|  |
| --- |
| Photo displaying partial image of two pie charts on a canvas-textured page |
| BANKING SYSTEM  DB DESIGN PROJECT |
| |  |  |  | | --- | --- | --- | | OLGA SLIPAK | 10/24/17 | Vanier College | |

Table of Contents

1. **System Planning** 
   1. Business case Reason/justification..……………………………………………………………………….4
   2. SWOT Analysis(what’s strength, weakness, opportunity, threat)…………………………….4

1.2.1 strength………………………………………………………………………………………………………………4

1.2.2. weakness……………………………………………………………………………………………………………5

1.2.3. opportunity…………………………………………………………………………………………..……………5

1.2.4. threat…………………………………………………………………………………………………………………5

* 1. Preliminary investigation………………………………………………………………………………………5
     1. Understand problems………………………………………………………………………………………5
     2. Project Slope / constraints………………….……………………………………………………………5
     3. Fact-finding: end user input, rate hours……………………………………………………………5
     4. Analysis fact, data usability……………………………………………………………………………5-6
     5. Separate project feasibility………………………………………………………………………………6
     6. Present resolved recommendation…………………………………………………………………..6
  2. Project feasibility……………………………………………………….………………………………………..6-7
     1. Operational feasibility………………………………………….…………………………………………..6
     2. Technical feasibility…………………………..………………….…………………………………………..6
     3. Economic feasibility (economic valuable)………….………………………………………………6
     4. Schedule feasibility (MS Project) ……………………….…………………………………………..6-7
  3. Project management…………………………………………………… ……………………………………..7-9
     1. Split project into tasks (has to be verifiable by user, measure tasks with the end user) ……………………………………………….………………………..…………………………….……..7-9
     2. Project scheduling time (MS Project, tasks and duration, Giant chart: tasks duration, PERT(evaluation permit, Diagram chart, tasks patterns)……………………………………………………………………………………………….……9-12
     3. Project Triangle (time, scope, costs)………………………………………………………….……12
  4. Project Risk/Probability (Matrix: risk and probability) ……………………………..……13

1. **System Analysis (Dia)**
   1. Client questionnaire (design questions: how many people, very often, mobile application or web, requirements of application)……………………………………………………….……………………………………………..…..13-14
   2. Key questions who where why which how when system will be used…………………..15
   3. Requirement modeling……………………………………………………………………………………15-17
      1. Input (number of hours – numerical, name employee, SSN)…………………..…15-16
      2. Output (total income what represent)………………………………………………..…………..16
      3. Process (method formula for total income)………………………………………….……16-17
      4. Performance (web site need or no, scalability of users)………………………………….17
      5. Security issue (for account)……………………………………………………………….……………17
   4. Data and Process Modeling…………………………………………………………………………..….17-18
      1. Function Decomposition Diagram (capture modules)………………………………….17-18
      2. Data Flow Diagram………………..…………………………………………………………………….….18
   5. UML diagram…………………………………………………………………………………………………19-20
      1. Use case diagram (person)………………………………………………………..…………………….21
      2. Class diagram …………………………………………………………………………….……………………20
      3. Sequence diagram (timing of request) ……………………………………….………………21-25
      4. Transition diagram…………………………………………………………………………………………..25
      5. Activity Diagram………………………………….………………………………………………………….26
   6. Data Dictionary …………………………………………………………………………………………….….26-27
2. **System Design (SQL developer, Access)** 
   1. Development strategies……………………………………………………………………………….………28
   2. Project Prototype(https://balsamiq.com/download/#)............................................28
      1. User Interface……………………………………………………………………………………………28-48
      2. Data structure………………………………………………………………………………………………….48
      3. Reports consideration………………………………………………………………………………………49
   3. Database normalization
      1. 1NF – one big table………………………………………………………………………………………….49-51
      2. 2NF – partial dependency………………………………………………………………………………..51-52
      3. 3NF – transition dependency…………………………………………………………………………52-53
      4. Dependency diagram ……………………………………………………………………………………………54
   4. Entity relationship diagram (Access/SQL development)……………………………………….55
   5. System architecture ………………………………………………………………………………………..55-56
3. **System Implementation (Java, Oracle)………………………………………………...…….………57**
   1. Front End System Java…………………………………………………………………………………………..57
   2. Back-End System (textfile, Oracle)………………………………… ……………………..…………….57
   3. Structure chart (for searching)……………………………………………………………………………..57
   4. Prototype ……………………….………………………………………………………………………………57-58
4. **Testing/Security** (for agile method process for spliced task; negative or positive feedback)
   1. Unit testing…………………………………………………………………………..……………………………….58
   2. System testing………………………………………………………………………………………………..58-59
   3. Integration testing………………………………………………………………………………….……………..59
5. **System Planning** 
   1. **Business case Reason/justification**

BANKING SYSTEM will be develop for small bank “VSB Bank” with 1000 clients. Process in a bank not already automatized and a lot repeated operations still take a lot of time for employees of a VSB Bank. Main operations for VSB Bank are operations with check account(on-demand account), credit account, deposit account, investment account and insurance.

Team of VSB Bank is 20 members and now is growing. Banking System can to issue questions like: calculate, search, update, delete, insert new charges, penalties, commissions for all type accounts, manage very quickly with data of entity client, entity account, entity charge, commission, penalties. VSB has loan book (60 credit agreements) and about 20 insurance agreements. Also, VSB bank works with collateral(guarantee). For efficient work, VSB bank needs automatized banking system for managing all data of all clients. This is purpose for developing Banking System.

Banking System has to automatize all this operations and help employees save time, increase productivity, increase profit of VBS Bank and quality of service for clients. For managers Banking System will give opportunity /possibility creating reports for analysis data, result, client, credits more easily and quickly.

* 1. **SWOT Analysis(what’s strength, weakness, opportunity, threat)**

**1.2.1 strength**

WEAKNESSES – STRENGTHS Internal scope of the project, our capacities and our limitations. Strong and weak points. High customer satisfaction

* Continuous improvement of product and quality
* Cost reduction strategy is good
* High employee morale.
* Less employee turnover
* Effective decision making

**1.2.2. weakness**

* Decision making will be delayed because of long procedure.
* Adaptability towards change is low
* Marketing of products is given less importance
* Experience and skilled employees are not in right position

**1.2.3. opportunity**

THREATS AND OPPORTUNITIES. External scope, the environment surrounding the project and his role in the success of the Project.

* Continuous innovation/improvement of Banking System
* Potential to grow in power tools, packaging and security systems
* It helps to complete with other companies
* Save time
* Increase customer satisfaction
* Reduce paper- based transactions, error&costs

**1.2.4. threats**

* Increase in competition
* Too many substitute products available in the market
* Continuous cost pressure
  1. **Preliminary investigation**
     1. **Understand problems**

Humanerrors and computational errors: many errors enabled by the system due to tedious computations required during data processing cost

Large storage space

Poorly generated records

Poor communication

Difficulty in data analysis

Takes a lot of time

* + 1. **Project Slope / constraints**

Files occupy a large storage space

Hardware needs

Maintenance

* + 1. **Fact-finding:**

User put Id\_client, Number\_account, Limit, Balance, Type\_Charge, Type\_Account, Type\_commission

* + 1. **Analysis fact, data usability**

Enable easy authorized modification of data

Enable easy retrieval data

* + 1. **Separate project feasibility**

Schedule is feasible. Project has duration 36 days, 162 hrs, costs 37 690.00 $.

* + 1. **Present resolved recommendation**

The existing organization of process / operations which is manual utilized by the VSB Bank for calculate, update, record charges, commissions, penalties, interests is not capable of satisfying of VSB Bank. As the number of the employees and clients continues to climb, the demands on the existing system will exceed its abilities, prompting the need for implementing an appropriate computerized system to complete with efficiency and time saving all the tasks currently done manually.

Based upon, we recommend VSB Bank which will automatized process of calculate, update, save, delete charges, commissions, penalties, interests, manage data of all clients.

* 1. **Project feasibility**
     1. **Operational feasibility**

Project is doable and makes effective use of the resources of VSB Bank.

Banking System improve service quality and will be used by VSB Bank. Banking System be a reduction in costs/expenses and an increase in benefits. Banking System offer effective controls to protect against fraud and guarantee accuracy and security of data and information. Banking System is feasible and expandable.

Users/employees and management adapt to the change very easy.

* + 1. **Technical feasibility**

New equipment be needed computers. The schedule is reasonable.

The required technology is available. The required resources are available - manpower- programmers, testers & debuggers, software (java, VB, C++) and hardware (computers, server, database).

* + 1. **Economic feasibility (economic valuable)**

Cost Benefit Analysis:

Development costs:

Development costs that are incurred during the development of the system are one time investment.

* Wages – 20$ per hour
* Equipment – 3000$

Operating costs (Operating costs are the expenses required for the day to day running of the system)

* Supplies – 1000$
* Overheads – 200$

Benefits = Income – Costs

Income = 100 000.00

Benefits = 100 000.00 – 300\*20 – 3000 – 1000 - 200 = 89 800$

* + 1. **Schedule feasibility (MS Project)**

Schedule is feasible. Project has 3 tasks.

First task is completed. Task 1 had duration 14 days, 94 hours, was completed 05 October.

Second task is completed, had duration 7 days, 22 hours and budget 13 040.00 $. The second task was completed 16 October 2017.

Third task has highlight color yellow and will finish at 06 November 2017.



* 1. **Project management**
     1. **Split project into tasks (has to be verifiable by user, measure tasks with the end user)**

**VSB Bank verified first task: CHARGE APPLICATION WITH 6 PROCESS:**

Search charge;

Calculate charges;

Update charges;

Delete charges;

Create new record of charges and save in database;

Search for general information about type charges.

In Database we have and use six tables with data: Client\_entity, Account\_type, Charge\_type\_insurance, Commission\_type, Penalties\_type, Account\_entity.

Every client has one or more different type account (table account\_type: check account, deposit account, investment account, credit account). Depending on type account client can have charge, commission, penalties. Rate, day to pay of charge, commission, penalties depending on type (tables: charge\_type, commission\_type, penalties\_type).

VSB Bank has following formula for calculation:

There are three type of charge:

CIP4000 - Charge of personal insurance for credit account, rate – 0.5%, periodicity for calculation - per year

CIPL4000 - Charge of pledge for credit account, rate – 5%, periodicity for calculation - per year

CIC4000 - Charge of student for credit account, rate – 0.3%, periodicity for calculation - per year

CIC0 – NO CHARGE

Formula for calculation:

Charge = rate\_charge \* limit

There are two type of commission:

C500 - Commission for using cheking account, rate – 0.3%

C501 - Commission for using credit account, rate – 0.5%

C502 - Commission for using investment account, rate – 0.4%

C0 - NO Commission

Formula for calculation:

Commission = rate\_commision \* balance at the last day of month

There are three type of penalties:

Penalties for late paying interest for credit account – 1% of amount of interest

Formula for calculation:

Penalties = rate\_penalt \* (amount of interest for paying)

Penalties for late returning amount of credit for credit account – 2% of amount for returning of credit

Penalties = rate\_penalt \* (amount of amount for returning of credit) Managers of VSB Bank need total information about amount charge, commission and penalties for all clients of VSB Bank.

**VSB Bank verified SECOND task: CLIENT APPLICATION WITH 5 PROCESS:**

Search for client

Update info client

Insert new client

Delete client

Report for client

Search for client – has to be: put client\_id and search for information first name, last name, address, telephone, e-mail, ssn, see all information from Database

Update info client – has to be: put id\_client and update first name, last name, address, telephone, e-mail, ssn for existing client in Database

Insert new client - has to be: insert / create new client: id\_client, first name, last name, address, telephone, e-mail, ssn in Database

Delete client - has to be: delete of existing client: id\_client, first name, last name, address, telephone, e-mail, ssn from Database

Report for client – has to be: put id\_client and create/see report with information: id client, first name, last name, number account, type account, description about account.

**VSB Bank verified THIRD task: COMMISSION APPLICATION WITH 7 PROCESS:**

SEARCHING DISCRIPTION OF COMMISSION by using selection type\_commission

SEARCHING ALL COMMISSION OF CLIENTS by using type\_commission

SEARCHING COMMISSION OF CLIENT by using id\_client

ADD NEW TYPE COMMISSION

UPDATE TYPE COMMISSION

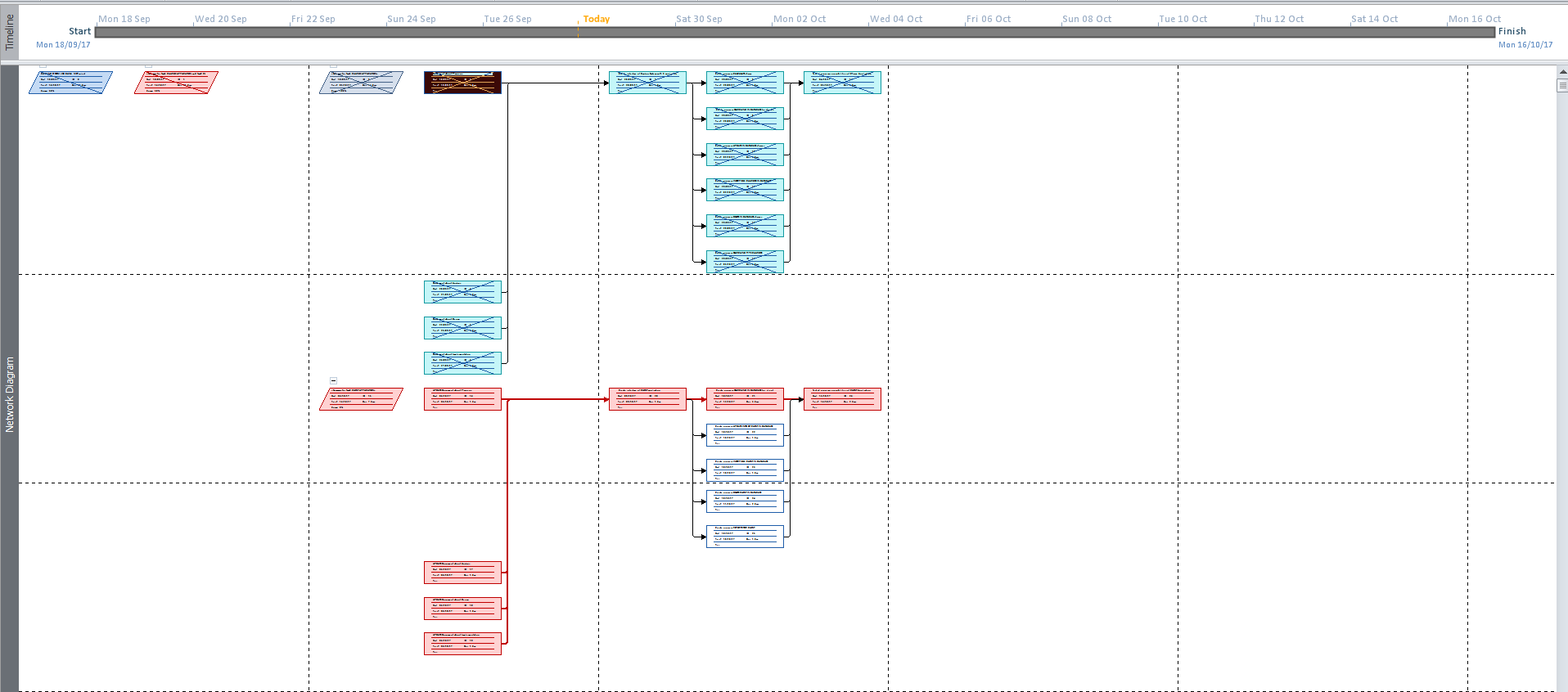
CALCULATE NEW COMMISSION

UPDATE EXISTING COMMISSION

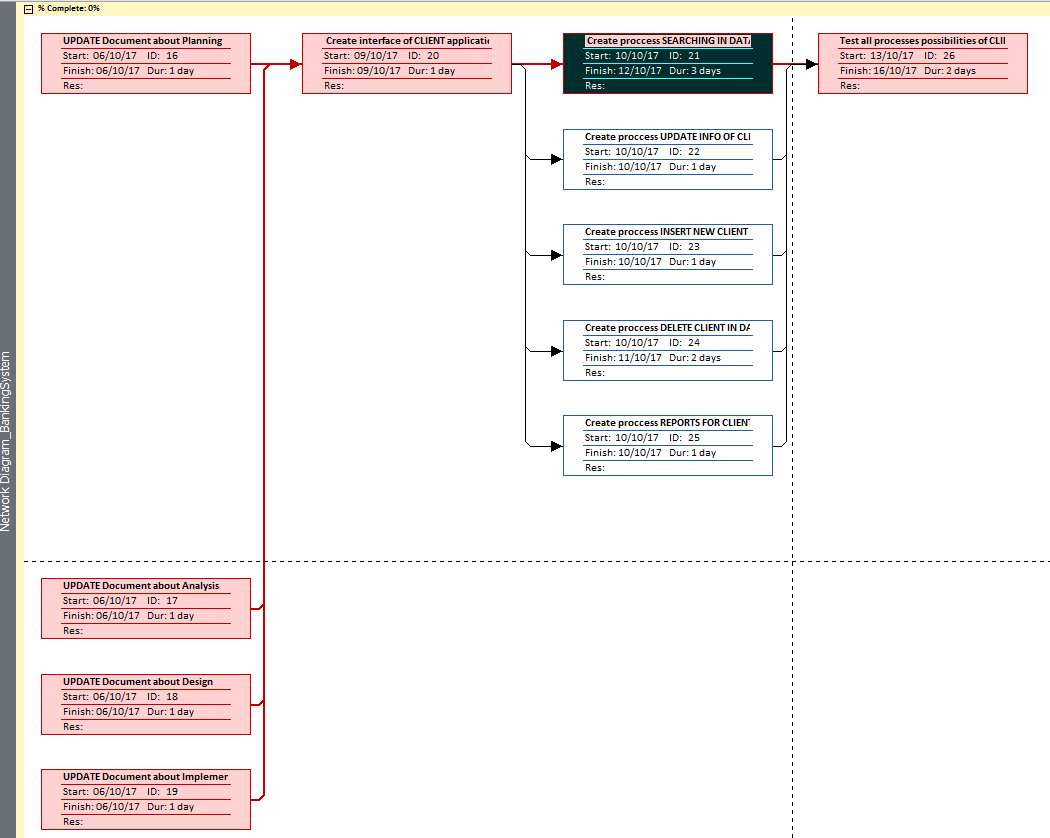
* + 1. **Project scheduling time (MS Project, tasks and duration, Gantt chart: tasks duration, PERT(evaluation permit, Diagram chart, tasks patterns)**

**NETWORK DIAGRAM**

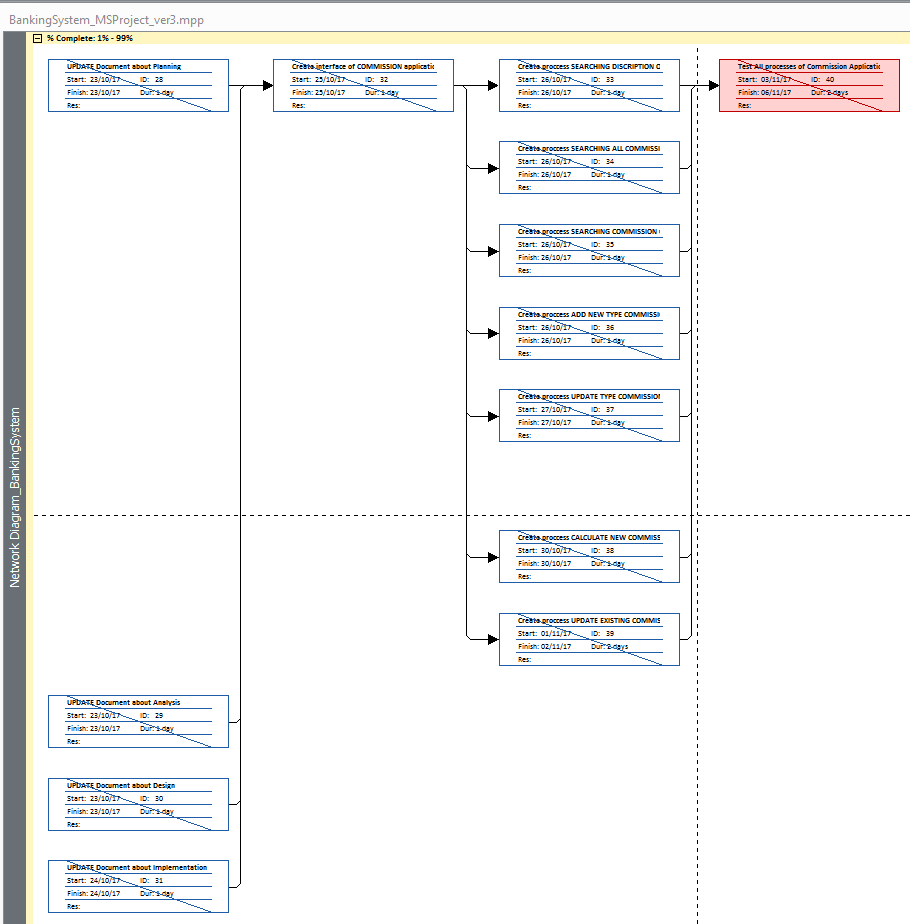
**First task, second task are completed.**



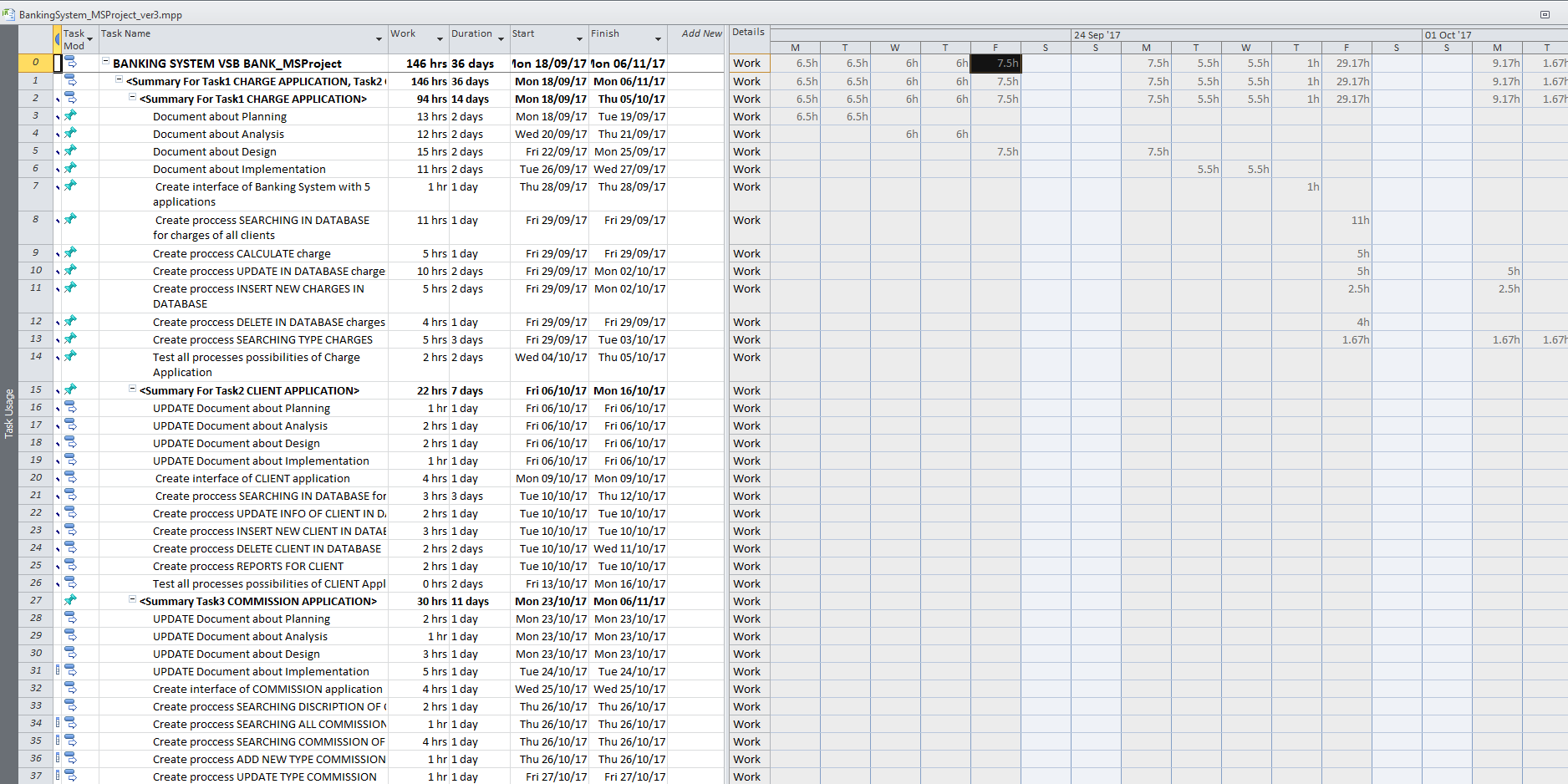
**Network Diagram for the second completed task**



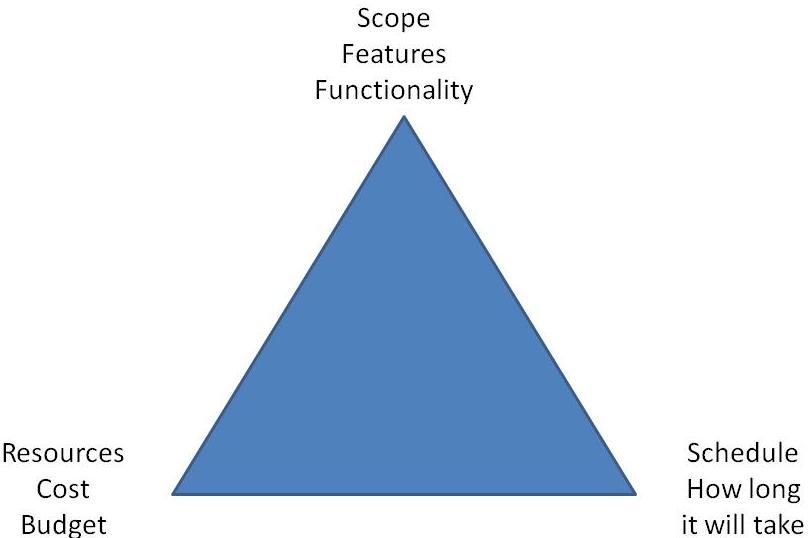
**Network Diagram for the third task**



**Task diagram**



* + 1. **Project Triangle (time, scope, costs)**



Time/schedule is 36 days tougher or 146 hours.

Scope – create/automatized Banking System for small VSB Bank for 20 employees

Cost – 36 490.00$

* 1. **Project Risk/Probability (Matrix: risk and probability)**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  |  | Impact | | | | |
|  |  | Trivial | Minor | Moderate | Major | Extreme |
| Probability | Rare |  |  |  | Team is unfamiliar with development framework(Consequences:  Tasks overrun  More time will be spent at the initial stage of the project) | Server is down for usage((Consequences:  Wed application can’t be hosted and unavailable until issue is resolved) |
| Unlikely |  |  | End user are unfamiliar with the usage application  (Consequences:  Productivity affected  Banking System can’t be fully utilized) |  |  |
| Moderate |  |  |  | Hardware components are not procured in time for development and testing  (Consequences:  Iteration tasks affected  A delay in project delivery) |  |
| Likely |  |  |  |  |  |
| Very Likely |  |  |  | Change in requests from client (Consequences: Project scope is affected  A delay in project delivery) |  |

1. **System Analysis**
   1. **Client questionnaire (design questions: how many people, very often, mobile application or web, requirements of application) (word)**

Q1 (System Analyst): What is the different rate of different type of charge, commission and penalties of Banking System

An1(End user): the rate depends on type of charge, type of commission and type of penalties

There are three type of charge:

CIP4000 - Charge of personal insurance for credit account, rate – 0.5%, periodicity for calculation - per year

CIPL4000 - Charge of pledge for credit account, rate – 5%, periodicity for calculation - per year

CIC4000 - Charge of student for credit account, rate – 0.3%, periodicity for calculation - per year

CIC0 – NO CHARGE

Formula for calculation:

Charge = rate\_charge \* limit

There are two type of commission:

C500 - Commission for using cheking account, rate – 0.3%

C501 - Commission for using credit account, rate – 0.5%

C502 - Commission for using investment account, rate – 0.4%

C0 - NO Commission

Formula for calculation:

Commission = rate\_commision \* balance at the last day of month

There are three type of penalties:

Penalties for late paying interest for credit account – 1% of amount of interest

Formula for calculation:

Penalties = rate\_penalt \* (amount of interest for paying for credit account)

Penalties for late returning amount of credit for credit account – 2% of amount for returning of credit

Penalties = rate\_penalt \* (amount of amount for returning of credit)

Q2 (System Analyst): What is limit?

An2(End user) Limit usually it is for credit account, limit means what is credit/bank’s amount of money which client can use. On limit depends rate of insurance for person and for pledge.

Limit is made of bank and can change.

Q3 (System Analyst): What is balance?

An3(End user): Balance is amount of money at the end of the day on the account

Q4: What is mean amount of interest for paying for credit account?

An4: Interest is amount of money what is pay client for using credit of bank.

Formula for calculation is:

Interest per one day = Rate of interest \* Balance at the end of the day

Q5: What is information employee need in Report about client?

Information for identification client(id, first and last name), how many account client has, of what is type this account, description of account, balance and limit of account



Q6: Is account number is unique for every client, and commission type depends on type of account?

Yes, for every client account number are different, and for every type account exists different type commission.

* 1. **Key questions who, where, why, which, how, when system will be used (word)**

Who: Banking System will use client department and finance department of VSB Bank – 20 employees

Where: in VSB Bank

Why: for calculation, update, save, insert, delete, search charge, commission and penalties

How: on the computer using Banking Application

When: Every day during the day

* 1. **Requirement modeling (word)**
     1. **Input (six entities/tables)**

**For Task 1 we need Client\_entity, Account\_type, Charge\_type\_insurance and Account\_entity**

**For Task 2 we need Client\_entity, Account\_type and Account\_entity**

**For Task 3 we need Client\_entity, Account\_type, Commission\_type and Account\_entity**

Client\_entity:

Id\_client – technical id of clients

Fname – first name of clients

Lname – last name of clients

Address – address of clients

Tel – telephone of clients

E-mail – e-mail of clients

SSN – social insurance number of clients

|  |
| --- |
| Charge\_type\_insurance:  Type\_charge - Type of Charge (PK)  Desc\_charge - Description of Charge  Rate\_charge - Rate of Charge  Period\_calc - Periodicity for calculation in days  Day\_pay - Due to pay |

|  |
| --- |
| Commission\_type  Type\_commission - Type of Commission (PK)  Desc\_commission - Description of Commission  Rate\_commission - Rate of Commission  Period\_calc\_commission - Periodicity for calculation  Day\_pay\_commission - Due to pay |

|  |
| --- |
| Penalties\_type  Type\_penalties - Type of Penalties (PK)  Desc\_penalties - Description of penalties  Rate\_penalties - Rate of penalties  Period\_calc\_penalties - Periodicity for calculation  Day\_pay\_penalties - Due to pay |

|  |
| --- |
| Account\_type  Type\_account - Type of Account  Desc\_account - Description of account  Number\_account – Number account PK |

Account\_entity:

Id\_client - Id client (PK)

Number\_account - Number of account (PK – unique for each clients)

Type of account(FK – PK in table account\_type)

Balance - Balance

Limit - Limit

Type\_penalties - Type of Penalties(FK, PK in Penalties\_type table)

Amount\_penalties - Amount of penalties

Type\_charge - Type of Charge(FK – PK in table charge\_type)

Amount\_charge - Amount of Charge

Type of Commission(FK; PK in table commission\_type)

Amount of Commission

* + 1. **Output**

Amount of charges, commissions, penalties for every account, for every clients (results of calculation). New records in Database for charges, clients, commission and penalties.

Sum of charges, commissions, penalties for all clients of the bank VSB (after calculation).

Form ReportClient with information about for identification client(id, first and last name), how many account client has, of what is type this account, description of account, balance and limit of account.

* + 1. **Process (method formula for charge and for commission)**

There are three type of charge:

CIP4000 - Charge of personal insurance for credit account, rate – 0.5%, periodicity for calculation - per year

CIPL4000 - Charge of pledge for credit account, rate – 5%, periodicity for calculation - per year

CIC4000 - Charge of student for credit account, rate – 0.3%, periodicity for calculation - per year

CIC0 – NO CHARGE

Formula for calculation:

Charge = rate\_charge \* limit

There are two type of commission:

Commission for using cheking account, rate – 0.3%, periodicity for calculation - per month

Commission for using credit account, rate – 0.5%, periodicity for calculation - per month

Formula for calculation:

Commission = rate\_commision \* balance at the last day of month

There are three type of penalties:

Penalties for late paying interest for credit account – 1% of amount of interest

Formula for calculation:

Penalties = rate\_penalt \* (amount of interest for paying)

Penalties for late returning amount of credit for credit account – 2% of amount for returning of credit

Penalties = rate\_penalt \* (amount of amount for returning of credit)

There are four type of commission:

C500 - Commission for using cheking account, rate – 0.3%

C501 - Commission for using credit account, rate – 0.5%

C502 - Commission for using investment account, rate – 0.4%

C0 - NO Commission

Formula for calculation:

Commission = rate\_commision \* balance at the last day of month

* + 1. **Performance (web site need or no, scalability of users)**

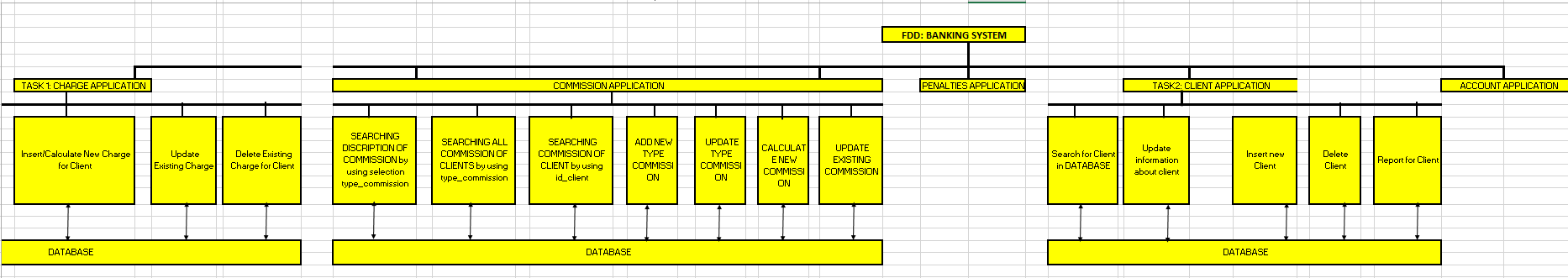
Website don’t need, Banking System for 20 users, for employees of the VSB Bank not for clients

* + 1. **Security issue (for account)**

Banking System is security

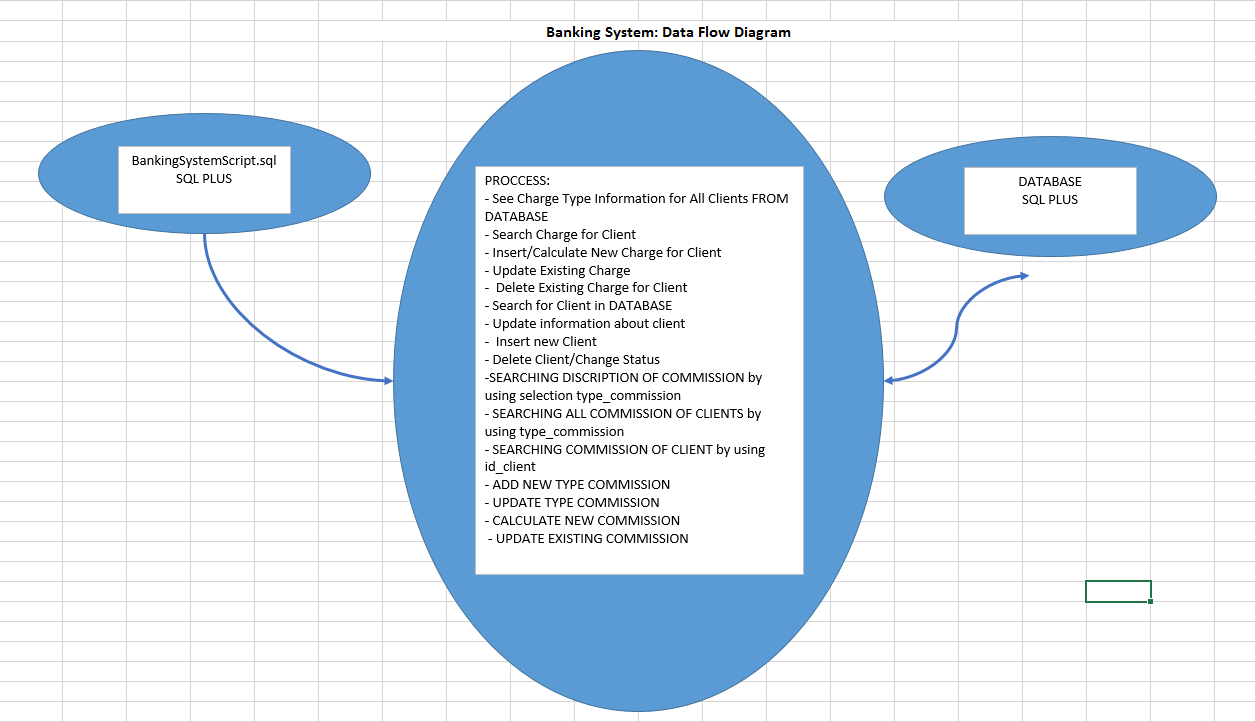
* 1. **Data and Process Modeling** 
     1. **Function Decomposition Diagram (capture modules)**

FDD is a graphical representation of business functions that starts with major functions, and then breaks them down into several levels of detail. An FDD of Banking System provide a checklist of user tasks that included in the interface design.

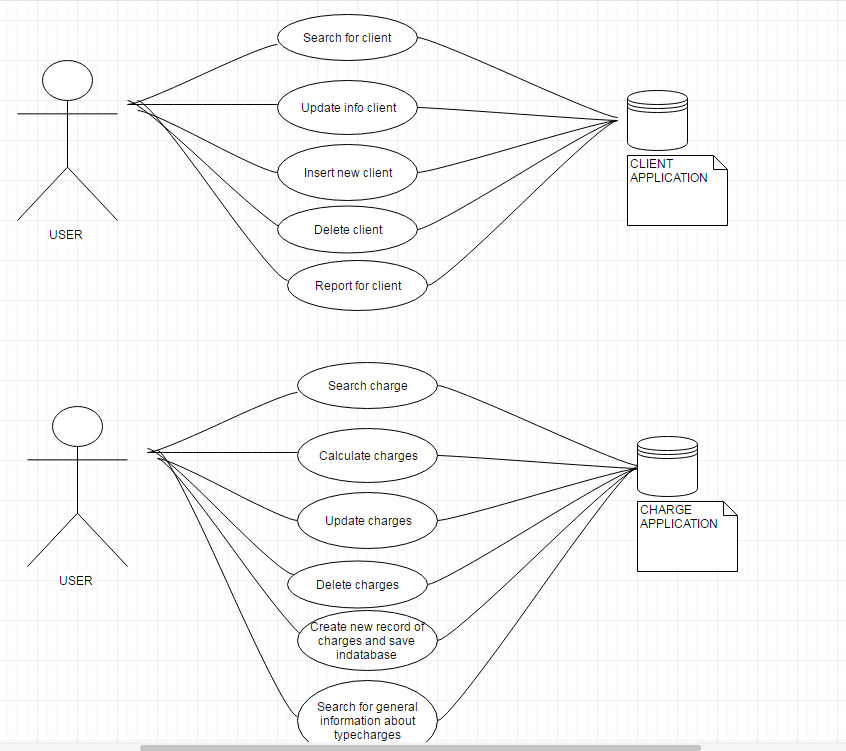


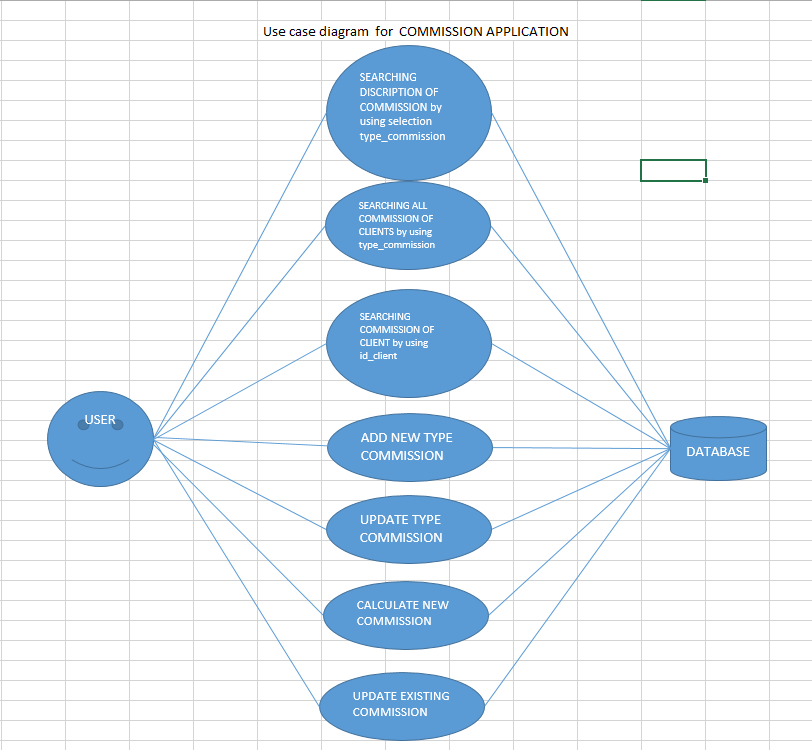
* + 1. **Data Flow Diagram DFD order 0 (1,2)(Data store – Process () – Data store) (capture flow of information)**

A data flow diagram (DFD) illustrates how data is processed by a system in terms of inputs and outputs. As its name indicates its focus is on the flow of information, where data comes from, where it goes and how it gets stored.

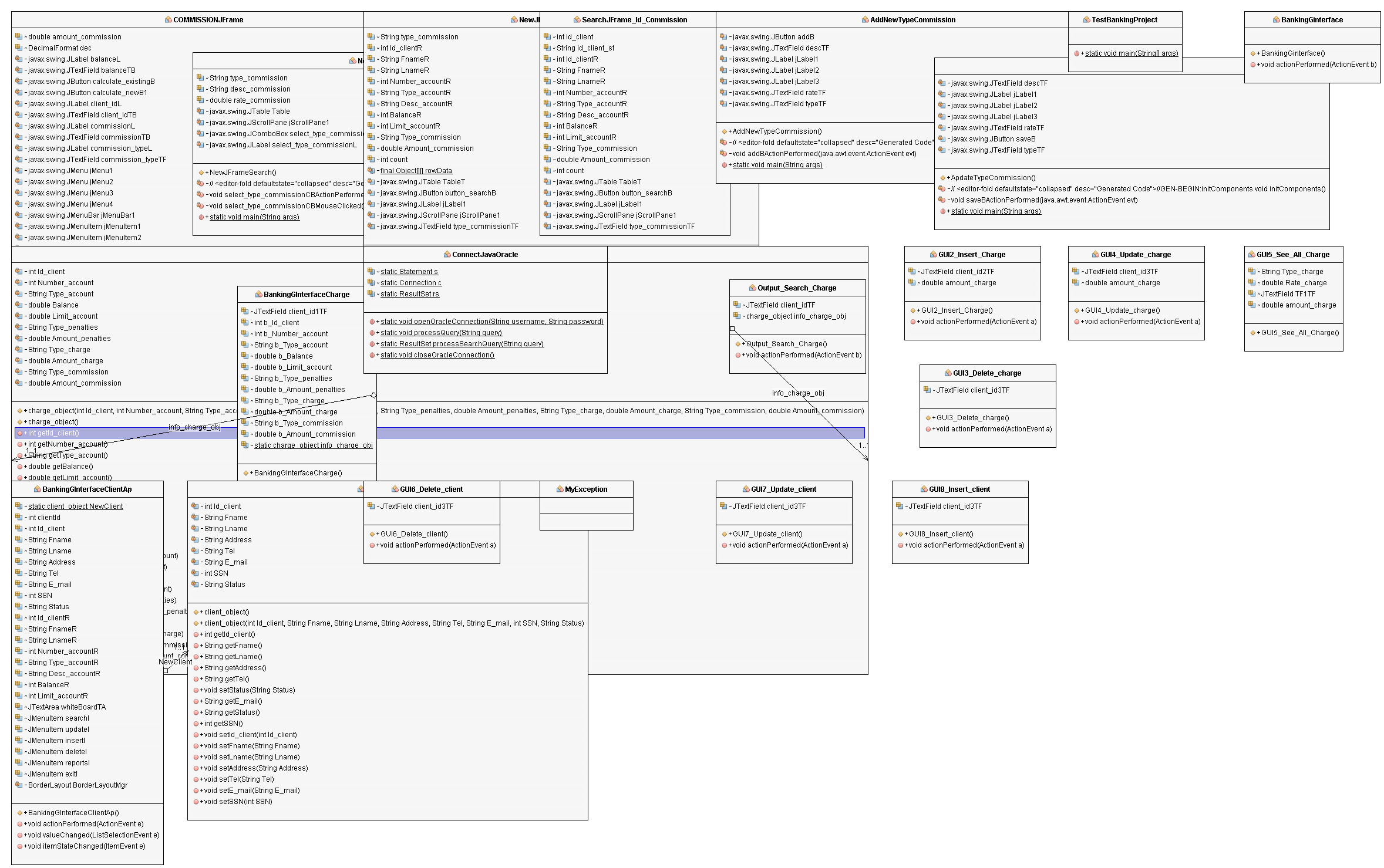


* 1. **UML diagram (Dia)**
     1. Use case diagram



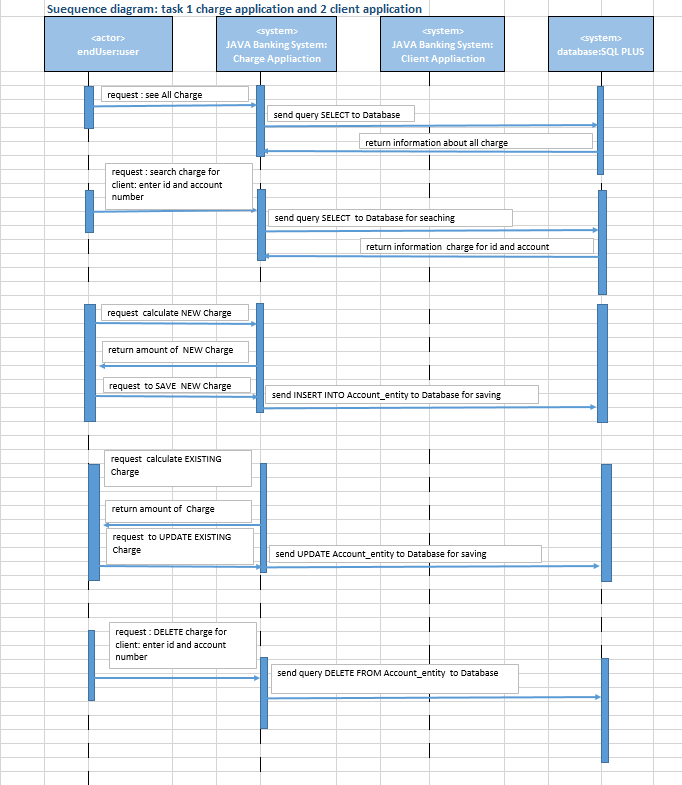


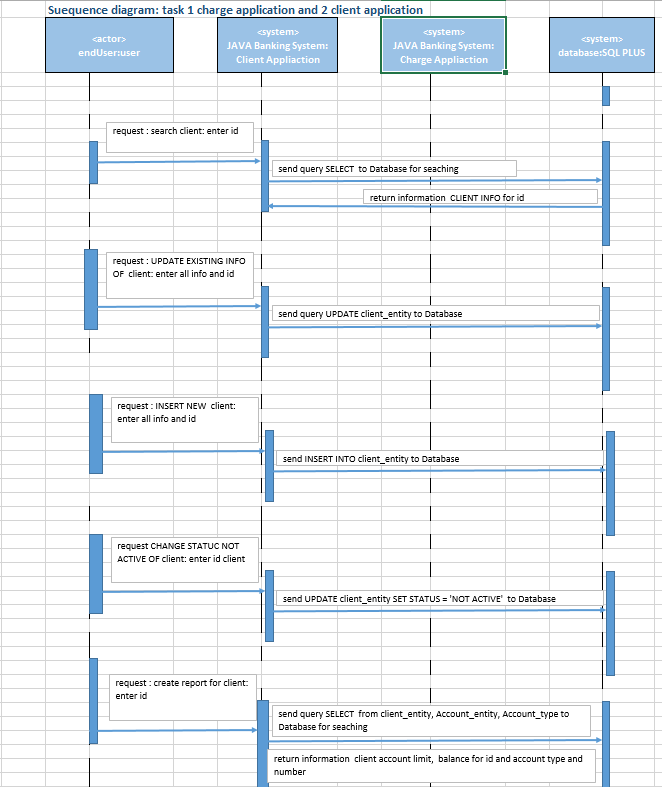
* + 1. **Class diagram**

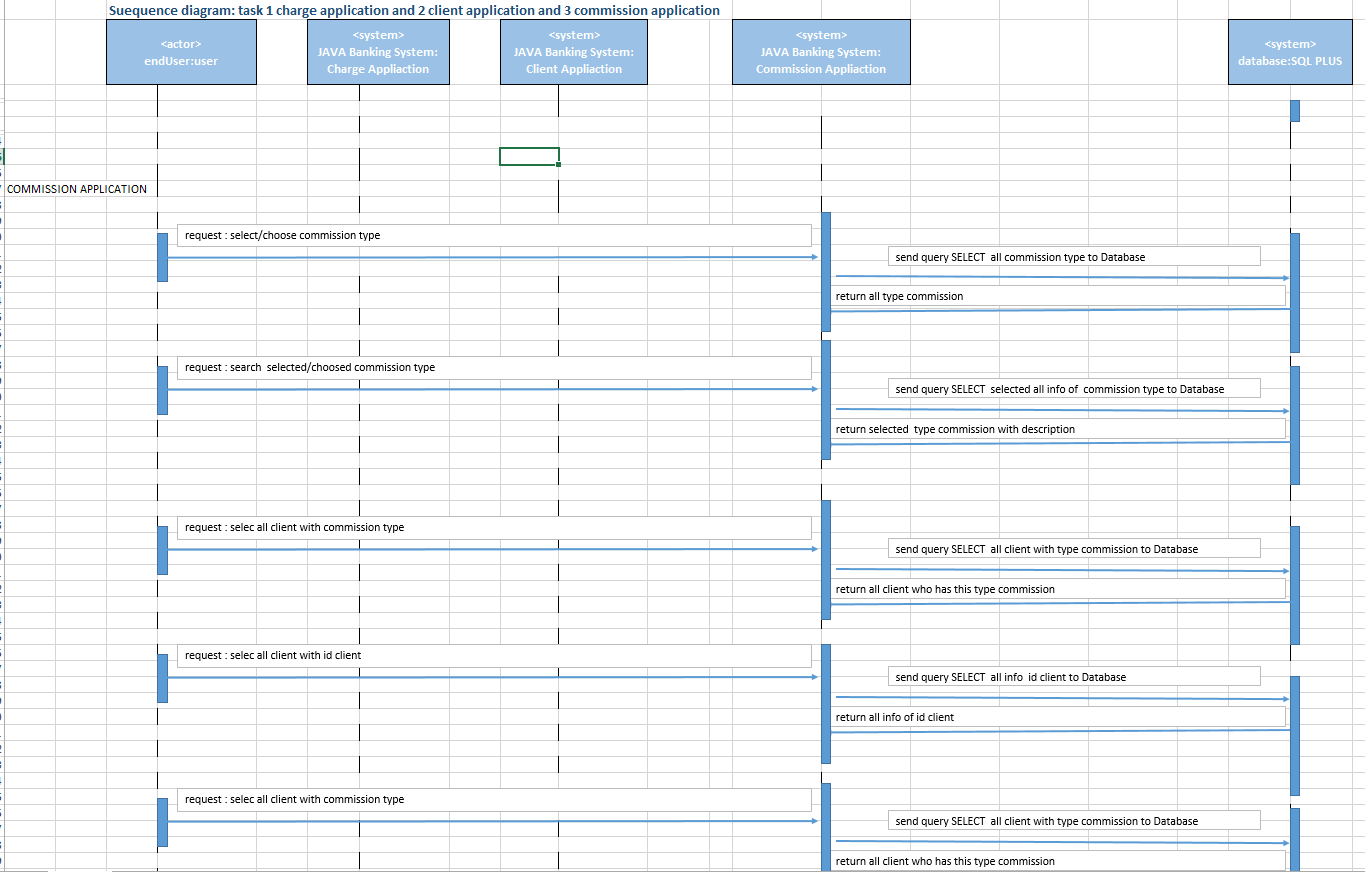


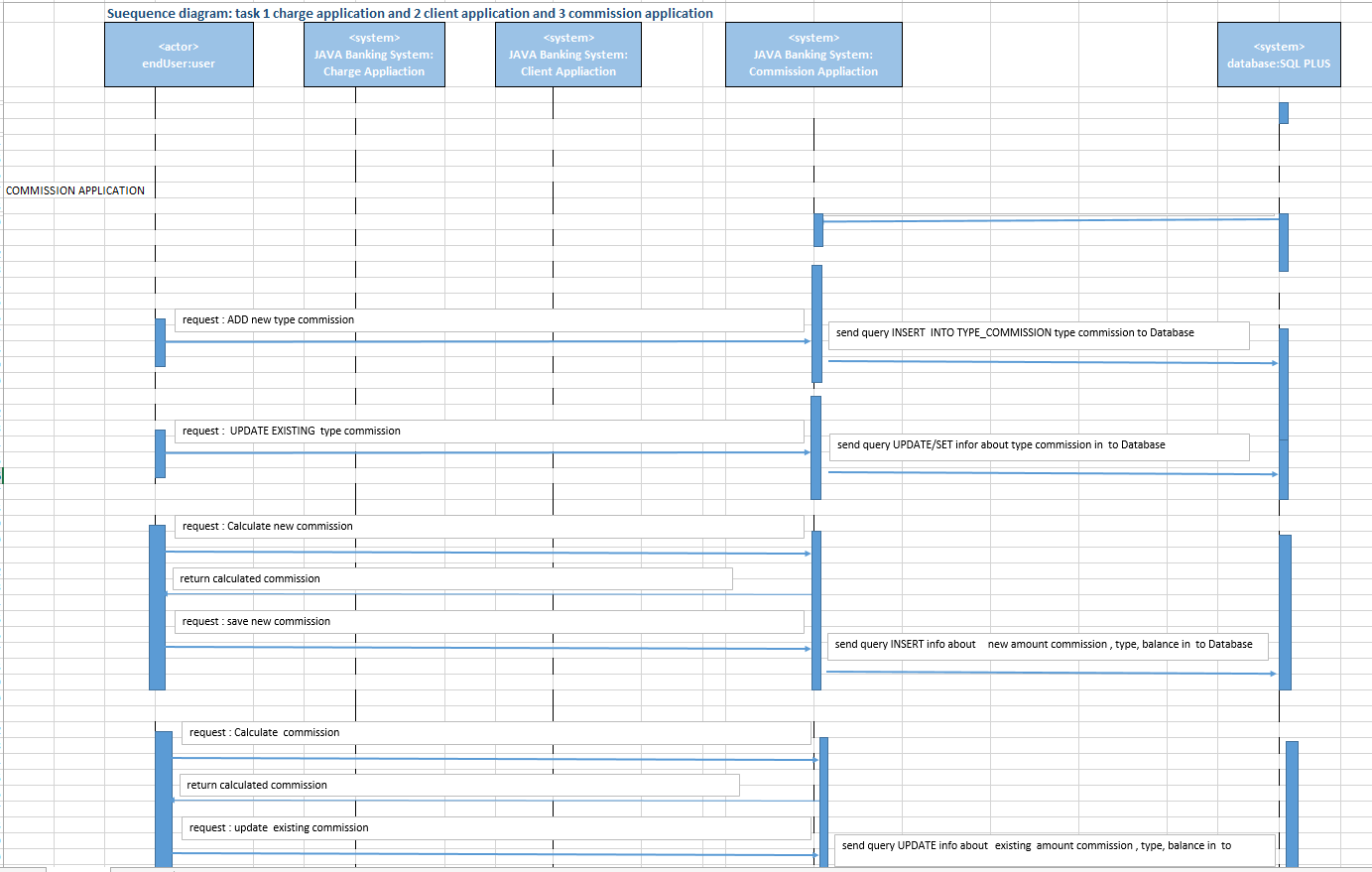
* + 1. **Sequence diagram (timing of request)**

A sequence diagram is an interaction diagram that shows how objects operate with one another and in what order. It is a construct of a message sequence chart. A sequence diagram shows object interactions arranged in time sequence.







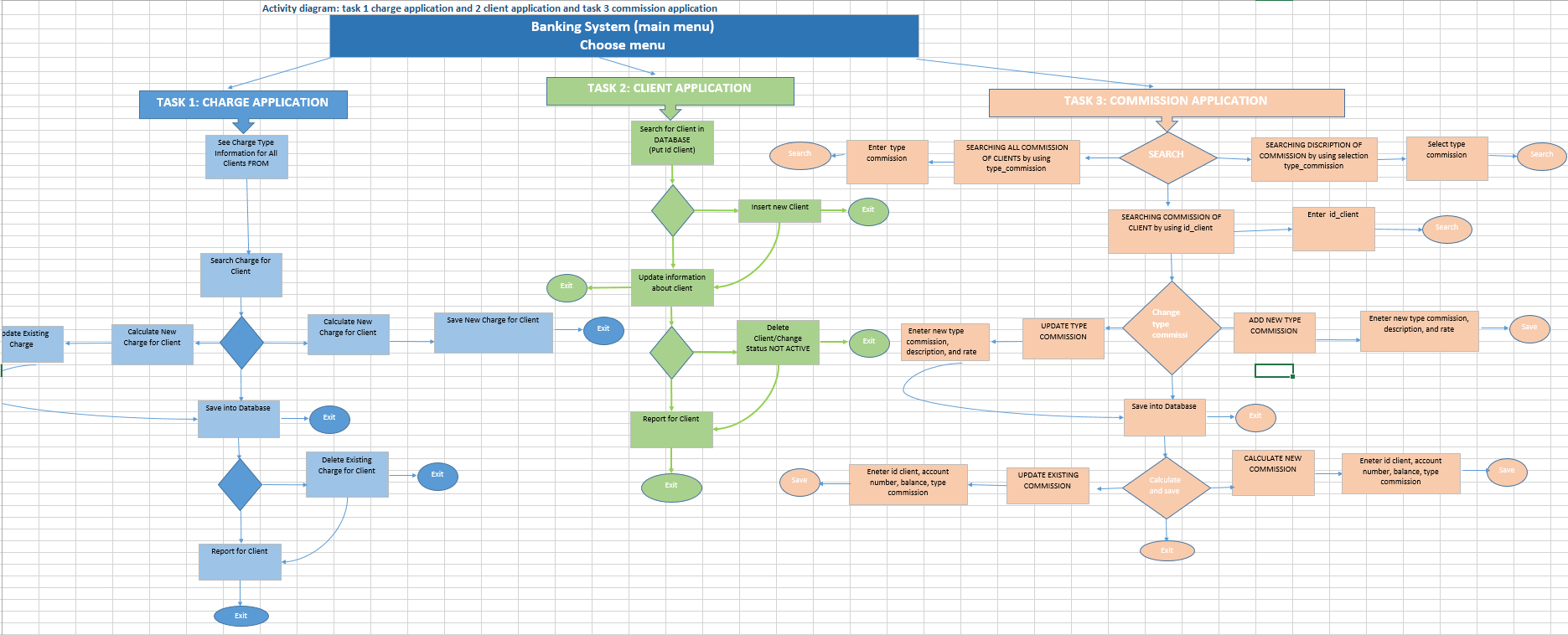


* + 1. **Transition diagram**

 Transition Diagram illustrates a process where an item passes between a number of stable states.



* + 1. **Activity Diagram**



* 1. **Data Dictionary (candidate for fields for feeling design, type, name, size of field (Tables)**

Account\_entity:

|  |  |  |
| --- | --- | --- |
| Name | Type |  |
| Id\_client | NUMBER(9) | Primary Key//composited |
| Number\_account | NUMBER(9) | Primary Key//composite |
| Type\_account | VARCHAR2(30) | Foreign Key // PK in table account\_type |
| Balance | NUMBER(9) |  |
| Limit | NUMBER(9) |  |
| Type\_penalties | VARCHAR2(30) | Foreign Key // PK in Penalties\_type table |
| Amount\_penalties | NUMBER(9) |  |
| Type\_charge | VARCHAR2(30) | Foreign Key // PK in table charge\_type |
| Amount\_charge | NUMBER(9) |  |
| Type\_Commission | VARCHAR2(30) | Foreign Key // PK in table commission\_type |
| Amount\_Commission | NUMBER(9) |  |

client\_entity:

|  |  |  |
| --- | --- | --- |
| Name | Type |  |
| Id\_client | NUMBER(9) | Primary Key |
| Fname | VARCHAR2(30) |  |
| Lname | VARCHAR2(30) |  |
| Address | VARCHAR2(30) |  |
| Tel | VARCHAR2(30) |  |
| E-mail | VARCHAR2(30) |  |
| SSN | NUMBER(9) |  |

Account\_type:

|  |  |  |
| --- | --- | --- |
| Name | Type |  |
| Type\_account | VARCHAR2(30) |  |
| Desc\_account | VARCHAR2(30) |  |
| Number\_account | NUMBER(9) | Primary Key |

Charge\_type\_insurance:

|  |  |  |
| --- | --- | --- |
| Name | Type |  |
| Type\_charge | VARCHAR2(30) | Primary Key |
| Desc\_charge | VARCHAR2(90) |  |
| Rate\_charge | FLOAT(9) |  |
| Period\_calc \_charge | VARCHAR2(30) |  |
| Day\_pay\_charge | VARCHAR2(30) |  |

Commission\_type

|  |  |  |
| --- | --- | --- |
| Name | Type |  |
| Type\_commission | VARCHAR2(30) | Primary Key |
| Desc\_commission | VARCHAR2(90) |  |
| Rate\_commission | FLOAT(9) |  |
| Period\_calc\_commission | VARCHAR2(30) |  |
| Day\_pay\_commission | VARCHAR2(30) |  |

Penalties\_type

|  |  |  |
| --- | --- | --- |
| Name | Type |  |
| Type\_penalties | VARCHAR2(30) | Primary Key |
| Desc\_penalties | VARCHAR2(90) |  |
| Rate\_penalties | FLOAT(9) |  |
| Period\_calc\_penalties | VARCHAR2(30) |  |
| Day\_pay\_penalties | VARCHAR2(130) |  |

1. **System Design (SQL developer, Access)** 
   1. **Development strategies**

**Desktop application**

**Mobile application**

**Web application**

**Cloud based application**

**SOA (service-based application) based application**

Developing strategies is the essential step between figuring out objectives and making the changes to reach them. Our objective is Banking System for VSB Bank with 20 employees for client and financial department for 20 end users. To reach result Banking System application we need desktop application, application that runs stand-alone in a desktop or laptop computer. Reasons for that are: less expensive to develop, no more updates, it is only for using by 20 people, it has more performance and stability and doesn’t have to rely on the browser to work.

* 1. **Project Prototype(https://balsamiq.com/download/#)**
     1. **User Interface**

**Intuitiveness/user friendly**

The Layout divided into areas that are used consistently for the same purpose: bottom for commands “Calculate”, “Exit”, “Print”, “Insert”, “Update”, “Delete”, Text Box for input, Label for input and output middle of the screen for input/output.

The screen divided so as to provide natural intuitive flow and minimize the user’s movement.

Interface “Banking System” has title “WELCOME TO THE BANKING APPLICATION OF VSB BANK!”. Main Interface has 5 applications and button exit.

Field labels are short and specific, clear and unambiguous: Enter names of input, Names of output.

The interface must be pleasing to the eye.

Reports has a minimum amount of white space.

Different font size used to distinguish between different types of information, san-serif fonts, e.g. Arial, Helvetica, are most readable for computer screens.

Capital letters used for title.

Too many different colors and patterns make for difficult reading and was minimized.

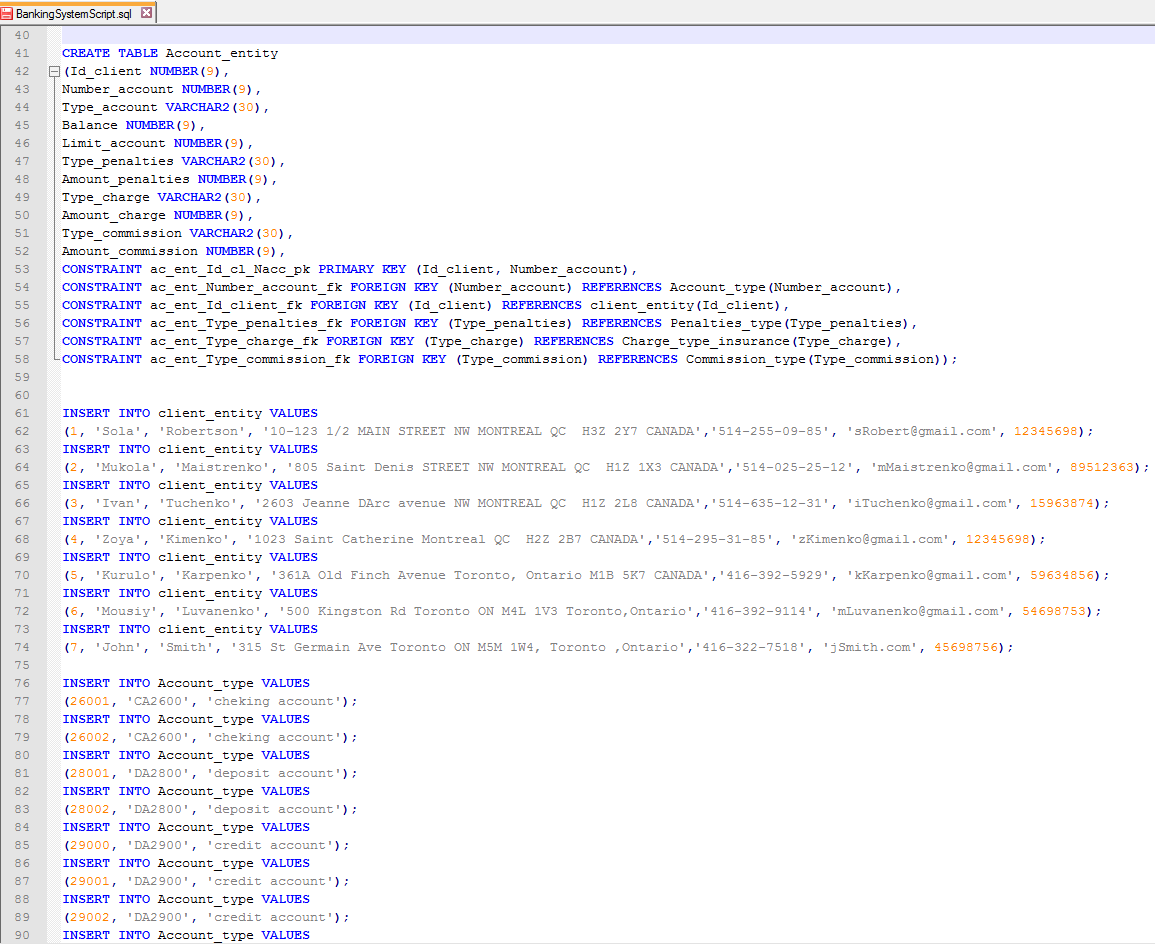
Colors used to differentiate between different types of information, to differentiate bottoms from regular text.

Guidance of Banking System is possible switched on and off.

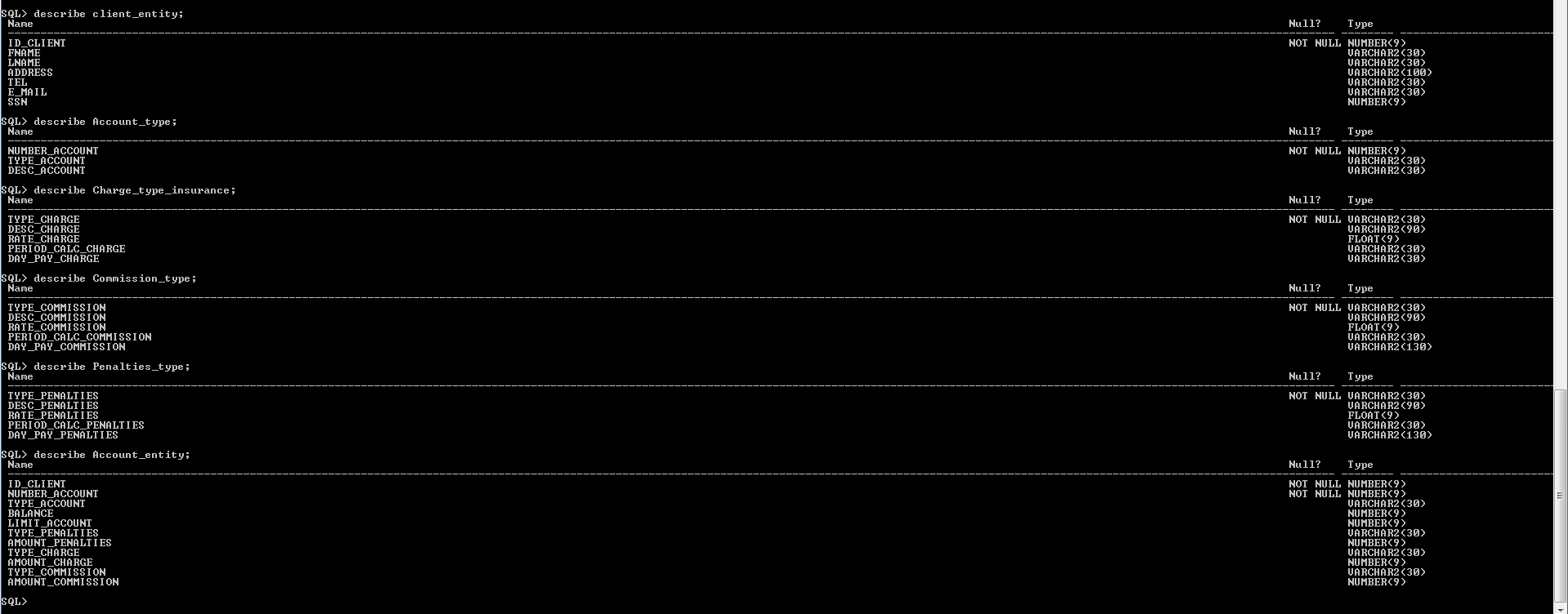
Project Prototype of Banking System:

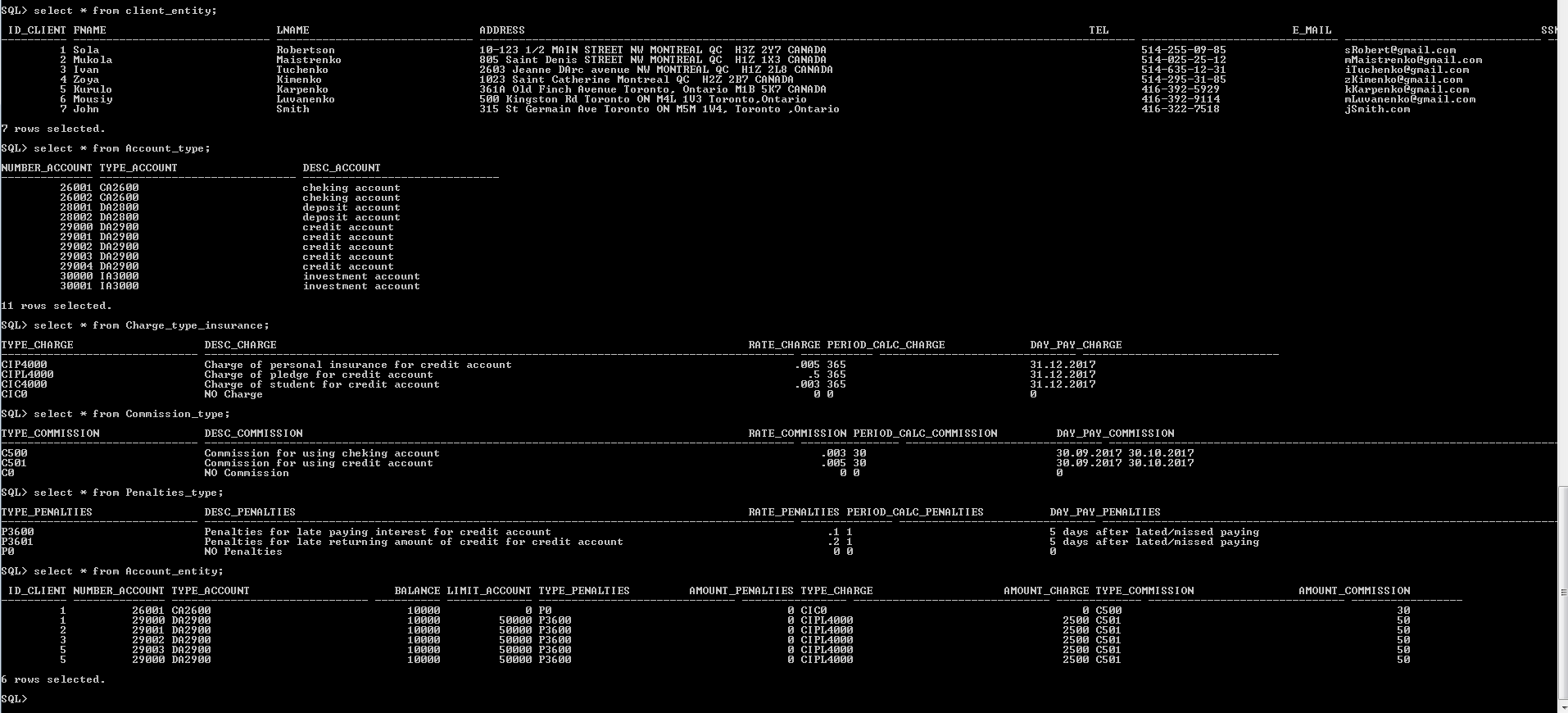
First, we created BankingSystemScript.sql for create six tables in Database:





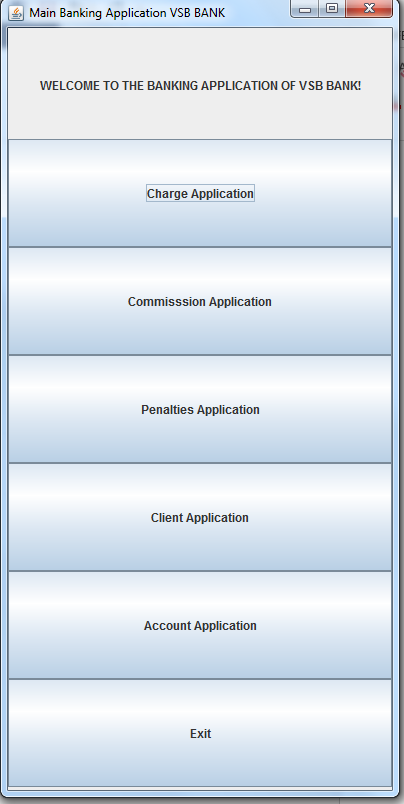
As Result we have 6 tables with data of VSB Bank:

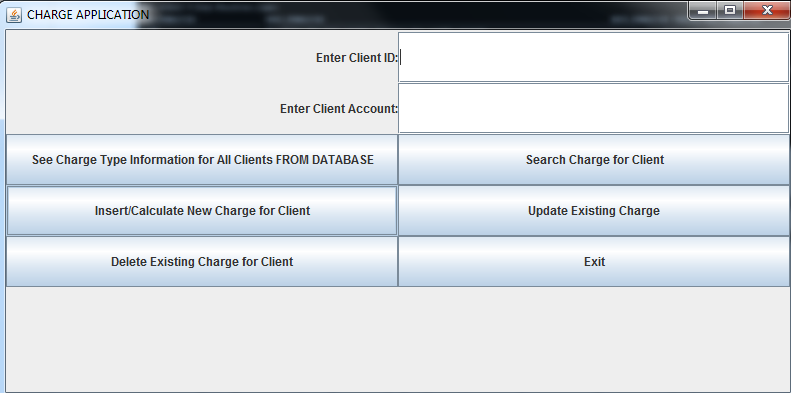




Create public class ConnectJavaOracle with open, close and process, search connection Java and SQL.

**Using button Charge Application, you open Charge application where you can create new charge, calculate, update, delete, search for information.**



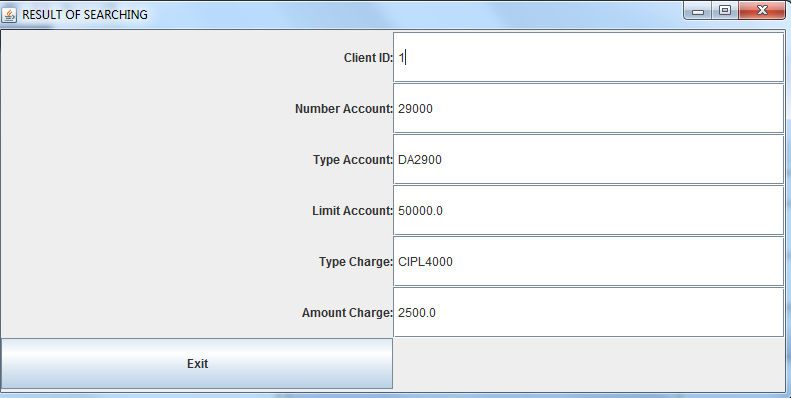


By Clicking on button See Charge Information you can see information about Charge Type what exist in a bank from Database.

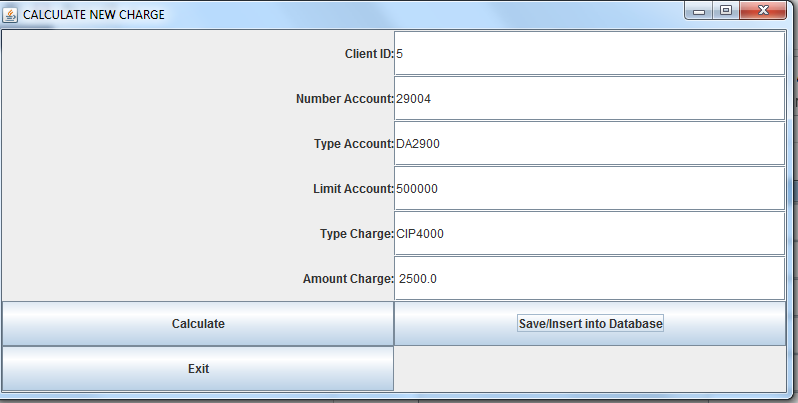


For search charge in Database for every client you need put composite key client\_id and client\_account ad after press button Search charge for client. Result will be:





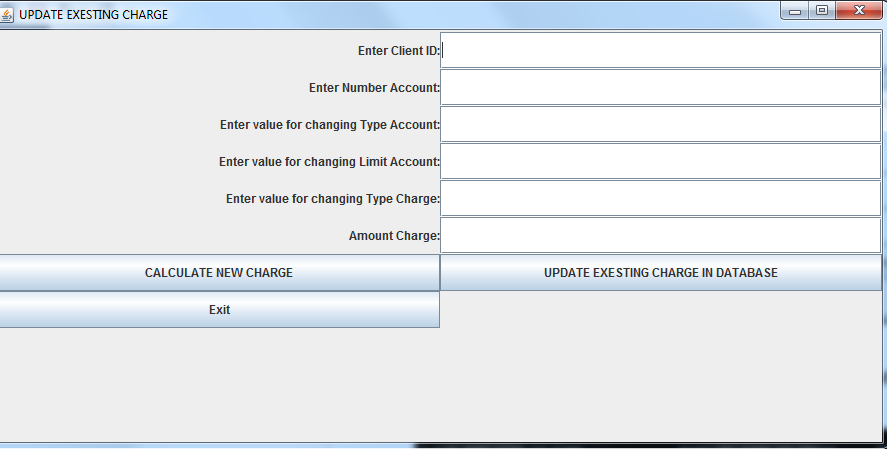
For inserting new charge you have to press button Insert/Calculate new charge. First you can calculate charge and after save in database.



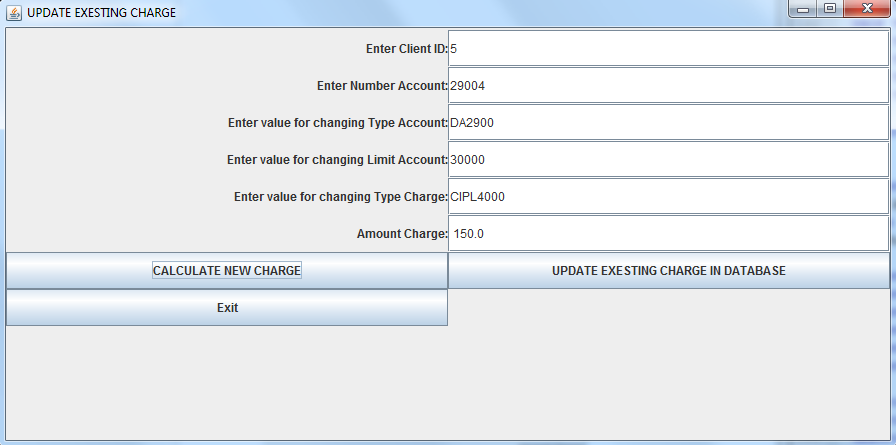
By clicking Save you save calculated charge in database. Result:

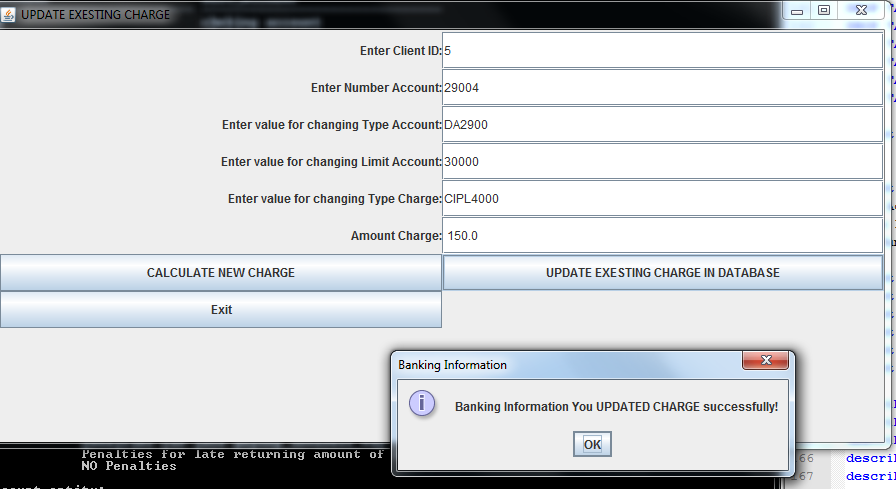


By clicking update you will have interface like

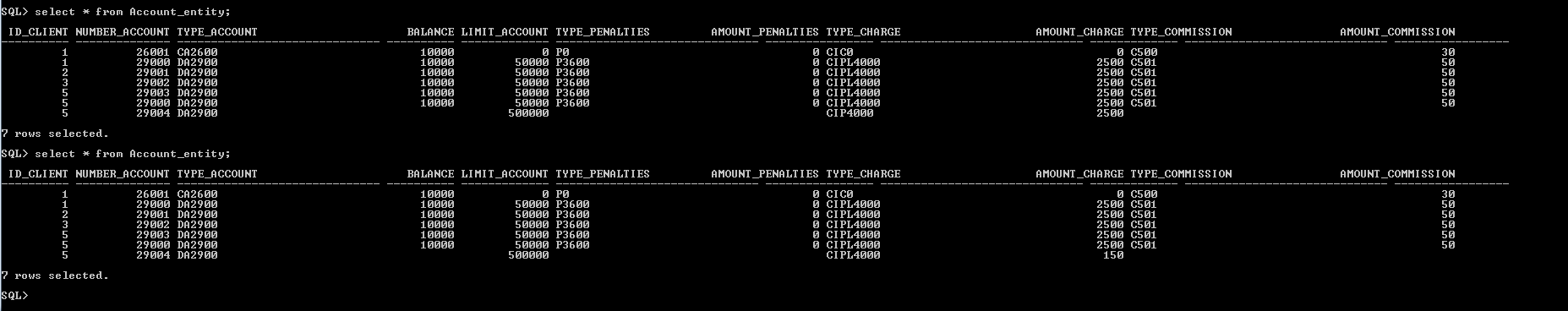


After entering all data you can calculate new charge and by clicking update - save new charge for existing client. Result will be:

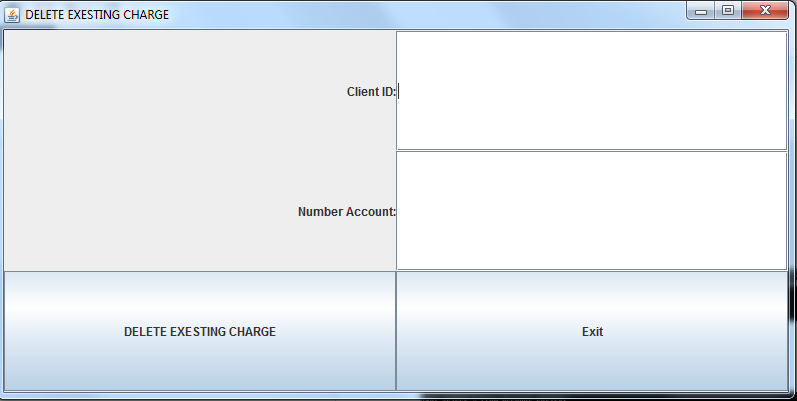




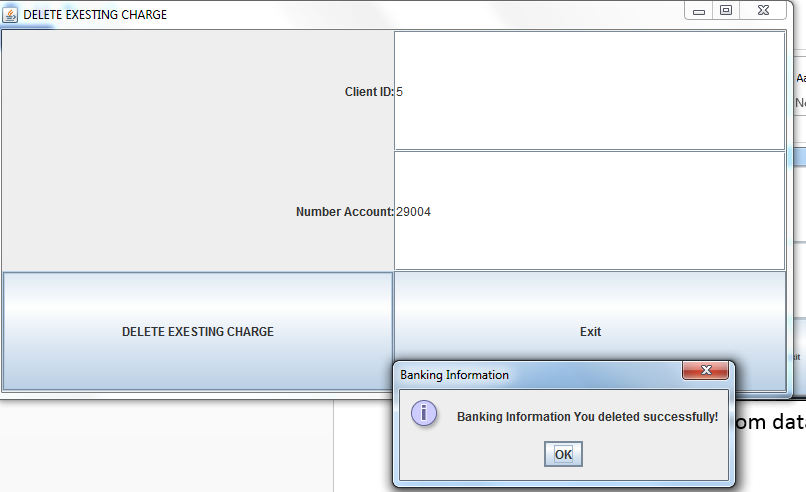
Result in Database



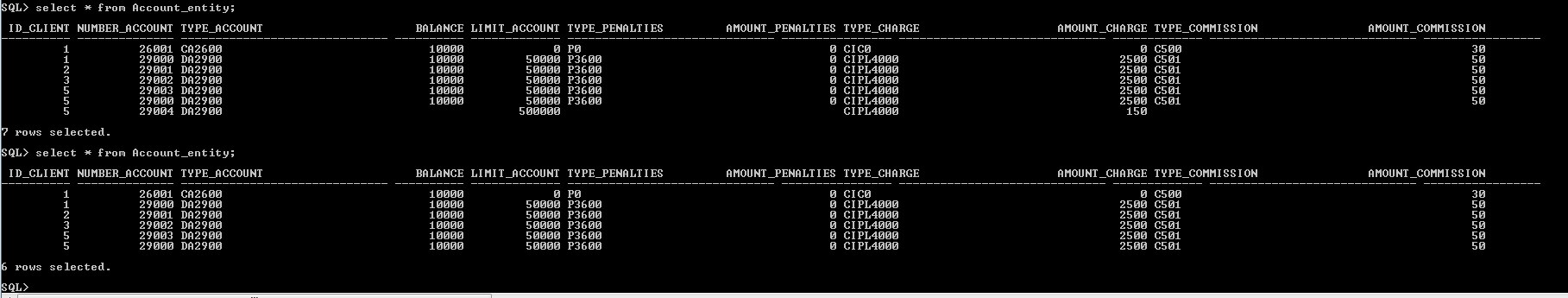
For deleting existing charge you need press button Delete you will have interface:



After press delete you deleted from database, result will be:

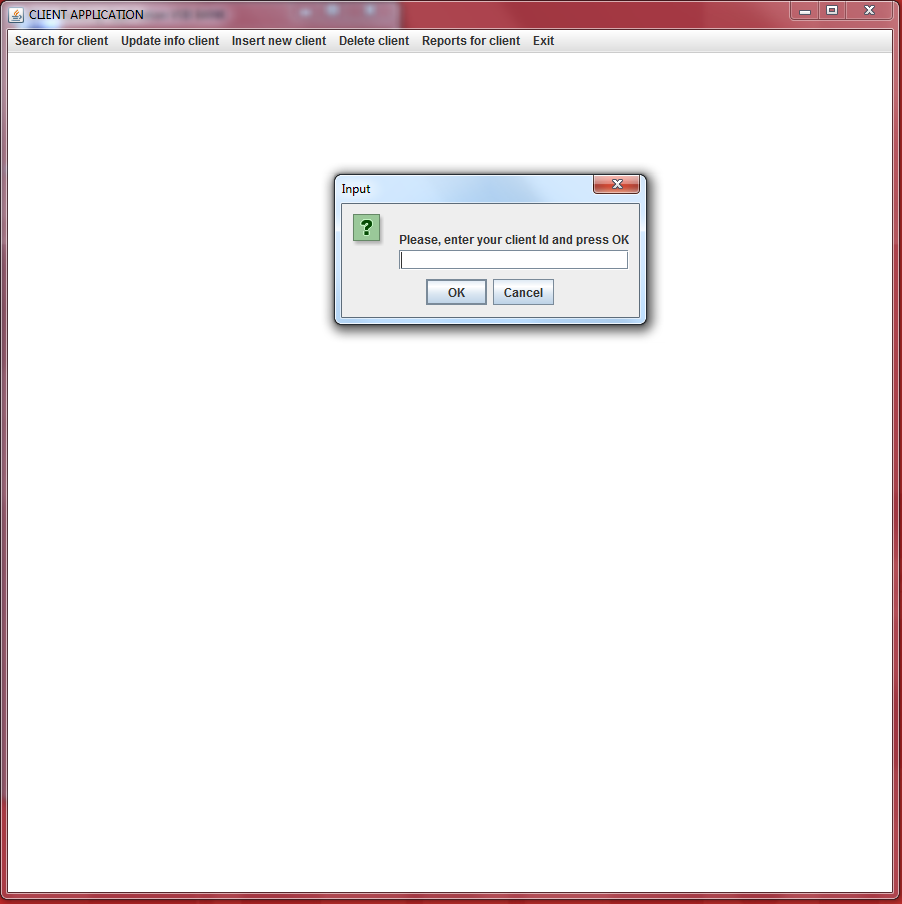


Result in Database

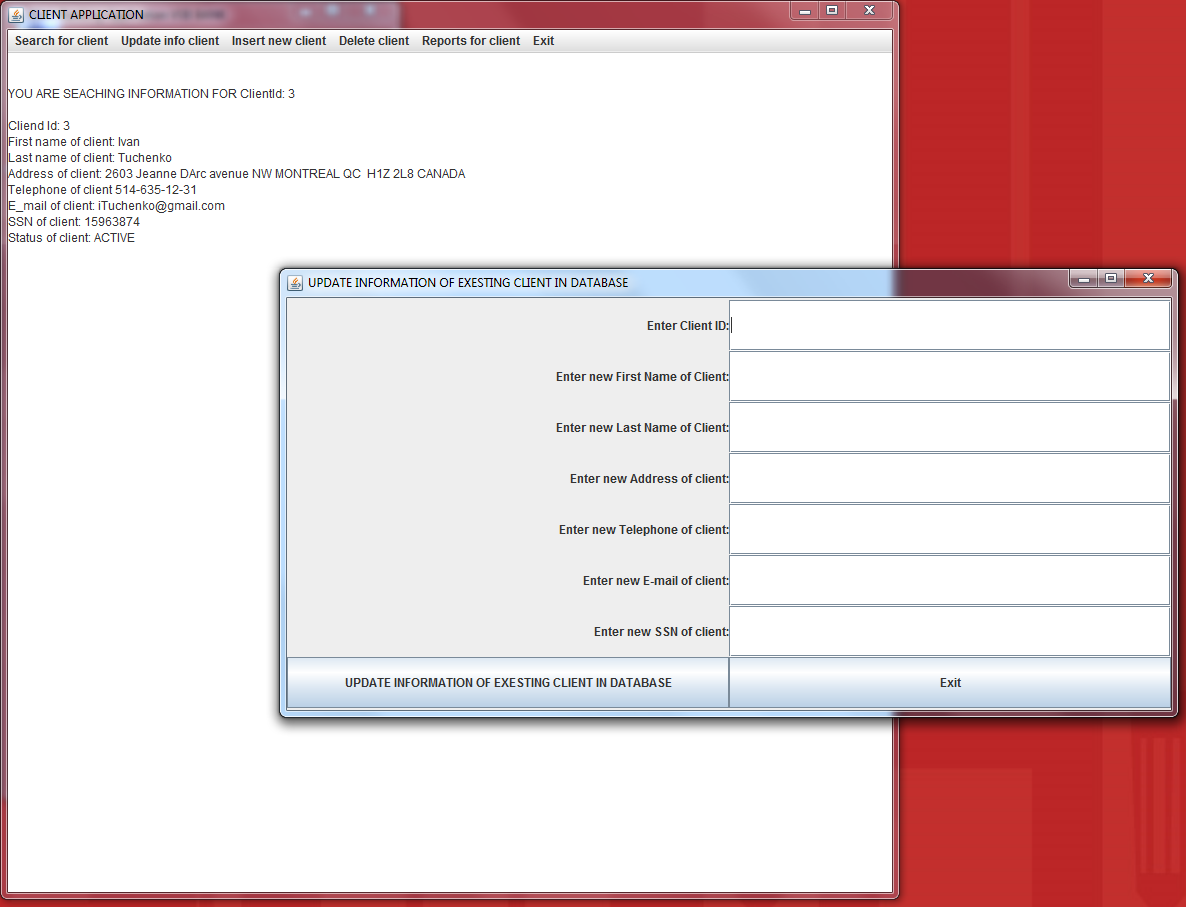


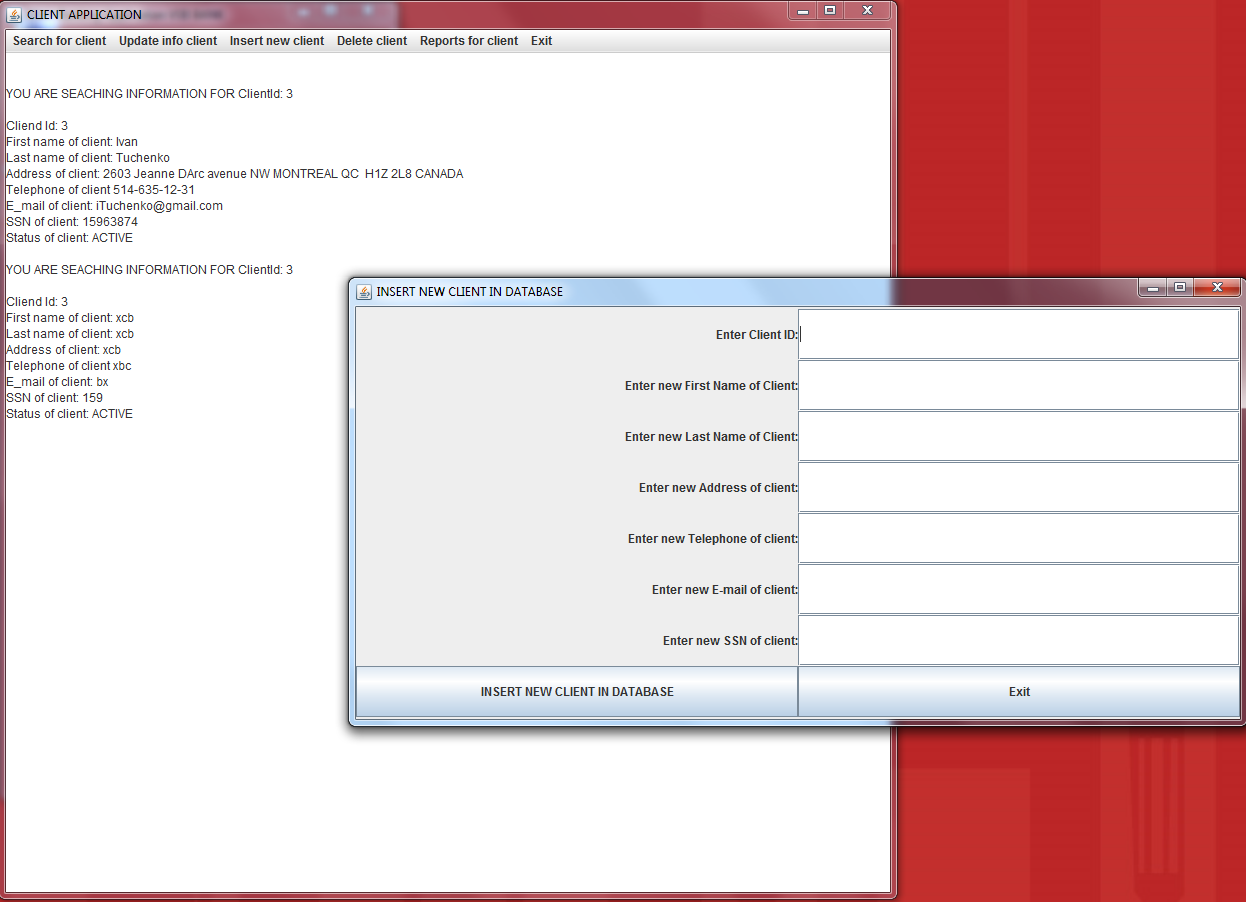
Button Exit, close Banking System.

**Using button Client Application, you open Client application where you can Search for client, Update info client, Insert new client, Delete client, Report for client.**

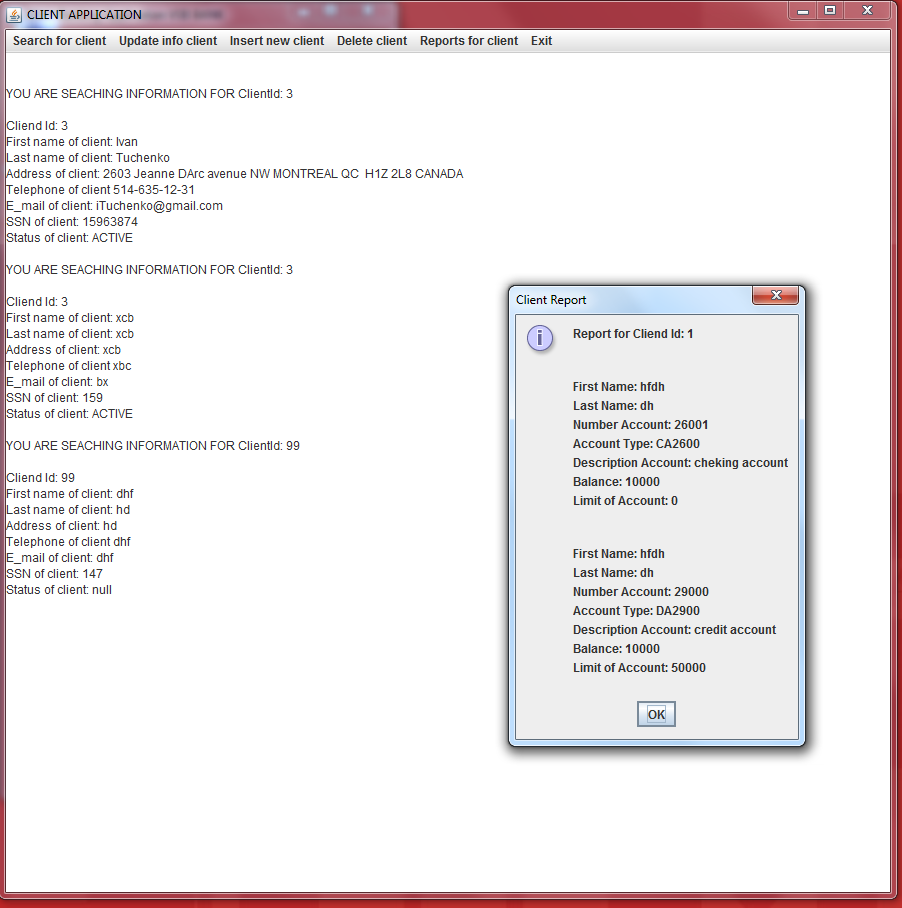


**In client application you have to click** search client information, update information of existing client, insert, delete, create report



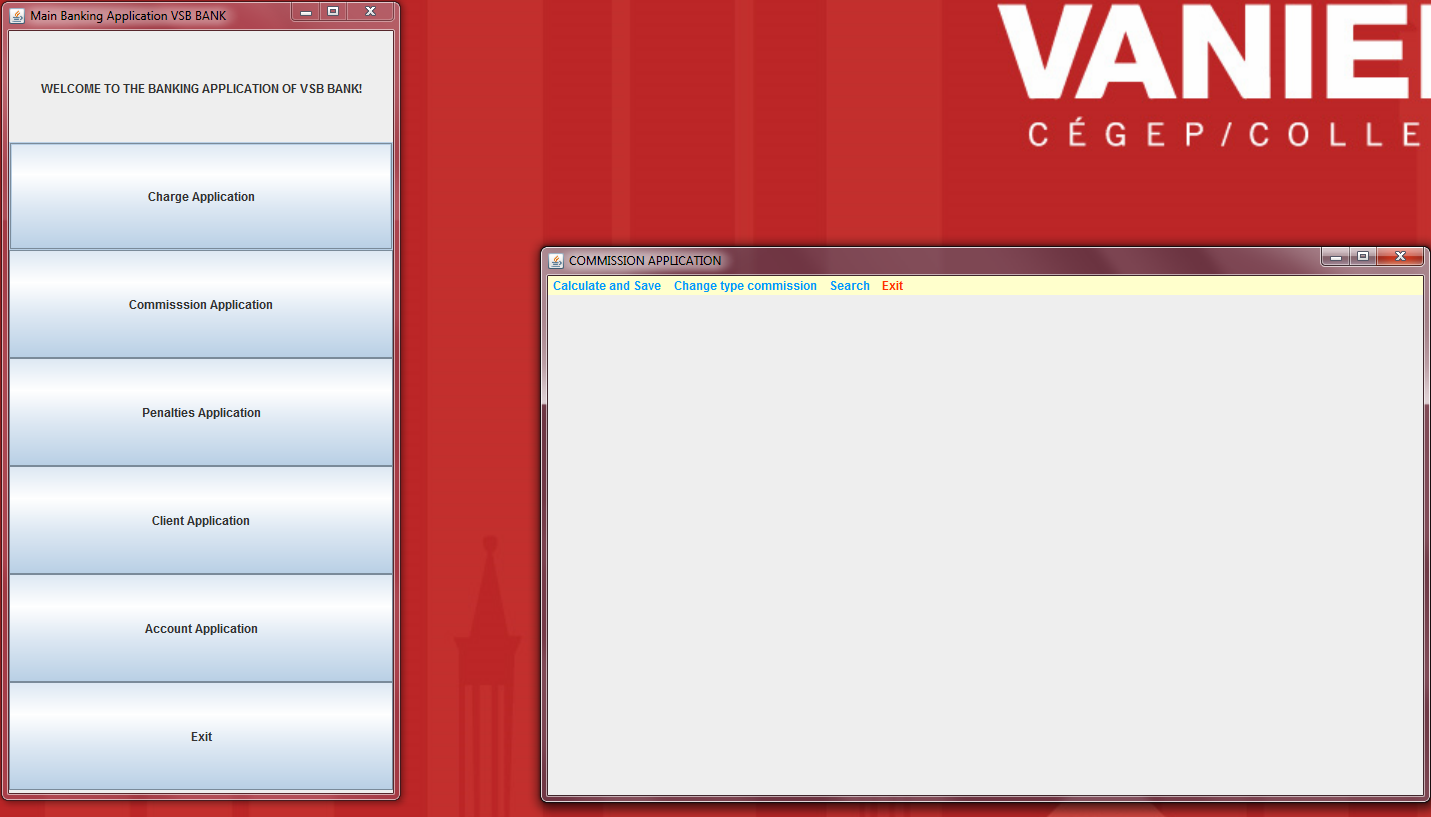




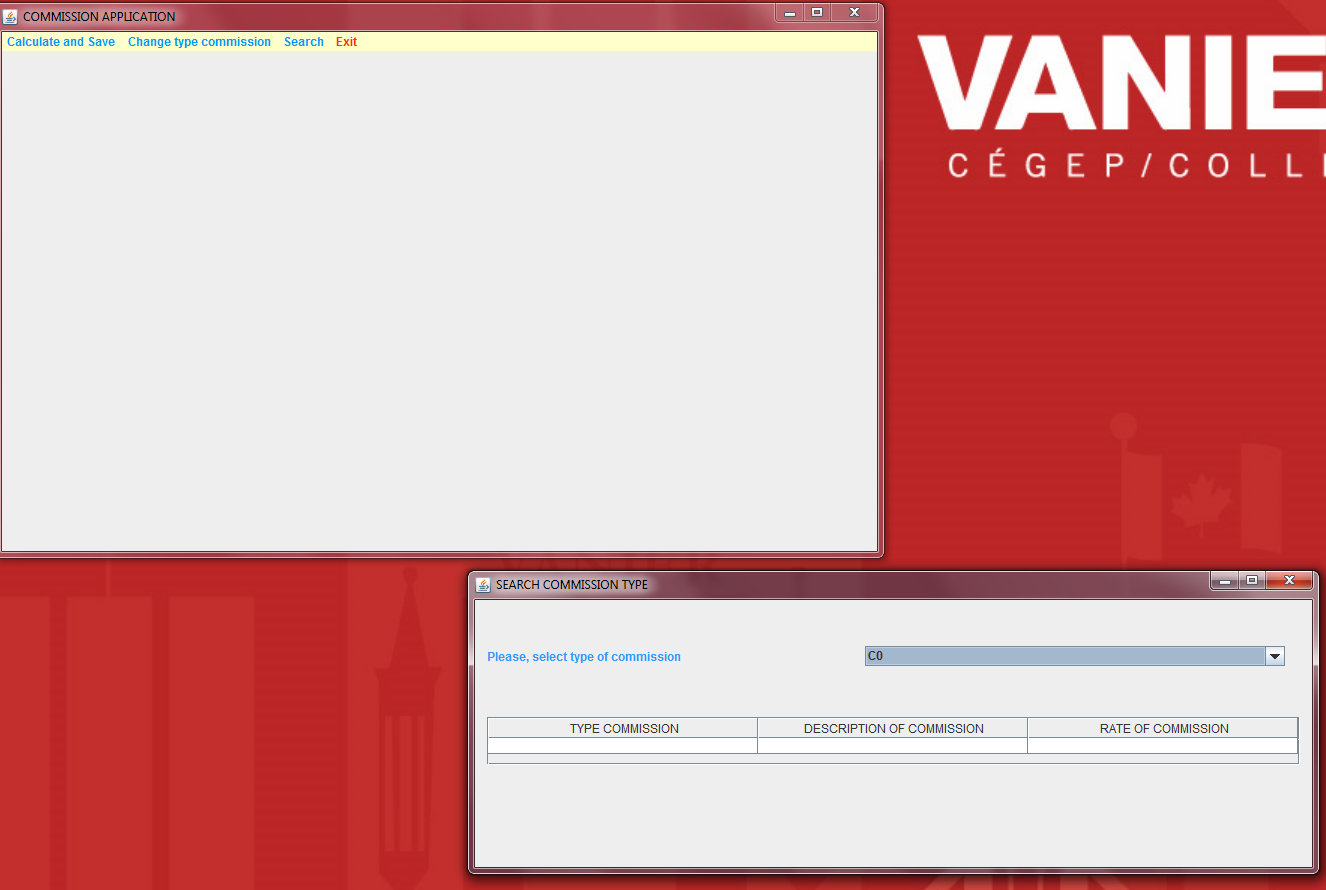


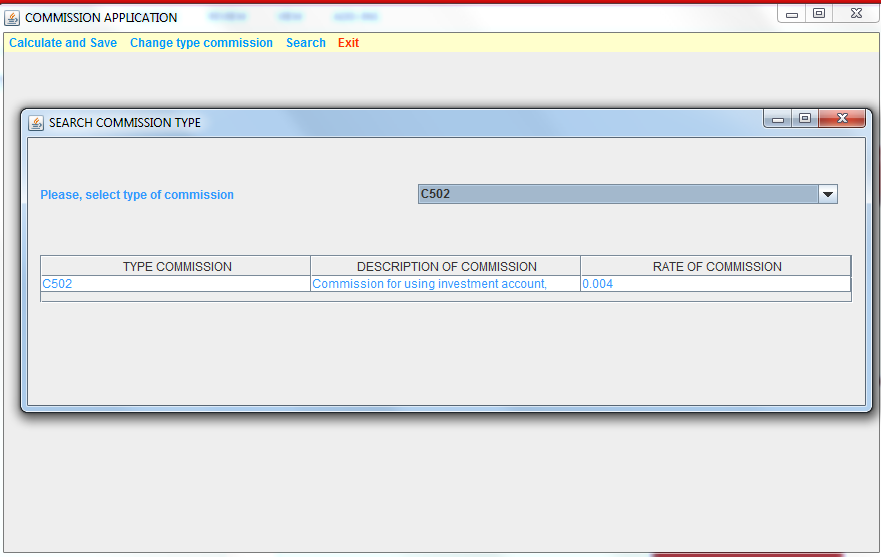
**In COMMISSION application you have to click**

* 1. **SEARCH**

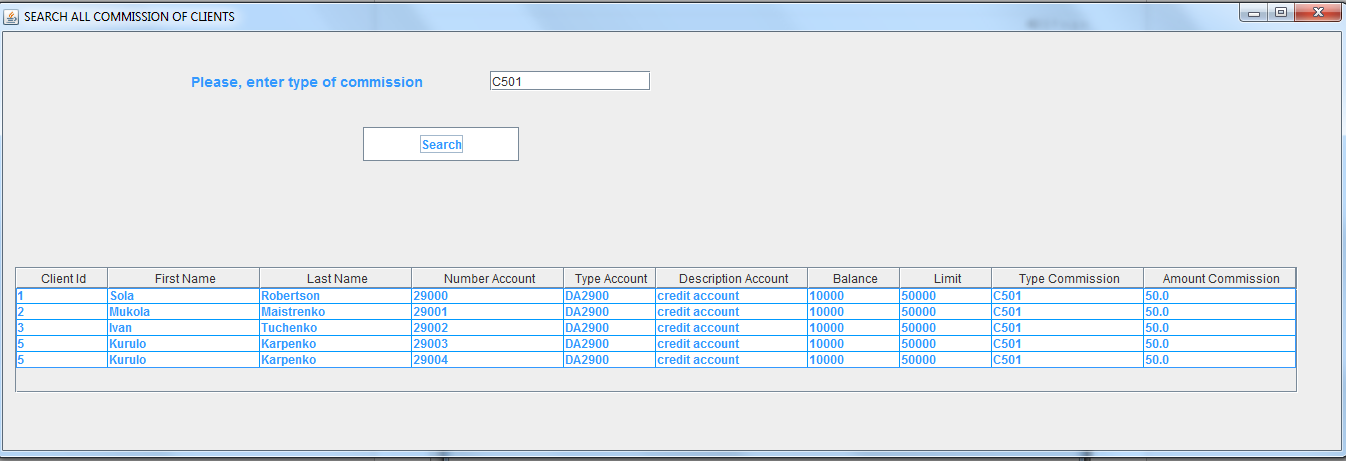


**SEARCH COMMISSION TYPE**

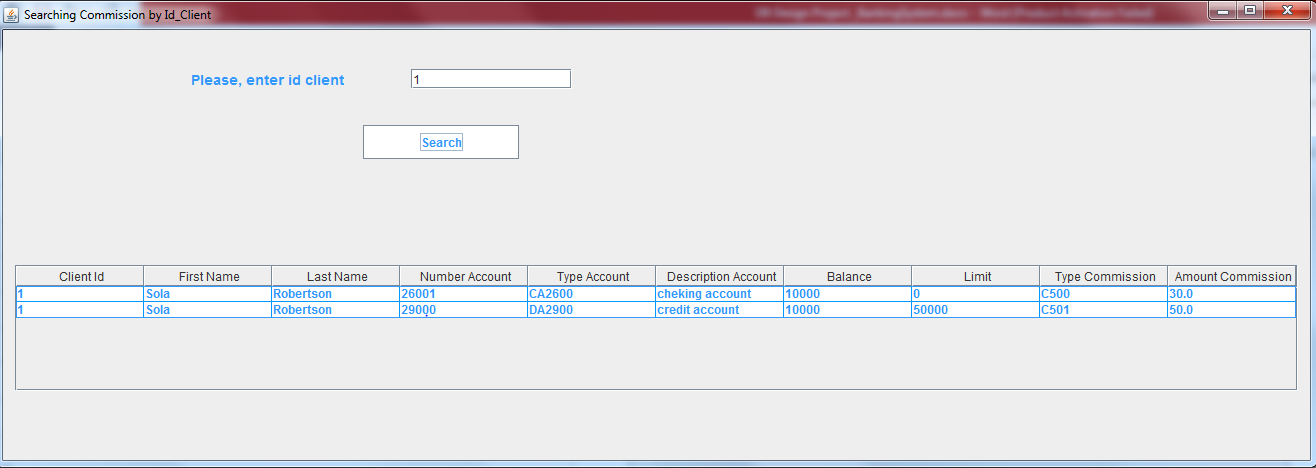




**SEARCH CLIENTS WITH TYPE COMMISSION**

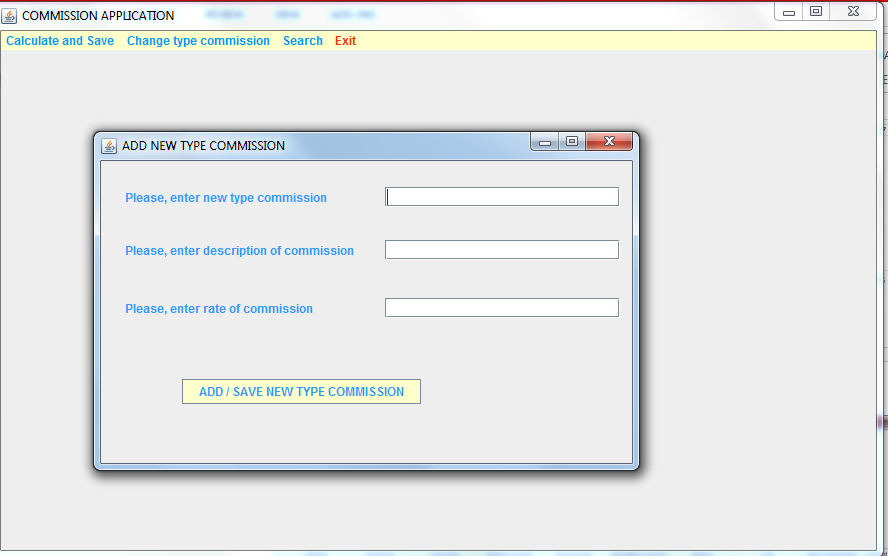


**SEARCH TYPE COMMISSION BY USING ID CLIENT**

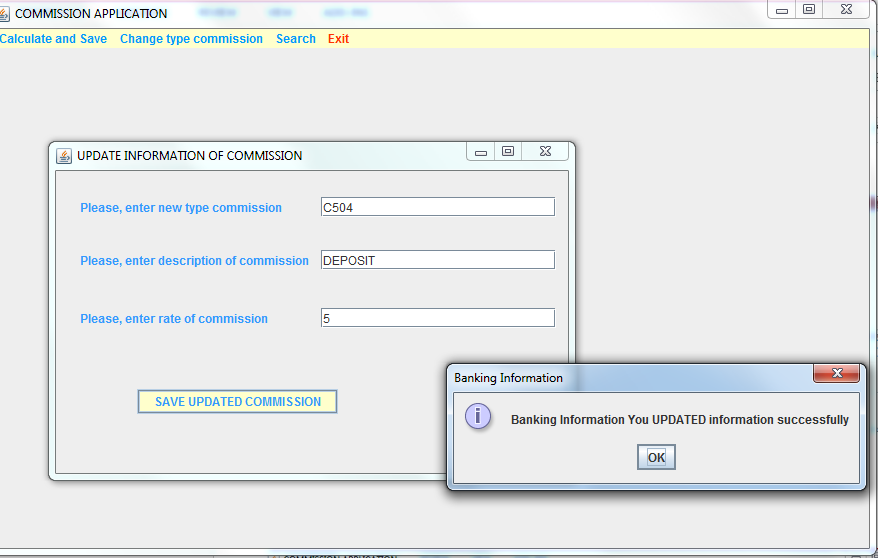


* 1. **CHANGE TYPE COMMISSION**

**ADD NEW TYPE COMMISSION**

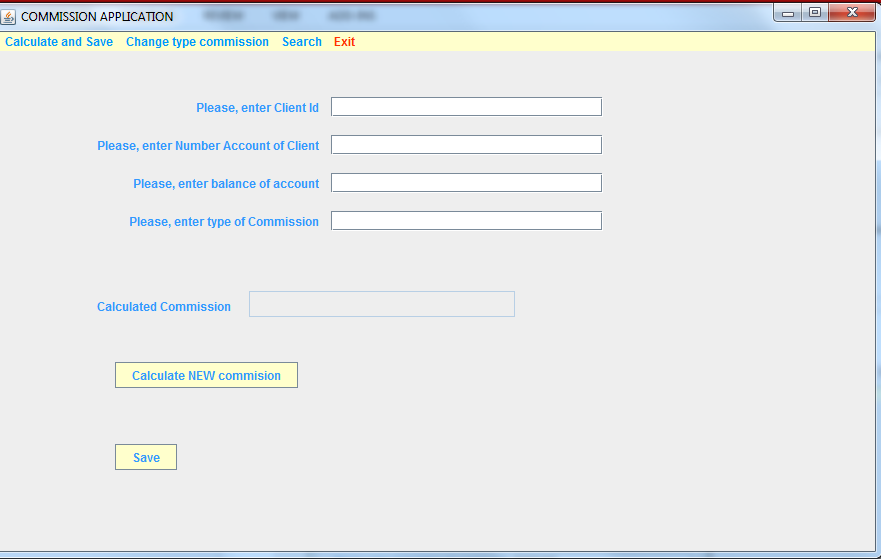


**UPDATE EXISTING TYPE COMMISSION**



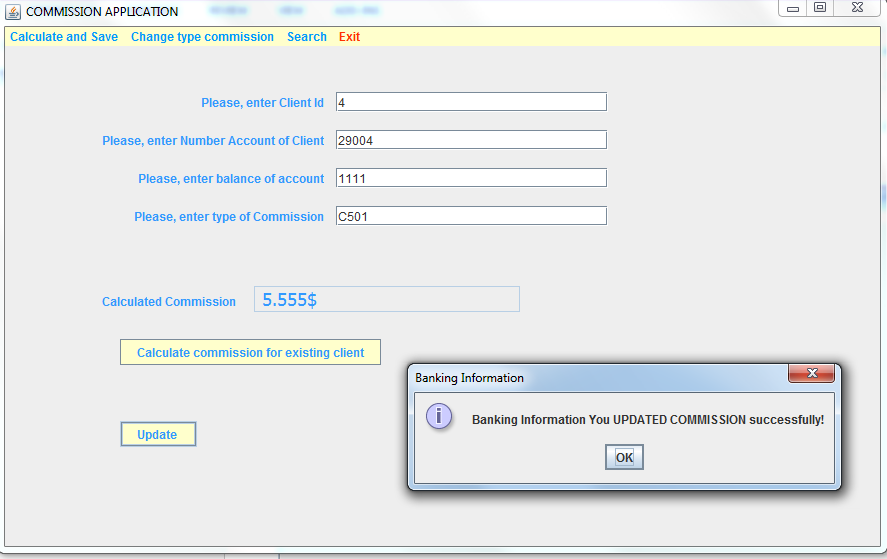
* 1. **CALCULATE AND SAVE**

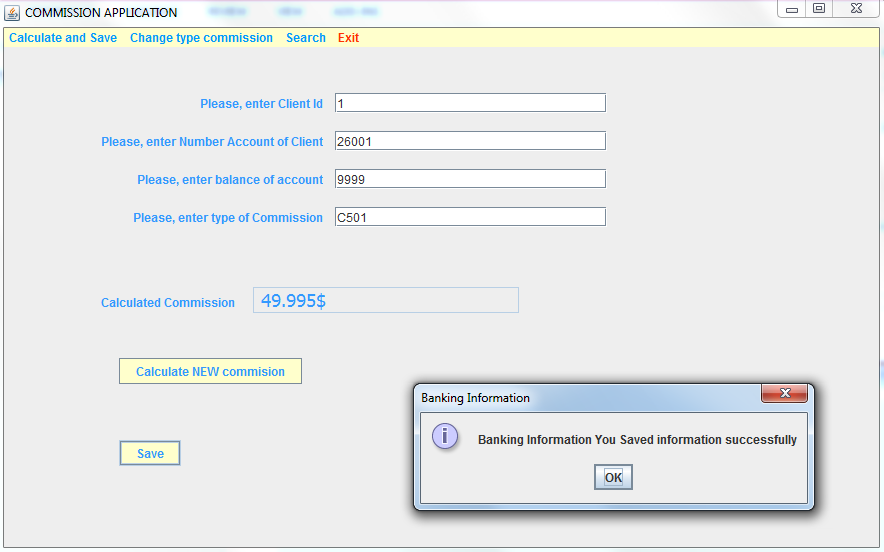
**CALCULATE NEW COMMISSION**



**UPDATE EXISTING COMMISSION**







* + 1. **Data structure**

Banking System will be coded in Java language by using data structure gui javax.swing.\* by implements ActionListener and actionPerformed.

* + 1. **Reports consideration**



* 1. **Database normalization** 
     1. **1NF – one big table**

**Banking System**

**The attributes:**

|  |  |
| --- | --- |
| Id\_client | Primary Key |
| Number\_account | Primary Key |
| Type of account |  |
| Balance |  |
| Limit - Limit |  |
| Type\_penalties - Type of Penalties |  |
| Amount\_penalties |  |
| Type\_charge |  |
| Amount\_charge |  |
| Type of Commission |  |
| Amount of Commission |  |
| Fname |  |
| Lname |  |
| Address |  |
| Tel |  |
| E-mail |  |
| SSN |  |
| Type\_account |  |
| Desc\_account |  |
| Number\_account |  |
| Type\_charge |  |
| Desc\_charge |  |
| Rate\_charge |  |
| Period\_calc\_CHARGE |  |
| Day\_pay\_charge |  |
| Type\_commission |  |
| Desc\_commission |  |
| Rate\_commission |  |
| Period\_calc\_COMMISSION |  |
| Day\_pay\_commission |  |
| Type\_penalties |  |
| Desc\_penalties |  |
| Rate\_penalties |  |
| Period\_calc \_PENALTIES |  |
| Day\_pay\_penalties |  |

**1NF** is okey Table BankingSystem in 1NF (Table Banking System meet the definition of a relation), each column has a unique name, all data in a column are the same kind, cells of the table hold a single value.

**Id\_client Number\_account**

**Type of account**

**Balance**

**Limit - Limit**

**Type\_penalties - Type of Penalties**

**Amount\_penalties**

**Type\_charge**

**Amount\_charge**

**Type of Commission**

**Amount of Commission**

**Fname**

**Lname**

**Address**

**Tel**

**E-mail**

**SSN**

**Type\_account**

**Desc\_account**

**Number\_account**

**Type\_charge**

**Desc\_charge**

**Rate\_charge**

**Period\_calc**

**Day\_pay\_charge**

**Type\_commission**

**Desc\_commission**

**Rate\_commission**

**Period\_calc**

**Day\_pay\_commission**

**Type\_penalties**

**Desc\_penalties**

**Rate\_penalties**

**Period\_calc**

**Day\_pay\_penalties**

* + 1. **2NF – partial dependency**

**1 NF is okey**

2 NF is NOT okey – NOT all nonkey attributes are determined by the entire primary key **Id\_client Number\_account**, we have **partial dependency**.

DATABASE AFTER 2NF (RESULT WE HAVE THREE TABLES :)

**Id\_client Number\_account**

**Type of account**

**Balance**

**Limit - Limit**

**Type\_penalties - Type of Penalties**

**Amount\_penalties**

**Type\_charge**

**Amount\_charge**

**Type of Commission**

**Amount of Commission**

**Type\_charge**

**Desc\_charge**

**Rate\_charge**

**Period\_calc**

**Day\_pay\_charge**

**Type\_commission**

**Desc\_commission**

**Rate\_commission**

**Period\_calc**

**Day\_pay\_commission**

**Type\_penalties**

**Desc\_penalties**

**Rate\_penalties**

**Period\_calc**

**Day\_pay\_penalties**

**Id\_client**

**Fname**

**Lname**

**Address**

**Tel**

**E-mail**

**SSN**

**Number\_account**

**Type\_account**

**Desc\_account**

**Number\_account**

* + 1. **3NF – transition dependency**

1 NF is okey

2 NF is okey Table in Second Normal Form

**3NF? no nonkey attributes are determined by any other nonkey attributes? No**:

DATABASE AFTER 3NF (RESULT WE HAVE SIX TABLES :)

**Id\_client Number\_account**

**Type of account**

**Balance**

**Limit**

**Type\_penalties**

**Amount\_penalties**

**Type\_charge**

**Amount\_charge**

**Type of Commission**

**Amount of Commission**

**Id\_client**

**Fname**

**Lname**

**Address**

**Tel**

**E-mail**

**SSN**

**Number\_account**

**Type\_account**

**Desc\_account**

**Number\_account**

**Type\_charge**

**Desc\_charge**

**Rate\_charge**

**Period\_calc\_charge**

**Day\_pay\_charge**

**Type\_commission**

**Desc\_commission**

**Rate\_commission**

**Period\_calc\_commission**

**Day\_pay\_commission**

**Type\_penalties**

**Desc\_penalties**

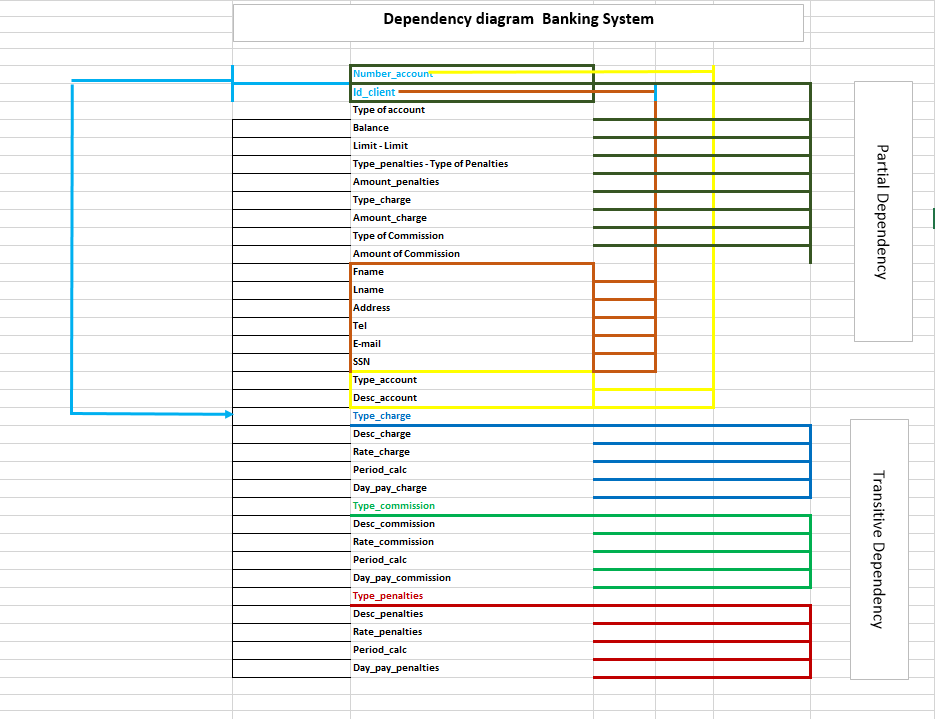
**Rate\_penalties**

**Period\_calc\_penalties**

**Day\_pay\_penalties**

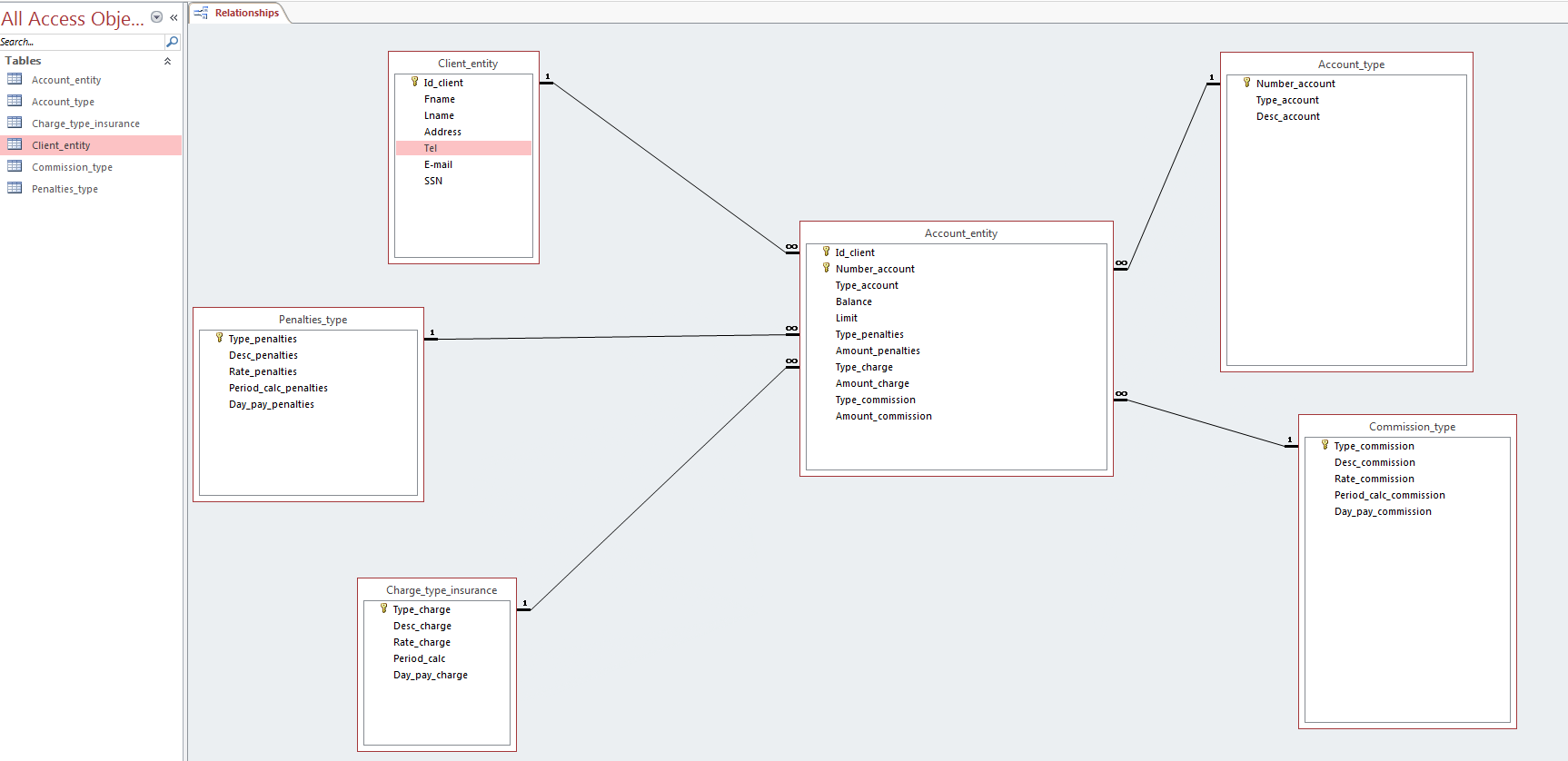
All determinates are candidate keys

* + 1. **Dependency diagram**



* 1. **Entity relationship diagram (Access/SQL development)**

**Relationship is one to many**



* 1. **System architecture**

**class database**

**Object – records**

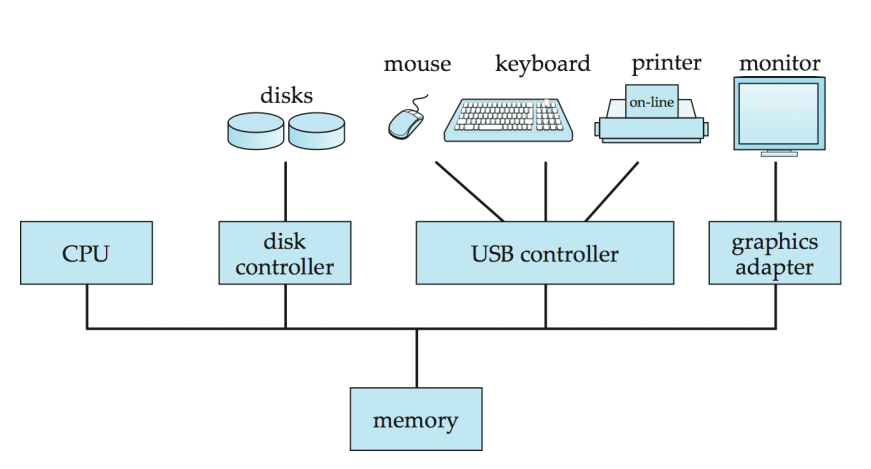
**Attribute – field**

A system architecture or systems architecture is the conceptual model that defines the structure, behavior, and more views of a system.

For Banking System is efficient centralized system - run on a single computer system and do not interact with other computer systems. General-purpose computer system: one to a few CPUs and a number of device controllers that are connected through a common bus that provides access to shared memory.

Banking System is single-user system (e.g., personal computer or workstation): desk-top unit, single user, usually has only one CPU and one or two hard disks; the OS may support only one user.

**A Centralized Computer Banking System**

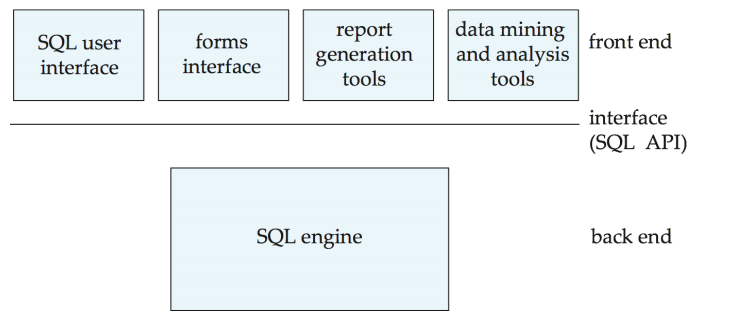


Database of Banking System functionality can be divided

back-end: manages access structures, query evaluation and optimization, concurrency control and recovery.

front-end: consists of tools such as forms, report-writers, and graphical user interface facilities.

The interface between the front-end and the back-end is through SQL or through an application program interface.



**4. System Implementation (Java, Oracle)**

**4.1. Front End System Java**

The frontend of the system is the sole point of contact for an end user with the system. Front-end development is what is used to create the visual display that the end user of a website experiences. Back-end development is what makes the presentation of front-end development possible.

For Banking System front end system is possible Java, C++, VB.

**4.2. Back-End System (textfile, Oracle)**

For back-end system for Banking System is possible Oracle.

**4.3. Structure chart (for searching)**

A Structure Chart (SC) is a chart which shows the breakdown of a system to its lowest manageable levels. They are used in structured programming to arrange program modules into a tree. Each module is represented by a box, which contains the module's name. The tree structure visualizes the relationships between modules.

**4.4. Prototype**

A prototype is an early sample, model, or release of a product built to test a concept or process or to act as a thing to be replicated or learned from.

Software prototyping is the activity of creating prototypes of software applications, i.e., incomplete versions of the software program being developed.

Prototype is a working model of software with some limited functionality.

Prototyping is used to allow the users evaluate developer proposals and try them out before implementation. It also helps understand the requirements which are user specific and may not have been considered by the developer during product design.

There are different types of software prototypes used in the industry: Throwaway/Rapid Prototyping, Evolutionary Prototyping, Incremental Prototyping, Extreme Prototyping.

For Banking System we can use evolutionary prototyping.



**5. Testing/Security** (for agile method process for spliced task; negative or positive feedback

**5.1. Unit testing**

Method of testing that verifies the individual units of the code is working properly. Test the smallest unit in source code.

Two most used testing framework in Java Junit, TestNG. Junit is test framework in java by creating test class and test case, by using method assertEquals(), assertTrue(), assertNotNull() (Tools>Create Junit Test).

In Banking System created exceptions, test package and tested.

**5.2. System testing**

System Testing is a level of the software testing where a complete and integrated software is tested. The purpose of this test is to evaluate the system's compliance with the specified requirements.

The following examples are different types of testing that should be considered during System testing:

Graphical user interface testing.

Usability testing.

Software performance testing.

Compatibility testing.

Exception handling.

Load testing.

Volume testing.

Stress testing

Scalability testing

Sanity testing

Smoke testing

Exploratory testing

Ad hoc testing

Regression testing

Installation testing

Maintenance testing [clarification needed]

Recovery testing and failover testing.

First test the GUI related issues on all these above mentioned modules by checking that font size, alignment, display images should work properly on all the modules.

After checking the GUI issues move towards functionality related issues by checking that the requirements of client have been met or not.

After checking functionality check whether the application is user friendly or not by checking that proper error message should be displayed on screen or not.

Banking System is user friendly the behavior and even the believed expectations of the customer.

**5.3. Integration testing**

Integration testing is a software development process which program units are combined and tested as groups in multiple ways.

Any of Black Box Testing, White Box Testing, and Gray Box Testing methods can be used. Normally, the method depends on definition of ‘unit’.

Integration Testing is performed after Unit Testing and before System Testing.

For Banking System we can use bottom testing where the lowest level components are tested first, then used to facilitate the testing of higher level components. The process is repeated until the component at the top of the hierarchy is tested.