
**Ahsanullah University of Science and Technology**
*Department of Computer Science & Enineering*

DATABASE LAB
CSE3104

Final Project Report

---

# Banking service

---

Submitted by

Shahajadi Sadia Afsana 18.01.04.138
Ashfakur Rahman Fahim 16.02.04.032

April 9, 2021

# Contents

# 1 Introduction

Banking Service System is a java based software platform which allows a user to transfer his all money into a virtual currency. As a result people need not carry cash money with them and also not to fear of losing their cash by theft or pick-pocketing while moving or travelling different places.

# 2 Objective

Banking Service System is an unique way of transacting money in a safer way. People need to pay money for different services beyond official/banking hours. They also need to send money to a person staying in remote places. All these services can easily be done if a person can keep sufficient balance into a portable bank. This banking service is prepared for the people to make financial transaction in any emergency need, at any time, from anywhere and without any risk.

# 3 Features

- **ADMIN**
- **AGENT**
- **USER**

## 3.1 Admin

Admin have to log in through the password we manually entered into database.Admin will get 3 options that are:

- **Send money :** admin can send money to any one . like admin or user or agent.
- **Add new admin :** one admin can add new admin
- **History:** admin can see his transaction history and profit history here.

## 3.2 Agent

agent have to sign up first. Then with that user name and password agent can log in to their personal account.Agent will get 5 options that are:

- **Send money :** Agent can send money to agents and users .
- **Payment :** Agent can pay money to agent or to user
- **Mobile recharge :** Agent can recharge money to anyone
- **History :** Agent can see his transaction history here.
- **Update information :** Agent can update his info and can change it easily by pressing "get info" button

### 3.3 User

user have to sign up first. Then with that user name and password user can log in to their personal account. User will get 6 options that are :

- **Send money :** User can send money to users .
- **Cash Out :** User can use cash out through agent.
- **Payment:** User can pay money for shopping .
- **Mobile recharge :** user can recharge money to user.
- **History:** User can see his transaction history here.
- **Update information :** User can update his info and can change it easily by pressing "get info" button .

## 4 Query Tables Construction

**Create table UserInfo**

( UserId int identity(20001,1) primary key, PhoneNo int unique not null, FirstName varchar(50) not null, LastName varchar(50) not null, MiddleName varchar(50) null, DateOfBirth varchar(10) not null, NationalId varchar(20) null, BirthCertificate varchar(20) null, PresentAddress varchar(50) not null, PremanentAddress varchar(50) not null, PasswordNo varchar(20) not null, ReferenceId varchar(20) not null, AccountType int not null );

**insert into UserInfo values**

(112233,'admin','admin','admin','1996-11-26','1234567','9876543','mohakhali','dhaka','1234','5678',3);

**insert into UserInfo values**

(112235,'sadia','sadia','sadia','1996-11-26','1234567','9876543','rampura','dhaka','1234','5678',3);

**Create table TransTable**

(TransID int identity(50000,1) primary key, UserId int FOREIGN KEY REFERENCES UserInfo(UserId) not null, SenderId varchar(50) not null, ReciverId varchar(50) not null, TransType varchar(50) not null, Ammount float not null, TransDate Date not null );

**Create table BlanceTable**

( BlanceID int identity(70000,1) primary key, UserId int FOREIGN KEY REFERENCES UserInfo(UserId) not null, CurrentBlance float not null );

**Create table ProfitTable**

( ProfitID int identity(110000,1) primary key, UserId int FOREIGN KEY REFERENCES UserInfo(UserId) not null, SenderId varchar(50) not null, TransType varchar(50) not null, Ammount float not null, TransDate Date not null
);

## 5 Schema Diagram

**Here is the Schema diagram of the project.**

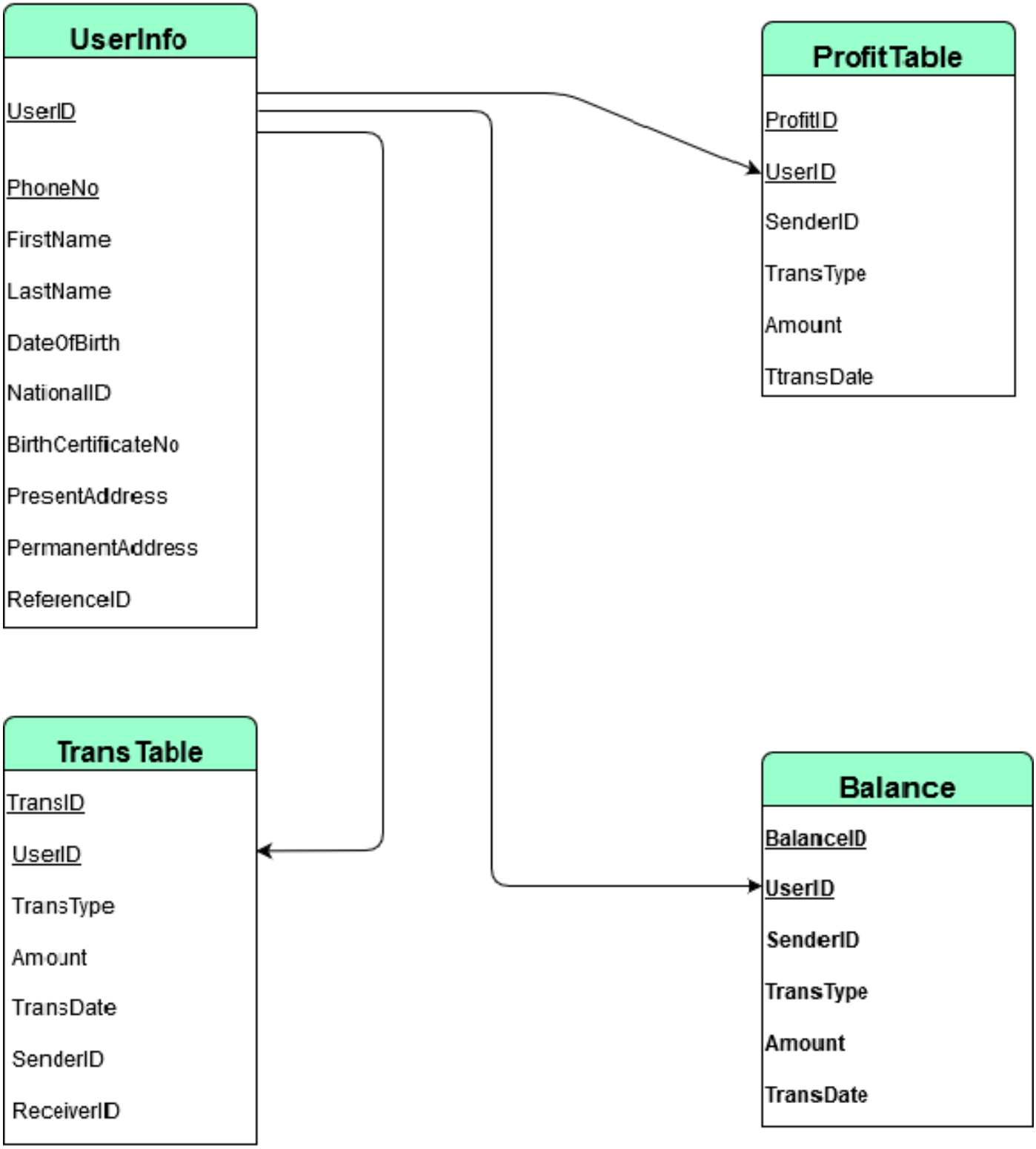


Figure 1: Schema diagram of Banking Service

# 6 Entity-Relationship Diagram

**Here is the ER- diagram of the project.**

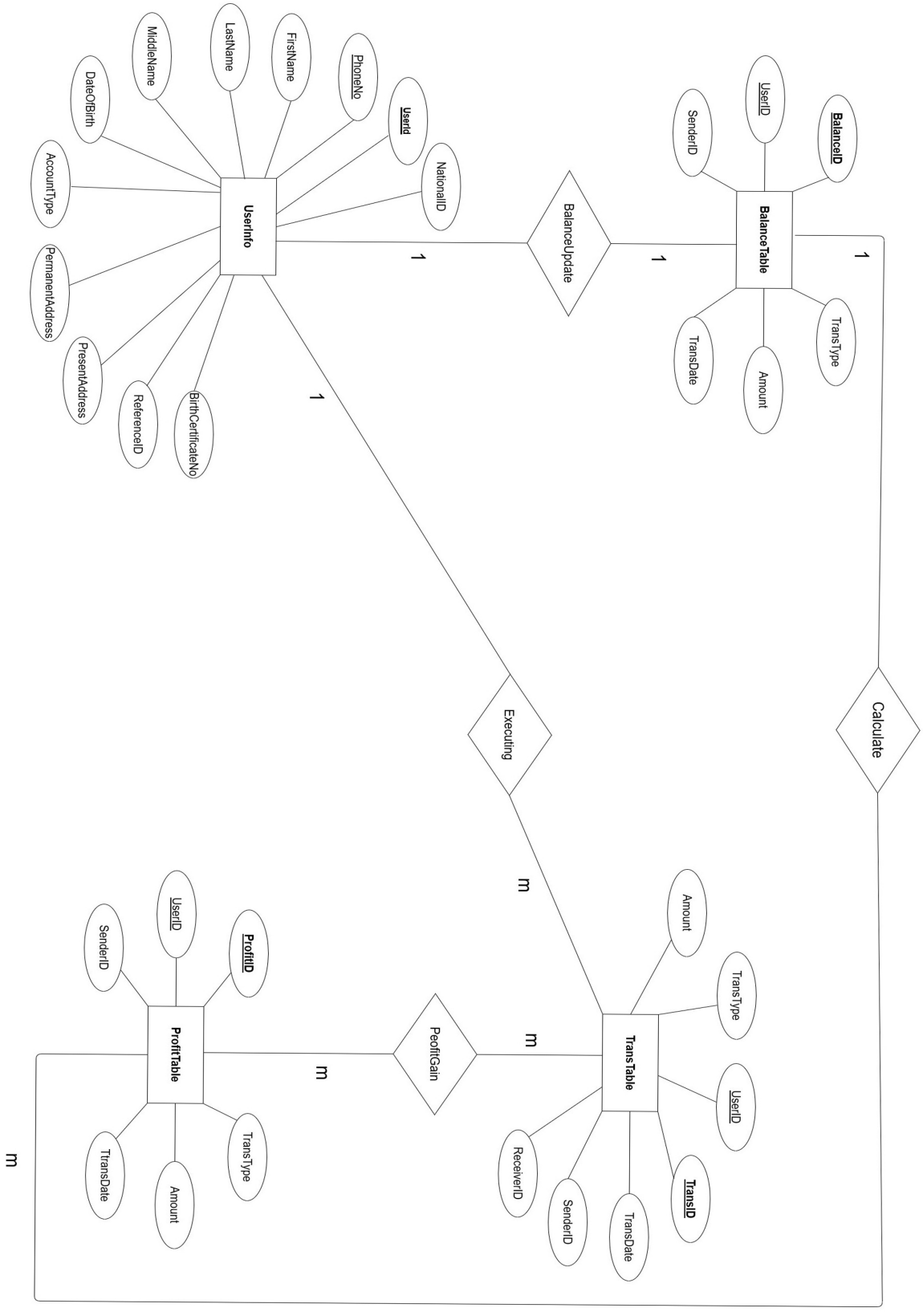


Figure 2: E-R diagram of Banking Service

### 6.1 Entity and its attributes

**Database Tables**

**Table 01: UserInfo**

| Field | Type | Constraint |
|---|---|---|
| UserId | Int | P.K (IS: 20001) |
| PhoneNo | Int | P.K |
| FirstName | Varchar(50) | Not Null |
| LastName | Varchar(50) | Not Null |
| MiddleName | Varchar(50) | Null |
| DateOfBirth | Date | Not Null |
| NationalId | Varchar(20) | Null |
| BirthCertificateNo | Varchar(20) | Null |
| PresentAddress | Varchar(50) | Not Null |
| PermanentAddress | Varchar(50) | Not Null |
| AccountType | Int | Not Null |
| ReferenceId | Varchar(20) | Not Null |

**Table 02: TransTable**

| Field | Type | Constraint |
|---|---|---|
| TransId | Int | P.K (IS: 50000) |
| UserId | Int | F.K (UserInfo) |
| TransType | Varchar(50) | Not Null |
| Amount | Float | Not Null |
| TransDate | Date | Not Null |
| SenderId | Varchar(50) | Not Null |
| RecieverId | Varchar(50) | Not Null |

**Table 03: Balance**

| Field | Type | Constraint |
|---|---|---|
| BalanceId | Int | P.K (IS: 70000) |
| UserId | Int | F.K (UserInfo) |
| CurrentBalance | Float | Not Null |

**Table 04: ProfitTable**

| Field | Type | Constraint |
|---|---|---|
| ProfitId | Int | P.K (IS: 110000) |
| UserId | Int | F.K (UserInfo) |
| SenderId | Varchar(50) | Not Null |
| TransType | Varchar(50) | Not Null |
| Amount | Float | Not Null |
| TransDate | Date | Not Null |

### 6.2 Relationship

**Relations**

**Relation 01: UserBalance**

| Field | Type | Constraint |
|---|---|---|
| UserId | Int | F.K (UserInfo) |
| CurrentBalance | float | |

**Relation 02: BalanceUpdate**

| Field | Type | Constraint |
|---|---|---|
| CurrentBalance | Float | |
| TransType | Varchar(50) | |
| Amount | Float | |
| TransId | int | F.K (TransTable) |
| UserId | Int | F.K (UserInfo) |

**Relation 03: SendMoney**

| Field | Type | Constraint |
|---|---|---|
| ReiceverId | Float | |
| TransType | Varchar(50) | |
| Ammount | Float | |
| Date | Date | |
| UserId | Int | F.K (UserInfo) |

**Relation 04: Payment**

| Field | Type | Constraint |
|---|---|---|
| ReiceverId | Float | |
| TransType | Varchar(50) | |
| Ammount | Float | |
| Date | Date | |
| UserId | Int | F.K (UserInfo) |

**Relation 05: CashOut**

| Field | Type | Constraint |
|---|---|---|
| ReiceverId | Float | |
| TransType | Varchar(50) | |
| Ammount | Float | |
| Date | Date | |
| UserId | Int | F.K (UserInfo) |

**Relation 06: MobileRecharge**

| Field | Type | Constraint |
|---|---|---|
| ReiceverId | Float | |
| TransType | Varchar(50) | |
| Ammount | Float | |
| Date | Date | |
| UserId | Int | F.K (UserInfo) |

**Relation 07: CashIn**

| Field | Type | Constraint |
|---|---|---|
| ReiceverId | Float | |
| TransType | Varchar(50) | |
| Ammount | Float | |
| Date | Date | |
| UserId | Int | F.K (UserInfo) |

**Relation 08: Profit**

| Field | Type | Constraint |
|---|---|---|
| TransId | Int | F.K (TransTable) |
| Profit | float | |

## 7 Conclusion

Carrying cash for transaction in different purposes is always at risk. Considering this risk factor banking service should be more developed as an easily operated financial device for the greater common users group. We have to work more to make it more secure and easily operated by the general users.

## 8 Contribution:

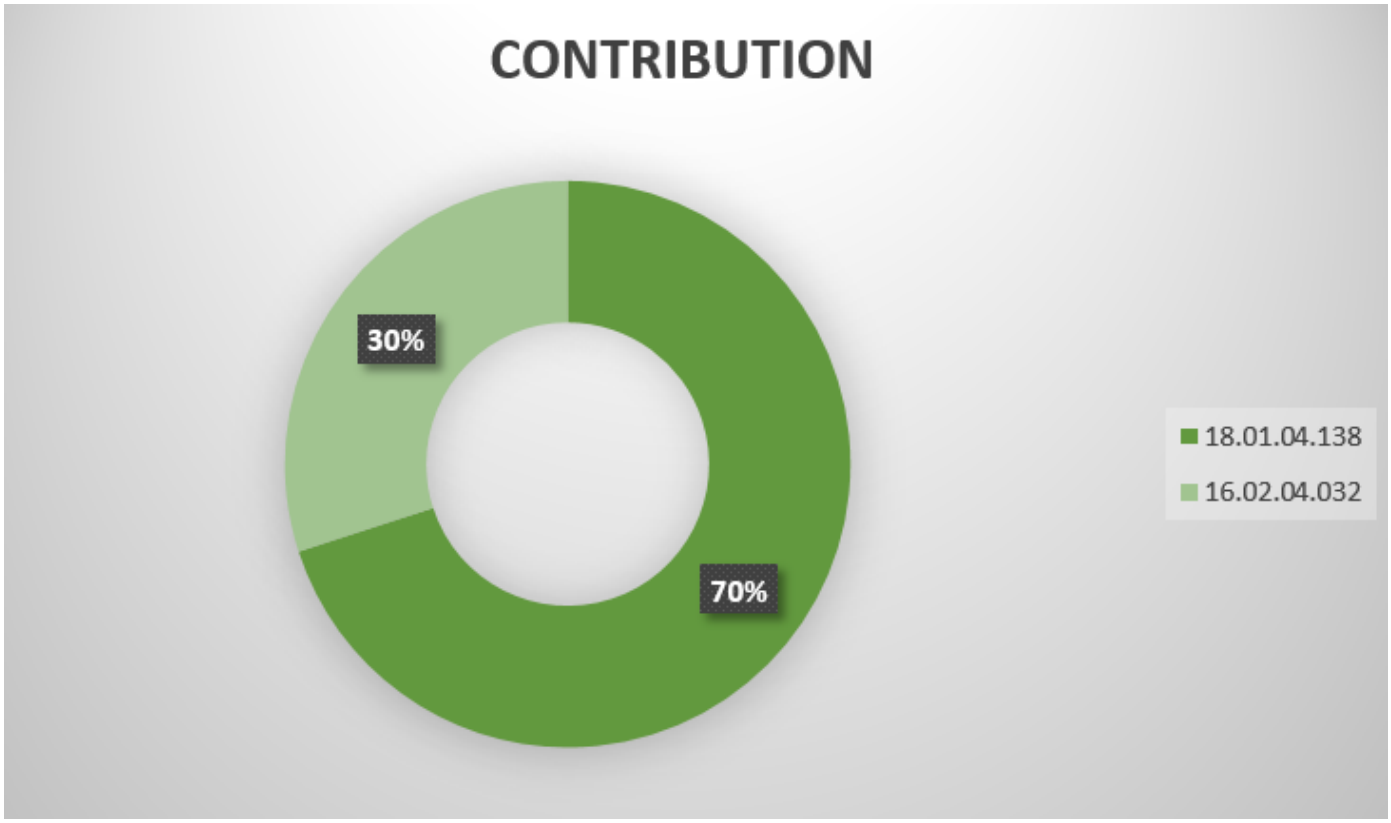


Figure 3: Our project contribution