EAST WEST UNIVERSITY

Course Code : CSE411

Section : 02

Project Report

For

Bank Management System

Group Members	
Name	**Id**
Tanjina Islam Esha	2018-1-60-010
Adnan Saif Dipto	2018-1-60-157
Md Mahmudur Rahman Limon	2018-1-60-253

Index

Problem Statement

Nowadays people have been using Banks more than ever. In cities, the number of bank users is huge and because of the facilities that most of the bank provides, the number will rise. Due to the technological advancement no one wants to carry cash with them as it is not safe, but also because of the interest they get from the bank.

But there is a problem with these banks. Like old times, most of the bank still uses hard copy for every activity. Now if we think about this, we can easily understand the problems that can cause by these hard copies of papers. For instance, these papers can get lost; these papers can be easily torn or destroyed or get damaged. Because of these hard copies, someone needs to carry them out to different sectors of the bank for several activities which consumes time. If any of the documents erases, these paper will be returned for rewrite which will create even more problems.

There are more sophisticated problems like; a manager needs to manage every employee using papers. So for every data checking, manager will have to go through the files paper by paper, which is not only time consuming but also monotonous and slipping of important data is common. That is why there are lot of complains about banks because of the time they take to do simple tasks.

Proposed Solution

The solution to this problem is simple. We need to create an application to deal with these problems. As the data needs to be accessed regularly and also need to store those data, we can create internet based software.

If we create an application, it will firstly deal with the paper problem. This will be helpful in the economical way. Secondly, it will be a lot faster than papers because we can get access of every data in seconds. Every data will be clicks away. So it will save us time. Which means it will save time while being efficient. We can store the data in database. This will ensure our data storage problem. Soft copies are easy to store and easy to manipulate while ensuring safety. Also we can send those data via internet which will be faster than carrying them out manually.

Also it will reduce the time of finding any data. We can check only the data we need in a matter of seconds. No need go through papers out of papers, files of files. This will increase efficiency of the bank.

Feasibility Studies

Here, we will be discussing about the feasibility of our software. We will discuss this in three sections. They are-

i. Technological Feasibility
ii. Operational Feasibility
iii. Economic Feasibility

- Technological Feasibility: The software that we have developed is a simple one yet efficient to the point where user will be able to use it with ease. For a simple application like this, a device with mare capabilities is enough to operate this application. But it is necessary to understand that this is a web based application, so a strong internet with stable bandwidth is recommended to operate this software. Also, updated internet browser is recommended for an efficient, effective and user friendly experience.

- Operational Feasibility: We have used PHP for coding and MYSQL database to store all the data we have. We have also used XAMPP for MYSQL server. For designing the website we have used HTML and for making it look good, we have added a little bits of CSS to it and for that we have used java script from bootstrap. PHP is a very good coding language with reliable and helpful resources that helped us while programming. For interactive software like ours, PHP is very much reliable and capable. Also as our software is still in its demo phase, MYSQL database is easy to use and manipulate and reliable for this early phase of software.

- Economic Feasibility: If a bank uses software for taking, keeping and manipulating all of the records instead of using papers, it will firstly reduce the cost of paper that the company needed. Secondly, for maintaining and carrying the records from one place to another, a good number of staff is needed. But the number of staff will be reduced by using a software resulting decrease of cost. We have tried to maintain a structure and left a good number of scopes for future development. So, it will be easy and cost friendly for a developer team to update the application in a very little time. This will save them a lot of time and reduce cost. So, using this application is a win from economic perspective.

Requirement Analysis

After discussing about the criteria and conditions of a bank, we have categorized the software requirements in two sections:

i. Functional Requirements
ii. Non-Functional Requirements

- Functional Requirements: Here in this section we will be discussing about the segments that our application will perform on a regular basis. We will discuss the functions that our software is capable to perform-
  - To enter into the system, user must need to register. Then the data will be stored into the database. After that when user will enter a valid ID and password, the database will be checked for validity and then user will be able to enter into the system.
  - Users (manager and employees) can see their profile and update the profile according to necessity.
  - Manager will be able to see the whole employee list, create ID for new employee, and remove an existing employee from the system.
  - Users (both, manager and employee) will be able to create new customer ID, check the list of all customer and their details, and remove an existing customer from the system.
  - Users (both, manager and employees) will be able to deposit, withdraw and transfer credit to another customer's account as per the customer demand.

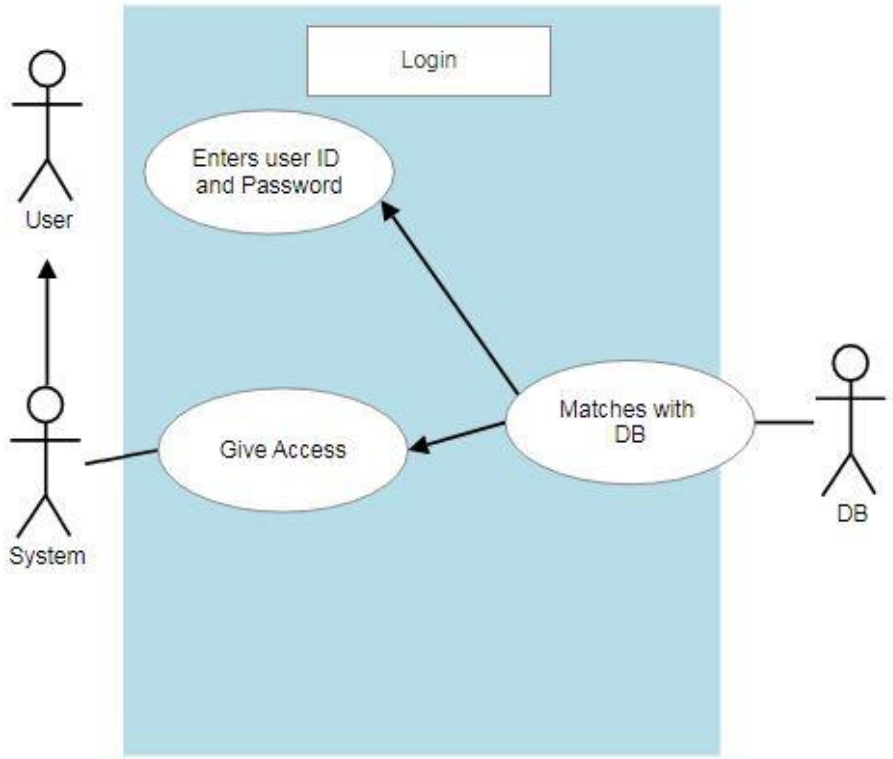
- Non-Functional Requirements: Here we will discuss about the non-functional requirements of the software as it will helpful to ensure the usability and the effectiveness of the entire software system. They are-

  - Manager and employees will have to register first in order to create an ID for the database.
  - With a valid ID and password, manager and employees will be able to enter the system.
  - There are two dash boards in the system. One for the manager and another for employees.
  - Exiting browser will log the user out automatically.
  - A manager will be able to see all the data of employees.
  - Employees will not be able see other employee details or the details of manager.
  - The software should be portable. So moving from one OS to other OS does not create any problem.
  - Admin will be able to manipulate the database manually.

Use Case

Use cases provide a high level view of the system. Use cases and their description are given below:

Use case number#1

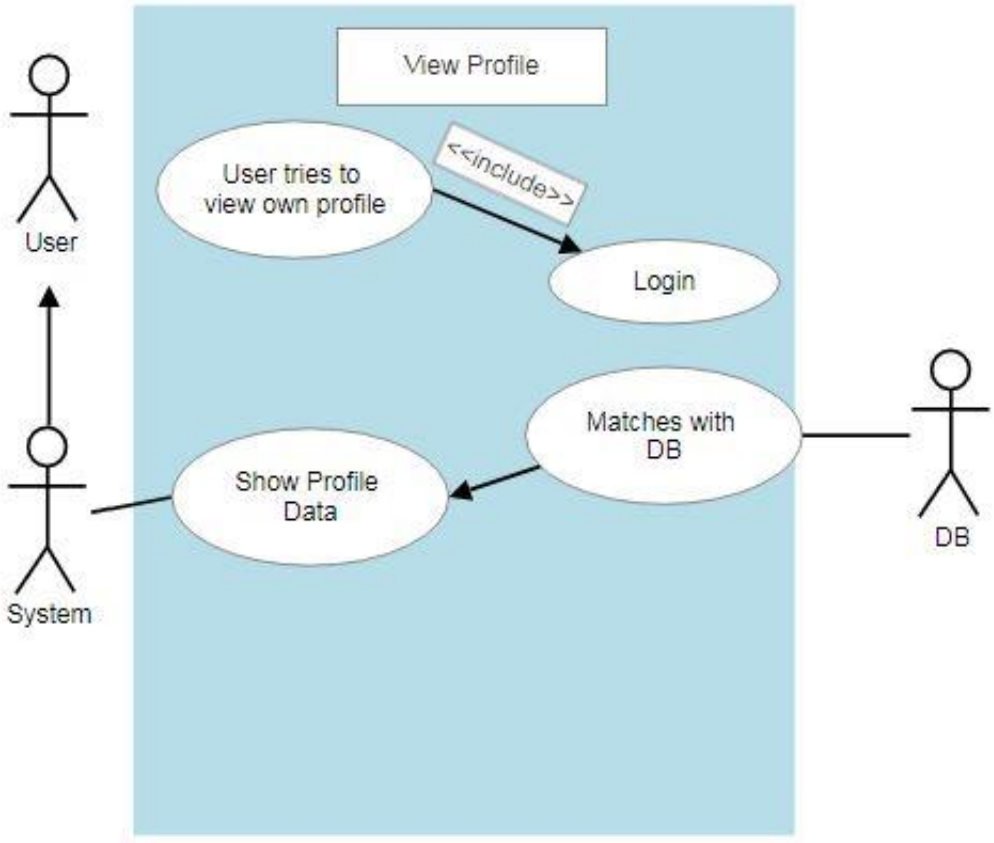


Use case description

Use Case [number #1]	Login to the system
Goal in Context	This use case allows the user to get access to the system so that they can use the features.
Preconditions	A stable bandwidth for better user experience, valid used id and password.
Success End Condition	Getting into the system
Failed End Condition	User will not get access to enter the system. And will be advised to enter correct id and password.
Primary Actor	User(both, manage or employees)
Secondary Actors	Software and Database
Trigger	Clicking on the login button

Description	Step	Action
	1	Users will enter to the login page and enter valid user id and password and press login
	2	Users will be able to insert into the system and use the features
Extensions or Variations	**Step**	**Branching Action**
	1	Users will be asked to enter valid user id and password.

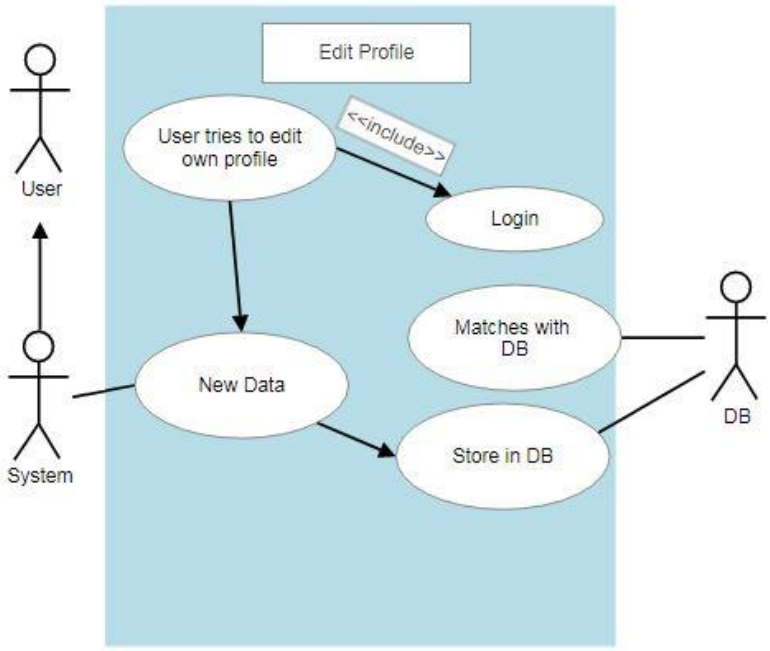
Use case number#2



Use case description

Use Case [number #2]	View Profile	
Goal in Context	This use case allows the user to view their profile. They will be able to see the detailed information that has been stored in the Database.	
Preconditions	A stable bandwidth for better user experience. Log in with a valid user id and password will be needed to get access of this feature.	
Success End Condition	User will be able to view their profile	
Failed End Condition	Users will be asked to login with a verified user id and password. There might be problem with the browser or the internet connection.	
Primary Actor	User	
Secondary Actors	Software and Database	
Trigger	When user will be logged in with a verified user id and password and will press on viewing profile.	
Description	**Step**	**Action**
	1	user will have to log in with a verified user id and password
	2	User will press “Profile”.
Extensions or Variations	**Step**	**Branching Action**
	1	System will collapse.
	2	System will ask to enter valid user id and password to log in

Use case number#3

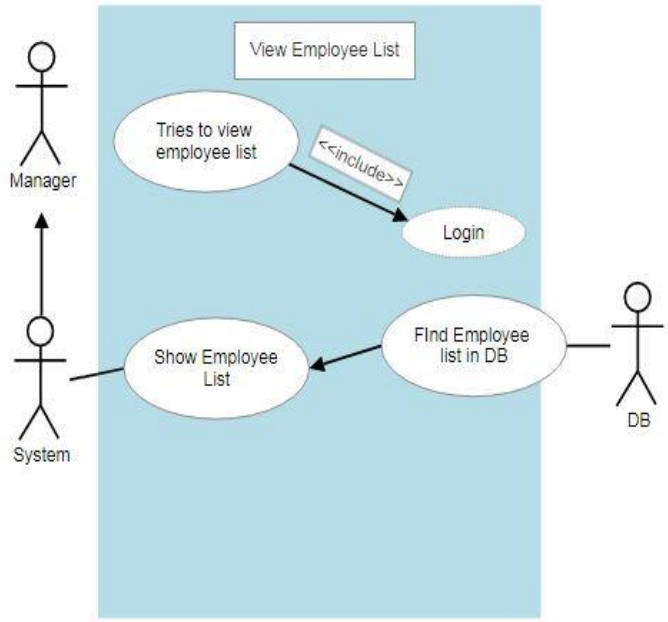


Use case description

Use Case [number #3]	Edit Profile	
Goal in Context	This use case allows the user to edit their profile. They will be able to edit their profile information.	
Preconditions	A stable bandwidth for better user experience. Log in with a valid user id and password will be needed to get access of this feature.	
Success End Condition	User will be able to edit their profile	
Failed End Condition	Users will be asked to login with a verified user id and password. There might be problem with the browser or the internet connection.	
Primary Actor	User	
Secondary Actors	Software and Database	
Trigger	When user will be logged in with a verified user id and password and will press on edit profile.	
Description	**Step**	**Action**
	1	user will have to log in with a verified user id and password
	2	User will press “Edit Profile”.
Extensions or Variations	**Step**	**Branching Action**
	1	System will collapse.

	2	System will ask to enter valid user id and password to log in

Use case number#4

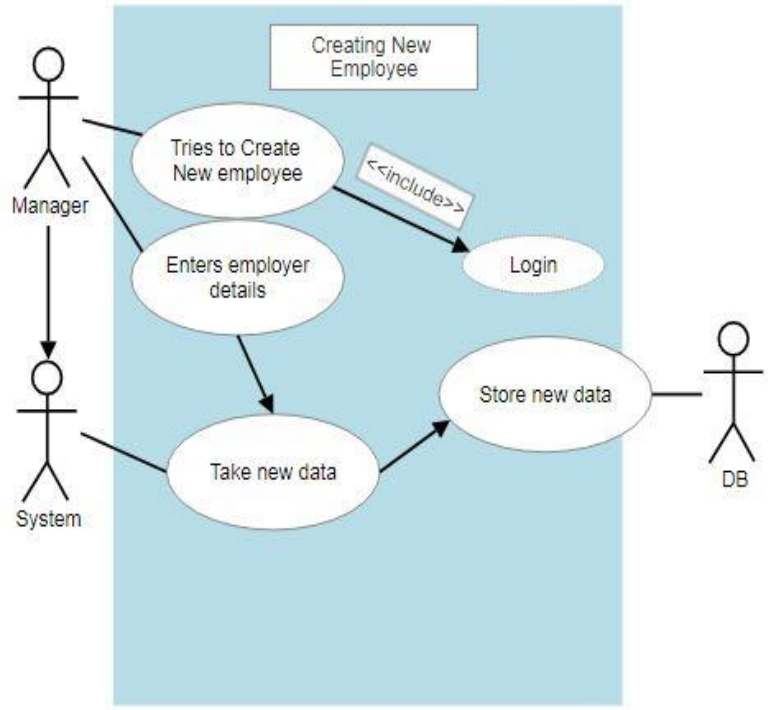


Use case description

Use Case [number #4]	View Employee list	
Goal in Context	This use case allows the Manager to View the employee list.	
Preconditions	A stable bandwidth for better user experience. Log in with a valid user id and password will be needed to get access of this feature.	
Success End Condition	Manager will be able to view the employee list.	
Failed End Condition	Manager will be asked to login with a verified user id and password. There might be problem with the browser or the internet connection.	
Primary Actor	Manager	
Secondary Actors	Software system and Database	
Trigger	When the manager will be logged in with a verified user id and password and will press on “View Employee List”.	
Description	**Step**	**Action**
	1	Manager will have to log in with a verified user id and password.

	2	Manager will press “View Employee list”.
Extensions or Variations	**Step**	**Branching Action**
	1	System will collapse.
	2	System will ask to enter valid user id and password to log in

Use case number#5

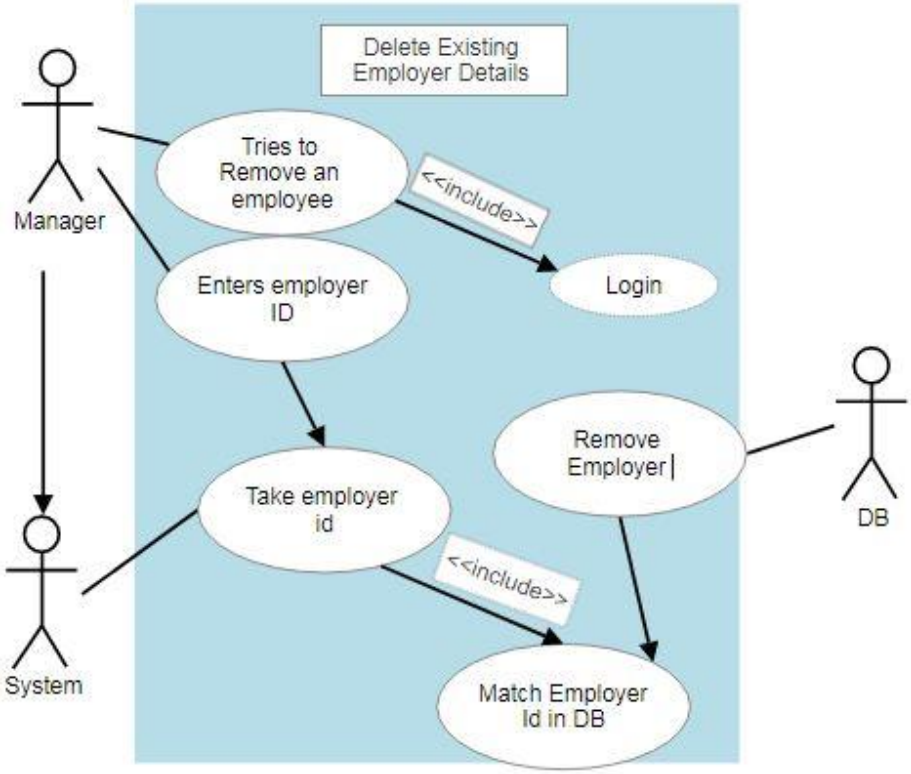


Use case description

Use Case [number #5]	Creating New Employee
Goal in Context	This use case allows the Manager to create new employee id.
Preconditions	A stable bandwidth for better user experience. Log in with a valid user id and password will be needed to get access of this feature.
Success End Condition	Manager will be able to create a new employee id.
Failed End Condition	Manager will be asked to login with a verified user id and password. There might be problem with the browser or the internet connection.

Primary Actor	Manager	
Secondary Actors	Software system and Database	
Trigger	When the manager will be logged in with a verified user id and password and will press on “Create New Employee”.	
Description	**Step**	**Action**
	1	Manager will have to log in with a verified user id and password.
	2	Manager will press “Create New Employee”.
	3	Manager will enter new data of a new employee
Extensions or Variations	**Step**	**Branching Action**
	1	System will collapse.
	2	System will ask to enter valid user id and password to log in.
	3	Data will not be stored due to same Employer ID

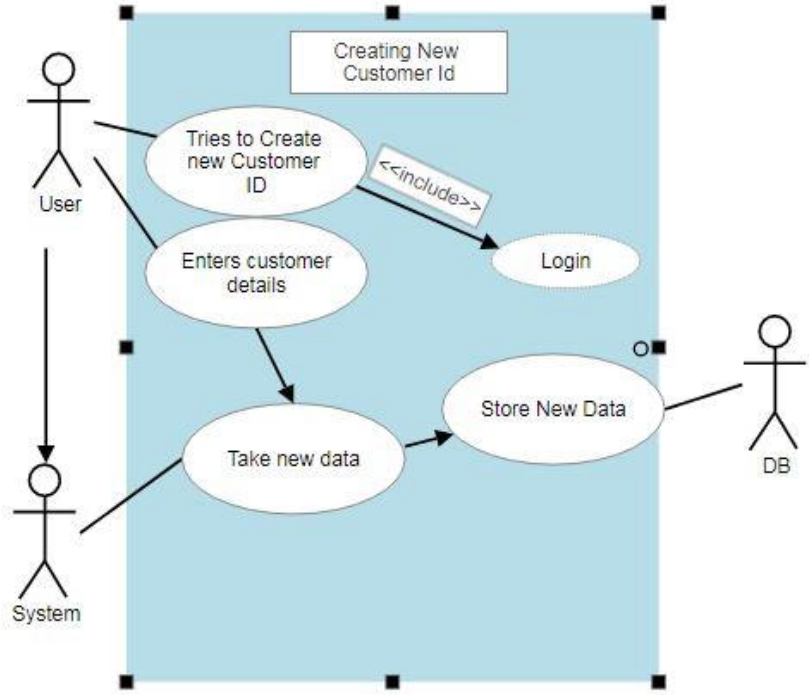
Use case number#6



Use case description

Use Case [number #6]	Delete Existing Employer Details	
Goal in Context	This use case allows the Manager to remove an employer's profile and detailed information.	
Preconditions	A stable bandwidth for better user experience. Log in with a valid user id and password will be needed to get access of this feature.	
Success End Condition	Manager will be able to remove an employer and all his information.	
Failed End Condition	Manager will be asked to login with a verified user id and password. There might be problem with the browser or the internet connection.	
Primary Actor	Manager	
Secondary Actors	Software system and Database	
Trigger	When the manager will be logged in with a verified user id and password and will press on "Remove Employee".	
Description	**Step**	**Action**
	1	Manager will have to log in with a verified user id and password.
	2	Manager will press "Remove Employee".
	3	Manager will enter a valid employee Id
	4	DB will search for that particular Id and when it will find a match, it will remove all the information of that Id
Extensions or Variations	**Step**	**Branching Action**
	1	System will collapse.
	2	System will ask to enter valid user id and password to log in
	3	No such id in the DB is found

Use case number#7

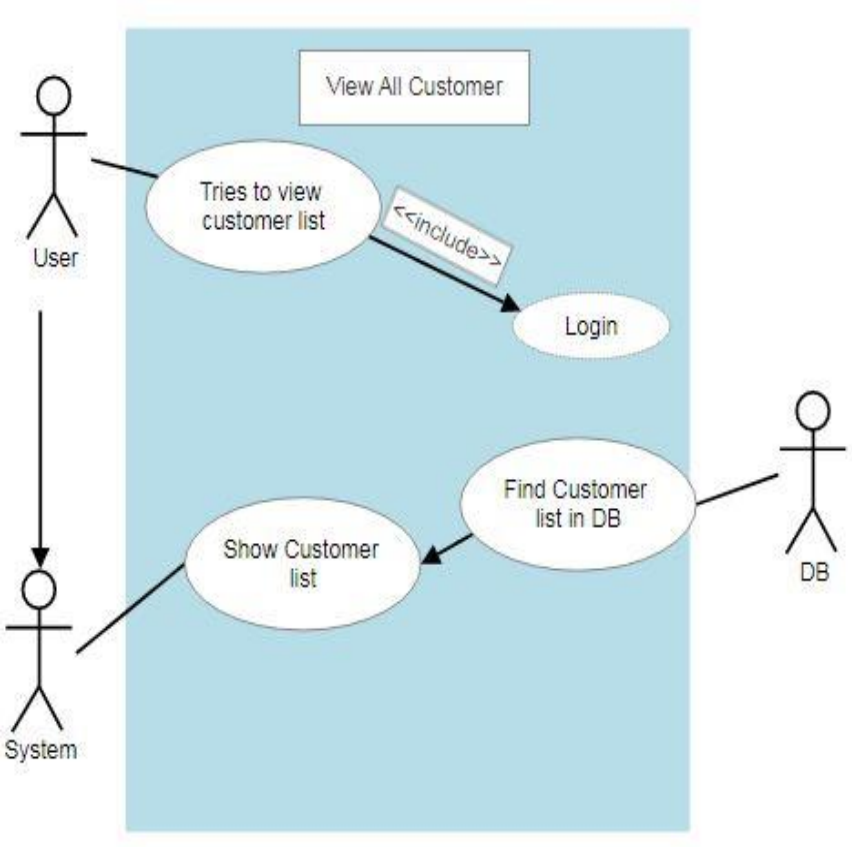


Use case description

Use Case [number #7]	Creating New Customer ID	
Goal in Context	This use case allows the users to create new customer id.	
Preconditions	A stable bandwidth for better user experience. Log in with a valid user id and password will be needed to get access of this feature.	
Success End Condition	Users will be able to create a new customer id.	
Failed End Condition	Users will be asked to login with a verified user id and password. There might be problem with the browser or the internet connection.	
Primary Actor	User	
Secondary Actors	Software system and Database	
Trigger	When the manager will be logged in with a verified user id and password and will press on “Create New Customer”.	
Description	**Step**	**Action**
	1	User will have to log in with a verified user id and password.
	2	User will press “Create New

		Employee”.
	3	User will enter new data of a new customer.
Extensions or Variations	**Step**	**Branching Action**
	1	System will collapse.
	2	System will ask to enter valid user id and password to log in.
	3	Data will not be stored due to same customer ID

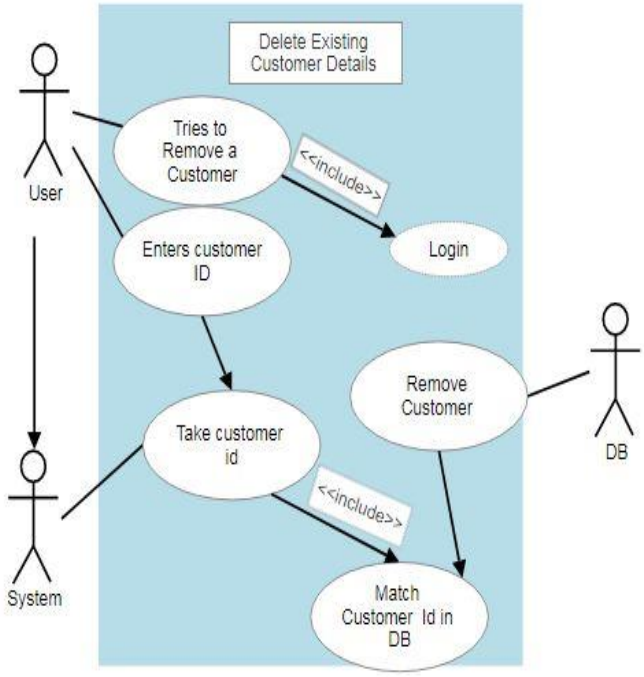
Use case number#8



Use case description

Use Case [number #8]	View All Customer	
Goal in Context	This use case allows the user to View all the customers	
Preconditions	A stable bandwidth for better user experience. Log in with a valid user id and password will be needed to get access of this feature.	
Success End Condition	Users will be able to view all the customers.	
Failed End Condition	Users will be asked to login with a verified user id and password. There might be problem with the browser or the internet connection.	
Primary Actor	Users	
Secondary Actors	Software system and Database	
Trigger	When the users will be logged in with a verified user id and password and will press on “View All Customer”.	
Description	**Step**	**Action**
	1	Users will have to log in with a verified user id and password.
	2	Users will press “View All Customer”.
Extensions or Variations	**Step**	**Branching Action**
	1	System will collapse.
	2	System will ask to enter valid user id and password to log in

Use case number#9

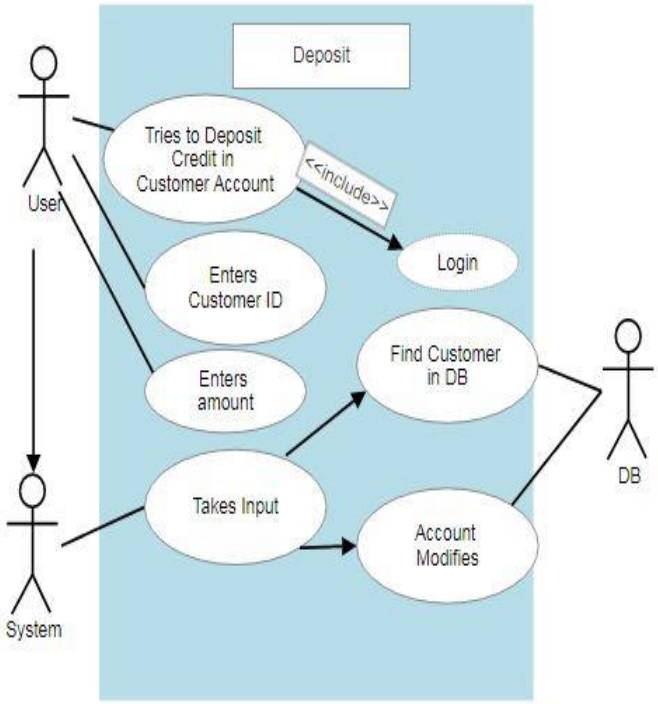


Use case description

Use Case [number #9]	Delete Existing Customer Details	
Goal in Context	This use case allows the users to remove a customer's profile and detailed information.	
Preconditions	A stable bandwidth for better user experience. Log in with a valid user id and password will be needed to get access of this feature.	
Success End Condition	Users will be able to remove a Customer and all his information.	
Failed End Condition	Users will be asked to login with a verified user id and password. There might be problem with the browser or the internet connection.	
Primary Actor	Users	
Secondary Actors	Software system and Database	
Trigger	When users will be logged in with a verified user id and password and will press on "Remove Customer".	
Description	**Step**	**Action**
	1	Users will have to log in with a verified user id and password.
	2	Users will press "Remove

		Customer".
	3	Users will enter a valid customer Id
	4	DB will search for that particular Id and when it will find a match, it will remove all the information of that Id
Extensions or Variations	**Step**	**Branching Action**
	1	System will collapse.
	2	System will ask to enter valid user id and password to log in
	3	No such id in the DB is found

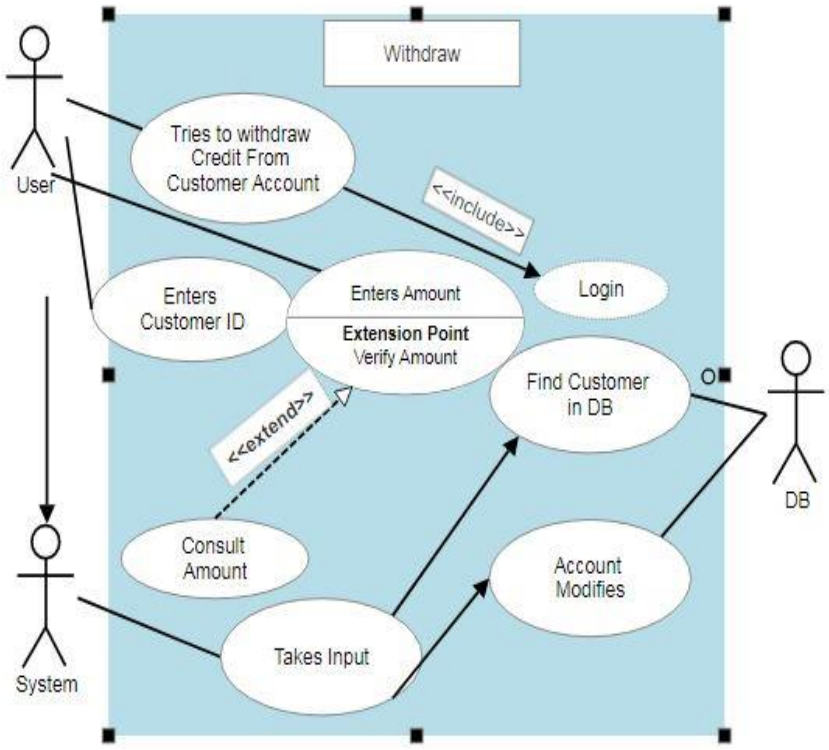
Use case number#10



Use case description

Use Case [number #10]	Deposit	
Goal in Context	This use case allows the users to Deposit credit into a customer's account.	
Preconditions	A stable bandwidth for better user experience. Log in with a valid user id and password will be needed to get access of this feature.	
Success End Condition	Users will be able to deposit credit into customer's account.	
Failed End Condition	Users will be asked to login with a verified user id and password. There might be problem with the browser or the internet connection. Failure in transection.	
Primary Actor	Users	
Secondary Actors	Software system and Database	
Trigger	When the user will be logged in with a verified user id and password and will press on "Deposit".	
Description	**Step**	**Action**
	1	Users will have to log in with a verified user id and password.
	2	Users will press "Deposit".
	3	Users will enter a valid customer Id
	4	DB will search for that particular Id and when it will find a match, it will deposit the desired amount to that particular ID
Extensions or Variations	**Step**	**Branching Action**
	1	System will collapse.
	2	System will ask to enter valid user id and password to log in
	3	No such id in the DB is found

Use case number#11

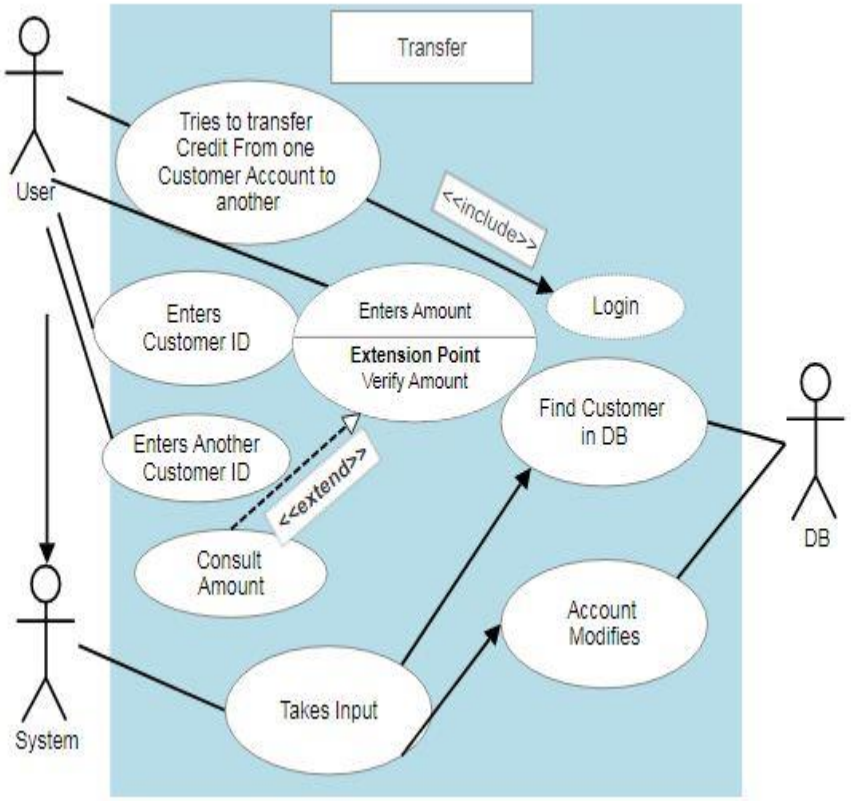


Use case description

Use Case [number #11]	Withdraw	
Goal in Context	This use case allows the users to Withdraw credit from a customer's account.	
Preconditions	A stable bandwidth for better user experience. Log in with a valid user id and password will be needed to get access of this feature. Desired amount should be lesser than the present balance.	
Success End Condition	Users will be able to withdraw credit from customer's account.	
Failed End Condition	Users will be asked to login with a verified user id and password. There might be problem with the browser or the internet connection. Failure in transection.	
Primary Actor	Users	
Secondary Actors	Software system and Database	
Trigger	When the user will be logged in with a verified user id and password and will press on "Withdraw".	
Description	**Step**	**Action**

	1	Users will have to log in with a verified user id and password.
	2	Users will press “Withdraw”.
	3	Users will enter a valid customer Id
	4	DB will search for that particular Id and when it will find a match, it will check the amount present in the account. If present balance> desired balance, credit will be withdrawn as the desired amount particular ID
Extensions or Variations	**Step**	**Branching Action**
	1	System will collapse.
	2	System will ask to enter valid user id and password to log in
	3	No such id in the DB is found
	4	Desired Balance>Present Balance Transection failed.

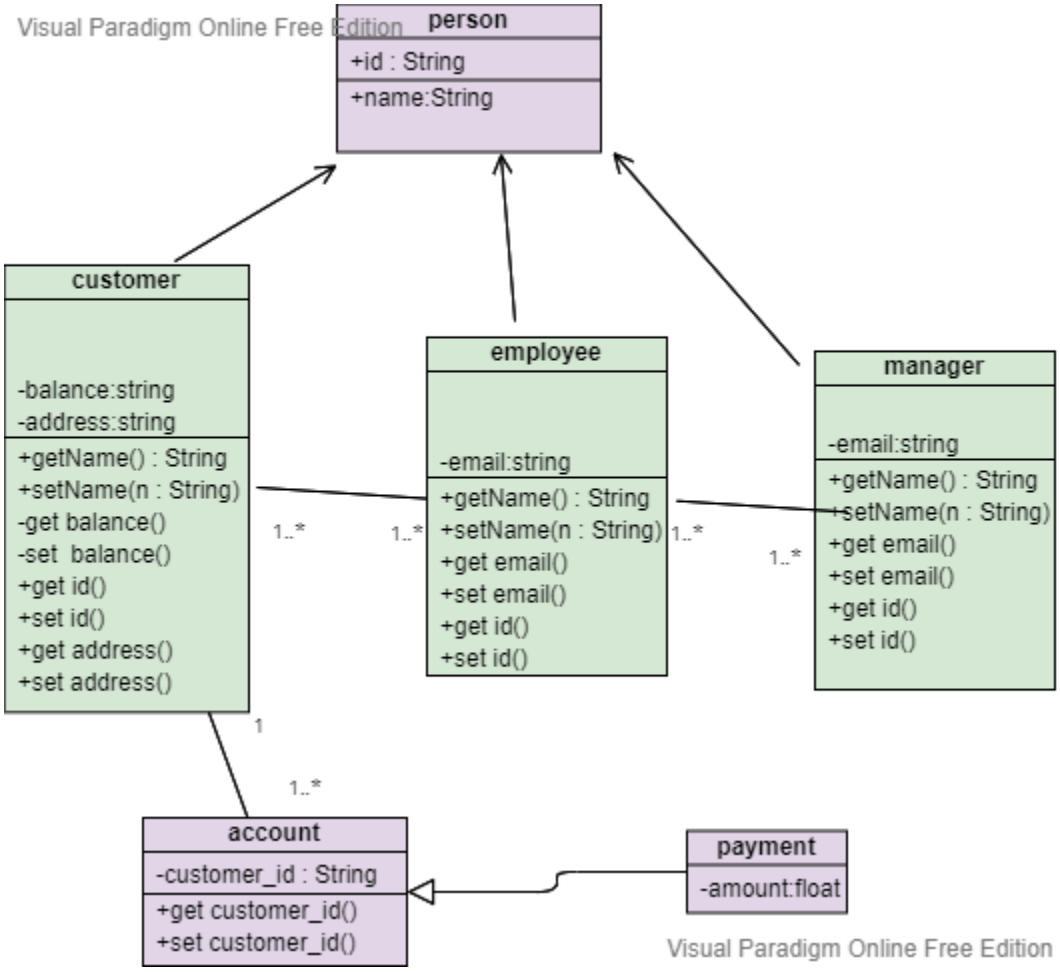
Use case number#12



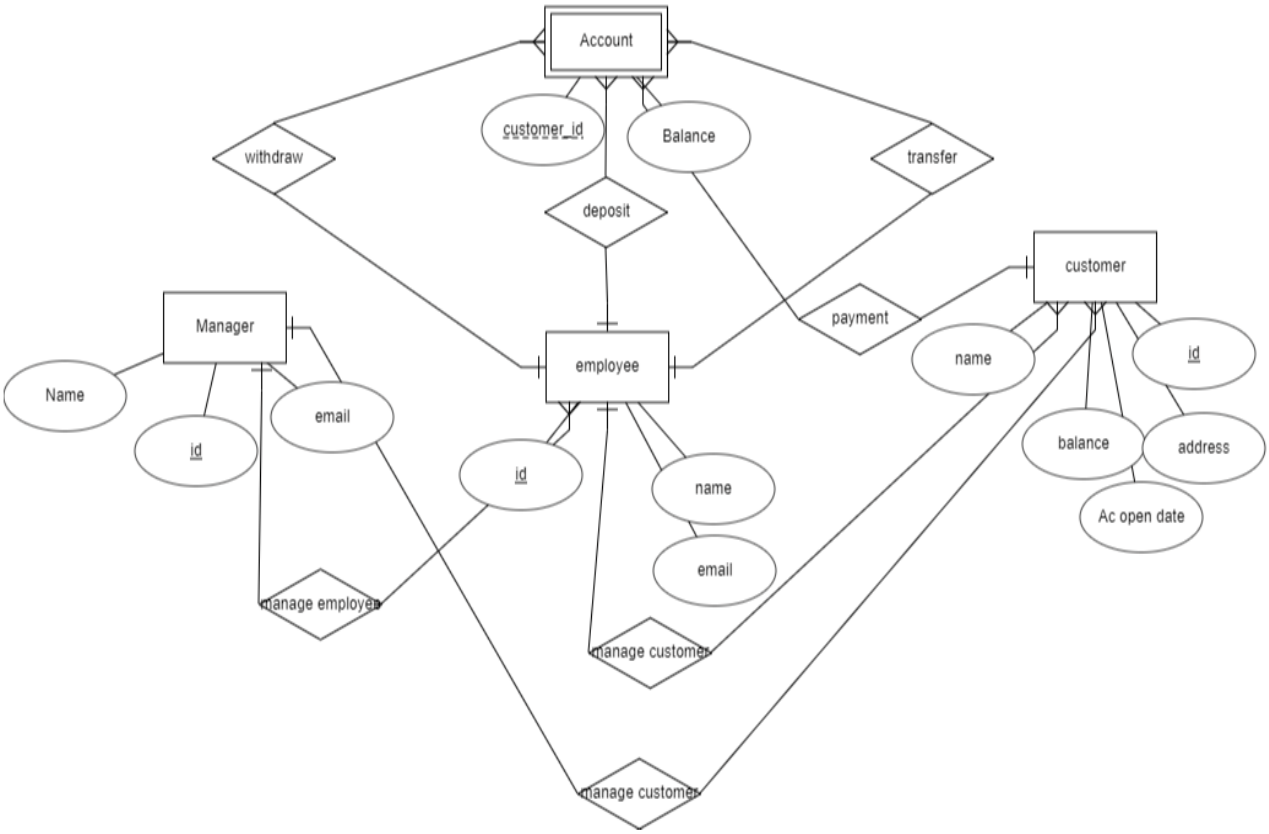
Use case description

Use Case [number #12]	Transfer	
Goal in Context	This use case allows the users to transfer credit from a customer's account to another.	
Preconditions	A stable bandwidth for better user experience. Log in with a valid user id and password will be needed to get access of this feature. Desired transfer amount should be lesser than the present balance. And a valid transfer id is given.	
Success End Condition	Users will be able to transfer credit from customer's account to another.	
Failed End Condition	Users will be asked to login with a verified user id and password. There might be problem with the browser or the internet connection. Failure in transection.	
Primary Actor	Users	
Secondary Actors	Software system and Database	
Trigger	When the user will be logged in with a verified user id and password and will press on "Transfer".	
Description	**Step**	**Action**
	1	Users will have to log in with a verified user id and password.
	2	Users will press "Transfer".
	3	Users will enter a valid customer Id, an amount and a valid customer's id to transfer the credit to.
	4	DB will search for that particular Id and when it will find a match, it will check the amount present in the account. If present balance> desired balance, credit will be transferred to the desired customer's id.
Extensions or Variations	**Step**	**Branching Action**
	1	System will collapse.
	2	System will ask to enter valid user id and password to log in
	3	No such id in the DB is found
	4	Desired Balance>Present Balance Transection failed.

UML Class Diagram



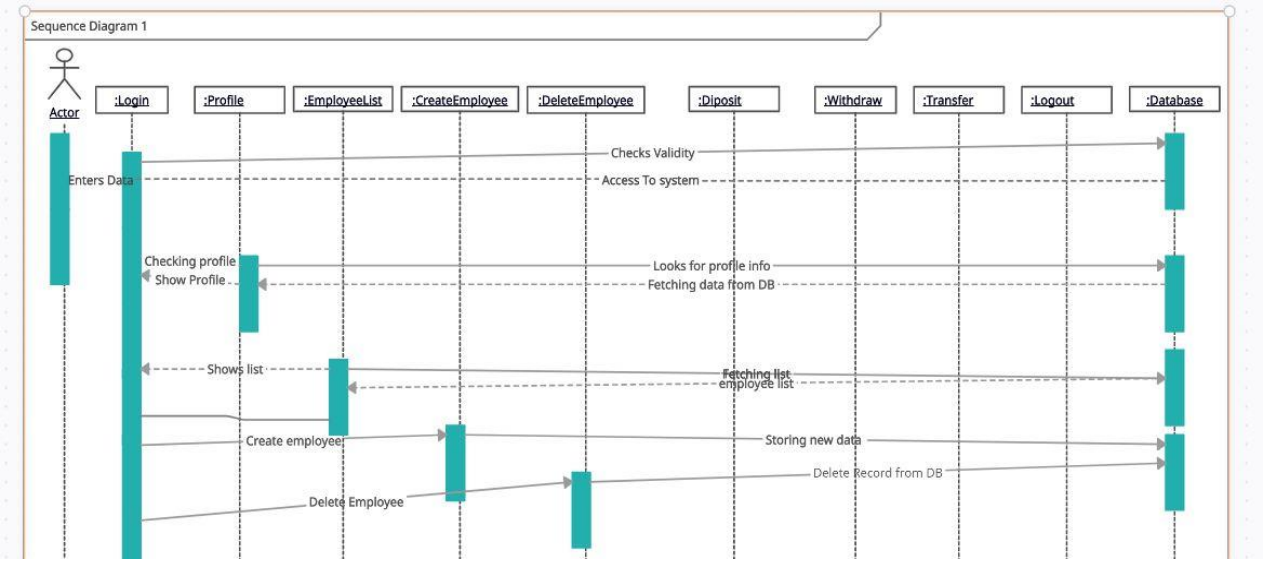
ER Diagram


Account
customer_id
Balance
withdraw
transfer
deposit
customer
Manager
payment
employee
name
id
Name
email
id
balance
address
id
name
Ac open date
email
manage employee
manage customer
manage customer

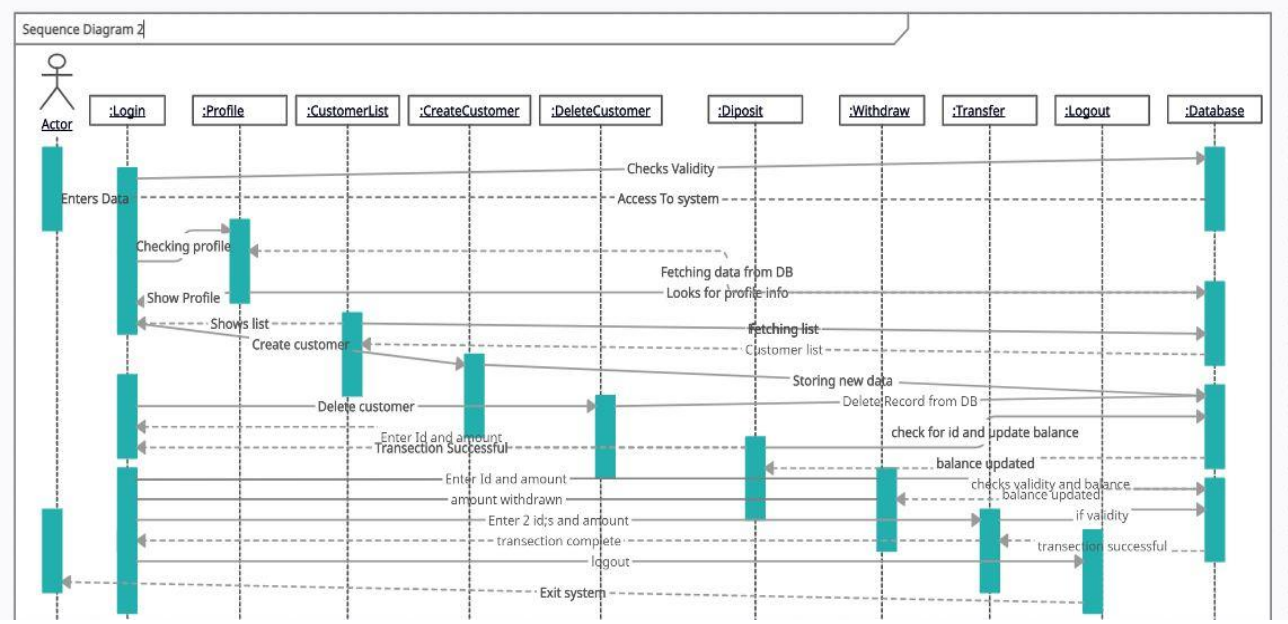
Sequence Diagram

We have divided the sequence diagram in two segments. In our software, we have two users; manager and employee. Here, manager can create, delete and view the employee list, but employees cannot. That is why this special feature has been shown in a particular sequence diagram and whole process of the manager and employee has been shown in another diagram.

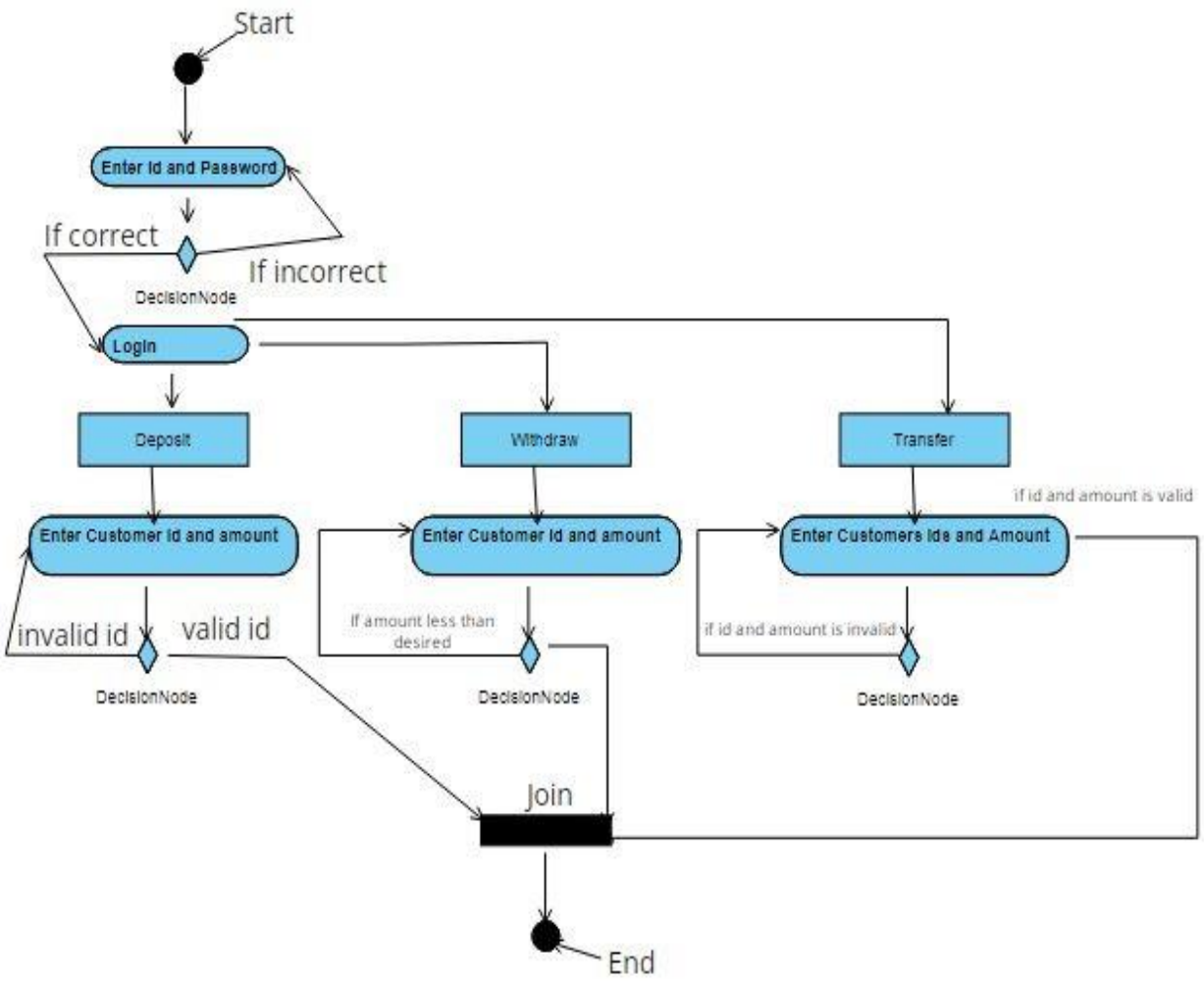
Sequence Diagram #01



Sequence Diagram #02



Activity Diagram



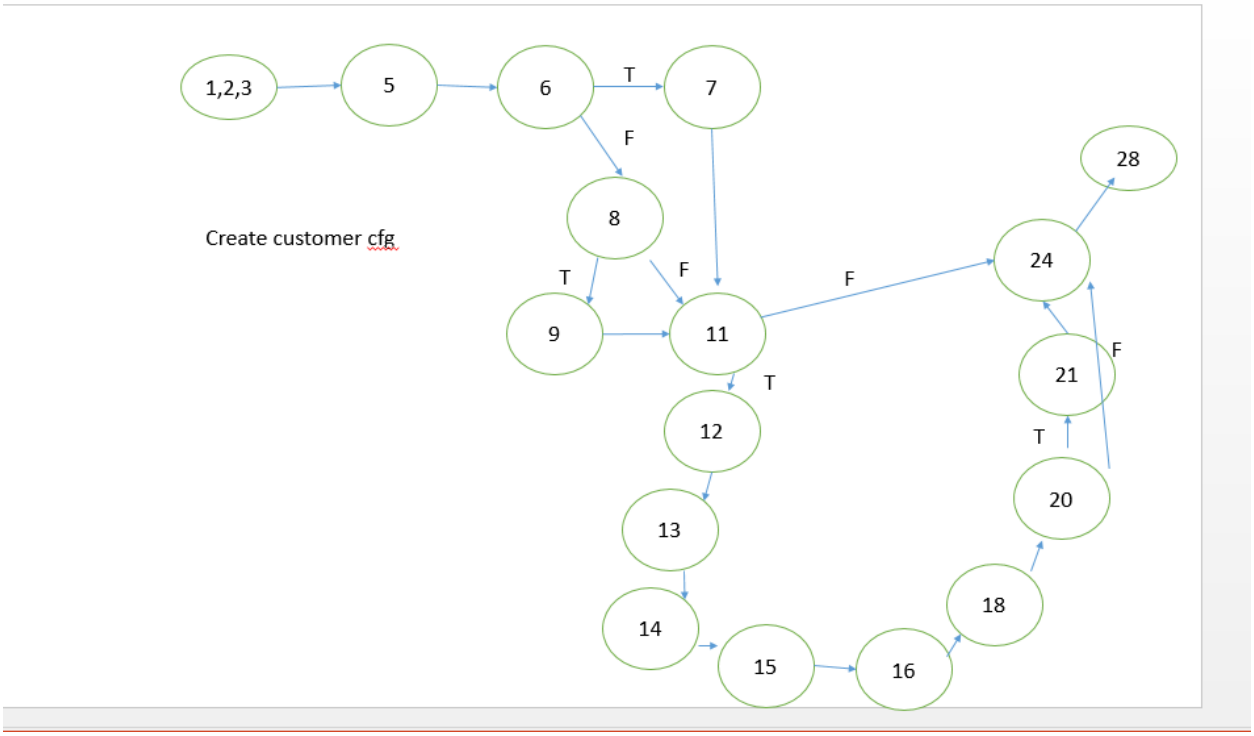
Test Cases (testing): White Box Method

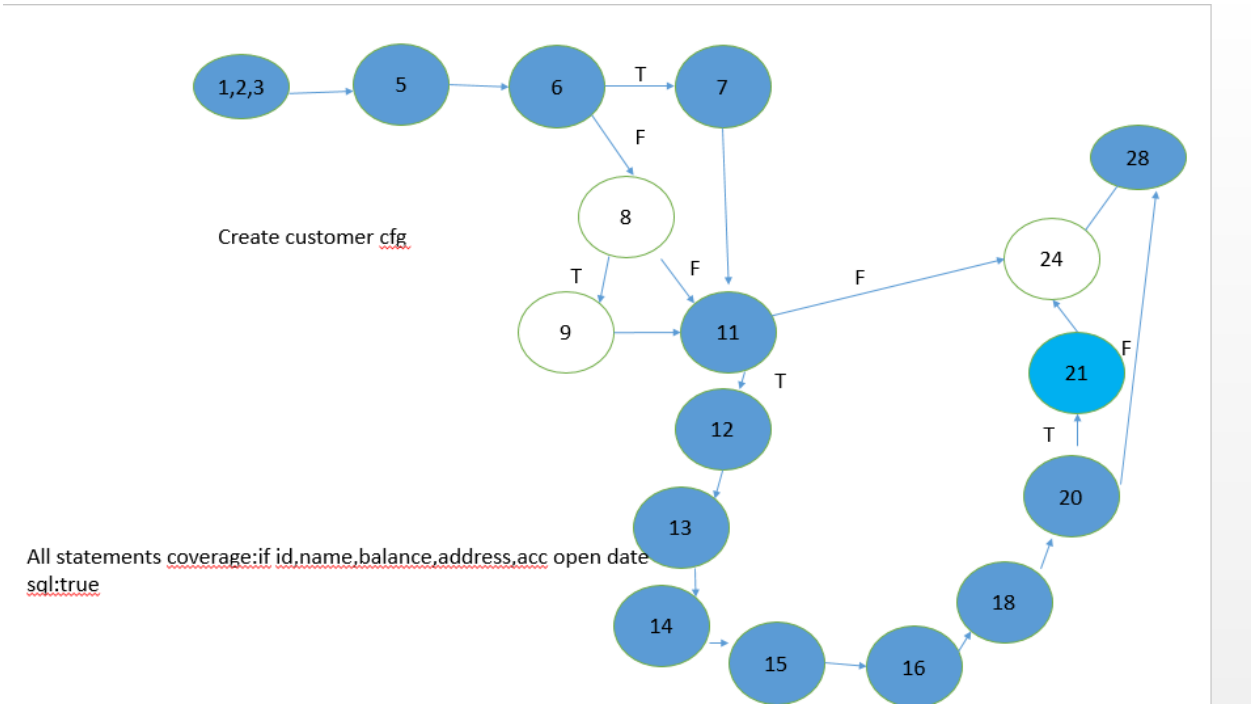
p_form-20211130T134701Z-001 > php_form > create(cus).php

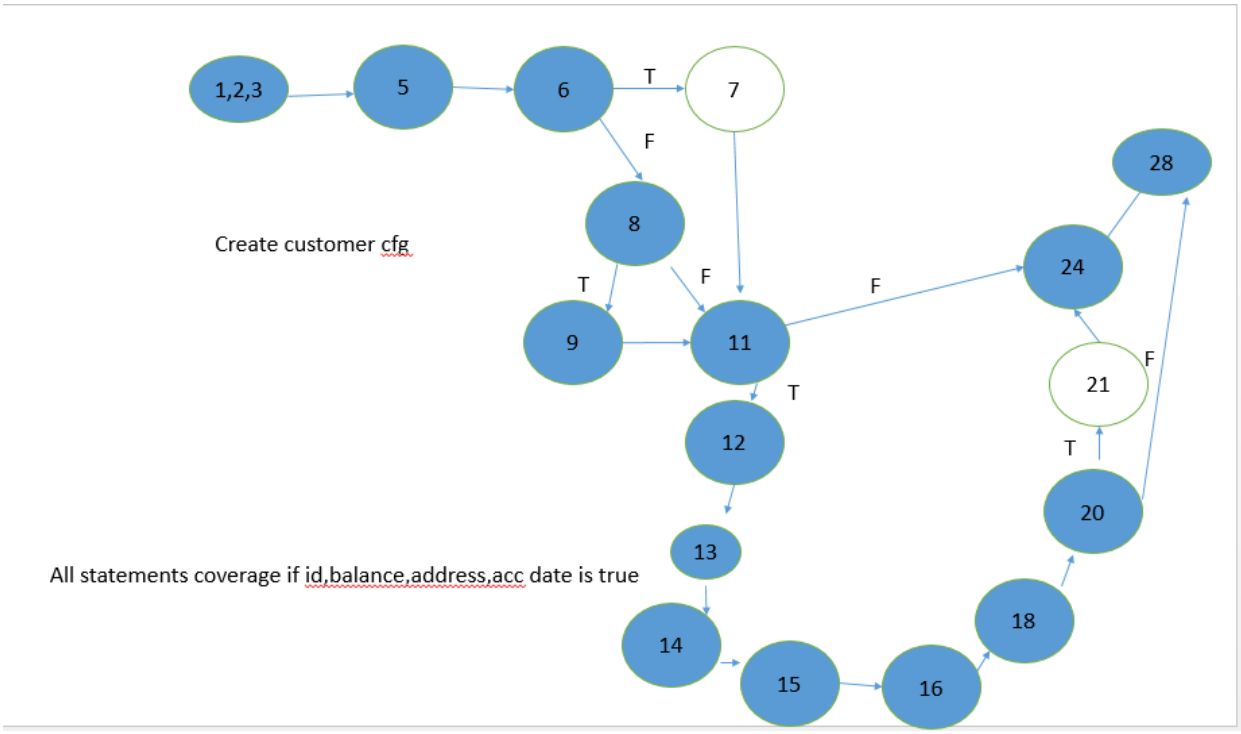
```php
<?php

$conn = new mysqli("localhost", "root", "","php_form");

session_start();
if (!$_SESSION['login']){
  header('Location: login.php');
} elseif($_SESSION['role_id'] == 1){
  header('Location: manager.php');
}
if(isset($_POST['id'])){
  $id = $_POST['id'];
  $name = $_POST['name'];
  $address = $_POST['address'];
  $balance = $_POST['balance'];
  $acc_open_date = $_POST['acc_open_date'];

  $sql = "INSERT INTO customer (id, name, address, balance, acc_open_date) VALUES ('$id', '$name', '$address', '$balance',

  if ($conn->query($sql) === TRUE) {
      echo "NEW CUSTOMER PROFILE CREATED";
  }
  else{
      echo "Error: " .  $sql . "<br>" . $conn ->error;
  }
}

$conn->close();

?>
```

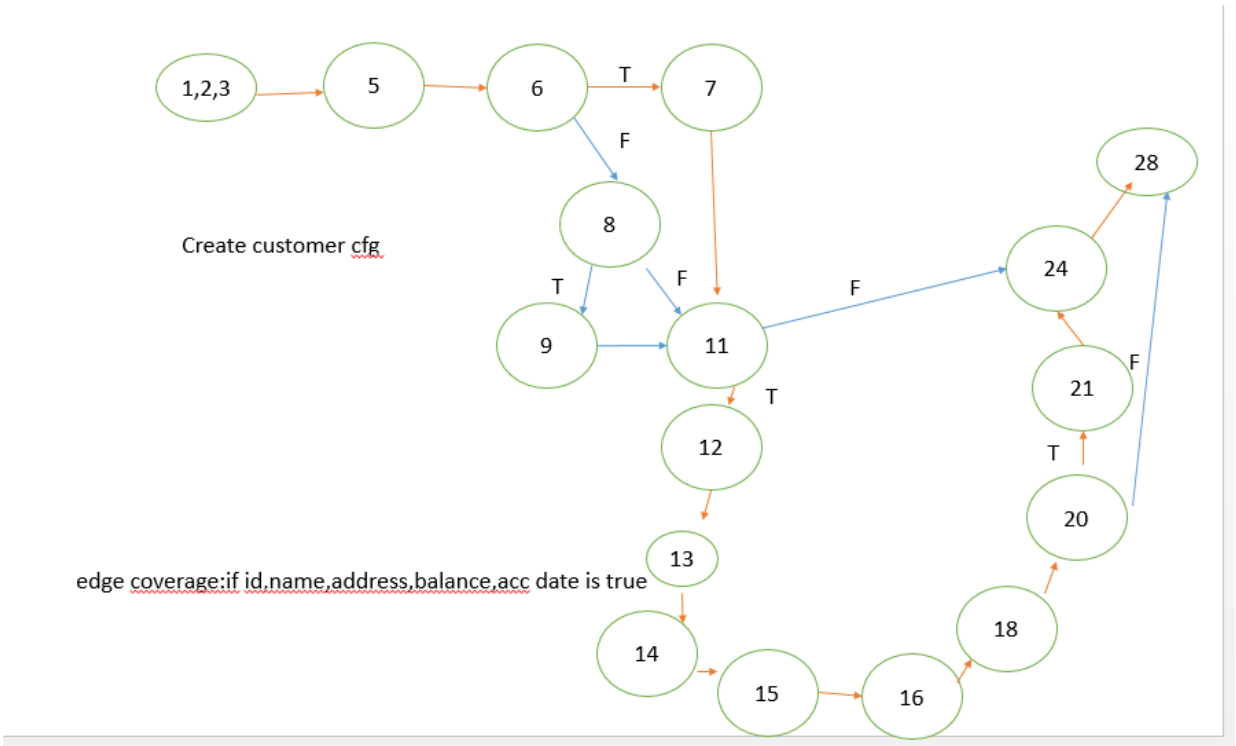
_form-20211130T134701Z-001 > php_form > delete(cus).php > ...

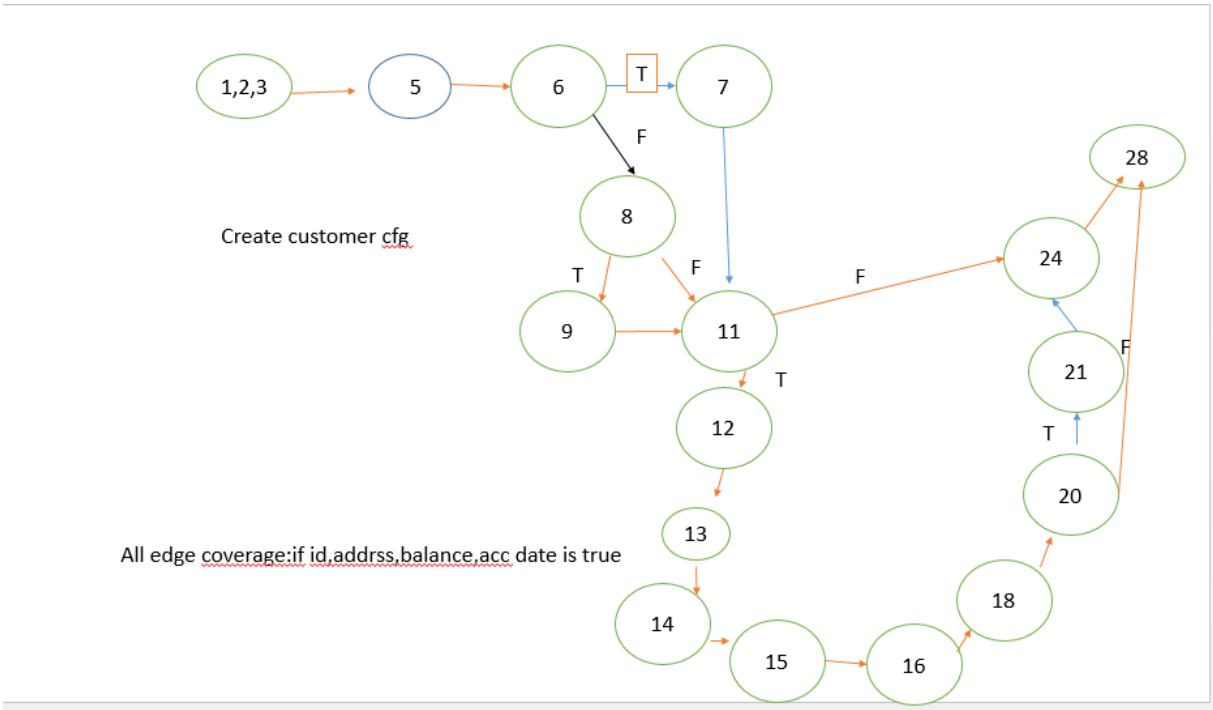
```php
<?php
$conn = new mysqli("localhost", "root", "","php_form");

session_start();
if (!$_SESSION['login']){
  header('Location: login.php');
} elseif($_SESSION['role_id'] == 1){
  header('Location: manager.php');
}
$id = $_POST['id'];
$sql = "DELETE FROM customer WHERE id='$id'";

if ($conn->query($sql) === TRUE) {
    echo "Record deleted successfully";
  } else {
    echo "Error deleting record: " . $conn->error;
  }

  $conn->close();
```

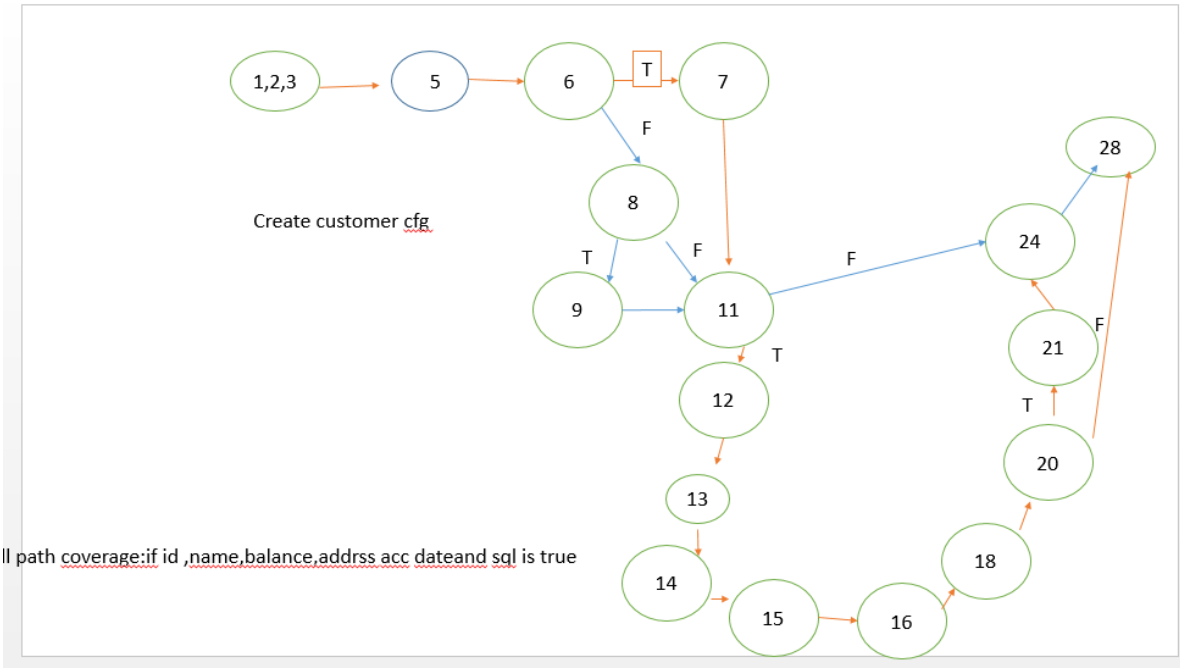

1,2,3
5
6
T
7
F
8
28
Create customer cfg
T
F
F
24
9
11
F
21
T
12
T
20
13
18
14
15
16

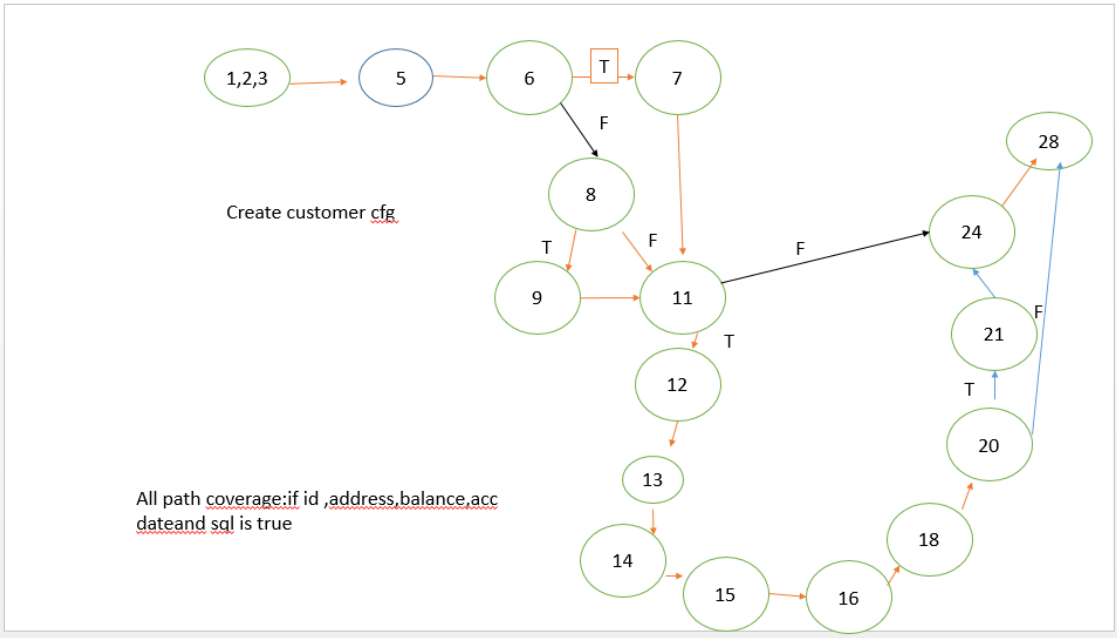

1,2,3
5
6
T
7
F
8
28
Create customer cfg
T
F
F
24
9
11
F
21
T
12
T
20
13
18
14
15
16
All statements coverage:if id,name,balance,address,acc open date
sql:true

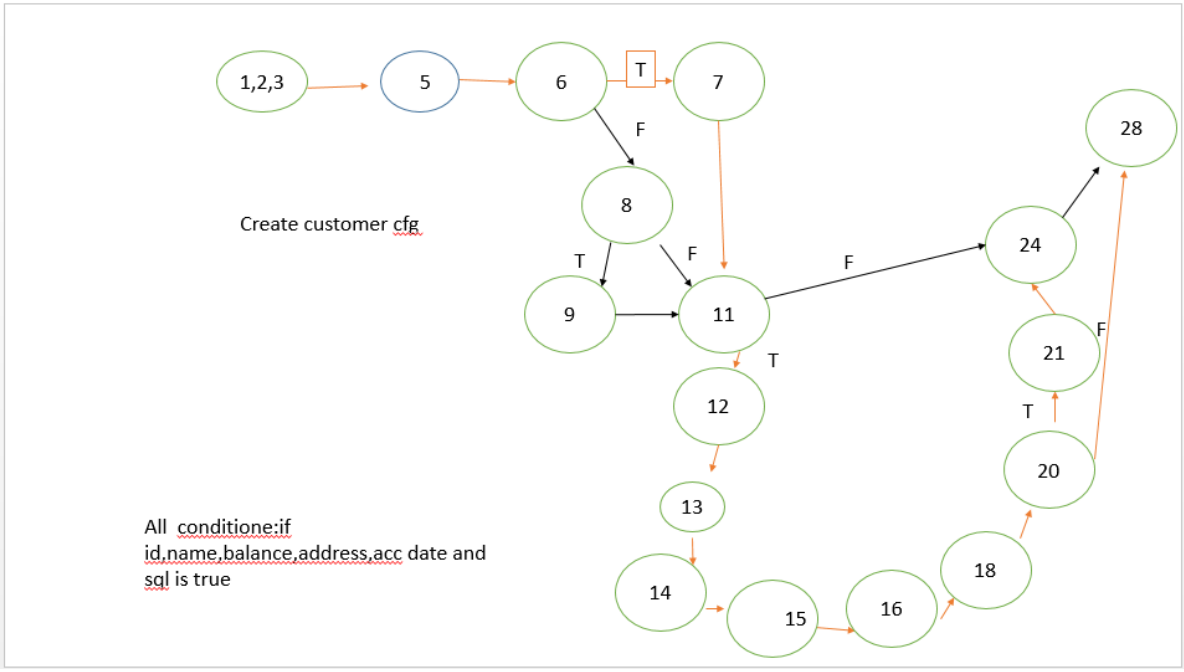

1,2,3
5
6
T
7
F
8
Create customer cfg
T
F
9
11
F
24
28
21
F
T
12
T
20
All statements coverage if id,balance,address,acc date is true
13
14
15
16
18

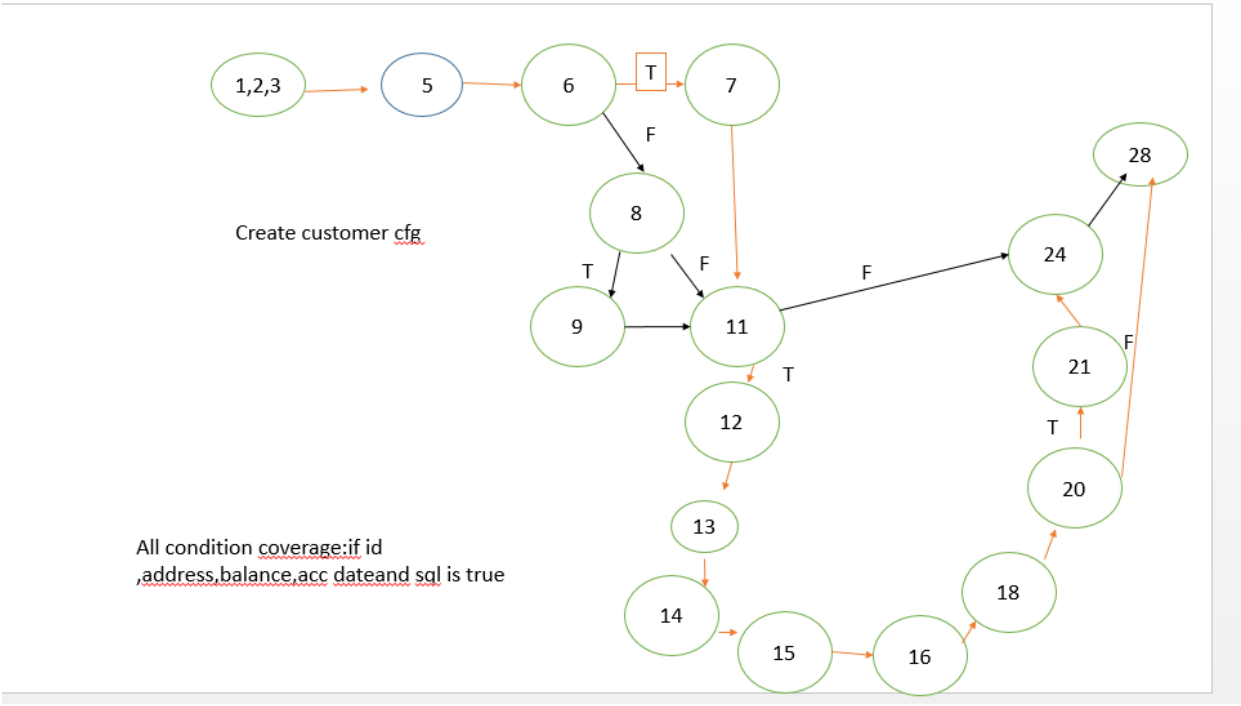

1,2,3
5
6
T
7
F
8
Create customer cfg
T
F
9
11
F
24
28
21
F
T
12
T
20
edge coverage:if id,name,address,balance,acc date is true
13
14
15
16
18

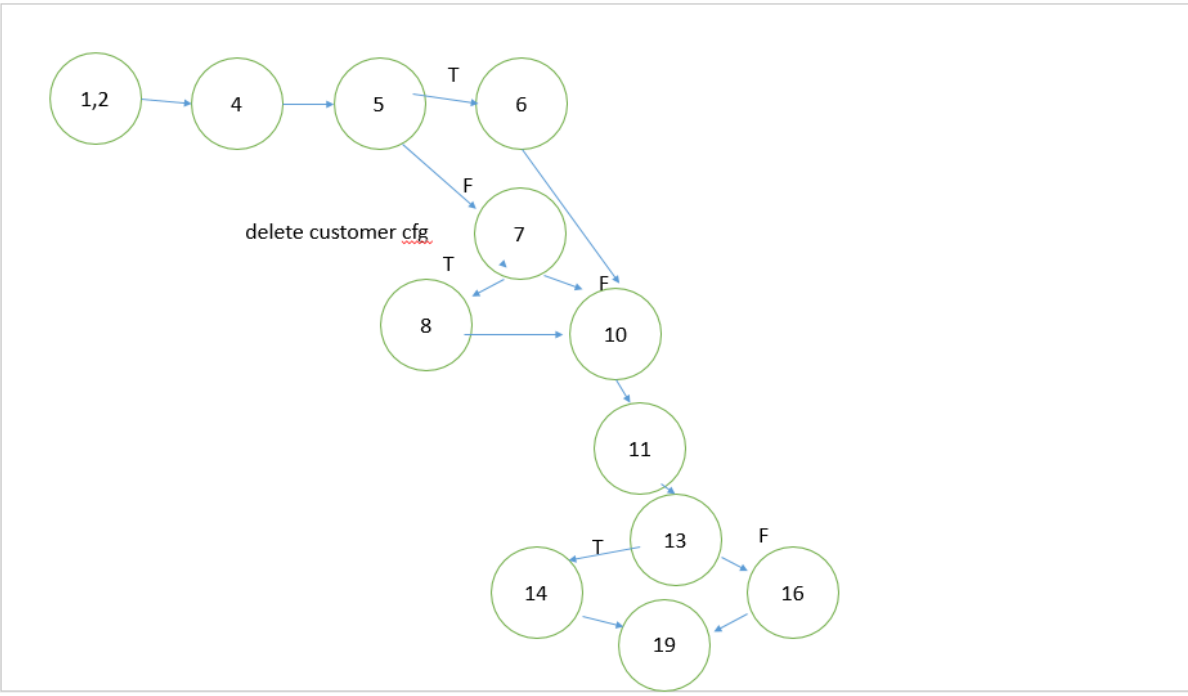

1,2,3
5
6
T
7
F
8
28
Create customer cfg
24
T
F
F
9
11
21
F
T
12
T
20
13
All edge coverage:if id,addrss,balance,acc date is true
14
18
15
16

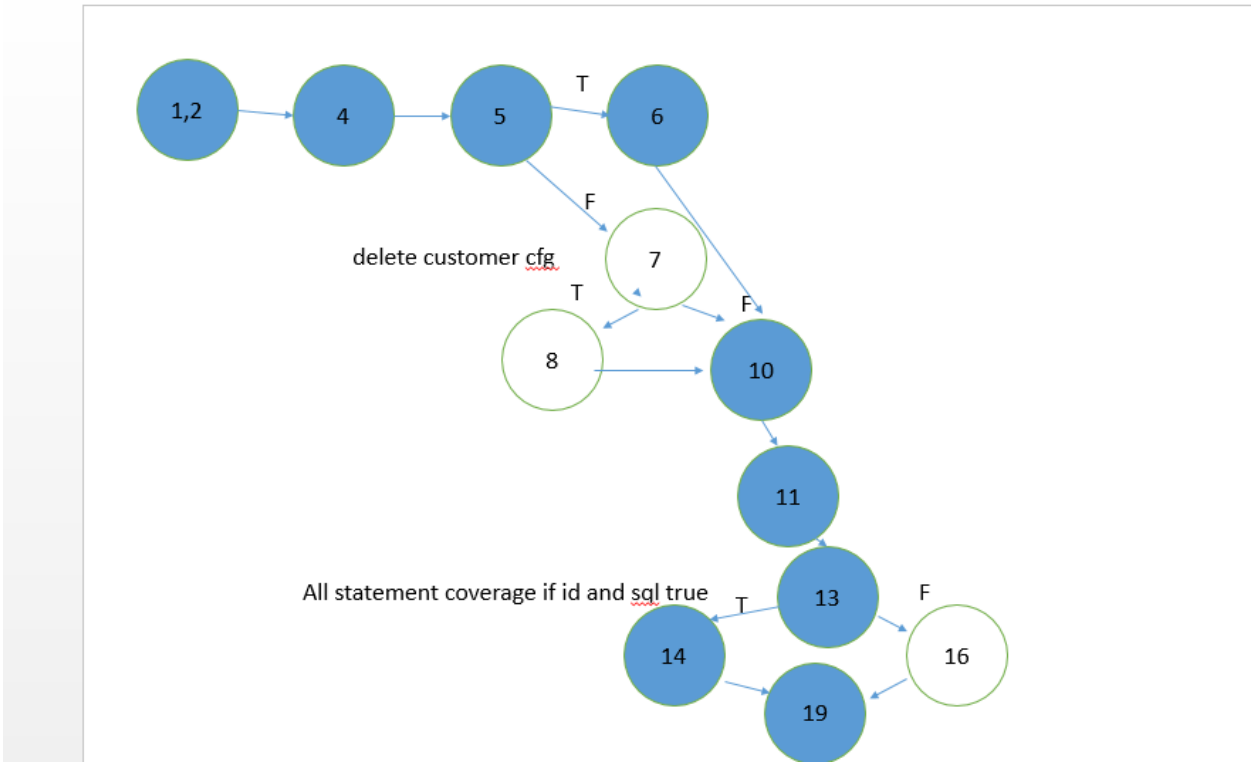

1,2,3
5
6
T
7
F
28
8
Create customer cfg
24
T
F
F
9
11
21
F
T
12
T
20
13
ll path coverage:if id ,name,balance,addrss acc dateand sql is true
14
18
15
16

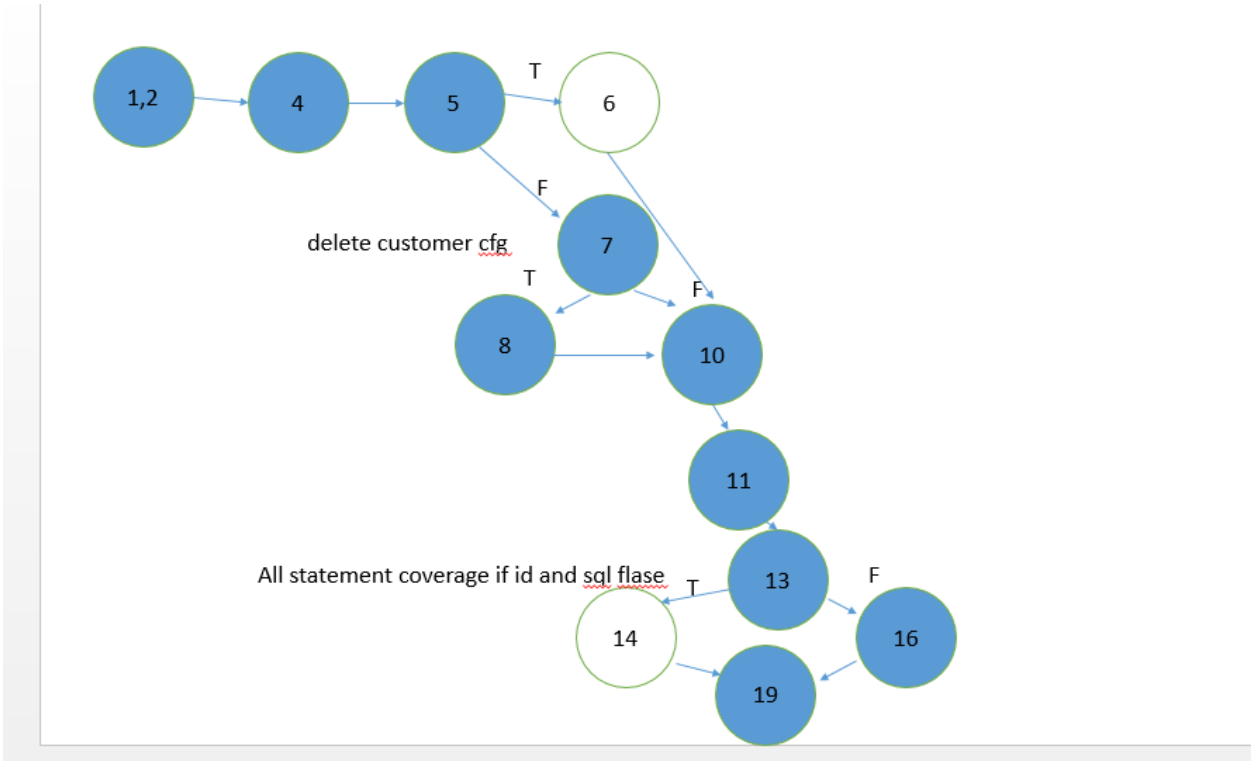

1,2,3
5
6
T
7
F
8
28
Create customer cfg
T
F
F
24
9
11
21
F
T
12
T
20
13
All path coverage:if id ,address,balance,acc dateand sql is true
14
18
15
16

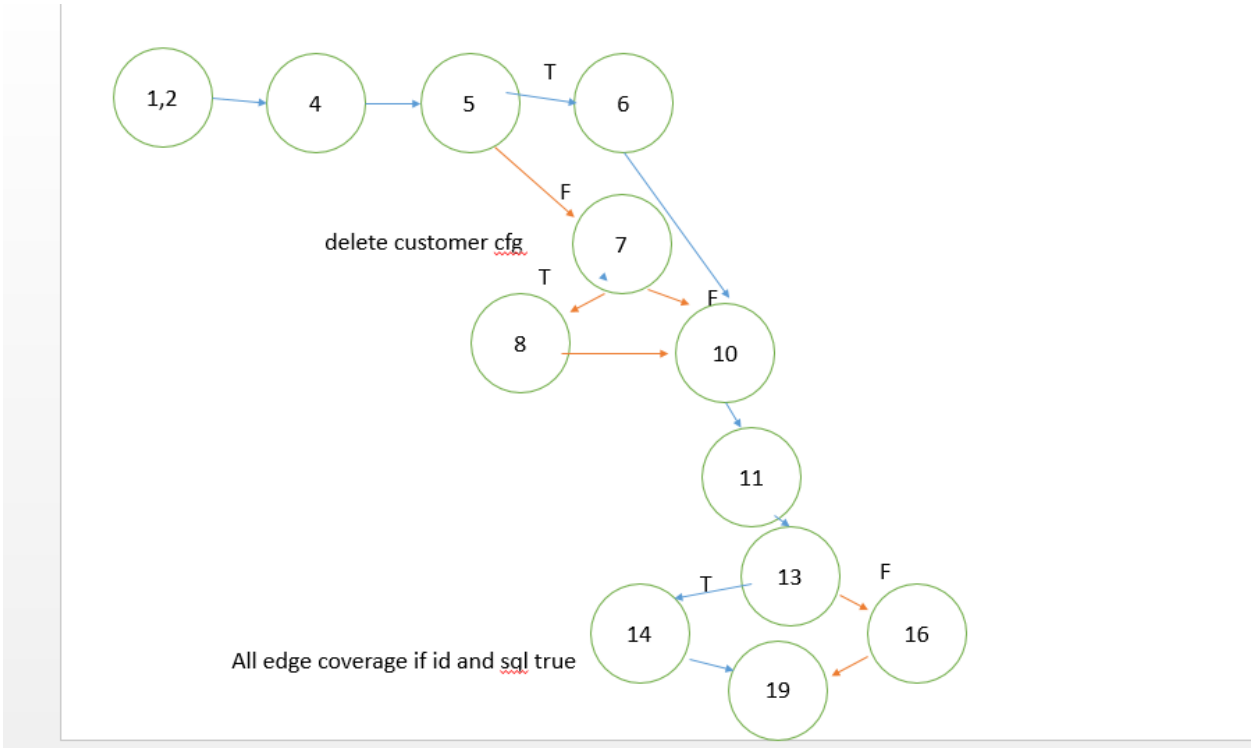

1,2,3
5
6
T
7
F
28
8
Create customer cfg
T
F
F
24
9
11
21
F
T
12
T
20
13
All conditione:if id,name,balance,address,acc date and sql is true
14
18
15
16

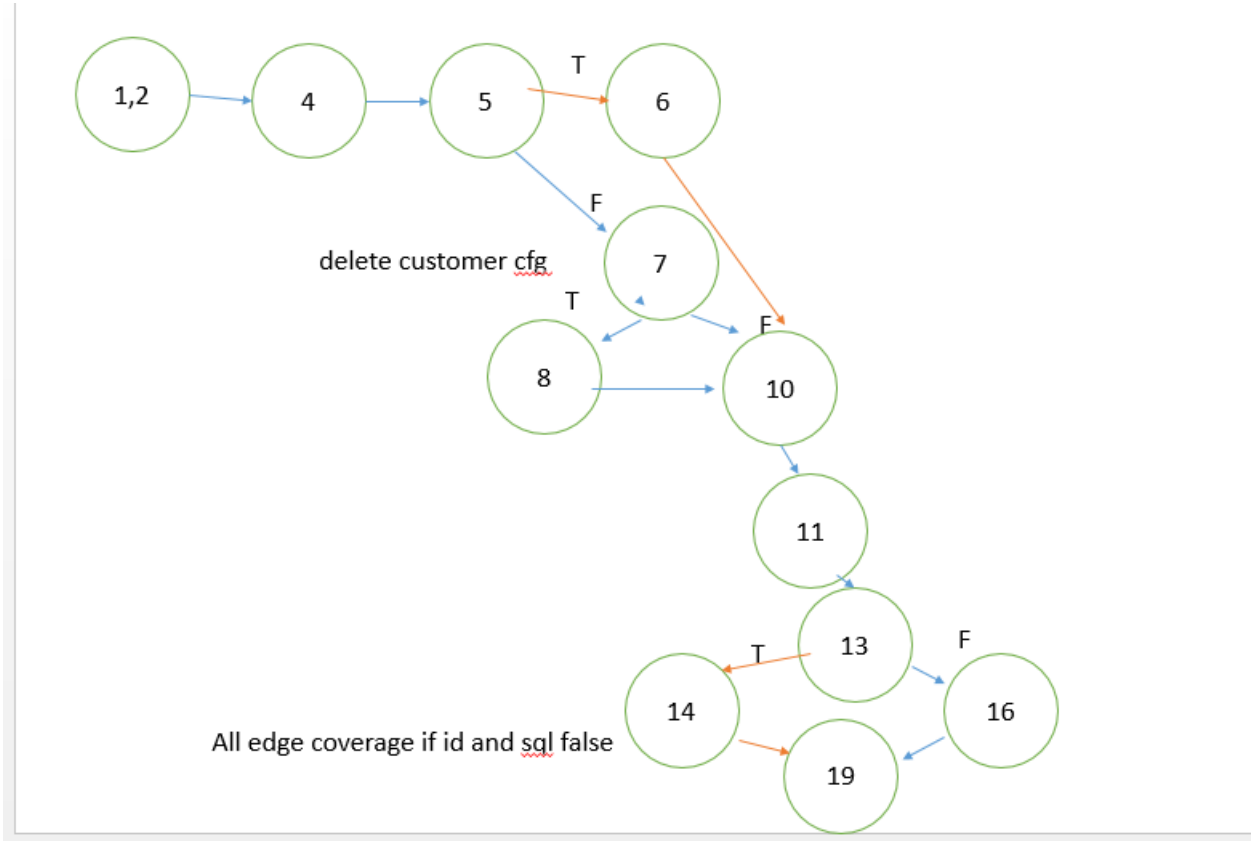

1,2,3
5
6
T
7
F
8
Create customer cfg
T
F
9
11
F
24
28
T
12
21
F
T
20
13
All condition coverage:if id
,address,balance,acc dateand sql is true
14
15
16
18

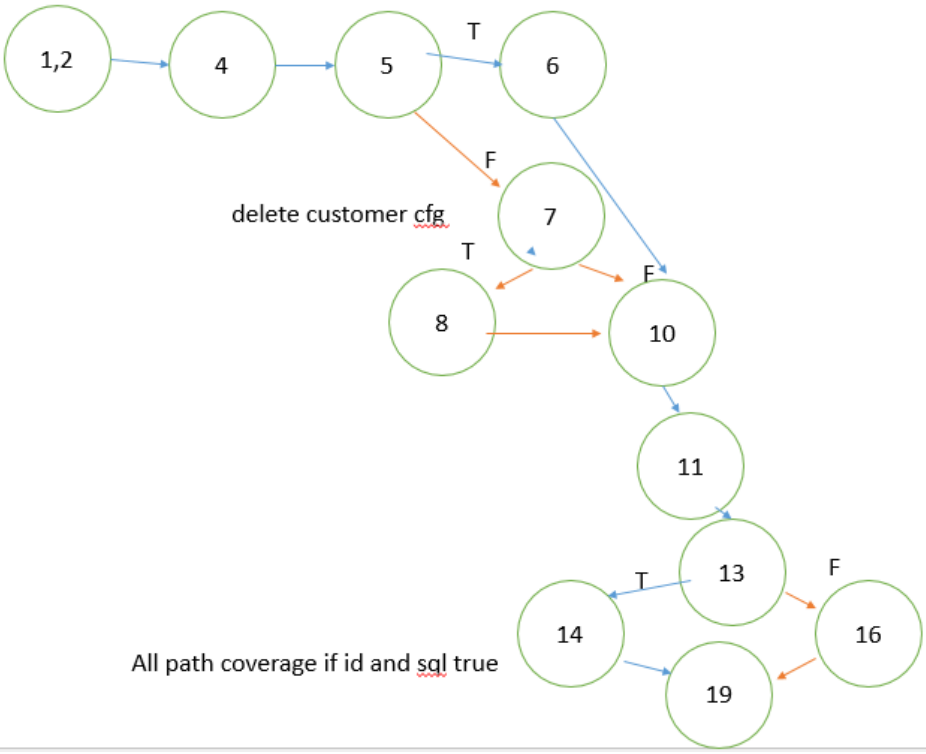

1,2
4
5
T
6
F
delete customer cfg
7
T
F
8
10
11
13
T
F
14
16
19


1,2
4
5
T
6
F
delete customer cfg
7
T
F
8
10
11
All statement coverage if id and sql true
T
13
F
14
16
19

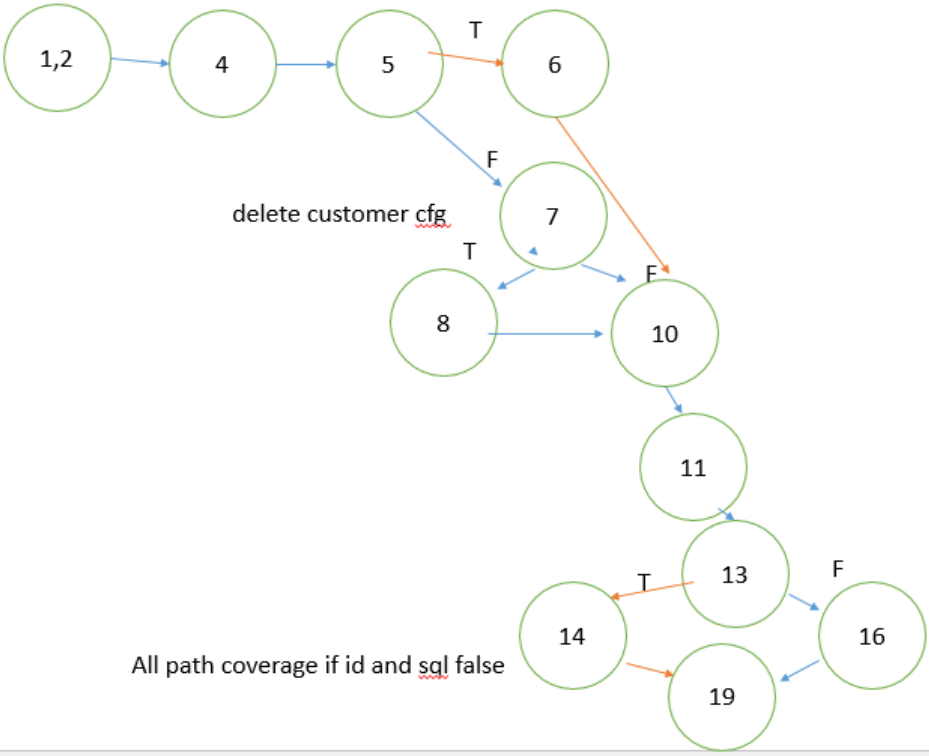

1,2
4
5
T
6
F
delete customer cfg
7
T
F
8
10
11
All statement coverage if id and sql flase
T
13
F
14
16
19


1,2
4
5
T
6
F
delete customer cfg
7
T
F
8
10
11
13
T
F
14
16
19
All edge coverage if id and sql true


1,2
4
5
T
6
F
delete customer cfg
7
T
F
8
10
11
13
T
F
14
16
19
All edge coverage if id and sql false

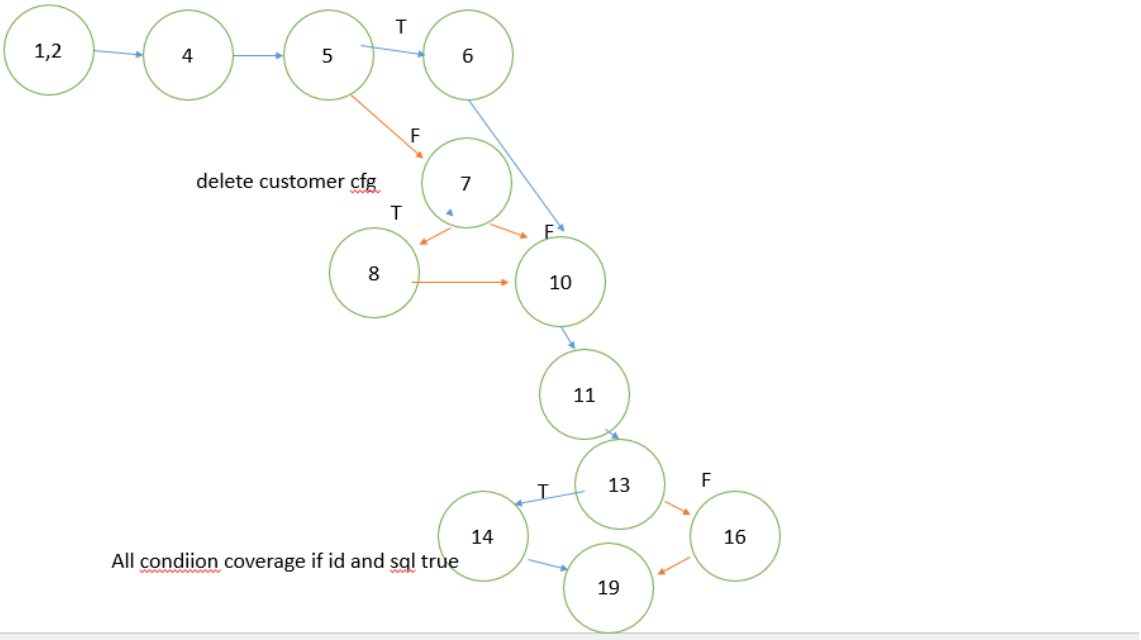


All path coverage if id and sql true



All path coverage if id and sql false


1,2
4
5
T
6
F
delete customer cfg
7
T
F
8
10
11
13
T
F
14
16
All condiion coverage if id and sql true
19


1,2
4
5
T
6
F
delete customer cfg
7
T
F
8
10
11
13
T
F
14
16
All condiion coverage if id and sql true
19

The End