Loan Management System Design Documentation

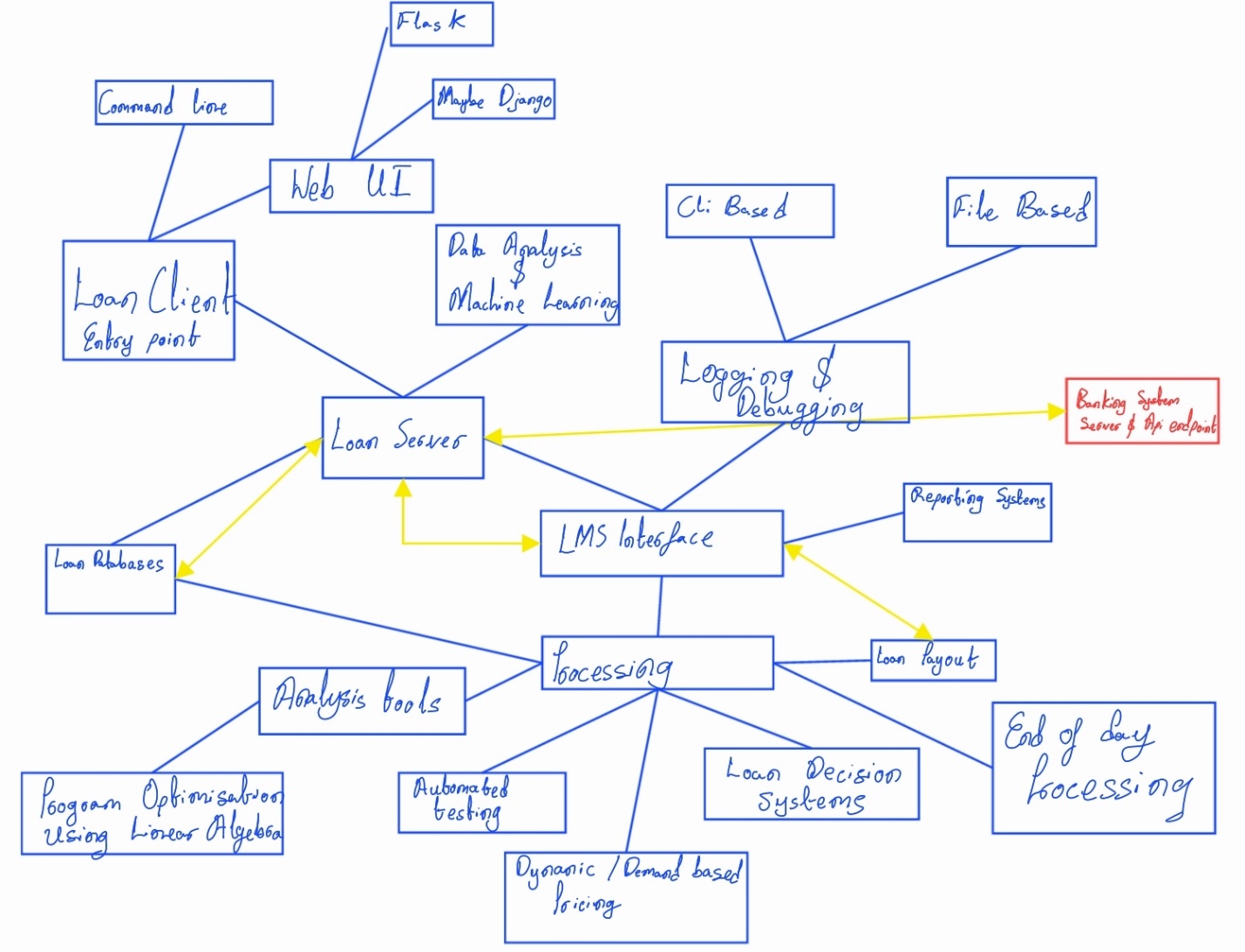
Project Description:

This project is a Loan Management System using a client – server model, that could get user data (human or generated) and processes it for credit scoring. It factors in the current economic conditions like yield spread, federal reserve rate, loan demand from previous day, e­tc.

Core Objectives and features:­

1. Generate artificial Data.
2. Client Server Functionality over a local Network.
3. A login in system for user to view their loan profiles.
4. A command line client for users and developers.
5. A web client for users and possibly for developers
6. A proper tracking system to track applied, current, and completed loans.
7. A Builtin Database Management System
8. A Basic Loan Reporting System to report statistics like number of loan applicants, average loan amount requested, average credit score of the applicants for the day, month, and year.
9. An api to send jsonified loan reports

Guidelines Resulting From Experimentation and increased Programming Experience



All User Data:

Data that will be received from user:

* + 1. User name
    2. User password for future loin methods
    3. User Financial Reserves
    4. User Requested Loan Amount
    5. User loan Duration
    6. User Credit Score

Calculated Data From User:

* + 1. Default Risk Score
    2. Loss Given Default
    3. Debt to income ratio
    4. Recovery rate for each loan
    5. Monthly debt payments for each loan application.
    6. Outstanding monthly debt payments prior to loan for the month
    7. Total monthly payments to us for the loan each month.
    8. Default Risk Score
    9. Loan Viability Score
    10. Adjusted Loan Viability Score
    11. Credit Score based default risk rate by group

Loan Server Design and Functionalities:

1. Receive and return Jsonified customer data.
2. Receive and return options for developer (DEV) menu.
3. Direct ability to directly search loan database for entries­.
4. Generate artificial user input when requested.
5. Handle processing of JSON requests from Clients prior to actual loan processing.
6. Handle compilation of back-end when necessary.
7. Handle API calls for necessary economic info.

Changes to make on Loan Server and LMS Interface

1. Fix bug on line 720
2. Make economic metrics be stored in a file to prevent repetitive API calls.

Changes to make on Loan Database Manager

1. Flatten as much as possible the functions involved In data storage and retrieval.

What to do:

* + 1. Just directly concatenate the loan values instead of doing this with a for loop.
    2. Comment out the logic because it is weird how I came up with that solution.

1. Split database value retrieval and writing to the required file in the retrieveAllUserDataFromDatabase Function.

File Management Changes:

1. Use a mutex to handle all output file objects file objects.
2. File input could be left to be done by multiple threads simultaneously.

Loan Management System (LMS) Interface

1. API for Reporting loan Statistics.
2. Methods of client verifying access to the server.
3. Methods of sending and receiving API calls to Bank Management System.
4. Methods of accepting and processing generated data
5. Methods of accepting and processing standalone applications.

Processing Functionalities and Features:

1. Compute loan application dependent Metrics.
2. Compute interest rate for each loan application.
3. Calculate final Adjusted Loan Viability Score.
4. Store processed Values in applicable Databases.

Dev Menu Functionalities and Functions:

1. Generate user Database
2. Store user generated data directly in CSV file for future analysis. Give access to different types of output such as those with no passwords or names present, etc.
3. Store generated user data directly in database. Possibly provide methods of prioritizing real user applications over generated applications when storing data in database. 