

Bangladesh University of Engineering and Technology

Department of Computer Science and Engineering

CSE408: Software Development Project

Machine Learning Based Credit Risk Prediction System

Submitted by:

Nishat Anjum Bristy (1505007)
Md Salman Shamil (1505021)
Syeda Nahida Akter Tisha (1505027)

Under the supervision of:

Dr. A. B. M. Alim Al Islam
Dr. Muhammad Abdullah Adnan
Dr. Md. Shamsuzzoha Bayzid

Contents

1 Vision Statement	3
1.1 Project Goals	3
1.2 Project Scope	3
1.3 High-level Features	3
1.4 Milestones and Deliverables	4
2 Architecture and Design	7
2.1 Use case diagram	7
2.2 Class Diagram	11
2.3 Sequence Diagram	14
2.4 Data Flow Diagram	16
3 User Guide	20
3.1 Client End	20
3.1.1 Registration	20
3.1.2 Login	21
3.1.3 Home Page	21
3.1.4 Edit Profile Page	21
3.1.5 Apply For Loan	21
3.1.6 Send Messages	21
3.2 Employee End	22
3.2.1 Registration	23
3.2.2 Login	23
3.2.3 Profile	24
3.2.4 Employee Workspace	24
3.2.5 Data Visualization with Graphs	25
3.2.6 To-do List	30
3.3 Apply For Loan and Grant	31
3.3.1 Apply For Loan	31
3.3.2 Form Reviewer	33
3.3.3 Negotiator	35
3.3.4 CRM	37
4 Machine Learning models and platform :	38
5 System Documentation	39
5.1 Installation	40
6 Function Documentation	43
6.1 Polls	43
6.1.1 polls/urls.py :	44
6.1.2 polls/views.py :	44
6.1.3 polls/models.py :	45

6.2	Employee	47
6.2.1	employee/urls.py	47
6.2.2	employee/views.py	47
6.2.3	Form Reviewer :	48
6.2.4	Negotiator :	48
6.2.5	Client Relations Manager(CRM):	49
7	Collaboration with Bank Asia and Our Future Work	49

1 Vision Statement

Issuing and processing loans is a major revenue generator for financial organizations. But the assessment of credit risk is very crucial for these organizations for reducing their capital loss. Our project aims at predicting credit risk at the time of loan granting process through machine learning based approach. We have created a system which can be incorporated with the existing banking systems to improve their loan management process.

1.1 Project Goals

This project aims at solving the problem of predicting trustworthiness of a client who has applied for a loan. We also provide a banking software system integrating our machine learning model to it. Our system is hosted as a website and it has two classes of users.

- **Clients:** Clients have active accounts in the bank and uses the website to apply for loan.
- **Employees:** There are three types of employees who process the loan application. *Form reviewer* reviews the application primarily, *Negotiator* negotiates about loan amount with the client and *Client Relations Manager* verifies the information provided by the client and finalizes the loan.

Primary goals of our project are

- to provide basic functionalities on the website for clients and employees.
- to implement a machine learning algorithm and training it with past records to predict trustworthiness of new applicants.
- to integrate the machine learning system with our website so that the negotiator can see the predicted credit risk before negotiating amounts with the client.

We also aim to provide additional features like messaging between users to make communication easier, visualization of data for employees for better understanding of the trend in loan applications, necessary information about the bank and branches etc.

1.2 Project Scope

Our project focuses on the features of a banking software that are needed for loan processing. From the application stage at user end to the finalization stage at employee end.

This project does not aim to implement ordinary banking software to provide general facilities. It rather includes the necessary and desirable features for loan granting process.

1.3 High-level Features

The essential features of our project which provides the basic functionality and workflow are as follows-

- an authentication system for clients and employees to use the system with appropriate access.
- a user-friendly loan application form for client to apply for loan online.

- tables listing all the applicants and their applications and other information at a specific stage of application processing for specific type of employees.
- rejection options at each stage as rejection can happen at any stage of processing.
- manual form edit access for form reviewer if there is any field where the value provided by the client cannot be processed easily in the next steps.
- finalization option for Client Relations Manager and saving to past loans.

The features mentioned above are necessary for the system to be working as a loan management system. Besides, the desirable features of our project are stated below-

- messaging integrated to the system for easier communication.
- a fully functioning client subsystem with bank information and other pages.
- a visualization of data through graphs from past records in the employee end to provide a better understanding of recent trend in loan applications.
- email notification to the applicant on final decision over the application.
- a history table for employee subsystem with past loan applications and decisions by our system.

1.4 Milestones and Deliverables

The implementation of our project was done under CSE408: Software Development course. Project deliverables were designed as follows:

Week	Tasks
Week 3	<ul style="list-style-type: none"> • Basic UI design for client subsystem(Login page, Client Homepage, Loan application form) • Database design and data entry
Week 4	<ul style="list-style-type: none"> • Form reviewer subsystem implementation • Preparing data-set for machine learning
Week 5	<ul style="list-style-type: none"> • Negotiator subsystem implementation • Adding messaging service
Week 6	<ul style="list-style-type: none"> • ML model design • Feature extraction, ranking features and adding additional features
Week 7	<ul style="list-style-type: none"> • ML model design • Preparing training set and test set • Determining best model for classification
Week 8	<ul style="list-style-type: none"> • ML model design • Testing and validating models
Week 9	<ul style="list-style-type: none"> • Client Relations Manager subsystem implementation • Graphical data visualization for employees
Week 10	<ul style="list-style-type: none"> • Integrating UI subsystems • Additional general pages after integration
Week 11	<ul style="list-style-type: none"> • Integrating the ML prediction model • System Testing
Week 12	<ul style="list-style-type: none"> • Beta testing • Address issues from testing

2 Architecture and Design

2.1 Use case diagram

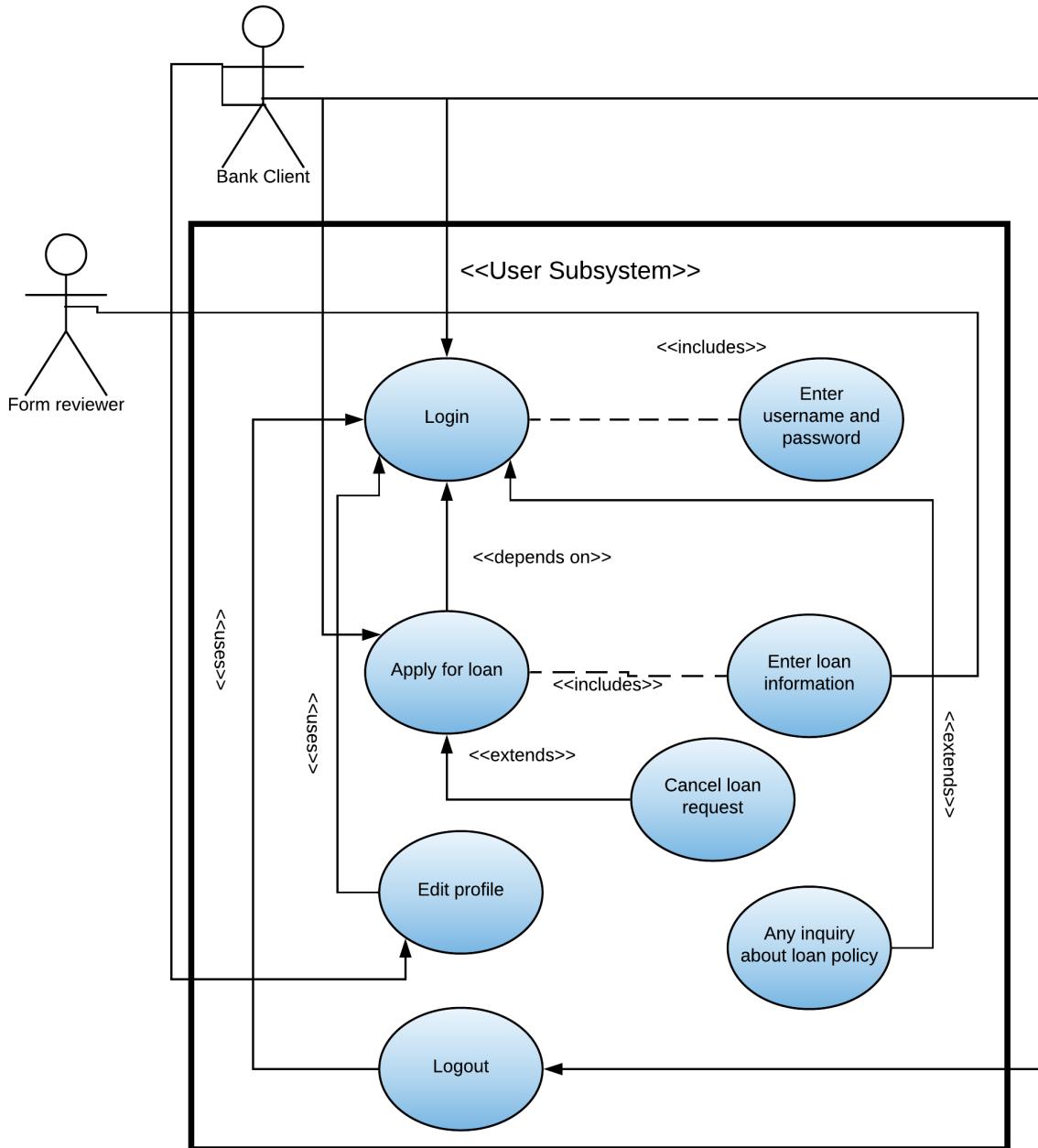


Figure 1: Use case diagram for *User* subsystem

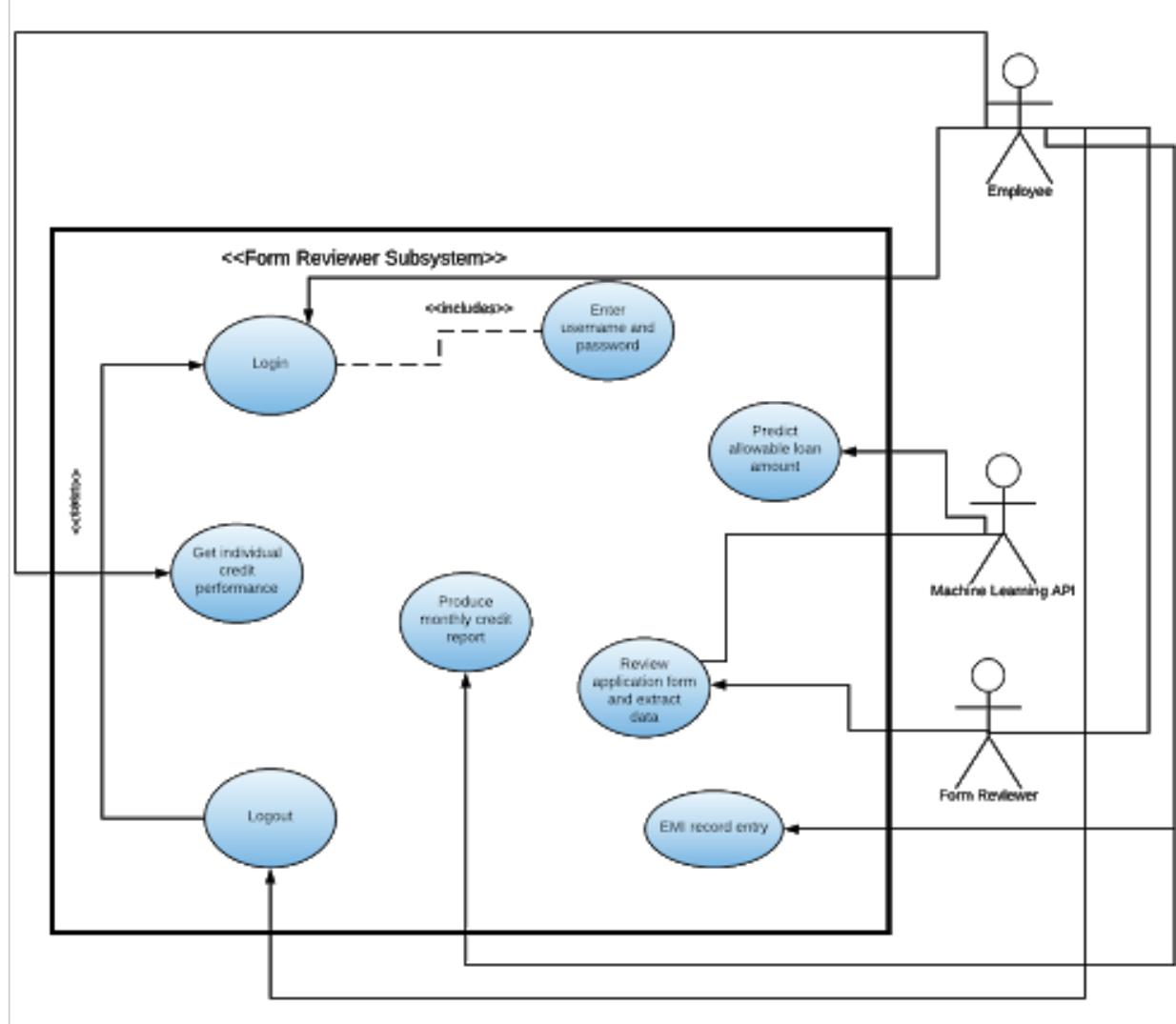


Figure 2: Use case diagram for *Form Reviewer* subsystem

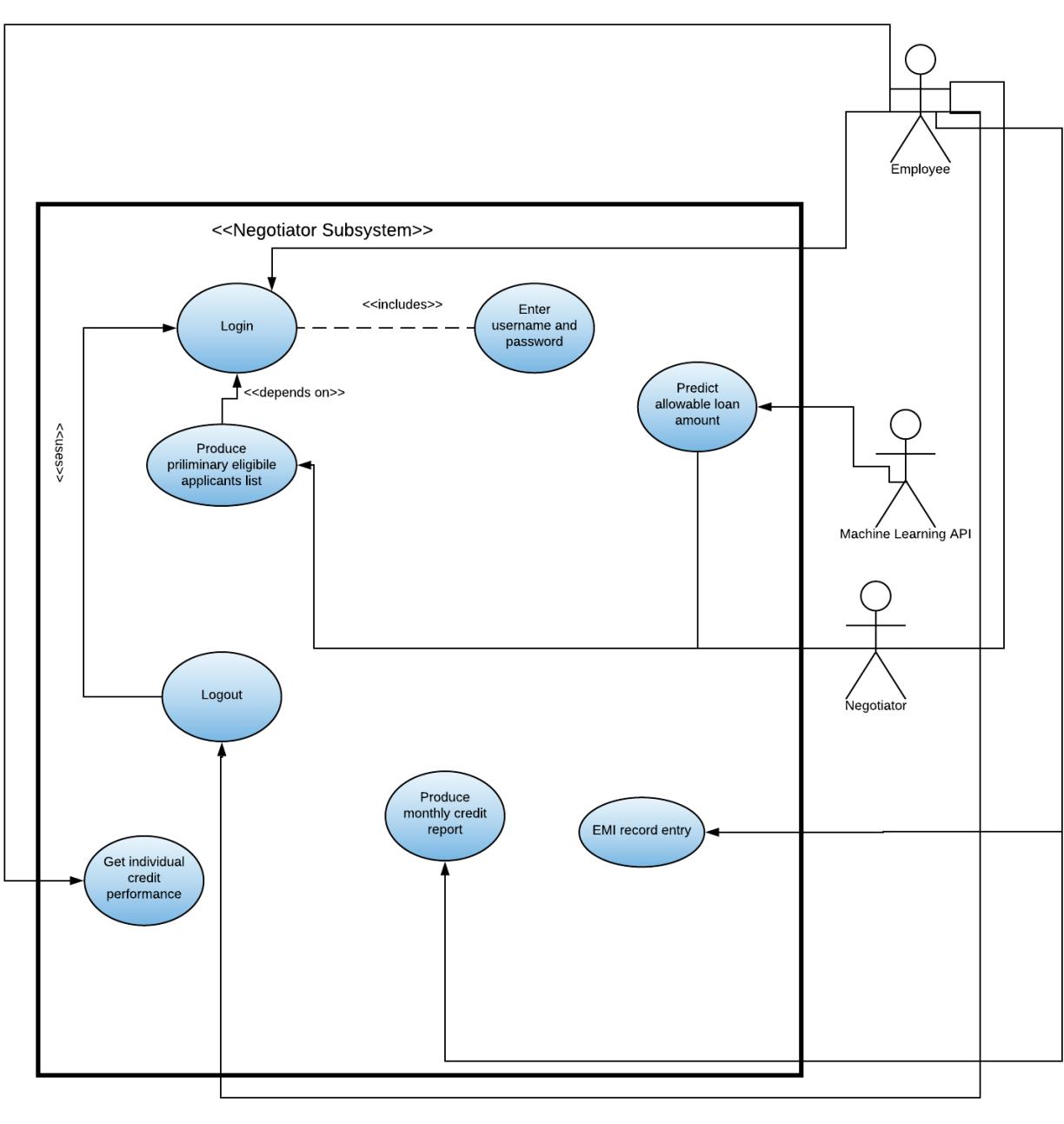


Figure 3: Use case diagram for *Negotiator* subsystem

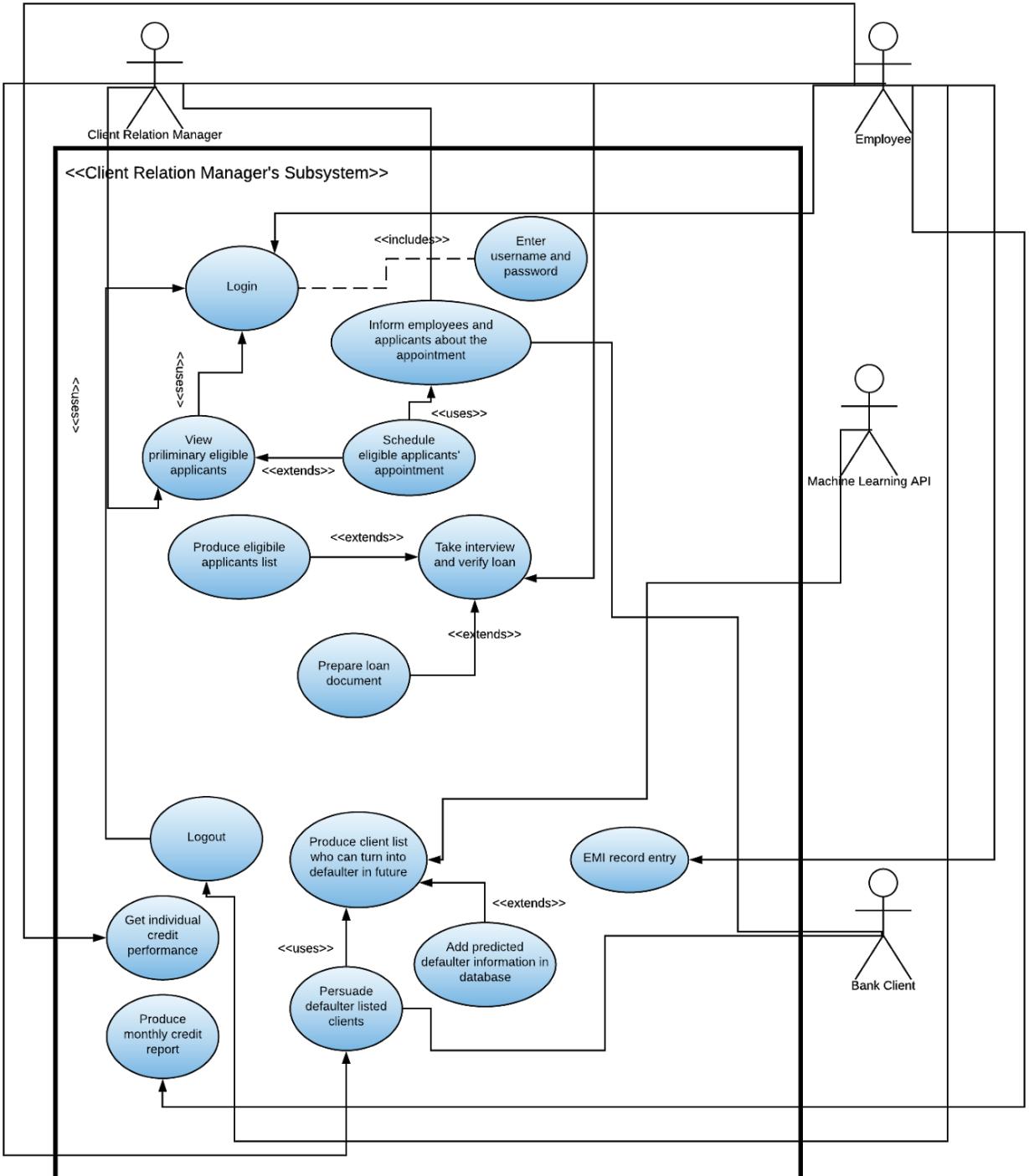


Figure 4: Use case diagram for *CRM* subsystem

2.2 Class Diagram

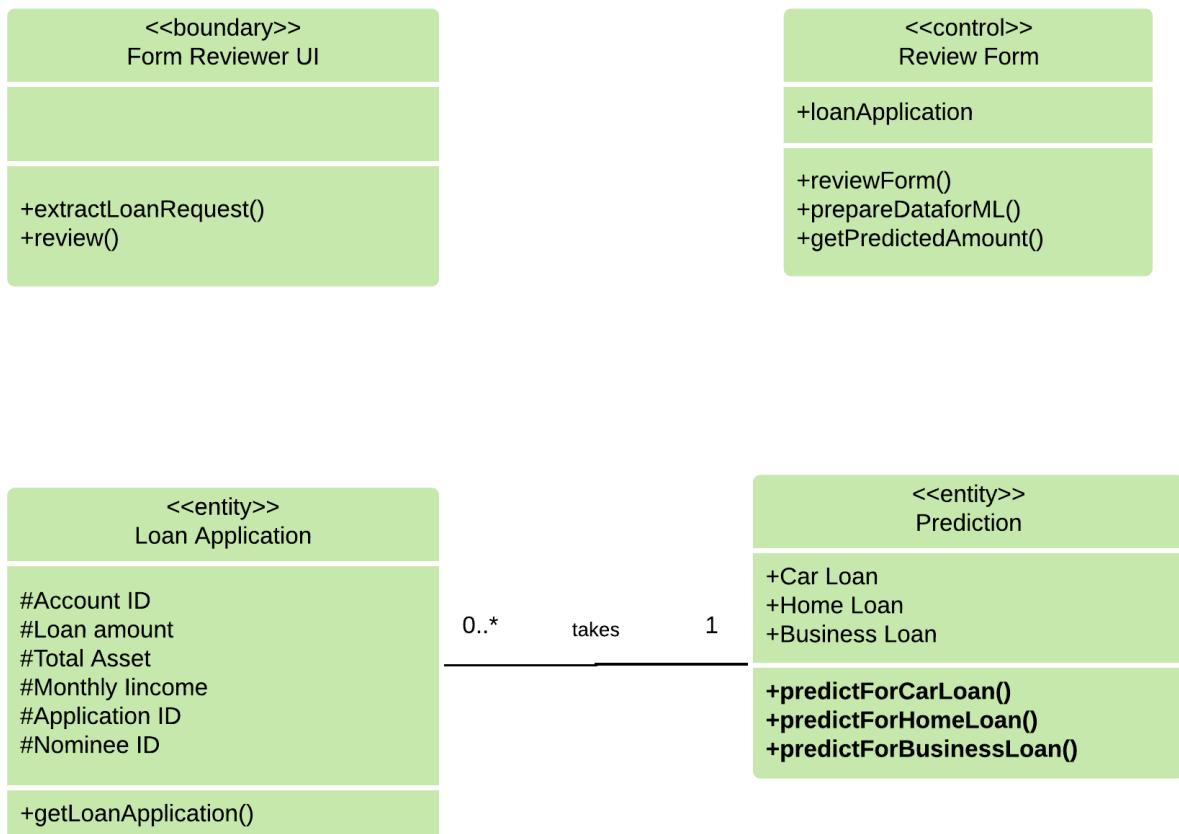


Figure 5: Class diagram for *Form Reviewer*

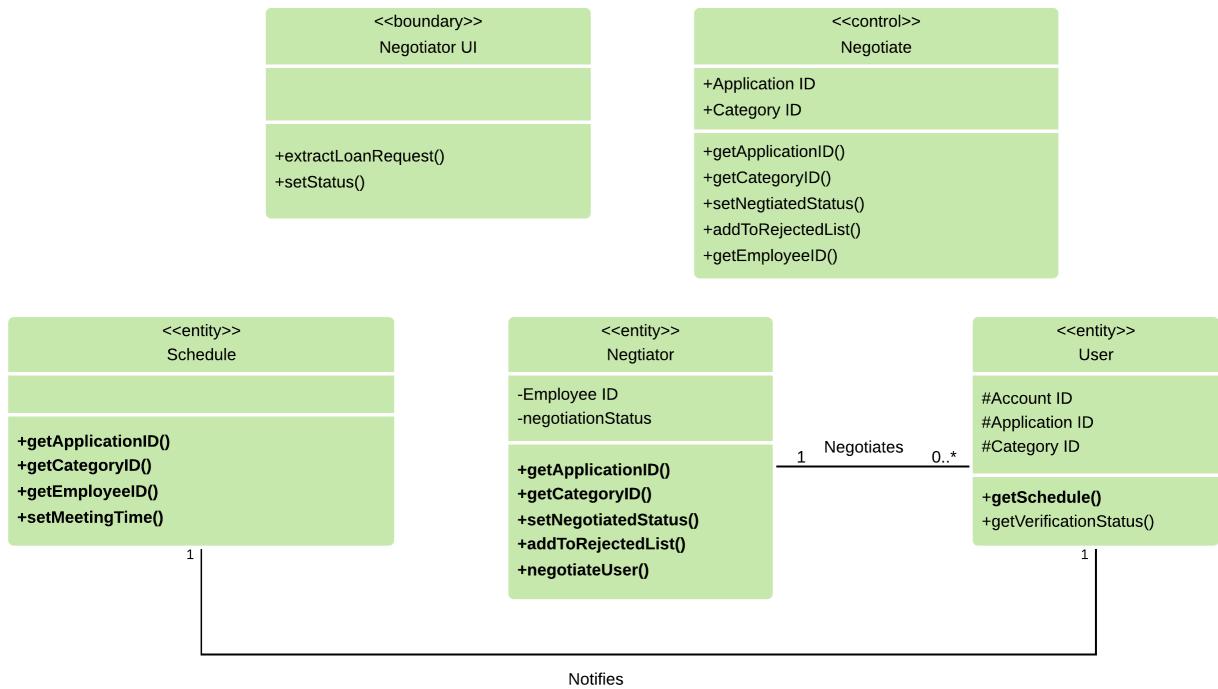


Figure 6: Class diagram for *Negotiator*

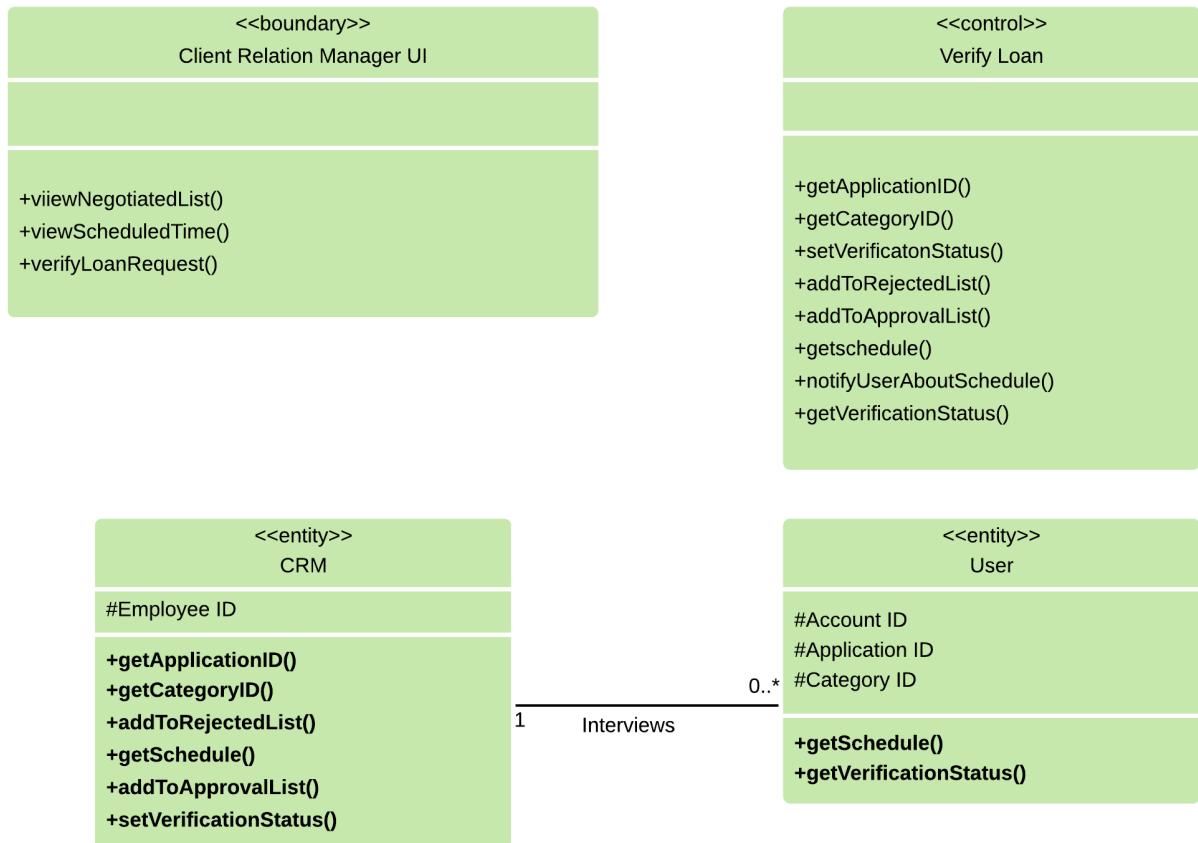


Figure 7: Class diagram for *Client Relations Manager*

2.3 Sequence Diagram

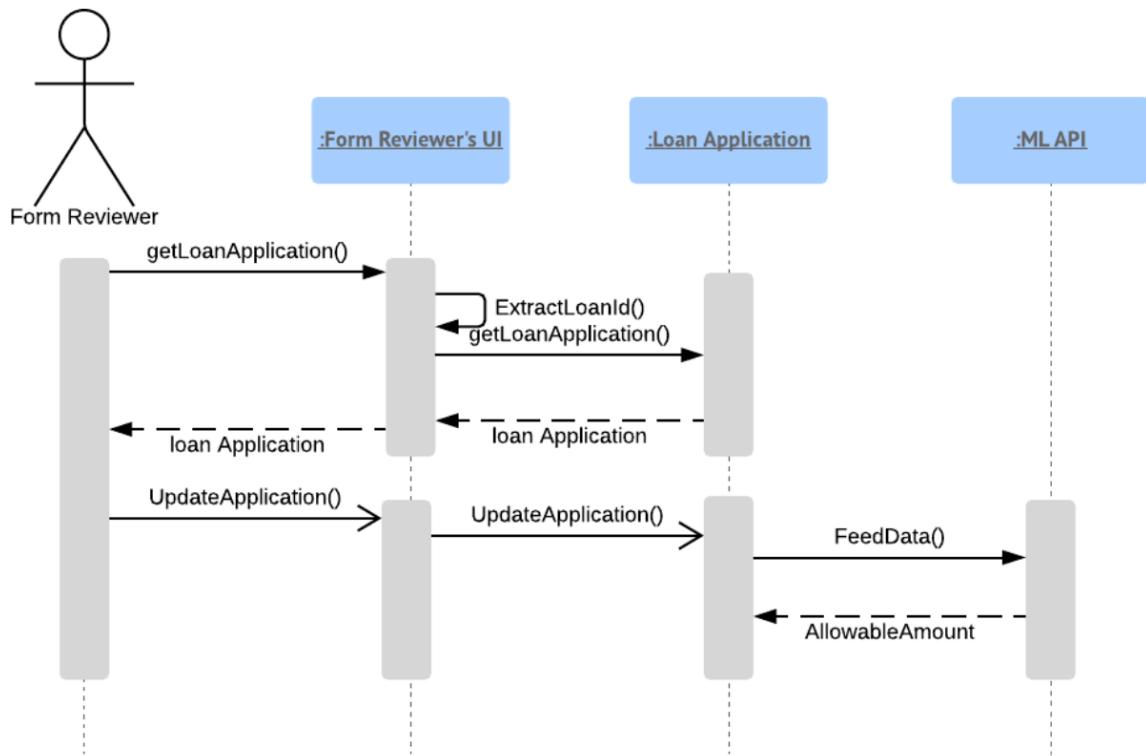


Figure 8: Sequence diagram for *Form Reviewer* subsystem

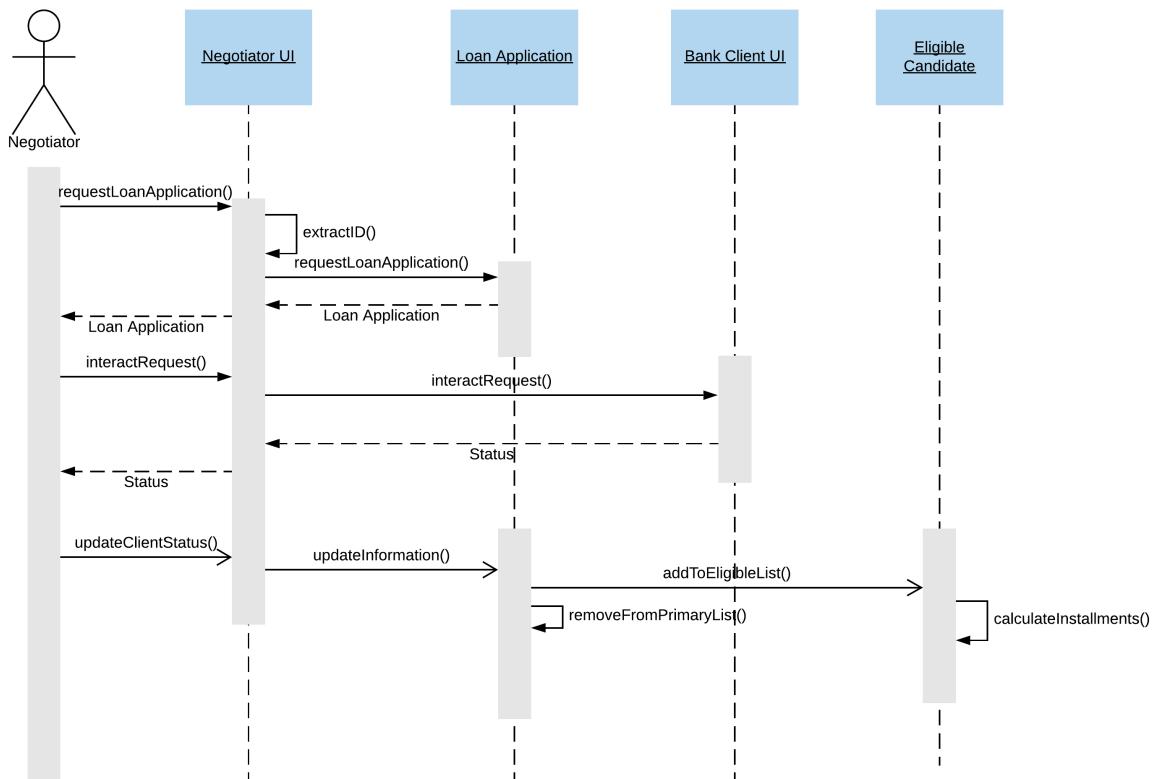


Figure 9: Sequence diagram for *Negotiator* subsystem

2.4 Data Flow Diagram

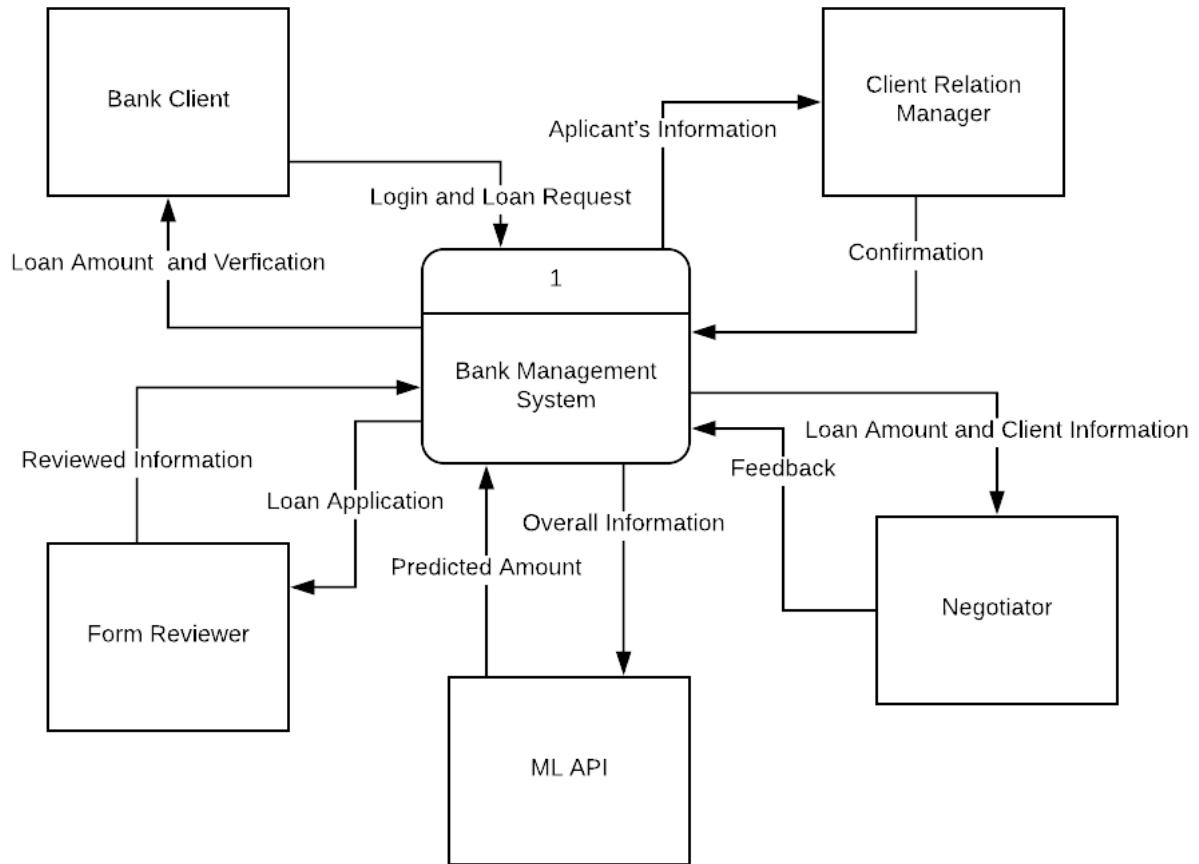


Figure 10: Context Diagram

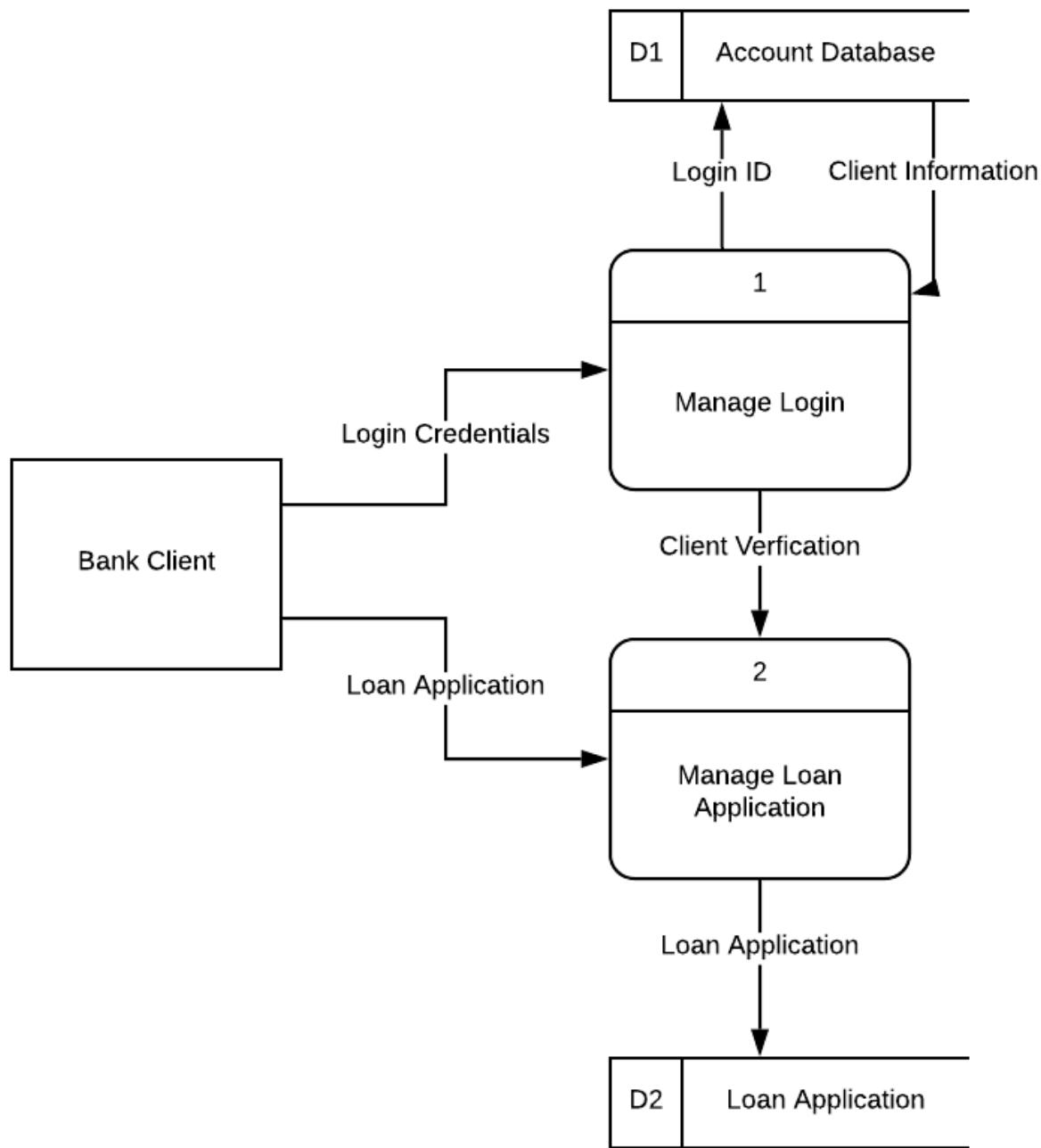


Figure 11: Data flow diagram for *User* subsystem

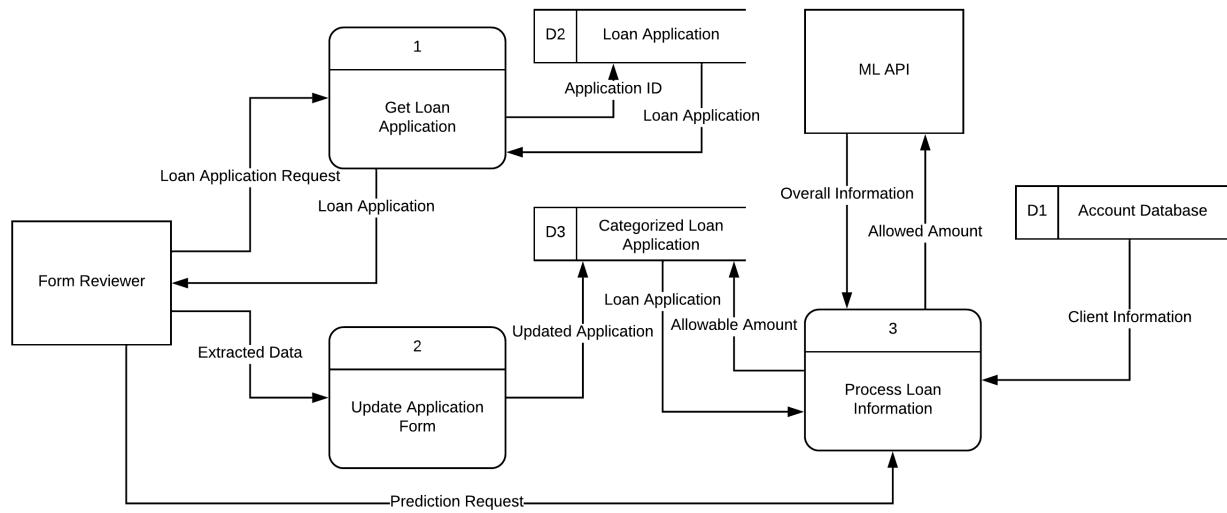


Figure 12: Data flow diagram for *Form Reviewer* subsystem

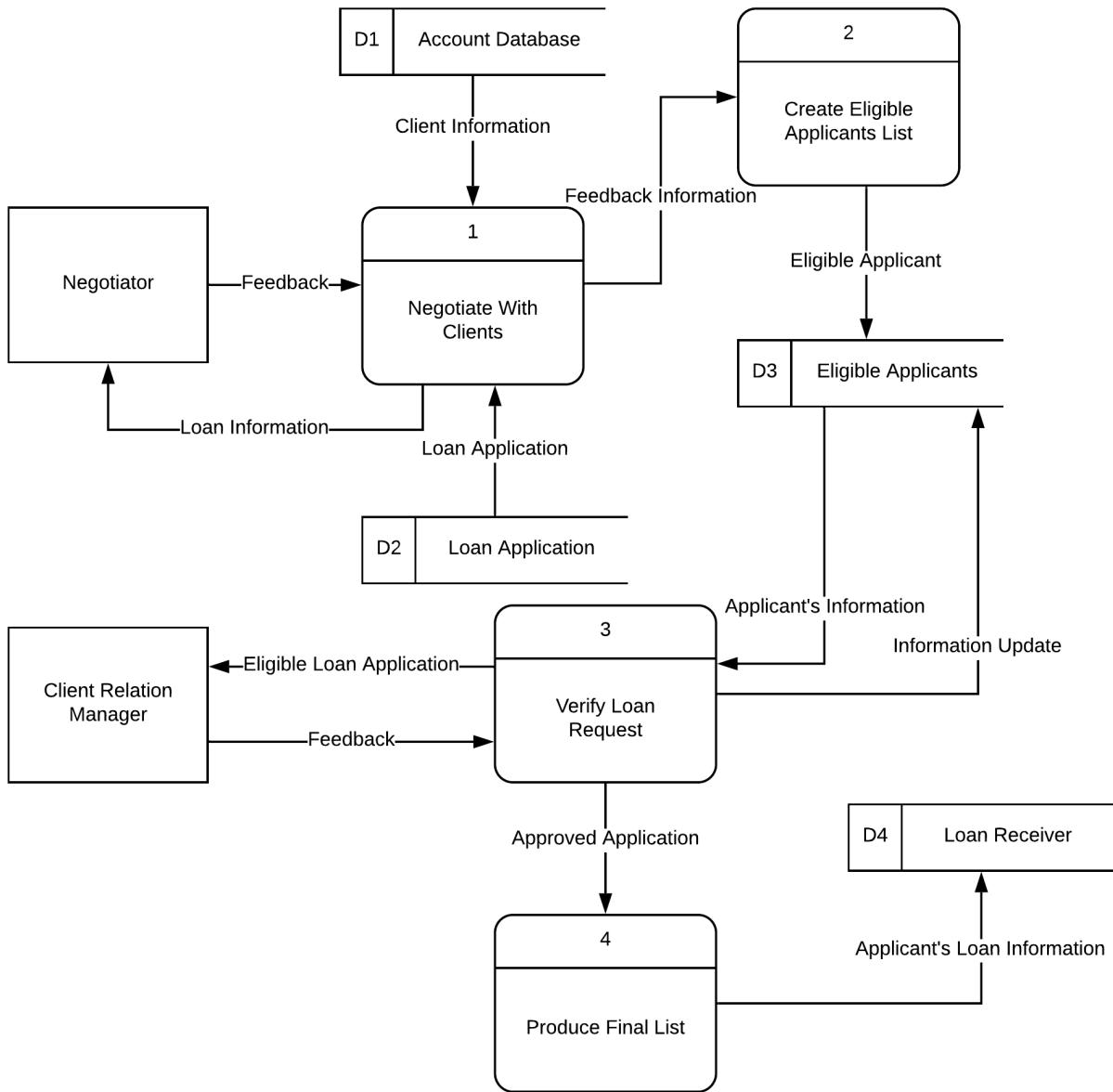


Figure 13: Data flow diagram for *Negotiator and CRM* subsystem

3 User Guide

3.1 Client End

Client end contains the legitimate account holder of the bank, any client holding a running account in the bank will have an online account.

3.1.1 Registration

Clients themselves are not allowed to register in the bank website or create an account in the website. Only bank has the power and if a client is a verified account holder, the bank will open an online account for him/her and provide him/her *username* and *password* to browse the site.

The screenshot shows a Django administration interface for adding a new account table. The top navigation bar includes links for 'Home', 'Polls', 'Account tables', and 'Add account table'. The main form has the following fields:

- Account type:** A text input field.
- User:** A dropdown menu with a '+' icon to add a new user.
- FullName:** A text input field.
- Age:** A text input field containing the value '0'.
- Sex:** A text input field.
- Marital status:** A text input field.
- Nominee id:** A dropdown menu with a '+' icon to add a new nominee.
- Balance:** A text input field containing the value '0'.
- Checking account:** A text input field containing the value '0'.

Figure 14: Create New Account Page (Admin)

3.1.2 Login

Client can login to the site using valid *username* and *password* provided by the bank.

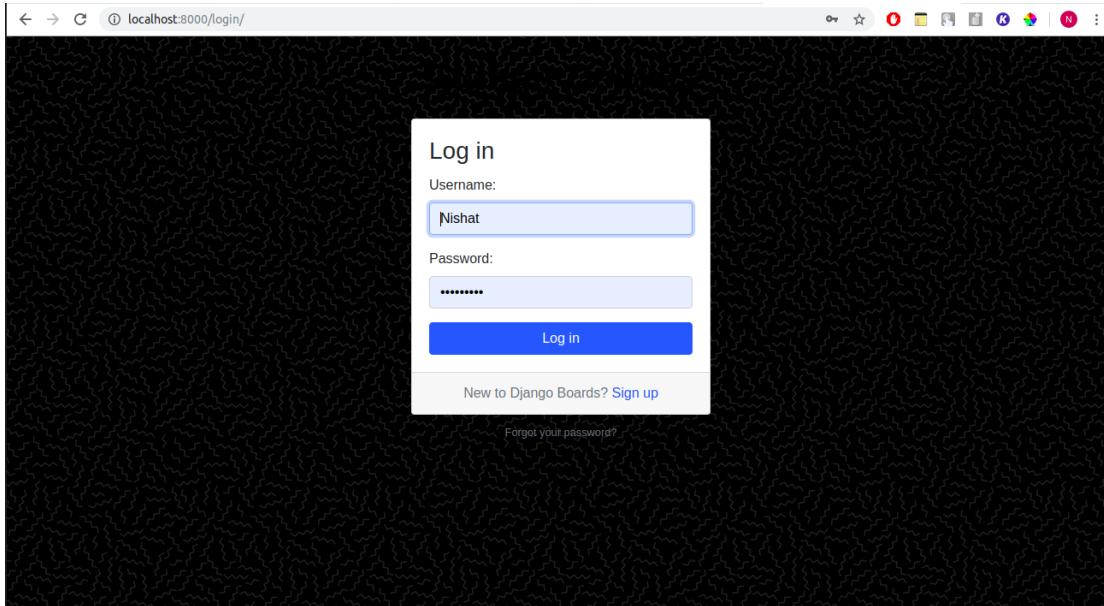


Figure 15: Login Page

3.1.3 Home Page

After getting successfully logged in, client can browse bank website, his/her account details, loan details etc.

3.1.4 Edit Profile Page

Client can edit profile information, password by clicking *Settings* option.

3.1.5 Apply For Loan

Client can apply for loan by clicking *Apply for loan* option in the top navigation bar.

3.1.6 Send Messages

Client can communicate with bank employees with the sites' internal messaging system.

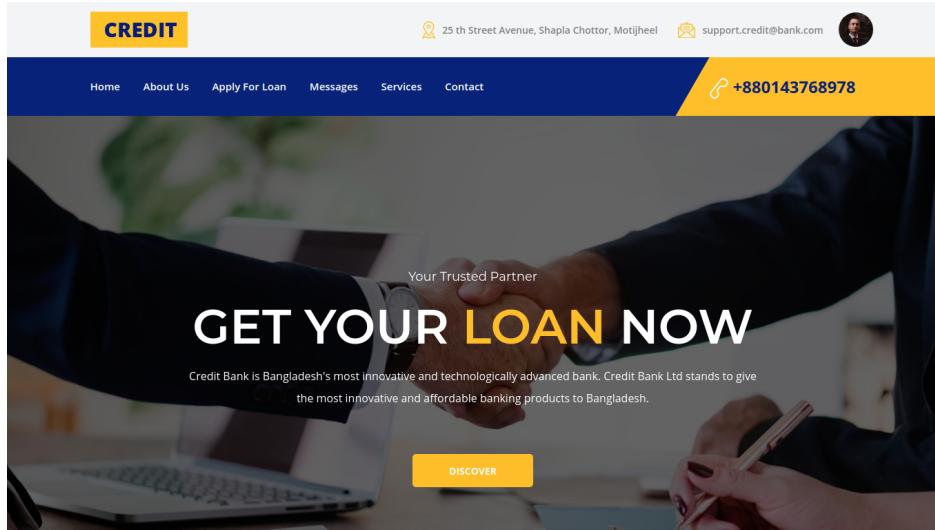


Figure 16: Login Page

E-mail	<input type="text" value="farhan@gmail.com"/>
First Name	<input type="text" value="Farhan"/>
Last Name	<input type="text" value="Feroz"/>
Password	<input type="password" value="Reset Password"/>

Figure 17: Edit Profile

3.2 Employee End

According to the requirements of the bank, we have categorized employees into three types:

- Form Reviewer
- Negotiator
- Client Relation Manager

Figure 18: Loan Application Page

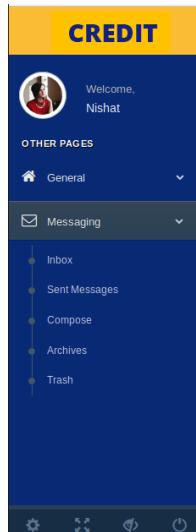


Figure 19: Sidebar

3.2.1 Registration

For an employee to have an online account in the bank, the bank will act as admin and will manually register legitimate bank employees and provide the *username* and *password* to the employees. Employees themselves can't create their online account.

3.2.2 Login

Employee logins using valid *username* and *password* provided by the bank.

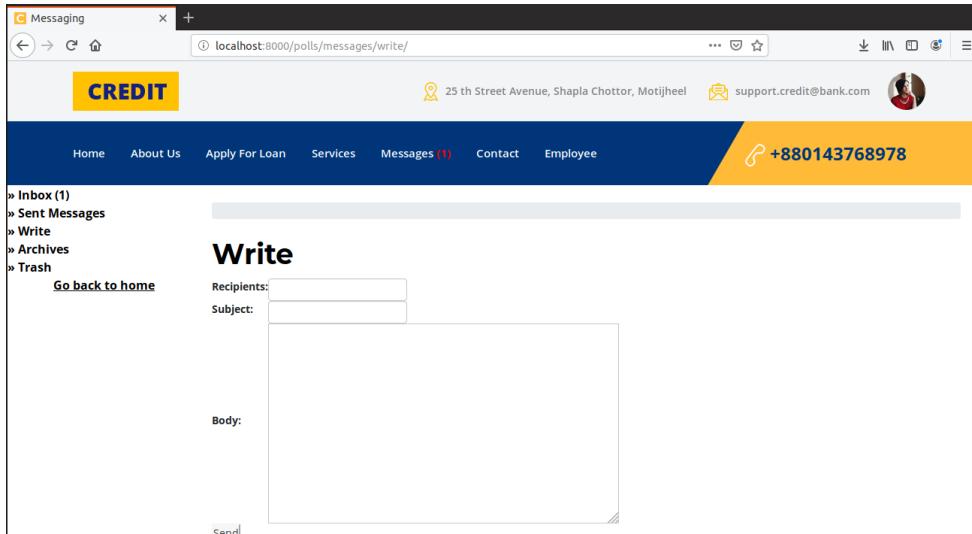


Figure 20: Inbox

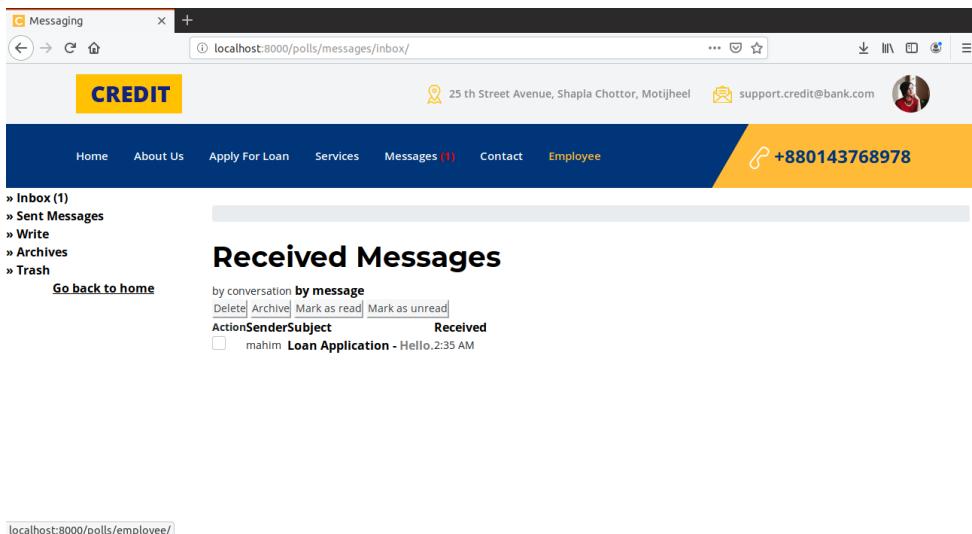


Figure 21: Received Message

3.2.3 Profile

Employee home page contains his profile information stored in bank database and his expertise and projects under the bank.

3.2.4 Employee Workspace

Side bar contains necessary tabs and links which are enough for each type of employee to conduct their daily job in an organized way.

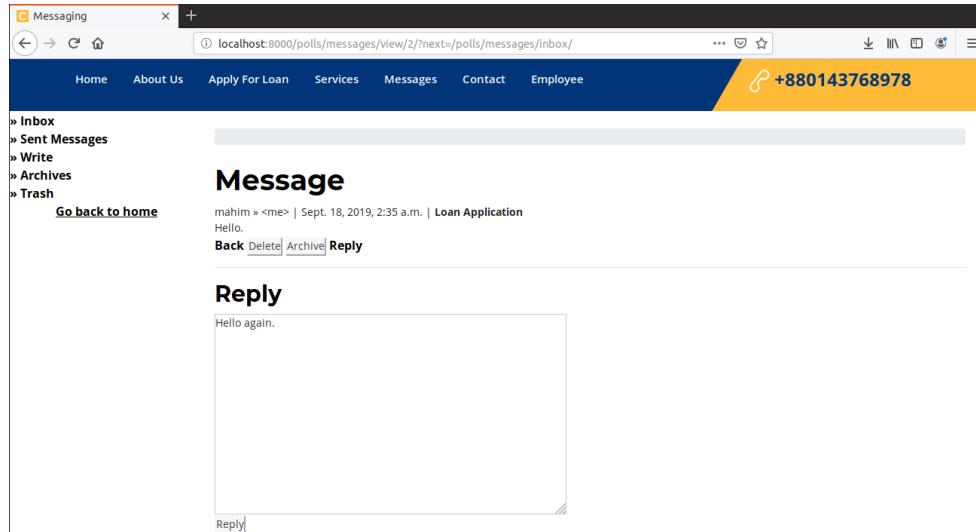


Figure 22: Message on the Sender Side

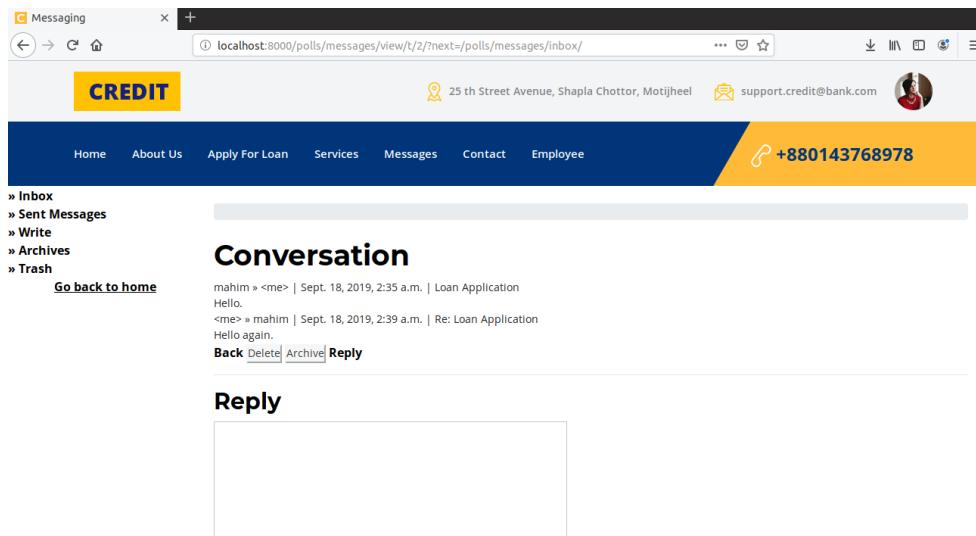


Figure 23: Whole Conversation

3.2.5 Data Visualization with Graphs

To gain more information about current status of the bank and its clients, different types of graphs based on different features are shown in the *Data Presentation* tab.

- **Loan Graphs**

- Bad Loan Graph(Amount Borrowed Vs Year)

Add employee

Employee id:

User name:

Department id: +/-

Employee name:

Age:

Sex:

Email:

Address:

Designation:

Figure 24: Add Employee Page (Admin)

Login Form

Lost your password?

New to site? Create Account

Credit Bank

©2016 All Rights Reserved. Credit Bank. Privacy and Terms

Figure 25: Login Page

- Loan Status by Region(Bad Loan Vs Region)
- Condition of Loan by Purpose(Loan Nature Vs Purpose)

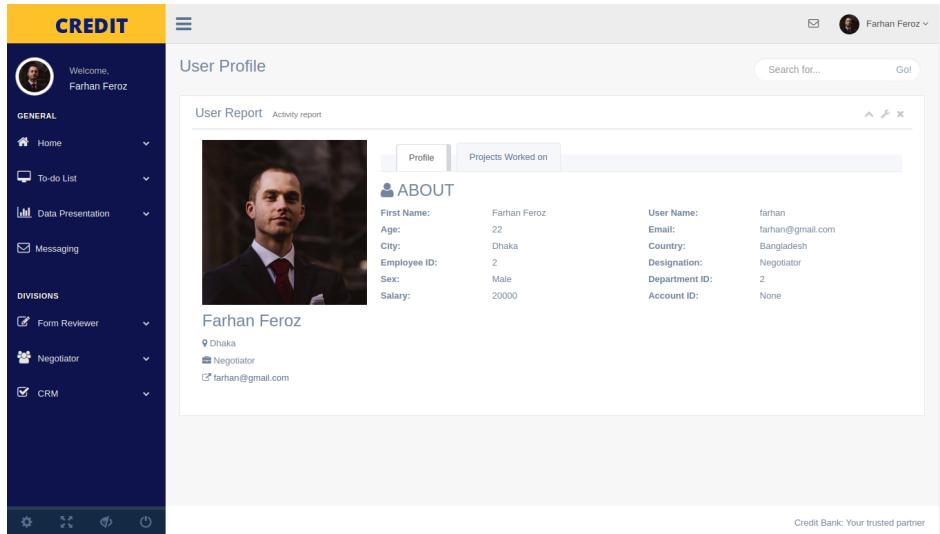


Figure 26: Employee Home Page

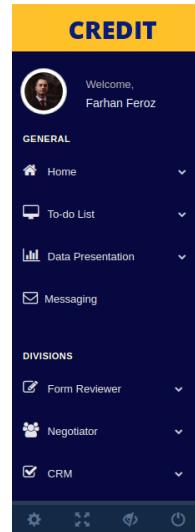


Figure 27: Sidebar

- Issuance of Loans
 - * Amount of Good Loans
 - * Amount of Bad Loans



Figure 28: Bad Loan Graph



Figure 29: Loan Status by Region

- Bad/Good Ratio(By Purpose)

• Defaulter Graphs

- Defaulter Rates



Figure 30: Condition of Loan by Purpose

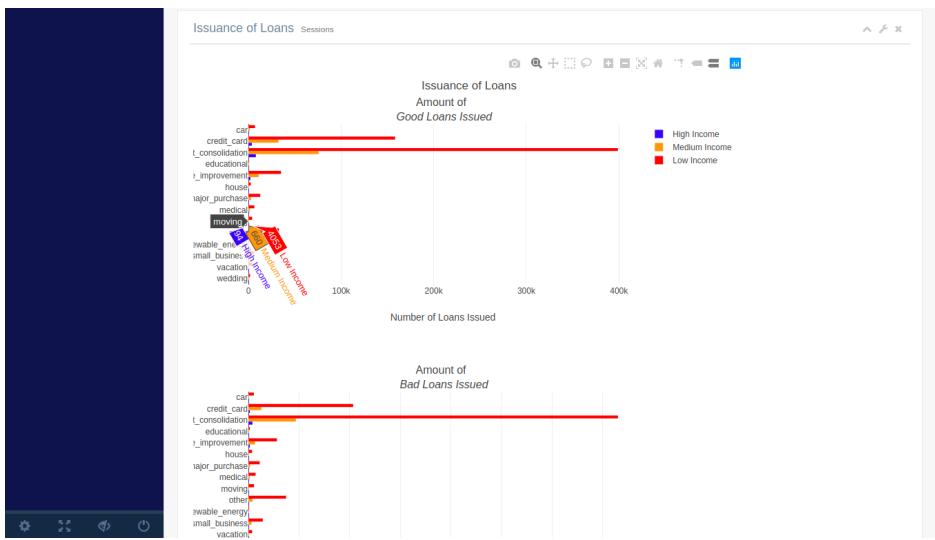


Figure 31: Amount of Good Loans

- Regional Risks

• Interest Rate Graphs

- Average Interest Rates by Loan Status Distribution
- Average Purpose Interest Rate by Income Category



Figure 32: Amount of Bad Loans



Figure 33: Bad/Good Ratio(By Purpose)

3.2.6 To-do List

Employee can keep track of his/her activities and maintain a calendar to keep his/her to-do list organized.

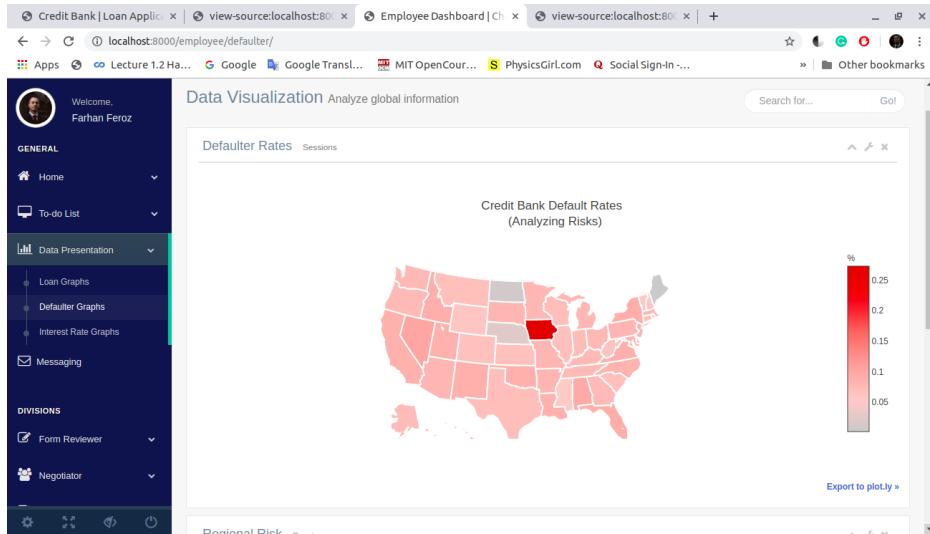


Figure 34: Defaulter Rates

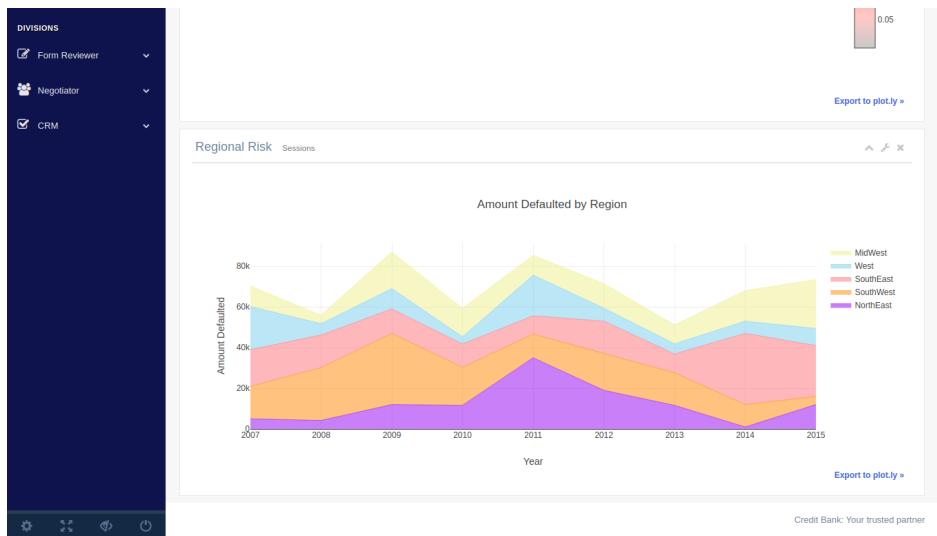


Figure 35: Regional Risks

3.3 Apply For Loan and Grant

3.3.1 Apply For Loan

- **Loan Application Form Fillup:** Client can access this page by clicking *Apply for loan* option in the top navigation bar of his/her homepage. There are 5 steps to follow and five different forms appear to gather client's information.

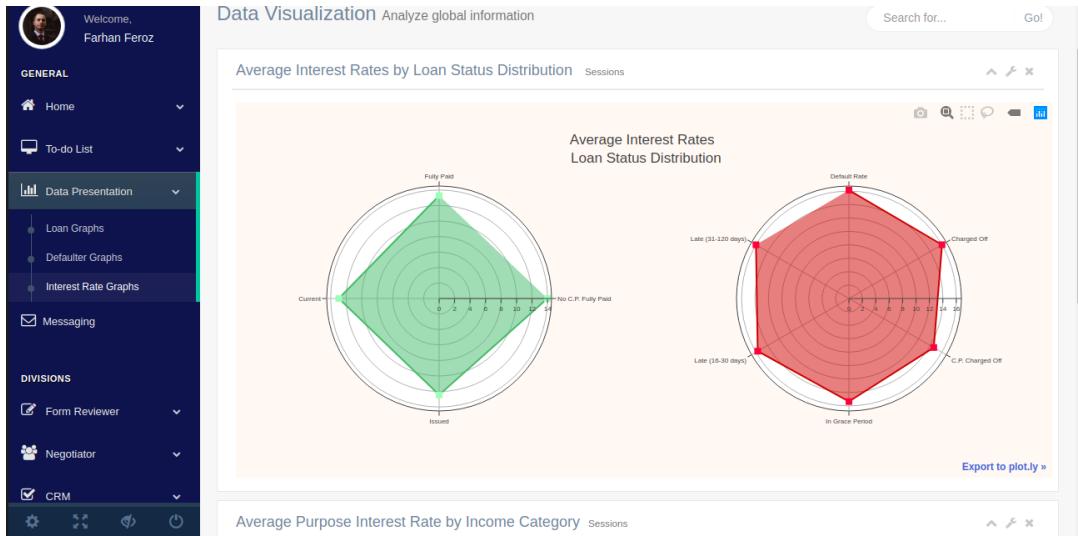


Figure 36: Average Interest Rates



Figure 37: Average Purpose Interest Rate

- **Successfully Stored In Database:** After following all the steps, a page will arrive showing successful submission of client's application.

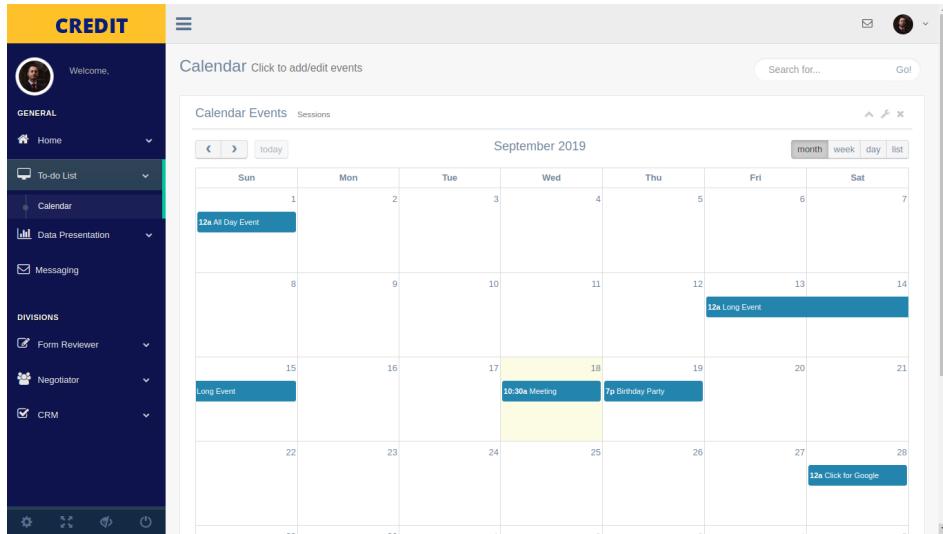


Figure 38: Basic To-do List

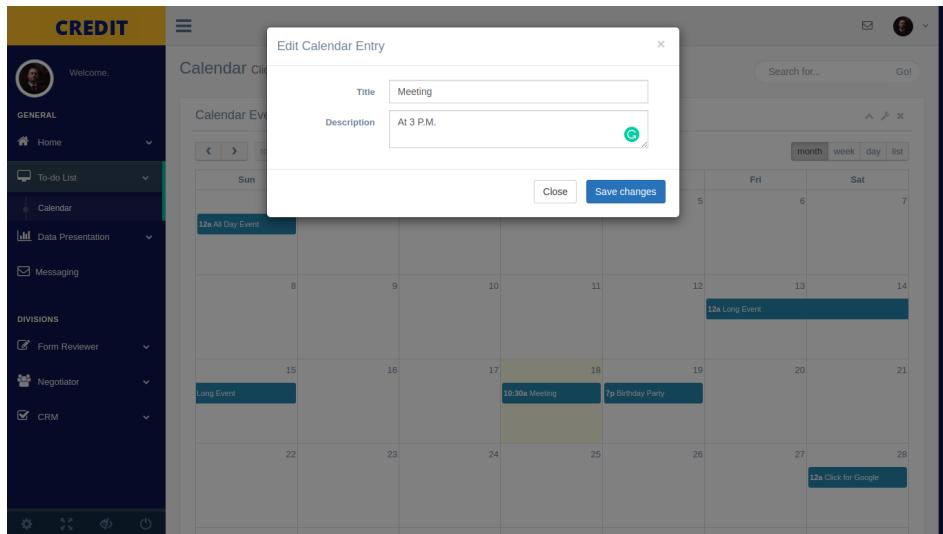


Figure 39: Creating an Entry

3.3.2 Form Reviewer

- **Loan Application List:** In form reviewer's window, all loan application form will appear in Pending Application table.

Figure 40: Basic Information Form

Figure 41: Debt Settlement Form

- **Search Loan Application:** Form reviewer can search through the list.
- **Review Loan Application:** Form reviewer can review specific application.
- **Generate Prediction:** After reviewing the application form reviewer can send the form for prediction. After generating prediction the application will disappear from the list.

Figure 42: Secondary Applicant Form

Figure 43: Applied Successfully

- **Reject Application:** Form reviewer can reject the application.

3.3.3 Negotiator

- **Loan Application List:** In form negotiator's window, all loan application form will appear in Pending Application table.

The screenshot shows the 'Applications Edit and Adjust for Prediction' screen. The left sidebar has sections for GENERAL (Home, To-do List, Data Presentation, Messaging), DIVISIONS (Form Reviewer, Negotiator, CRM), and a central section for Pending Applications, Account Table, and Loan History. The main area displays a table titled 'Pending Requests' with columns: Application ID, Request Date, Prediction, Reject, and Action. The table contains 9 entries, each with a timestamp from September 18, 2019, and a 'Predict' status. A search bar at the top right shows 'Search for...' and a 'Go!' button.

Application ID	Request Date	Prediction	Reject	Action
14	Sept. 18, 2019, 9:21 a.m.	Predict	Reject	View
16	Sept. 18, 2019, 1:21 p.m.	Predict	Reject	View
17	Sept. 18, 2019, 1:27 p.m.	Predict	Reject	View
18	Sept. 18, 2019, 1:28 p.m.	Predict	Reject	View
19	Sept. 18, 2019, 1:30 p.m.	Predict	Reject	View
20	Sept. 18, 2019, 1:31 p.m.	Predict	Reject	View
21	Sept. 18, 2019, 1:31 p.m.	Predict	Reject	View
22	Sept. 18, 2019, 1:32 p.m.	Predict	Reject	View
23	Sept. 18, 2019, 1:33 p.m.	Predict	Reject	View

Figure 44: Application List

This screenshot is identical to Figure 44, but it includes a search filter. In the search bar at the top right, '23' is entered. The table below shows only one entry, corresponding to the filtered result.

Application ID	Request Date	Prediction	Reject	Action
23	Sept. 18, 2019, 1:33 p.m.	Predict	Reject	View

Figure 45: Showing Search Result

- **Send message:** The negotiator will negotiate with the client if he agrees with the allowable amount. This is done by sending the client in our messaging API.
- **Update amount:**
- **Remove From the List:** After updating the amount or rejecting, the application will disappear from the list.

Credit

Welcome, Mahim Mahbub

GENERAL

- Home
- To-do List
- Data Presentation
- Messaging

DIVISIONS

- Form Reviewer
- Negotiator
- CRM

CREDIT

Welcome, Mahim Mahbub

GENERAL

- Home
- To-do List
- Data Presentation
- Messaging

DIVISIONS

- Form Reviewer
- Negotiator
- CRM

Credit Bank Loan Application

Application 16 Online

Review Form Carefully!

Step 1 Step 2 Step 3 Step 4 Step 5

Loan Details

Loan Amount *	150000
Number of Terms *	36
Interest Rate (%)	6.0
Installment amount	4418.666666666667
Grade *	<input type="radio"/> A <input type="radio"/> B <input checked="" type="radio"/> C <input type="radio"/> D <input type="radio"/> E <input type="radio"/> F <input type="radio"/> G
Purpose *	debt_consolidation
Open Accounts*	5.0
Total Accounts*	8.0
Bankcard open to buy*	0.0
Percent of trades never delinquent*	100.0

Figure 46: Review Form

Credit

Welcome, Mahim Mahbub

GENERAL

- Home
- To-do List
- Data Presentation
- Messaging

DIVISIONS

- Form Reviewer
- Negotiator
- CRM

CREDIT

Welcome, Mahim Mahbub

GENERAL

- Home
- To-do List
- Data Presentation
- Messaging

DIVISIONS

- Form Reviewer
- Negotiator
- CRM

Credit Bank Loan Application

Application 16 Online

Review Form Carefully!

Step 1 Step 2 Step 3 Step 4 Step 5

Account Details

Installment Accounts *	4.0
Satisfactory Accounts *	5.0
Average Current Balance *	47485.0
Account Opened *	4.0
Accounts Opened(12 months) *	2.0
Accounts with Past Due *	0.0
BC Ratio *	0
Mortgage Accounts *	1.0
Bankcard Account Opened *	0
Bankcard Delinquency *	0

Figure 47: Review Form

3.3.4 CRM

- **Loan Application List:** In form CRM's window, all loan application form will appear in Pending Requests table.
- **Grant Loan:** CRM will grant the loan by clicking *Accept* option from the table.
- **Remove From the List:** The accepted or rejected application will disappear from the list.

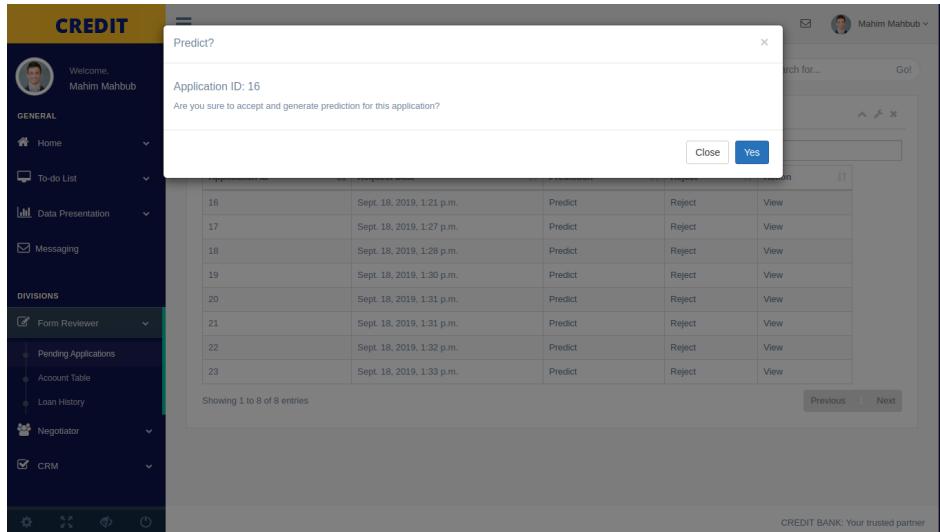


Figure 48: Generate Prediction

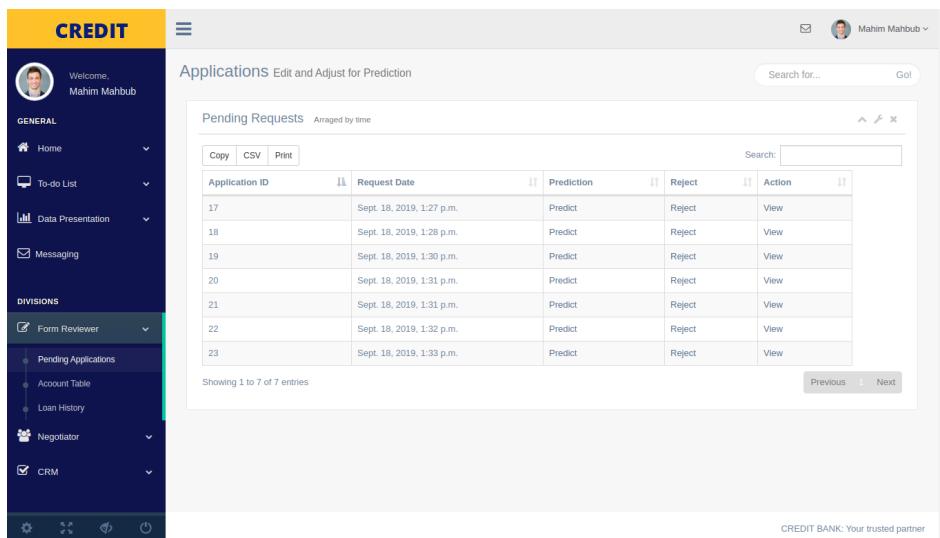


Figure 49: Form Forwarded to Negotiator

4 Machine Learning models and platform :

- **Platform :** Jupyter notebook.
- **Dataset :** We used a dataset with 2.26 million entries . The dataset has 145 columns in total. We used 93 columns for training. Here is the link of the dataset. [dataset](#)
- **Algorithm :** The Logistic Regression classifier has been used for prediction.
- **Accuracy:** After doing logistic regression, our accuracy was 82% on the test dataset.

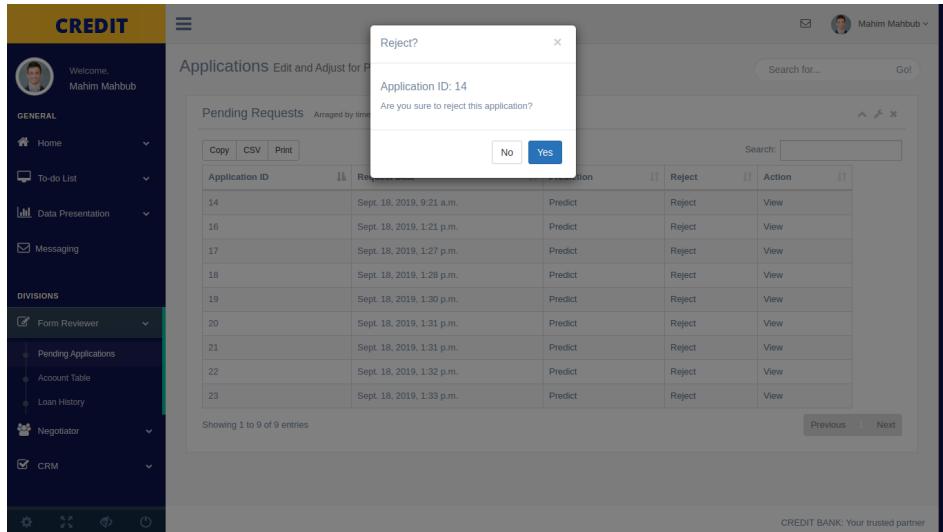


Figure 50: Reject Application

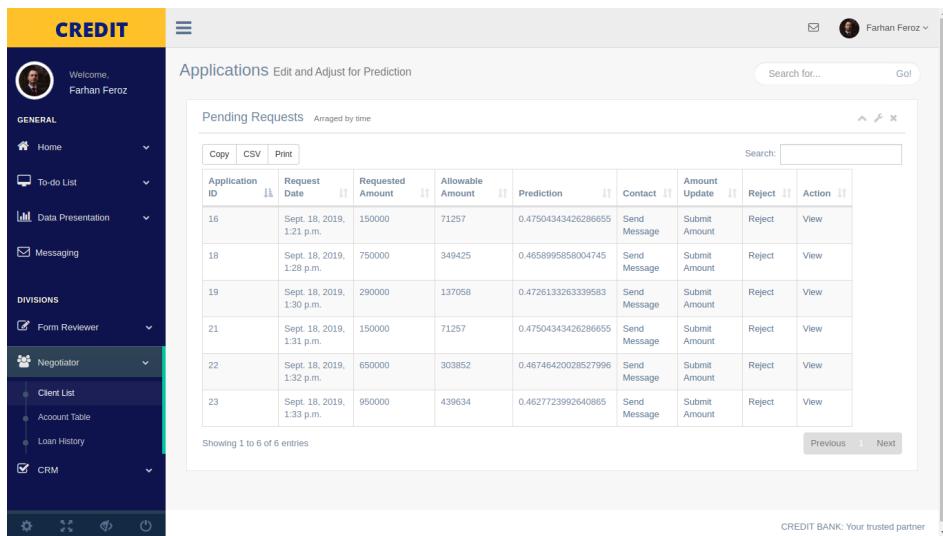


Figure 51: Application List

- **Scope for improvement:** A paper suggested using decision tree and neural network as training algorithm. But for Resource constraints, we could not use these algorithms.

5 System Documentation

The Django Framework is used for building this project which comes handy with SQLite database.

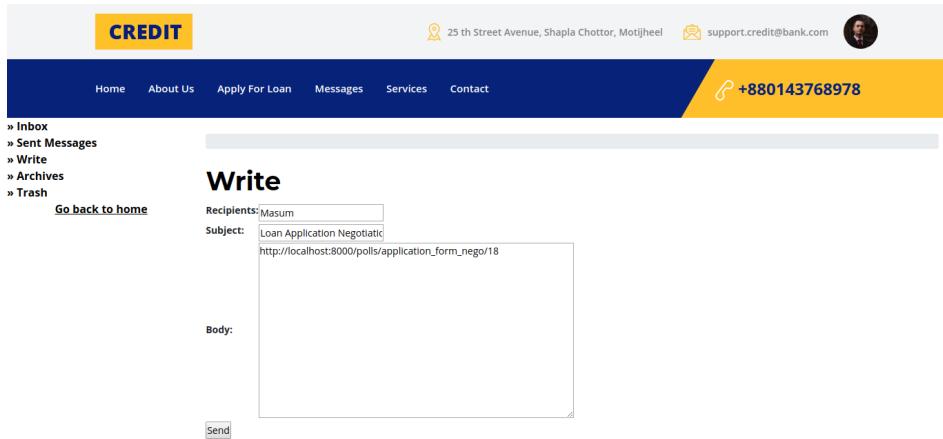


Figure 52: Send message for negotiation

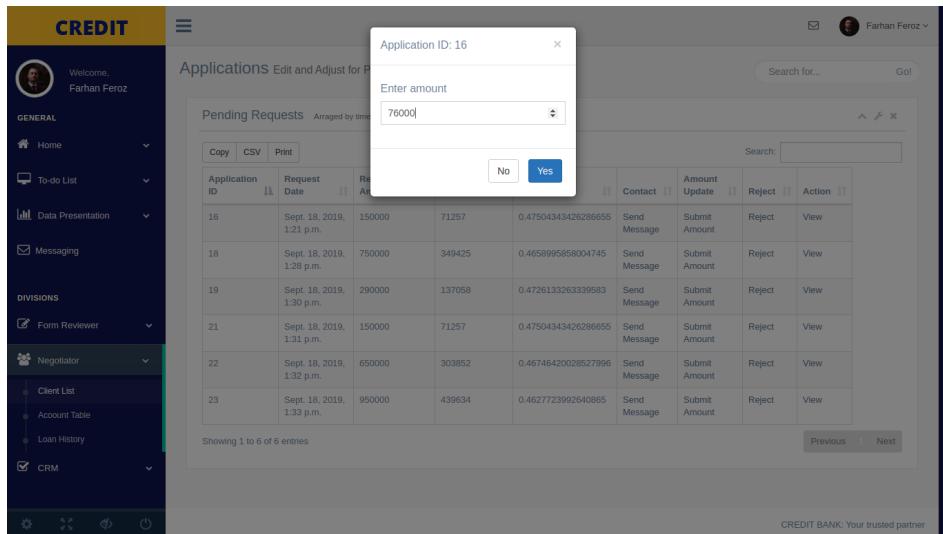


Figure 53: Update allowable amount according to the negotiation

5.1 Installation

- Django : The main Framework.

```
pip install Django
```

For the installation of django, this link can be followed. [Django-installation](#)

The screenshot shows the CREDIT application interface. On the left is a dark sidebar with a yellow header 'CREDIT'. The sidebar includes sections for 'GENERAL' (Home, To-do List, Data Presentation, Messaging), 'DIVISIONS' (Form Reviewer, Negotiator), and 'CRM' (Client List, Account Table, Loan History). The 'CRM' section has a checked checkbox. The main area is titled 'Applications Edit and Adjust for Prediction' and shows a table for 'Pending Requests'. The table has columns: Application ID, Request Date, Requested Amount, Allowable Amount, Prediction, Contact, Amount Update, Reject, and Action. The table contains five entries with IDs 18, 19, 21, 22, and 23, each with different details like date (Sept. 18, 2019), amount (e.g., 750000, 290000), and prediction values. At the bottom of the table, it says 'Showing 1 to 5 of 5 entries'. The top right corner shows the user 'Farhan Feroz'.

Figure 54: Remove From List

This screenshot is similar to Figure 54 but for user 'Sifat Ishmaan'. The sidebar and table structure are identical, showing the same 'Pending Requests' list with five entries. The table columns are: Application ID, Request Date, Requested Amount, Negotiated Amount, Prediction, Contact, Accept, Reject, and Action. The table contains entries with IDs 16, 19, and 21, each with different details like date (Sept. 18, 2019), amount (e.g., 150000, 290000), and prediction values. At the bottom of the table, it says 'Showing 1 to 3 of 3 entries'. The top right corner shows the user 'Sifat Ishmaan'.

Figure 55: Application List

- Graphos : For rendering graphs from the historical data.

```
pip install django-graphos
```

- Mathfilters : For advanced calculations.

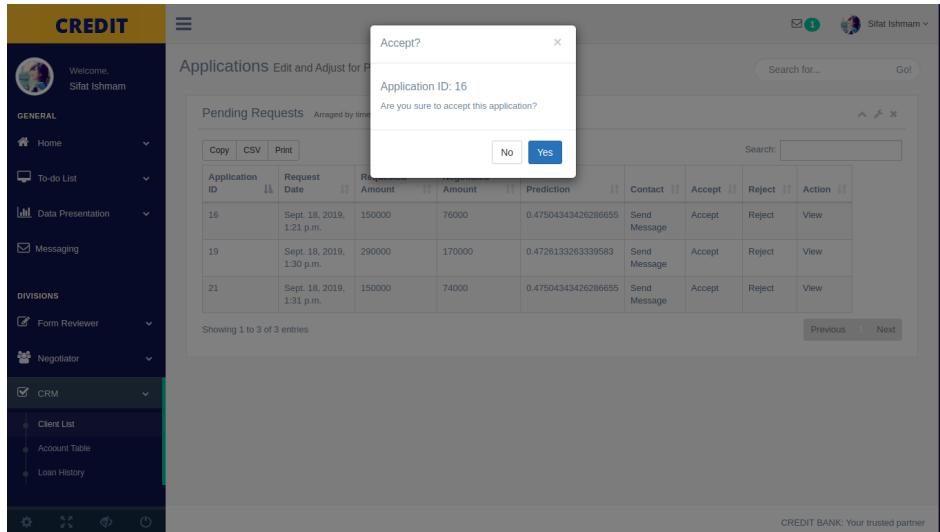


Figure 56: Accept Loan

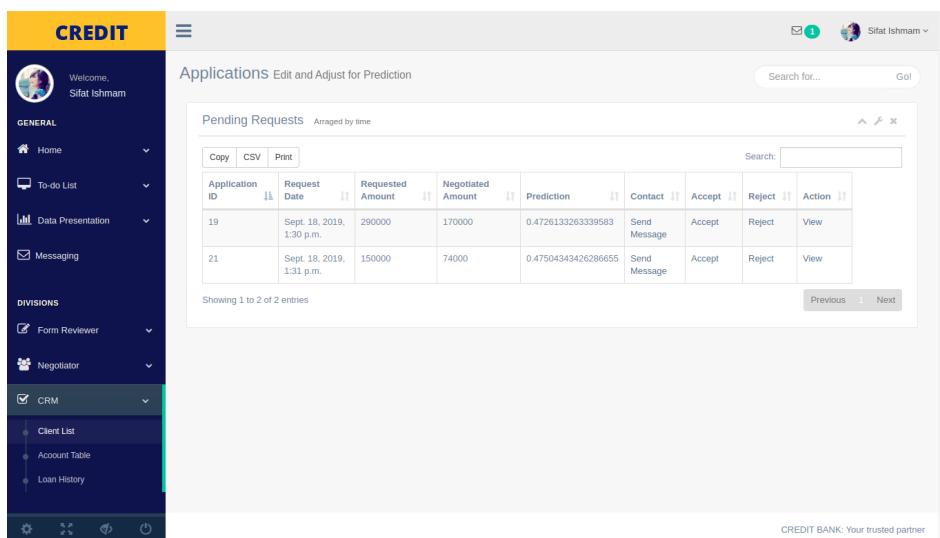


Figure 57: Remove From List

```
pip install django-mathfilters
```

- Widget-Tweaks : For tweaking the form field rendering in templates for altering CSS classes and HTML attributes is supported.

```
pip install django-widget-tweaks
```

- django-postman : For allowing authenticated users of the bank website to exchange private

messages within the site. This facilitates the employee client communication. Documentation of django-postman can be found [here](#).

```
pip install django-postman
```

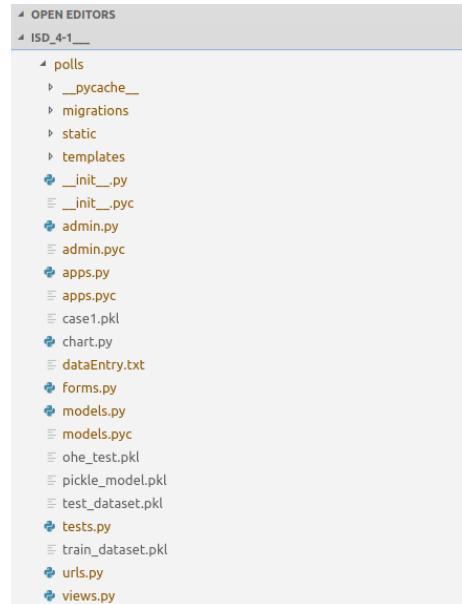
6 Function Documentation

We used a good number of python, javascript functions and css files. Here are some key functions from these.

We used three different django-apps for separating user and employee sites.

- **App** : This is the app containing the templates , css , js and python files for loan application.
- **Employee** : This is the app containg Employee's templates, UI elements, employee's dashboard etc. We have 3 types of employee. Form reviewer, Negotiator, Client Relations Manager. Their respective functionalities are handled in this app.
- **Polls** : This is the app which handles the employee and user sites interactions.

6.1 Polls



The screenshot shows a file explorer window with the following directory structure:

```
OPEN EDITORS
ISD_4-1__
polls
  __pycache__
  migrations
  static
  templates
  __init__.py
  __init__.pyc
  admin.py
  admin.pyc
  apps.py
  apps.pyc
  case1.pkl
  chart.py
  dataEntry.txt
  forms.py
  models.py
  models.pyc
  ohe_test.pkl
  pickle_model.pkl
  test_dataset.pkl
  tests.py
  train_dataset.pkl
  urls.py
  views.py
```

6.1.1 polls/urls.py :

This python file handles the urls of the polls app and loads a valid views.py function.

```
path('' , views.homepage, name='homepage'),
url(r'^login_success/$', views.login_success, name='login_success'),
path('messages/' , include('postman.urls')),
path('handleForm/' , views.handleForm , name='handleForm'),
path('edit_profile/' , views.edit_profile , name='edit_profile'),
path('contact/' , views.contact, name='contact'),
path('profile/' , views.profile , name='profile'),
```

6.1.2 polls/views.py :

- **def homepage(request):**

If the user is logged in, this function renders user's homepage and if not, this renders the bank homepage.

- **def login_success(request):**

When a user clicks the login button, this function checks if the login request is from a legitimate user.

- Input : The parameters from a HTTP POST request.
- Output : If successful, takes the user to his homepage.

- **def logout_success(request):**

When a user clicks the logout button, this function logs out the user.

- Input : User credentials.
- Output : If successful, takes the user to the bank homepage.

- **def about(request):**

This is takes to the bank's about page.

- **def services(request):**

This is takes to the bank's services page.

- **def contact(request):**

This is takes to the bank's contact information page. **def handleform(request):**

When a legitimate user applies for a loan, this function is triggered.

- Input : The parameters from the loan Application form.

- Output : The loan application is saved in the database , the status of the application is saved as "APPLIED" and takes the user to a new page.

6.1.3 polls/models.py :

models.py contains the database tables of our system. These are python classes with variables as database fields.

- **class Profile(models.Model):**

Stores profile images with reference to the user id.

- **user:** foreign key from "User" table of django authentication module.
- **image** stores the image url for user profile picture.

- **class AccountTable(models.Model):**

General information about the account holder along with a unique account id.

- **account_type:** Type of account.
- **user:** User id as foreign key.
- **fullName:** Full name of the client.
- **age:** Age of the client.
- **sex:** Sex of the client.
- **marital_status:** Marital Status of the client.
- **nominee_id:** Account id of nominee.
- **balance:** Balance of the account.
- **checking_account:** Balance of the current account.
- **savings_account:** Interest-bearing deposit account balance.
- **branch_id:** Branch unique id.

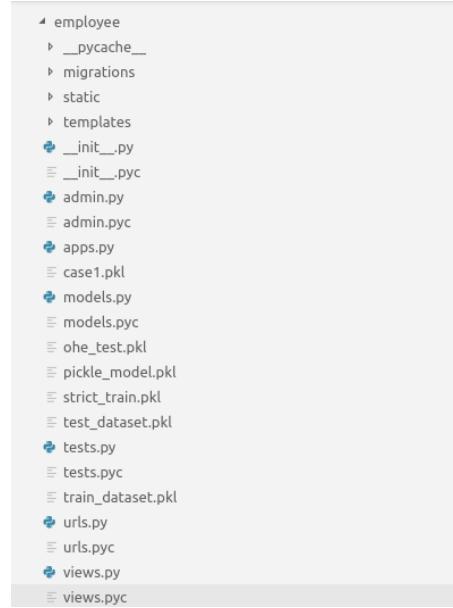
- **class Employee(models.Model):**

Contains user information for those who are actually employees of the bank.

- **employee_id:** ID number of the employee.
- **user_name:** User id as foreign key.
- **department_id:** The department the employee currently serves.
- **employee_name:** Name of the employee.
- **age:** Age of the employee.
- **sex:** Sex of the employee.
- **email:** Email address of the employee.
- **address:** Home address of the employee.
- **designation:** Designation of the employee.

- **salary:** Monthly salary in BDT.
 - **account_id:** If the employee also holds an account in the bank, the account id is stored here.
- **class CSVReader(models.Model):**
- When a client applies for a loan, a form is generated. We convert this form's data to a csv file. The csv file's url and other necessary fields are stored in this table along with unique loan id.
- **account_id:** Account ID of loan applicant.
 - **upload:** URL of the uploaded CSV file from the form.
 - **apply_date:** The date of the application submission.
 - **negotiated_amount:** The amount which was negotiated with the client and submitted by a negotiator of the bank based on the predicted trustworthiness of the client.
 - **prediction:** Predicted trustworthiness of a loan applicant.
 - **allowable_amount:** An amount set on the basis of predicted credit risk and company policy.
 - **status:** The stage that the application currently is in. Possible values are: Applied, Reviewed, Negotiated, Granted, Rejected etc.
- **class Branch(models.Model):**
- Information about the branches of the bank is stored here.
- **branch_id:** A unique identification number of the branch.
 - **branch_name:** Name of the branch.
 - **branch_address:** Location of the branch.
- **class Department(models.Model):**
- Information about the departments of a specific branch of the bank is stored here.
- **department_id:** A unique identification number of the department.
 - **branch_id:** ID of the branch
 - **department_name:** Name of the department.

6.2 Employee



6.2.1 employee/urls.py

This python file handles all the urls of the employee module for three types of our employee.

```
path('dashboard/', views.dashboard, name='dashboard'),
path('generatePrediction/<int:loan_id>', views.generatePrediction,
     name='generatePrediction'),
path('FRLList/', views.FRLList, name='FRLList'),
path('', auth_views.LoginView.as_view(template_name='employee/login.html'), name='login'),
path('getCSV/<int:loan_id>', views.getCSV, name='getCSV'),
path('logout_emp/', auth_views.LogoutView.as_view(template_name='employee/login.html'),
      name='logout_emp'),
path('form_reviewer_reject /<int:CarLoanApplication_pk>', views.form_reviewer_reject,
      name='form_reviewer_reject'),
path('NList/', views.NList, name='NList'),
path('CRMLList/', views.CRMLList, name='CRMLList'),
```

6.2.2 employee/views.py

def dashboard(request): This function handles the homepage of all employees.

- Input : Employee's login credentials.
- Output : Shows the options in the UI according to the designation of the employee.

6.2.3 Form Reviewer :

- **def dashboard(request):**

This function handles the homepage of all employees.

- Input : Employee's login credentials.
- Output : Shows the options in the UI according to the designation of the employee.

- **def generatePrediction(request, loan_id):**

This functionality is only for form reviewer. When he clicks on the "Generate" button from his window, This function is triggered and a prediction is generated through the machine learning API.

- Input : Loan application ID.
- Output : Generates the prediction for loan approval.

- **def FRList(request):**

This is the function for showing the applications whose status are still "APPLIED" to the form reviewer.

- Input : HTTP request with employee credentials.
- Output : If the employee is a form reviewer, the applications whose status are "APPLIED", are shown.

- **def form_reviewer_reject(request,loan_id_pk):**

If form reviewer finds the loan application unfit for even feeding into the machine learning API, he can use this functionality.

- Input : Loan application ID.
- Output : The corresponding application is rejected. And added to the rejected list with status="REJECTED_REV".

6.2.4 Negotiator :

- **def NList(request):**

This is the function for showing the applications whose status are still "REVIEWED" to the negotiator.

- Input : HTTP request with employee credentials.
- Output : If the employee is a negotiator, the applications whose status are "REVIEWED", are shown with prediction result and allowable amount.

- **def update_amount(request, loan_id):**

The negotiator can negotiate with the client through messaging and the final amount will have to be submitted.

- Input : Loan application ID, negotiated amount.
- Output : Negotiated amount is the amount that the bank grants to the user. After this step, the status of a loan application is changed to "NEGOTIATED".
- **def nego_reject(request,loan_id_pk):**
If negotiation is not possible with the client, negotiator rejects the application.
 - Input : Loan application ID.
 - Output : The corresponding application is rejected. And added to the rejected list with status="REJECTED_NEGO".

6.2.5 Client Relations Manager(CRM):

- **def CRMList(request):**
This is the function for showing the applications whose status are still "NEGOTIATED" to the CRM.
 - Input : HTTP request with employee credentials.
 - Output : If the employee is a CRM, the applications whose status are "NEGOTIATED", are shown with negotiated amount.
- **def crm_accept(request, loan_id):**
In this stage, CRM will accept the loan application when the personal verification is done.
 - Input : Loan application ID.
 - Output : The bank finally grants the loan to the user. After this step, the status of a loan application is changed to "ACCPETED".
- **def crm_reject(request,loan_id_pk):**
If CRM finds that the informations and documents provided by the applicant is not trustworthy, he rejects the application.
 - Input : Loan application ID.
 - Output : The corresponding application is rejected. And added to the rejected list with status="REJECTED_CRM".

7 Collaboration with Bank Asia and Our Future Work

With the hope to implement our methods in the the existing banking system of Bangladesh, we talked to Bank Asia and they showed sincere interest in the automatic credit risk prediction system. They have already provided their data dictionary and will provide their datasets soon.