SRS for Smart Banking Chatbot

1.1 Aim of the project

The aim of this project is that to provide all the information of bank as well as their account details to the registered user by asking some question to the bot, so no need to go to the bank for any inquiry or to call a customer care.

1.2 Project Scope

Chatbots improve human interaction with systems by giving a response based on the user input. Unlike the traditional banking methods, Chatbots can bring in a better and faster user experience providing 24×7 intelligent customer service.

1.3 Project Objective

Customers can make any interaction easily with bank from creating a bank account to the asking any questions (regarding to the bank).

1.4 Project Modules:

Customer Modules:

- Login / Registration
- Bot Chat
- Activity log
- Finder
- Status
- Feedback

Admin Modules:

- Login
- Verification of customers
- Customer bank account provider
- Feedback provider

1.5 Project basic requirements:

1.5.1 Software requirements:

- Windows Server 2008(SP2-64 bit edition)
- SQL Server 2014
- PHP (v5.6.32)
- Python (v3.6.3) and Libraries required for AI and Natural Language Processing(NLP)
- Bootstrap (v3.3.7)
- AngularJS (v1.6.4)
- jQuery (v1.10.2)
- Node.js (v8.10.0)

1.5.2 Hardware requirements:

- Combined Database server and Web server

- 8 cores (Intel Xeon E5504 or comparable CPU)
- 32 GB of RAM
- 550 GB hard disk space for the operating system, SQL Server, and all databases files.

- Web-server have an underlying operating system that manages database server and web-application.

1.6 Project Feasibility Study

1.61 Technical Feasibility:

The work for the project can be undertaken with the existing computers and servers, software technologies.

1.62 Operational Feasibility:

It is possible for the customer to use functions like update their bank account details (with specific verification), send amount of rupees to the other bank account (with specific details), see their own bank amount, etc. with talking to the Jon Snow (Chat Bot).

1.6.3 Implementation Feasibility:

Implementing our project is very feasible as it is easily uploaded and managed on web-server and database server and then from customer's point of view also it is very feasible since they just have to enter questions regarding to their query.

1.6.4 Economic Feasibility:

For the customers asking any questions, it is completely free for month; and they will have to pay for the assistant (Jon snow - The Chat Bot) as they use.

1.6.5 Resource Feasibility:

This project uses hardware and all latest software resources that are easily available and can be integrated very easily and effectively.

1.6.6. Market Feasibility:

Customers don't have to be panic. They can just ask any questions to the assistant.

1.7 Detailed Modules Description

Customer Modules:

1. Login / Registration: Customers may register on the system and use the services provided by Jon Assistant.
2. Bot Chat: Customers can chat with Jon Assistant as if talking to a real banking operator. It provides services:
   - Services related to the banking cards
   - Services related to the transaction
   - Services related to make request for the cards and loans
   - Services related to the account security
   - View the activities on the account
3. Tracing of transactions: Customers can get information about transactions they have made.
4. Finder: This module is work as a finding mechanism of the bank. Customers can find the location of the branch or ATM of the bank.

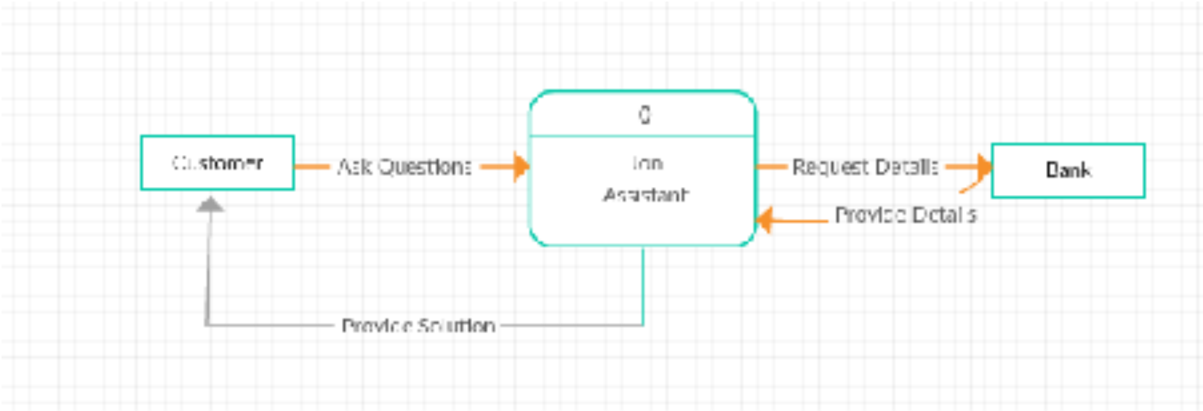
5. Status: Customers can see the status of the requests they have made. It can be in three modes: Pending, Approved, or Rejected.

6. Feedback: Customers can always make feedbacks to the bank based on how the services are working.

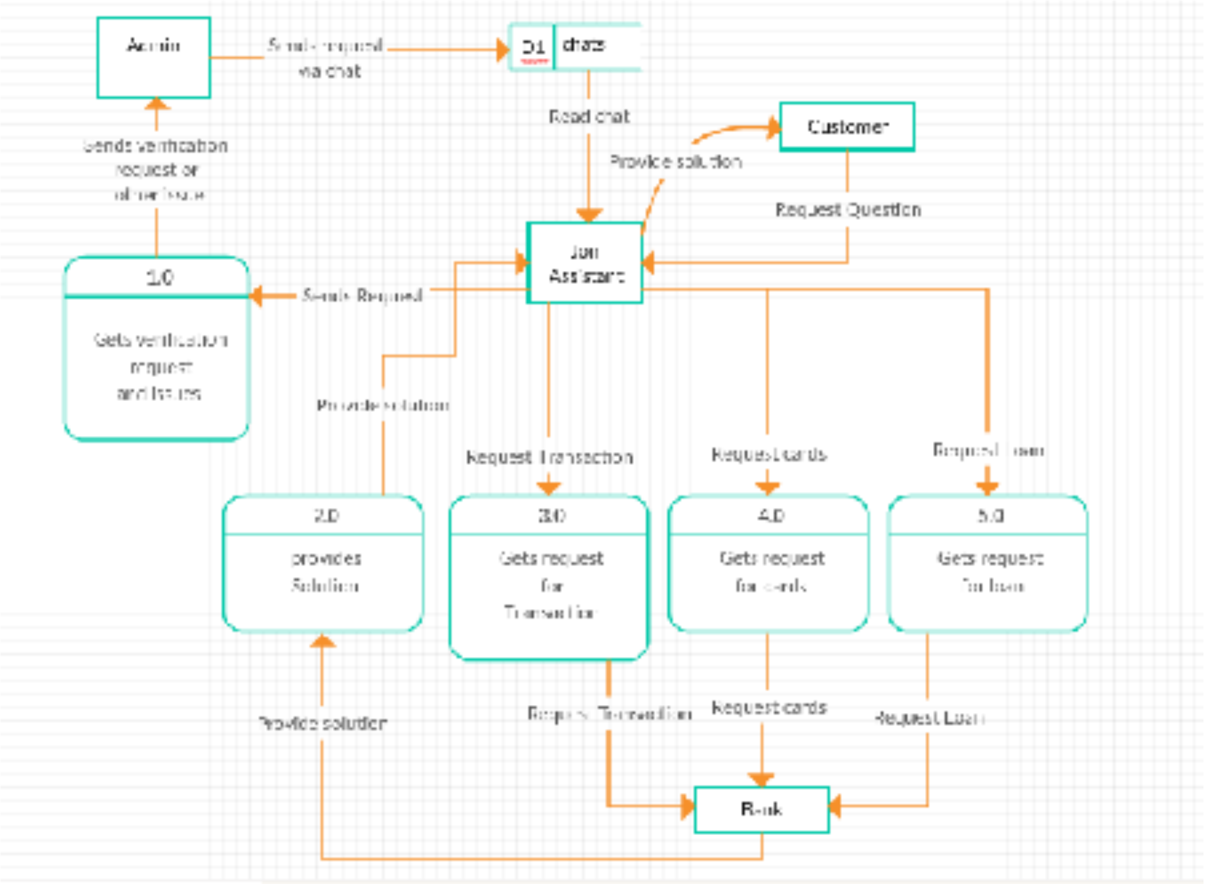
Admin Modules:

- Login: Admin may login into the system to provide services.
- Verification of customers: Admin can verify a request from the Jon Assistant, which was actually received from the customers at the time of conversation.
- Customer bank account provider: After successful verification of details provided by customer, Admin will provide customer bank account details at the customer panel.
- Feedback provider: Admin can receive the feedback from the customers and can see the defect in the system. By using this information, admin can enhance the system as customers wants.

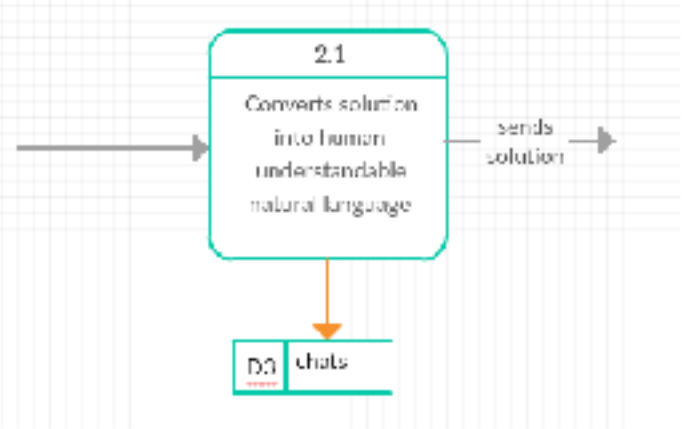
1.7.1 Dataflow Diagram:

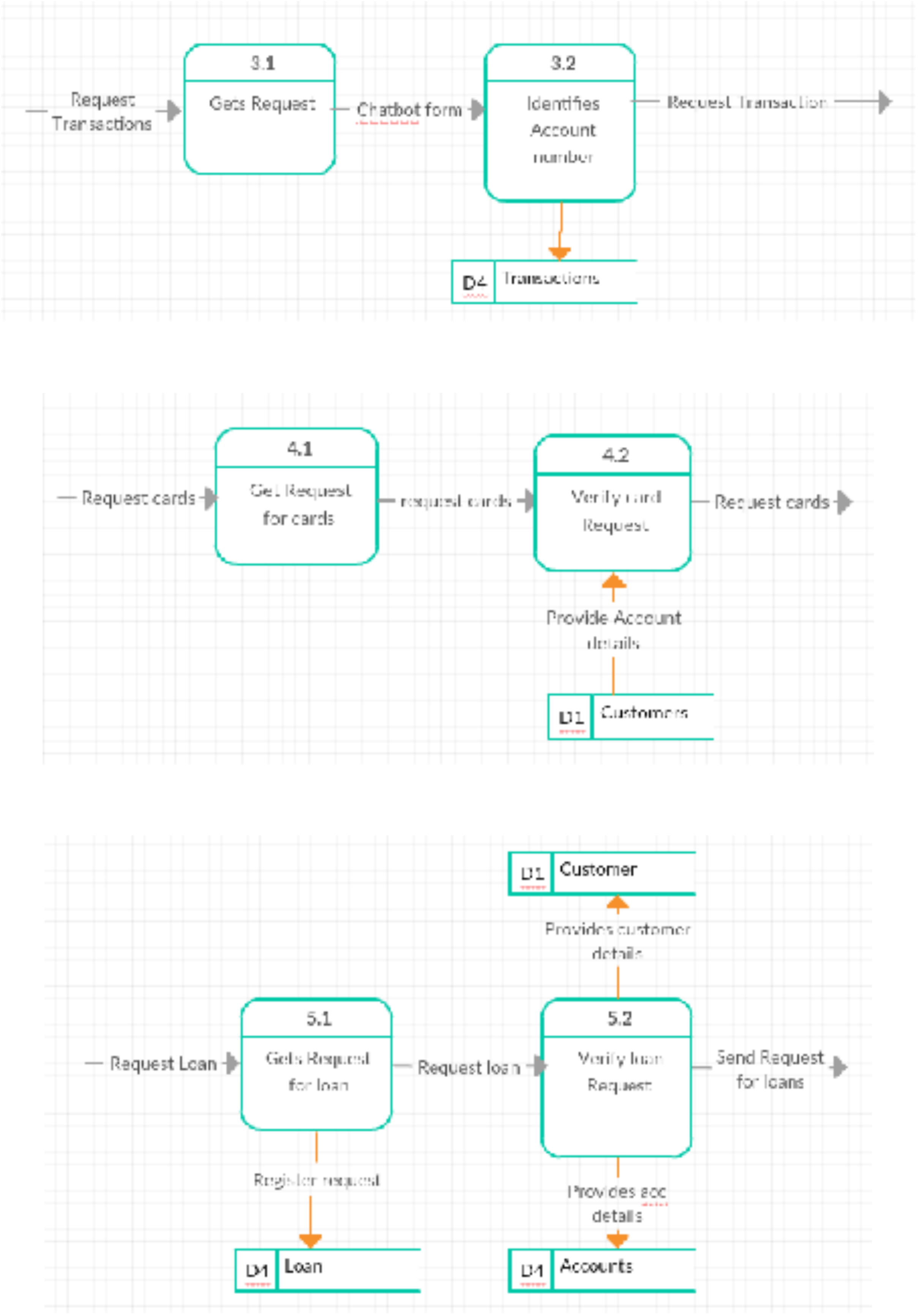


Level 0


Admin
Sends request via chat
D1
chats
Read chat
Customer
Provide solution
Sends verification request or other issue
Request Question
Bot Assistant
1.0
Gets verification request and Issues
Sends Request
Provide solution
Request Transaction
Request cards
Request Loan
2.0
provides Solution
3.0
Gets request for Transaction
4.0
Gets request for cards
5.0
Gets request for loan
Provide solution
Request Transaction
Request cards
Request Loan
Bank

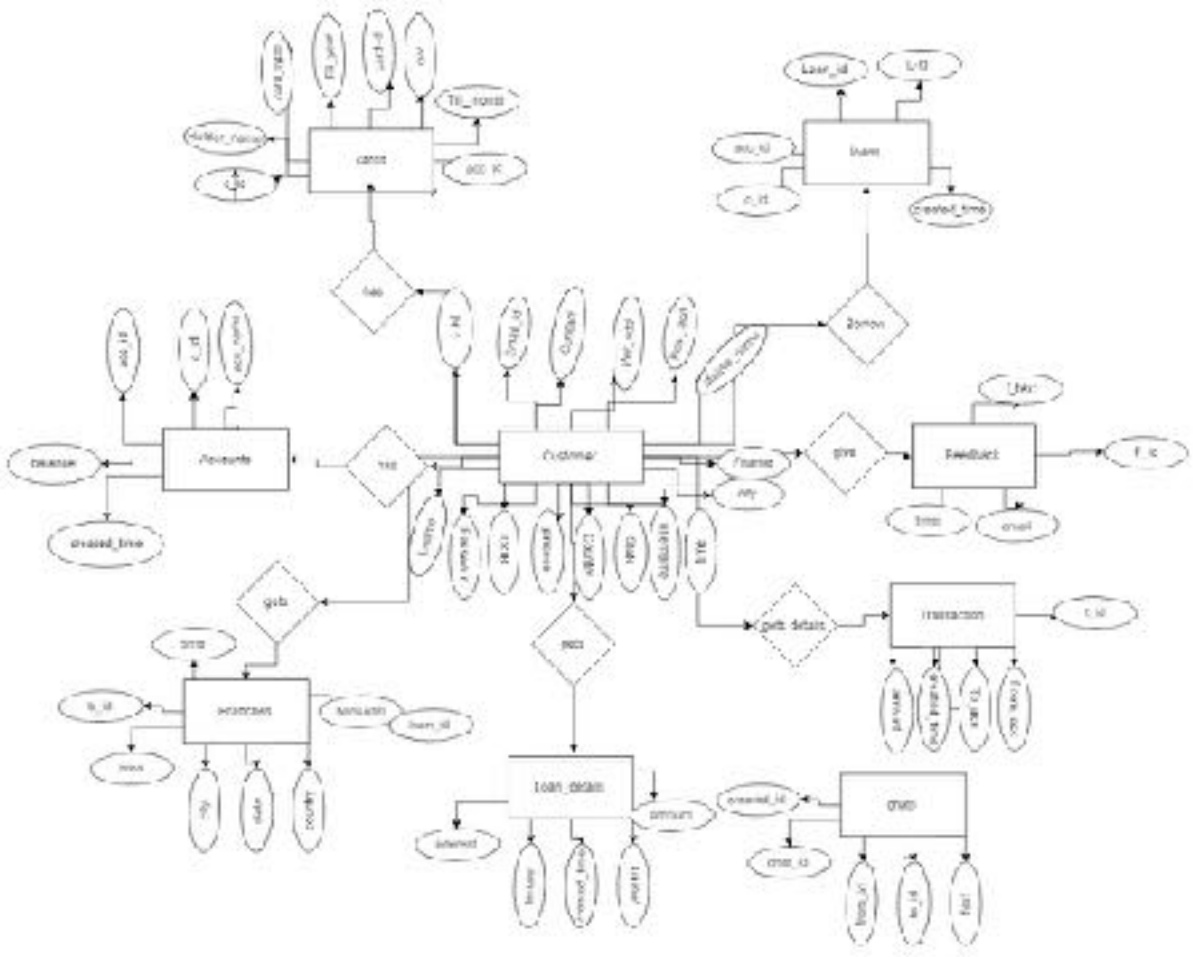
Level 1


2.1
Converts solution into human understandable natural language
sends solution
D1
chats

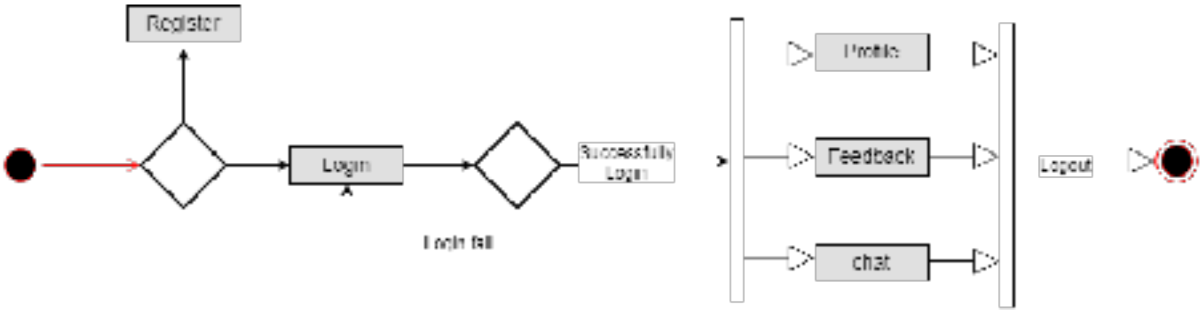

3.1
Gets Request
Request Transactions
Chatbot form
3.2
Identifies Account number
Request Transaction
D4
Transactions
4.1
Get Request for cards
Request cards
request cards
4.2
Verify card Request
Request cards
Provide Account details
D1
Customers
D1
Customer
Provides customer details
5.1
Gets Request for loan
Request Loan
Request loan
5.2
Verify loan Request
Send Request for loans
Register request
Provides acc details
D4
Loan
D4
Accounts

Level 2

1.7.2 Entity Relationship Diagram:



1.7.3 State Diagram:



1.8 Database design

What is database design?

Database design is the process of producing a detailed data model of a database. This data model contains all the needed logical and physical design choices and physical storage parameters needed to generate a design in a data definition language, which can then be used to create a database. A fully attributed data model contain detailed attributes for each entry.

Database design for Smart Banking Chatbot:

Database name	: minor_project
No of tables	: 9
Table name	: customers, branches, accounts, cards, loans, loan_details, feedback, transaction, chats

Database dictionary:

Sr. no	Name	Data type	Constraints	Description
1	cid	int(100)	AUTO_INCREMENT, PRIMARY KEY	To store auto increment value customer id
2	username	varchar(30)	UNIQUE_KEY	To store customer username
3	fname	varchar(20)	NOT NULL	To store the customer first name
4	lname	varchar(20)	NOT NULL	To store the customer last name

5	email	varchar(40)	UNIQUE_KEY	To store the customer email id
6	password	varchar(20)	NOT NULL	To store the customer password
7	contact	varchar(20)	NOT NULL	To store the customer contact no.
8	post_add	varchar(50)	NOT NULL	To store the customer postal add.
9	per_add	varchar(50)	NOT NULL	To store the customer permanent add.
10	city	varchar(20)	NOT NULL	To store the customer city name
11	state	varchar(20)	NOT NULL	To store the customer's state
12	country	varchar(20)	NOT NULL	To store the customer's country name
13	middle_name	varchar(20)	NOT NULL	To store the customer's middle name
14	pincode	varchar(20)	NOT NULL	To store the customer's address pincode
15	gender	varchar(20)	NOT NULL	To store the customer's gender
16	dob	date	NOT NULL	To store the customer's date of birth

17	time	datetime	DEFAULT CURRENT_TIMESTAMP	To store the current date and time
18	created_time	Datetime	DEFAULT CURRENT_TIMESTAMP	To store the date and time of entry made in table
19	jon_service	int(1)	DEFAULT 0	To store the bit which indicates customer has started Jon service or not

Table 1. customers

Sr. no	Name	Data type	Constraints	Description
1	bid	int(100)	AUTO_INCREMENT	To store the auto increment value bank id
2	area	varchar(30)	PRIMARY_KEY	To store the area of bank
3	city	varchar(20)	NOT NULL	To store the city name of bank
4	state	varchar(20)	NOT NULL	To store the state name of bank
5	country	varchar(20)	NOT NULL	To store the country of bank

6	branch_atm	varchar(6)	NOT_NULL	To store the value which indicates that this row is for branch or atm
7	time	datetime	DEFAULT CURRENT_TIMESTAMP	To store the current date and time

Table 2. branch

Sr. no	Name	Data type	Constraints	Description
1	acc_id	int(100)	AUTO_INCREMENT, PRIMARY_KEY	To store the auto increment value customer account id
2	c_id	int(30)	NOT NULL, FOREIGN KEY REFERENCES customers(cid)	To store id of customer
3	acc_name	varchar(20)	NOT NULL	To store the account name of customer
4	acc_type	varchar(20)	NOT NULL	To store the type of the account
5	balance	int(20)	NOT NULL	To store the account balance of customer
6	created_time	Timestamp	DEFAULT CURRENT_TIMESTAMP	To store the date and time of entry made in table

Table 3: accounts

Sr. no	Name	Data type	Constraints	Description
1	card_id	int(20)	AUTO_INCREMENT , PRIMARY_KEY	To store the auto increment value card id
2	c_id	int(30)	NOT NULL, FOREIGN KEY REFERENCES customers(cid)	To store the id of customer
3	acc_id	int(30)	NOT NULL, FOREIGN KEY REFERENCES accounts(acc_id)	To store the customer account id
4	holder_name	varchar(20)	NOT NULL	To store the account holder name
5	till_month	Int(10)	NOT NULL	To store the expiration month
6	till_year	Int(10)	NOT_NULL	To store the expiration year
7	cvv	int(10)	NOT_NULL	To store the cvv number of card
8	card_type	varchar(20)	NOT-NULL	To store the type of card

Table 4: cards

Sr. no	Name	Data type	Constraints	Description
1	f_id	int(10)	AUTO_INCREMENT, PRIMARY_KEY	To store the auto increment value of feedback id
2	email	varchar(30)	NOT_NULL	To store the email id of customer
3	f_text	varchar(50)	NOT NULL	To store the customer feedback
4	time	datetime	DEFAULT CURRENT_TIMESTAMP	To store the time of feedback added

Table 5: feedback

Sr. no	Name	Data type	Constraints	Description
1	l_id	int(10)	AUTO_INCREMENT, PRIMARY_KEY	To store the auto increment value of l_id
2	loan_id	int(10)	NOT_NULL, FOREIGN KEY REFERENCES loan_details(loan_id)	To store the loan_id of loan from loan_details table
3	acc_id	int(10)	NOT NULL, FOREIGN KEY REFERENCES accounts(acc_id)	To store the account id of customer

4	c_id	int(10)	NOT NULL, FOREIGN KEY REFERENCES customers(cid)	To store id of customer
5	created_time	datetime	DEFAULT CURRENT_TIMESTAMP	To store the date and time when loan was taken

Table 6: loans

Sr. no	Name	Data type	Constraints	Description
1	loan_id	int(10)	AUTO_INCREMENT, PRIMARY_KEY	To store the auto increment value of loan id
2	amount	varchar(10)	NOT_NULL	To store the amount of loan
3	ianterest	varchar(10)	NOT NULL	To store the interest of loan
4	tenure	varchar(10)	NOT NULL	To store the tenure of loan
5	created_time	datetime	CURRENT_TIMESTAMP	To store the date and time when loan was taken

Table 7: Loan_details

Sr. no	Name	Data type	Constraints	Description

1	t_id	int(10)	AUTO_INCREMENT, PRIMARY_KEY	To store the auto increment value of transaction id
2	from_acc	int(10)	NOT_NULL, FOREIGN KEY REFERENCES accounts(acc_id)	To store the account id from which amount is debited to other customer
3	to_acc	int(10)	NOT NULL, FOREIGN KEY REFERENCES accounts(acc_id)	To store the account id where amount is credited from other customer
4	amount	int(10)	NOT NULL	To store the transfer amount
5	created_time	datetime	DEFAULT CURRENT_TIMESTAMP	To store the date and time when transaction was performed

Table 8: transaction

Sr. no	Name	Data type	Constraints	Description
1	chat_id	int(10)	AUTO_INCREMENT,	To store the auto increment value of chat_id
2	from_id	int(10)	NOT_NULL	To store id from which message was sent

3	to_id	int(10)	NOT NULL	Stores id, to indicate id has received message
4	text	varchar(100)	NOT NULL	To store the text of chat
5	created_time	datetime	DEFAULT CURRENT_TIMESTAMP	To store the date and time when the message was sent

Table 9: chats