**BEARCAT BANK**

Project Report



Group Members : Snehal Mahajan

Shrinidhi Shetty

Krishnaja Pinnamaneni

## Reshma Punukollu

**Carl H. Lindner College of Business, University of Cincinnati**

**TABLE OF CONTENTS**

[**INTRODUCTION**](#_ig7rza7pb8wt) **3**

[**PROJECT SCHEDULE**](#_u2pdduse9nrs) **4**

[**USER STORIES**](#_z552muu3n5sl) **5**

**INITIAL SCREEN LAYOUT** **6**

**ENTITY-RELATIONSHIP (ER) DIAGRAM 9**

**RELATIONAL MODEL 11**

**DATA DICTIONARY 13**

**DATA IMPLEMENTATION 18**

**INTRODUCTION**

# The goal of this project is to upgrade the business model for Bearcat bank in order to develop a solution to help grow Bearcat Bank into a great company and even more progressive business models.

# This includes improving user-interface layouts, upgrading database systems via advanced data modeling and database design techniques, making Bearcat banking services more user-friendly and enhancing feasibility for the business. Through this project, our aim is to model and present a Database design for the effective structuring of the Bearcat Bank system.

.

As part of our first iteration, we will be understanding business requirements. Also we will be designing the following elements

* User Stories
* UI Screen layout
* ER Diagram/Relational Model
* Data Dictionary

In the later stage, a relational model will be designed to show application of each individual element and we will create a Data Dictionary for the implementation of the Database design.

# **PROJECT SCHEDULE**

# 

# 

|  |  |
| --- | --- |
| Team Member | Contribution |
| Snehal Mahajan | Data Dictionary, Formatting , ER Diagram |
| Shrinidhi Shetty | Data Dictionary, Documentation , ER Diagram |
| Reshma Punukollu | User Story, Screen Layout, ER Diagram |
| Krishnaja Pinnamaneni | User Story, Screen Layout |

# 

# 

# **USER STORIES**

As a Customer, I want to:

* Login with my customer id and password/pin to the bank portal
* Deposit Cash/Check
* Check Account Balance
* Withdraw Cash
* Open/Close/Edit a Loan or Investment Account
* Check my profile, Account summary and transaction details
* Raise ticket if I have an issues
* Apply for debit or credit card

As a Bank Employee, I want to:

* Create, update and delete customers of the bank
* Generate a basic customer profile
* Maintain record of customers
* Add maintenance charges for customer accounts
* Browse, open and solve tickets raised by customers

As a Bank Manager, I want to :

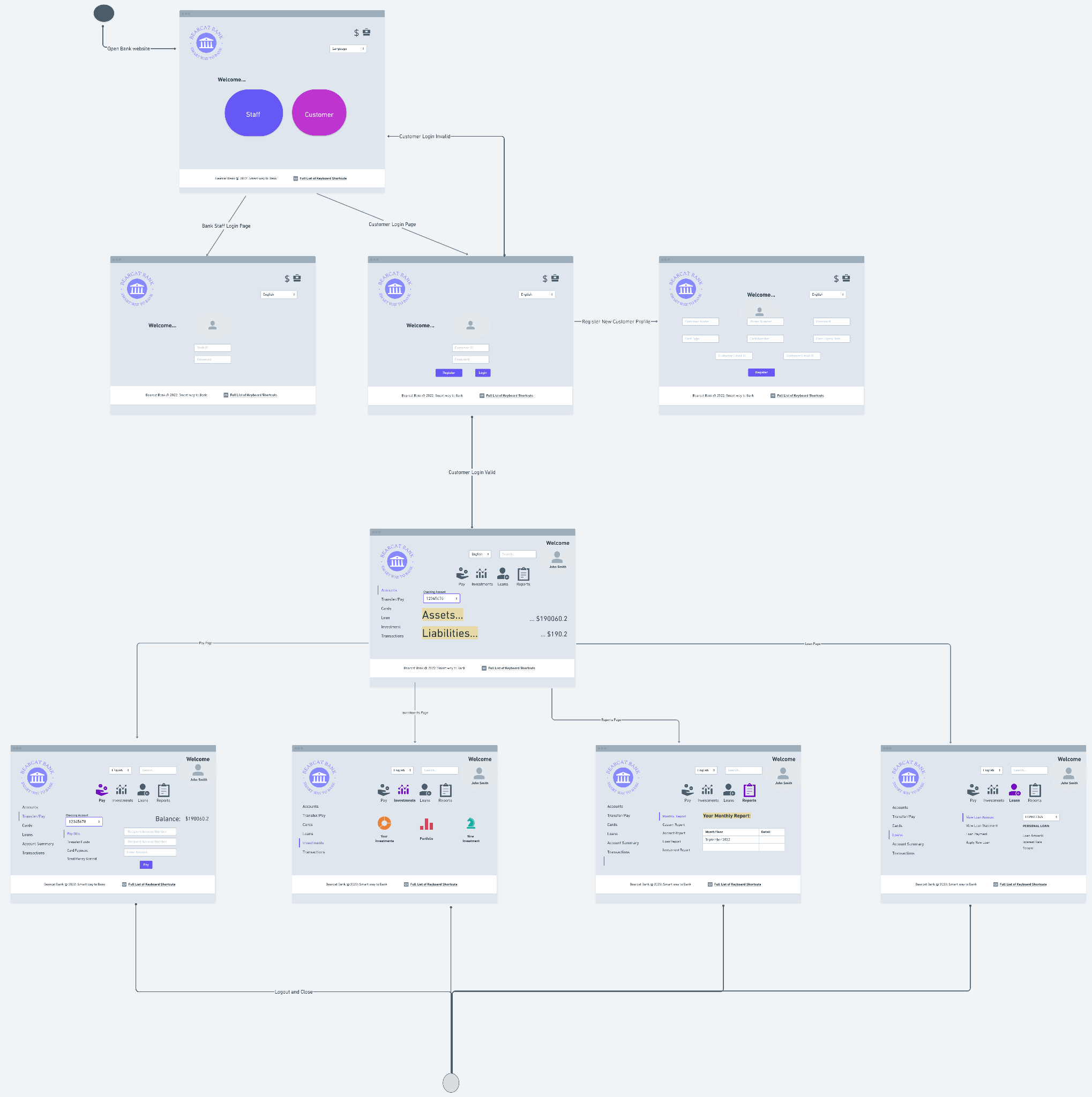
* Add and delete employees
* Approve and reject loans and investments
* Post building maintenance charges
* Generate a report of all the loans and investments
* Generate report of employees
* Generate monthly and daily charges
* Create promotional offers for customers
* Track bank overhead charges

**INITIAL SCREEN LAYOUTS**

**Customer Screen: (Bearcat Bank Website)**

Figure shows the flow of Customer screen views. The first screen asks the customer to first register and then log in using the CustomerID and password. After logging in, he/she can view their account and card details, transfer or pay money, view their investments and loans and check their account statements. If generated, Customers can also view Offers they can make use of.

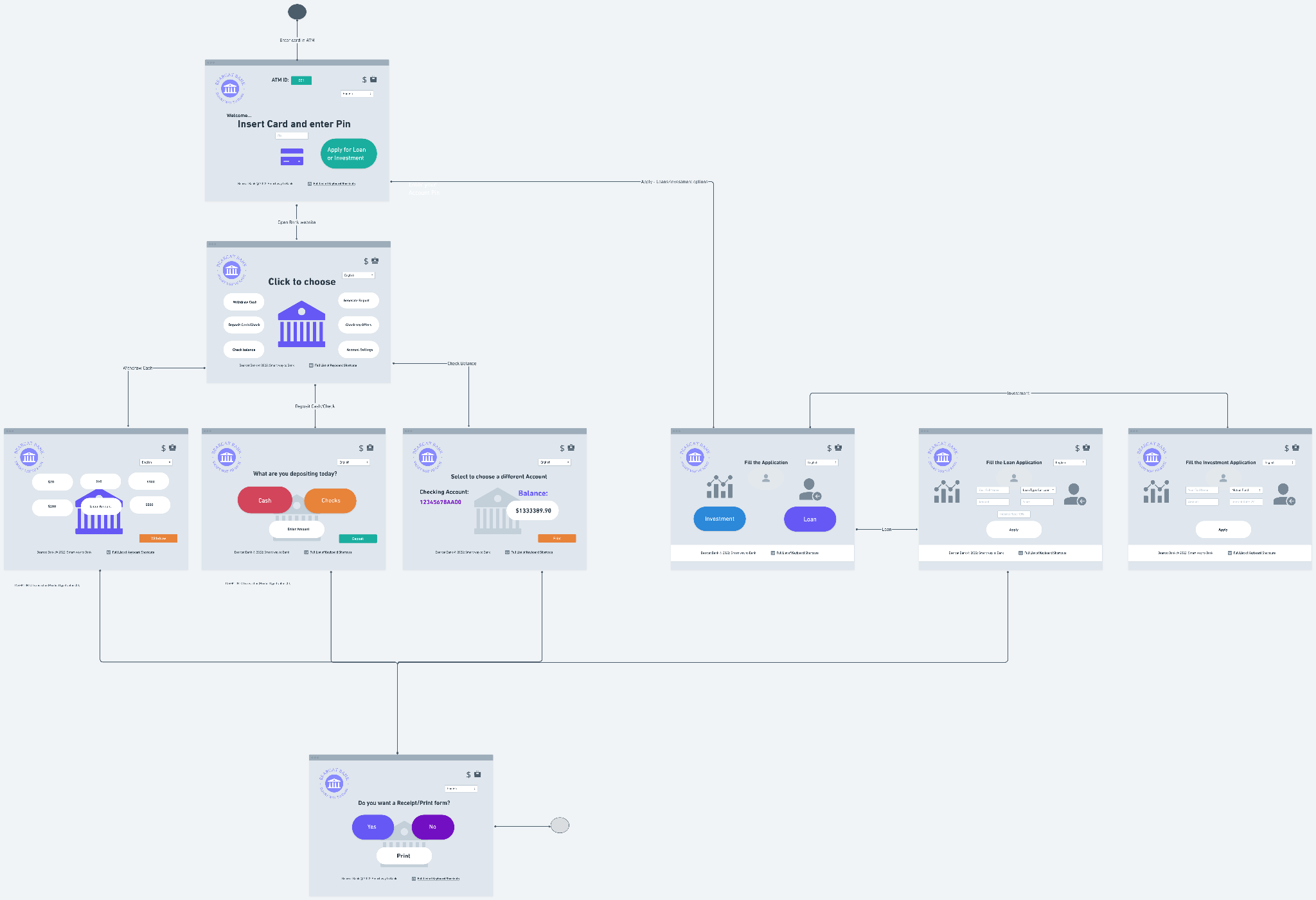
Link: <https://whimsical.com/getting-started-boards-NzJk2aGGYRyZj8gCvS1nnE>



**ATM Screen:**

Figure shows the flow of ATM screen views. The customer has to enter his Card and ATM pin. After validating the Card and pin, he/she can withdraw cash, deposit cash/check, check their account balance, request for service, change their card settings and select loan/other investment options.

Link: <https://whimsical.com/getting-started-boards-NzJk2aGGYRyZj8gCvS1nnE>

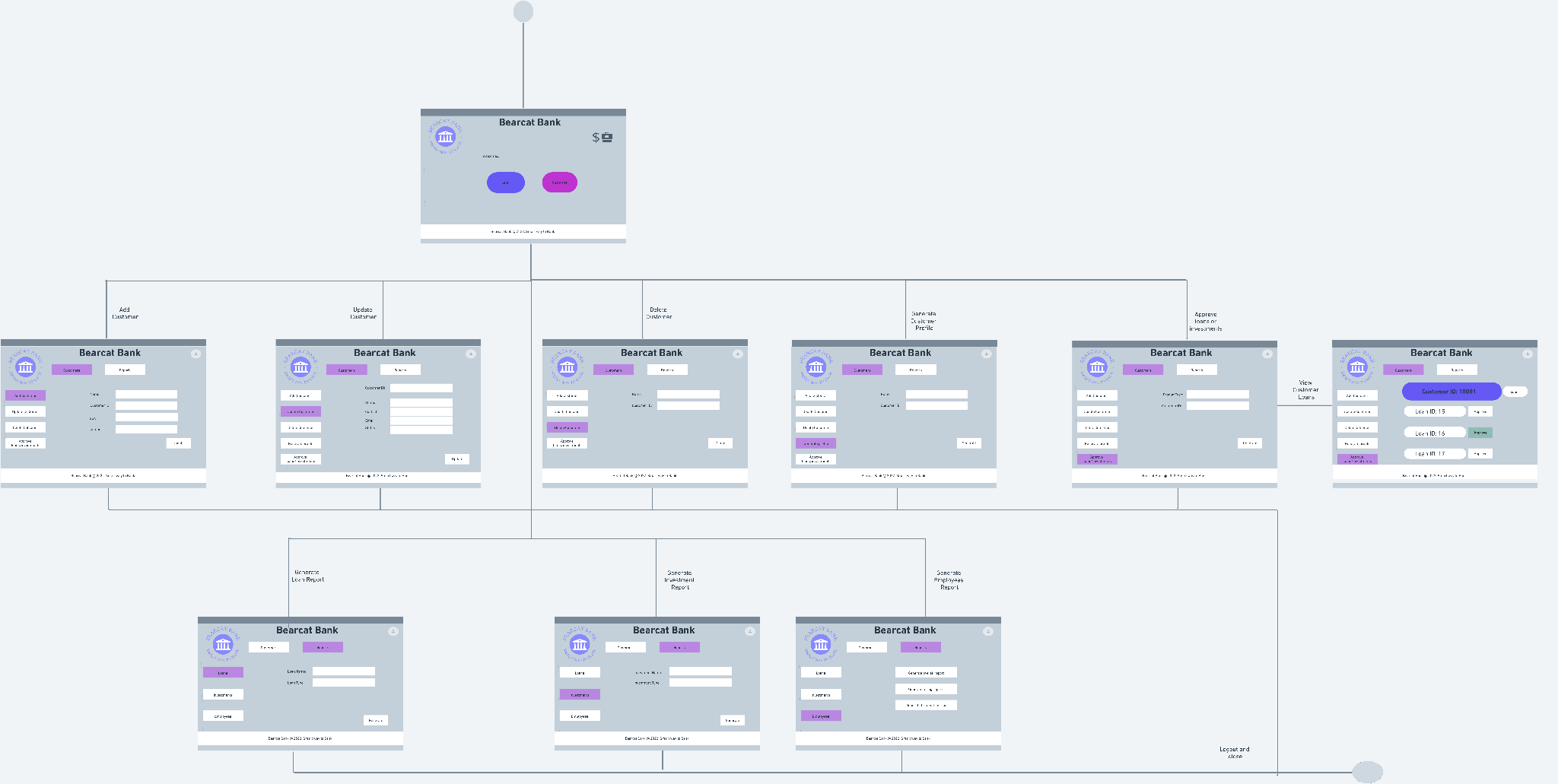


**Staff/Manager Screen:**

Figure shows the flow of Manager screen views. When the manager logs in, he can see three tabs - Customers, Reports and Offers. The Manager can add a customer, update an existing customer, delete a customer and create a basic customer profile when he/she lands on the first tab - Customers.

When the manager lands on the second tab - Reports, he/she can create loan reports, investment reports and employee related reports (Hiring report, termination report, overall report). When the manager lands on the Offers tab, he/she can create and delete promotional offers.

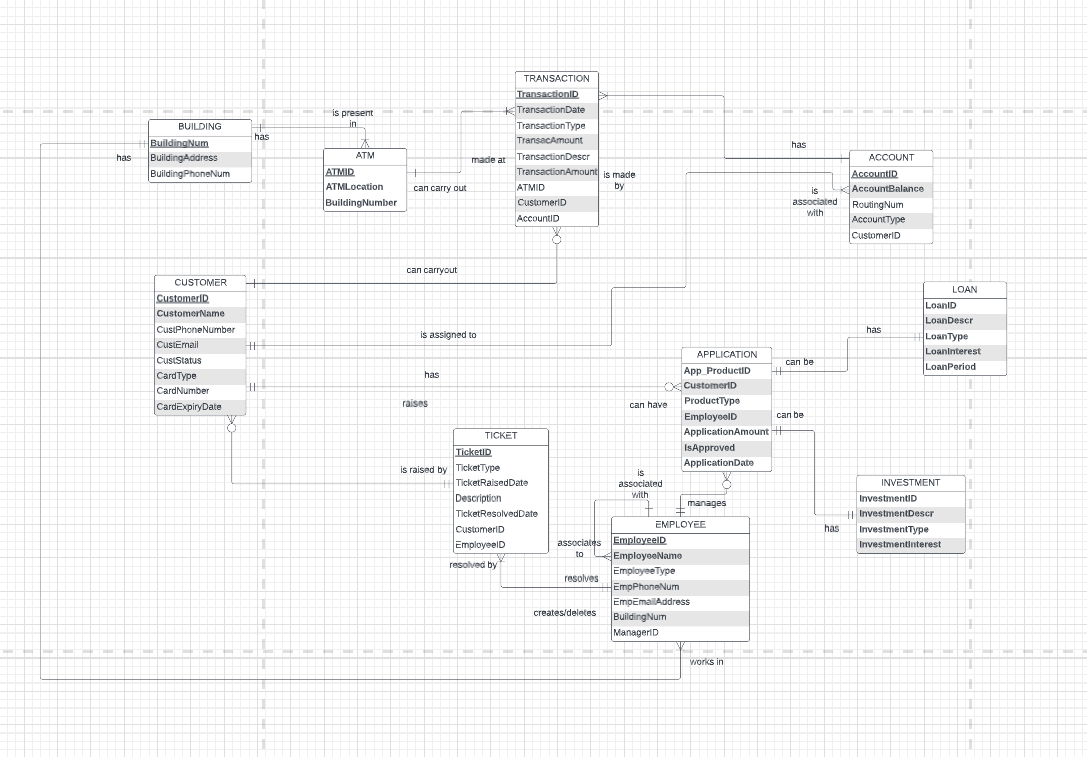
Link: <https://whimsical.com/getting-started-boards-G4FqE5nyf1nDHxYDe74RsZ>

****

**ENTITY-RELATIONSHIP (ER) DIAGRAM**

Link:

<https://lucid.app/lucidchart/3528f644-2bf8-43fb-b073-1ec2d2533d97/edit?viewport_loc=-2459%2C1213%2C3204%2C1262%2C0_0&invitationId=inv_ec46ad0b-2cb7-484f-8b9f-101eed362bd5>



**Explanation:**

Entity-Relationship diagrams are used to visualize data and relationships among the entities of the system.

Our ER diagram describes the logical structure of the Bearcat Bank system. The Bearcat Bank is a system which requires improvements and updates in their current flow to make them profitable.

We as a team are trying to develop a business solution to improve the existing system of Bearcat Bank. Our Model is broken down into 6 main subject areas which are our most important entities:

The Customer, the Employee, the Account, the Transaction. the Application, loan and investment.

The Employee is an important asset to the Bearcat Bank. We need many details for an Employee that needs to be considered to define the function that an employee performs in the bank. Details like: Employee name, Phone Number and Employee type. An employee can have zero or many applications. An employee has an unary relationship with itself as a Manager is an Employee too.

Another important entity is the Customer. Important details which need to be considered are Name, Phone Number, Email, Status(Juicy perks, inactive, guest) and join date. An employee has a relationship with offer and profile as every customer will have a profile of his own. A customer also has a unary relationship with itself. Customers will be able to raise a Ticket and get it resolved by an employee.

An Application is related to profile, an employee and can be used for specific products such as Loan, Investment.

Transaction is also a very important entity that needs to be considered when we look at the flow of events in a bank. Related to the transactions we have to consider the date on which the transaction was performed as well as the type of the transaction. We also consider the amount of the transaction. A transaction can be carried out on one ATMs only at a time but an ATM has one or many transactions at a time.

Other important entities are Building and product. There are Loans and Investments.

Hence we can see that with all the relationships and entities in place , we have drawn an enhanced ER diagram. Taking a basis of the ER diagram, we have drawn a relational model.

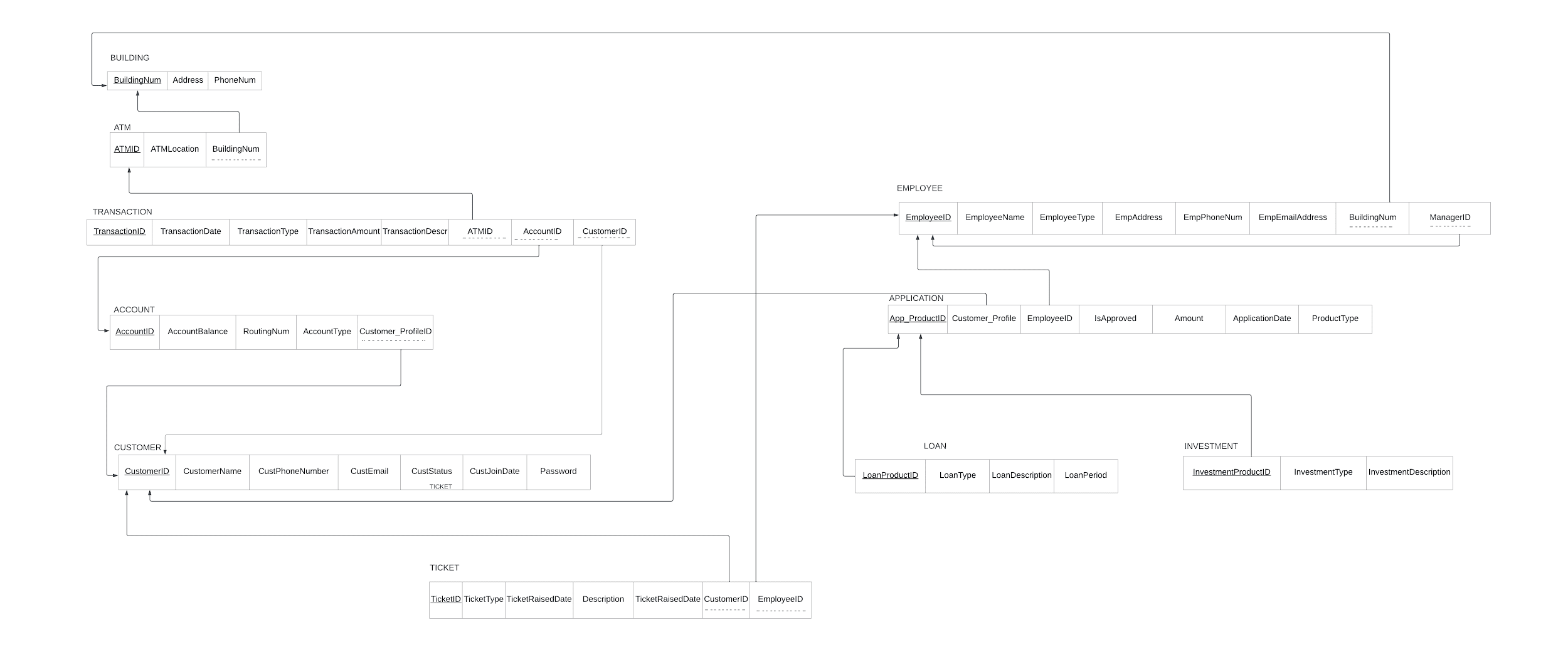
**RELATIONAL MODEL**

The Relational model helps us to organize data within a database and it provides a structure to model our system. Normalization helps in removing insertion, deletion and modification anomalies and hence minimizes redundancies by applying rules regarding functional and transitive dependencies.

Our Bearcat Bank Relational Model below has been normalized to the 3rd Normal Form (3NF) after removing the repeating groups and multivalued attributes, and defining the Primary keys in each relation. Next, we identified the Partial Dependencies for the relation with a composite primary key: Application and simplified it further to not have partial dependencies in any relation. For achieving the 3NF, we recognized the Transitive Dependencies where non-key attributes could be dependent on other non-key attributes, and we have removed such transitive dependencies to make it well structured for the database application of our system. This relational model simplifies our system in defining the key characteristics and relations to work with.

.

<https://lucid.app/documents/view/3625af41-3464-45dd-ac38-e2d181f0378e>



**DATA DICTIONARY**

The following tables show the physical design of our database. First, we

map each entity type in the ER diagram to a table. Then, we conduct the following

Denormalization:

1. Integrating Customer, Profile and Card information into the Customer table
   1. As there is 1:1 relationship between Customer and Profile, it makes sense to maintain

Profile information in the same context as Customer

* 1. There is 1:M relationship between Customer and Card and Card is not associated with

any other entity

1. Integrating Loan and Investment into Product table
   1. As Loan, Investment are products which can be applied by the Customer, we have

integrated the Loan and Investment table into Product table

As the Bearcat Bank is relatively small, we do not require Horizontal and Vertical partitioning as

information is addressed by the table structure

Our database (Bearcat\_Bank) includes the following 10 tables:

**CUSTOMER\_PROFILE**

This table records the information of Customers. The Indexes are CustomerID. This table is named as CustomerProfile in our database

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Field** | **Definition** | **Data Type** | **Extra Value Constraint** | **Not Null, Unique and default controls** | **PK or FK** |
| CustomerID | Identifier for customer | INT | None | Not Null,Unique | PK |
| CustomerName | Name of the customer | VARCHAR(30) | None | Not Null |  |
| Password | Password for a customer to login | VARCHAR(50) | None | Not Null |  |
| CustomerPhoneNumber | Phone Number of the customer | VARCHAR(50) | None | Not Null |  |
| CustomerEmail | Email of the customer | VARCHAR(50) | None | Not Null |  |
| CustomerStatus | Status of the customer | VARCHAR(50) | None | Not Null |  |
| CustomerProfileDate | Customer profile creation date | DATE | None | Not Null |  |
| CardType | Type of the card | VARCHAR(20) | CREDIT CARD or DEBIT CARD | Not Null |  |
| CardNumber | Card number mentioned | VARCHAR(50) | None | Not Null, Unique |  |
| CardExpiryDate | Card expiry date | DATE | MM/YYYY | Not Null |  |

**TRANSACTION**

This table records the information of transactions made by Customer Account or at the ATM. The Indexes are TransactionID, CustomerID, AccountID, ATMID. This tabled is named as tb\_Transaction in our database

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Field** | **Definition** | **Data Type** | **Extra Value Constraint** | **Not Null, Unique and default controls** | **PK or FK** |
| TransactionID | The identifier for a transaction | INT | None | Not Null,Unique,  Auto Increment | PK |
| TransactionDate | Date of the transaction | DATETIME | None | Not Null |  |
| TransactionType | Type of the transaction | VARCHAR(6) | “DEBIT” or “CREDIT” | Not Null |  |
| TransactionDescr | Description of the transaction that has taken place | VARCHAR(50) | None | Not Null |  |
| TransactionAmount | Amount of the transaction | DECIMAL(10,2) | None | Not Null |  |
| ATMID | Identifier of ATM making transaction | INT | None |  | FK |
| AccountID | Identifier of Account making transaction | INT | None | Not Null | FK |
| CustomerID | Identifier of Customer making transaction | INT | None | Not Null | FK |

**TICKET**

This table records the information of tickets raised. The indexes are TicketID, CustomerID, EmployeeID. This tabled is named as tb\_Ticket in our database

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Field** | **Definition** | **Data Type** | **Extra Value Constraint** | **Not Null, Unique and default controls** | **PK or FK** |
| TicketID | Identifier of a ticket raised | INT | None | Not Null,Unique | PK |
| TicketType | Type of the ticket raised | VARCHAR(30) | None | Not Null |  |
| TicketRaisedDate | Ticket created date | DATE | None | Not Null |  |
| Description | Description of the ticket raised | VARCHAR(300) | None | Not Null |  |
| TicketResolvedDate | Ticket resolution date | DATE | None | Not Null |  |
| CustomerID | Identifier of Customer who raised ticket | INT | None | Not Null | FK |
| EmployeeID | Identifier of Employee who resolved ticket | INT | None | Not Null | FK |

**EMPLOYEE**

This table records the information of Employees working. The Indexes are EmployeeID, ManagerID. This tabled is named as tb\_Employee in our database

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Field** | **Definition** | **Data Type** | **Extra Value Constraint** | **Not Null, Unique and default controls** | **PK or FK** |
| EmployeeID | The identifier of the bank employee | INT | None | Not Null,Unique | PK |
| EmployeeName | Name of the employee | VARCHAR(50) | None | Not Null |  |
| EmployeeType | Type of the bank employee | VARCHAR(50) | None | Not Null |  |
| EmpPhoneNum | Phone Number of the employee | INT | None | Not Null |  |
| EmpEmailAddress | Email Address of the employee | VARCHAR(70) | None | Not Null, Unique |  |
| BuildingNum | Identifier of Building where Employee works | INT | None | Not Null, Unique | FK |
| ManagerID | Identifier of Employee’s Manager | INT | None | Not Null | FK |

**APPLICATION**

This table records the information of Applications created. The indexes are CustomerID, ProductID, ApplicationDate. This tabled is named as tb\_Application in our database

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Field** | **Definition** | **Data Type** | **Extra Value Constraint** | **Not Null, Unique and default controls** | **PK or FK** |
| App\_ProductID | Identifier of Product associated with application | INT | None | Not Null | PK |
| CustomerID | Identifier of Customer Profile associated with Application | INT | None | Not Null | FK |
| ProductType | Type of the Product Applied for by customer | VARCHAR(1) | “L” or “I” | Not Null |  |
| EmployeeID | Identifier of Employee managing application | INT | None | Not Null | FK |
| ApplicationAmount | Amount for the application | DECIMAL(9,2) | None | Not Null |  |
| IsApproved | Whether an application is approved or not | SMALLINT | 0 or 1 | Not Null |  |
| ApplicationDate | Date of the application | DATE | None | Not Null |  |

**ACCOUNT**

This table records the information of Accounts of a customer. The indexes are AccountID, CustomerID. This tabled is named as tb\_Account in our database

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Field** | **Definition** | **Data Type** | **Extra Value Constraint** | **Not Null, Unique and default controls** | **PK or FK** |
| AccountID | Identifier of the amount | INT | None | Not Null,Unique | PK |
| AccountBalance | Balance in the account | INT | None | Not Null,Unique |  |
| RoutingNum | Routing number of the account | VARCHAR(20) | None | Not Null |  |
| AccountType | Type of the account | VARCHAR(50) | None | Not Null |  |
| CustomerID | Identifier of account customer | INT | None | Not Null | FK |

**BUILDING**

This table records the information of the Building of the bank. The indexes are BuildingNumber. This tabled is named as tb\_Building in our database

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Field** | **Definition** | **Data Type** | **Extra Value Constraint** | **Not Null, Unique and default controls** | **PK or FK** |
| BuildingNumber | Identifier of the building | INT | None | Not Null,Unique | PK |
| BuildingAddress | Address of the building | VARCHAR(50) | None | Not Null,Unique |  |
| BuildingPhoneNumber | Phone Number of the building | VARCHAR(50) | None | Not Null |  |

**ATM**

This table records the information of the ATMs managed by the bank. The indexes are ATMID. This tabled is named as tb\_ATM in our database

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Field** | **Definition** | **Data Type** | **Extra Value Constraint** | **Not Null, Unique and default controls** | **PK or FK** |
| ATMID | Identifier of the ATM | INT | None | Not Null | PK |
| ATMLocation | Location of the ATM | VARCHAR(50) | None | Not Null |  |
| BuildingNumber | Identifier of building where ATM may be stationed | INT | None |  | FK |

**LOAN**

This table records the information of Loans applied for by the Customer. The indexes are LoanID. This tabled is named as tb\_Loan in our database

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Field** | **Definition** | **Data Type** | **Extra Value Constraint** | **Not Null, Unique and default controls** | **PK or FK** |
| LoanID | Identifier of the product - loan application | INT | None | Not Null,Unique | PK, FK |
| LoanDescr | Name of the product | VARCHAR(100) | None | Not Null |  |
| LoanType | Type of the investment | VARCHAR(30) | None | Not Null |  |
| LoanInterest | Interest rate on loan | DECIMAL(5,2) | None | Not Null |  |
| LoanPeriod | Period for which the loan is taken | INT | Should be between 1 (Year) and 20 (Years) | Not Null |  |

**INVESTMENT**

This table records the information of Investments applied for by the Customer. The indexes are InvestmentID. This tabled is named as tb\_Investment in our database

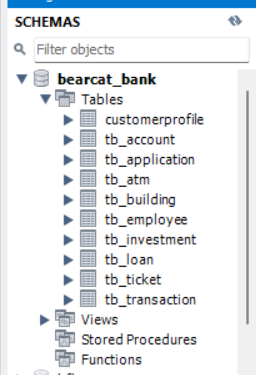
|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Field** | **Definition** | **Data Type** | **Extra Value Constraint** | **Not Null, Unique and default controls** | **PK or FK** |
| InvestmentID | Identifier of the product - investment application | INT | None | Not Null,Unique | PK, FK |
| InvestmentDescr | Name of the investment | VARCHAR(100) | None | Not Null |  |
| InvestmentType | Type of the investment | VARCHAR(30) | None | Not Null |  |
| InvestmentInterest | Interest rate on investment (current) | DECIMAL(5,2) | None | Not Null |  |

**DATA IMPLEMENTATION**

**Create and Use Bearcat Bank Database:**

Create SCHEMA Bearcat\_Bank;

USE Bearcat\_Bank;



**Create the Tables for Bearcat Bank:**

1. **Query for creating CustomerProfile**

Create Table CustomerProfile (

CustomerID INT AUTO\_INCREMENT PRIMARY KEY,

Password VARCHAR(50) NOT NULL,

CustomerName VARCHAR (30) NOT NULL,

CustomerPhoneNumber VARCHAR (50) NOT NULL,

CustomerEmail VARCHAR (50) NOT NULL,

CustomerStatus VARCHAR(50) NOT NULL,

CustomerProfileDate Date NOT NULL,

CardType VARCHAR(20),

CardNumber VARCHAR(50),

CardExpiryDate DATE

)AUTO\_INCREMENT=10000;

1. **Query for creating Account Table tb\_Account**

Create Table tb\_Account (

AccountID INT AUTO\_INCREMENT not null unique,

AccountBalance Int NOT NULL,

RountingNum VARCHAR (50) NOT NULL,

AccountType VARCHAR (50) NOT NULL,

CustomerID int NOT NULL,

Primary Key (AccountID),

foreign key (CustomerID) references CustomerProfile(CustomerID)

)AUTO\_INCREMENT=200000;

1. **Query for creating Employee Table tb\_Employee**

create table tb\_Employee (

EmployeeID INT AUTO\_INCREMENT PRIMARY KEY,

EmployeeName VARCHAR(50) NOT NULL,

EmployeeType VARCHAR(40) NOT NULL,

EmpPhoneNum INT NOT NULL,

EmpEmailAddress VARCHAR(70) NOT NULL,

BuildingNumber INT NOT NULL,

ManagerID INT,

FOREIGN KEY(ManagerID) References tb\_Employee(EmployeeID)

)AUTO\_INCREMENT=5000;

1. **Query for creating Application Table tb\_Application**

Create table tb\_Application (

App\_ProductID INT AUTO\_INCREMENT PRIMARY KEY,

CustomerID INT NOT NULL,

ProductType VARCHAR(1) NOT NULL,

EmployeeID INT NOT NULL,

ApplicationAmount DECIMAL(9,2) NOT NULL,

IsApproved BOOLEAN NOT NULL,

ApplicationDate DATE NOT NULL,

FOREIGN KEY(CustomerID) References CustomerProfile(CustomerID),

FOREIGN KEY (EmployeeID) References tb\_Employee(EmployeeID)

);

1. **Query for creating Building Table tb\_Building**

CREATE TABLE tb\_Building (

BuildingNumber INT PRIMARY KEY,

BuildingAddress VARCHAR(50) NOT NULL,

BuildingPhoneNumber VARCHAR(50) NOT NULL

);

1. **Query for creating ATM Table tb\_ATM**

CREATE TABLE tb\_ATM (

ATMID INT PRIMARY KEY,

ATMLocation VARCHAR(50) NOT NULL,

BuildingNum INT,

FOREIGN KEY(BuildingNum) REFERENCES tb\_Building(BuildingNumber)

);

1. **Query for creating Loan Table tb\_Loan**

CREATE TABLE tb\_Loan (

LoanID INT PRIMARY KEY,

LoanDescr VARCHAR(100) NOT NULL,

LoanType VARCHAR(30) NOT NULL,

LoanInterest DECIMAL(5,2) NOT NULL,

LoanPeriod INT NOT NULL,

CHECK (LoanPeriod >= 1 AND LoanPeriod <= 20),

FOREIGN KEY (LoanID) REFERENCES tb\_Application(App\_ProductID)

);

-- DROP TABLE tb\_Loan;

1. **Query for creating Investment Table tb\_Investment**

CREATE TABLE tb\_Investment (

InvestmentID INT PRIMARY KEY,

InvestmentDescr VARCHAR(100) NOT NULL,

InvestmentType VARCHAR(30) NOT NULL,

InvestmentInterest DECIMAL(5,2) NOT NULL,

FOREIGN KEY (InvestmentID) REFERENCES tb\_Application(App\_ProductID)

);

1. **Query for creating Transaction Table tb\_Transaction**

Create Table tb\_Transaction (

TransactionID INT AUTO\_INCREMENT,

TransactionDate DATE,

TransactionType VARCHAR (6) NOT NULL,

TransactionDescr VARCHAR(50) NOT NULL,

TransactionAmount DECIMAL (10,2) NOT NULL,

ATMID INT NOT NULL,

AccountID INT NOT NULL,

CustomerID INT NOT NULL,

Primary Key (TransactionID),

foreign key (ATMID) references tb\_ATM(ATMID),

foreign key (AccountID) references tb\_Account(AccountID),

foreign key (CustomerID) references CustomerProfile(CustomerID)

)AUTO\_INCREMENT=500;

-- DROP TABLE tb\_Transaction;

1. **Query for creating Ticket Table tb\_Ticket**

CREATE TABLE tb\_Ticket (

TicketID INT AUTO\_INCREMENT Primary key,

TicketType Varchar(30) NOT NULL,

TicketRaisedDate Date,

Description Varchar(300) NOT NULL,

TicketResolvedDate Date,

CustomerID INT NOT NULL,

EmployeeID INT NOT NULL,

FOREIGN KEY (CustomerID) REFERENCES CustomerProfile(CustomerID),

FOREIGN KEY (EmployeeID) REFERENCES tb\_Employee(EmployeeID)

);

**Load (Insert) the data in CustomerProfile**

INSERT INTO CustomerProfile (Password, CustomerName, CustomerPhoneNumber,

CustomerEmail, CustomerStatus, CustomerProfileDate, CardType, CardNumber, CardExpiryDate)

VALUES

('Teams@7', 'Shrinidhi Shetty','5139697585','shri@mail.uc.edu','active','2020-01-05','Credit','0421-4321-4982-3233', '2023-12-01'),

('Teams@8', 'Radhika Adwant','5138642165','radh@mail.uc.edu','not active','2020-01-05','Debit','9538-3921-4402-5323', '2023-12-01'),

('Teams@9', 'Andrew Smith','5139875565','andre@mail.uc.edu','active','2020-01-05','Debit','9938-1010-4444-5373', '2023-12-01'),

('Teams@10', 'alexander duncan','5139997865','mahajasl@mail.uc.edu','active','2020-01-05','Debit','9938-3321-4402-5033', '2023-12-01'),

('Teams@11', 'priyanka patil','5139997865','mahajasl@mail.uc.edu','active','2020-01-05','Debit','9938-3321-4402-5033', '2023-12-01'),

('Teams@12', 'Snehal Mahajan','5139997865','mahajasl@mail.uc.edu','active','2020-01-05','Debit','9938-3321-4402-5033', '2024-12-01'),

('Teams@13', 'Apurva shriwastwa','5139752665','apurva@mail.uc.edu','not active','2020-01-05','Credit','9668-3431-4402-5521', '2024-12-01'),

('Teams@14', 'Arun Kumar','513654485','arun@mail.uc.edu','active','2020-01-05','Debit','6758-1111-4402-5033', '2024-12-01'),

('Teams@15', 'Snehal patil','5139997865','snepatil@mail.uc.edu','active','2020-01-05','Debit','9938-3321-4402-5033', '2024-12-01'),

('Teams@16', 'Snehal lokhande','5139997865','snehal@mail.uc.edu','active','2020-01-05','Debit','9938-3321-4402-5033', '2024-12-01'),

('Teams@17', 'yanran liu','5139997865','yanran@mail.uc.edu','active','2020-01-05','Credit','9938-3121-4642-5033', '2025-12-01'),

('Teams@18', 'Krunal shivagan','5132227675','shiva@mail.uc.edu','not active','2020-01-05','Debit','5432-7777-5422-5033', '2025-12-01'),

('Teams@19', 'Ritika Mahajan','5132215865','ritika@mail.uc.edu','active','2020-01-05','Debit','9938-3663-2402-5033', '2025-12-01'),

('Teams@20', 'Ritu Mahajan ','5133927165','ritu@mail.uc.edu','active','2020-01-05','Debit','9938-3321-4402-5033', '2025-12-01'),

('Teams@21', 'Yash Karve','5138069465','yash@mail.uc.edu','active','2020-01-05','Debit','9938-3321-4402-5033', '2026-12-01'),

('Teams@22', 'Dhruval Cheda','513234265','dhruval@mail.uc.edu','active','2020-01-05','Credit','9938-3321-4402-5033', '2026-12-01'),

('Teams@23', 'Sheeba Shetty','513092865','sheeba@mail.uc.edu','active','2020-01-05','Debit','9938-3321-4402-5033', '2026-12-01'),

('Teams@24', 'Ramesh Patil','5130958665','ramesh@mail.uc.edu','active','2020-01-05','Credit','9938-3321-4402-5033', '2026-12-01'),

('Teams@25', 'Rohit Bhoite','5134396865','rohit@mail.uc.edu','active','2020-01-05','Debit','9938-3321-4402-5033', '2024-12-01'),

('Teams@26', 'Archana Shetty','5148377585','arc@mail.uc.edu','active','2020-01-05','Credit','0421-4321-45743-3233', '2024-12-01');

**Load (Insert) the data in Employee Table**

insert into tb\_Employee (EmployeeName, EmployeeType, EmpPhoneNum, EmpEmailAddress,BuildingNumber,ManagerID)

values ('Snehal', 'Manager', '583233223', 'sneh@mail.uc.edu', 101, 5000),

('Kabira', 'Clerk', '583233223', 'sneh@mail.uc.edu', 101,5000),

('Julie', 'Manager', '583233223', 'sneh@mail.uc.edu', 101,5000),

('Thanmayee', 'Clerk', '583233223', 'sneh@mail.uc.edu', 101, 5000),

('Simran', 'Clerk', '583233223', 'sneh@mail.uc.edu', 101,5000),

('Apurva', 'Clerk', '583233223', 'sneh@mail.uc.edu', 101,5000),

('Priyanka', 'Clerk', '583233223', 'sneh@mail.uc.edu', 101,5000),

('Varsha', 'Clerk', '583233223', 'sneh@mail.uc.edu', 101,5000),

('Krishnaja', 'Clerk', '583233223', 'sneh@mail.uc.edu', 101,5000),

('Reshma', 'Clerk', '583233223', 'sneh@mail.uc.edu', 101,5000),

('Bob', 'Clerk', '583233223', 'sneh@mail.uc.edu', 101,5000),

('Marley', 'Clerk', '583233223', 'sneh@mail.uc.edu', 101,5000),

('Tim', 'Clerk', '583233223', 'sneh@mail.uc.edu', 101,5000),

('Adam', 'Clerk', '583233223', 'sneh@mail.uc.edu', 101,5000),

('Jack', 'Clerk', '583233223', 'sneh@mail.uc.edu', 101,5000),

('John', 'Clerk', '583233223', 'sneh@mail.uc.edu', 101,5000),

('Justin', 'Clerk', '583233223', 'sneh@mail.uc.edu', 101,5000),

('Tina', 'Clerk', '583233223', 'sneh@mail.uc.edu', 101,5000),

('Sita', 'Clerk', '583233223', 'sneh@mail.uc.edu', 101,5000),

('Meena', 'Clerk', '583233223', 'sneh@mail.uc.edu', 101,5000),

('Radha', 'Clerk', '583233223', 'sneh@mail.uc.edu', 101,5000);

SELECT \* from CustomerProfile;

INSERT INTO tb\_Account (AccountID, AccountBalance, RountingNum, AccountType, CustomerID)

Values

(200000, 2229218, '444444', 'Savings', 10000),

(200001, 39402892, '333333', 'Checking', 10001),

(200002, 939232, '111111', 'Savings', 10002),

(200003, 9281098, '222222', 'Savings', 10003),

(200004, 92392, '555555', 'Savings', 10004),

(200005, 23131, '666666', 'Checking', 10019),

(200006, 2322022, '777777', 'Checking', 10005),

(200007, 3443322, '773322', 'Checking', 10006),

(200008, 3424222, '336688', 'Checking', 10007),

(200009, 555533, '112222', 'Checking', 10008),

(200010, 19282, '553333', 'Savings', 10009),

(200011, 3827271, '998888', 'Savings', 10010),

(200012, 857574, '889999', 'Savings', 10011),

(200013, 28281, '778888', 'Savings', 10012),

(200014, 958573, '667777', 'Checking', 10013),

(200015, 1183722, '556666', 'Checking', 10014),

(200016, 756462, '445555', 'Checking', 10015),

(200017, 2828364, '334444', 'Checking', 10016),

(200018, 985743, '223333', 'Checking', 10017),

(200019, 129372, '112222', 'Savings', 10018);

SELECT \* from tb\_Account;

INSERT INTO tb\_Building(BuildingNumber, BuildingAddress, BuildingPhoneNumber) values(101, "Clifton Ave",5139990000);

INSERT INTO tb\_Application (CustomerID, ProductType, EmployeeID, ApplicationAmount, IsApproved, ApplicationDate)

values (10001, 'L', 5001, 200.55, 0,'2022-11-20'),

(10002, 'L', 5002, 2000, 0,'2022-11-21'),

(10003, 'L', 5002, 1700, 0,'2022-11-22'),

(10004, 'L', 5002, 500, 0,'2022-11-23'),

(10005, 'L', 5003, 3000, 0,'2022-11-24'),

(10016, 'I', 5009, 200, 0,'2022-11-25'),

(10017, 'I', 5009, 300, 0,'2022-11-26'),

(10018, 'I', 5009, 400, 0,'2022-11-27'),

(10019, 'I', 5009, 500, 0,'2022-11-28'),

(10011, 'I', 5009, 250, 0,'2022-11-29'),

(10006, 'L', 5004, 4000, 0,'2022-11-30'),

(10007, 'L', 5005, 2000, 0,'2022-11-30'),

(10008, 'L', 5006, 5000, 0,'2022-12-01'),

(10009, 'L', 5007, 3000, 0,'2022-12-01'),

(10010, 'L', 5008, 2000, 0,'2022-12-01'),

(10012, 'I', 5009, 400, 0,'2022-11-01'),

(10013, 'I', 5009, 1000, 0,'2022-11-02'),

(10014, 'I', 5009, 800, 0,'2022-11-03'),

(10015, 'I', 5009, 900, 0,'2022-11-04'),

(10000, 'I', 5009, 1200, 0,'2022-11-05');

INSERT INTO tb\_Loan(LoanID, LoanDescr, LoanType, LoanInterest, LoanPeriod) values(1, "Car Loan taken with a loan period of 10 years", "Car Loan", 10.20, 1),

(2, "Car Loan taken with a loan period of 10 years", "Car Loan", 10.20, 2),

(3, "Car Loan taken with a loan period of 10 years", "Car Loan", 10.20, 3),

(4, "Car Loan taken with a loan period of 10 years", "Car Loan", 10.20, 4),

(5, "Car Loan taken with a loan period of 10 years", "Car Loan", 10.20, 5),

(11, "Car Loan taken with a loan period of 10 years", "Car Loan", 10.20, 4),

(12, "Car Loan taken with a loan period of 10 years", "Car Loan", 10.20, 2),

(13, "Car Loan taken with a loan period of 10 years", "Car Loan", 10.20, 3),

(14, "Car Loan taken with a loan period of 10 years", "Car Loan", 10.20, 1),

(15, "Car Loan taken with a loan period of 10 years", "Car Loan", 10.20, 10);

INSERT INTO tb\_Investment(InvestmentID, InvestmentDescr, InvestmentType, InvestmentInterest) values(6, "Mutual Fnds Investment", "Mutual Funds", 10.20),

(7, "Mutual Fnds Investment", "Mutual Funds", 10.20),

(8, "Mutual Fnds Investment", "Mutual Funds", 10.20),

(9, "Mutual Fnds Investment", "Mutual Funds", 10.20),

(10, "Sysematic Inv Plan Investment", "SIP", 10.20),

(16, "Sysematic Inv Plan Investment", "SIP", 10.20),

(17, "Sysematic Inv Plan Investment", "SIP", 10.20),

(18, "Securitites", "Securitites", 10.20),

(19, "Bonds", "Bonds", 10.20),

(20, "Term Inv", "Term Investment", 10.20);

INSERT INTO tb\_ATM(ATMID, ATMLocation, BuildingNum) values(200, "Clifton Ave 1st Floor",101),

(201, "Clifton Ave UC",101),

(202, "Clifton Ave 2nd Floor",101),

(203, "Clifton Ave 3rd Floor",101),

(204, "Clifton Ave 4th Floor",101),

(205, "Clifton Ave 5th Floor",101),

(206, "Clifton Ave 6th Floor",101),

(207, "Burnett Woods", null),

(208, "Blue Ash", null),

(209, "Riddle Road", null),

(210, "Thompson Heights", null),

(211, "Mason", null),

(212, "Calhoun Street", null),

(213, "McMillan Street", null),

(214, "Campus Green Garage", null),

(215, "Jefferson Avenue", null),

(216, "Vine Street", null),

(217, "Probasco", null),

(218, "Stratford Avenue", null),

(219, "Ludlow", null);

INSERT INTO tb\_Transaction (TransactionDate, TransactionType, TransactionDescr, TransactionAmount, ATMID, AccountID, CustomerID)

Values

(01/02/2022, 'Debit', 'Check','30828.2',200,200000,10000),

(01/02/2022, 'Credit', 'Check','305352',201, 200001,10001),

(01/02/2022, 'Debit', 'Direct deposit','21442',202, 200002,10002),

(01/02/2022, 'Credit', 'Direct deposit','100',203, 200003,10003),

(01/02/2022, 'Debit', 'Check','10837.87',200,200004, 10004),

(01/02/2022, 'Debit', 'Check','44737',206,200005, 10005),

(01/02/2022, 'Debit', 'Direct deposit','10101',207, 200006,10006),

(01/02/2022, 'Credit', 'Check','111303',209,200007, 10007),

(01/02/2022, 'Debit', 'Check','54838',210,200008, 10008),

(01/02/2022, 'Credit', 'Check','31132',201,200009, 10009),

(01/02/2022, 'Debit', 'Check','90901',202,200010, 10010),

(01/02/2022, 'Debit', 'Direct deposit','330494',203, 200011,10011),

(01/02/2022, 'Credit', 'Check','11132',204, 200012,10012),

(01/02/2022, 'Debit', 'Direct deposit','90899',205, 200013,10013),

(01/02/2022, 'Debit', 'Direct deposit','30828.2',201, 200014,10014),

(01/02/2022, 'Debit', 'Direct deposit','30828.2',202, 200015,10015),

(01/02/2022, 'Debit', 'Check','30828.2',210, 200016,10016),

(01/02/2022, 'Credit', 'Direct deposit','30828.2',211, 200017,10017),

(01/02/2022, 'Debit', 'Direct deposit','22121',212, 200018,10018),

(01/02/2022, 'Credit', 'Check','7592',213, 200019,10019);

INSERT INTO tb\_Ticket (TicketID, TicketType, TicketRaisedDate, Description, TicketResolvedDate, CustomerID, EmployeeID)

Values (1, 'InternetBanking', '2022-09-13', 'Ticket for Internet Banking', '2022-09-13',10000,5000),

(2, 'ATM', '2022-09-17', 'Ticket for ATM Issue', '2022-09-19',10001,5001),

(3, 'Application', '2022-09-24', 'Ticket for Application Issue', '2022-09-25',10002,5002),

(4, 'Transaction', '2022-09-26', 'Ticket for Transaction Issue', '2022-09-26',10003,5003),

(5, 'Account', '2022-09-27', 'Ticket for Account Issue', '2022-09-28',10004,5004),

(6, 'Maintenance', '2022-09-28', 'Ticket for Maintenance Issue', '2022-09-28',10005,5005),

(7, 'BranchChange', '2022-09-29', 'Ticket for BranchChange', '2022-09-30',10006,5006),

(8, 'DemandDraft', '2022-09-30', 'Ticket for DemandDraft Issue', '2022-09-30',10007,5007),

(9, 'Loan', '2022-10-02', 'Ticket for Loan Issue', '2022-10-05',10008,5008),

(10, 'AddressChange', '2022-10-07', 'Ticket for AddressChange', '2022-10-07',10009,5009),

(11, 'Application', '2022-10-09', 'Ticket for Application Issue', '2022-10-11',10010,5010),

(12, 'Account', '2022-10-12', 'Ticket for Account Issue', '2022-10-13',10011,5011),

(13, 'Transaction', '2022-10-15', 'Ticket for Transaction Issue', '2022-10-17',10012,5012),

(14, 'ATM', '2022-10-19', 'Ticket for ATM Issue', '2022-10-21',10013,5013),

(15, 'Maintenance', '2022-10-22', 'Ticket for Maintenance Issue', '2022-10-25',10014,5014),

(16, 'Loan', '2022-10-27', 'Ticket for Loan Issue', '2022-10-29',10015,5015),

(17, 'NameChange', '2022-10-28', 'Ticket for NameChange', '2022-10-29',10016,5016),

(18, 'Cheque', '2022-10-30', 'Ticket for Cheque Issue', '2022-10-31',10017,5017),

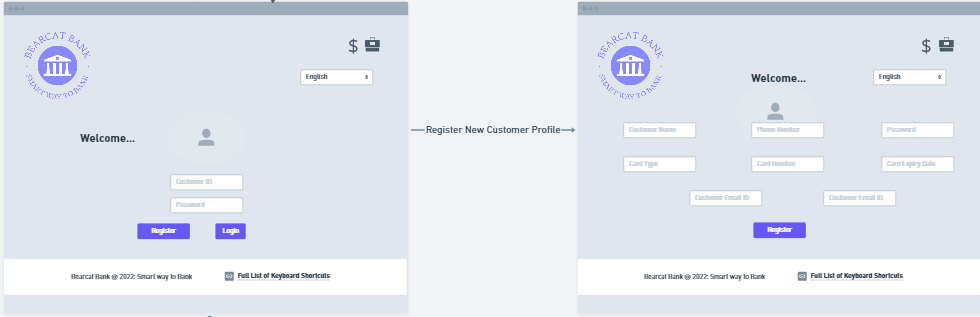
(19, 'Loan', '2022-11-01', 'Ticket for Loan Issue', '2022-11-05',10018,5018),

(20, 'InternetBanking', '2022-11-08', 'Ticket for Internet Banking Issue', '2022-11-12',10019,5019);

**Search information and execute queries as per user requirements on Bearcat Bank screens:**

**CUSTOMER SCREENS:**

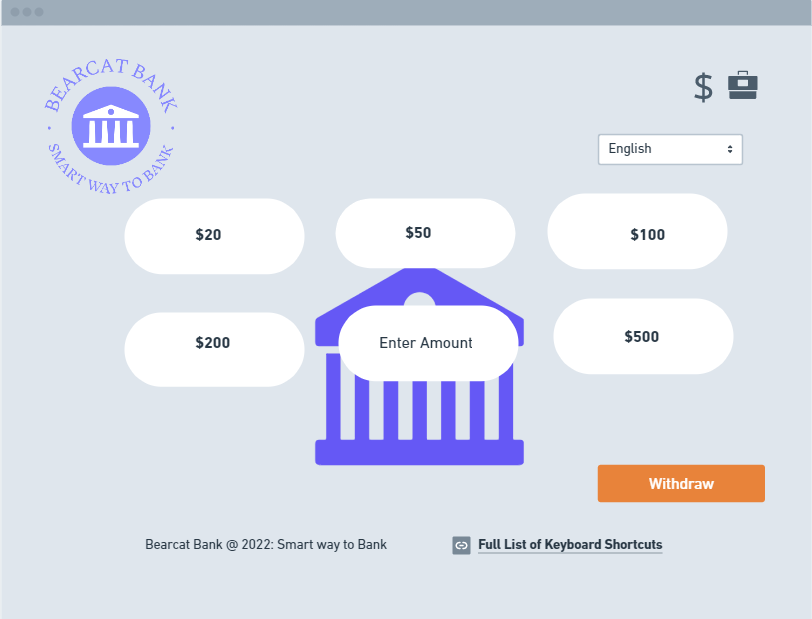
-- CUSTOMER SCREENS



-- Register (New Customer Registration)

INSERT INTO CustomerProfile (Password, CustomerName, CustomerPhoneNumber, CustomerEmail, CustomerStatus, CustomerProfileDate, CardType, CardNumber, CardExpiryDate)

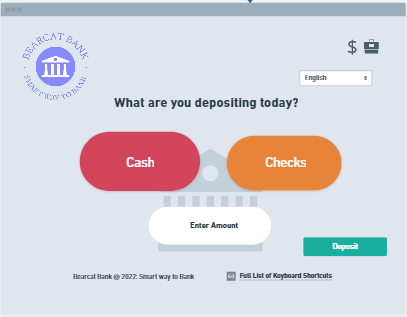
VALUES ('Teams@New','Shrinidhi Shetty','5139697585','shri@mail.uc.edu','active','2020-01-10','Credit','0439-0290-0900-7654', '2026-12-01');



-- Withdraw cash at ATM

INSERT INTO tb\_Transaction (TransactionDate, TransactionType, TransactionDescr, TransactionAmount, ATMID, AccountID, CustomerID)

Values ('2022-11-30', 'Credit', 'Cash withdrawal', 5000, 200, 200001,10001);



-- Cash Direct deposit at ATM

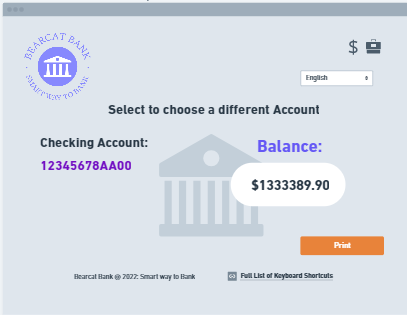
INSERT INTO tb\_Transaction (TransactionDate, TransactionType, TransactionDescr, TransactionAmount, ATMID, AccountID, CustomerID)

Values ('2022-11-30', 'Debit', 'Direct deposit', 2000, 200, 200001,10001);

-- Check deposit at ATM

INSERT INTO tb\_Transaction (TransactionDate, TransactionType, TransactionDescr, TransactionAmount, ATMID, AccountID, CustomerID)

Values ('2022-11-30', 'Debit', 'Check', 2000, 200, 200001,10001);



-- View Balance at ATM and on Customer Portal for Customer

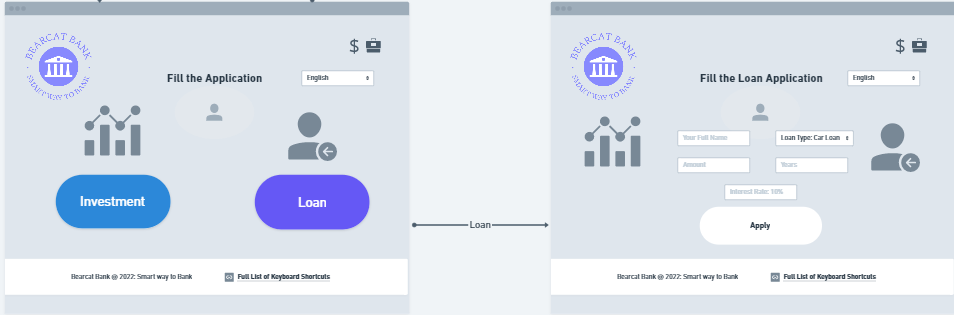
SELECT CustomerProfile.CustomerName, tb\_Account.CustomerID, tb\_Account.AccountType as AccountType, tb\_Account.AccountID as AccountID,

(SUM(CASE WHEN tb\_Transaction.TransactionType = 'DEBIT' THEN tb\_Transaction.TransactionAmount ELSE 0 END) -

SUM(CASE WHEN tb\_Transaction.TransactionType = 'CREDIT' THEN tb\_Transaction.TransactionAmount ELSE 0 END)) AS Balance

FROM tb\_Account INNER JOIN tb\_Transaction ON tb\_Account.CustomerID = tb\_Transaction.CustomerID INNER JOIN CustomerProfile ON

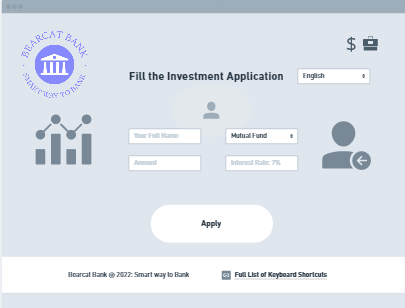
tb\_Account.CustomerID = CustomerProfile.CustomerID;



-- Apply Loan at ATM and on Customer Portal

INSERT INTO tb\_Application (CustomerID, ProductType, EmployeeID, ApplicationAmount, IsApproved, ApplicationDate)values ('10003', 'L', '5001', '58827', 0,'2022-11-30') ;

INSERT INTO tb\_Loan(LoanID, LoanDescr, LoanType, LoanInterest, LoanPeriod) values(21, "Car Loan taken with a loan period of 10 years", "Car Loan", 10.20, 5);



-- Apply Investment at ATM and on Customer Portal

INSERT INTO tb\_Application (CustomerID, ProductType, EmployeeID, ApplicationAmount, IsApproved, ApplicationDate)values ('10003', 'I', '5001', '58827', 0,'2022-11-30') ;

INSERT INTO tb\_Investment(InvestmentID, InvestmentDescr, InvestmentType, InvestmentInterest) values (22, "Mutual Fnds Investment", "Mutual Funds", 7.50);

-- View CustomerProfile

select \* from CustomerProfile;

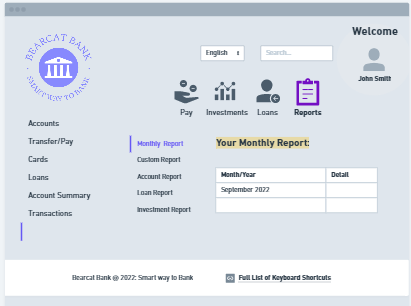
-- Display the first screen after logging into Customer Portal:

select \* from CustomerProfile where CustomerID=Login;

select \* from tb\_Investment where CustomerID=Login;

select \* from tb\_Loans where CustomerID=Login;

select \* from tb\_Card where CustomerID=Login;



-- Group Data by Year and Quarter from the transactions table for Analysis of Trends

SELECT EXTRACT(YEAR FROM TransactionDate) AS Year,

EXTRACT(QUARTER FROM TransactionDate) AS Quarter,

COUNT(TransactionAmount) AS Number\_of\_Transactions

FROM tb\_Transaction

GROUP BY EXTRACT(YEAR FROM TransactionDate), EXTRACT(QUARTER FROM TransactionDate)

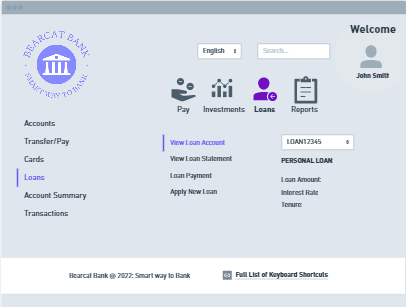
ORDER BY EXTRACT(YEAR FROM TransactionDate) ASC, EXTRACT(QUARTER FROM TransactionDate);

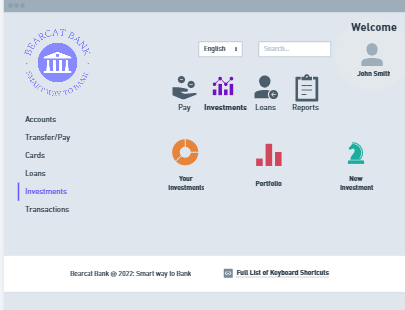
-- View Customer Transactions

Select T.TransactionID, t.TransactionDate, t.TransactionDescr, T.TransactionAmount, T.AccountID, C.CustomerID, C.CustomerName

FROM tb\_Transaction T, tb\_Account A, CustomerProfile C

WHERE T.AccountID = A.AccountID AND A.CustomerID = C.CustomerID AND C.CustomerID = 10001;





-- View Approved Loans

Select L.\*, A.CustomerID, A.ApplicationDate, A.isApproved FROM tb\_Application A, tb\_Loan L

WHERE A.App\_ProductID = L.LoanID AND A.CustomerID = 10001 AND A.isApproved = 1 AND A.ProductType = 'L';

-- View Approved Investments

Select I.\*, A.CustomerID, A.ApplicationDate, A.isApproved FROM tb\_Application A, tb\_Investment I

WHERE A.App\_ProductID = I.InvestmentID AND CustomerID = 10016 AND isApproved = 1 AND A.ProductType = 'I';

**STAFF SCREENS:**

****

-- Add Customer

INSERT INTO CustomerProfile (Password, CustomerName, CustomerPhoneNumber, CustomerEmail, CustomerStatus, CustomerProfileDate, CardType, CardNumber, CardExpiryDate)

VALUES ('Teams@NewCust','Blake Lively','5139697585','blake@mail.uc.edu','active','2020-01-10','Credit','0439-0290-0900-7654', '2026-12-01');



-- Delete Customer

DELETE FROM CustomerProfile WHERE CustomerID = '10001';



-- Update Customer Details (here updating CustomerStatus)

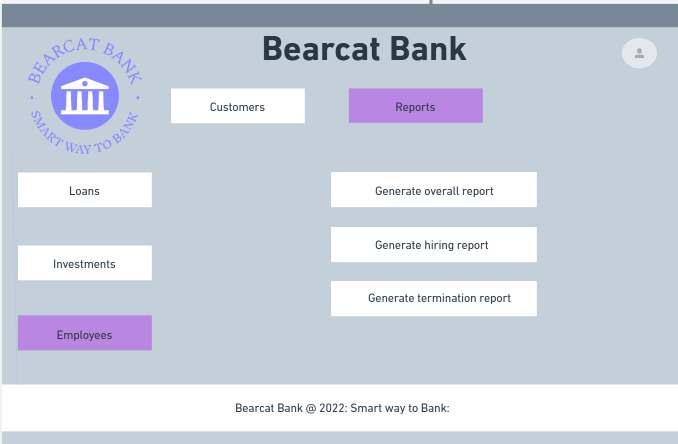
UPDATE CustomerProfile SET CustomerStatus = 'inactive' WHERE CustomerID = '10019';

-- Generate Loan Report

select \* from tb\_Loan where LoanID IS NOT NULL;

-- Generate Investment Report

select \* from tb\_Investment where InvestmentID IS NOT NULL;



-- Generate Employees Report working in the building 101

Create View EmpReport as

select \* from tb\_Employee

inner join tb\_Building

on tb\_Employee.BuildingNumber = tb\_Building.BuildingNumber;

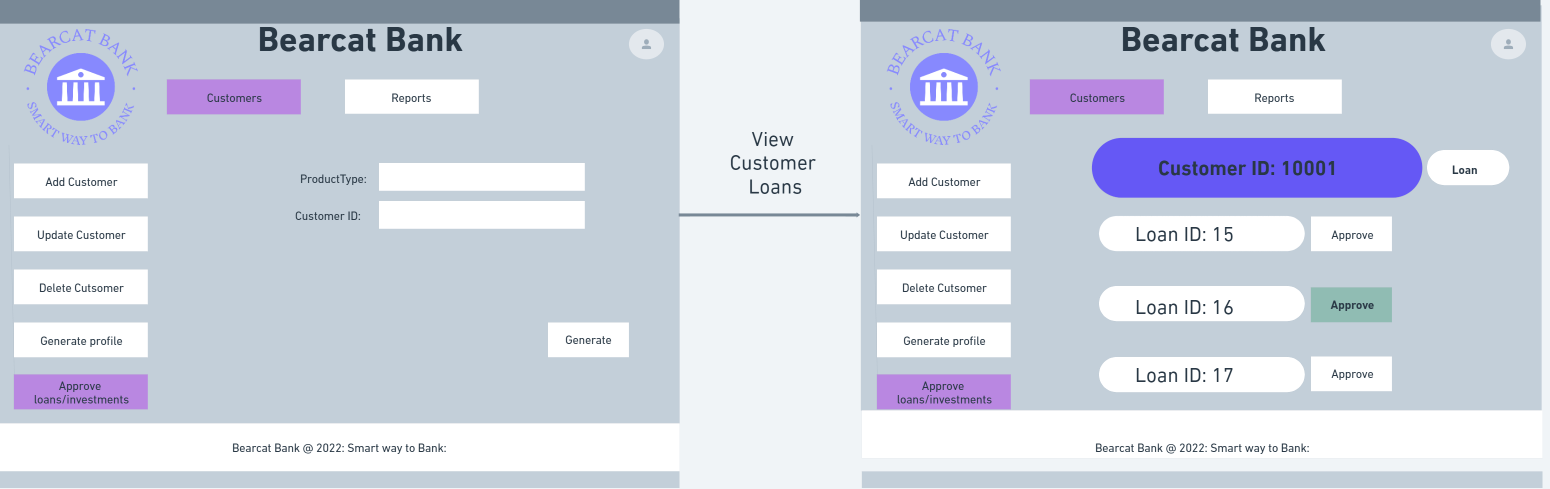
Select \* from EmpReport;

-- View Staff details managed by a particular Manager

SELECT M.EmployeeName as ManagerName, E.EmployeeName as EmployeeName, E.EmployeeID as EmployeeID, E.EmpPhoneNum as 'Employee Contact'

FROM tb\_Employee E, tb\_Employee M

WHERE E.ManagerID = M.EmployeeID;



-- Approve Loan for Customer ID 10001

select \* from tb\_Application where CustomerID = 10001 and ProductType = 'L';

update tb\_Application

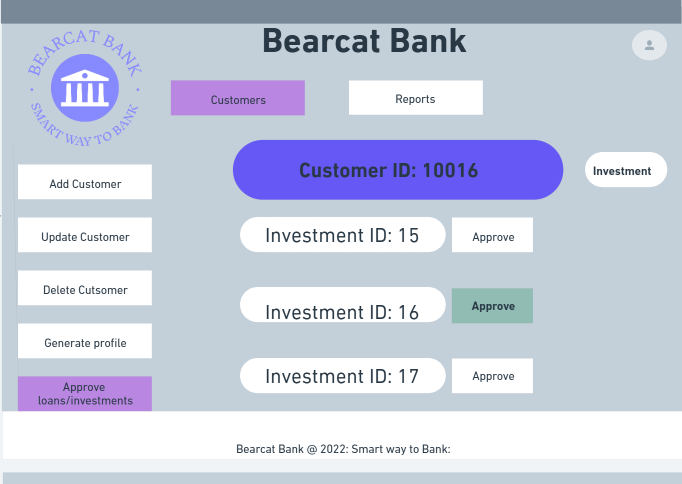
set isApproved = 1

where ProductType = 'L' and CustomerID = 10001 AND App\_ProductID = 1;

-- View Approved Loans for a customer

Select L.\*, A.CustomerID, A.ApplicationDate, A.isApproved FROM tb\_Application A, tb\_Loan L

WHERE A.App\_ProductID = L.LoanID AND A.CustomerID = 10001 AND A.isApproved = 1 AND A.ProductType = 'L';



-- Approve Investment for Customer

select \* from tb\_Application where CustomerID = 10016 and ProductType = 'I';

update tb\_Application

set isApproved = 1

where ProductType = 'I' and CustomerID = 10016 AND App\_ProductID = 6;

-- View Approved Investments for a customer

Select I.\*, A.CustomerID, A.ApplicationDate, A.isApproved FROM tb\_Application A, tb\_Investment I

WHERE A.App\_ProductID = I.InvestmentID AND CustomerID = 10016 AND isApproved = 1 AND A.ProductType = 'I';

\*\*\*