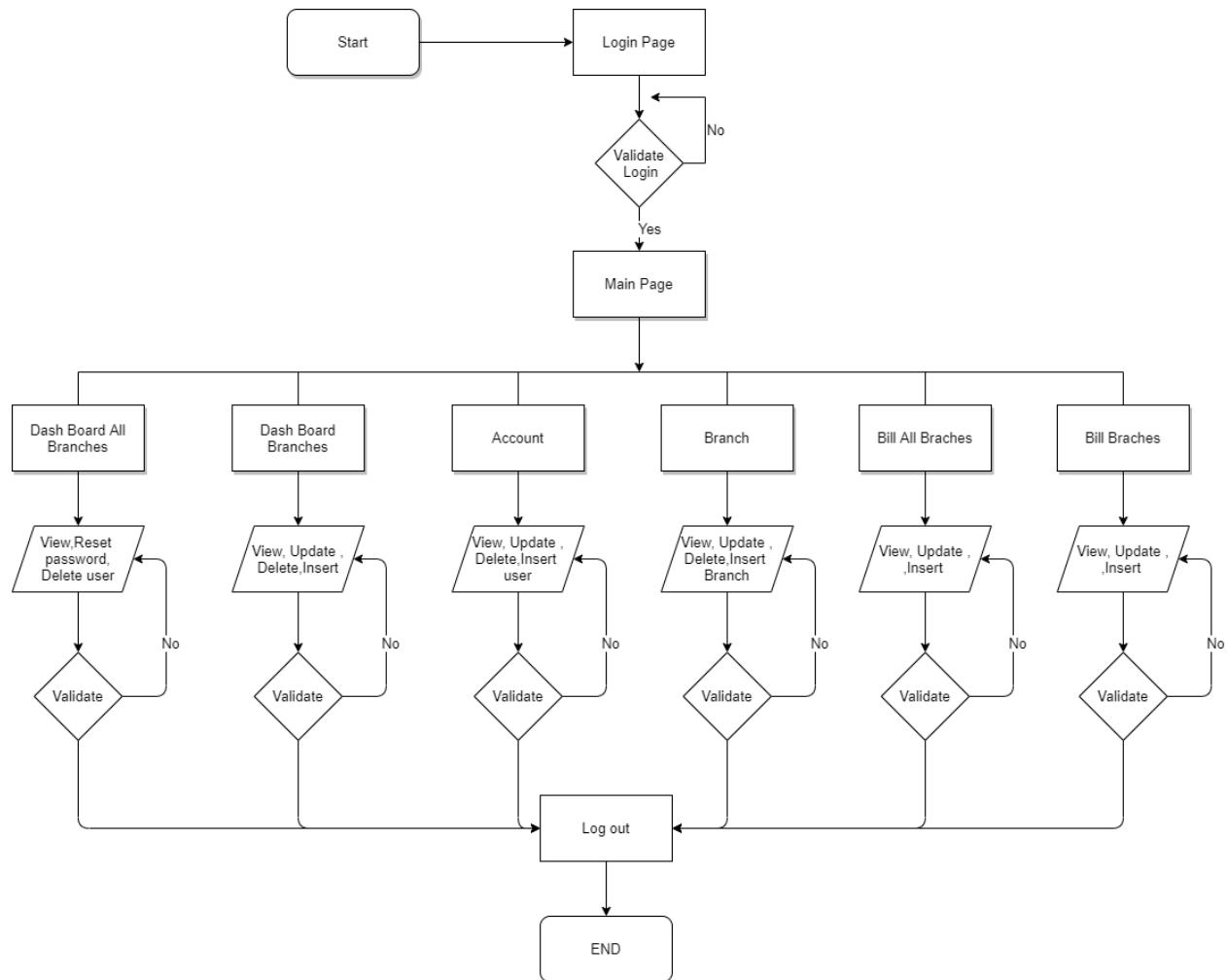
## 2. Main data flowchart for user



	Prepared by (Student) Project Group No:3	Approved by (Faculty)
Signature		
Date		

Design Plan:TARS Delivery	Document Name:Review 2	SWD/Form No.1
Effective Date:22/01/2021	Version:1.0	Page No:3

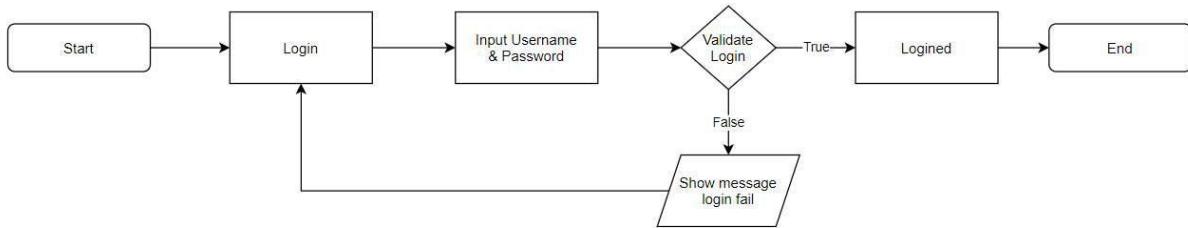
### 3. Main data flowchart for admin



	Prepared by (Student) Project Group No:3	Approved by (Faculty)
Signature		
Date		

Design Plan:TARS Delivery	Document Name:Review 2	SWD/Form No.1
Effective Date:22/01/2021	Version:1.0	Page No:4

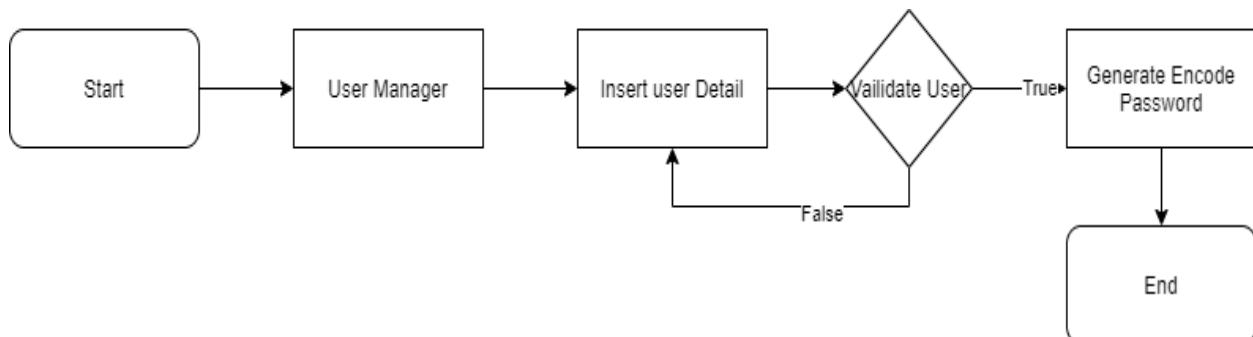
#### 4. Login process (For admins and users)



#### 5. Logout process (For admins and users)



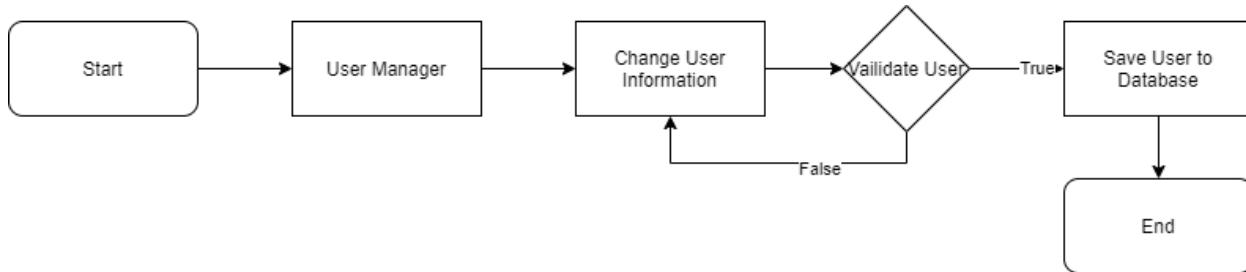
#### 6. Insert User (For User Only)



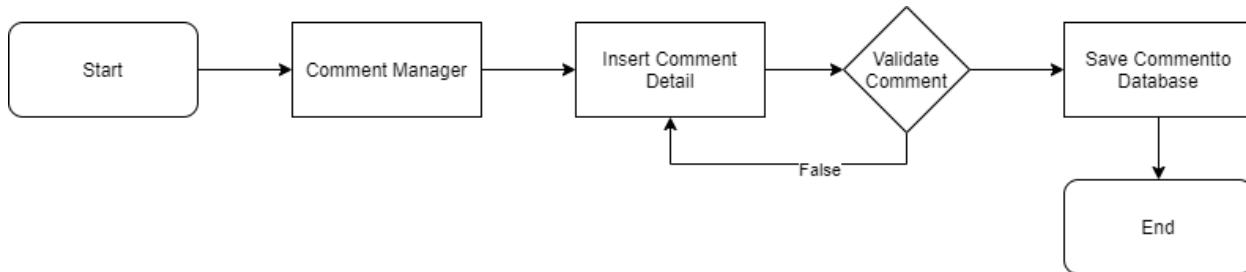
	Prepared by (Student) Project Group No:3	Approved by (Faculty)
Signature		
Date		

Design Plan:TARS Delivery	Document Name:Review 2	SWD/Form No.1
Effective Date:22/01/2021	Version:1.0	Page No:5

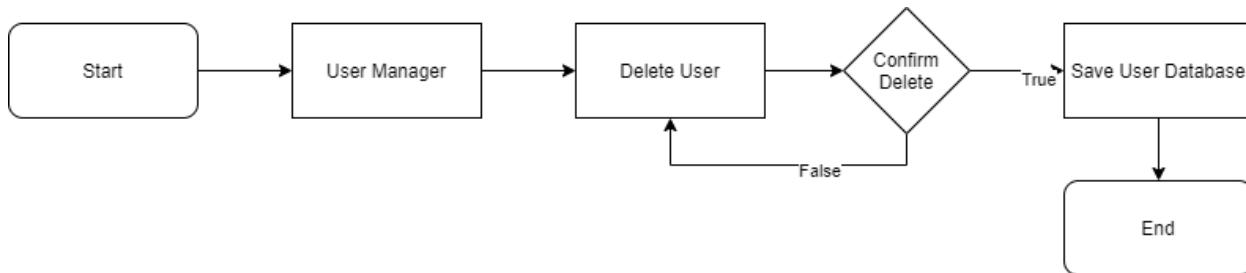
## 7. Update user's information (For User only)



## 8. Insert Comment (for user only)



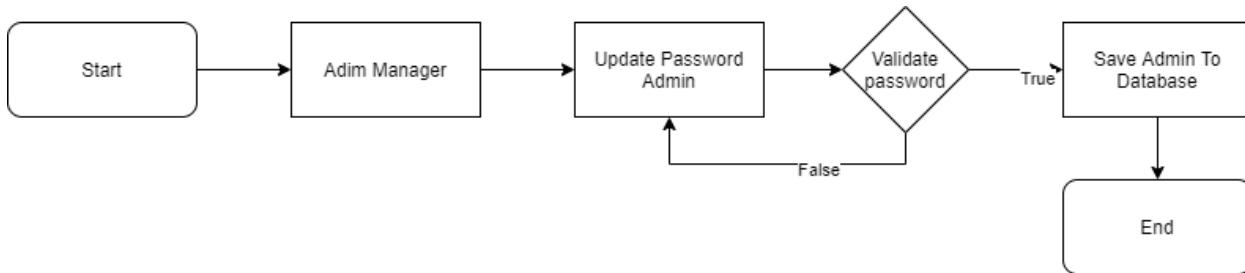
## 9. Delete User(For Admin Only)



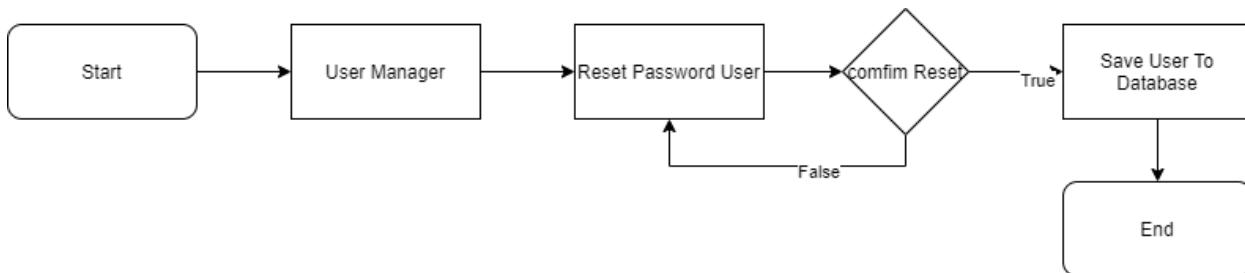
	Prepared by (Student) Project Group No:3	Approved by (Faculty)
Signature		
Date		

Design Plan:TARS Delivery	Document Name:Review 2	SWD/Form No.1
Effective Date:22/01/2021	Version:1.0	Page No:6

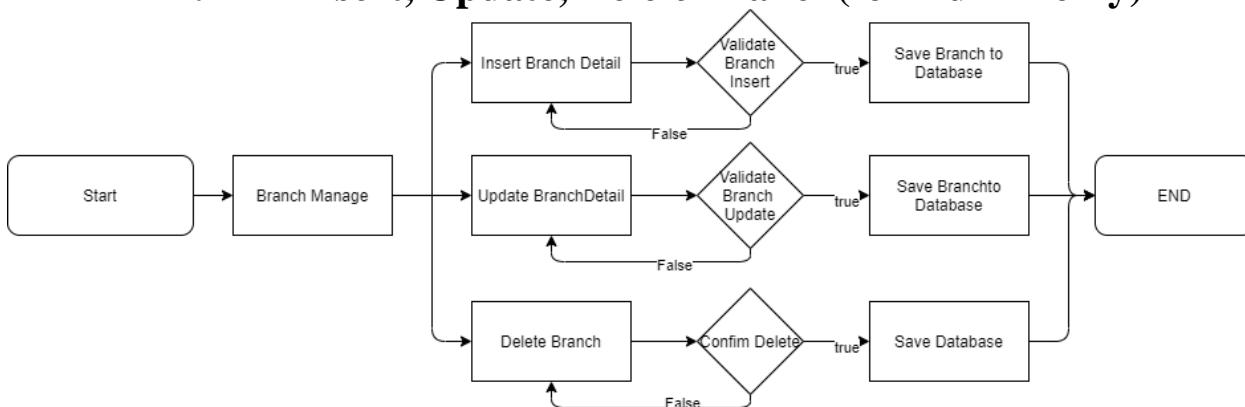
## 10. Reset Password Admin(for Admin only)



## 11. Chang Password User(for User only)



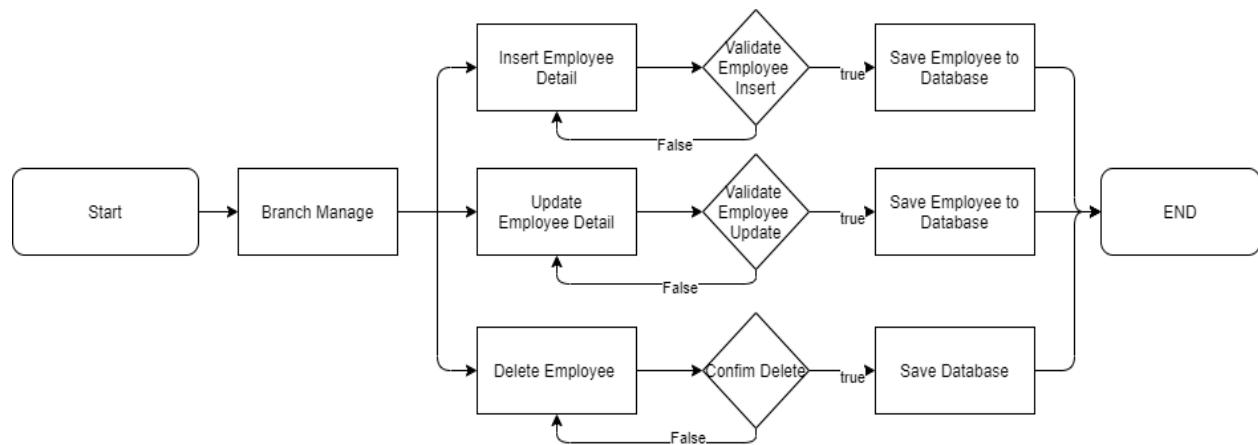
## 12. Insert, Update, Delete Branch(for Admin only)



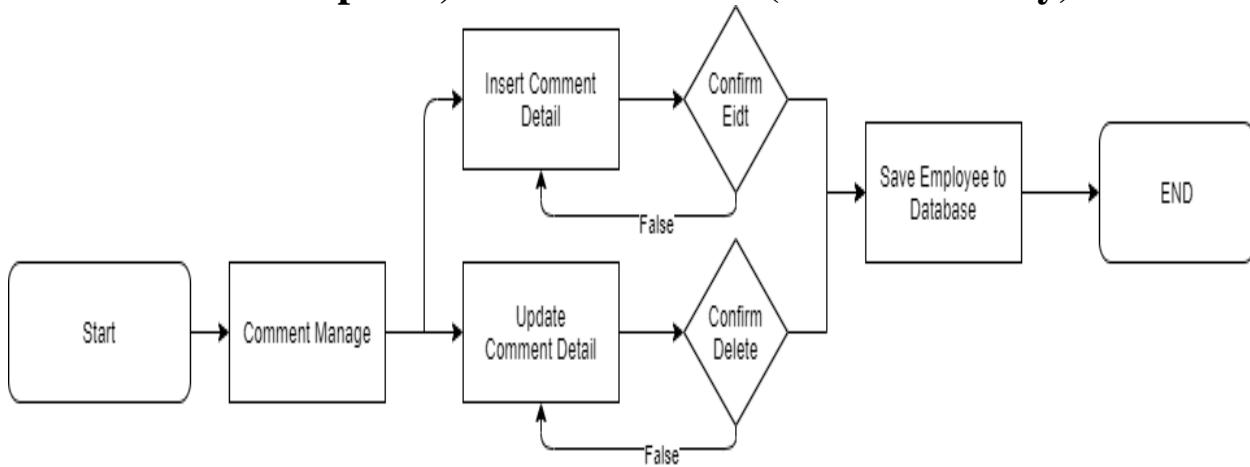
## 13. Insert, Update, Delete Employees(for Admin only)

Prepared by (Student) Project Group No:3	Approved by (Faculty)
Signature	
Date	

Design Plan:TARS Delivery	Document Name:Review 2	SWD/Form No.1
Effective Date:22/01/2021	Version:1.0	Page No:7



#### 14. Update, Delete Comment(for Admin only)



	Prepared by (Student) Project Group No:3	Approved by (Faculty)
Signature		
Date		

Design Plan:TARS Delivery	Document Name:Review 2	SWD/Form No.1
Effective Date:22/01/2021	Version:1.0	Page No:8

### III. Data Flow Diagram(DFD)

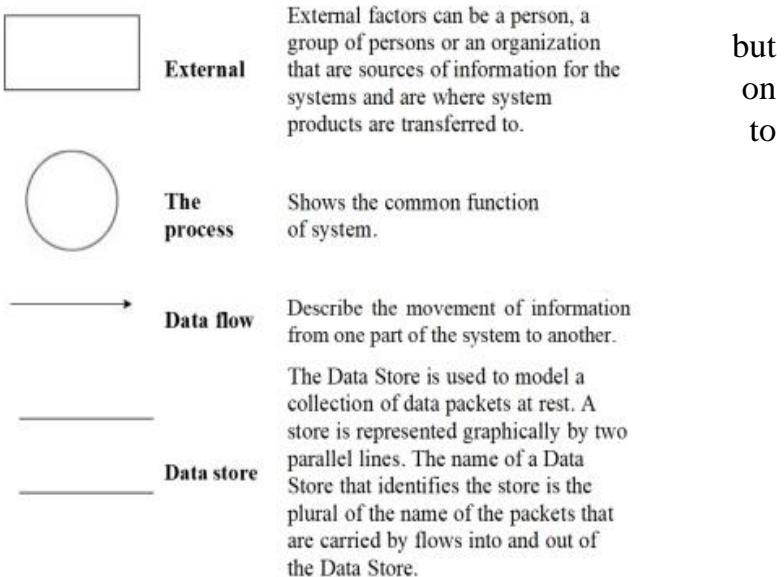
**Define:** Data Flows Diagram (DFD) describes the information flow in the system. The next step of system analysis is to consider in detail the information necessary for the implementation for functions discussed above and the one necessary for the improvement of the functions. Modelling tool frequently used for this purpose is DFD. DFD will support 4 main activities:

**Analysis:** DFD is used to determine requirement of users.

**Design:** DFD is used to map out plan and illustrate solution to analysis and users while designing a new system.

**Communication:** one of the strength of DFD is its simplicity and ease to understand to analysts and users;

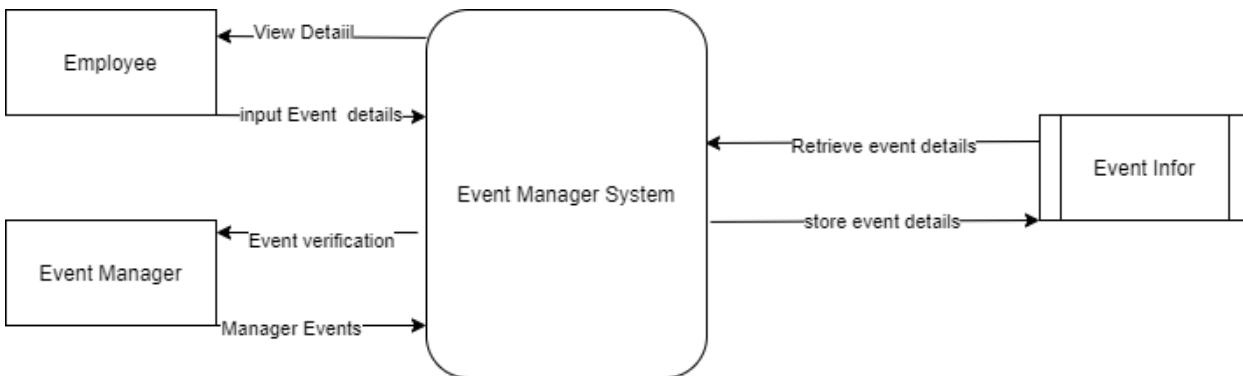
**Document:** DFD is used to provide special description of requirement and system design. DFD provide an overview of key functional components of the system it does not provide any detail these components. We have use other tools like database dictionary, process specification to get an idea of which information will be exchanged and how.



## 1. Context Level Diagram

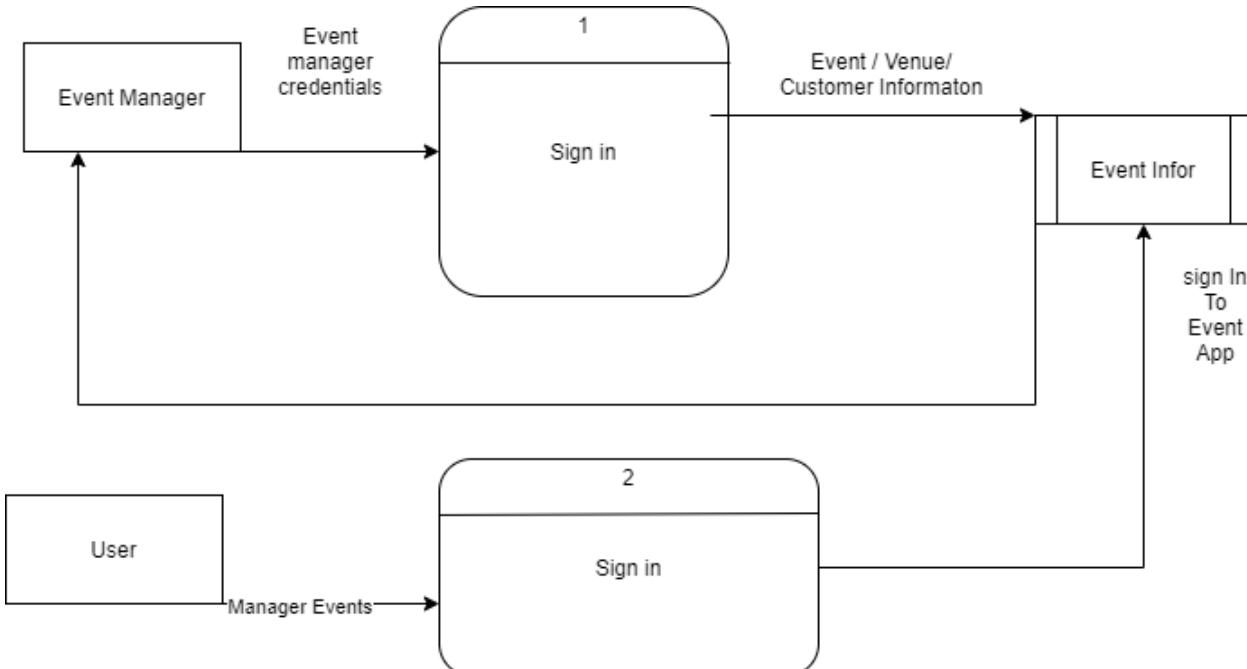
	Prepared by (Student)  Project Group No:3	Approved by (Faculty)
Signature		
Date		

Design Plan:TARS Delivery	Document Name:Review 2	SWD/Form No.1
Effective Date:22/01/2021	Version:1.0	Page No:9



## 2. Context DFD

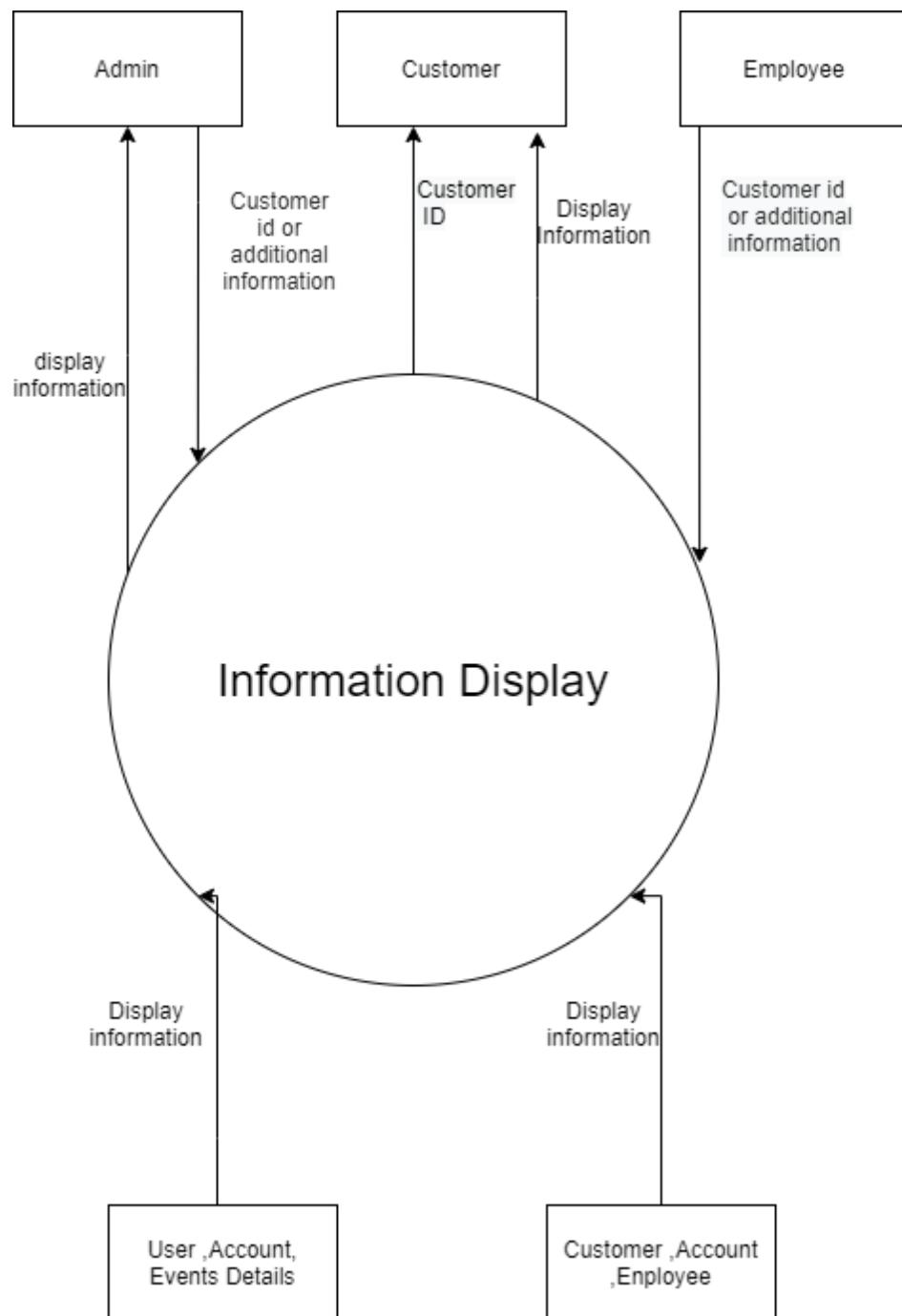
### 1. Login



	Prepared by (Student) Project Group No:3	Approved by (Faculty)
Signature		
Date		

Design Plan:TARS Delivery	Document Name:Review 2	SWD/Form No.1
Effective Date:22/01/2021	Version:1.0	Page No:10

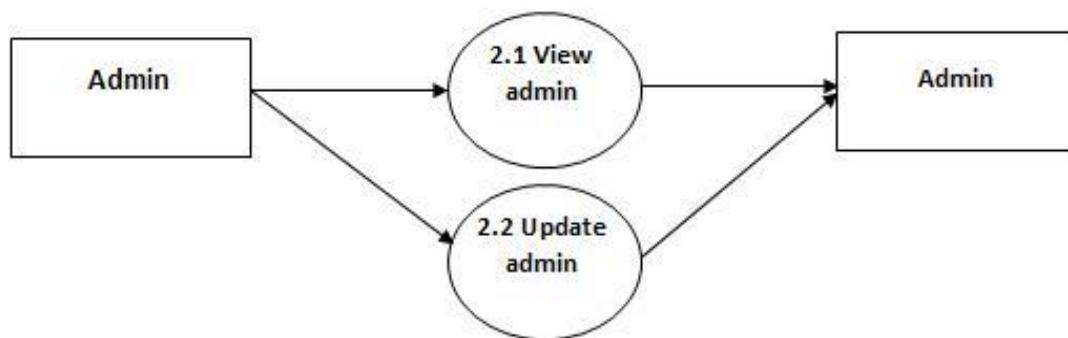
## 2. Customer And Accounts information



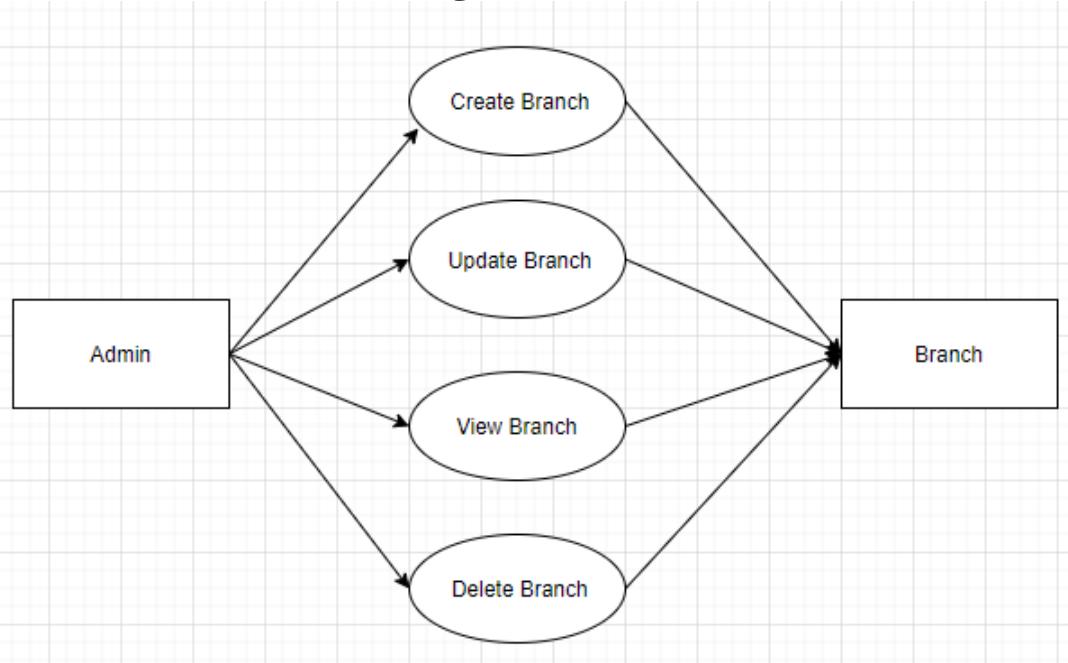
	Prepared by (Student) Project Group No:3	Approved by (Faculty)
Signature		
Date		

Design Plan:TARS Delivery	Document Name:Review 2	SWD/Form No.1
Effective Date:22/01/2021	Version:1.0	Page No:11

### 3. Admin manage Admin



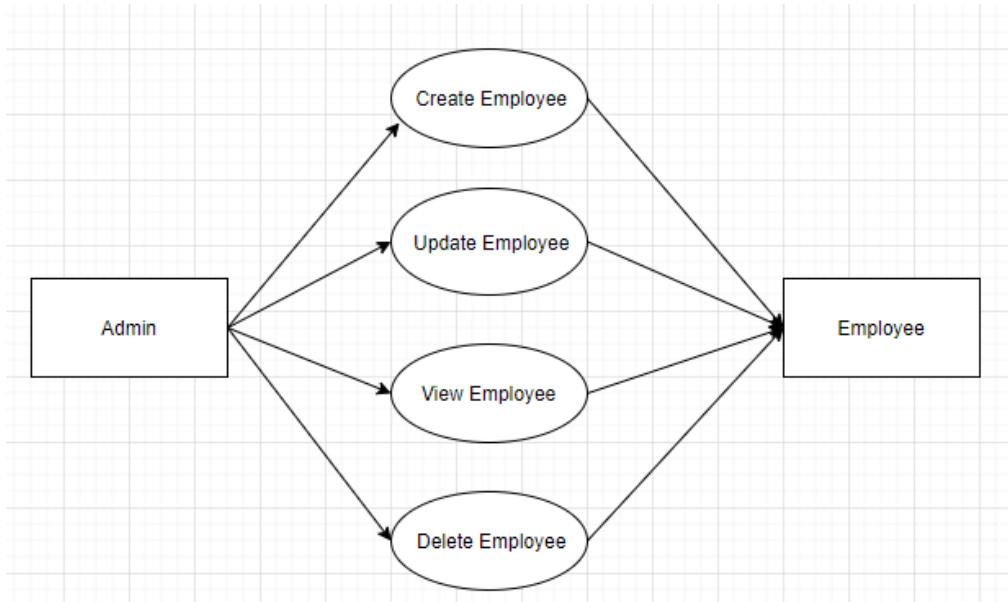
### 4. Admin manage Branch



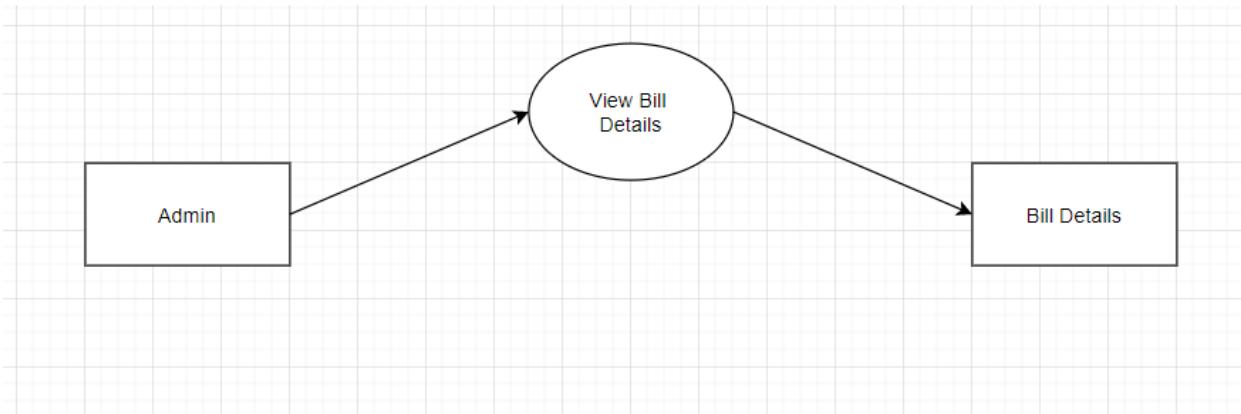
### 5. Admin manage Employee

	Prepared by (Student)	Approved by (Faculty)
	Project Group No:3	
Signature		
Date		

Design Plan:TARS Delivery	Document Name:Review 2	SWD/Form No.1
Effective Date:22/01/2021	Version:1.0	Page No:12



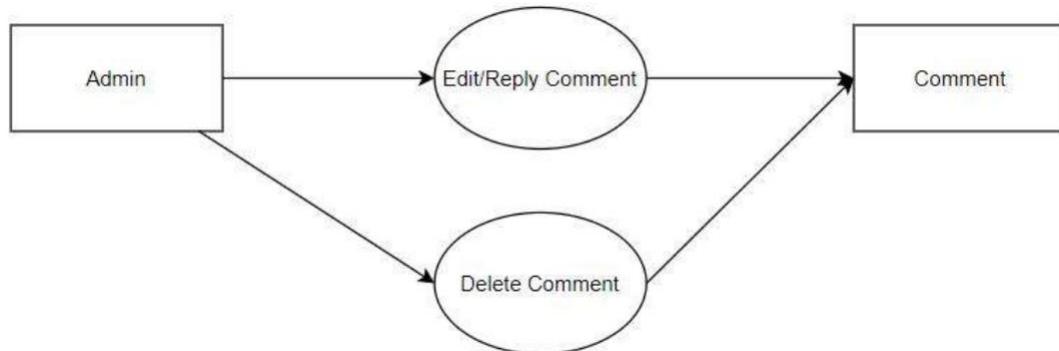
## 6. Admin manage Bill



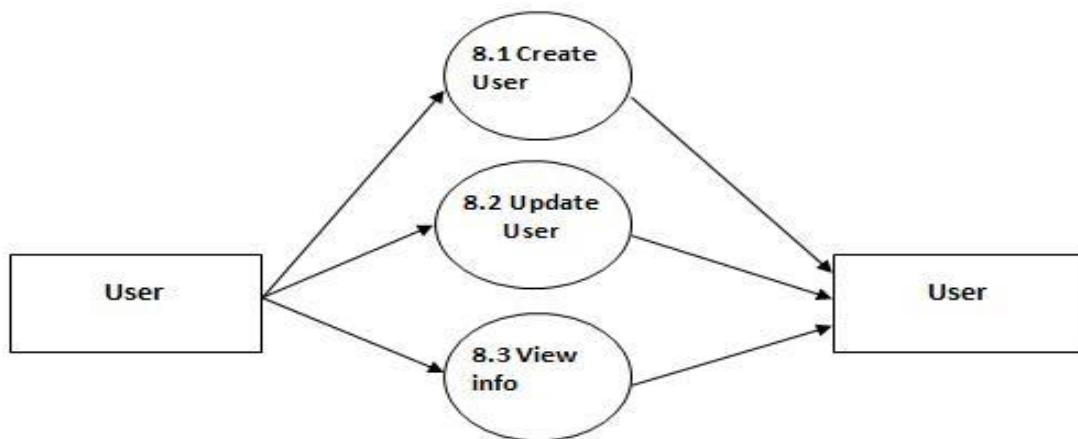
	Prepared by (Student) Project Group No:3	Approved by (Faculty)
Signature		
Date		

Design Plan:TARS Delivery	Document Name:Review 2	SWD/Form No.1
Effective Date:22/01/2021	Version:1.0	Page No:13

## 7. Admin manage Comment



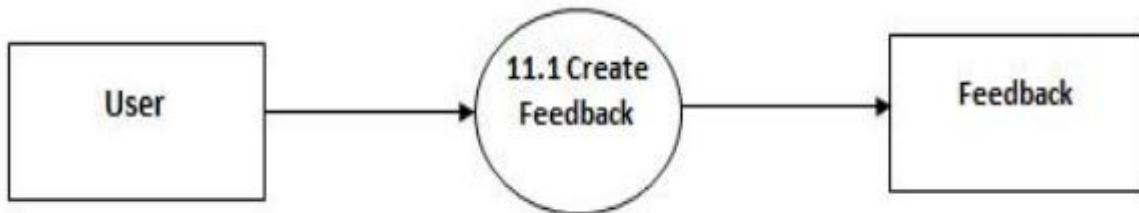
## 8. User manage User



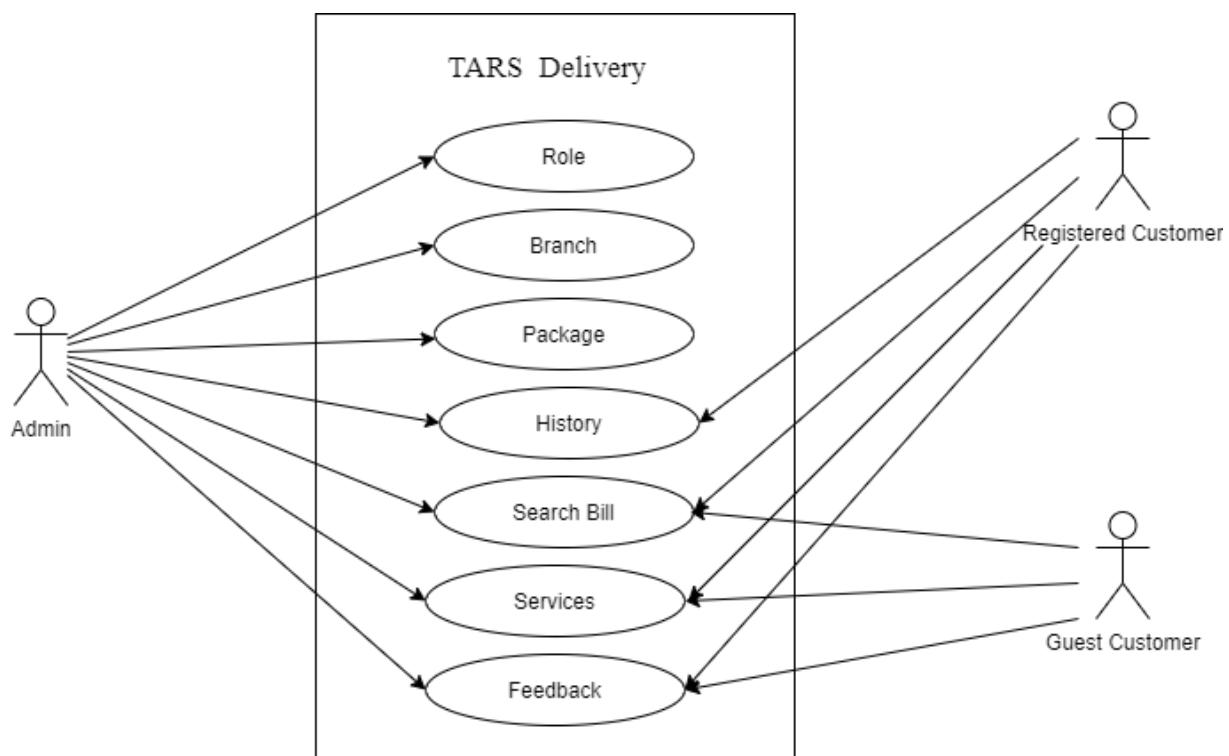
	Prepared by (Student) Project Group No:3	Approved by (Faculty)
Signature		
Date		

Design Plan:TARS Delivery	Document Name:Review 2	SWD/Form No.1
Effective Date:22/01/2021	Version:1.0	Page No:14

## 9. User manage Feedback



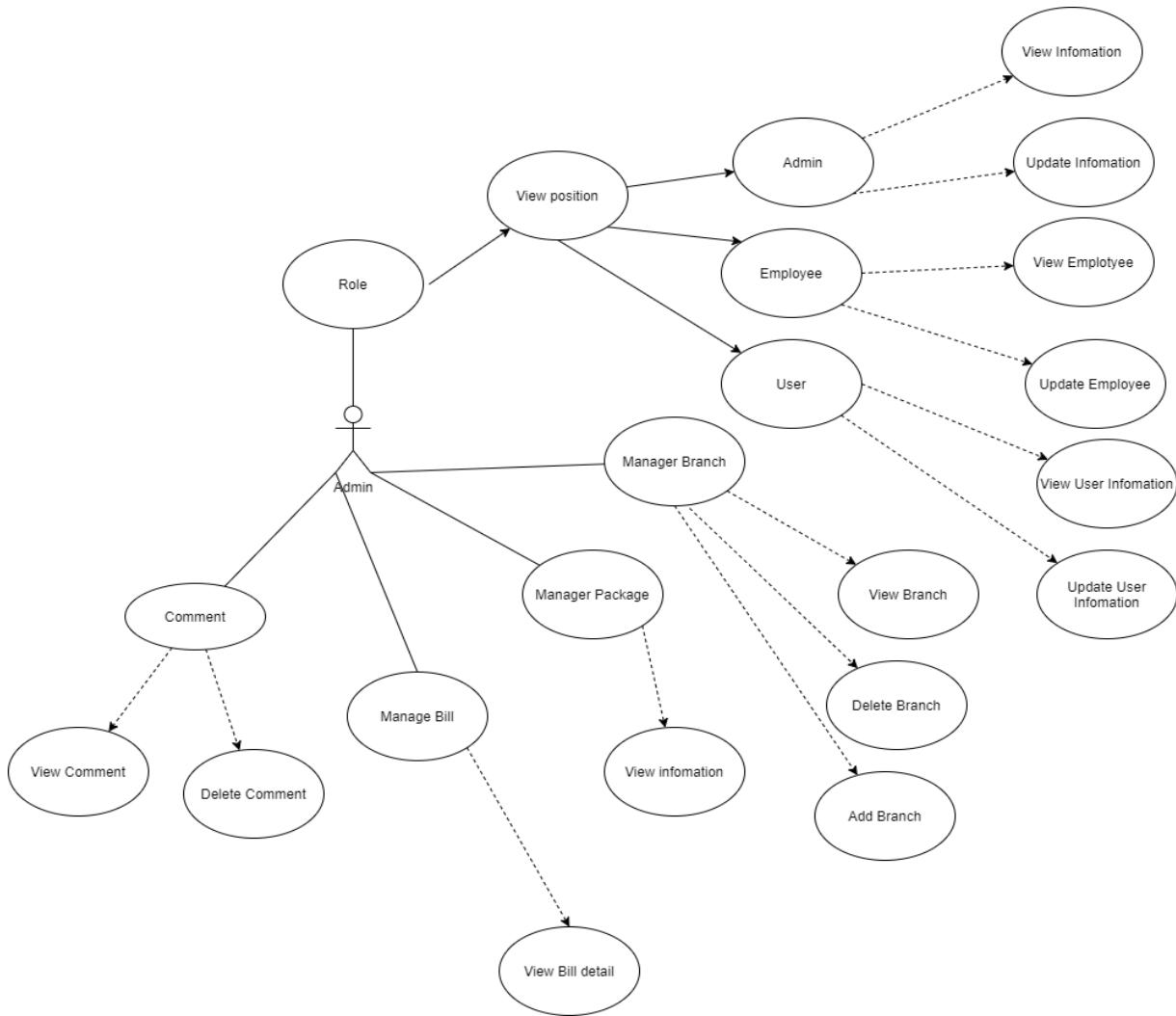
## IV. Use Case Diagram



	Prepared by (Student) Project Group No:3	Approved by (Faculty)
Signature		
Date		

Design Plan:TARS Delivery	Document Name:Review 2	SWD/Form No.1
Effective Date:22/01/2021	Version:1.0	Page No:15

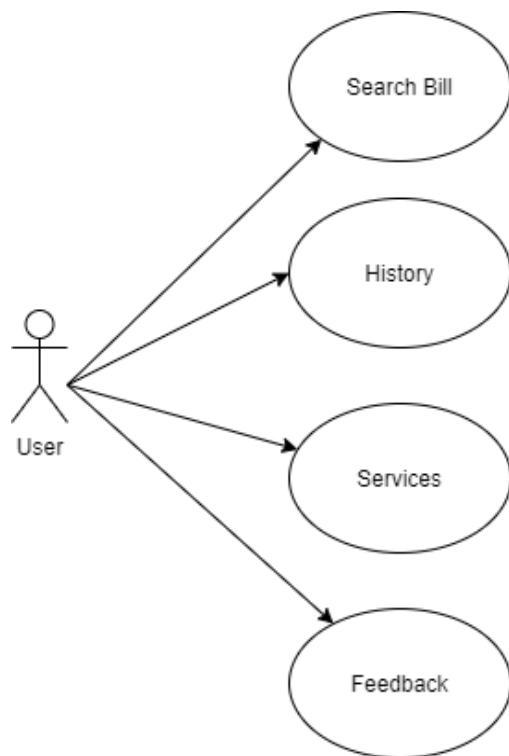
## 1. Admin use case



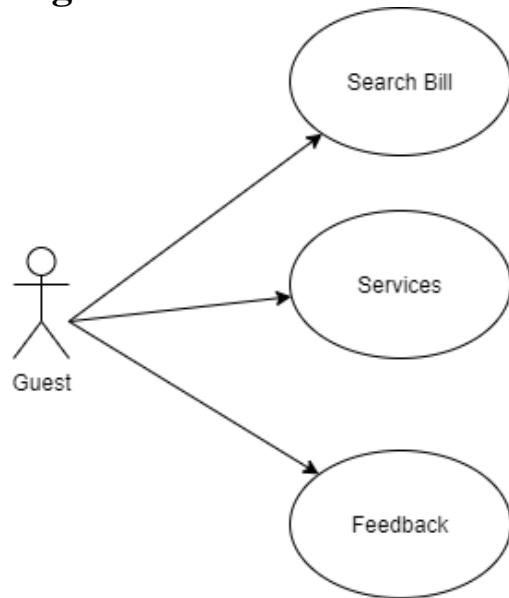
## 2. User use diagram

	Prepared by (Student) Project Group No:3	Approved by (Faculty)
Signature		
Date		

Design Plan:TARS Delivery	Document Name:Review 2	SWD/Form No.1
Effective Date:22/01/2021	Version:1.0	Page No:16



### 3. Guest user diagram



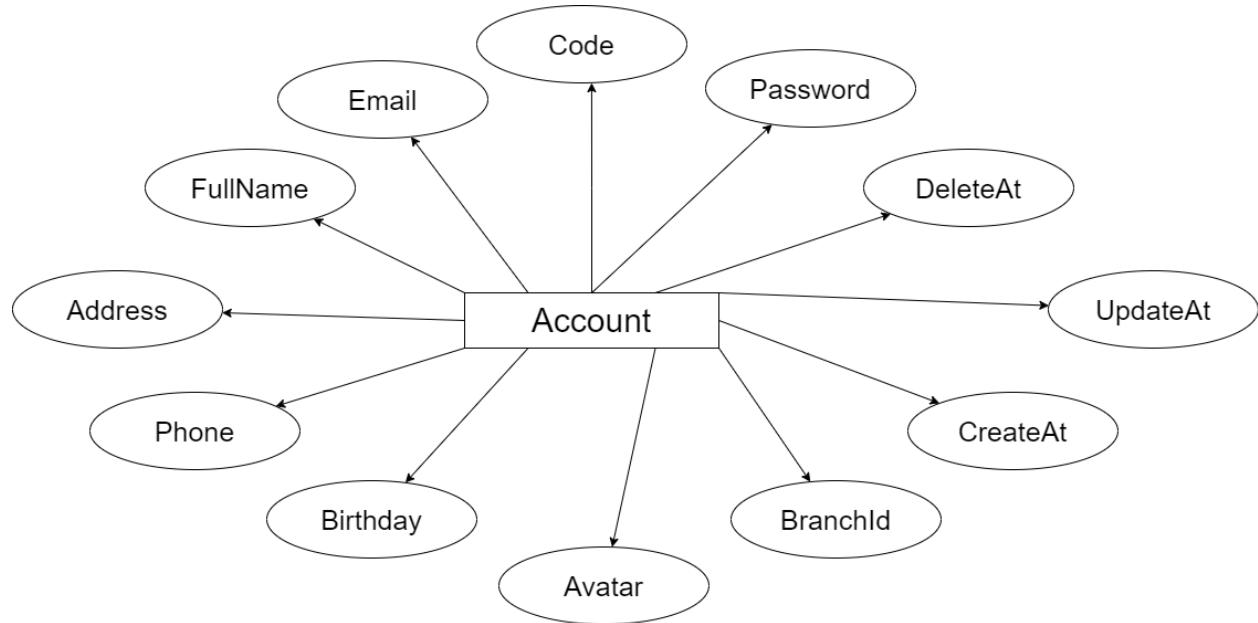
	Prepared by (Student) Project Group No:3	Approved by (Faculty)
Signature		
Date		

Design Plan:TARS Delivery	Document Name:Review 2	SWD/Form No.1
Effective Date:22/01/2021	Version:1.0	Page No:17

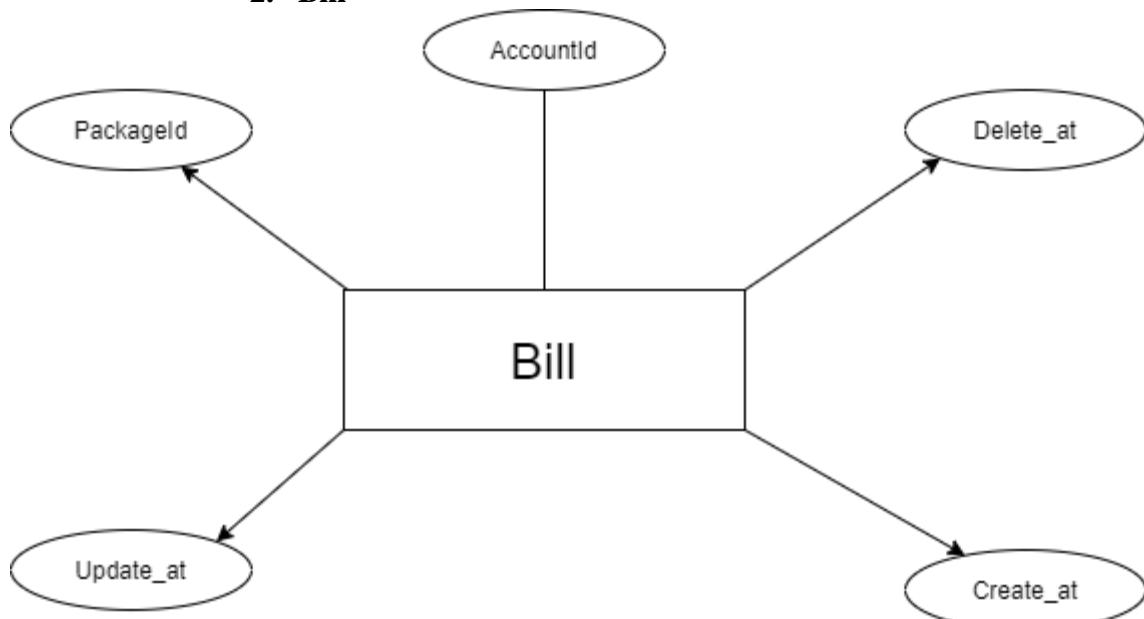
## V. Entity – Relationship Diagrams

### 1. Entities and Properties

#### 1. Account



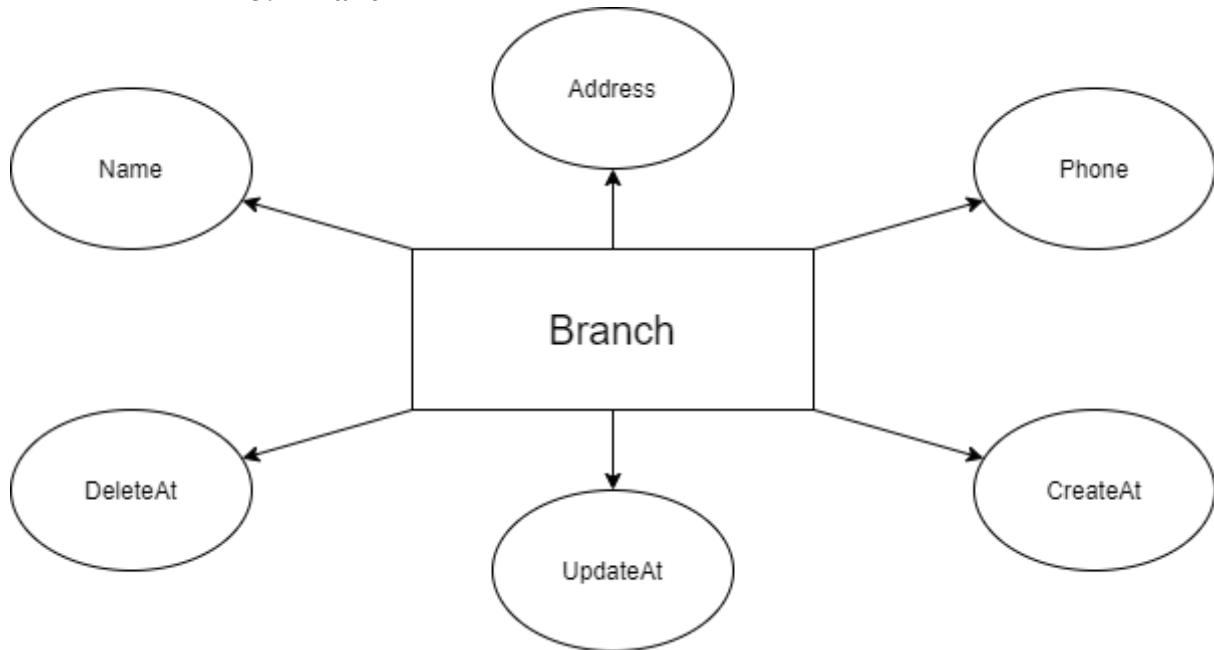
#### 2. Bill



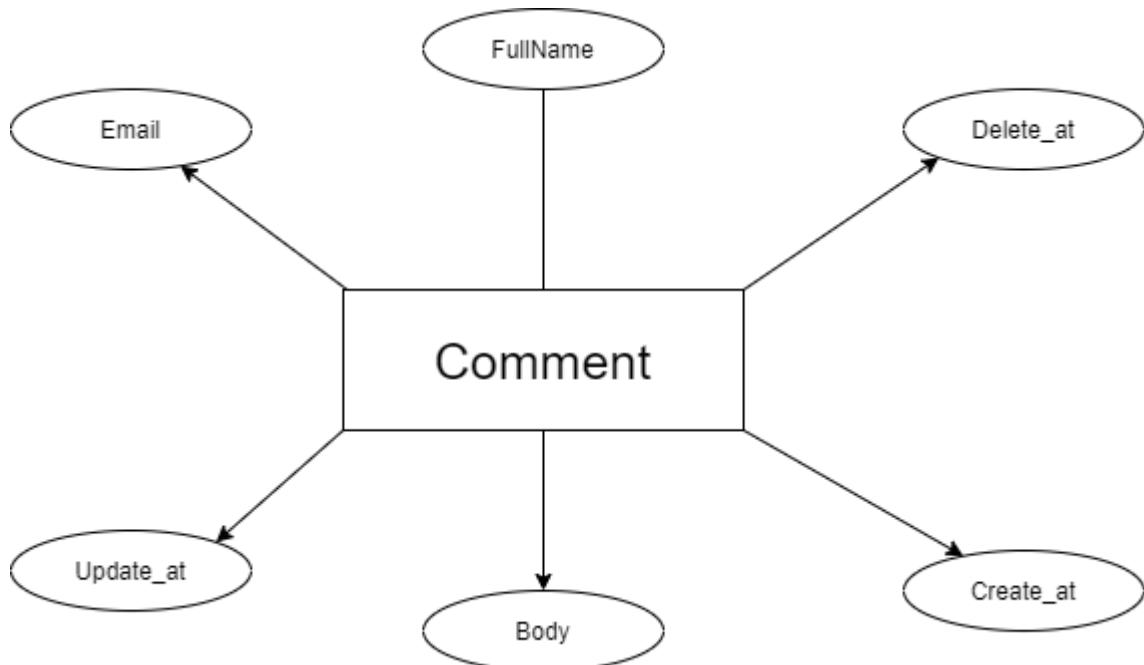
	Prepared by (Student) Project Group No:3	Approved by (Faculty)
Signature		
Date		

Design Plan:TARS Delivery	Document Name:Review 2	SWD/Form No.1
Effective Date:22/01/2021	Version:1.0	Page No:18

### 3. Branch



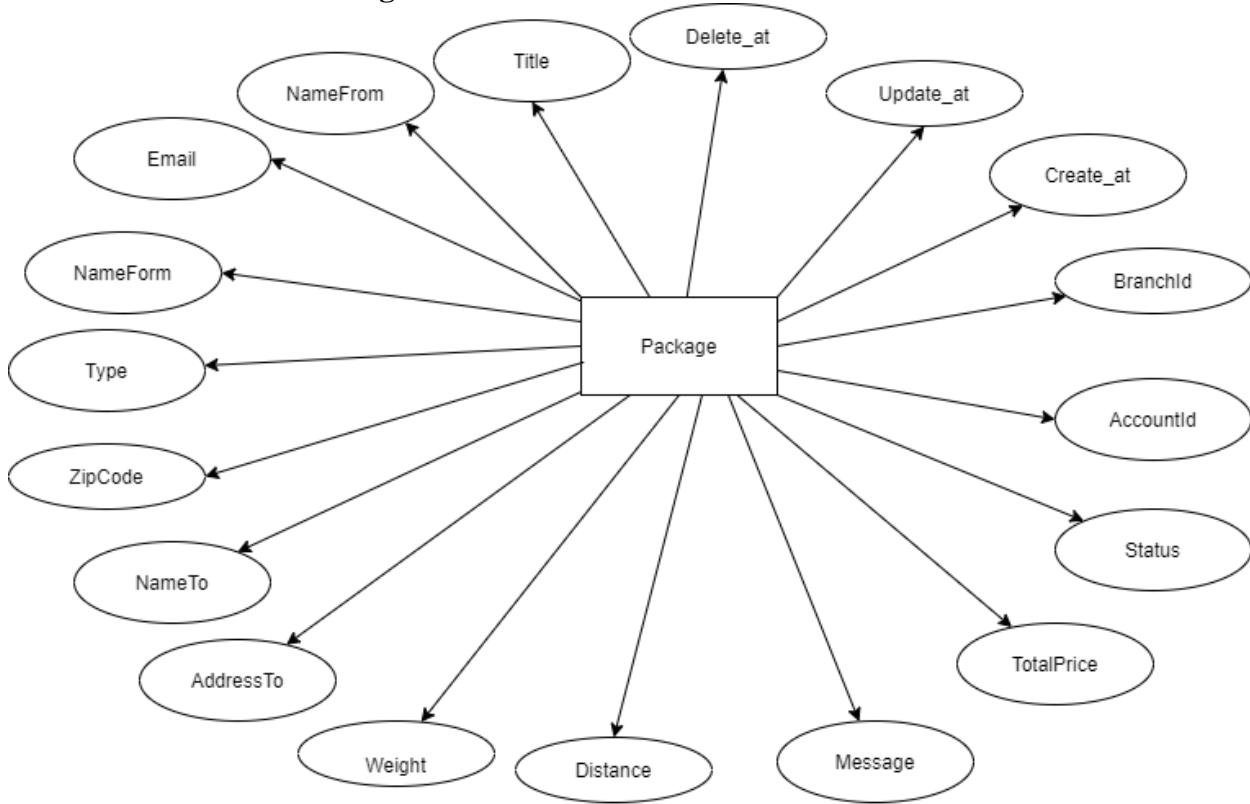
### 4. Comment



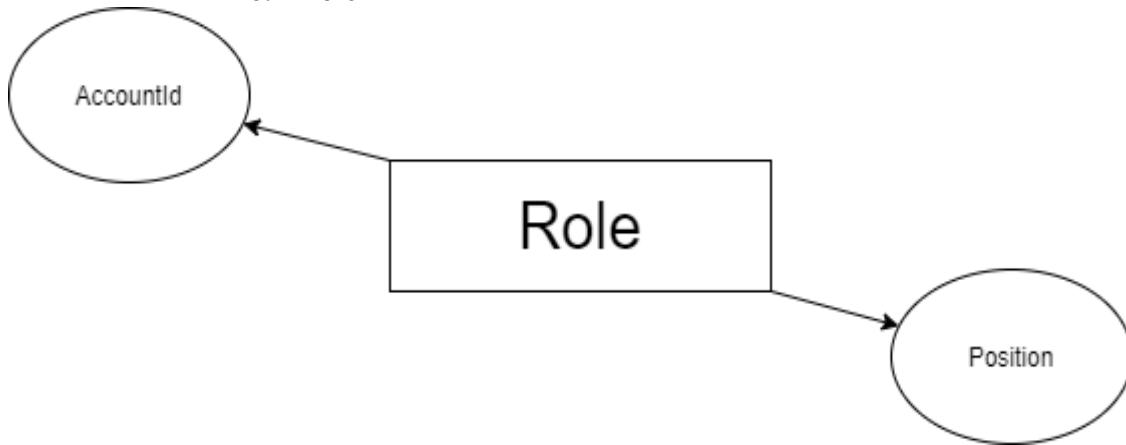
	Prepared by (Student) Project Group No:3	Approved by (Faculty)
Signature		
Date		

Design Plan:TARS Delivery	Document Name:Review 2	SWD/Form No.1
Effective Date:22/01/2021	Version:1.0	Page No:19

## 5. Package



## 6. Role



	Prepared by (Student) Project Group No:3	Approved by (Faculty)
Signature		
Date		

Design Plan:TARS Delivery	Document Name:Review 2	SWD/Form No.1
Effective Date:22/01/2021	Version:1.0	Page No:20

## Table Design

Table: Package				
Field Name	Data Type	Null	Key	Description
Id	Int	Not null	PK	Stores the id of the Package
Title	Nvarchar(max)	Null		Stores the Title of the Package
NameFrom	Nvarchar(max)	Null		Stores the NameFrom of the Package
Email	Nvarchar(max)	Null		Stores the Email of the Package
AddressFrom	Nvarchar(max)	Null		Stores the AddressFrom of the Package
Type	Nvarchar(max)	Null		Stores the Type of the Package
ZipCode	Nvarchar(max)	Null		Stores the ZipCode of the Package
NameTo	Nvarchar(max)	Null		Stores the NameTo of the Package
AddressTo	Nvarchar(max)	Null		Stores the AddressTo of the Package
Weight	Float	Null		Stores the Weight of the Package
Distance	Float	Not Null		Stores the Distance of the Package
Message	Nvarchar(max)	Null		Stores the Message of the Package
TotalPrice	Float	Not Null		Stores the TotalPrice of the Package
Status	Nvarchar(max)	Not Null		Stores the Status of the Package
BranchId	Int	Null	FK	Stores the Branch id selected by the Package
AccountId	Int	Null	FK	Stores the Account id selected by the Package
PriceListName	Nvarchar(max)	Null		Stores the PriceListName of the Package
Create_at	Datetime2(7)	Not Null		Stores the Create of the Package
Update_at	Datetime2(7)	Null		Stores the Update of the Package
Delete_at	Datetime2(7)	Null		Stores the Delete of the Package

Table: Role				
Field Name	Data Type	Null	Key	Description
id	int	Not Null	PK	Stores the id of the Role
AccountId	int	Not Null	FK	Stores the Account id selected by the Role
Position	int	Not Null		

	Prepared by (Student) Project Group No:3	Approved by (Faculty)
Signature		
Date		

Design Plan:TARS Delivery	Document Name:Review 2	SWD/Form No.1
Effective Date:22/01/2021	Version:1.0	Page No:21

**Table: Branch**

Field Name	Data Type	Null	Key	Description
Id	int	NO	PK	Branch Id
Name	nvarchar(MAX)	YES		Branch Name
Address	nvarchar(MAX)	YES		Branch Address
Phone	nvarchar(MAX)	YES		Branch Phone
Create_at	datetime2(7)	NO		Branch Create Time
Update_at	datetime2(7)	YES		Branch Update Time
Delete_at	datetime2(7)	YES		Branch Delete Time

**Table: Bill**

Field Name	Data Type	Null	Key	Description
Id	int	NO	PK	Bill Id
AccountId	int	YES	FK	Bill Account Id
Packageld	int	NO	FK	Bill Package Id
Create_at	datetime2(7)	NO		Bill Create Time
Update_at	datetime2(7)	YES		Bill Update Time
Delete_at	datetime2(7)	YES		Bill Delete Time

**Table: Comment**

Field Name	Data Type	Null	Key	Description
Id	int	NO	PK	Comment Id
FullName	nvarchar(MAX)	YES		Comment Full Name
Email	nvarchar(MAX)	YES		Comment Email
Body	nvarchar(MAX)	YES		Comment Body
Create_at	datetime2(7)	NO		Comment Create at
Update_at	datetime2(7)	YES		Comment Update at
Delete_at	datetime2(7)	YES		Comment Delete at

**Prepared by (Student)**

**Project Group No:3**

**Approved by (Faculty)**

<b>Signature</b>		
<b>Date</b>		

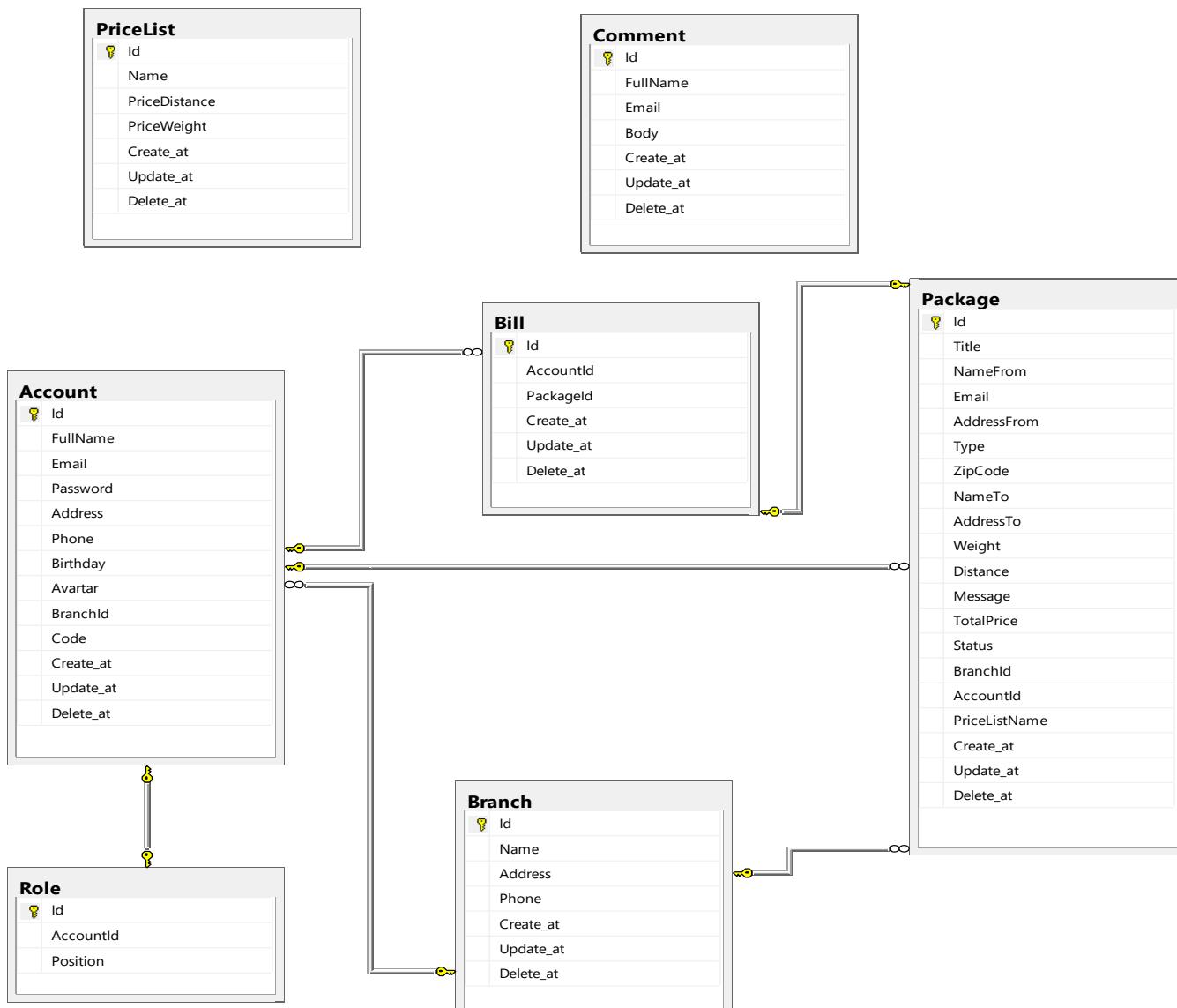
Design Plan:TARS Delivery	Document Name:Review 2	SWD/Form No.1
Effective Date:22/01/2021	Version:1.0	Page No:22

Table: Account				
Field Name	Data Type	Null	Key	Description
Id	int	NO	PK	Account Id
FullName	nvarchar(MAX)	YES		Account Full Name
Email	nvarchar(MAX)	YES		Account Email
Password	nvarchar(MAX)	YES		Account Password
Address	nvarchar(MAX)	YES		Account Address
Phone	nvarchar(MAX)	YES		Account Phone
Birthday	datetime2(7)	NO		Account Birthday
Avatar	nvarchar(MAX)	YES		Account Avatar
BranchId	int	YES	FK	Account Branch Id
Code	nvarchar(MAX)	YES		Account Code
Create_at	datetime2(7)	NO		Account Create Time
Update_at	datetime2(7)	YES		Account Update Time
Delete_at	datetime2(7)	YES		Account Delete Time

	Prepared by (Student) Project Group No:3	Approved by (Faculty)
Signature		
Date		

Design Plan:TARS Delivery	Document Name:Review 2	SWD/Form No.1
Effective Date:22/01/2021	Version:1.0	Page No:23

## Relation Diagram



	Prepared by (Student)	Approved by (Faculty)
	Project Group No:3	
Signature		
Date		

Design Plan:TARS Delivery	Document Name:Review 2	SWD/Form No.1
Effective Date:22/01/2021	Version:1.0	Page No:24

## TASK SHEET REVIEW 2

	Prepared by (Student) Project Group No:3	Approved by (Faculty)
Signature		
Date		

Design Plan:TARS Delivery	Document Name:Review 2	SWD/Form No.1
Effective Date:22/01/2021	Version:1.0	Page No:25

Project Ref. No.:		Project Title:	Activity Plan Prepared By:	Date of Preparation of Activity Plan:			
Sr.N o.	Task			Actual Start Date	Actual Days	Team Mate Names	Status
1	Architecture & Design of Project	TARS Delivery	Huy	21/01/2021	1	All Member	Completed
2	Algorithms – Data Flowchart			21/01/2021	3	Huy	Completed
3	Data Flow Diagram			21/01/2021	3	Hoai	Completed
4	Use Case Diagram			21/01/2021	3	Tu	Completed
5	Entity-Relationship Diagram			21/01/2021	3	Kien	Completed
6	Task sheet review			21/01/2021	1	Tu	Completed

	Prepared by (Student) <b>Project Group No:3</b>	Approved by (Faculty)
Signature		
Date		

Design Plan:TARS Delivery	Document Name:Review 2	SWD/Form No.1
Effective Date:22/01/2021	Version:1.0	Page No:26

Signature of Instructor	Signature Team Leader
<b>Mr. Tran Phuoc Sinh</b>	<b>Pham Quang Huy</b>

	Prepared by (Student)	Approved by (Faculty)
	<b>Project Group No:3</b>	
<b>Signature</b>		
<b>Date</b>		

Design Plan:TARS Delivery	Document Name:Review 2	SWD/Form No.1
Effective Date:22/01/2021	Version:1.0	Page No:27

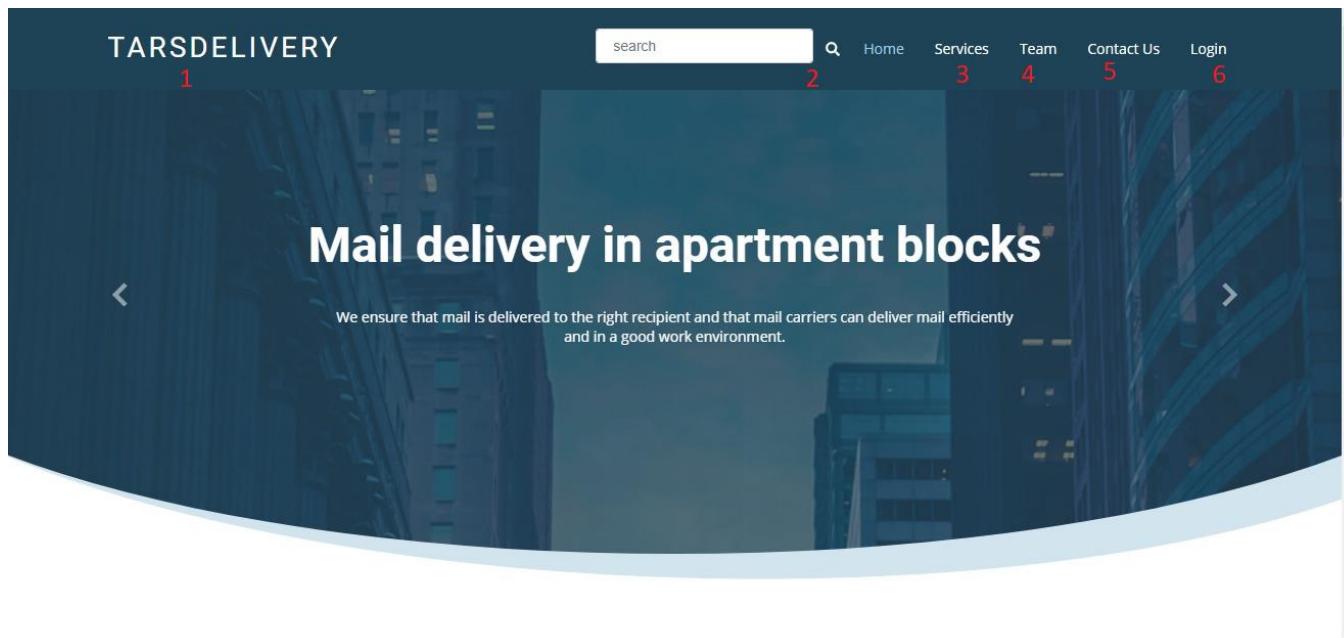
# REVIEW 3

	Prepared by (Student) Project Group No:3	Approved by (Faculty)
Signature		
Date		

Design Plan:TARS Delivery	Document Name:Review 2	SWD/Form No.1
Effective Date:22/01/2021	Version:1.0	Page No:28

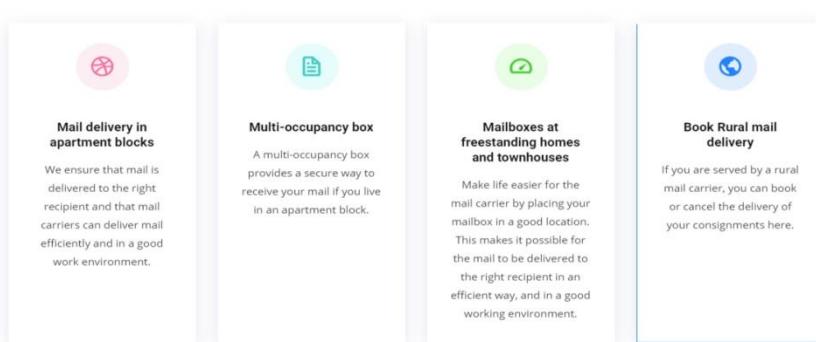
User:

## 1.Index Page



	Prepared by (Student) Project Group No:3	Approved by (Faculty)
Signature		
Date		

Design Plan:TARS Delivery	Document Name:Review 2	SWD/Form No.1
Effective Date:22/01/2021	Version:1.0	Page No:29



### Mail delivery to apartment blocks

We aim to ensure that mail is delivered to the right recipient and that mail carriers can deliver mail efficiently and in a good work environment.



### Delivery to normal mailboxes

*If your building does not have a multi-occupancy mailbox, the mail is delivered through the mailbox in the apartment door or to a mailbox located on the relevant floor of the building.*

Mark the mailbox clearly with the names of the residents - the text should be at least 8 mm tall. Good lighting is important. The lighting on each floor should be at least 200 lux and it should be at least 100 lux in stairwells. The mailbox must be a complete unit, with no risk of being cut or squeezed when it is used.



### Mail that is not delivered

*Mail that cannot be delivered because, for example, the entry door is locked or the elevator does not work, is stored at the mail delivery office and delivered on the next delivery round. If your entry code has been or will soon be changed, it is important that you let us know. Inform us about new entry code.*

### Delivery in the case of major renovations

*Often, major renovations of the plumbing, elevators or similar make it difficult or impossible for mail carriers to make deliveries. Please contact our customer service for assistance deciding how the delivery of mail can be carried out during renovation work.*



	Prepared by (Student) <b>Project Group No:3</b>	Approved by (Faculty)
<b>Signature</b>		
<b>Date</b>		

Design Plan:TARS Delivery	Document Name:Review 2	SWD/Form No.1
Effective Date:22/01/2021	Version:1.0	Page No:30

Contact Us

590 Cách Mạng Tháng Tám  
Phường 11, Quận 3  
Thành phố Hồ Chí Minh

Phone: +1 5589 55488 55  
Email: tarsdeliverydemo@gmail.com

About TARSDELIVERY

We deliver mail during the day to all households, on all days except weekends and public holidays. Information is provided here about mail delivery and the services you can use to ensure smooth delivery in line with your needs.

Twitter Facebook Instagram LinkedIn

No	Name	Type	Validation	Event	Description	Status
1	Home	Hyperlink		Click	Click to go to Home	Enable
2	Search	Hyperlink		Click	Click to go to search	Enable
3	Services	Hyperlink		Click	Choose and click go to sub page	Enable
4	Team	Hyperlink		Click	Click to go to Team Page	Enable
5	Contact Us	Hyperlink		Click	Click to go to Contact Us page	Enable
6	Login	Hyperlink		Click	Click to go to Login page	Enable

	Prepared by (Student) <b>Project Group No:3</b>	Approved by (Faculty)
Signature		
Date		

Design Plan:TARS Delivery	Document Name:Review 2	SWD/Form No.1
Effective Date:22/01/2021	Version:1.0	Page No:31

## Source code:

```

public IActionResult Search(int? search)
{
    if(search == null)
    {
        return RedirectToAction("index", "Home");
    }
    var model = JsonConvert.DeserializeObject<Package>(httpClient.GetStringAsync(uriPackage + search).Result);
    if(model == null)
    {
        return RedirectToAction("index", "Home");
    }
    return View(model);
}

```

	Prepared by (Student) Project Group No:3	Approved by (Faculty)
Signature		
Date		

Design Plan:TARS Delivery	Document Name:Review 2	SWD/Form No.1
Effective Date:22/01/2021	Version:1.0	Page No:32

## 2.Services

**TARSDELIVERY**

Check the distance to send the parcel

1 Enter a location of origin  
2 Enter a location of destination

View Map View Information

Services Home / Services

### Check the distance to send the parcel

590 Cách Mạng Tháng Tám, Phường 11, Quận 3, Thành phố Hồ Chí Minh, Việt Nam  
487 Huỳnh Tấn Phát, Tân Thuận Đông, Quận 7, Thành phố Hồ Chí Minh, Việt Nam

Distance Is Kilo: 12.61  
IN MINUTES: 22:51

FROM: 590 Cách Mạng Tháng Tám, Phường 11, Quận 3, Thành phố Hồ Chí Minh, Việt Nam  
TO: 487 Huỳnh Tấn Phát, Tân Thuận Đông, Quận 7, Thành phố Hồ Chí Minh, Việt Nam

Weights(gram): 100  
10% Fuel Fees, 20% Route Fees, 10% VAT (price + 10% + (20% \* kilo) + 10%)  
Price: 4.02 USD

View Map Reset Information

### Details the parcel

CACHUY1615040072PV

Your Name <b>1</b>	Your Email <b>2</b>	
Recipient's Name <b>3</b>	Package <b>4</b>	ZipCode <b>5</b>
Message <b>6</b>		
<b>Confirm</b> <b>7</b>		

	Prepared by (Student) <b>Project Group No:3</b>	Approved by (Faculty)
<b>Signature</b>		
<b>Date</b>		

Design Plan:TARS Delivery	Document Name:Review 2	SWD/Form No.1
Effective Date:22/01/2021	Version:1.0	Page No:33

No	Name	Type	Validation	Event	Description	Status
1	Your Name	TextBox		Focus	Input Your Name	Enable
2	Your Email	TextBox		Focus	Input Your Email	Enable
3	Recipient's Name	TextBox		Focus	Input Your Recipient's Name	Enable
4	Packge and mail	option		Click	Option package and mail	Enable
5	Zipcode	option		Focus	Input Your Zipcode	Enable
6	Message	TextBox		Focus	Input Your Message	Enable
7	Submit	button		Click	Submit	Enable

	Prepared by (Student) <b>Project Group No:3</b>	Approved by (Faculty)
Signature		
Date		

Design Plan:TARS Delivery	Document Name:Review 2	SWD/Form No.1
Effective Date:22/01/2021	Version:1.0	Page No:34

## Sources code:

```
// Print results on submit the form
$('#distance_form').submit(function (e) {
  e.preventDefault();

  origin = $('#origin').val();
  destination = $('#destination').val();

  // Process
  (async () => {
    try {
      await displayRoute();
      await calculateDistance();
    } catch (error) {
      console.log(error);
      $('#view-information').prop('disabled', true);
    }
  })();
});
```

	Prepared by (Student)	Approved by (Faculty)
	Project Group No:3	
Signature		
Date		

Design Plan:TARS Delivery	Document Name:Review 2	SWD/Form No.1
Effective Date:22/01/2021	Version:1.0	Page No:35

### 3. Team

Our Team

[Home](#) / Our Team



**Pham Quang Huy**  
TEAM LEADER



**Pham Xuan Hoai**  
MEMBER



**Ngo Trung Kien**  
MEMBER



**Le Hoang Anh Tu**  
MEMBER



	Prepared by (Student) <b>Project Group No:3</b>	Approved by (Faculty)
Signature		
Date		

Design Plan:TARS Delivery	Document Name:Review 2	SWD/Form No.1
Effective Date:22/01/2021	Version:1.0	Page No:36

## 4.Contact Us

Contact

Home / Contact


**Our Address**  
 590 Cách Mạng Tháng Tám, Phường 11, Quận 3, Thành phố Hồ Chí Minh


**Email Us**  
 tarsdeliverydemo@gmail.com  
 toilahuy098@gmail.com


**Call Us**  
 +1 5589 55488 55  
 +1 6678 254445 41

1

2

3

4



No	Name	Type	Validation	Event	Description	Status
1	Full Name	TextBox		Focus	Input Full Name	Enable
2	Email	TextBox		Focus	Input Email	Enable
3	Body Message	TextBox		Focus	Input Body Message	Enable
4	Send Message	Button		Click	Submit	Enable

Prepared by (Student)	Approved by (Faculty)
<b>Project Group No:3</b>	
<b>Signature</b>	
<b>Date</b>	

Design Plan:TARS Delivery	Document Name:Review 2	SWD/Form No.1
Effective Date:22/01/2021	Version:1.0	Page No:37

## Source Code:

```
[HttpPost]
public IActionResult Contact(Comment comment)
{
    try
    {
        if (ModelState.IsValid)
        {
            var model = httpClient.PostAsJsonAsync(uriComment, comment).Result;
            if (model.IsSuccessStatusCode)
            {
                return RedirectToAction("Contact");
            }
            else
            {
                ViewBag.Msg = "Error";
                return View();
            }
        }
        else
        {
            return View();
        }
    }
    catch(Exception e)
    {
        ModelState.AddModelError(string.Empty, e.Message);
    }
    return View();
}
```

	Prepared by (Student) Project Group No:3	Approved by (Faculty)
Signature		
Date		

Design Plan:TARS Delivery	Document Name:Review 2	SWD/Form No.1
Effective Date:22/01/2021	Version:1.0	Page No:38

## 5.Login

Login

Home / Login

The screenshot shows a login interface with the following elements and callouts:

- Email:** Input field labeled "Enter email address". Callout 1 is positioned to its right.
- Password:** Input field labeled "Enter password". Callout 2 is positioned to its right.
- Forgot Password?**: Hyperlink. Callout 3 is positioned below it.
- Submit**: Button. Callout 4 is positioned to its left.
- Sign in**: Button with a Google logo. Callout 5 is positioned to its right.
- Need an account? Sign up!**: Hyperlink. Callout 6 is positioned below it.

No	Name	Type	Validation	Event	Description	Status
1	Email	TextBox		Focus	Input Email	Enable
2	Password	TextBox		Focus	Input Password	Enable
3	Forgot password	Hyperlink		Focus	Input Body Message	Enable
4	Submit	submit		Click	submit	Enable
5	Login	Hyperlink		Click	Login google	Enable
6	Sign up	Hyperlink		Click	Sign up account	Enable

	Prepared by (Student)	Approved by (Faculty)
	<b>Project Group No:3</b>	
<b>Signature</b>		
<b>Date</b>		

Design Plan:TARS Delivery	Document Name:Review 2	SWD/Form No.1
Effective Date:22/01/2021	Version:1.0	Page No:39

## Source code:

```
public IActionResult Login()
{
    var saccount = HttpContext.Session.GetString("sAccount");
    if (saccount != null)
    {
        return RedirectToAction("Index", "Home");
    }
    return View();
}
```

```
[HttpPost]
0 references
public IActionResult Login(Account acc)
{
    try
    {
        var accounts = JsonConvert.DeserializeObject<IEnumerable<Account>>(httpClient.GetStringAsync(uriAccount).Result);
        var account = accounts.SingleOrDefault(a => a.Email.Equals(acc.Email) && BCrypt.Net.BCrypt.Verify(acc.Password, a.Password));
        if (account != null)
        {
            Role role = JsonConvert.DeserializeObject<Role>(httpClient.GetStringAsync(uriRole + account.Id).Result);
            if (role.Position == 1 || role.Position == 3 && account.Delete_at == null)
            {
                HttpContext.Session.SetString("sRole", role.Position.ToString());
                HttpContext.Session.SetString("sAccount", JsonConvert.SerializeObject(account));
                return RedirectToAction("Index", "Home");
            }
            else
            {
                ViewBag.Msg = "No authorize to connect.";
                return View();
            }
        }
        else
        {
            ViewBag.Msg = "Email or password is wrong! Please try again.";
        }
    }
    catch (Exception e)
    {
        ViewBag.Msg = e.Message;
    }
    return View();
}
```

	Prepared by (Student) Project Group No:3	Approved by (Faculty)
Signature		
Date		

Design Plan:TARS Delivery	Document Name:Review 2	SWD/Form No.1
Effective Date:22/01/2021	Version:1.0	Page No:40

## 6. Forgot password :

Forgot Password

Home / Forgot Password

No	Name	Type	Validation	Event	Description	Status
1	Email	TextBox		Focus	Input Email	Enable
2	Return to login	Hyperlink		Click	Return to login	Enable
3	Reset password	Submit		Click	submit	Enable

	Prepared by (Student) Project Group No:3	Approved by (Faculty)
Signature		
Date		

Design Plan:TARS Delivery	Document Name:Review 2	SWD/Form No.1
Effective Date:22/01/2021	Version:1.0	Page No:41

## Source code:

```
// User ForgotPassword
public IActionResult ForgotPassword()
{
    var saccount = HttpContext.Session.GetString("sAccount");
    if (saccount != null)
    {
        return RedirectToAction("Index", "Home");
    }
    return View();
}
```

```
[HttpPost]
0 references
public IActionResult ForgotPassword(Account acc)
{
    var accounts = JsonConvert.DeserializeObject<IEnumerable<Account>>(httpClient.GetStringAsync(uriAccount).Result);
    var account = accounts.SingleOrDefault(a => a.Email.Equals(acc.Email));
    try
    {
        if (acc.Email != null)
        {
            if (account != null)
            {
                Role role = JsonConvert.DeserializeObject<Role>(httpClient.GetStringAsync(uriRole + account.Id).Result);
                if (role.Position == 3)
                {
                    int random = rnd.Next(1000, 9999);
                    string code = BCrypt.Net.BCrypt.HashPassword(random.ToString());
                    account.Code = code;
                    var model = httpClient.PutAsJsonAsync(uriAccount, account).Result;
                    if (model.IsSuccessStatusCode)
                    {
                        string path = $"{this.Request.Scheme}://{this.Request.Host}{this.Request.PathBase}/User/Account/ResetPassword/?Co
                        Help.SendEmail.ResetPassword(account.Email, path).Wait();
                        return RedirectToAction("ForgotPasswordConfirm", "Account");
                    }
                }
            }
        }
    }
}
```

	Prepared by (Student) Project Group No:3	Approved by (Faculty)
Signature		
Date		

Design Plan:TARS Delivery	Document Name:Review 2	SWD/Form No.1
Effective Date:22/01/2021	Version:1.0	Page No:42

```

        }
        else
        {
            ViewBag.Msg = "Admin and employee cannot reset";
            return View();
        }
    }
    else
    {
        ViewBag.Msg = "Email wrong! Please check again!";
        return View();
    }
}
catch (Exception e)
{
    ViewBag.Msg = e.Message;
}
return View();
}

```

	Prepared by (Student) Project Group No:3	Approved by (Faculty)
Signature		
Date		

Design Plan:TARS Delivery	Document Name:Review 2	SWD/Form No.1
Effective Date:22/01/2021	Version:1.0	Page No:43

## 7. Sign up :

Register

Home / Register

Create Account

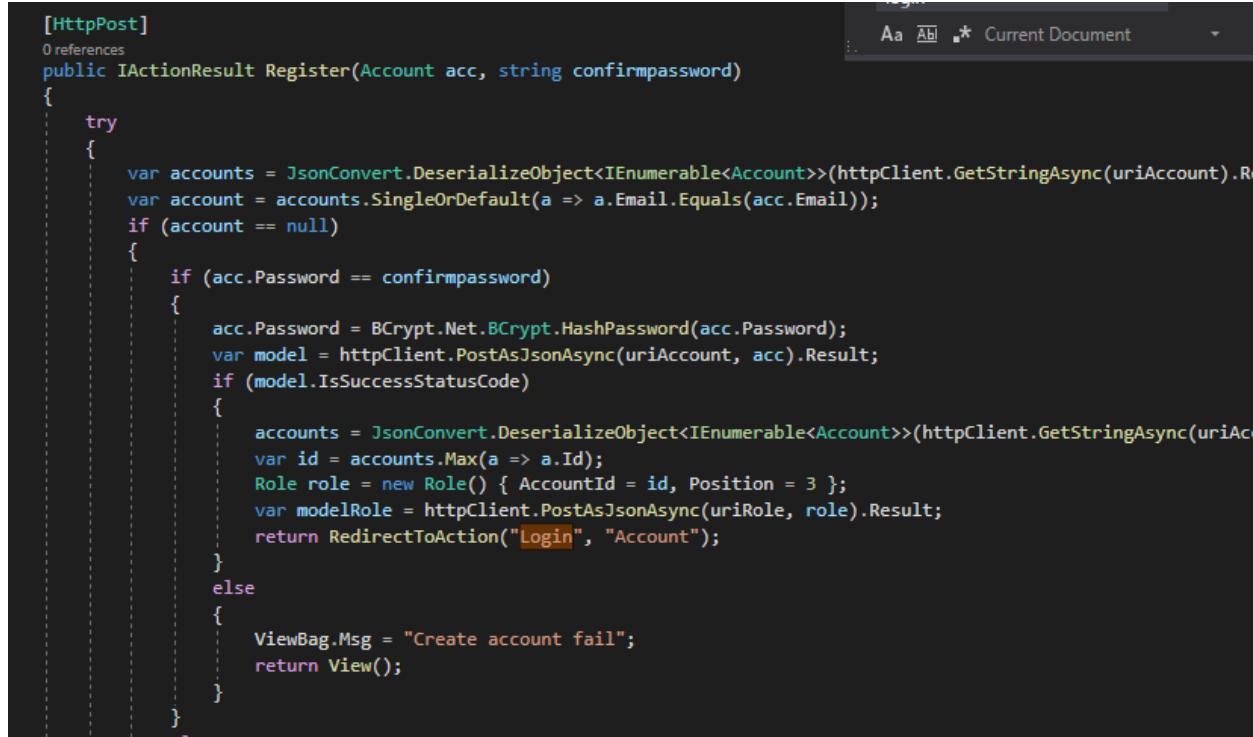
Fullscreen	Enter fullname	Date of birth
	1	dd/mm/yyyy 2 <input type="button" value=""/>
Email	Enter email address 3	
Phone	Enter phone 4	
Address	Enter address 5	
Password	Enter password 6	Confirm Password 7
<input type="button" value="Create Account"/> 8		
Have an account? Go to login 9		

No	Name	Type	Validation	Event	Description	Status
1	Full Name	TextBox		Focus	Input FullName	Enable
2	Date of birth	Date		Focus	Input Date	Enable
3	Email	TextBox		Focus	Input Email	Enable
4	Phone	TextBox		Focus	Phone	Enable
5	Address	TextBox		Focus	Input Address	Enable
6	Password	TextBox		Focus	Input password	Enable
7	Confirm Password	TextBox		Click	Confirm Password	Enable
8	Submit	Submit		Click	Submit	Enable
9	Go to Login	Hyperlink		Click	Go to Login	Enable

	Prepared by (Student)	Approved by (Faculty)
	Project Group No:3	
Signature		
Date		

Design Plan:TARS Delivery	Document Name:Review 2	SWD/Form No.1
Effective Date:22/01/2021	Version:1.0	Page No:44

## Source code:



```
[HttpPost]
public IActionResult Register(Account acc, string confirmPassword)
{
    try
    {
        var accounts = JsonConvert.DeserializeObject<IEnumerable<Account>>(httpClient.GetStringAsync(uriAccount).Result);
        var account = accounts.SingleOrDefault(a => a.Email.Equals(acc.Email));
        if (account == null)
        {
            if (acc.Password == confirmPassword)
            {
                acc.Password = BCrypt.Net.BCrypt.HashPassword(acc.Password);
                var model = httpClient.PostAsJsonAsync(uriAccount, acc).Result;
                if (model.IsSuccessStatusCode)
                {
                    accounts = JsonConvert.DeserializeObject<IEnumerable<Account>>(httpClient.GetStringAsync(uriAccount).Result);
                    var id = accounts.Max(a => a.Id);
                    Role role = new Role() { AccountId = id, Position = 3 };
                    var modelRole = httpClient.PostAsJsonAsync(uriRole, role).Result;
                    return RedirectToAction("Login", "Account");
                }
                else
                {
                    ViewBag.Msg = "Create account fail";
                    return View();
                }
            }
        }
    }
}
```

	Prepared by (Student) Project Group No:3	Approved by (Faculty)
Signature		
Date		

Design Plan:TARS Delivery	Document Name:Review 2	SWD/Form No.1
Effective Date:22/01/2021	Version:1.0	Page No:45

```

        }
    }
}
else
{
    ViewBag.Msg = "Email exists";
    return View();
}
catch (Exception e)
{
    ViewBag.Msg = e.Message;
}
return View();
}

```

```

// User Register
0 references
public IActionResult Register()
{
    var saccount = HttpContext.Session.GetString("sAccount");
    if (saccount != null)
    {
        return RedirectToAction("Index", "Home");
    }
    return View();
}

```

	Prepared by (Student) Project Group No:3	Approved by (Faculty)
Signature		
Date		

Design Plan:TARS Delivery	Document Name:Review 2	SWD/Form No.1
Effective Date:22/01/2021	Version:1.0	Page No:46

## 8.Bill History:

History		Home / History
---------	--	----------------

ID	Title	Name To	Address From	Address To	Total Price	Status
20210127	Package 02	Name To 02	Address From 02	Address To 02	\$324.32	Doing VPP
20210128	Package 02	Name To 02	Address From 02	Address To 02	\$324.32	Finish Branch
20210129	Package 04	Name To 04	Address From 04	Address To 04	\$524.32	Stored VPP
ID	Title	Name To	Address From	Address To	Total Price	Status

	Prepared by (Student) Project Group No:3	Approved by (Faculty)
Signature		
Date		

Design Plan:TARS Delivery	Document Name:Review 2	SWD/Form No.1
Effective Date:22/01/2021	Version:1.0	Page No:47

Admin :

## 1.Login

No	Name	Type	Validation	Event	Description	Status
1	Email	TextBox		Focus	Input Email	Enable
2	Password	TextBox		Focus	Input Password	Enable
3	Login	Submit		Click	Login	Enable

	Prepared by (Student)	Approved by (Faculty)
	Project Group No:3	
Signature		
Date		

Design Plan:TARS Delivery	Document Name:Review 2	SWD/Form No.1
Effective Date:22/01/2021	Version:1.0	Page No:48

## Source code:

```
public IActionResult Login()
{
    if (User.Identity.Name != null)
    {
        return RedirectToAction("Index", "Home");
    }
    return View();
}
```

```
[HttpPost]
0 references
public IActionResult Login(Account acc)
{
    var accounts = JsonConvert.DeserializeObject<IEnumerable<Account>>(httpClient.GetStringAsync(uriAcco
    if (!string.IsNullOrEmpty(acc.Email) && !string.IsNullOrEmpty(acc.Password))
    {
        var account = accounts.SingleOrDefault(a => a.Email.Equals(acc.Email) && BCrypt.Net.BCrypt.Verif
        if (account != null && account.Delete_at == null)
        {
            Role role = JsonConvert.DeserializeObject<Role>(httpClient.GetStringAsync(uriRole + account.
            ClaimsIdentity identity = null;
            bool isAuthenticated = false;
            if (role.Position == 1)
            {
                identity = new ClaimsIdentity(new[]
                {
                    new Claim("AccountId", account.Id.ToString()),
                    new Claim(ClaimTypes.Name, account.Email),
                    new Claim(ClaimTypes.Role, "Admin"),
                }, CookieAuthenticationDefaults.AuthenticationScheme);
                isAuthenticated = true;
            }
        }
    }
}
```

	Prepared by (Student) Project Group No:3	Approved by (Faculty)
Signature		
Date		

Design Plan:TARS Delivery	Document Name:Review 2	SWD/Form No.1
Effective Date:22/01/2021	Version:1.0	Page No:49

```

        else
        {
            ViewBag.Msg = "User cannot login";
            return View();
        }
    else
    {
        ViewBag.Msg = "Email or password wrong! Please try again";
        return View();
    }
}
return View();
}

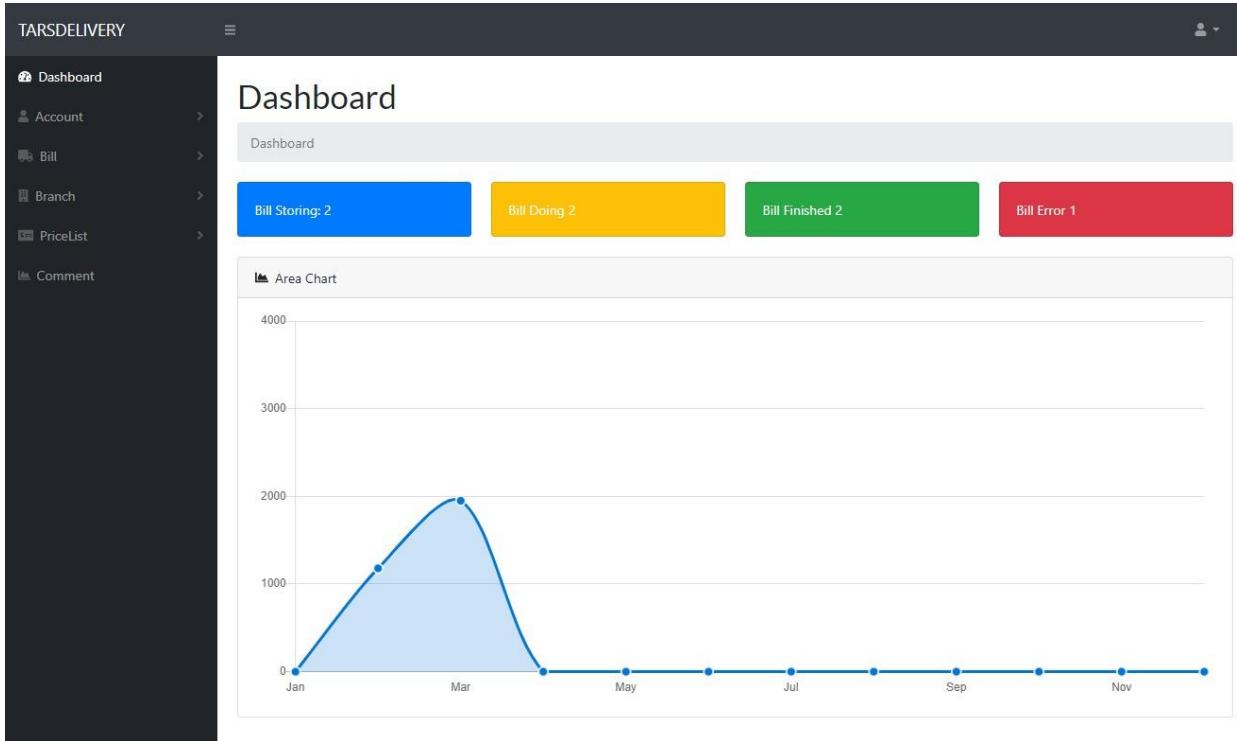
if (role.Position == 2)
{
    identity = new ClaimsIdentity(new[]
    {
        new Claim("AccountId", account.Id.ToString()),
        new Claim(ClaimTypes.Name,account.Email),
        new Claim(ClaimTypes.Role,"Manager"),
    }, CookieAuthenticationDefaults.AuthenticationScheme);
    isAuthenticated = true;
}
if (isAuthenticated)
{
    var authProperties = new AuthenticationProperties
    {
        AllowRefresh = true,
        ExpiresUtc = DateTimeOffset.Now.AddDays(1),
        IsPersistent = true,
    };
    var principal = new ClaimsPrincipal(identity);
    var login = HttpContext.SignInAsync(CookieAuthenticationDefaults.AuthenticationScheme, p
    return RedirectToAction("Index", "Home", new { area = "Admin" }));
}

```

	Prepared by (Student) Project Group No:3	Approved by (Faculty)
Signature		
Date		

Design Plan:TARS Delivery	Document Name:Review 2	SWD/Form No.1
Effective Date:22/01/2021	Version:1.0	Page No:50

## 2.Dashboard:



	Prepared by (Student) Project Group No:3	Approved by (Faculty)
Signature		
Date		

Design Plan:TARS Delivery	Document Name:Review 2	SWD/Form No.1
Effective Date:22/01/2021	Version:1.0	Page No:51

## Source code:

```
[Authorize(Roles = "Admin, Manager")]
0 references
public IActionResult Index()
{
    var models = JsonConvert.DeserializeObject<IEnumerable<BillPackage>>(httpClient.GetStringAsync($"{{url}}"));
    if (User.IsInRole("Admin"))
    {
        ViewBag.Storing = models.Where(m => m.GetPackage.Status == 1 && m.GetPackage.Delete_at == null);
        ViewBag.Doing = models.Where(m => m.GetPackage.Status == 2 && m.GetPackage.Delete_at == null).Count();
        ViewBag.Finished = models.Where(m => m.GetPackage.Status == 3 && m.GetPackage.Delete_at == null);
        ViewBag.Error = models.Where(m => m.GetPackage.Status == 4 && m.GetPackage.Delete_at == null).Count();

        double[] totalMonth = new double[12];
        for (int i = 0; i < totalMonth.Length; i++)
        {
            var month = models.Where(e => e.GetPackage.Create_at.Month == i + 1 && e.GetPackage.Delete_at == null).Count();
            totalMonth[i] += Math.Round(month, 2, MidpointRounding.AwayFromZero);
        }
        ViewBag.Data = JsonConvert.SerializeObject(totalMonth);
    }
}
```

```
if (User.IsInRole("Manager"))
{
    var accounts = JsonConvert.DeserializeObject<IEnumerable<Account>>(httpClient.GetStringAsync(url));
    var account = accounts.SingleOrDefault(a => a.Email.Equals(User.Identity.Name));
    ViewBag.Storing = models.Where(m => m.GetPackage.Status == 1 && m.GetPackage.Delete_at == null && m.GetPackage.Account_id == account.Id);
    ViewBag.Doing = models.Where(m => m.GetPackage.Status == 2 && m.GetPackage.Delete_at == null && m.GetPackage.Account_id == account.Id);
    ViewBag.Finished = models.Where(m => m.GetPackage.Status == 3 && m.GetPackage.Delete_at == null && m.GetPackage.Account_id == account.Id);
    ViewBag.Error = models.Where(m => m.GetPackage.Status == 4 && m.GetPackage.Delete_at == null && m.GetPackage.Account_id == account.Id);

    double[] totalMonth = new double[12];
    for (int i = 0; i < totalMonth.Length; i++)
    {
        var month = models.Where(e => e.GetPackage.Create_at.Month == i + 1 && e.GetPackage.Delete_at == null && e.GetPackage.Account_id == account.Id).Count();
        totalMonth[i] += month;
    }
    ViewBag.Data = JsonConvert.SerializeObject(totalMonth);
}

return View();
}
```

	Prepared by (Student) Project Group No:3	Approved by (Faculty)
Signature		
Date		

Design Plan:TARS Delivery	Document Name:Review 2	SWD/Form No.1
Effective Date:22/01/2021	Version:1.0	Page No:52

### 3.Account

TARSDELIVERY										
Dashboard		Show 10 entries	Search:							
Account		Account List								
Account List										
ID	FullName	Email	Role	Status	Created					
2	Pham Xuan Hoai	hoaixp@gmail.com	Employee	InActive	3/6/2021 9:06:26 PM	1 Details 2 Reset Password Active 3				
3	Le Hoang Anh Tu	tulha@gmail.com	User	Active	3/6/2021 9:06:26 PM	Details Inactive				
4	Ngo Trung Kien	kiennt@gmail.com	User	Active	3/6/2021 9:06:26 PM	Details Inactive				
5	ngotrungkien4798@gmail.com	ngotrungkien4798@gmail.com	Employee	Active	3/6/2021 9:52:29 PM	Details Reset Password Inactive				
ID	FullName	Email	Role	Status	Created					

No	Name	Type	Validation	Event	Description	Status
1	Details	Hyperlink		Click	Click	Enable
2	Reset Password	Date		Click	Reset Password default	Enable
3	Active	button		Click	Click Active and Inactive	Enable

Prepared by (Student)	Approved by (Faculty)
Project Group No:3	
Signature	
Date	

Design Plan:TARS Delivery	Document Name:Review 2	SWD/Form No.1
Effective Date:22/01/2021	Version:1.0	Page No:53

## Source code:

```
[Authorize(Roles = "Admin")]
0 references
public IActionResult Index()
{
    var model = JsonConvert.DeserializeObject<IEnumerable<Account>>(httpClient.GetStringAsync(uriAccount).Result);
    var accounts = model.Where(a => !a.Email.Equals(User.Identity.Name)).ToList();
    var roles = JsonConvert.DeserializeObject<IEnumerable<Role>>(httpClient.GetStringAsync(uriRole).Result);
    ViewBag.Role = roles;
    return View(accounts);
}
```

	Prepared by (Student) Project Group No:3	Approved by (Faculty)
Signature		
Date		

Design Plan:TARS Delivery	Document Name:Review 2	SWD/Form No.1
Effective Date:22/01/2021	Version:1.0	Page No:54

#### 4.Details Account:

The screenshot shows a user profile page from a system named TARSDELIVERY. On the left is a sidebar with navigation links: Dashboard, Account (selected), Bill, Branch, PriceList, and Comment. The main area features a circular profile picture of a man with a beard. To the right of the picture is a table displaying the following information:

Fullname:	Pham Xuan Hoai
Email:	hoaixp@gmail.com
Date of Birth	3/21/1998 12:00:00 AM
Phone	0933691822
Address	Binh Thanh
Role	Employee
Branch:	Branch 01

At the bottom left of the main area, there is a link labeled "Back to list".

	Prepared by (Student) Project Group No:3	Approved by (Faculty)
Signature		
Date		

Design Plan:TARS Delivery	Document Name:Review 2	SWD/Form No.1
Effective Date:22/01/2021	Version:1.0	Page No:55

## Source code:

```
// Admin Detail
[Authorize(Roles = "Admin")]
0 references
public IActionResult Details(int id)
{
    var model = JsonConvert.DeserializeObject<IEnumerable<Account>>(httpClient.GetStringAsync(uriAccount));
    var account = model.SingleOrDefault(a => a.Id == id);
    var roles = JsonConvert.DeserializeObject<IEnumerable<Role>>(httpClient.GetStringAsync(uriRole).Result);
    var role = roles.SingleOrDefault(r => r.AccountId.Equals(account.Id));
    ViewBag.Role = role;
    var branches = JsonConvert.DeserializeObject<IEnumerable<Branch>>(httpClient.GetStringAsync(uriBranch));
    var branch = branches.SingleOrDefault(r => r.Id.Equals(account.BranchId));
    ViewBag.Branch = branch;
    return View(account);
}
```

	Prepared by (Student) Project Group No:3	Approved by (Faculty)
Signature		
Date		

Design Plan:TARS Delivery	Document Name:Review 2	SWD/Form No.1
Effective Date:22/01/2021	Version:1.0	Page No:56

## 5.Create Account:

The screenshot shows a user interface for creating a new employee account. The left side features a dark sidebar with navigation links: Dashboard, Account, Bill, Branch, PriceList, and Comment. The main content area is titled "Create employee". It contains two input fields: "Email" with the placeholder "Enter email address" and a red validation mark "1", and "Branch" with the placeholder "Branch 01" and a red validation mark "2". At the bottom right is a blue button labeled "Create Account".

No	Name	Type	Validation	Event	Description	Status
1	Email	TextBox		Focus	Input Email	Enable
2	Branch	option		Click	Click to show option	Enable

	Prepared by (Student) <b>Project Group No:3</b>	Approved by (Faculty)
Signature		
Date		

Design Plan:TARS Delivery	Document Name:Review 2	SWD/Form No.1
Effective Date:22/01/2021	Version:1.0	Page No:57

## Source code:

```
// Admin Create account for employee
0 references
public IActionResult Create()
{
    var branches = JsonConvert.DeserializeObject<IEnumerable<Branch>>(httpClient.GetStringAsync(uriBranch).Result);
    ViewBag.List = new SelectList(branches, "Id", "Name");
    return View();
}
```

```
[HttpPost]
0 references
public IActionResult Create(Account acc)
{
    try
    {
        var branches = JsonConvert.DeserializeObject<IEnumerable<Branch>>(httpClient.GetStringAsync(uriBranch).Result);
        ViewBag.List = new SelectList(branches, "Id", "Name");
        string tempPassword = "12345678";
        var accounts = JsonConvert.DeserializeObject<IEnumerable<Account>>(httpClient.GetStringAsync(uriAccount).Result);
        var account = accounts.SingleOrDefault(a => a.Email.Equals(acc.Email));
        if (account == null)
        {
            acc.Password = BCrypt.Net.BCrypt.HashPassword(tempPassword);
            acc.FullName = acc.Email;
            acc.Birthday = DateTime.Now;
            var model = httpClient.PostAsJsonAsync(uriAccount, acc).Result;
            if (model.IsSuccessStatusCode)
            {
                accounts = JsonConvert.DeserializeObject<IEnumerable<Account>>(httpClient.GetStringAsync(uriAccount).Result);
                var id = accounts.Max(a => a.Id);
                Role role = new Role() { AccountId = id, Position = 2 };
                var modelRole = httpClient.PostAsJsonAsync(uriRole, role).Result;
                string path = $"{this.Request.Scheme}://{this.Request.Host}{this.Request.PathBase}/Admin/Account/Login";
                Help.SendEmail.EmployeeLogin(acc.Email, tempPassword, path).Wait();
                return RedirectToAction("Index", "Account");
            }
        }
    }
```

```
    else
    {
        ViewBag.Msg = "Email is exists";
        return View();
    }
}
catch (Exception e)
{
    ViewBag.Msg = e.Message;
}
return View();
}
```

	Prepared by (Student) Project Group No:3	Approved by (Faculty)
Signature		
Date		

Design Plan:TARS Delivery	Document Name:Review 2	SWD/Form No.1
Effective Date:22/01/2021	Version:1.0	Page No:58

## 6.Bill:

The screenshot shows a software application window titled "TARSDELIVERY". On the left is a dark sidebar with navigation links: Dashboard, Account, Bill, Bill List (selected), Create Bill, Branch, PriceList, and Comment. The main content area has a header "Working" and "Stop Working". Below is a table with 7 rows of data. The table columns are: ID, Title, Name, Email, Type, Salary, Status, and Created. Each row contains a set of buttons: Details, Update, and Delete. The table footer shows "Showing 1 to 7 of 7 entries" and navigation buttons for Previous, Next, and page number 1.

ID	Title	Name	Email	Type	Salary	Status	Created	Action
20210126	Package 01	Name From 01	mail01@gmail.com	Mail	\$652.32	Stored VPP	3/9/2021 9:27:56 AM	<button>Details</button> <button>Update</button> <button>Delete</button>
20210127	Package 02	Name From 02	tulha@gmail.com	Mail	\$324.32	Doing VPP	3/9/2021 9:27:56 AM	<button>Details</button> <button>Update</button>
20210128	Package 02	Name From 02	tulha@gmail.com	Mail	\$324.32	Finish Branch	3/9/2021 9:27:56 AM	<button>Details</button>
20210129	Package 04	Name From 04	tulha@gmail.com	Mail	\$524.32	Stored VPP	3/9/2021 9:27:56 AM	<button>Details</button> <button>Update</button> <button>Delete</button>
20210130	Package 05	Name From 05	kiennt@gmail.com	Mail	\$654.32	Doing VPP	3/9/2021 9:27:56 AM	<button>Details</button> <button>Update</button>
20210131	Package 06	Name From 06	kiennt@gmail.com	Mail	\$324.32	Finish Branch	3/9/2021 9:27:56 AM	<button>Details</button>
20210132	Package 07	Name From 07	kiennt@gmail.com	Mail	\$324.32	Error VPP	3/9/2021 9:27:56 AM	<button>Delete</button>

Name	Type	Validation	Event	Description	Status
Details	Button		Click	Click to redirect to Details Page	Enable
Update	Button		Click	Click to redirect to Update Page	Enable
Delete	Button		Click	Click to Delete Bill	Enable

Prepared by (Student)	Approved by (Faculty)
Project Group No:3	
Signature	
Date	

Design Plan:TARS Delivery	Document Name:Review 2	SWD/Form No.1
Effective Date:22/01/2021	Version:1.0	Page No:59

## Source code:

```
[Authorize(Roles = "Admin, Manager")]
0 references
public IActionResult Index()
{
    var models = JsonConvert.DeserializeObject<IEnumerable<BillPackage>>(httpClient.GetStringAsync($"{uriBill}"));

    var modelsWorking = models.Where(m => m.GetBill.Delete_at == null && m.GetPackage.Delete_at == null);
    var modelsStopWorking = models.Where(m => m.GetBill.Delete_at != null && m.GetPackage.Delete_at != null);

    if (User.IsInRole("Manager"))
    {
        var userId = int.Parse(User.FindFirst("AccountId").Value);
        var account = JsonConvert.DeserializeObject<Account>(httpClient.GetStringAsync($"{uriAccount}/{userId}"));

        modelsWorking = models.Where(m => m.GetBill.Delete_at == null && m.GetPackage.Delete_at == null);
        modelsStopWorking = models.Where(m => m.GetBill.Delete_at != null && m.GetPackage.Delete_at != null);
    }

    ViewBag.ModelsDeleted = modelsStopWorking;

    return View(modelsWorking);
}
```

	Prepared by (Student) Project Group No:3	Approved by (Faculty)
Signature		
Date		

Design Plan:TARS Delivery	Document Name:Review 2	SWD/Form No.1
Effective Date:22/01/2021	Version:1.0	Page No:60

## 7.Details Bill:

**TARSDELIVERY**

**Amount**  
\$652.32

**ID:** 20210126  
**FROM:** Address From 01      **TO:** Address To 01  
**ZIPCODE:** 1234      **NAME:** Name To 01  
**NAME:** Name From 01      **DATE:** 3/6/2021 9:06:26 PM  
**EMAIL:** mail01@gmail.com

#	Package	Type	Distance	Weight	Method	Message
20210126	Package 01	Mail	304.23 KM	1500 GR	VPP	Message 01

	Prepared by (Student) <b>Project Group No:3</b>	Approved by (Faculty)
<b>Signature</b>		
<b>Date</b>		

Design Plan:TARS Delivery	Document Name:Review 2	SWD/Form No.1
Effective Date:22/01/2021	Version:1.0	Page No:61

## Source code:

```
[Authorize(Roles = "Admin, Manager")]
0 references
public IActionResult Details(int billId)
{
    var models = JsonConvert.DeserializeObject<IEnumerable<BillPackage>>(httpClient.GetStringAsync($"{uriBills}/GetBillPackages").Result);
    var modelBill = models.SingleOrDefault(m => m.GetBill.Id == billId);

    return View(modelBill);
}
```

	Prepared by (Student) Project Group No:3	Approved by (Faculty)
Signature		
Date		

Design Plan:TARS Delivery	Document Name:Review 2	SWD/Form No.1
Effective Date:22/01/2021	Version:1.0	Page No:62

## 8.Update Bill:

TARSDELIVERY

Details the parcel

Package Name: Package 01

Name: Name From 01 Email: mail01@gmail.com

Recipient's Name: Name To 01 Type: Mail ZipCode: 1234

Address Origin: Address From 01 Address Destination: Address To 01 Distance: 304.23 Status: Doing

Message: Message 01

Total Price: \$652.32

Confirm

	Prepared by (Student)	Approved by (Faculty)
	<b>Project Group No:3</b>	
Signature		
Date		

Design Plan:TARS Delivery	Document Name:Review 2	SWD/Form No.1
Effective Date:22/01/2021	Version:1.0	Page No:63

## Source code:

```
[Authorize(Roles = "Admin, Manager")]
0 references
public IActionResult Update(int packageId)
{
    var priceList = JsonConvert.DeserializeObject<IEnumerable<PriceList>>(httpClient.GetStringAsync($"{uriBills}/GetPriceLists").Result);
    ViewBag.PriceLists = new SelectList(priceList.Where(m => !m.Name.Equals("VPP") && m.Delete_at == null), "Name", "Name");

    var models = JsonConvert.DeserializeObject<IEnumerable<Branch>>(httpClient.GetStringAsync($"{uriBranches}").Result);
    ViewBag.Branchs = new SelectList(models, "Id", "Name");

    var model = JsonConvert.DeserializeObject<Package>(httpClient.GetStringAsync($"{uriPackages}/GetPackage/{packageId}").Result);
    ViewBag.Distance = model.Distance;
    ViewBag.TotalPrice = model.TotalPrice;

    ViewBag.BranchId = "";

    if (User.IsInRole("Manager"))
    {
        ViewBag.BranchId = "Manager";
    }
    else if (User.IsInRole("Admin"))
    {
        ViewBag.BranchId = "Admin";
    }

    return View(model);
}
```

	Prepared by (Student) Project Group No:3	Approved by (Faculty)
Signature		
Date		

Design Plan:TARS Delivery	Document Name:Review 2	SWD/Form No.1
Effective Date:22/01/2021	Version:1.0	Page No:64

## 9.Create Bill:

TARSDELIVERY

Create the parcel

Package Name: 1615257972PN

Name: Your Name Email: Your Email

Recipient's Name: Recipient's Name Type: Package ZipCode: ZipCode Branch: Branch 01

Address Origin: Enter a location Address Destination: Enter a location Weight(gram): 1000 Method: Normal

Message: Message

Total Price: \$1.21

Confirm

Name	Type	Validation	Event	Description	Status
Name	Textbox		Focus	Input Bill Name	Enable
Email	Textbox		Focus	Input Email Name	Enable
Recipient's Name	Textbox		Focus	Input Recipient's Name	Enable
Type	Option		Click	Click to Choice Type	Enable
ZipCode	Textbox		Focus	Input ZipCode	Enable
Branch	Option		Click	Click to Choice Branch	Enable
Address Origin	Textbox		Focus	Input Address Origin	Enable
Address Destination	Textbox		Focus	Input Address Destination	Enable
Weight(gram)	Textbox		Focus	Input Weight	Enable
Method	Option		Click	Click to Choice Method	Enable
Message	Textbox		Focus	Input Message	Enable
Confirm	Button		Click	Click to Confirm the Bill	Enable

Prepared by (Student)	Approved by (Faculty)
Project Group No:3	
Signature	
Date	

Design Plan:TARS Delivery	Document Name:Review 2	SWD/Form No.1
Effective Date:22/01/2021	Version:1.0	Page No:65

## 10.Branch:

TARSDELIVERY						
<span>Show 10 entries</span> <span>Search:</span>						
	Name	Address	Phone	Inactive	Action	
1	Branch 01	Address 01	0901858004		<span>Edit</span> <span>Inactive</span>	
2	Branch 02	Address 02	0901858004	3/9/2021 9:35:53 AM	<span>Edit</span> <span>Active</span>	
Id	Name	Address	Phone	Inactive	Action	

Showing 1 to 2 of 2 entries

Previous 1 Next

Name	Type	Validation	Event	Description	Status
Edit	Button		Click	Click to redirect to Edit Page	Enable
Active	Button		Click	Click to Active Branch	Enable
Inactive	Button		Click	Click to Inactive Branch	Enable

	Prepared by (Student)	Approved by (Faculty)
	Project Group No:3	
Signature		
Date		

Design Plan:TARS Delivery	Document Name:Review 2	SWD/Form No.1
Effective Date:22/01/2021	Version:1.0	Page No:66

## Source code:

```
[Authorize(Roles = "Admin")]
0 references
public IActionResult Index()
{
    var model = JsonConvert.DeserializeObject<IEnumerable<Branch>>(httpClient.GetStringAsync(uriBranch).
    var branches = model.ToList();
    return View(branches);
}
0 references
```

	Prepared by (Student) Project Group No:3	Approved by (Faculty)
Signature		
Date		

Design Plan:TARS Delivery	Document Name:Review 2	SWD/Form No.1
Effective Date:22/01/2021	Version:1.0	Page No:67

## 11.Edit Branch:

TARSDELIVERY

Edit Branch

Name  
Branch 01

Address  
Address 01

Phone  
0901858004

Save

[Back to List](#)

	Prepared by (Student) <b>Project Group No:3</b>	Approved by (Faculty)
Signature		
Date		

Design Plan:TARS Delivery	Document Name:Review 2	SWD/Form No.1
Effective Date:22/01/2021	Version:1.0	Page No:68

## Source code:

```

0 REFERENCES
public IActionResult Edit(int id)
{
    var model = JsonConvert.DeserializeObject<Branch>(httpClient.GetStringAsync($"{uriBranch}GetBranch/{id}").Result);
    return View(model);
}

[HttpPost]
[ValidateAntiForgeryToken]
0 references
public IActionResult Edit(Branch branch)
{
    try
    {
        var model = httpClient.PutAsJsonAsync($"{uriBranch}", branch).Result;
        if (model.IsSuccessStatusCode)
        {
            return RedirectToAction("Index");
        }
        ViewBag.Msg = "Error Update!";
    }
    catch (Exception)
    {
        ViewBag.Msg = "Error!";
    }
    return View();
}

```

	Prepared by (Student) Project Group No:3	Approved by (Faculty)
Signature		
Date		

Design Plan:TARS Delivery	Document Name:Review 2	SWD/Form No.1
Effective Date:22/01/2021	Version:1.0	Page No:69

## 12.PriceList:

The screenshot shows a user interface for managing price lists. The left sidebar has a dark theme with navigation links for Dashboard, Account, Bill, Branch, PriceList, and Comment. The main content area has a light background with a table header. The table columns are Id, Name, Price, Create Time, Inactive Time, and Action. The table body contains four rows of data. Each row has an 'Edit' button in the Action column. The second row has an 'Inactive' button. The third row has an 'Inactive' button. The fourth row has an 'Active' button.

Id	Name	Price	Create Time	Inactive Time	Action
1	VPP	1.08	3/9/2021 9:27:56 AM		<button>Edit</button>
2	Normal	1.21	3/9/2021 9:27:56 AM		<button>Edit</button> <button>Inactive</button>
3	Speed	1.35	3/9/2021 9:27:56 AM		<button>Edit</button> <button>Inactive</button>
4	Courier	1.46	3/9/2021 9:27:56 AM	3/9/2021 9:30:50 AM	<button>Edit</button> <button>Active</button>

Name	Type	Validation	Event	Description	Status
Edit	Button		Click	Click to redirect to Edit Page	Enable
Active	Button		Click	Click to Active Item	Enable
Inactive	Button		Click	Click to Inactive Item	Enable

	Prepared by (Student)	Approved by (Faculty)
	Project Group No:3	
Signature		
Date		

Design Plan:TARS Delivery	Document Name:Review 2	SWD/Form No.1
Effective Date:22/01/2021	Version:1.0	Page No:70

## Source Code:

```
[Authorize(Roles = "Admin")]
0 references
public IActionResult Index()
{
    var res = JsonConvert.DeserializeObject<IEnumerable<PriceList>>(
        httpClient.GetStringAsync(uri).Result);
    return View(res);
}
```

	Prepared by (Student) Project Group No:3	Approved by (Faculty)
Signature		
Date		

Design Plan:TARS Delivery	Document Name:Review 2	SWD/Form No.1
Effective Date:22/01/2021	Version:1.0	Page No:71

## 13.Edit PriceList:

TARSDELIVERY

Dashboard

Account

Bill

Branch

PriceList

Comment

Edit Pricelist

Name: Normal

PriceWeight: 1.21

Save

Back to List

	Prepared by (Student)	Approved by (Faculty)
	<b>Project Group No:3</b>	
<b>Signature</b>		
<b>Date</b>		

Design Plan:TARS Delivery	Document Name:Review 2	SWD/Form No.1
Effective Date:22/01/2021	Version:1.0	Page No:72

## Source Code:

```

0 references
public IActionResult Edit(string name)
{
    var res = JsonConvert.DeserializeObject<IEnumerable<PriceList>>(httpClient.GetStringAsync(uri).Result);
    var model = res.SingleOrDefault(e => e.Name.Equals(name));
    return View(model);
}

[HttpPost]
0 references
public IActionResult Edit(PriceList priceList)
{
    try
    {
        var res = JsonConvert.DeserializeObject<IEnumerable<PriceList>>(httpClient.GetStringAsync(uri).Result);
        var modelOld = res.SingleOrDefault(e => e.Id == priceList.Id);
        var exist = res.SingleOrDefault(e => e.Name.Equals(priceList.Name) && e.Id != priceList.Id);
        if (exist == null)
        {
            priceList.Create_at = modelOld.Create_at;
            if (ModelState.IsValid)
            {
                var model = httpClient.PutAsJsonAsync<PriceList>(uri, priceList).Result;
                if (model.IsSuccessStatusCode)
                {
                    return RedirectToAction("Index", "PriceList");
                }
            }
            else
            {
                return View(modelOld);
            }
        }
        else if ((modelOld.PriceDistance != priceList.PriceDistance && exist == null) || (modelOld.PriceWeight != priceList.PriceWeight &
        {
            priceList.Create_at = modelOld.Create_at;
            if (ModelState.IsValid)
            {
                var model = httpClient.PutAsJsonAsync<PriceList>(uri, priceList).Result;
                if (model.IsSuccessStatusCode)
                {
                    return RedirectToAction("Index", "PriceList");
                }
            }
            else
            {
                return View(modelOld);
            }
        }
        else
        {
            ViewBag.Msg = "Name is exists";
            return View(modelOld);
        }
    }
    catch (Exception e)
    {
        ModelState.AddModelError(string.Empty, e.Message);
    }
    return View();
}

```

	Prepared by (Student) Project Group No:3	Approved by (Faculty)
Signature		
Date		

Design Plan:TARS Delivery	Document Name:Review 2	SWD/Form No.1
Effective Date:22/01/2021	Version:1.0	Page No:73

## 14.Create PriceList:

The screenshot shows the 'Create Pricelist' page within the TARSDELIVERY application. The left sidebar contains a navigation menu with items: Dashboard, Account, Bill, Branch, PriceList (which is currently selected), and Comment. The main content area is titled 'Create Pricelist' and contains two input fields: 'Name' and 'PriceWeight'. Below these fields is a blue 'Create' button.

Name	Type	Validation	Event	Description	Status
Name	Textbox		Focus	Input Item Name	Enable
PriceWeight	Textbox		Focus	Input Price Weight	Enable
Create	Button		Click	Click to Create Price Item	Enable

	Prepared by (Student) Project Group No:3	Approved by (Faculty)
Signature		
Date		

Design Plan:TARS Delivery	Document Name:Review 2	SWD/Form No.1
Effective Date:22/01/2021	Version:1.0	Page No:74

## 15.Comment:

TARSDELIVERY					
Dashboard		Search: <input type="text"/>			Logout
Account					
Bill					
Branch					
PriceList					
Comment					
Show 10 entries					
Id		Full Name	Email	Create_at	Action
1		Taylor Swift	Taylor@gmail.com	3/9/2021 9:27:56 AM	<button>Details</button> <button>Delete</button>
2		Selena Gomez	GomezSinger@gmail.com	3/9/2021 9:27:56 AM	<button>Details</button> <button>Delete</button>
3		Halsey	HalseyShazam@gmail.com	3/9/2021 9:27:56 AM	<button>Details</button> <button>Delete</button>
4		Alan Walker	AlanWalker@gmail.com	3/9/2021 9:27:56 AM	<button>Details</button> <button>Delete</button>
Id		Full Name	Email	Create_at	Action
Showing 1 to 4 of 4 entries					
<a href="#">Previous</a> <a href="#">1</a> <a href="#">Next</a>					

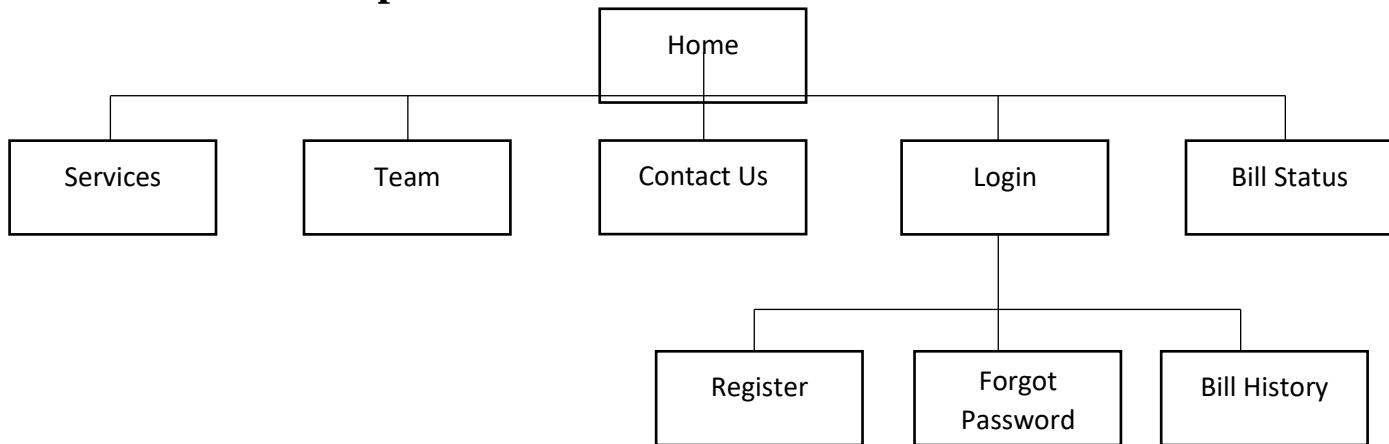
Name	Type	Validation	Event	Description	Status
Details	Button		Click	Click to redirect to Details Page	Enable
Delete	Button		Click	Click to Delete Comment	Enable

Prepared by (Student)	Approved by (Faculty)
Project Group No:3	
Signature	
Date	

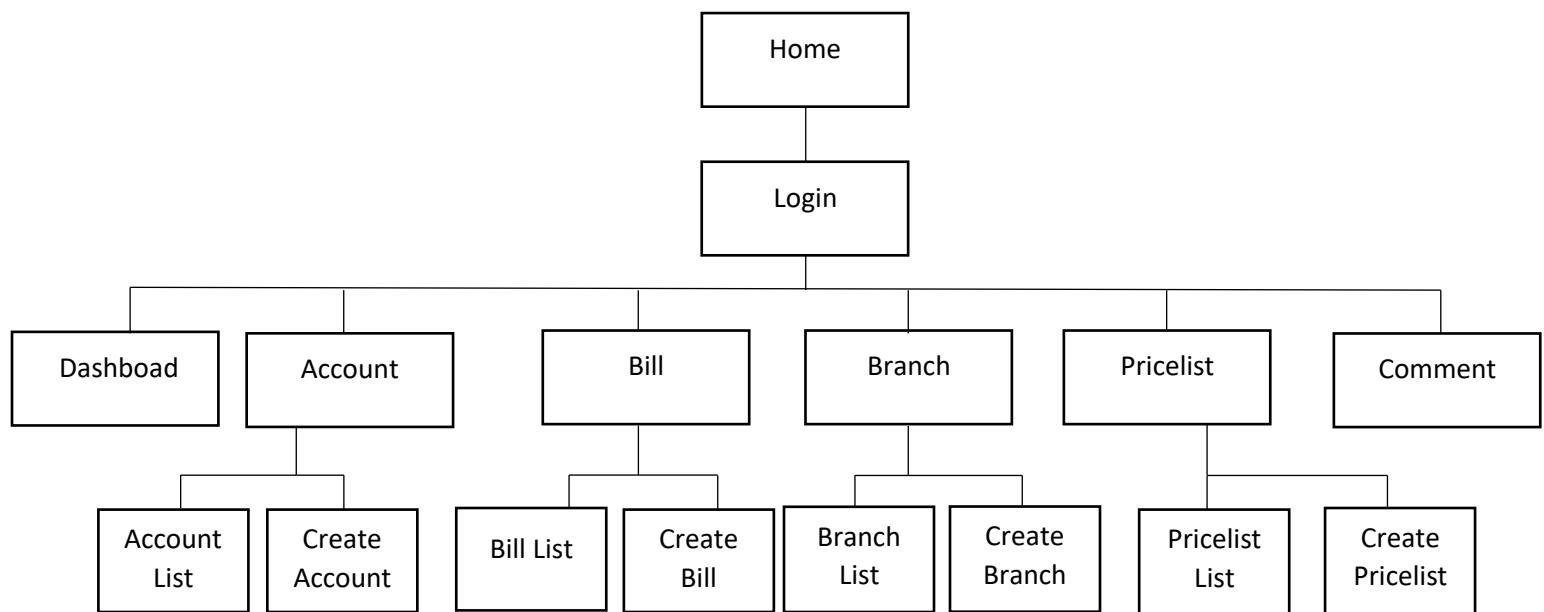
Design Plan:TARS Delivery	Document Name:Review 2	SWD/Form No.1
Effective Date:22/01/2021	Version:1.0	Page No:75

## • SiteMap

### - User Sitemap



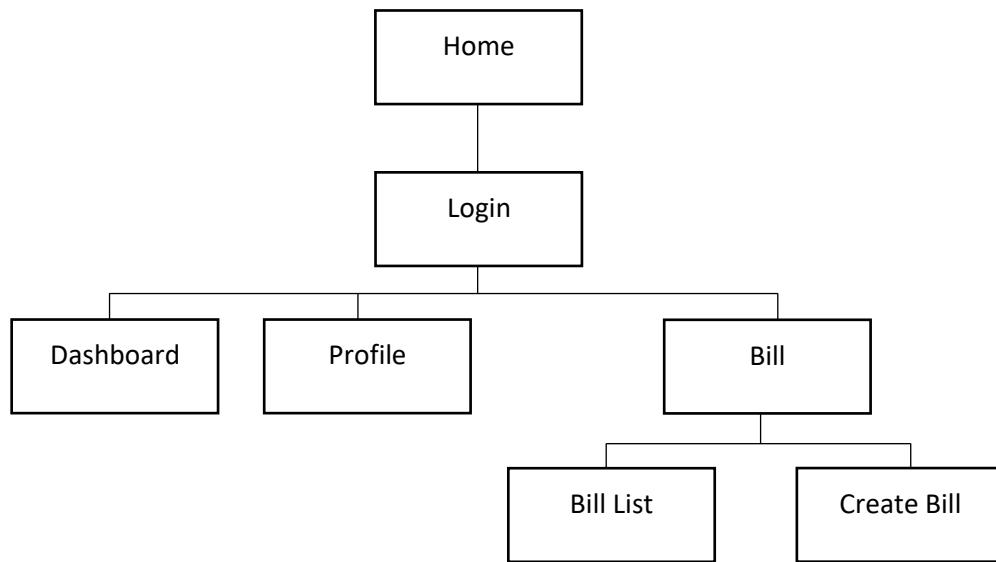
### - Admin Sitemap



	Prepared by (Student) Project Group No:3	Approved by (Faculty)
Signature		
Date		

Design Plan:TARS Delivery	Document Name:Review 2	SWD/Form No.1
Effective Date:22/01/2021	Version:1.0	Page No:76

## - Employees Sitemap



	Prepared by (Student) Project Group No:3	Approved by (Faculty)
Signature		
Date		

Design Plan:TARS Delivery	Document Name:Review 2	SWD/Form No.1
Effective Date:22/01/2021	Version:1.0	Page No:77

- **API Swagger:**
  - **Account**

**Account**

GET /api/Account
PUT /api/Account
POST /api/Account
GET /api/Account/{id}

- **Bills**

**Bills**

GET /api/Bills/GetBillPackages
GET /api/Bills/GetPriceLists
PUT /api/Bills/DeleteBill/{id}
PUT /api/Bills/UpdateBill
POST /api/Bills/CreateBill
PUT /api/Bills/UndoBill/{id}
GET /api/Bills/GetBill/{id}

- **Branch**

**Branch**

GET /api/Branch
POST /api/Branch
PUT /api/Branch
GET /api/Branch/GetBranch/{id}
PUT /api/Branch/DeleteBranch/{id}

	Prepared by (Student) Project Group No:3	Approved by (Faculty)
Signature		
Date		

Design Plan:TARS Delivery	Document Name:Review 2	SWD/Form No.1
Effective Date:22/01/2021	Version:1.0	Page No:78

## - Comment

**Comment**

GET	/api/Comment
POST	/api/Comment
GET	/api/Comment/{id}
PUT	/api/Comment/DeleteComment/{id}

## - Packages

**Packages**

PUT	/api/Packages/DeletePackage/{id}
GET	/api/Packages/GetPackage/{id}
PUT	/api/Packages/UpdatePackage
POST	/api/Packages/CreatePackage
GET	/api/Packages/GetNewPackage
PUT	/api/Packages/UndoPackage/{id}
GET	/api/Packages

## - PriceList

**PriceList**

GET	/api/PriceList
PUT	/api/PriceList
POST	/api/PriceList
PUT	/api/PriceList/DeletePriceList/{id}
GET	/api/PriceList/{name}

	Prepared by (Student) Project Group No:3	Approved by (Faculty)
Signature		
Date		

Design Plan:TARS Delivery	Document Name:Review 2	SWD/Form No.1
Effective Date:22/01/2021	Version:1.0	Page No:79

## - Role

**Role**

- GET /api/Role**
- POST /api/Role**
- GET /api/Role/{accountid}**

## - Services

**Services**

- GET /api/Services/GetPriceList/{name}**
- GET /api/Services/GetNewPackage**
- POST /api/Services/CreateBill**
- POST /api/Services/CreatePackage**

## - Schemas

**Schemas**

- Branch >
- Package >
- Bill >
- Role >
- Account >
- Comment >
- PriceList >

	Prepared by (Student) Project Group No:3	Approved by (Faculty)
Signature		
Date		

Design Plan:TARS Delivery	Document Name:Review 2	SWD/Form No.1
Effective Date:22/01/2021	Version:1.0	Page No:80

## Task Sheet REVIEW 3

Project Ref. No.:		Project Title:	Activity Plan Prepared By:	Date of Preparation of Activity Plan:			
Sr. No.	Task			Actual Start Date	Actual Days	Team Mate Names	Status
1	Database Design Diagram	TARSDELIVERY	Huy	21/01/2021	1	All Member	Completed
2	Database Structure			22/01/2021	1	All Member	Completed
3	Template			23/01/2021	1	All Member	Completed
4	Source Structure			23/01/2021	1	All Member	Completed
5	Google Authen API			04/03/2021	1	Huy	Completed
6	Sign In/Sign Out - User			28/01/2021	2	Huy	Completed
7	Sign In/Sign Out – Admin			25/01/2021	3	Huy	Completed
8	Account Manager			30/01/2021	3	Huy	Completed
9	Dashboard			02/03/2021	1	Huy	Completed
10	Google Maps API			23/01/2021	2	Hoai	Completed
11	Paypal API			25/01/2021	3	Hoai	Completed

	Prepared by (Student) Project Group No:3	Approved by (Faculty)
Signature		
Date		

Design Plan:TARS Delivery	Document Name:Review 2	SWD/Form No.1
Effective Date:22/01/2021	Version:1.0	Page No:81

Project Ref. No.:		Project Title:	Activity Plan Prepared By:	Date of Preparation of Activity Plan:			
Sr. No.	Task			Actual Start Date	Actual Days	Team Mate Names	Status
12	Service Page	TARSDELIVERY	Huy	28/01/2021	1	Hoai	Completed
13	Bill Management			29/01/2021	2	Hoai	Completed
14	QR Code			04/03/2021	1	Hoai	Completed
15	Report			30/01/2021	1	Hoai	Completed
16	Search Bill			25/01/2021	2	Kien	Completed
17	Bill History			27/01/2021	2	Kien	Completed
18	Branch Management			29/01/2021	3	Kien	Completed
19	Task Review			21/01/2021	14	Kien, Tu	Completed
20	Contact Us			25/01/2021	1	Tu	Completed
21	Comment Management			27/01/2021	2	Tu	Completed
22	Pricelist Management			30/01/2021	3	Tu	Completed

	Prepared by (Student) Project Group No:3	Approved by (Faculty)
Signature		
Date		

Design Plan:TARS Delivery	Document Name:Review 2	SWD/Form No.1
Effective Date:22/01/2021	Version:1.0	Page No:82

Signature of Instructor	Signature Team Leader
<b>Mr. Tran Phuoc Sinh</b>	<b>Pham Quang Huy</b>

	Prepared by (Student)	Approved by (Faculty)
	<b>Project Group No:3</b>	
<b>Signature</b>		
<b>Date</b>		