BANK MANAGEMENT SYSTEM

ADITYA BATTULA (RA2111026010449)

09-11-2022

Object Oriented Design and Programming (18CSC202J)

S.NO	TITLE

1	Problem Statement & Aim
2	Use Case Diagram
3	Class Diagram
4	Sequence Diagram
5	Activity Diagram
6	Package Diagram
7	State Chart Diagram
8	Component Diagram
9	Communication Diagram
10	Result

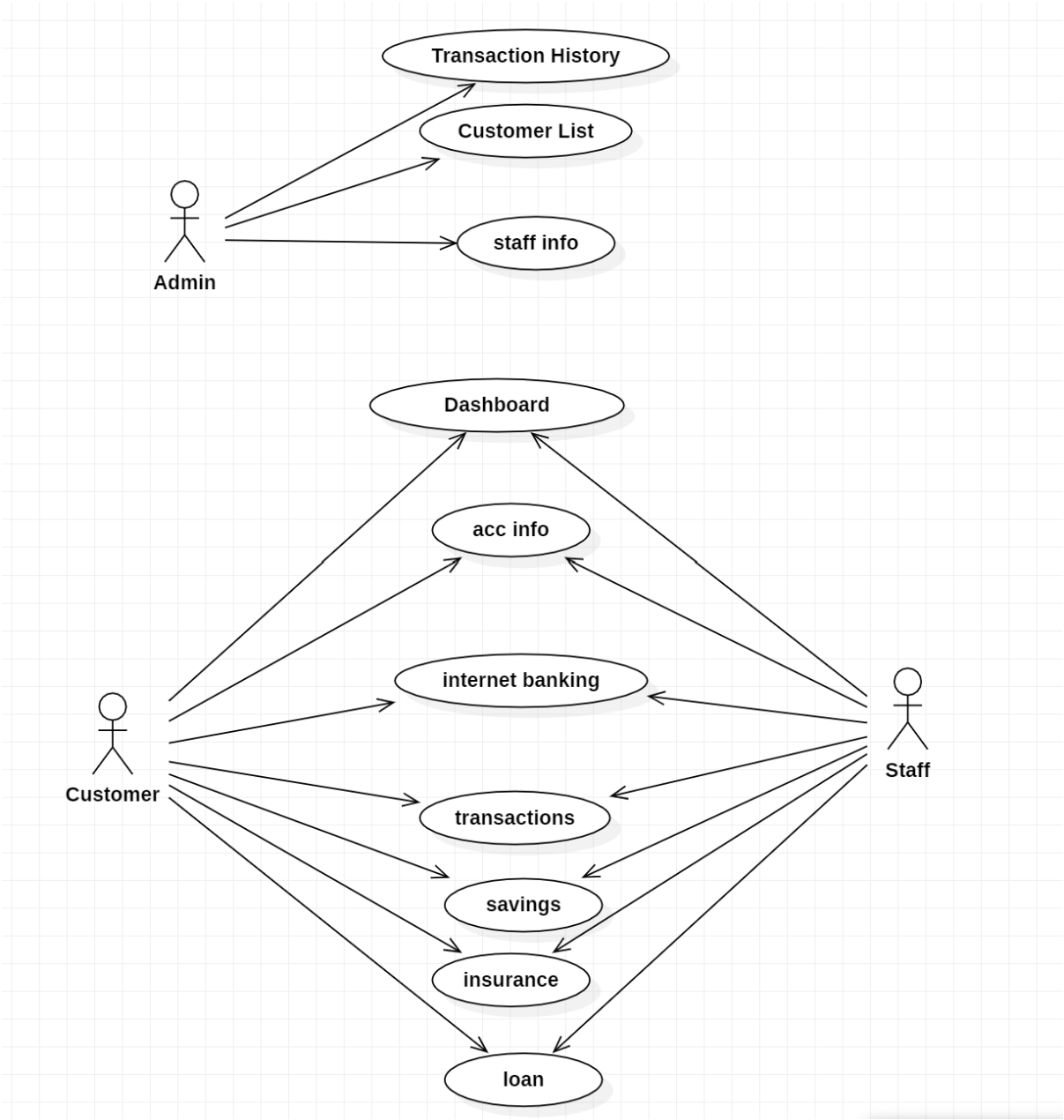
PROBLEM DESCRIPTION

Although many types of bank management systems offer various ways to perform bank operations through various locations and websites, it is crucial for there to be a system where bank management is easy, simple and effective. We are allotted to create and build various UML diagrams to better deploy the functions of bank management. The UML diagrams and code generations we provide will help to better understand the process and roles behind the system. We are allotted to provide UML diagrams of various models, for example class diagram, sequence diagrams, use-case diagrams. These diagrams help to better showcase the process behind the bank management system in various ways.

AIM

To develop a software for solving financial applications of a customer in banking environment in order to nurture the needs of an end banking user by providing various ways to perform banking tasks. Also, to enable the user's workspace to have additional functionalities which are not provided under a conventional banking software.

USE CASE DIAGRAM



USE CASE DIAGRAM – BANK MANAGEMENT SYSTEM

Use Case Documentation

Description:

This Use Case Diagram is a graphic depiction of the interactions among the elements of Banking Management System. It represents the methodology used in system analysis to identify, clarify, and organize system requirements of Banking management system. The main actors of Banking management System in this Use case diagram are: admin, staff, customer, who perform the different types of use cases such Bank transaction history, Customer list, staff information, Dashboard access, account information, internet banking, transactions, savings account, insurance operations, loan operations etc. Major elements of the UML use case diagram of Banking management system are shown on the picture above.

The relationships between and among the actors and the use cases of Banking management System:

- **Admin Entity:** Use cases of Admin are Transaction History of the bank, Customer List, staff information.
- **Customer Entity:** Dashboard, account information, internet banking, transactions, savings, insurance, loan.
- **Staff Entity:** Dashboard, account information, internet banking, transactions, savings, insurance, loan.

Customer List Use case

Use case name	Customer List
Actor	Admin
Pre-condition	Admin must be logged in
Post condition	-
Include	Authentication
Frequency of use	Not often
Normal course of events	Admin clicks on customer list under banks info.

Staff list Use case

Use case name	Staff List
Actor	Admin
Pre-condition	Admin must be logged in
Post condition	-
Include	Authentication
Frequency of use	Not often
Normal course of events	Admin clicks on staff list under banks info.

Customer List Use case

Use case name	Customer List
Actor	Admin
Pre-condition	Admin must be logged in
Post condition	-
Include	Authentication
Frequency of use	Not often
Normal course of events	Admin clicks on customer list under banks info.

Transaction history Use case

Use case name	Transaction History
Actor	Admin
Pre-condition	Admin must be logged in
Post condition	-
Include	Authentication
Frequency of use	Not often
Normal course of events	Admin clicks on transaction history under banks info.

Dashboard Use case

Use case name	Dashboard
Actor	Staff, customer
Pre-condition	Customer must be logged in
Post condition	-
Include	Authentication
Frequency of use	Not often
Normal course of events	customer clicks on dashboard to view various banking operations.

Account information Use case

Use case name	Account Information
Actor	Staff, customer
Pre-condition	Customer must be logged in
Post condition	-
Include	Authentication
Frequency of use	Not often
Normal course of events	customer clicks on account information to view customers name, address, account number and several details.

Internet banking Use case

Use case name	Internet banking
Actor	Staff, customer
Pre-condition	Customer must be logged in and select online banking
Post condition	-
Include	Authentication
Frequency of use	Most times
Normal course of events	customer clicks on internet banking in the dashboard dropdown.

Transactions Use case

Use case name	Transactions
Actor	Staff, customer
Pre-condition	Customer must be logged in and select transaction history.
Post condition	-
Include	Authentication
Frequency of use	often
Normal course of events	customer clicks on transaction history in the dashboard dropdown.

Savings Use case

Use case name	Savings
Actor	Staff, customer
Pre-condition	Customer must be logged in and select savings account.
Post condition	-
Include	Authentication
Frequency of use	often
Normal course of events	customer clicks on savings in the dashboard dropdown.

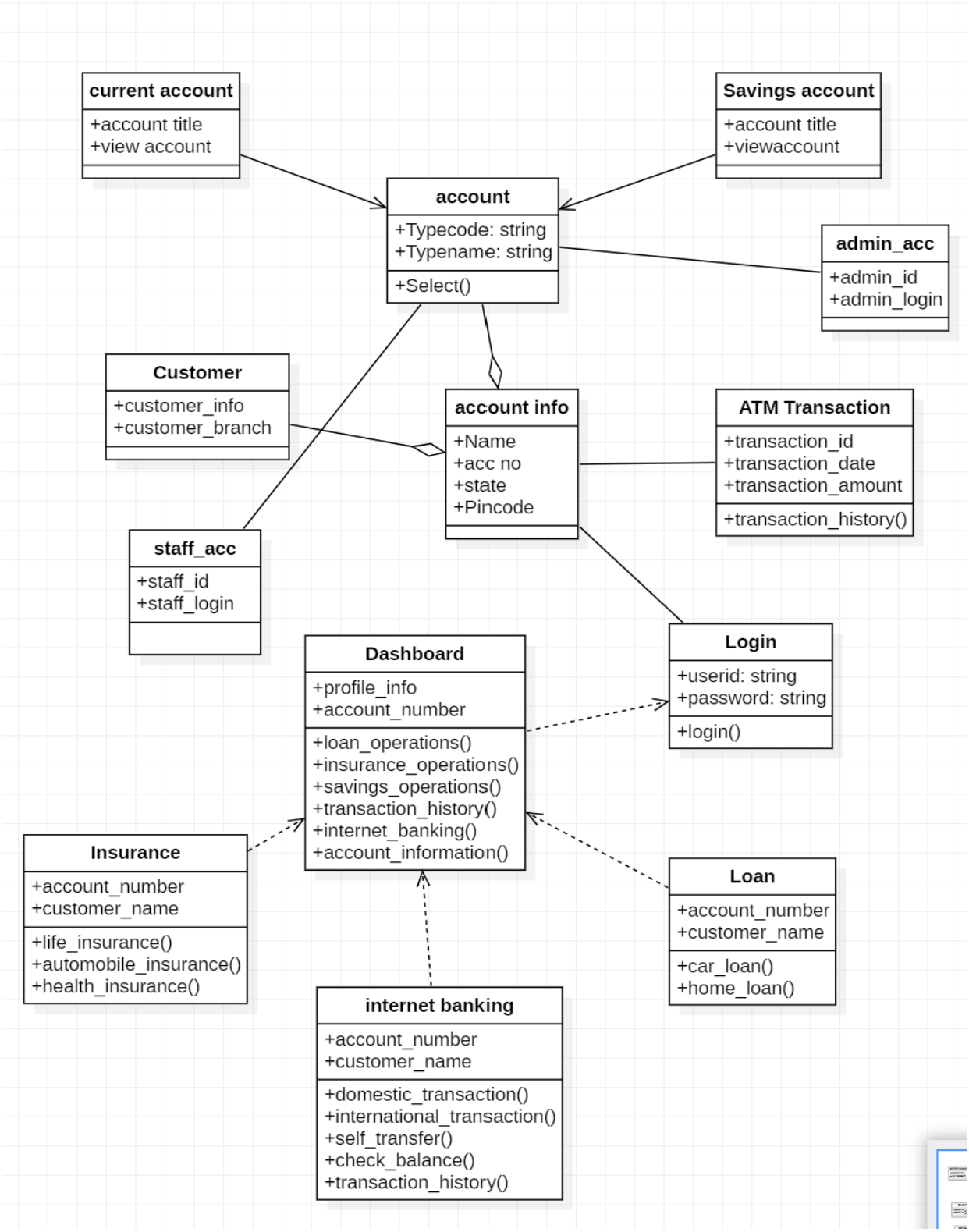
Insurance Use case

Use case name	Insurance
Actor	Staff, customer
Pre-condition	Customer must be logged in and select on insurance operations.
Post condition	-
Include	Authentication
Frequency of use	often
Normal course of events	customer clicks on insurance in the dashboard dropdown.

Loan Use case

Use case name	Loan
Actor	Staff, customer
Pre-condition	Customer must be logged in and select loan operations.
Post condition	-
Include	Authentication
Frequency of use	often
Normal course of events	customer clicks on loan operations in the dashboard dropdown.

CLASS DIAGRAM



Class Diagram – Bank Management System

Class Diagram Description

Description:

Banking Management System Class Diagram describes the structure of a Banking system classes, their attributes, operations (or methods), and the relationships among objects. The main classes of the Banking Management system are Customer, admin, internet banking, loan, insurance, staff acc, ATM transactions. These classes individual hold different attributes and different operations that are required for these classes to function in a sequential and meaningful manner.

The relationships between and among the main classes and the operations and attributes of Banking management System:

- **Dashboard class**: consists all the operations that can be accessed by the logged in users.
- **Staff account class**: consists of all the attributes regarding the staff account.
- **Customer class**: consists of all the attributes regarding the customer class.

- **ATM transaction class**: includes all the attributes and operations regarding transactions.
- **Loan class**: all the operations and attributes are included in this class regarding loan operations.
- **Insurance class**: all the operations and attributes are included in this class regarding insurance operations.
- **Internet banking class**: includes all the attributes and operations that can be accessed through the dashboard for the logged in customer.
- **Account info class**: this class includes all the account information of each individual customer and their account details.

C++ Code Generation for Class Diagram:

Internet banking class:

```cpp
/**
 * Project Untitled
 */


#include "internet banking.h"

/**
 * internet banking implementation
 */


void internet banking::domestic_transaction() {

}

void internet banking::international_transaction() {

}

void internet banking::self_transfer() {

}

void internet banking::check_balance() {

}

void internet banking::transaction_history() {

}
```

Account information class:

```
/**
 * Project Untitled
 */


#include "account info.h"

/**
 * account info implementation
 */


void account info::Operation1() {

}
```

ATM Transaction class:

```
/**
 * Project Untitled
 */


#include "ATM Transaction.h"

/**
 * ATM Transaction implementation
 */


void ATM Transaction::transaction_history() {

}
```

Current account class:

```
/**
 * Project Untitled
 */


#include "current account.h"

/**
 * current account implementation
 */


void current account::account title() {

}

void current account::viewaccount() {

}
```

Customer class:

```
/**
 * Project Untitled
 */


#include "Customer.h"

/**
 * Customer implementation
 */
```

Admin class:

```
/**
 * Project Untitled
 */


#include "admin_acc.h"

/**
 * admin_acc implementation
 */
```

Staff class:

```
/**
 * Project Untitled
 */


#include "staff_acc.h"

/**
 * staff_acc implementation
 */


void staff_acc::Operation1() {

}
```

Insurance class:

```
/**
 * Project Untitled
 */


#include "Insurance.h"

/**
 * Insurance implementation
 */


void Insurance::life_insurance() {

}

void Insurance::automobile_insurance() {

}

void Insurance::health_insurance() {

}
```

Dashboard class:

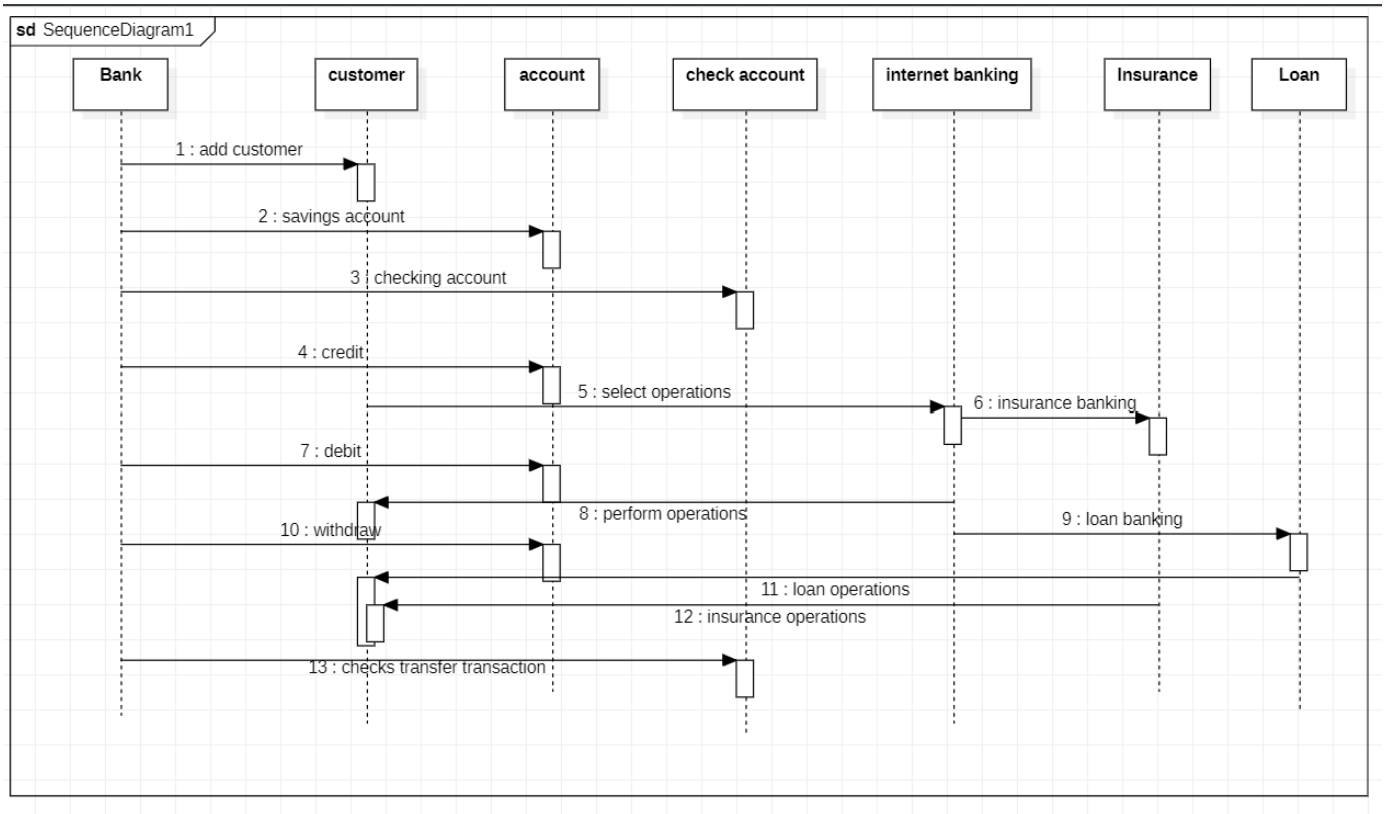
```
/**
 * Project Untitled
 */


#include "Dashboard.h"

/**
 * Dashboard implementation
 */


void Dashboard::loan_operations() {

}

void Dashboard::insurance_operations() {

}

void Dashboard::savings_operations() {

}

void Dashboard::transaction_history() {

}

void Dashboard::internet_banking() {

}
```

Loan class:

```cpp
/**
 * Project Untitled
 */


#include "Loan.h"

/**
 * Loan implementation
 */


void Loan::car_loan() {

}

void Loan::home_loan() {

}
```

SEQUENCE DIAGRAM

Sequence Diagram – Bank Management system



Description:

This is the UML sequence diagram of Banking management system which shows the interaction between the objects of customer interaction with the banking system, account creation, internet banking, loan operations, insurance operations, checking account details and account information.

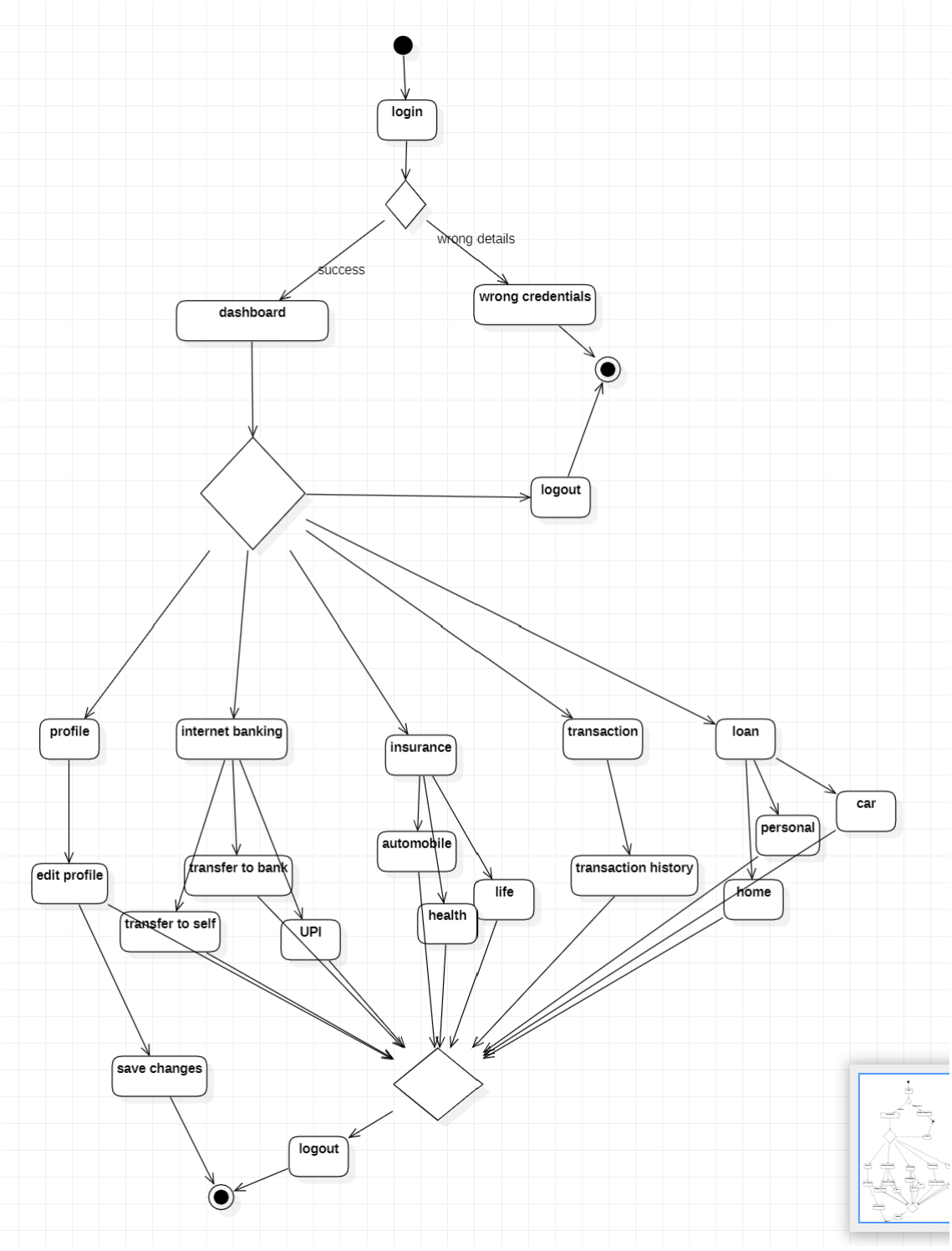
Sequence operations include:

- Add customer
- Savings account
- Current account
- Checking account
- Credit
- Internet banking
- insurance
- loan
- withdraw
- debit

The sequence diagram above helps to better follow the operations handled by the actors and different classes that we provide in the class diagrams. These

actions also get a reaction that is sent back to the customer as their result after they have performed their operation that they have intended.

ACTIVITY DIAGRAM



Activity diagram – Bank Management System

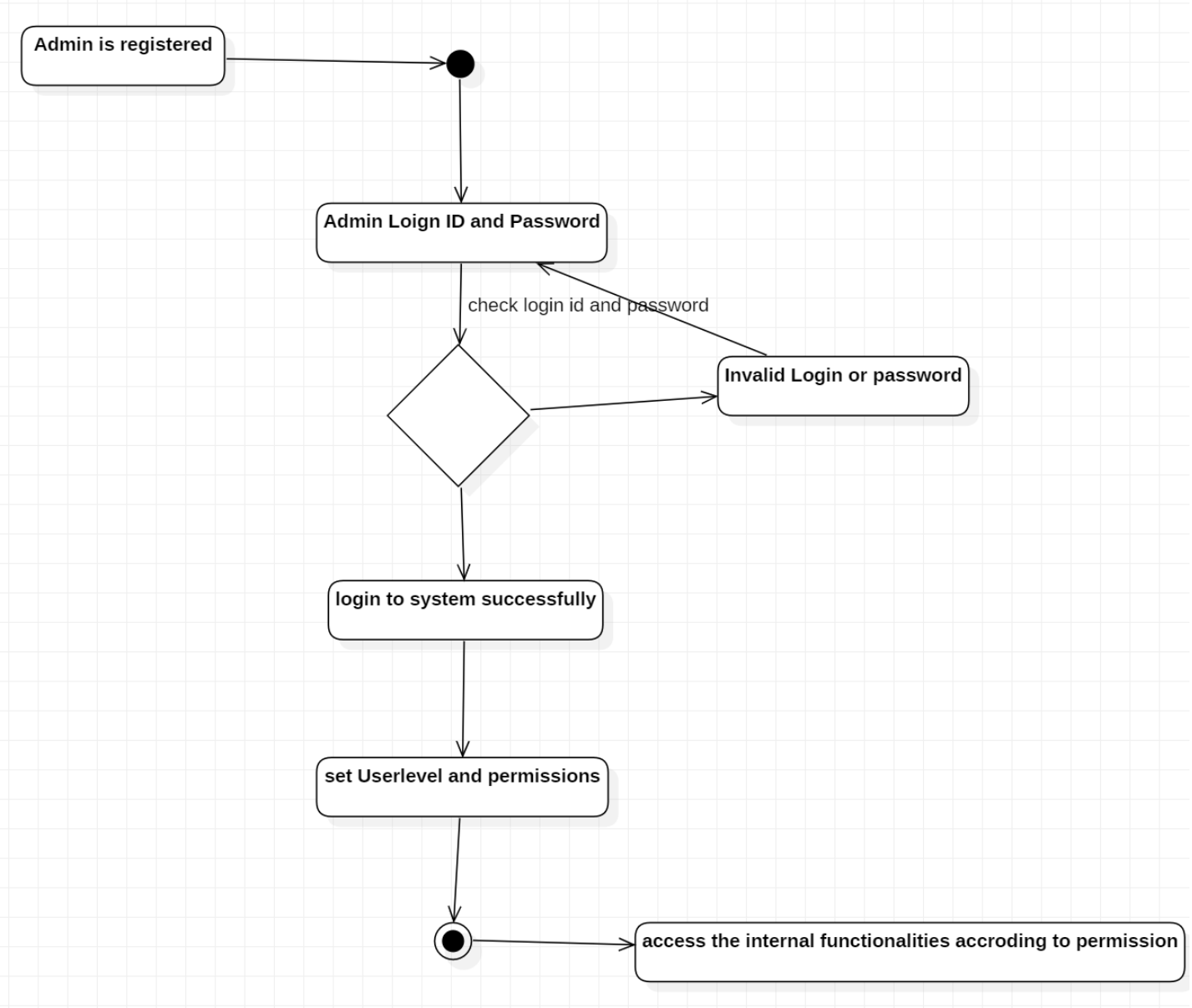
Activity Diagram Description:

This is the Activity UML diagram of Banking Management system which shows the flows between the activity of profile, internet banking, transactions, insurance, loan. The main activity involved in this UML Activity Diagram of Banking Management system are as follows:

- Profile Activity
- Internet Banking Activity
- Transactions Activity
- Insurance Activity
- Loan activity

Features of the Activity UML Diagram of Banking Management System

- The customer user can navigate through the dashboard to view and perform various functions including, loan, insurance, transactions and other options.
- It shows the activity of how a customer user gets from login to perform their desired actions after their login.
- All objects such as (Loan, Insurance, transactions, profile) are interlinked.
- It shows the flow of balance through the activities that can be performed by the customer through login and logout.



Admin Login activity diagram of Banking management system

Description:

This is the Login Activity Diagram of Banking Management System, which shows the flows of Login Activity, where admin will be able to login using their username and password. After login user can manage all the operations on loan, insurance, loan, transactions, profile and future more. All pages such as profile, insurance, loan are secret and user can access page after their login. The diagram above helps demonstrate how the login page works in a Banking management system. The various objects in a Banking Management System. The various objects in the Loan, insurance, profile and further more present in the dashboard – interact over the course of the activity, and user will not be able to access their page without verifying their identity through the login page.

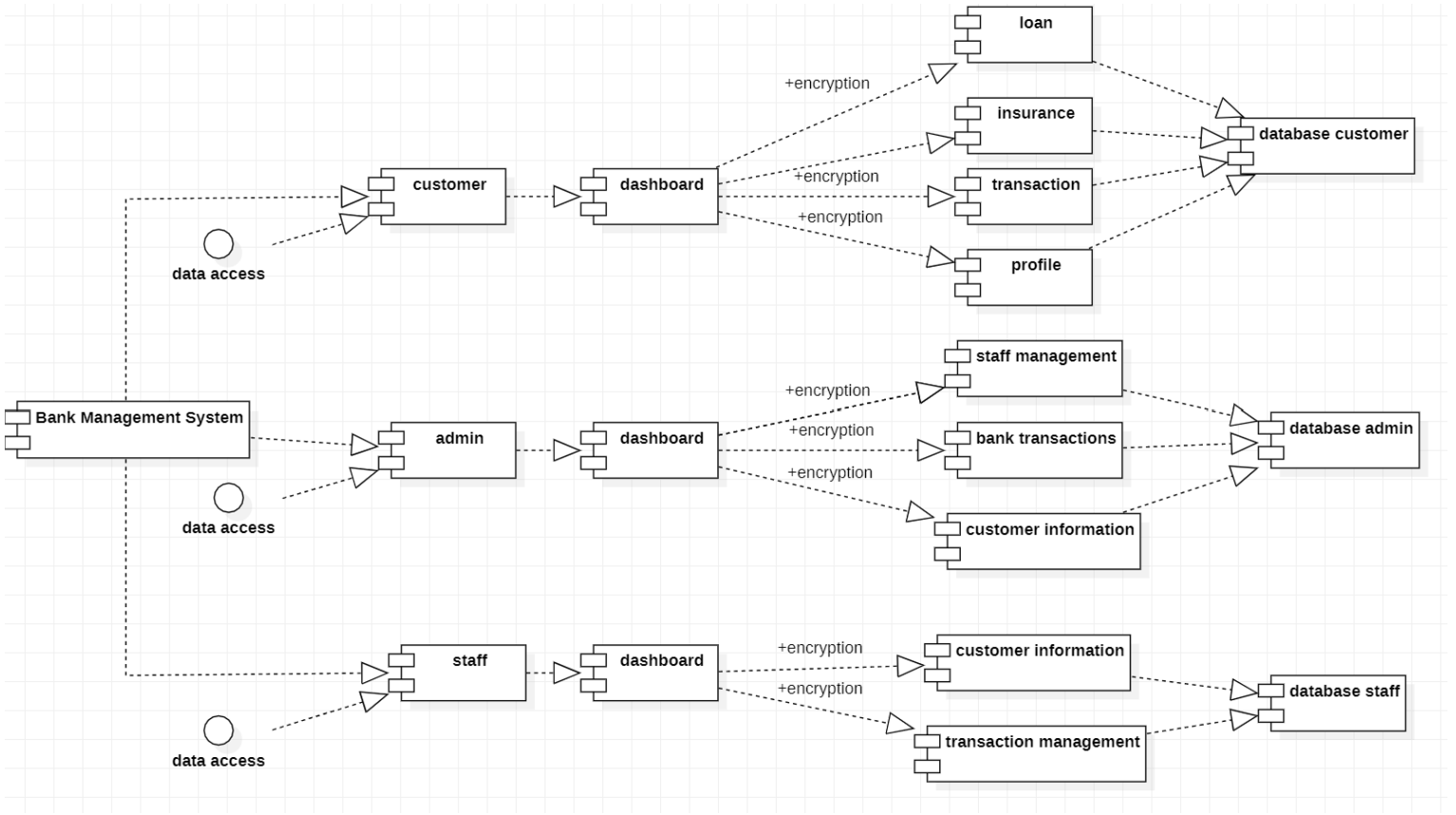
This activity diagram includes:

- Admin is registered
- Admin Login ID & password
- Check Login ID & password
- Decision block: if login id and password is valid
- If invalid go back to login page

- Login to system successfully
- Set admin level and permission
- Access the internet functionalities according to permission

The admin login page is crucial part of the banking management process as it accesses the data regarding the wither the customer,admin or staff. These three classification is done through the login page and the activity diagram above shows the login sequence activity diagram.

COMPONENT DIAGRAM



Component Diagram – Bank Management System

Description:

This is a component diagram of banking management system which shows components, provided and required interfaces, ports, and relationships between admin database, customer database, and staff database. These three different components have different login id and passwords with different login pages respectively. This type of diagrams is used in Component-Based development to describe systems with Service-Oriented Architecture. Banking management system UML component diagram, describes the organization and writing of the physical components in a system.

Components of UML Component Diagram of Banking Management System:

Customer account component:

- Dashboard component:
    - Loan component

      The loan component manages all the loan operations done by the user or customer in the management system. They can apply for different types of loan including house loan, car loan, student loan etc, depending on their need for financial support.
    - Insurance component

The Insurance component present inside the customer interface gives them the ability to browse through different insurance plans provided by the particular bank.

- Transactions component

  Transactions component present in the customer operations is very crucial as it enables various types of transactions which includes domestic, international etc.

- Profile component

  The profile component lets the user edit his profile details which include name, phone number, email and address etc.

Admin account component:

- Dashboard component
  - Staff management component

    The staff management component present inside the admin user interface lets the admins to manage the staff work hours, wage and their progress on how they are full filling their role as a staff member monitoring all the transactions.

  - Bank transactions component

    The bank transactions component lets the admin monitor the overall transactions that are coming in and going out the bank.

How much money is being transported on a daily basis in and out the bank.

- Customer information component

  The customer information component present in the admin actions lets the admin monitor and gather the customer information provided by the user while they login at the start. This lets the admin monitor from where the transactions are being made and by whom.

Staff account component:

- Dashboard component
  - Customer information component

    The customer information component is also present inside the staff action list same as it is present inside the admin actions. But here the staff has a limited access as they can edit the customers details in their profile upon request from the bank customer.
  - Transaction management component

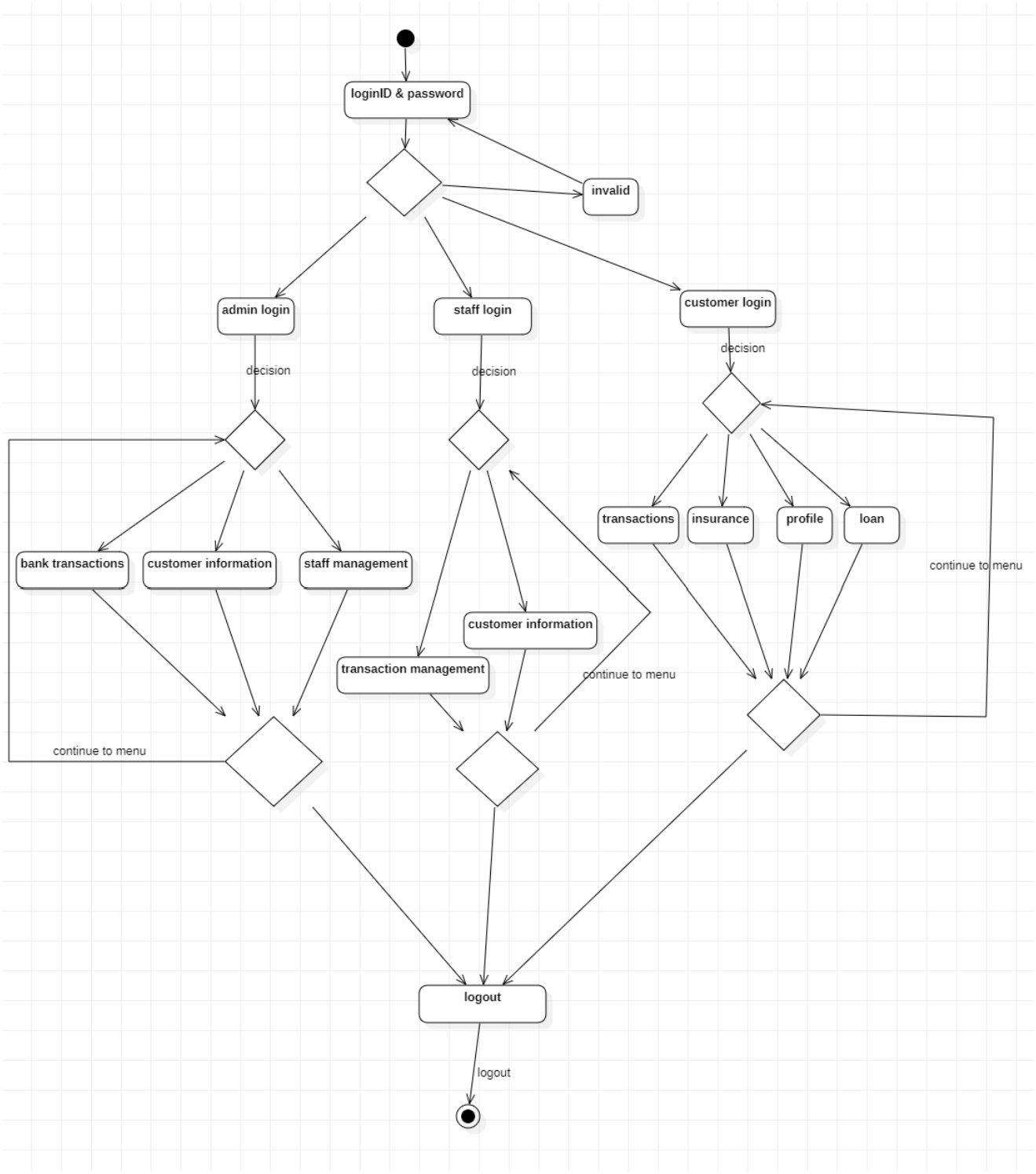
    The transaction management component lets the staff monitor any money that needs to be inputted into the bank or withdrawn from the bank. This process can only be done either through the

customers request to transfer money from their account to the bank or vice versa.

The component diagram in UML plays a crucial role in showcasing the different databases and different fields that are provided to of from different divisions. In this case the divisions include customer, staff, admin.

STATE CHART

DIAGRAM



STATE CHART DIAGRAM – Bank Management System

Description:

State chart diagram is one of the five UML diagrams used to model the dynamic nature of a system. They define different states of an object during its lifetime and these states are changed by events. State chart diagrams are useful to model the reactive systems. Reactive systems can be defined as a system that responds to external or internal events.

State chart diagram describes the flow of control from one state to another state. States are defined as a condition in which an object exists and it changes when some event is triggered. The most important purpose of State chart diagram is to model lifetime of an object from creation to termination.

Components of State chart diagram for Bank management system:

- Login id & password

  First the state we go through is login state when the customer, admin or staff is asked to sign in using their username and password allotted to them or set by them depending on their role.

- Customer state
  - Profile

    During the profile state the customer is allowed to edit their profile information including their address and further bank details and UPI ID to fasten transactions.

- Transactions

  During the transaction state the customer is enabled to perform different actions regarding multiple money transfers from account to account through either domestic or international transfer.

- Loan

  The loan state in the state chart diagram above enables the customer to perform loan operations on whether to obtain home loan, car loan or any other types of loan that the bank can offer to customers in need of financial aid.

- Insurance

  The insurance state lets the registered user to navigate through several insurance operations that the bank is able to offer, some of these include life insurance, health insurance, automobile insurance and etc that are a necessity for the customers.

- Admin state:
  - Bank transactions

    This state present inside the admin phase of the state chart diagram enables the admin user to monitor bank transactions on a

day to day basis and analyse the income, profit and losses that the bank is gain through they insurances, loan plans etc.

- Customer information

  Customer information state lets the admin supervise the customer information and implement new plans to attract new customer. This information is really important for the banks to gain a customer base that is loyal to their bank.

- Staff management

  Staff Management is the state where the admin user is allowed to manage all the staff work hours and wages day to day as it is important to make sure the employees are getting paid on time with bonus and promotions.

- Staff State:
  - Transaction management

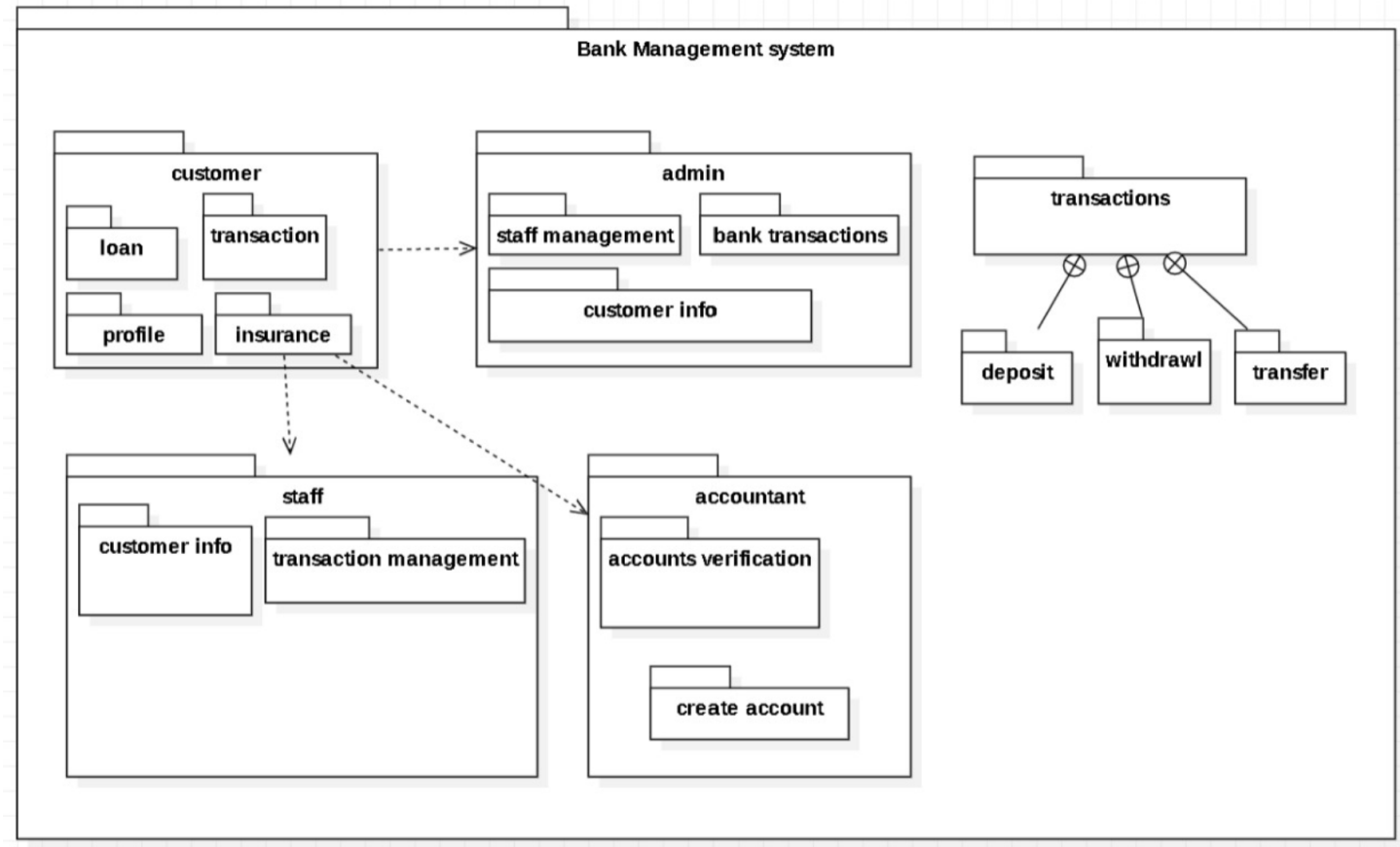
    In this state the staff is allowed to manage customer transactions upon their request if they are facing some problems themselves. It is really important for the staff to take care of their customer and provide good experience.

  - Customer information

The customer information is also provided to the staff to better manage day to day transactions of there are any problems that are being faced by the customers.

- Then if the user wants to continue with their actions they are resend to the menu page where they are allowed to select the action again or they can logout. The state chart Is directed to end statement.

PACKAGE DIAGRAM



Package Diagram – Bank Management System

Description:

Package diagram is used to simplify complex class diagrams, you can group classes into packages. A package is a collection of logically related UML elements.

The diagram below is a business model in which the classes are grouped into packages: Packages appear as rectangles with small tabs at the top. The package name is on the tab or inside the rectangle. The dotted arrows are dependencies. One package depends on another if changes in the other could possibly force changes in the first.

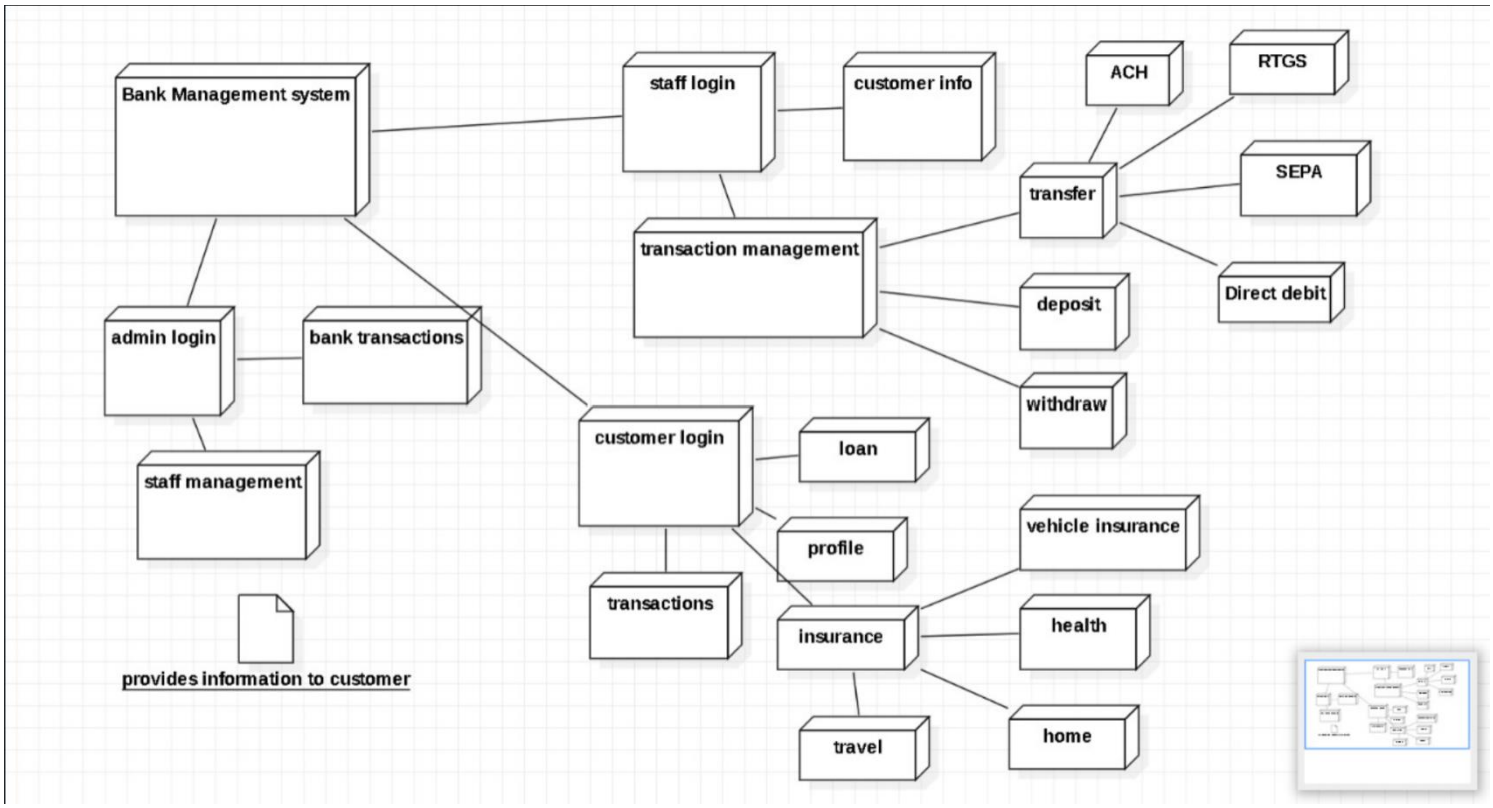
This Package Diagram with Bank Management system includes:

- Customer package
    - Loan
    - Transaction
    - Insurance
    - Profile
- Admin Package
    - Staff management
    - Bank transactions
    - Customer information

- Transaction package include:
    - Deposit
    - Withdrawl
    - Transfer
- Accountant package
    - Accounts verification
    - Create account
- Staff package
    - Customer information
    - Transaction management

DEPLOYMENT

DIAGRAM



Deployment Diagram – Bank Management System

Description:

The term Deployment itself describes the purpose of the diagram. Deployment diagrams are used for describing the hardware components, where software components are deployed. Component diagrams and deployment diagrams are closely related. Component diagrams are used to describe the components and deployment diagrams shows how they are deployed in hardware. UML is mainly designed to focus on the software artifacts of a system. However, these two diagrams are special diagrams used to focus on software and hardware components. Most of the UML diagrams are used to handle logical components but deployment diagrams are made to focus on the hardware topology of a system. Deployment diagrams are used by the system engineers.

The purpose of deployment diagrams can be described as −

- Visualize the hardware topology of a system.
- Describe the hardware components used to deploy software components.
- Describe the runtime processing nodes.

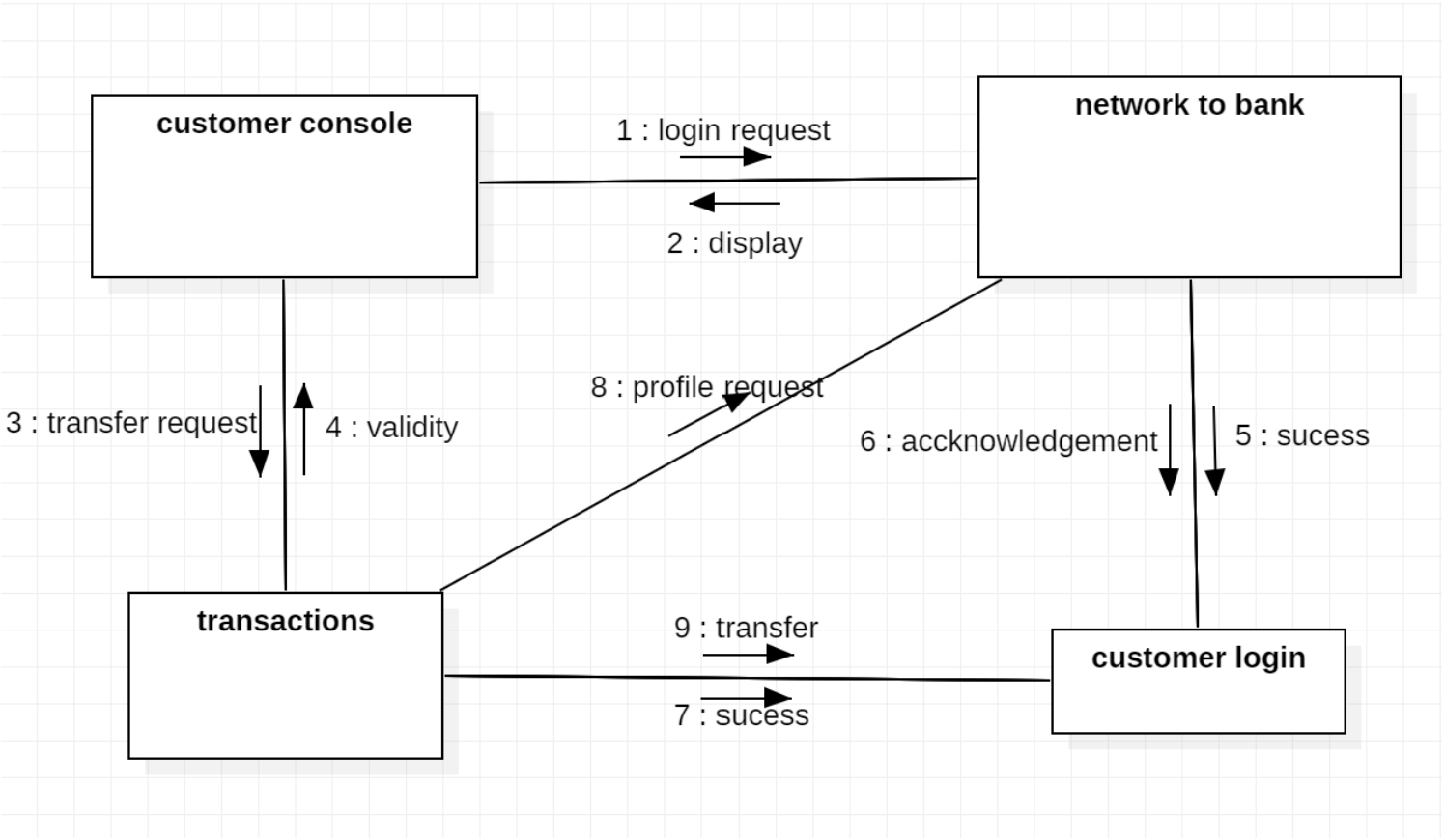
Components present in this bank management system are:

- Customer component

    - Loan
    - Transaction
    - Insurance
    - Profile
- Admin component
    - Staff management
    - Bank transactions
    - Customer information
- Transaction component include:
    - Deposit
    - Withdrawl
    - Transfer
- Accountant component
    - Accounts verification
    - Create account
- Staff component
    - Customer information
    - Transaction management

COMMUNICATION

DIAGRAM



Communication Diagram –

Bank Management System

Description

UML Communication Diagrams, previously known as collaboration diagrams are a type of behavioural diagram that shows the interactions that take place between objects in a piece of software or system. This type of diagram emphasizes the messages exchanged between objects. Communication diagrams are best used when one use case has multiple scenarios that need depicting.

UML Communication Diagrams, previously known as collaboration diagrams are a type of behavioural diagram that shows the interactions that take place between objects in a piece of software or system. This type of diagram emphasizes the messages exchanged between objects. Communication diagrams are best used when one use case has multiple scenarios that need depicting.

Show the passing of messages between objects in a system or piece of software.

Depicting the interactions between objects

Visualize how messages are sent and received between objects, as well as the consequences

Communication diagram for Bank management system:

- Customer console to network to bank
- Network to bank to customer console
- Bank network to customer login
- Transactions to customer login
- Transactions to customer console

RESULT:

All the UML diagrams have been constructed successfully using StarUML and the descriptions of all the UML diagram.