1 PAGE SUMMARY

Yeong Hun (Luke) Lee, Kyusub (David) Chung, Chian Yee

**Loan Simulator Application**

Project Design (Attached from Page 2)

**Project Description (4-5 sentences)**

This application will enable the customer to simulate whether his/her potential loan applications are likely to get approved or disapproved. The user will have to type in a few questions that the application is asking, and based on the information, he/she will be assigned the application result. If the application is approved, then the user will be provided the interest rates, the possible loan amount, the approved loan period, and the credit grade. After this, the user can also run a simulator which displays the remaining principal and interests.

**Work Breakdown**

* Yeong Hun (Luke) Lee
  + Created an algorithm to assign the user’s credit grades, interest rates, the loan amount and the loan period
  + Sourced the data file which contains the real life loan application scenarios - more than 1 million data
  + Read the source file and applied the algorithm created into the file.
* Kyusub (David) Chung
* Chian Yee
  + Building the user interface with java swing.
  + Design the flow and integrate the team’s work into one application.
  + Building the file saving feature of the app.

**Link to Github**

<https://github.com/UPenn-CIT599/final-project-team-25-clk>

1. **Customer Log In & Customer creation**

When a customer uses our app, he/she will be prompted asking whether he/she is a new user.

* 1. New User

If the customer is a new user, then a form will appear to ask for customer’s detail such as name, date of birth, job title, annual income, etc. **A customer class object** is created. Customer information will be saved into CSV.

* 1. Old User

If a customer has logged in before, we will ask the customer for a unique id. We then retrieve the customer detail from the CSV file and return a **Customer class** object as logged in user.

1. **Customer Apply Loan**

When a customer applies for a loan, the customer will be prompted to fill in a form that gets details about loan amount applied, loan duration, credit card usage, the number of times customers defaulted in the past. This information will be stored in the **LoanApplication class** object and will be saved into a CSV file as well.

1. **Customer Manage Loan**

A list of the approved loan (**Loan Class Object**) will be presented for customer to manage, ie (pay the instalment , terminate, etc)

1. **Calculate Credit Scores of other users in the database**

With a raw data file (from Lending Club website) that contains approximately 1 million people who have previously been granted their loans from the bank, **Algorithm class** will play a role of calculating each person’s credit score in accordance with the credit score model (this model has been named as “PennCLK score”) of which it is based on FICO score that the commercial banks use in real life.

Once each person is given the PennCLK score, then they will be categorized from A to E, while the score has the highest of 850 points and the lowest of 300 points.

1. **Calculate Credit Score of the user from the LoanApplication object**

Once the user completes entering his information, the LoanApplication object will be parsed onto Algorithm class, and the user’s own PennCLK score and the credit grade will be given.

1. **Loan Application Result**

After the user is given his credit score, he/she will be assigned a credit grade from A to E. With this output, the user’s credit results will be compared to the database and if the user’s score is below 300 points (the credit grade will be below E), then he/she will be rejected for his application. Else, he will be prompted with the message that he is likely to get approved his loan application with approximate interest rates. This interest rate will be calculated based on the weighted average of interest rates per each grade (from A to E) within the database file.

1. **Create Loan Schedule**

Given the loan application result that produces money to be lent, loan period, and the interest rate, calculate the monthly payment that the user needs to pay. Payment class is used to as values in a hashmap to store information (monthly total amount to be paid, monthly principal amount, monthly interest amount, whether the customer paid that month(T/F), how much he paid) for the set of keys, e.g. number of months.

1. **User Interaction to Manage Outstanding Loan**

Given that all loans are to be paid back in increments of 1 month period, the user will be asked 1. to choose the date in which he wants to make the decision on whether to pay and 2. To choose how much to pay. If the user decides not to fully comply with his monthly payment due, the method keeps track of the sum of money not paid in its entirety. When certain conditions are met for termination, loan program ends and calculate the return on investment for the lender.

# 

# 

# 

# CRC Cards

|  |  |
| --- | --- |
| **Customer (basic information about a customer)** | |
| **Responsibilities** | **Collaborators** |
| has user id | Loan Application |
| has name | Loan |
| has annual income |  |
| has job title |  |
| has credit rating |  |
| can manage loan |  |
| can apply loans |  |
|  |  |
|  |  |

|  |  |
| --- | --- |
| **Loan Application (customer apply for a new loan)** | |
| **Responsibilities** | **Collaborators** |
| has loan\_application\_id | Customer |
| has loan\_application\_amount |  |
| has loan\_application\_duration |  |
| has reason\_for\_applying |  |
| has approval\_status (true or false) |  |
| has date\_of\_application |  |
|  |  |

|  |  |
| --- | --- |
| **Admin - ie the Bank** | |
| **Responsibilities** | **Collaborators** |
| Can approve loan | Loan Application |
| Can terminate loan | Customer |
| Can impose penalty fee | Loan |
| Can generate credit scores |  |
|  |  |
|  |  |

|  |  |
| --- | --- |
| **Storage - a class to handle reading and writing from csv** | |
| **Responsibilities** | **Collaborators** |
| can store customer info into csv | Loan Application |
| can get customer info into csv | Loan |
| can store loan info into csv |  |
| can get loan info from csv |  |
| can get loan application from csv |  |
| can store loan application from csv |  |
|  |  |
|  |  |

|  |  |
| --- | --- |
| **UserInterface - will be replaced with Java Swing later** | |
| **Responsibilities** | **Collaborators** |
| can prompt login | Main |
| can prompt for customer instruction | Customer |
| can prompt for existing loan detail |  |
| can prompt for new loan application |  |
|  |  |
|  |  |

|  |  |
| --- | --- |
| **Algorithm (to calculate PennCLK score based on five (or more) components)** | |
| **Responsibilities** | **Collaborators** |
| review database users’ payment history |  |
| review database users’ owed amounts |  |
| review database users’ credit history |  |
| review database users’ pursuit of the new loan |  |
| review database users’ credit mix |  |
| calculate the PennCLK score |  |
| assign the credit grade of a database user |  |

|  |  |
| --- | --- |
| **ApplicationResult - will the user be approved or rejected for his/her application** | |
| **Responsibilities** | **Collaborators** |
| calculate the PennCLK score of the user | Customer, LoanApplication, Algorithm |
| assign the credit grade of the user | Customer, LoanApplication, Algorithm |
| give a result of the loan application |  |
| if approved, calculate interest rates | Customer, LoanApplication, Algorithm |

|  |  |
| --- | --- |
| **Loan class** | |
| **Responsibilities** | **Collaborators** |
| Has principal |  |
| Has interest rate |  |
| Has loan period(months) |  |
| Has loanID |  |
| Has cusomerID |  |
| Has Payment | Payment |
| Has HashMap loanSchedule <month, Payment> |  |
| Take principal, interest rate, loan period(months) and calculate monthly PaymentDue for the borrower. |  |
| Take loan period(months) and PaymentDue to create a hashmap loanSchedule |  |

|  |  |
| --- | --- |
| **Payment** | |
| **Responsibilities** | **Collaborators** |
| Has monthlyPaymentForPrincipal |  |
| Has monthlyPaymentForInterest |  |
| Has monthlyPaymentTotal |  |
| Has boolean payOrDefault |  |
| Has paymentMade |  |

# Libraries

1. Java Swing for UI

# Responsibilities

1. Chian Yee, Lee - handling UI for gathering customer information, log in, loan application information. Also, handling of storage of information into csv files.
2. Yeong Hun (Luke), Lee - set up an algorithm to calculate the credit score of the user of the application and users from the database file (from Lending Club). Afterward, give a result to the loan application based on the algorithm.
3. Kyusub Chung - Given the result of loan application, if the user is approved of a loan, calculate monthly payment. Handle user interaction and manage the loan, i.e. user chooses the date on which he wants to pay the monthly payment, user can choose how much to pay for each month. Set conditions for loan termination and upon such event, calculate return on investment for the lender.

# Overview

Create a loan simulator program that calculates the amount of loan that he or she may borrow.

# Goals

1. Help the client simulate the possibility of being granted a loan regardless of whether this is the first time the user raising a loan or the extension of the existing loan.
2. Automate the process of assigning credit grading, amount to lend, a period of loan for a customer based on personal information and income data. The credit grading, amount to lend and interest rates are based on past historical data obtained publicly from Lending Club.

# Specifications

There are three main functions of this program. They are as follows.

## Get Customer Information (Chian)

1. Ask borrower whether he/she is a new customer or existing customer
2. If new borrower, collect borrower’s name, date of birth, amount of loan to borrow, loan period, job title, income per month.
3. If an existing borrower, ask for previous loan’s amount, loan period, payment pattern (whether pay on time or not)
4. Create a Customer class, PastLoan class and save both this information into CSV file.
5. Basic UI will use Java Swing.

## Analyse past loan data and the application decision based on the user input (Luke)

1. Create a FileReading & Writing class to source the file from Lending Club.
2. From the source file given by Lending Club, the program will incorporate an algorithm of determining the user’s credit grades, expected borrowing rates and available loan period based on the user’s information given.
3. The analysis result, which represents the general credit profile of the user will be parsed into a data decision class - whether to approve or disapprove the loan for the client.
4. Once approved, the user will be shown the decision with the expected approved loan amount, the interest rates and the term of the loan.

## Apply Payment Module to Track Customer Balance (David)

1. Given a loan, this module will calculate how much interest the user needs to pay for the given time frame.
2. If the user pays the interest, move onto the next payment schedule. If the user does not pay the interest in full, incur a penalty fee that will have to be paid along with the principal at maturity. If the interest is paid partially, he is to go default. This will lower his credit rating(rules need to be applied here) and increase the interest rate for future payments.
3. Track the balance of the customer. In the end, if the user pays the principal, print out the real interest rate that the investor has paid over the total loan period. If the user does not pay the principal, he is to go default. For the loan that was given out to the user, calculate the actual return on investment that the bank has accrued given all the payment that the user has made over time.

# Milestones

## Project Proposal - 27 Oct 2019

Basic description of what we are building and allocation of work.

## Project Design - 10 Nov 2019

## Final Project Submission- 8 Dec2019

TBC