Republic of the Philippines

**University of Southeastern Philippines**

Obrero, Davao City

**College of Information and Computing**

| | | |
|---|---|---|
| System | : | Personal Loan Management System |
| Company/Office | : | AB Finance Corporation |
| Project Stage | : | Final Stage |
| Date | : | December 17, 2020 |
| Subject | : | CS 213 – Database Management System |
| Group Name | : | Team 007 |
| Group Members | : | Kenith S. Lumantas – 2019-00260<br>Cyrelle John A. Domingo – 2019-00397 (Team Leader)<br>Edgardo V. Amigo – 2019-00409 |

# I. REQUIREMENTS SPECIFICATION

## 1.1 Overview

With the rapid growth of technology, people's daily lives became more convenient than before. A system created by a group of programmers and designers helps many people starting from small businesses until the largest company in the world. However, there is still some field of works that needs a better system to help them. The database designers propose a personal loan management system that will help AB Finance Corporation and many lending companies to enhance their business and transform their manual system to a paperless one.

AB Finance Corporation is a Lending Company that offers personal loans at Juan Dela Cruz, Toril, Davao City. Eugene Quirino started the corporation on November 11, 2005, which has grown into multiple branches, such as in Talomo and Buhangin.

The type of organization designers proposed the system to is a corporation. The line of business would be a part of the economic sector that provides services, primarily financial. The corporation focuses on commercial banking, particularly on personal loans. A personal loan is an unsecured loan and is based solely on the Customer's creditworthiness, unlike a secured loan that is based on the security or collateral of the customer.

## 1.2 Existing Information Systems

Due to the confidentiality agreement's limitations, limited information is to be disclosed, such as the number of staff in a branch and other existing used by the corporation. However, describing a little of the system is allowed. The existing system is manual and does not maintain proper security details, and cannot track details easily. The existing system does not contain fast retrieval of information, which helps generate different reports needed for the business.

## 1.3 The Proposed System

The database designers plan to develop a system that store, update, and delete customers' data. Track all customer repayments, loan details, interest rate, lending records by the click of a mouse. Also, Provides a facility to generate reports quickly.

The proposed system's objective is to transform the manual system into a paperless system to reduce workload. Increase the decision process and consistency. Increase security as there are different dashboards or views for each position and increase communication and information access between offices.

### Product Functions

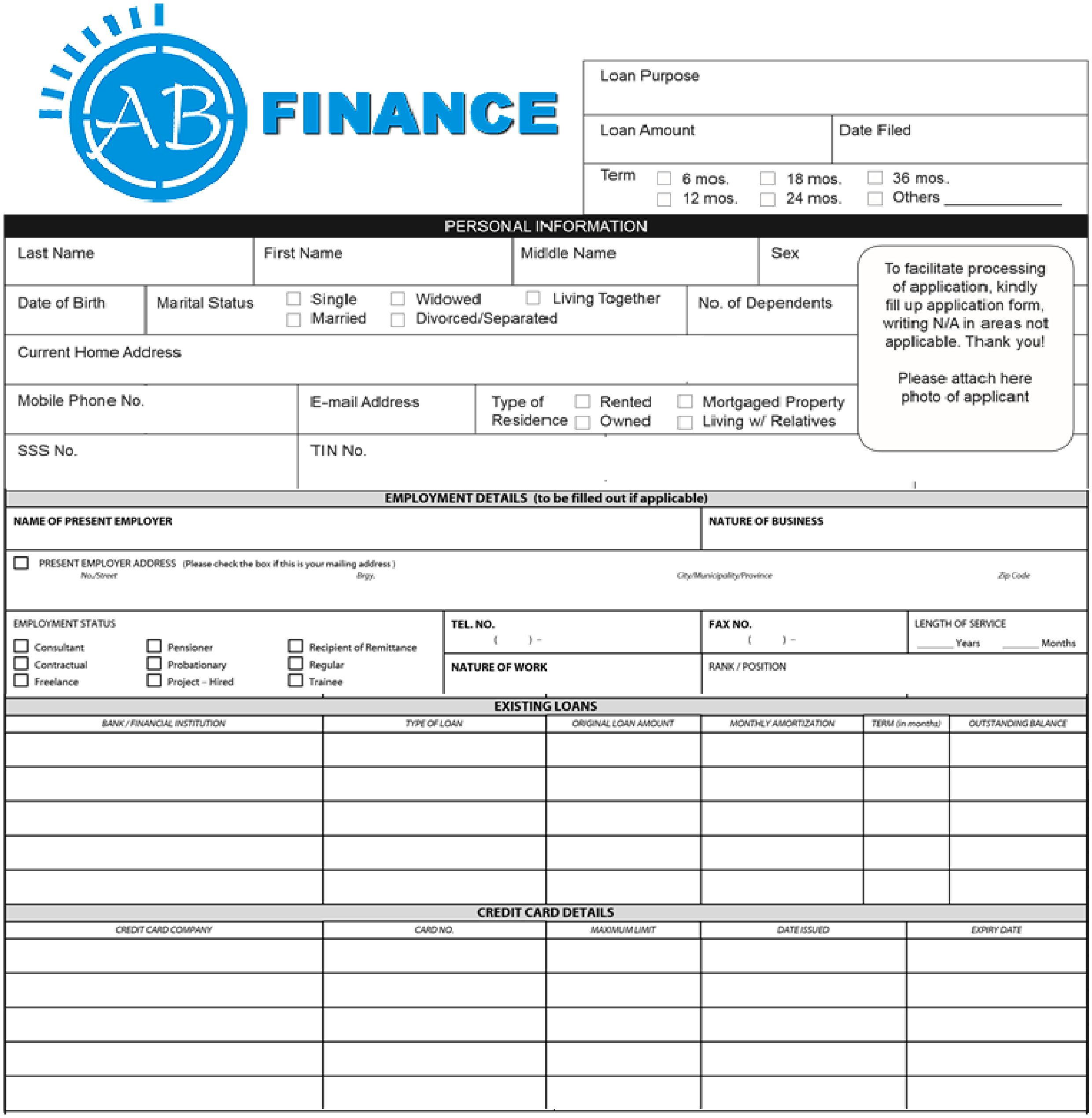
- **Customer Information Module**
  - Shows the customer details such as customer name, address, contact number, UMIDNo, TIN and email.
- **Loan Transaction Module**
  - Tracking all loans availed by the customer.
  - Provides a concise summary of the customers' bill and how it compares to last months' bill or year.
  - One customer can have more than one loan as the system can generate a unique loan application number.
  - The system can find the customer in the system by their name or customer number generated by the system.
  - Give information for all the active loans.
- **Query Module**
  - Give information for a particular customer by entering their customer number. However, since the corporation instructs that the system will not cover an online system, this module will be used by the staff instead.
- **Staff Information Module**
  - Give staff information for all branches or a specific staff using the staff number.

## 1.4 **Scope**

The system's scope includes transforming the manual loan operations to a paperless system, tracking all loans availed by the customer, and assigning different assign roles for each staff. A branch manager manages the Branch. A loan officer will qualify and add the customer to the system and the loan processor, which processes the customer's documents. The underwriter will be the one who determines the decision of the loan application. The proposed system can do the following – add customers, track the submitted documents for processing, documentation and underwriting. Display customer, staff and active loan in a particular branch. Track customer, staff, loan using the generated number generated by the system and view its repayment history. One of the database's limitations is that it is only shared through a local network and does not involve making the database online.

# II. CONCEPTUAL DATA MODEL

## 2.1 Identified User Views



AB FINANCE

Loan Purpose

Loan Amount

Date Filed

Term □ 6 mos. □ 12 mos. □ 18 mos. □ 24 mos. □ 36 mos. □ Others ____________

PERSONAL INFORMATION

Last Name

First Name

Middle Name

Sex

Date of Birth

Marital Status □ Single □ Married □ Widowed □ Divorced/Separated □ Living Together

No. of Dependents

Current Home Address

Mobile Phone No.

E-mail Address

Type of Residence □ Rented □ Owned □ Mortgaged Property □ Living w/ Relatives

To facilitate processing of application, kindly fill up application form, writing N/A in areas not applicable. Thank you!

Please attach here photo of applicant

SSS No.

TIN No.

EMPLOYMENT DETAILS (to be filled out if applicable)

NAME OF PRESENT EMPLOYER

NATURE OF BUSINESS

□ PRESENT EMPLOYER ADDRESS (Please check the box if this is your mailing address )

No./Street

Brgy.

City/Municipality/Province

Zip Code

EMPLOYMENT STATUS

□ Consultant □ Contractual □ Freelance □ Pensioner □ Probationary □ Project – Hired □ Recipient of Remittance □ Regular □ Trainee

TEL. NO. ( ) -

FAX NO. ( ) -

LENGTH OF SERVICE _____ Years _____ Months

NATURE OF WORK

RANK / POSITION

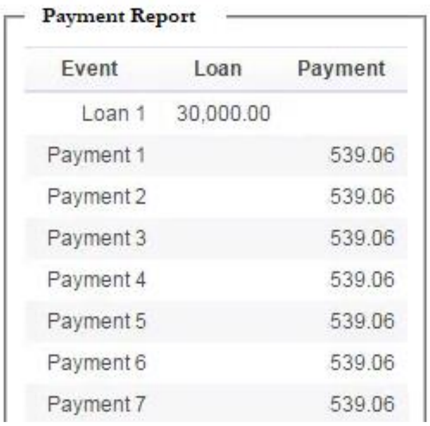
EXISTING LOANS

| BANK / FINANCIAL INSTITUTION | TYPE OF LOAN | ORIGINAL LOAN AMOUNT | MONTHLY AMORTIZATION | TERM (in months) | OUTSTANDING BALANCE |
|---|---|---|---|---|---|
| | | | | | |
| | | | | | |
| | | | | | |
| | | | | | |
| | | | | | |

CREDIT CARD DETAILS

| CREDIT CARD COMPANY | CARD NO. | MAXIMUM LIMIT | DATE ISSUED | EXPIRY DATE |
|---|---|---|---|---|
| | | | | |
| | | | | |
| | | | | |
| | | | | |
| | | | | |

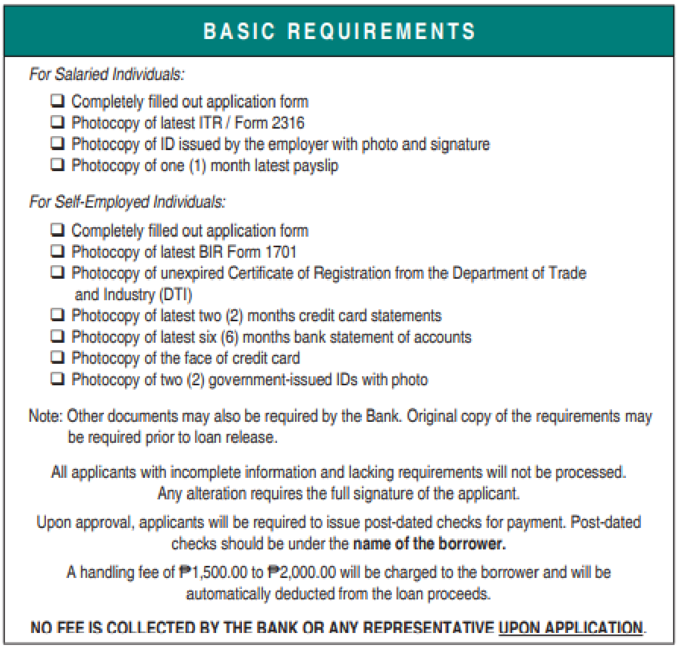
Figure 1: Finance Loan Application Form



Payment Report

| Event | Loan | Payment |
|---|---|---|
| Loan 1 | 30,000.00 | |
| Payment 1 | | 539.06 |
| Payment 2 | | 539.06 |
| Payment 3 | | 539.06 |
| Payment 4 | | 539.06 |
| Payment 5 | | 539.06 |
| Payment 6 | | 539.06 |
| Payment 7 | | 539.06 |

Figure 2: AB Finance Payment Report



BASIC REQUIREMENTS

*For Salaried Individuals:*

- ❑ Completely filled out application form
- ❑ Photocopy of latest ITR / Form 2316
- ❑ Photocopy of ID issued by the employer with photo and signature
- ❑ Photocopy of one (1) month latest payslip

*For Self-Employed Individuals:*

- ❑ Completely filled out application form
- ❑ Photocopy of latest BIR Form 1701
- ❑ Photocopy of unexpired Certificate of Registration from the Department of Trade and Industry (DTI)
- ❑ Photocopy of latest two (2) months credit card statements
- ❑ Photocopy of latest six (6) months bank statement of accounts
- ❑ Photocopy of the face of credit card
- ❑ Photocopy of two (2) government-issued IDs with photo

Note: Other documents may also be required by the Bank. Original copy of the requirements may be required prior to loan release.

All applicants with incomplete information and lacking requirements will not be processed. Any alteration requires the full signature of the applicant.

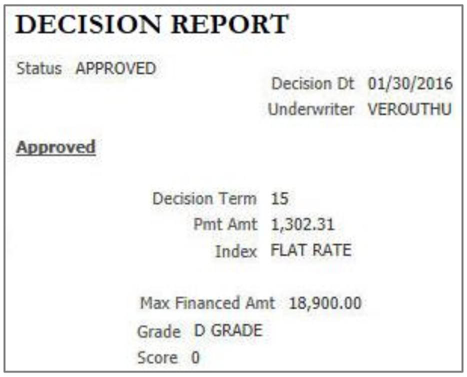
Upon approval, applicants will be required to issue post-dated checks for payment. Post-dated checks should be under the **name of the borrower.**

A handling fee of ₱1,500.00 to ₱2,000.00 will be charged to the borrower and will be automatically deducted from the loan proceeds.

**NO FEE IS COLLECTED BY THE BANK OR ANY REPRESENTATIVE UPON APPLICATION.**

Figure 3: Required Documents

Figure 4: Decision Report



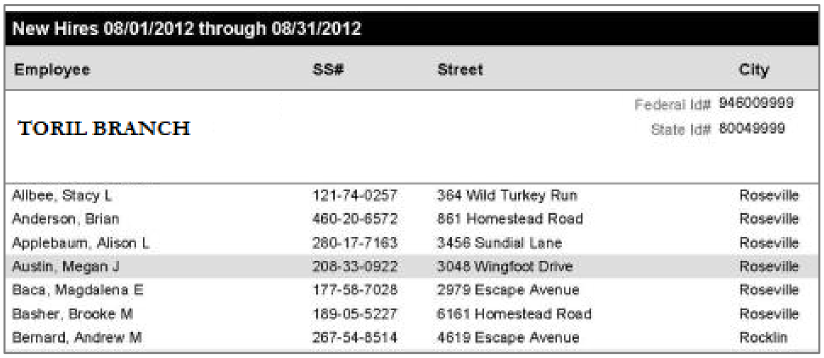
DECISION REPORT

| | |
|---|---|
| Status | APPROVED |
| Decision Dt | 01/30/2016 |
| Underwriter | VEROUTHU |

Approved

| | |
|---|---|
| Decision Term | 15 |
| Pmt Amt | 1,302.31 |
| Index | FLAT RATE |
| Max Financed Amt | 18,900.00 |
| Grade | D GRADE |
| Score | 0 |

Figure 5: Staff Details

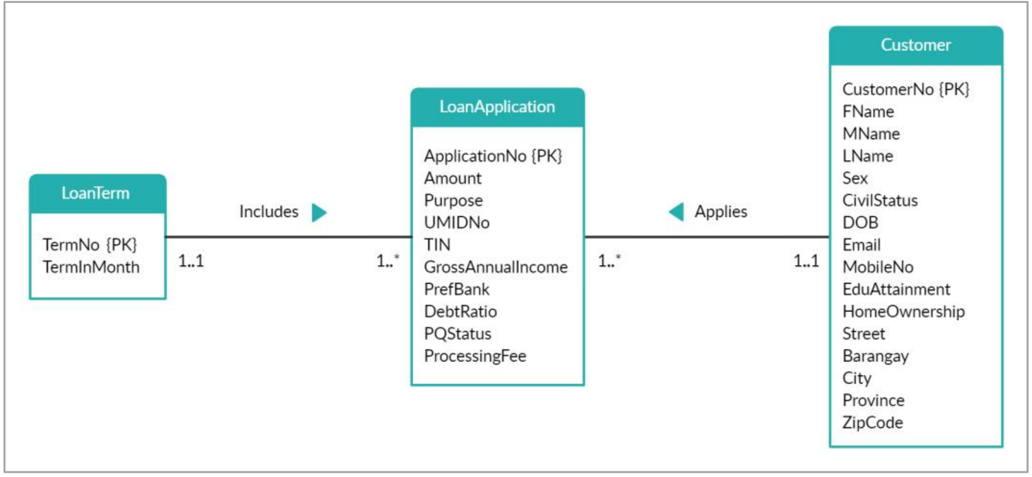


New Hires 08/01/2012 through 08/31/2012

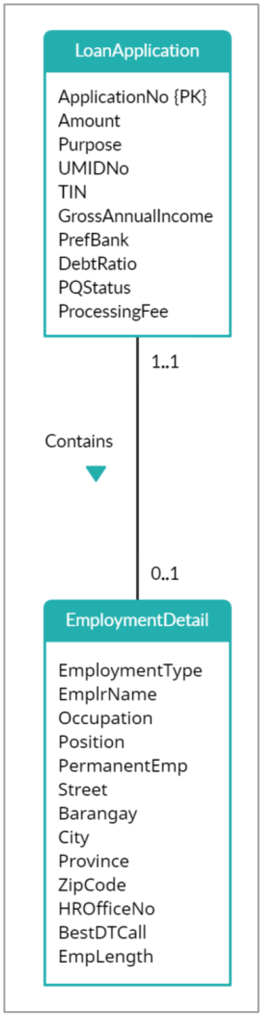
| Employee | SS# | Street | City |
|---|---|---|---|
| TORIL BRANCH | | | Federal Id# 946009999<br>State Id# 80049999 |
| Allbee, Stacy L | 121-74-0257 | 364 Wild Turkey Run | Roseville |
| Anderson, Brian | 460-20-6572 | 861 Homestead Road | Roseville |
| Applebaum, Alison L | 280-17-7163 | 3456 Sundial Lane | Roseville |
| Austin, Megan J | 208-33-0922 | 3048 Wingfoot Drive | Roseville |
| Baca, Magdalena E | 177-58-7028 | 2979 Escape Avenue | Roseville |
| Basher, Brooke M | 189-05-5227 | 6161 Homestead Road | Roseville |
| Bernard, Andrew M | 267-54-8514 | 4619 Escape Avenue | Rocklin |

## 2.2 Conceptual Data Model

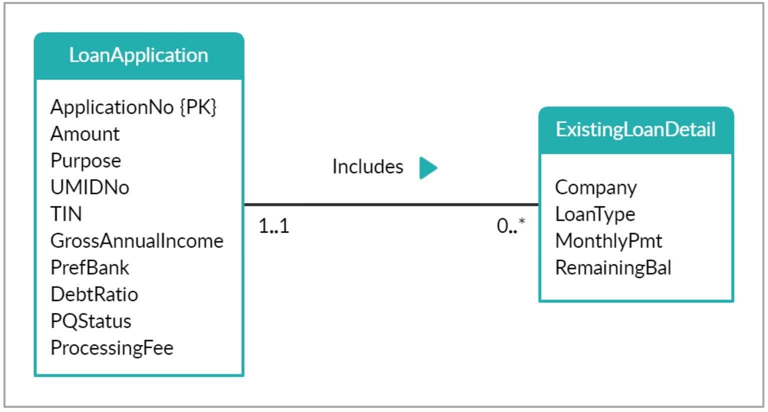
Input User View 1 (Figure 1)



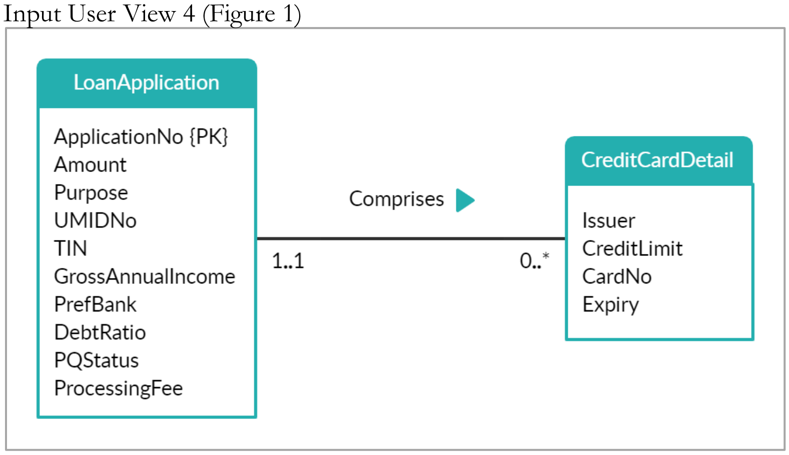
Input User View 2 (Figure 1)



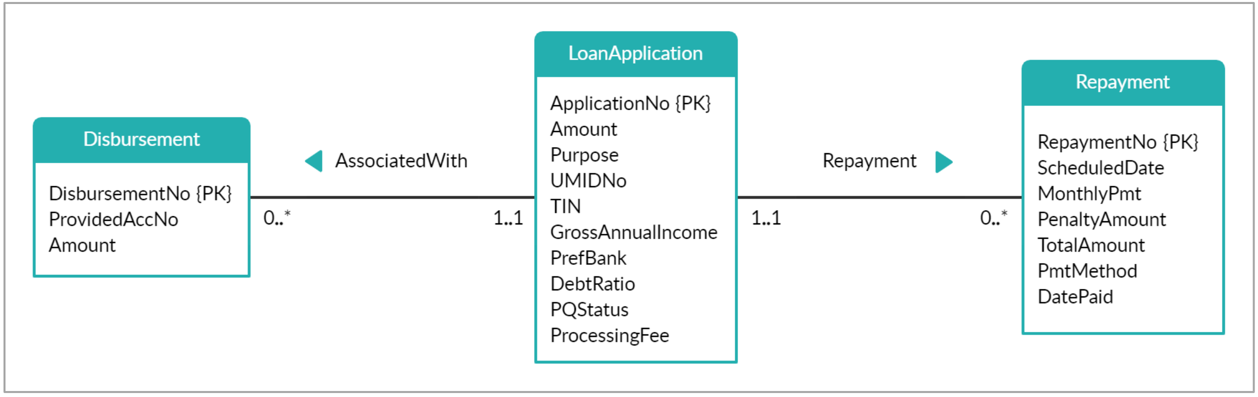
Input User View 3 (Figure 1)



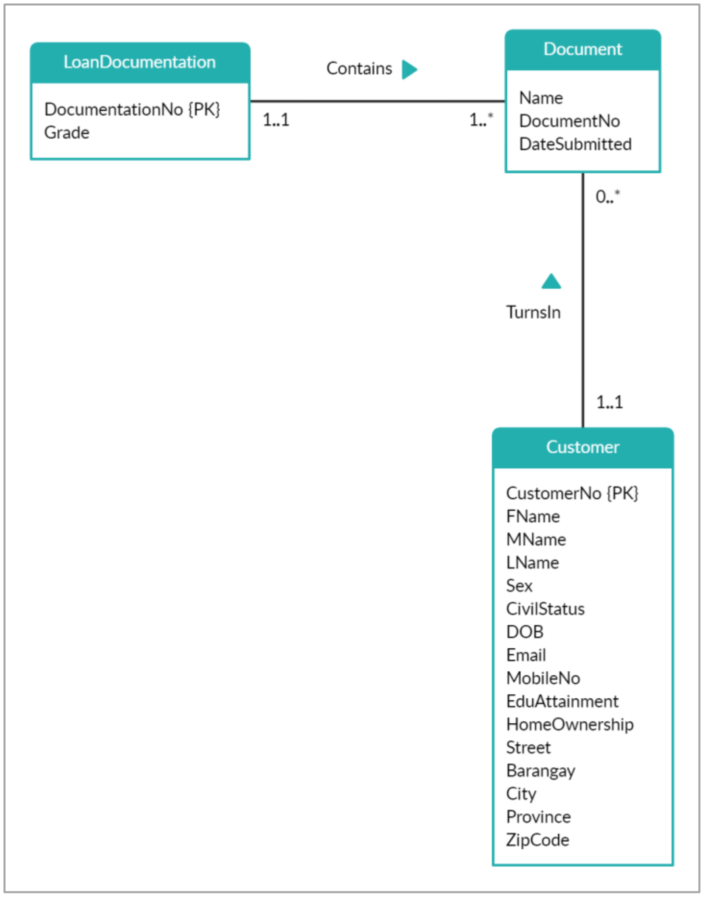
Input User View 4 (Figure 1)



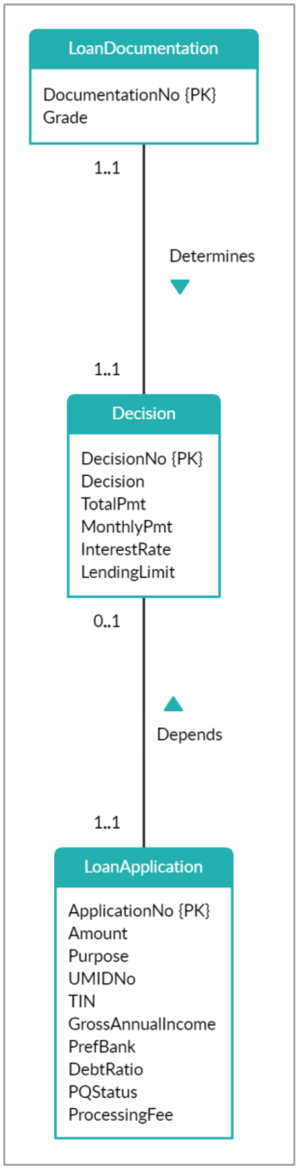
Output User View 1 (Figure 2)



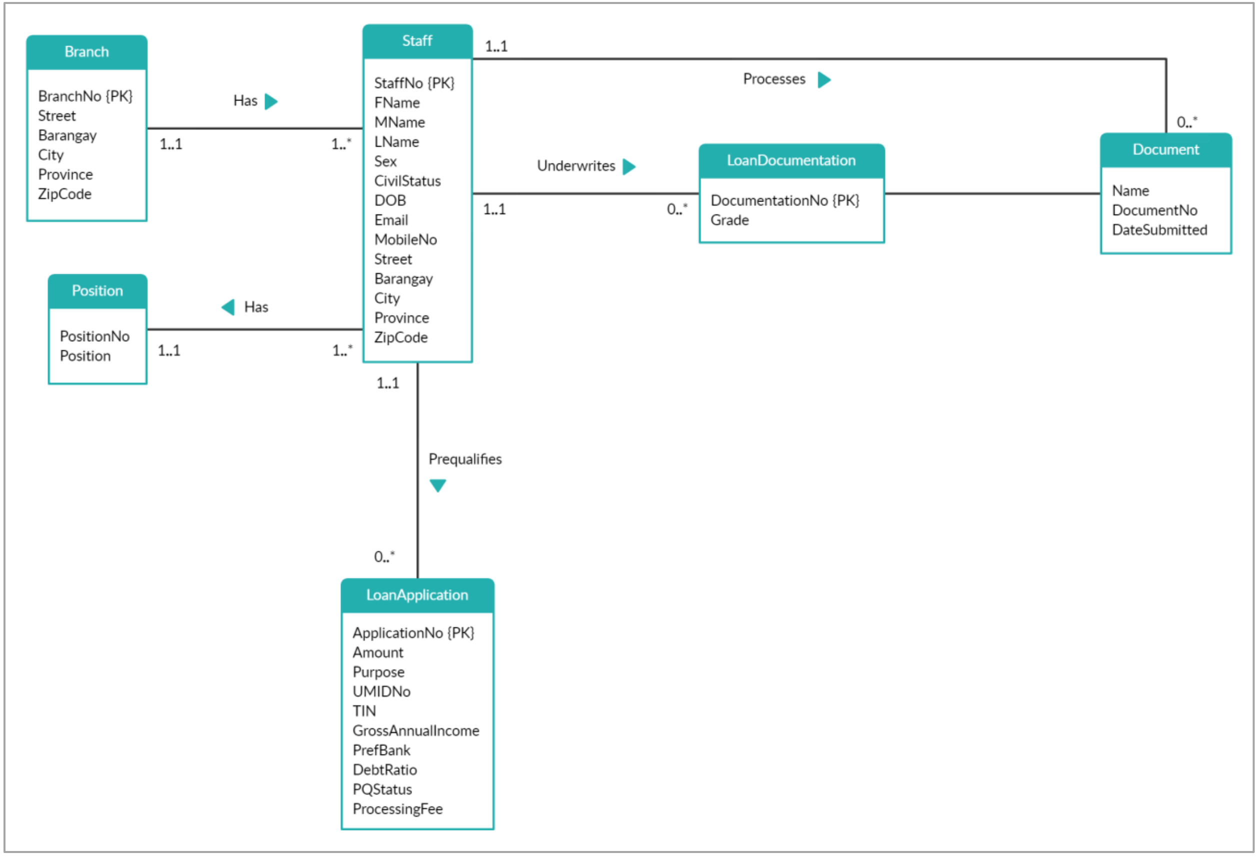
Output User View 2 (Figure 3)



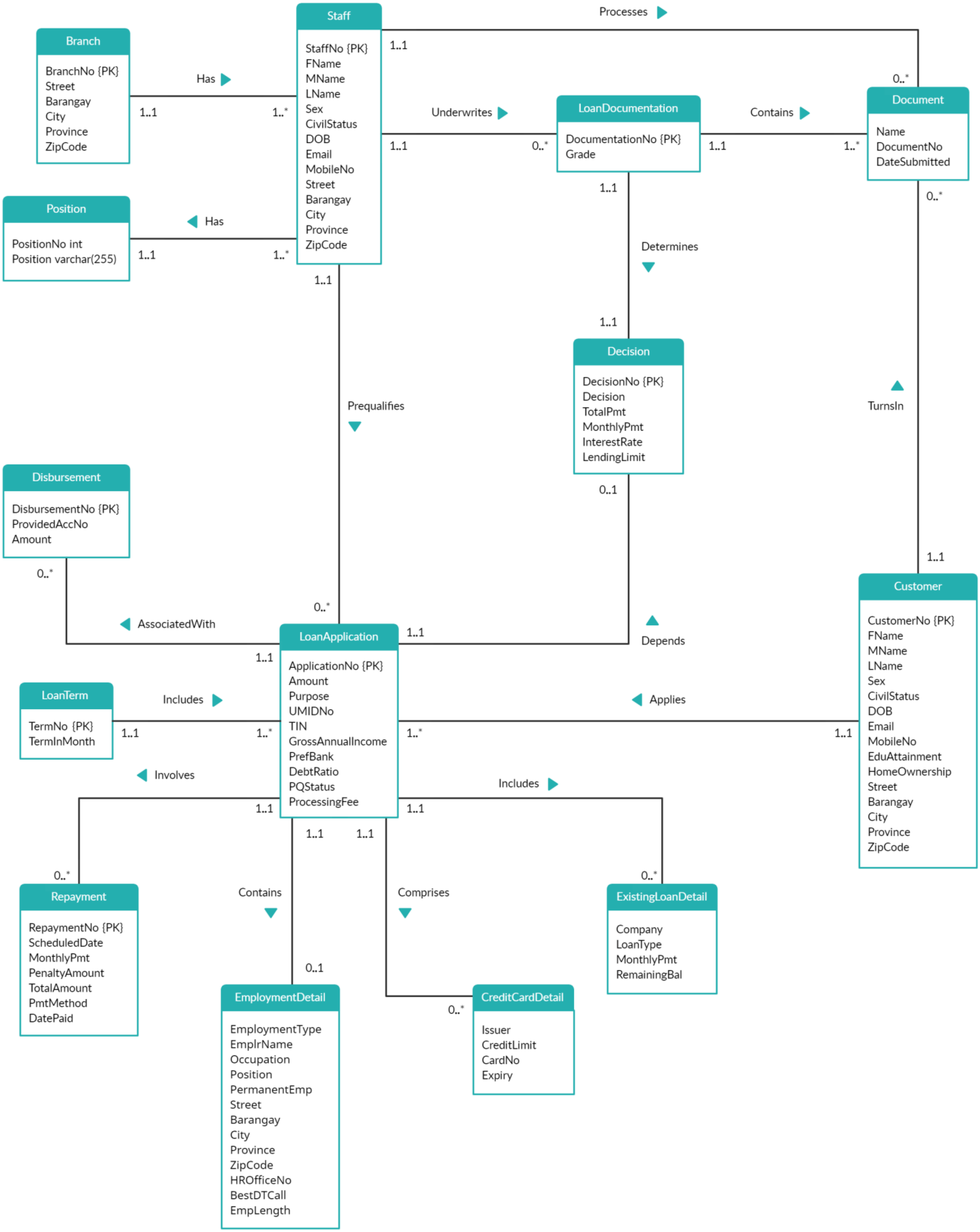
Output User View 3 (Figure 4



Output User View 4 (Figure 5)



## Single ERD of User Views



## 2.3 Documentation

### Entity Types Description

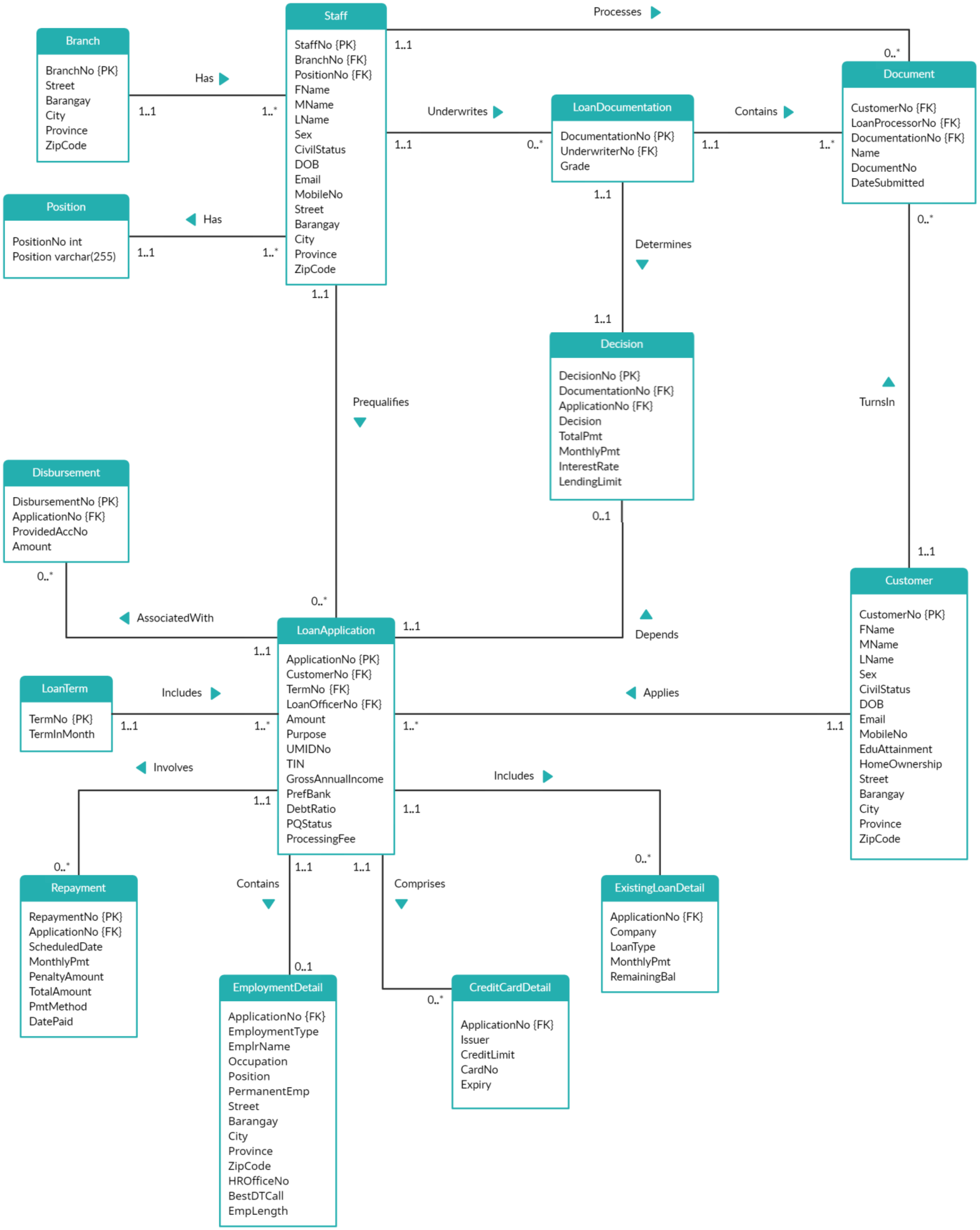
| Entity Type | Description |
|---|---|
| Branch | General term describing all branch by AB Finance Corporation. |
| CreditCardDetail | General term describing all credit card own by a customer. |
| Customer | General term describing all customer. |
| Decision | General term describing the final loan decision of the loan application. |
| Disbursement | General term describing the funding or sending of the requested loan. Disbursement is along with the creation of the disbursement account in which the money will be disburse in that account. |
| Document | General term describing all documents required by a staff particularly loan officer that is for processing, verification and underwriting that will be used for the final loan decision of the loan application. |
| EmploymentDetail | Employment details of customer that is part of loan application. |
| ExistingLoanDetail | General term describing all existing loan in AB Finance Corporation or other financial institution offering different types of loan. |
| LoanApplication | General term describing the loan application applied by a customer. |
| LoanDocumentation | General term for the all documents compiled in a single file related to the loan application. |
| LoanTerm | General term for describing the term of the loan. |
| Position | General term for the position of staff. |
| Repayment | General term describing the repayment of the disbursement amount or loan application. |
| Staff | General term describing all staff employed by AB Finance Corporation which has specific roles in processing the loan application. |

### Relationship Description

| Entity Type | Multiplicity | Relationship | Multiplicity | Entity Type |
|---|---|---|---|---|
| Branch | 1..1 | has | 1..* | Staff |
| Customer | 1..1<br>1..1 | turnsin<br>applies | 0..*<br>1..* | Document<br>LoanApplication |
| LoanApplication | 1..1<br>1..1<br>1..1<br>1..1<br>1..1<br>1..*<br>1..1 | comprises<br>depends<br>associatedwith<br>contains<br>includes<br>includes<br>involves | 0..*<br>0..1<br>0..*<br>0..1<br>0..*<br>1..1<br>0..* | CreditCardDetail<br>Decision<br>Disbursement<br>EmploymentDetail<br>ExistingLoanDetail<br>LoanTerm<br>Repayment |
| LoanDocumentation | 1..1<br>1..1 | determines<br>contains | 1..1<br>1..* | Decision<br>Document |
| Staff | 1..1<br>1..1<br>1..1<br>1..* | processes<br>prequalifies<br>underwrites<br>has | 0..*<br>0..*<br>0..*<br>1..1 | Document<br>LoanApplication<br>LoanDocumentation<br>Position |

# III. LOGICAL DATABASE DESIGN

## 3.1 FINAL ERD



## 3.2 FINAL DATA DICTIONARY

**Branch**

| Name | Key | Data Type | Null | Default | References | Description |
|---|---|---|---|---|---|---|
| BranchNo | PK | tinyint | | | | Primary key for branch table. |
| Street | | varchar(255) | | | | Name of the street. |
| Barangay | | varchar(255) | | | | Name of the barangay. |
| City | | varchar(255) | | | | Name of the city. |
| Province | | varchar(255) | | | | Name of the province. |
| ZipCode | | char(4) | | | | Zip code of the street address |

**CreditCardDetail**

| Name | Key | Data Type | Null | Default | References | Description |
|---|---|---|---|---|---|---|
| ApplicationNo | FK | int | | | LoanApplication | Foreign key to LoanApplication table |
| Issuer | | varchar(255) | | | | Provider of the credit card. |
| CreditLimit | | int | Yes | Null | | Credit limit. |
| CardNo | | char(19) | | | | Credit card number. |
| Expiry | | date | Yes | Null | | Expiration date. |

**Customer**

| Name | Key | Data Type | Null | Default | References | Description |
|---|---|---|---|---|---|---|
| CustomerNo | PK | int | | | | Primary key for customer table. |
| FName | | varchar(255) | | | | First name. |
| MName | | varchar(255) | Yes | Null | | Middle name. |
| LName | | varchar(255) | | | | Last name. |
| Sex | | enum | | | | Gender. |
| CivilStatus | | enum | | | | Civil Status |
| DOB | | date | | | | Date of birth. |
| UMIDNo | | char(13) | | | | UMID. Also accepts SSS/GSIS number. |
| TIN | | char(12) | | | | Tax Identification Number. |
| Email | | varchar(255) | | | | Email Address. |
| MobileNo | | char(12) | | | | Mobile number. |
| EduAttainment | | varchar(255) | | | | Highest education attainment. |
| HomeOwnership | | enum | | | | Owner of the house. |
| Street | | varchar(255) | | | | Name of the street. |
| Barangay | | varchar(255) | | | | Name of the barangay. |
| City | | varchar(255) | | | | Name of the city. |
| Province | | varchar(255) | | | | Name of the province. |
| ZipCode | | char(4) | | | | Zip code of the street address |

**Decision**

| Name | Key | Data Type | Null | Default | References | Description |
|---|---|---|---|---|---|---|
| DecisionNo | PK | int | | | | Primary key of Decision table. |
| DocumentationNo | FK | int | | | LoanDocumentation | Foreign key to LoanDocumentation table. |
| ApplicationNo | FK | int | | | LoanApplication | Foreign key to LoanApplication table. |
| Decision | | enum | | | | Final decision of the loan. |
| TotalPmt | | decimal(11,4) | | | | Total payment. |
| MonthlyPmt | | decimal(11,4) | | | | Monthly payment. |
| InterestRate | | decimal(5,4) | | | | Interest rate. |
| LendingLimit | | decimal(11,4) | | | | Maximum amount available to borrow. |

**Dibusrsement**

| Name | Key | Data Type | Null | Default | References | Description |
|---|---|---|---|---|---|---|
| DisbursementNo | PK | int | | | | Primary key of Disbursement table. |
| ApplicationNo | FK | int | | | LoanApplication | Foreign key to LoanApplication table. |
| ProvidedAccNo | | char(19) | | | | Account number provided by branch. |
| Amount | | decimal(11,4) | | | | Amount disbursed. |
| DateDisbursed | | date | | | | Date of disbursement. |

**Document**

| Name | Key | Data Type | Null | Default | References | Description |
|---|---|---|---|---|---|---|
| CustomerNo | FK | int | | | Customer | Foreign key to Customer table. |
| LoanProcessorNo | FK | int | | | Staff | Foreign key to Staff table. |
| DocumentationNo | FK | int | | | LoanDocumentation | FK to LoanDocumentation. |
| DocumentNo | | int | | | | Unique identifier of the row. |
| Name | | varchar(255) | | | | Name of the document. |
| DateSubmitted | | date | | | | Date submitted. |

**EmploymentDetail**

| Name | Key | Data Type | Null | Default | References | Description |
|---|---|---|---|---|---|---|
| ApplicationNo | FK | int | | | LoanApplication | Foreign key to LoanApplication table. |
| EmploymentType | | enum | | | | Type of employment. |
| CompanyName | | varchar(255) | Yes | Null | | Name of company. |
| Occupation | | varchar(255) | | | | Occupation. |
| Position | | varchar(255) | Yes | Null | | Position. |
| Street | | varchar(255) | | | | Name of the street. |
| Barangay | | varchar(255) | | | | Name of the barangay. |
| City | | varchar(255) | | | | Name of the city. |
| Province | | varchar(255) | | | | Name of the province. |
| ZipCode | | char(4) | | | | Zip code of the street address |
| HROfficeNo | | char(12) | Yes | Null | | Contact number of HR Office. |
| YearsEmployed | | tinyint | | | | Total years at work/business. |

**ExistingLoanDetail**

| Name | Key | Data Type | Null | Default | References | Description |
|---|---|---|---|---|---|---|
| ApplicationNo | FK | int | | | LoanApplication | Foreign key to LoanApplication table. |
| CompanyName | | varchar(255) | | | | Loan provider of the existing loan. |
| LoanType | | varchar(255) | | | | Type of loan. |
| MonthlyPmt | | decimal(11,4) | | | | Monthly payment. |
| RemainingBal | | decimal(11,4) | | | | Remaining balance. |

**LoanApplication**

| Name | Key | Data Type | Null | Default | References | Description |
|---|---|---|---|---|---|---|
| ApplicationNo | PK | int | | | | Primary key of LoanApplication table. |
| CustomerNo | FK | int | | | Customer | Foreign key to Customer table. |
| TermNo | FK | int | | | LoanTerm | Foreign key to LoanTerm table. |
| LoanOfficerNo | FK | int | | | Staff | Foreign key to Staff table. |
| Amount | | decimal(11,4) | | | | Loan amount applied. |
| Purpose | | varchar(255) | Yes | Null | | Purpose. |
| GrossAnnualIncome | | decimal(11,4) | | | | Gross annual income of customer. |
| PrefBank | | varchar(255) | | | | Preferred bank for disbursement. |
| DebtRatio | | decimal(3,0) | | | | Debt ratio of customer. |
| InitialStatus | | enum | | | | Initial status of the loan. |
| ProcessingFee | | decimal(9,4) | Yes | 0 | | Processing fee. |

**LoanDocumentation**

| Name | Key | Data Type | Null | Default | References | Description |
|---|---|---|---|---|---|---|
| DocumentationNo | PK | int | | | | Primary key of LoanDocumentation table. |
| UnderwriterNo | FK | int | | | Staff | Foreign key to Staff table. |
| Grade | | char(2) | | | | Loan grade based on submitted documents. |

**LoanTerm**

| Name | Key | Data Type | Null | Default | References | Description |
|---|---|---|---|---|---|---|
| TermNo | PK | tinyint | | | | Primary key of LoanTerm table. |
| TermInMonths | | smallint | | | | Term of the loan. |

**Position**

| Name | Key | Data Type | Null | Default | References | Description |
|---|---|---|---|---|---|---|
| PositionNo | PK | smallint | | | | Primary key of Position table. |
| Position | | varchar(255) | | | | Position. |

**Repayment**

| Name | Key | Data Type | Null | Default | References | Description |
|---|---|---|---|---|---|---|
| RepaymentNo | PK | int | | | | Primary key of Repayment table. |
| ApplicationNo | FK | int | | | LoanApplication | Foreign key to LoanApplication table. |
| ScheduledDate | | date | | | | Scheduled date of repayment. |
| MonthlyPmt | | decimal(11,4) | | | | Amount of monthly repayment. |
| PenaltyAmount | | decimal(8,4) | | | | Penalty amount. |
| TotalAmount | | decimal(11,4) | | | | Total amount. |
| PmtMethod | | varchar(255) | | | | Payment method. |
| DatePaid | | date | | | | Date the loan paid. |

**Staff**

| Name | Key | Data Type | Null | Default | References | Description |
|---|---|---|---|---|---|---|
| StaffNo | PK | int | | | | Primary key of Staff table. |
| BranchNo | FK | tinyint | | | Branch | Foreign key to Branch table. |
| PositionNo | FK | int | | | Position | Foreign key to Position table. |
| FName | | varchar(255) | | | | First name. |
| MName | | varchar(255) | Yes | Null | | Middle name. |
| LName | | varchar(255) | | | | Last name. |
| Salary | | decimal(11,4) | | | | Salary. |
| Sex | | enum | | | | Sex |
| CivilStatus | | enum | | | | Civil status. |
| DOB | | date | | | | Date of birth. |
| Email | | varchar(255) | | | | Email address. |
| MobileNo | | char(12) | | | | Mobile number. |
| Street | | varchar(255) | | | | Name of the street. |
| Barangay | | varchar(255) | | | | Name of the barangay. |
| City | | varchar(255) | | | | Name of the city. |
| Province | | varchar(255) | | | | Name of the province. |
| ZipCode | | char(4) | | | | Zip code of the street address |

**Domain of Attributes**

| Attribute | Domain |
|---|---|
| Amount | 30000.0000-2000000.9999 |
| CivilStatus | Single, Married, Widow, Separated, Annuled |
| DebtRatio | 1-100 |
| Decision | Approved, Suspended, Rejected |
| EmploymentType | Government, Private, Self-Employed |
| HomeOwnership | Owned, Mortgaged, Rented, Used Free/Living With Parents/Relatives |
| InitialStatus | Approved, Rejected |
| InterestRate | 1.0000-4.9999 |
| LendingLimit | 30000.0000-2000000.9999 |
| Position | Branch Manager, Underwriter, Loan Processor, Loan Officer |
| Sex | Male, Female, Unknown |

**Physical Data Model Generated In MySQL Workbench**

## 3.3 TRANSFORMATION OF ERDS TO NORMALIZED RELATIONS

- Due to the process made in the Conceptual Data Model where relations with repeating groups and multi-valued attributes were removed and made into a separate entity, most of the relations are sufficiently normalized. The process involved is shown below.

### 0NF

**LoanApplication**

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| ApplicationNo | Amount | LoanTerm | Purpose | GrossAnnualIncome | PrefBank | DebtRatio | InitialStatus |

| | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| ProcessingFee | CustomerName | Sex | CivilStatus | DOB | UMIDNo | TIN | Email | MobileNo | EduAttainment |

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| HomeOwnership | Address | ZipCode | EmploymentType | CompanyName | Occupation | Position | Address |

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| ZipCode | HROfficeNo | YearsEmployed | CompanyNameEL | LoanType | MonthlyPmt | RemainingBal | Issuer |

| | | |
|---|---|---|
| CreditLimit | CardNo | Expiry |

| | | |
|---|---|---|
| Repeating Group 1 | → | CustomerName, Sex, CivilStatus, DOB, UMIDNo, TIN, Email, MobileNo, EduAttainment,HomeOwnership, Address, ZipCode. |
| Repeating Group 2 | → | LoanTerm |
| Nullable Group 1 | → | EmploymentType, CompanyName, Occupation, Position, Address, Address, ZipCode, HROfficeNo, YearsEmployed. |
| Multi-valued Attribute 1 | → | CompanyNameEL, LoanType, MonthlyPMT, RemainingBal. |
| Multi-valued Attribute 2 | → | Issuer, CreditLimit, CardNo, Expiry. |

**Physical Data Model Generated In MySQL Workbench**

**ALTERNATIVE 1NF**

**LoanApplication**

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| ApplicationNo | Amount | LoanTerm | Purpose | GrossAnnualIncome | PrefBank | DebtRatio | InitialStatus | ProcessingFee |

**Customer**

| | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| CustomerName | Sex | CivilStatus | DOB | UMIDNo | TIN | Email | MobileNo | EduAttainment | HomeOwnership | Address | ZipCode |

**EmploymentDetail**

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| EmploymentType | CompanyName | Occupation | Position | Address | ZipCode | HROfficeNo | YearsEmployed |

**ExistingLoanDetail**

| | | | |
|---|---|---|---|
| CompanyNameEL | LoanType | MonthlyPmt | RemainingBal |

**CreditCardDetail**

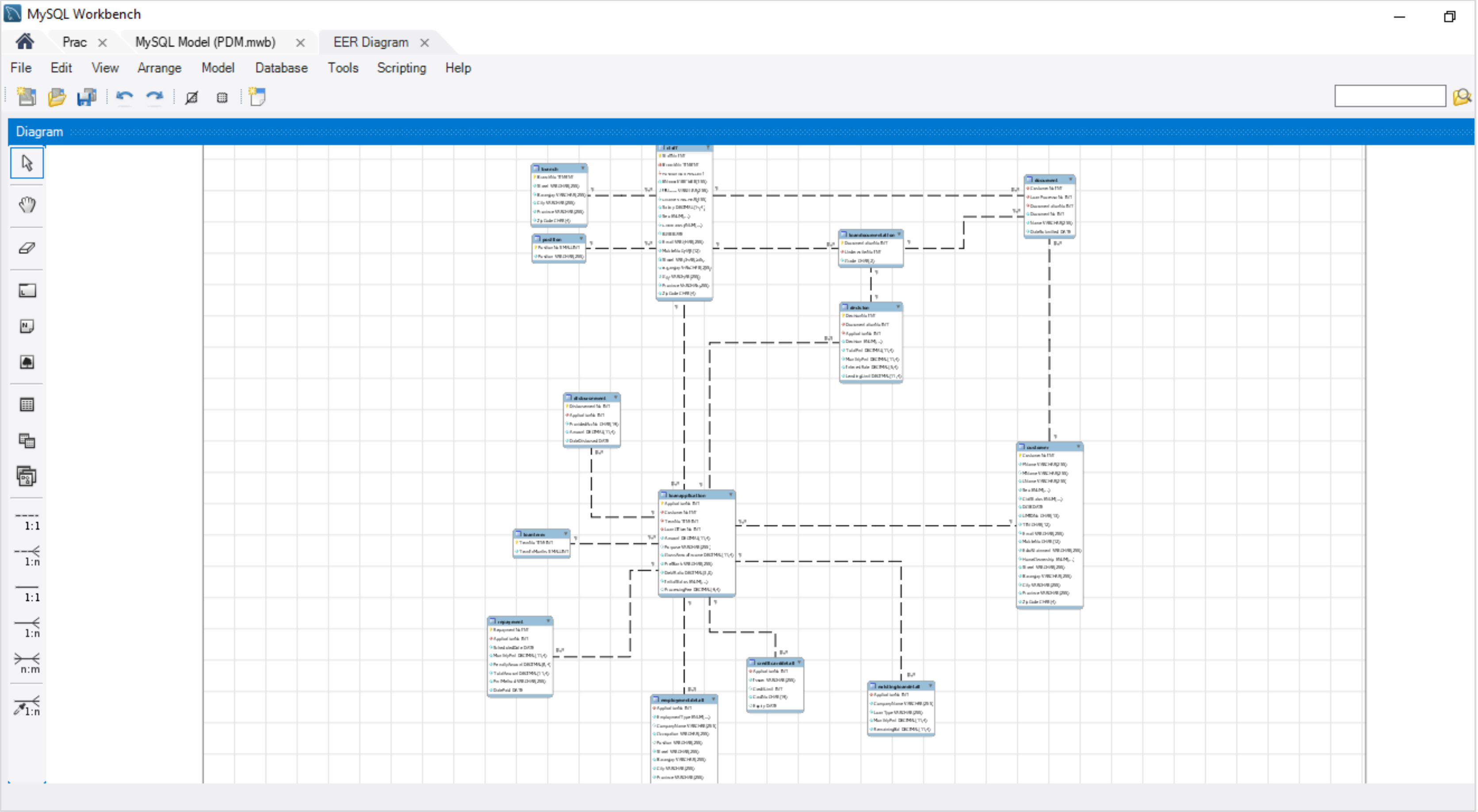
| | | | |
|---|---|---|---|
| Issuer | CreditLimit | CardNo | Expiry |

An alternative approach of 1NF is used on this relation where the repeating data is placed along with a copy of the original key attributes in a separate relation. This approach also normalized into the same set of 3NF relations. The nullable group is also separated on this relation as preferred by the database designers.

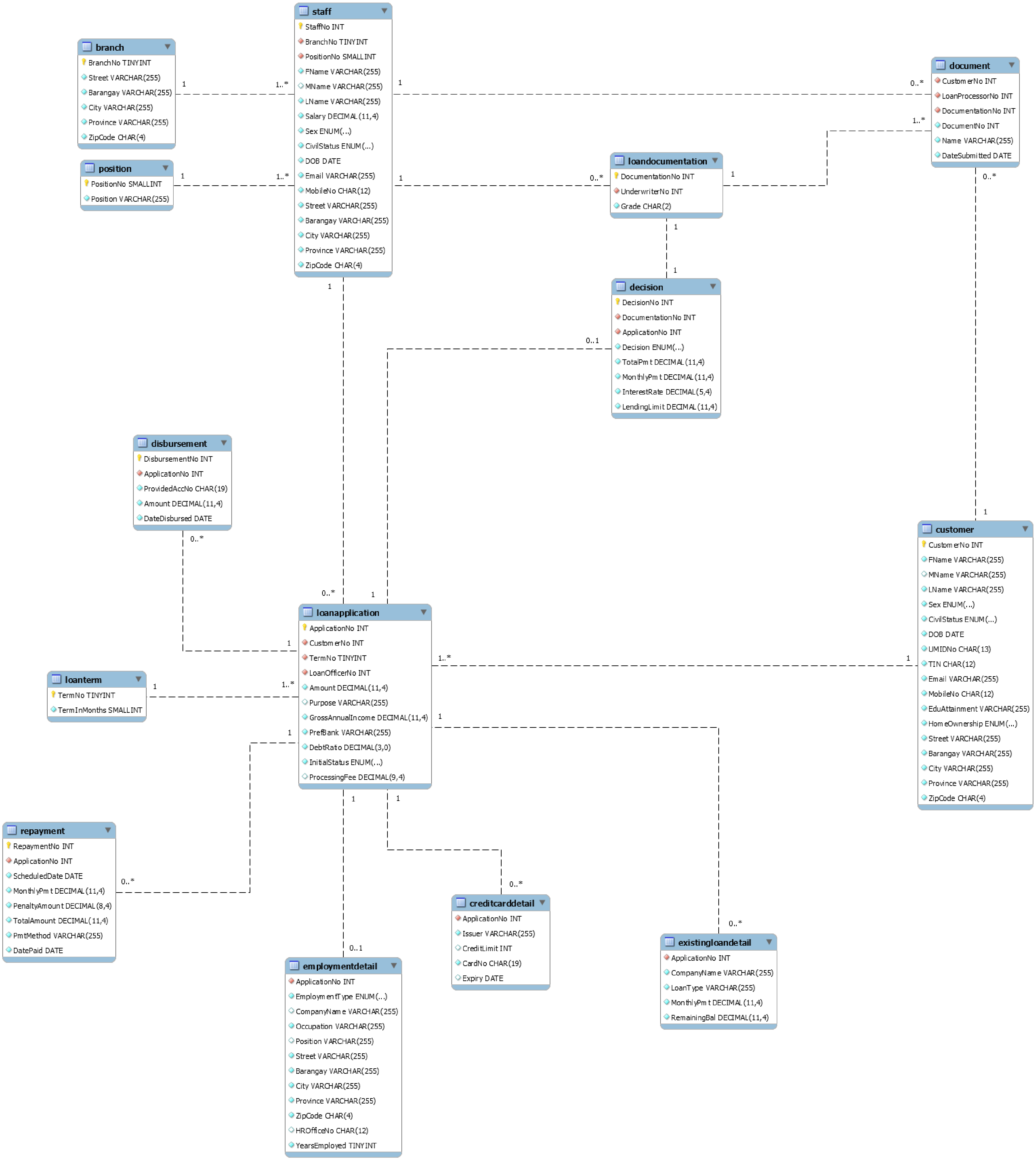
**Physical Data Model Generated In MySQL Workbench**

## 3.4 FINAL NORMALIZED DIAGRAM

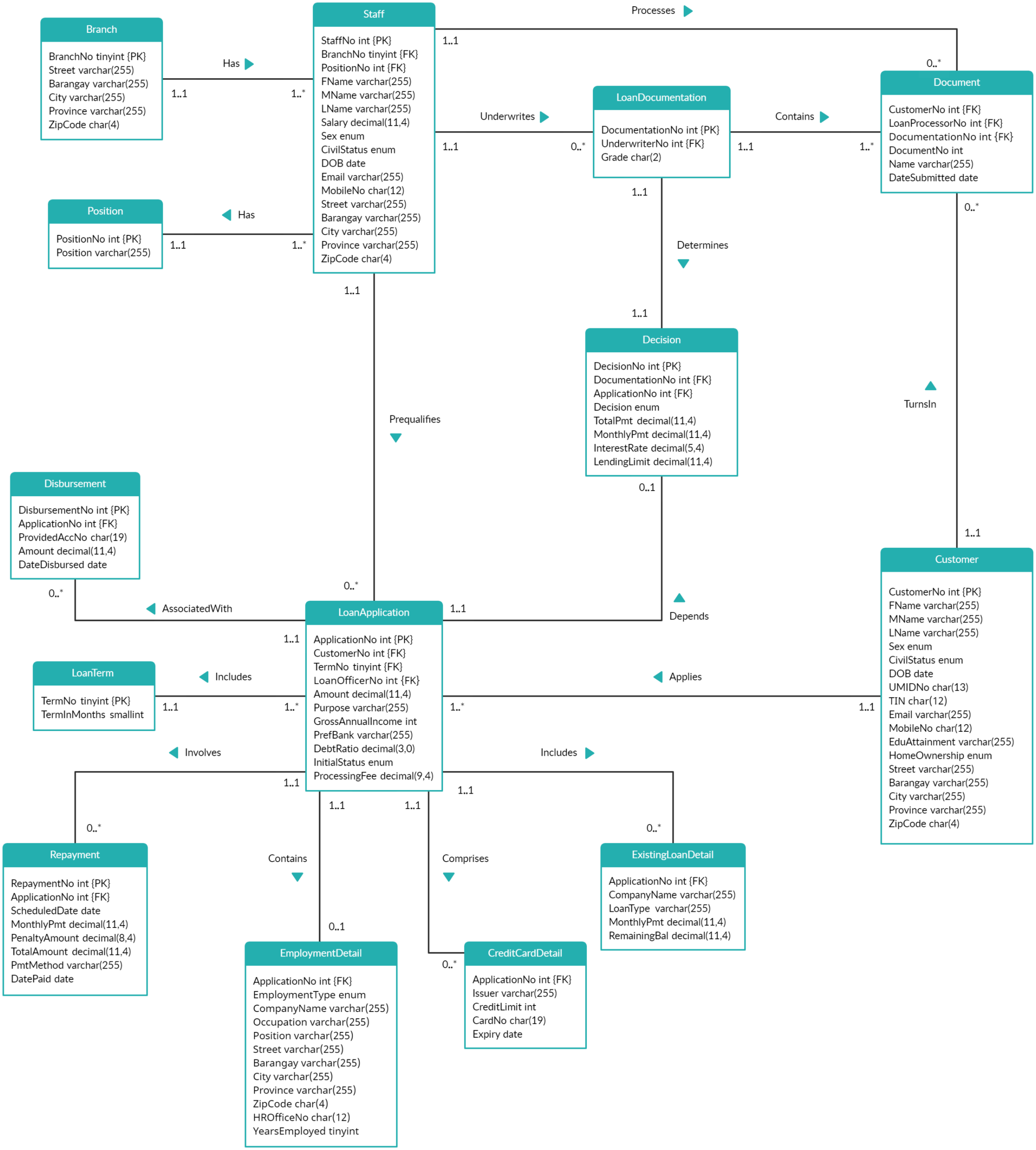
Screenshot of the Diagram Generated by the Database



**Physical Data Model Generated In MySQL Workbench**



**Created Physical Data Model By The Database Designers**



# IV. PHYSICAL DATABASE DESIGN

## 4.1 INTEGRITY CONSTRAINTS

- Branch & Staff

  - Every Branch must have a Staff.
  - Once a Branch record is updated, the Staff record will follow.
  - Once a Branch record gets deleted, the deletion will fail due to the Staff record's restrictions.

- Customer & Document

  - Every Customer may or may not turns in a document.
  - Once a Customer record is updated, a Document record will follow.
  - Once a Customer record gets deleted, the deletion will fail due to the document record's restrictions.

- Customer & LoanApplication

  - Every Customer must have a Loan Application.
  - Once a Customer record is updated, the Loan Application record will follow.
  - Once a Customer record gets deleted, the deletion will fail due to the Loan Application record's restrictions.

- LoanApplication & CreditCardDetail, Decision, Disbursement, EmploymentDetail, ExistingLoanDetail and Repayment

  - Every Loan Application may or may not have a Credit Card Detail, Decision, Disbursement, Employment Detail, Existing Loan and Repayment. However, once a Loan Application is updated, all the said records will follow.
  - Once a Loan Application record gets deleted, the deletion will fail due to the restrictions made by all the said records.

- LoanDocumentation & Decision and Document

  - Every Loan Documentation must have a Decision and Document.
  - Once a Loan Documentation record is updated, the Decision and Document record will follow.
  - Once a Loan Documentation record gets deleted, the deletion will fail due to the Decision and Document record's restrictions.

- LoanTerm & LoanApplication

  - For every Loan Term record, there should be a corresponding Loan Application record on it.
  - Once a Loan Term record is updated, the Loan Application record will follow.
  - Once a Loan Term record gets deleted, the deletion will fail due to the Loan Application record's restrictions.

- Position & Staff

    - For every Position record, there should be a corresponding Staff record on it.
    - Once a Loan Term record is updated, a Staff record will follow.
    - Once a Loan Term record gets deleted, the deletion will fail due to the restrictions made by the Staff record.

- Staff & Document

    - Every Staff may or may not assign working in Document.
    - Once a Staff record is being updated, Document records will follow.
    - Once a Staff record is being deleted, the deletion will fail due to the Document record's restrictions.

- Staff & LoanApplication

    - Every Staff may or may not prequalify a Loan Application.
    - Once a Staff record is being updated, Loan Application records will follow.
    - Once a Staff record is being deleted, the deletion will fail due to the Loan Application record's restrictions.

- Staff & LoanDocumentation

    - Every Staff may or may underwrites Loan Documentation.
    - Once a Staff record is being updated, Loan Documentation records will follow.
    - Once a Staff record is being deleted, the deletion will fail due to the Loan Documentation record's restrictions.

## 4.2 GENERAL CONSTRAINTS

- Customer must age 21 and above.
- The customer must be no older than 65 upon loan maturity.
- The customer must have an active Philippine mobile number.
- Customers can reapply for a loan after six months of the last application.
- The allowable loan amount is up to ₱2,000,000.
- Customers can only have two loans.
- The initial status of the loan application must be approved to submit documents.
- Staff must age 21 and above.
- Staff must not higher than 65 years old.

## 4.3 TRANSACTIONS

STAFF

A. Enter the details for the customer's new loan application and update customer information, including the existing loan, credit card, and employment details.
B. Update the details of the loan application due to the loan adjustment.
C. Identify the assigned loan application of staff.

BRANCH

D. Update value of qualified debt ratio and other qualifications for the initial status of the loan application.
E. Identify documents submitted by customers for underwriting.
F. Identify the assigned staff of a particular loan application to update the disbursement.

| Transaction / Relation | (A) | | | | (B) | | | | (C) | | | | (D) | | | | (E) | | | | (F) | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | I | R | U | D | I | R | U | D | I | R | U | D | I | R | U | D | I | R | U | D | I | R | U | D |
| Branch | | | | | | | | | | | | | | X | | | | X | | | | X | | |
| Staff | | X | | | | X | | | | X | | | | | | | | | | | | X | | |
| Customer | | | X | | | | | | | | | | | | | | X | | | | | | | |
| Underwriting | | | | | | | | | | | | | | | | | X | | | | | | | |
| EmploymentDetails | | | X | | | | | | | | | | | | | | | | | | | | | |
| ELDetails | | | X | | | | | | | | | | | | | | | | | | | | | |
| CCDetails | | | X | | | | | | | | | | | | | | | | | | | | | |
| Document | | | | | | | | | | | | | | | | | | X | | | | | | |
| FinalLoanDecision | | | | | | | | | | | | | | | | | | | | | | | | |
| LoanApplication | X | | | | | | X | | | X | | | | | X | | | | | | | X | | |
| Disbursement | | | | | | | | | | | | | | | | | | | | | | | X | |
| Repayment | | | | | | | | | | | | | | | | | | | | | | | | |
| LoanAdjustment | | | | | | X | | | | | | | | | | | | | | | | | | |

## 4.1 CHOSEN FILE ORGANIZATION

- B+ Tree is chosen as file organization in this project as it supports referential integrity constraints and since the file organization depend on the storage engine, InnoDB is used, the default storage engine of the chosen Database Management System, MySQL.

## 4.2 CHOSEN INDEX

- The index shown in the database depends on the attributes such as attributes that are a primary key is a primary index and attributes that are not is called an index. A unique index is also present for those attributes that are unique but not a primary key.

# V. SQL DOCUMENTATION

## 5.1 SQL STATEMENTS

```
/* Creating Database*/
CREATE DATABASE PLMS CHARACTER SET utf8mb4 COLLATE utf8mb4_unicode_520_ci ;

/* Using the Database*/
USE PLMS;

/* Creating tables for the Database*/
CREATE TABLE Branch (
    BranchNo tinyint NOT NULL AUTO_INCREMENT,
    Street varchar(255) NOT NULL,
    Barangay varchar(255) NOT NULL,
    City varchar(255) NOT NULL,
    Province varchar(255) NOT NULL,
    ZipCode char(4) NOT NULL,
    PRIMARY KEY (BranchNo)
);
```

```
CREATE TABLE Customer (
    CustomerNo int NOT NULL AUTO_INCREMENT,
    FName varchar(255) NOT NULL,
    MName varchar(255),
    LName varchar(255) NOT NULL,
    Sex enum('Male', 'Female', 'Unknown') NOT NULL,
    CivilStatus enum('Single', 'Married', 'Widow', 'Separated', 'Annuled') NOT NULL,
    DOB date NOT NULL,
    UMIDNo char(13) NOT NULL,
    TIN char(12) NOT NULL,
    Email varchar(255) NOT NULL,
    MobileNo char(12) NOT NULL,
    EduAttainment varchar(255) NOT NULL,
    HomeOwnership enum('Owned', 'Mortgaged', 'Rented', 'Used Free/Living With Parents/Relatives')
    NOT NULL,
    Street varchar(255) NOT NULL,
    Barangay varchar(255) NOT NULL,
    City varchar(255) NOT NULL,
    Province varchar(255) NOT NULL,
    ZipCode char(4) NOT NULL,
    PRIMARY KEY (CustomerNo),
    KEY uiFullName (FName, MName, LName),
    KEY uiUMIDNO (UMIDNo),
    KEY uiMobileNo (MobileNo),
    KEY uiAddress (Street, Barangay),
    KEY uiZipCode (ZipCode)
);

CREATE TABLE LoanTerm (
    TermNo tinyint NOT NULL AUTO_INCREMENT,
    TermInMonths smallint NOT NULL,
    PRIMARY KEY (TermNo),
    KEY uiTerm (TermInMonths)
);

/* Values for LoanTerm*/
INSERT INTO LoanTerm (TermInMonths) VALUES (12), (18), (24), (36);
```

```
CREATE TABLE Position (
    PositionNo smallint NOT NULL AUTO_INCREMENT,
    Position varchar(255) NOT NULL,
    PRIMARY KEY (PositionNo)
);

/* Values for Position*/
INSERT INTO `Position`(Position) VALUES ('Branch Manager'), ('Loan Officer'), ('Loan Processor'),
('Underwriter');

CREATE TABLE Staff (
    StaffNo int NOT NULL AUTO_INCREMENT,
    BranchNo tinyint NOT NULL,
    PositionNo smallint NOT NULL,
    FName varchar(255) NOT NULL,
    MName varchar(255),
    LName varchar(255) NOT NULL,
    Salary decimal(11,4) NOT NULL,
    Sex enum('Male', 'Female') NOT NULL,
    CivilStatus enum('Single', 'Married', 'Widow', 'Separated', 'Annuled') NOT NULL,
    DOB date NOT NULL,
    Email varchar(255) NOT NULL,
    MobileNo char(12) NOT NULL,
    Street varchar(255) NOT NULL,
    Barangay varchar(255) NOT NULL,
    City varchar(255) NOT NULL,
    Province varchar(255) NOT NULL,
    ZipCode char(4) NOT NULL,
    PRIMARY KEY (StaffNo),
    KEY uiBranchNo (BranchNo),
    KEY uiPositionNo (PositionNo),
    KEY uiFullName (FName, MName, LName),
    KEY uiAddress (Street, Barangay),
    KEY uiZipCode (ZipCode),
    FOREIGN KEY (BranchNo) REFERENCES Branch(BranchNo) ON UPDATE CASCADE ON DELETE
    NO ACTION,
    FOREIGN KEY (PositionNo) REFERENCES `Position` (PositionNo) ON UPDATE CASCADE ON
    DELETE NO ACTION
);
```

```
CREATE TABLE LoanApplication (
    ApplicationNo int NOT NULL AUTO_INCREMENT,
    CustomerNo int NOT NULL,
    TermNo tinyint NOT NULL,
    LoanOfficerNo int NOT NULL,
    Amount decimal(11,4) NOT NULL,
    Purpose varchar(255),
    GrossAnnualIncome decimal(11,4) NOT NULL,
    PrefBank varchar(255) NOT NULL,
    DebtRatio decimal(3,0) NOT NULL,
    InitialStatus enum('Approved', 'Rejected') NOT NULL,
    ProcessingFee decimal(9,4) DEFAULT '0.0000',
    PRIMARY KEY (ApplicationNo),
    KEY uiCustomerNo (CustomerNo),
    KEY uiTermNo (TermNo),
    KEY uiLoanOfficerNo (LoanOfficerNo),
    KEY uiAmount (Amount),
    KEY uiGAI (GrossAnnualIncome),
    KEY uiStatus (InitialStatus),
    FOREIGN KEY (CustomerNo) REFERENCES Customer(CustomerNo) ON UPDATE CASCADE ON
    DELETE NO ACTION,
    FOREIGN KEY (TermNo) REFERENCES LoanTerm(TermNo) ON UPDATE CASCADE ON DELETE
    NO ACTION,
    FOREIGN KEY (LoanOfficerNo) REFERENCES Staff(StaffNo) ON UPDATE CASCADE ON DELETE
    NO ACTION
);
```

```
/* Creating Triggers for LoanApplication*/
DELIMITER //
CREATE TRIGGER utiOfficerOnly
BEFORE INSERT ON LoanApplication
FOR EACH ROW
BEGIN
    IF NOT EXISTS (
            SELECT
                    Position.*
            FROM((Staff
                    INNER JOIN LoanApplication ON Staff.StaffNo = LoanApplication.LoanOfficerNo)
            INNER JOIN Position ON Staff.PositionNo = Position.PositionNo)
                    WHERE (NEW.LoanOfficerNo = Staff.StaffNo AND Position.Position = 'Loan Officer'))
  THEN SIGNAL SQLSTATE '45000' SET MESSAGE_TEXT = 'value is not a loan officer';
  END IF;
END //
DELIMITER ;

DELIMITER //
CREATE TRIGGER utuOfficerOnly
BEFORE UPDATE ON LoanApplication
FOR EACH ROW
BEGIN
    IF NOT EXISTS (
            SELECT
                    Position.*
            FROM((Staff
                    INNER JOIN LoanApplication ON Staff.StaffNo = LoanApplication.LoanOfficerNo)
                     INNER JOIN Position ON Staff.PositionNo = Position.PositionNo)
            WHERE (NEW.LoanOfficerNo = Staff.StaffNo AND Position.Position = 'Loan Officer'))
  THEN SIGNAL SQLSTATE '45000' SET MESSAGE_TEXT = 'staff is not a loan officer';
  END IF;
END //
DELIMITER ;
```

```
CREATE TABLE CreditCardDetail (
    ApplicationNo int NOT NULL,
    Issuer varchar(255) NOT NULL,
    CreditLimit int,
    CardNo char(19) NOT NULL,
    Expiry date,
    KEY uiApplicationNo (ApplicationNo),
    FOREIGN KEY (ApplicationNo) REFERENCES LoanApplication(ApplicationNo)
    ON UPDATE CASCADE ON DELETE NO ACTION
);

CREATE TABLE LoanDocumentation (
    DocumentationNo int NOT NULL AUTO_INCREMENT,
    UnderwriterNo int NOT NULL,
    Grade char(2) NOT NULL,
    PRIMARY KEY (DocumentationNo),
    KEY uiUnderwriterNo (UnderwriterNo),
    FOREIGN KEY (UnderwriterNo) REFERENCES Staff(StaffNo) ON UPDATE CASCADE ON DELETE
    NO ACTION
);

/* Creating Triggers for LoanDocumentation*/
DELIMITER //
CREATE TRIGGER utiUnderwriterOnly
BEFORE INSERT ON LoanDocumentation
FOR EACH ROW
BEGIN
    IF NOT EXISTS (
            SELECT
                Position.*
            FROM((Staff
                INNER JOIN LoanDocumentation ON Staff.StaffNo =
                LoanDocumentation.UnderwriterNo)
                INNER JOIN Position ON Staff.PositionNo = Position.PositionNo)
            WHERE (NEW.UnderwriterNo = Staff.StaffNo AND Position.Position = 'Underwriter'))
  THEN SIGNAL SQLSTATE '45000' SET MESSAGE_TEXT = 'staff is not an underwriter';
  END IF;
END //
DELIMITER ;
DELIMITER //
CREATE TRIGGER utuUnderwriterOnly
BEFORE UPDATE ON LoanDocumentation
FOR EACH ROW
BEGIN
    IF NOT EXISTS (
            SELECT
                    Position.*
            FROM((Staff
                    INNER JOIN LoanDocumentation ON Staff.StaffNo =
                    LoanDocumentation.UnderwriterNo)
                     INNER JOIN Position ON Staff.PositionNo = Position.PositionNo)
            WHERE (NEW.UnderwriterNo = Staff.StaffNo AND Position.Position = 'Underwriter'))
  THEN SIGNAL SQLSTATE '45000' SET MESSAGE_TEXT = 'staff is not an underwriter';
  END IF;
END //
DELIMITER ;

CREATE TABLE Decision (
    DecisionNo int NOT NULL AUTO_INCREMENT,
    DocumentationNo int NOT NULL,
    ApplicationNo int NOT NULL,
    Decision enum('Approved', 'Suspended', 'Rejected') NOT NULL,
    TotalPmt decimal(11,4) NOT NULL,
    MonthlyPmt decimal(11,4) NOT NULL,
    InterestRate decimal(5,4) NOT NULL,
    LendingLimit decimal(11,4) NOT NULL,
    PRIMARY KEY (DecisionNo),
    KEY uiDocumentationNo (DocumentationNo),
    KEY uiApplicationNo (ApplicationNo),
    KEY uiDecision (Decision),
    KEY uiTotalPmt (TotalPmt),
    KEY uiMonthlyPmt (MonthlyPmt),
    FOREIGN KEY (DocumentationNo) REFERENCES LoanDocumentation(DocumentationNo) ON
    UPDATE CASCADE ON DELETE NO ACTION,
    FOREIGN KEY (ApplicationNo) REFERENCES LoanApplication(ApplicationNo) ON UPDATE
    CASCADE ON DELETE NO ACTION
);
```

```
CREATE TABLE Disbursement (
    DisbursementNo int NOT NULL AUTO_INCREMENT,
    ApplicationNo int NOT NULL,
    ProvidedAccNo char(19) NOT NULL,
    Amount decimal(11,4) NOT NULL,
    DateDisbursed date NOT NULL,
    PRIMARY KEY (DisbursementNo),
    KEY uiApplication (ApplicationNo),
    FOREIGN KEY (ApplicationNo) REFERENCES LoanApplication(ApplicationNo) ON UPDATE
    CASCADE ON DELETE NO ACTION
);

CREATE TABLE Document (
    DocumentNo int NOT NULL AUTO_INCREMENT,
    CustomerNo int NOT NULL,
    LoanProcessorNo int NOT NULL,
    DocumentationNo int NOT NULL,
    Name varchar(255) NOT NULL,
    DateSubmitted date NOT NULL,
    UNIQUE KEY uuiDocumentNo (DocumentNo),
    KEY uiCustomerNo (CustomerNo),
    KEY uiLoanProcessorNo (LoanProcessorNo),
    KEY uiDocumentationNo (DocumentationNo),
    KEY uiName (Name),
    FOREIGN KEY (CustomerNo) REFERENCES Customer(CustomerNo) ON UPDATE CASCADE ON
    DELETE NO ACTION,
    FOREIGN KEY (LoanProcessorNo) REFERENCES Staff(StaffNo) ON UPDATE CASCADE ON
    DELETE NO ACTION,
    FOREIGN KEY (DocumentationNo) REFERENCES LoanDocumentation(DocumentationNo) ON
    UPDATE CASCADE ON DELETE NO ACTION
);
```

```
/* Creating Triggers for Document*/
DELIMITER //
CREATE TRIGGER utiLoanProcessorOnly
BEFORE INSERT ON Document
FOR EACH ROW
BEGIN
    IF NOT EXISTS (
            SELECT
                    Position.*
            FROM((Staff
                    INNER JOIN Document ON Staff.StaffNo = Document.LoanProcessorNo)
                    INNER JOIN Position ON Staff.PositionNo = Position.PositionNo)
            WHERE (NEW.LoanProcessorNo = Staff.StaffNo AND Position.Position = 'Loan Processor'))
  THEN SIGNAL SQLSTATE '45000' SET MESSAGE_TEXT = 'staff is not a loan processor';
  END IF;
END //
DELIMITER ;

DELIMITER //
CREATE TRIGGER utuLoanProcessorOnly
BEFORE UPDATE ON Document
FOR EACH ROW
BEGIN
    IF NOT EXISTS (
            SELECT
                    Position.*
            FROM((Staff
                    INNER JOIN Document ON Staff.StaffNo = Document.LoanProcessorNo)
                    INNER JOIN Position ON Staff.PositionNo = Position.PositionNo)
            WHERE (NEW.LoanProcessorNo = Staff.StaffNo AND Position.Position = 'Loan Processor'))
  THEN SIGNAL SQLSTATE '45000' SET MESSAGE_TEXT = 'staff is not a loan processor';
  END IF;
END //
DELIMITER ;
```

```
CREATE TABLE EmploymentDetail (
    ApplicationNo int NOT NULL,
    EmploymentType enum('Governemnt', 'Private', 'Self-Employed') NOT NULL,
    CompanyName varchar(255),
    Occupation varchar(255) NOT NULL,
    Street varchar(255) NOT NULL,
    Barangay varchar(255) NOT NULL,
    City varchar(255) NOT NULL,
    Province varchar(255) NOT NULL,
    ZipCode char(4) NOT NULL,
    HROfficeNo char(12),
    YearsEmployed tinyint(2) NOT NULL,
    KEY uiApplicationNo (ApplicationNo),
    FOREIGN KEY (ApplicationNo) REFERENCES LoanApplication(ApplicationNo) ON UPDATE
    CASCADE ON DELETE NO ACTION
);

CREATE TABLE ExistingLoanDetail (
    ApplicationNo int NOT NULL,
    CompanyName varchar(255) NOT NULL,
    LoanType varchar(255) NOT NULL,
    MonthlyPmt decimal(11,4) NOT NULL,
    RemainingBal  decimal (11,4) NOT NULL,
    KEY uiApplicationNo (ApplicationNo),
    FOREIGN KEY (ApplicationNo) REFERENCES LoanApplication(ApplicationNo) ON UPDATE
    CASCADE ON DELETE NO ACTION
);
```

```
CREATE TABLE Repayment (
    RepaymentNo int NOT NULL AUTO_INCREMENT,
    ApplicationNo int NOT NULL,
    ScheduledDate date NOT NULL,
    MonthlyPmt decimal(11,4) NOT NULL,
    PenaltyAmount decimal(8,4) NOT NULL,
    TotalAmount decimal(11,4) NOT NULL,
    PmtMethod varchar(255) NOT NULL,
    DatePaid date NOT NULL,
    PRIMARY KEY (RepaymentNo),
    KEY uiApplicationNo (ApplicationNo),
    KEY uiScheduledDate (ScheduledDate),
    FOREIGN KEY (ApplicationnO) REFERENCES LoanApplication(ApplicationNo) ON UPDATE
    CASCADE ON DELETE NO ACTION
);

/* Creating Stored Procedures for the Database*/
DELIMITER //
CREATE PROCEDURE uspCustomerReport(InputtedCustomerNo int)
BEGIN
    SELECT
            CustomerNo AS 'Customer No',
            CONCAT_WS(' ', FName, MName, LName) AS 'Full Name',
            UMIDNo AS 'UMID No',
            TIN AS 'TIN',
            Sex AS 'Sex',
            CONCAT_WS(' ', Street, Barangay, City, Province, ZipCode) AS 'Address',
            DOB AS 'DOB'

    FROM Customer
    WHERE (CustomerNo = InputtedCustomerNo);
END //
DELIMITER ;
DELIMITER //
CREATE PROCEDURE uspLoanDetailsReport(CustomerNo int)
BEGIN
    SELECT
            LoanApplication.CustomerNo AS 'Customer No',
            LoanTerm.TermInMonths AS 'Loan Term',
            LoanApplication.Amount AS 'Loan Amount',
            COALESCE(Decision.InterestRate, 0) AS 'Interest Rate',
            CONCAT_WS(' ', Customer.FName, Customer.MName, Customer.LName) AS 'Full Name',
            CONCAT_WS(' ', Customer.Street, Customer.Barangay, Customer.City, Customer.Province,
            Customer.ZipCode) AS 'Address',
            Customer.MobileNo AS 'Contact No'
    FROM (((LoanApplication
            INNER JOIN LoanTerm ON LoanApplication.TermNo = LoanTerm.TermNo)
            LEFT JOIN Decision ON LoanApplication.ApplicationNo = Decision.ApplicationNo)
            INNER JOIN Customer ON LoanApplication.CustomerNo = Customer.CustomerNo)
    WHERE (LoanApplication.CustomerNo = CustomerNo);
END //
DELIMITER ;

DELIMITER //
CREATE PROCEDURE uspReceivedPaymentReport(CustomerNo int)
BEGIN
    SELECT
            Repayment.RepaymentNo AS 'Receipt No',
            LoanApplication.CustomerNo AS 'Customer No',
            Repayment.DatePaid AS 'Receipt Date',
            Repayment.ScheduledDate AS 'Scheduled Date',
            Repayment.TotalAmount AS 'Repayment Amount',
            Repayment.PenaltyAmount AS 'Penalty Charge',
            Repayment.TotalAmount AS 'Total Amount'
    FROM (Repayment RIGHT JOIN LoanApplication ON Repayment.ApplicationNo =
            LoanApplication.ApplicationNo)
    WHERE (LoanApplication.CustomerNo = CustomerNo AND Repayment.RepaymentNo IS NOT NULL);
END //
DELIMITER ;
```

```
/* Creating Views for the Database*/
CREATE VIEW uvActiveLoan AS
SELECT
    Customer.CustomerNo AS 'Customer No',
    CONCAT_WS(' ', Customer.FName, Customer.MName, Customer.LName) AS 'Full Name',
    LoanApplication.ApplicationNo AS 'Application No',
     Disbursement.DateDisbursed AS 'Start Date',
    (SELECT (Disbursement.DateDisbursed + INTERVAL LoanTerm.TermInMonths MONTH)) AS 'End Date'
FROM (((LoanApplication
    INNER JOIN Customer ON LoanApplication.CustomerNo = Customer.CustomerNo)
    LEFT JOIN Disbursement ON LoanApplication.ApplicationNo = Disbursement.ApplicationNo)
    INNER JOIN LoanTerm ON LoanApplication.TermNo = LoanTerm.TermNo)
WHERE (Disbursement.DateDisbursed + INTERVAL LoanTerm.TermInMonths MONTH) > NOW();

CREATE VIEW uvStaffDetails AS
SELECT
    CONCAT_WS(' ', FName, MName, LName) AS 'Full Name',
    CONCAT_WS(' ', Street, Barangay, City, Province, ZipCode) AS 'Address',
    'Position' AS 'Position',
    Sex AS 'Sex'
FROM Staff
```

# REFERENCES

Begg, C., & Connolly, T. (2005). Database Systems: A Practical Approach to Design, Implementation and Management (Fourth Edition). Addison-Wesley.

Cinergix Pty Ltd (n.d.) Creately: Chart, Diagram & Visual Canvas Software. creately.com

**CURRICULUM VITAE**

## Personal Background

**Name:** Kenith S. Lumantas

**Age:** 21

**Nickname:** Ken-Ken

**Address:** Bataan, Lapu-lapu Street, Digos City, Davao del Sur

**Mobile Number:** 09560315143

**Email:** kslumantas@usep.edu.ph

## Educational Background

**Elementary School:** Ramon Magsaysay Central Elementary School **Year Graduated:** 2011 – 2012

**Junior High School:** Digos City National High School **Year Graduated:** 2016 – 2017

**Senior High School:** Senior High School in Digos City **Year Graduated:** 2018 – 2019

**College:** University of Southeastern Philippines

**Course Taken:** Bachelor of Science in Computer Science

**Motto in Life:** Keep all things simple

**Achievements:** NCII passer

**Hobbies:** Reading novels and playing games

**Sports:** Volleyball

**Likes/Dislikes:** I like simple things and exploring.

# CURRICULUM VITAE

## Personal Background

**Name:** Cyrelle John A. Domingo

**Age:** 21

**Nickname:** Cy

**Address:** Uphal Village, Lizada, Toril, Davao City

**Mobile Number:** 09673574943

**Email:** cjadomingo@usep.edu.ph

## Educational Background

**Elementary School:** Don Juan Dela Cruz Central Elementary School **Year Graduated:** 2011 – 2012

**Junior High School:** Doña Carmen Denia National High School **Year Graduated:** 2016 – 2017

**Senior High School:** Brokenshire College Toril **Year Graduated:** 2018 – 2019

**College:** University of Southeastern Philippines

**Course Taken:** Bachelor of Science in Computer Science

**Motto in Life:** Patience is a virtue

**Hobbies:** Playing video games

**Sports:** Frisbee

**Likes/Dislikes:** I like playing any genre of video games.

# CURRICULUM VITAE

## Personal Background

**Name:** Edgardo V. Amigo

**Age:** 21

**Nickname:** Ed

**Address:** Tibanban, Governor Generoso, Davao Oriental

**Mobile Number:** 09121850764

**Email:** evedgardo @usep.edu.ph

## Educational Background

**Elementary School:** Enrique Orencia Elementary School

**Year Graduated:** 2012 – 2013

**Junior High School:** Sigaboy Agricultural Vocational High School

**Year Graduated:** 2016 – 2017

**Senior High School:** Sigaboy Agricultural Vocational High School

**Year Graduated:** 2018 – 2019

**College:** University of Southeastern Philippines

**Course Taken:** Bachelor of Science in Computer Science

**Motto in Life:** Where there's a will, there's a way

**Achievements:** CSS NCII Passer

**Hobbies:** Biking

**Sports:** Badminton

**Likes/Dislikes:** I like to explore a new place