List of Content

TITLE

LOAN MANAGEMENT SYSTEM

Introduction And Objectives

Providing a loan should be a simple process. One should check the client's eligibility to get the loan and then approve or deny the loan. Once approved, the customer should receive the funds.

However, in traditional lending systems, particularly in larger organizations, this process is often chaotic–and for valid reasons. As the customer base increases, servicing loans become complex. Every customer has different terms and payment dates. It is cumbersome to keep everything in order.

That is why lenders use loan management software to streamline their process.

A loan management system helps to sort out the repayments that are coming in. But it can do more. There are modular, scalable, and customizable components that organizations can use for complete automation.

Top-of-the-line products can even use machine learning algorithms to reduce risks. Small business loans kept increasing and reached 28 billion USD in 2019. So, getting a digital platform in this competitive market is crucial for survival.

Purpose

To automate the entire loan lifecycle. The software can help with processing customer information, create new loans, and more. They can also provide lenders with accurate statements and reports. Moreover, they can manage interest rates and provide the tools for collection automation.

These automated loan management/lending systems outshine legacy systems in many ways. Being a digitized system, it also caters to the newer generation of customers. It also reduces manual errors and risks.

Scope of the System

- Our Loan Management System has two main interfaces. One is for customers and other is for Loan Provider Company (Admin).
- Customer can apply for a Loan through from the website. Customer need to fill their loan requirements with their basic personal details.
- Customer can view Loan Details, Interest Rate, Repayment Schedule etc.
- Customer can make loan payment through online payment methods (Debit Card, UPI, Wallet Options etc.). After the payment, system generates receipt and send to the customer email-id.
- There will be a manual process if customer applies for secured loan.
- EMI Due Reminders will be sent to customers on their registered email-id.
- Customer can view the status of the loan application.
- Administrator can view the unpaid customer details, Total Loan Disbursed (Monthly/Yearly) etc.

Benefits of the System

- ✓ **Eliminating human error**
    - It's no secret, that calculations are something that algorithms handle better than we, humans. In a lending system, there are just too many variables, which is why it is error-prone. The best loan servicing software, however, is created to completely rule out any errors, which is, undoubtedly, beneficial from every standpoint.

- ✓ **Preventing delays in payment**
    - Not being able to collect a debt is something that most lenders are especially wary of. However, if they leverage a traditional loan management approach, they may not see it coming. Loan servicing systems, on the other hand, integrate analytic modules capable of detecting even the most subtle fluctuations in clients' credibility and preventing payment delays in a timely manner.

- ✓ **Saving time**
    - Loan management requires a great level of meticulousness and attention to detail. As a rule, a full-fledged team is required to deal with every aspect of a loan process. Needless to say, loan management carried out manually and based on paperwork takes up a lot of time. A digital lending system, on the other hand, automates the routines and enables your team to dedicate time to other important tasks.

- ✓ **Increased revenue**
    - This stems from all of the above: an automated loan processing system enables lenders to process more applications, assign and manage more loans, and see them all the way through closing all while detecting scams and preventing delays. The staff is free to oversee the process and focus on client relationships and look for new business opportunities. This enables financial companies to gain a distinct competitive edge and increase revenue.

Tools/Platform & Hardware

Front-end (Bootstrap Framework)

Bootstrap is a free and open-source CSS framework directed at responsive, mobile-first front-end web development. It contains HTML, CSS and JavaScript-based design templates for typography, forms, buttons, navigation, and other interface components.

Back-end (Python/Django Framework)

Django is a Python-based web framework, free and open-source, that follows the model–template–views architectural pattern. It is maintained by the Django Software Foundation, an independent organization established in the US as a 501 non-profit.

Database (MySQL)

MySQL is an open-source relational database management system. As with other relational databases, MySQL stores data in tables made up of rows and columns. Users can define, manipulate, control, and query data using Structured Query Language, more commonly known as SQL. A flexible and powerful program, MySQL is the most popular open-source database system in the world.

Hardware Requirements

- Minimum Intel i3 7th Gen (32/64 Bit)
- Windows 7,8.1,10 or Linux/Chrome OS/MAC
- 1280 x 800 minimum screen resolution
- 2.40 Ghz Processor
- 4 GB RAM minimum, 8 GB RAM Recommended

Functional Requirements

1) Register:

Input: Customer need to fill their basic details (First Name, Last Name, Email-ID, Contact No, Create New Password).

Processing: All details will be checked and if any error found then an error message is displayed else customer is successfully registered with us.

Output: Confirmation of Registration status and customer can now redirect to the home screen

2) Login:

Input: Customer need to enter the Email-ID and Password that was provided at the time of registration.

Processing: Email-ID and Password will verify. If any error found then an error message is displayed else customer is successfully logged in our system.

Output: Customer redirect to home screen on successful authentication.

3) Customer apply for loan:

Description: Customer need to select the loan on the basis of their requirements.

Input: Customer have to fill up their basic details (Aadhar No, PAN No, Current Salary, Office Address, Home Address, Loan Amount, Loan Tenure etc.)

Processing: All details will be checked and if any error found then an error message is displayed else the loan application is successfully created.

Output: Loan Application details will share to customer registered email-id.

4) Administrator review Loan Application:

Description: Administrator review the loan application created by customer and administrator can decide whether to Approved or Reject the Application.

Input: If Loan Application is rejected, administrator needs to add the remarks else Administrator confirm and approved the loan application.

Output: Status of the loan application is sent to customer registered Email-ID.

5) Customer repay the Loan EMI:

Description: Customer need to pay their loan EMI timely. So, Customer can pay their EMI on Repayment Date through our website with different payment methods available.

Input: Customer have to fill up their Credentials in order to repay their EMI smoothly.

Processing: Credentials will be checked and if any error found then an error message is displayed else the payment is successful.

Output: Payment is Successful and the receipt will be sent to the customer registered Email-ID.

Non-Functional Requirements

Usability Requirement

The System shall allow the users to accept the system from any device. Since All users are familiar with the website, no special training is required. The system is user friendly which makes the system easy.

- **Availability Requirement:** The system is available 100% for the user and is used 24 hours a day and 7 days a week.
- **Efficiency Requirement Mean time to Repair:** (MTTR) even if the system fails the system will be recovered or backup within a hour or less.
- **Accuracy:** The system should accurately provide real time information taking consideration various concurrency issues. The System shall provide 100% at this reliability.
- **Reliability Requirement:** The System has to be so percent reliable due to importance of data and the damages that can caused by incorrect or incomplete data, the system will run 7 days a week and 24 hours a day.

Reason of using Python/Django as a Backend

Django is a high-level Python web framework that enables rapid development of secure and maintainable websites. Built by experienced developers, Django takes care of much of the hassle of web development, so you can focus on writing your app without needing to reinvent the wheel. It is free and open source, has a thriving and active community, great documentation, and many options for free and paid-for support.

Django helps you write software that is:

- **Complete**

  Django follows the "Batteries included" philosophy and provides almost everything developers might want to do "out of the box". Because everything you need is part of the one "product", it all works seamlessly together, follows consistent design principles, and has extensive and up-to-date documentation.

- **Versatile**

  Django can be (and has been) used to build almost any type of website — from content management systems and wikis, through to social networks and news sites. It can work with any client-side framework, and can deliver content in almost any format (including HTML, RSS feeds, JSON, XML, etc).

- **Secure**

  Django helps developers avoid many common security mistakes by providing a framework that has been engineered to "do the right things" to protect the website automatically.

  For example, Django provides a secure way to manage user accounts and passwords, avoiding common mistakes like putting session information in cookies where it is vulnerable (instead cookies just contain a key, and the actual data is stored in the database) or directly storing passwords rather than a password hash.

- **Scalable**

  Django uses a component-based "shared-nothing" architecture (each part of the architecture is independent of the others, and can hence be replaced or changed if needed).

  Having a clear separation between the different parts means that it can scale for increased traffic by adding hardware at any level: caching servers, database servers, or application servers. Some of the busiest sites have successfully scaled Django to meet their demands (e.g. Instagram and Disqus, to name just two).

- **Maintainable**

  Django code is written using design principles and patterns that encourage the creation of maintainable and reusable code. In particular, it makes use of the *Don't Repeat Yourself (DRY)* principle so there is no unnecessary duplication, reducing the amount of code.

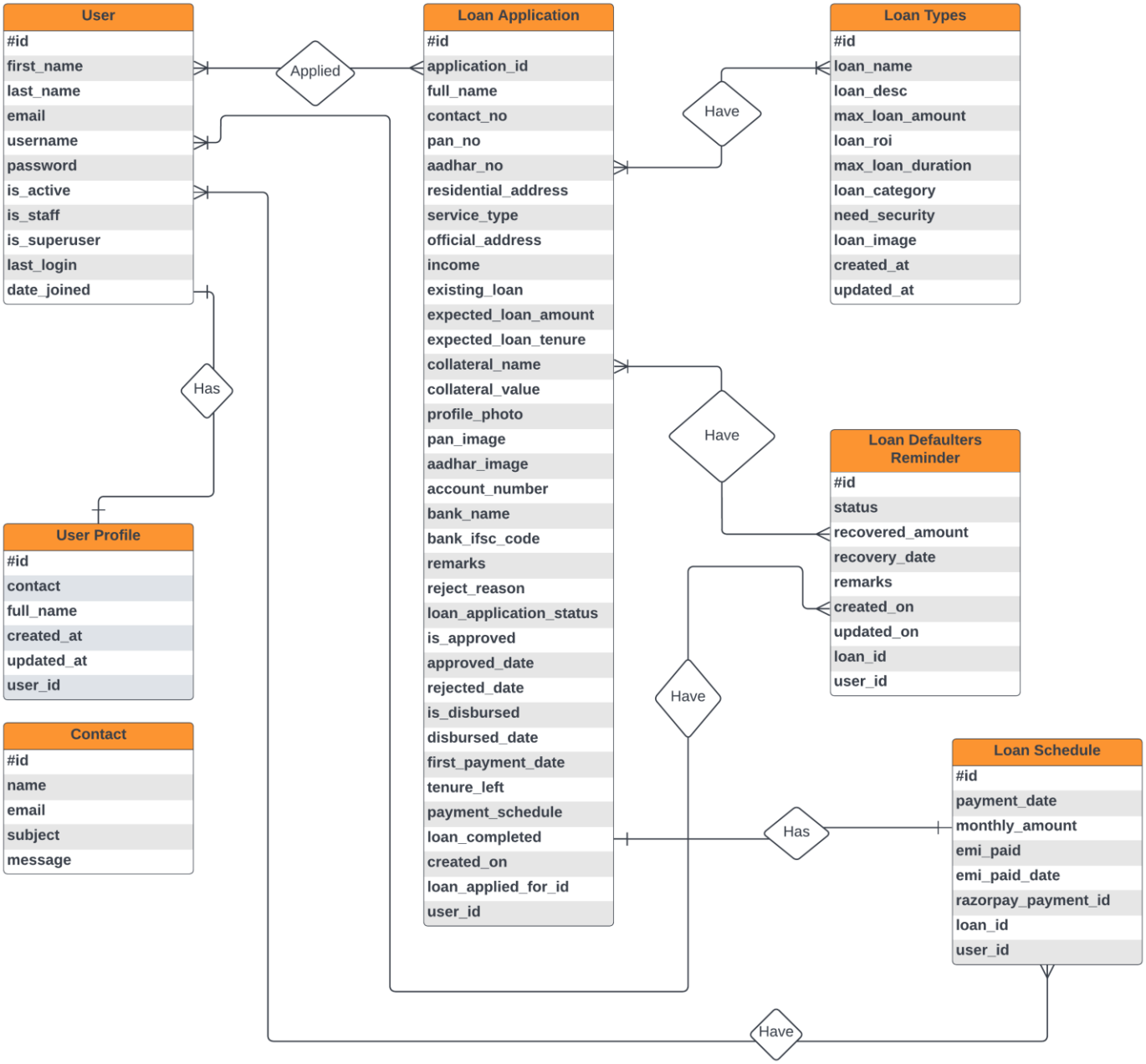
- **Portable**

  Django is written in Python, which runs on many platforms. That means that you are not tied to any particular server platform, and can run your applications on many flavors of Linux, Windows, and macOS.

  Furthermore, Django is well-supported by many web hosting providers, who often provide specific infrastructure and documentation for hosting Django sites.

ER-Diagram

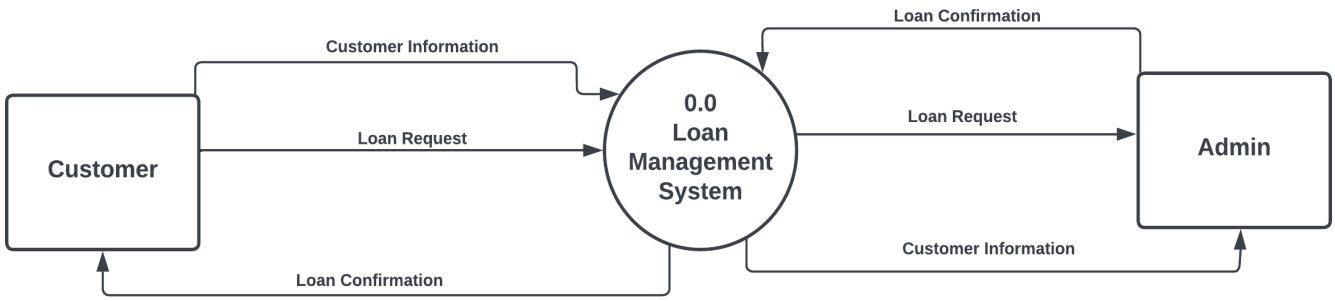
An Entity–relationship model (ER model) describes the structure of a database with the help of a diagram, which is known as Entity Relationship Diagram (ER Diagram). An ER model is a design or blueprint of a database that can later be implemented as a database. The main components of E-R model are: entity set and relationship set.



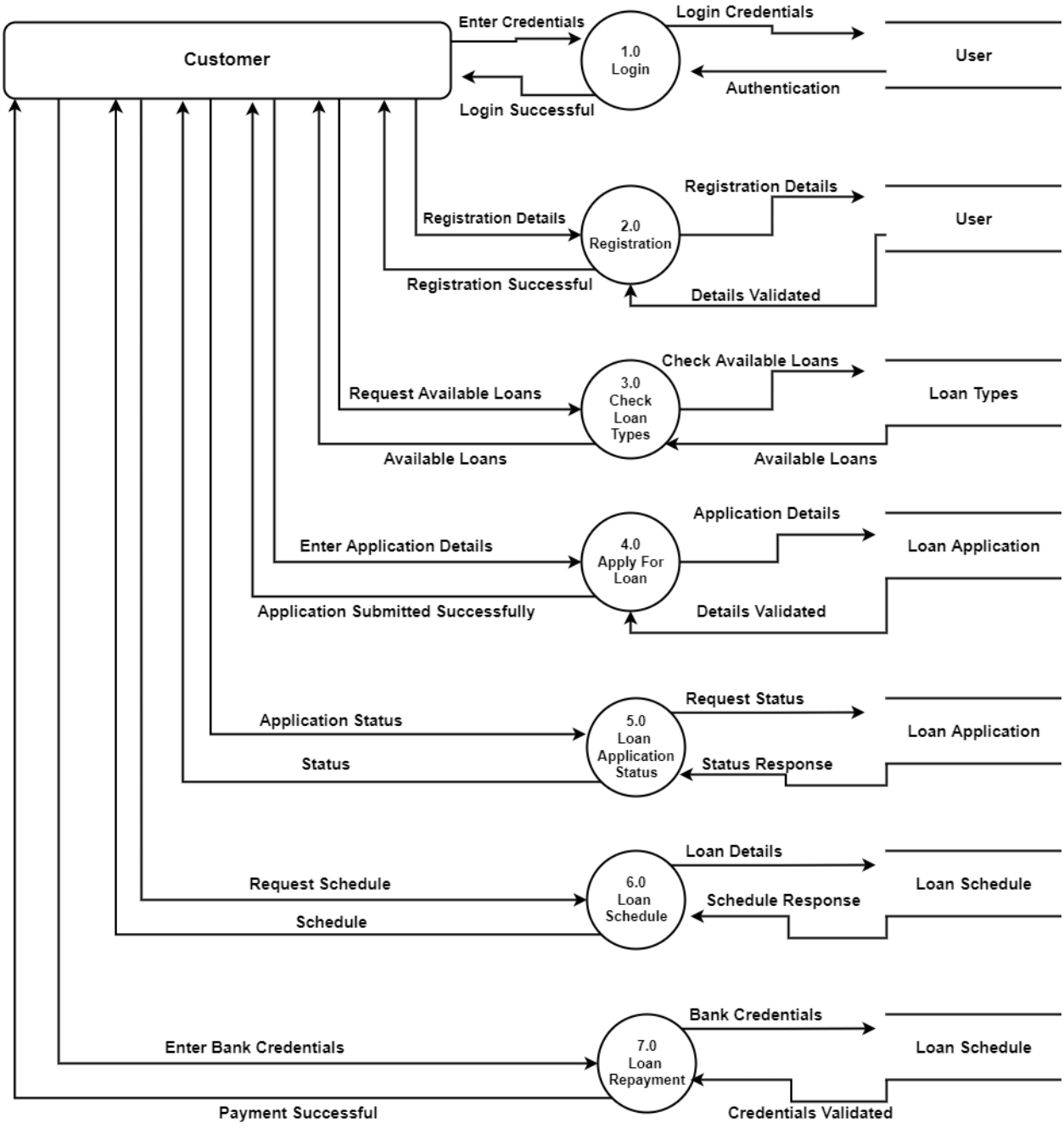
Data Flow Diagram

Data flow diagram is graphical representation of flow of data in an information system. It uses defined symbols like rectangles, circles and arrows, plus short text labels, to show data inputs, outputs, storage points and the routes between each destination.

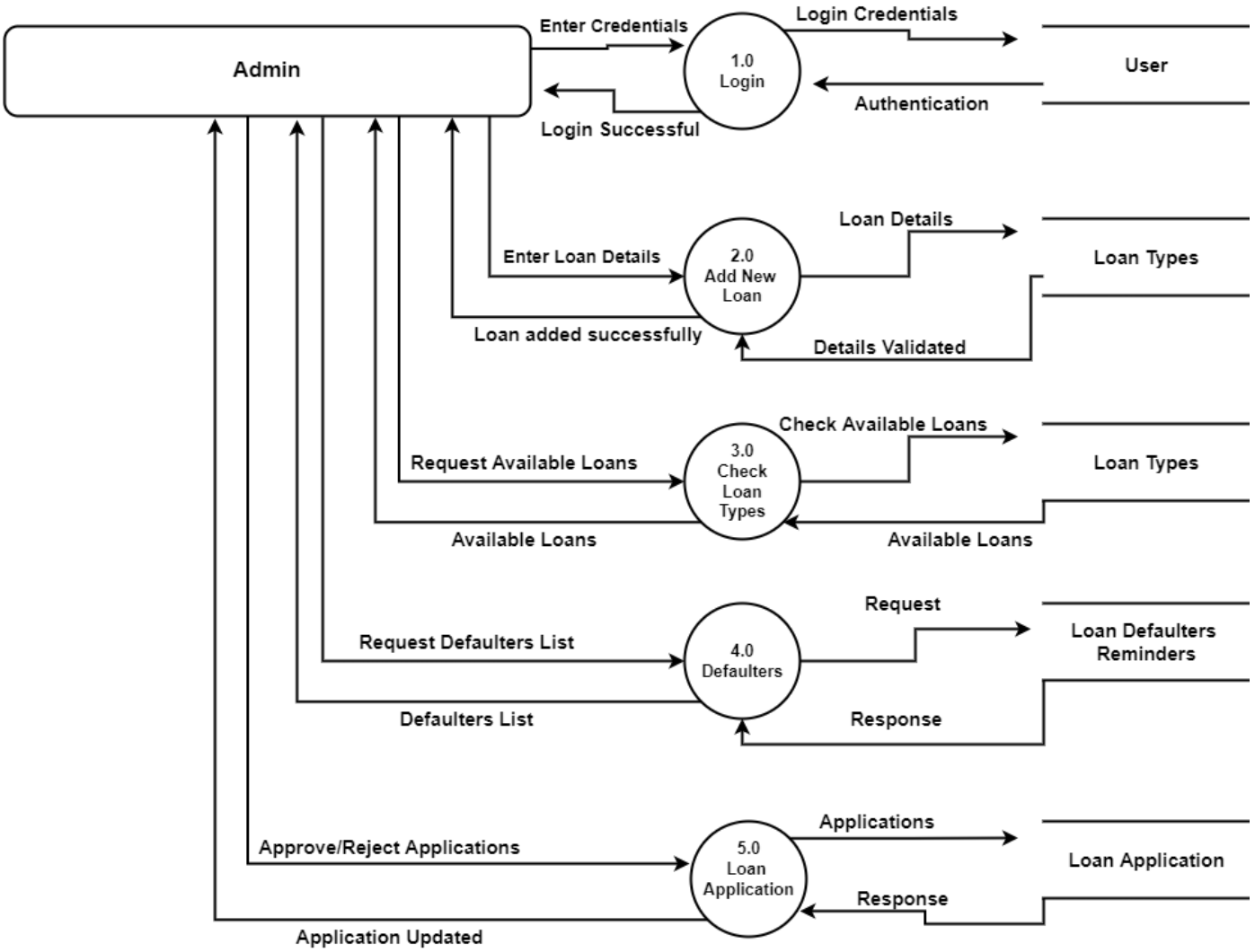
DFD 0 Level: The 0 Level DFD shows flow of data of application. DFD Level 0 is also called a Context Diagram. It's a basic overview of the whole system or process being analyzed or modeled.



DFD 1 Level: DFD Level 1 provides a more detailed breakout of pieces of the Context Level Diagram. This DFD describes main functions carried out by the system, as we break down the high-level process of the Context Diagram into its subprocesses.



DFD 1st Level (Customer)

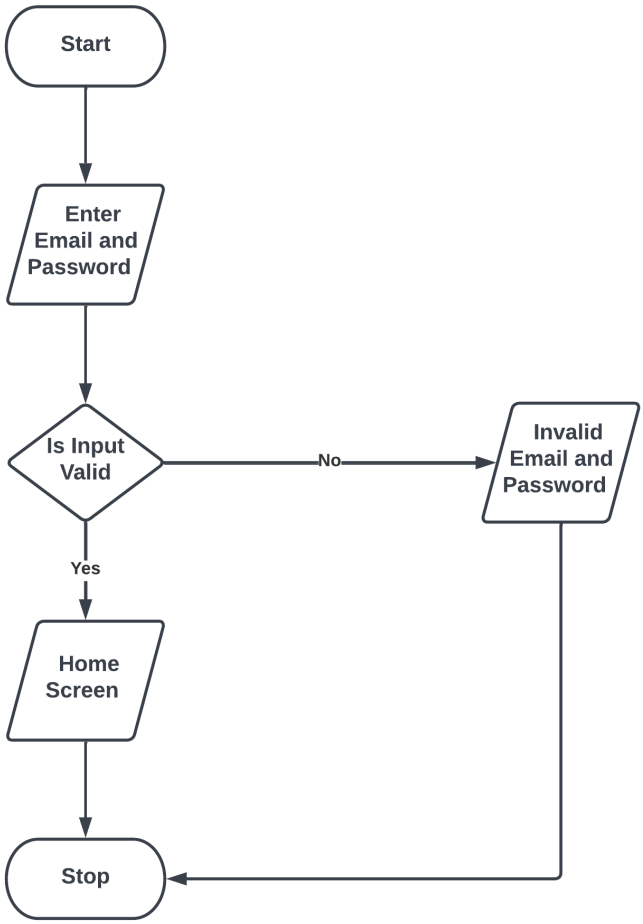


DFD 1st Level (Admin)

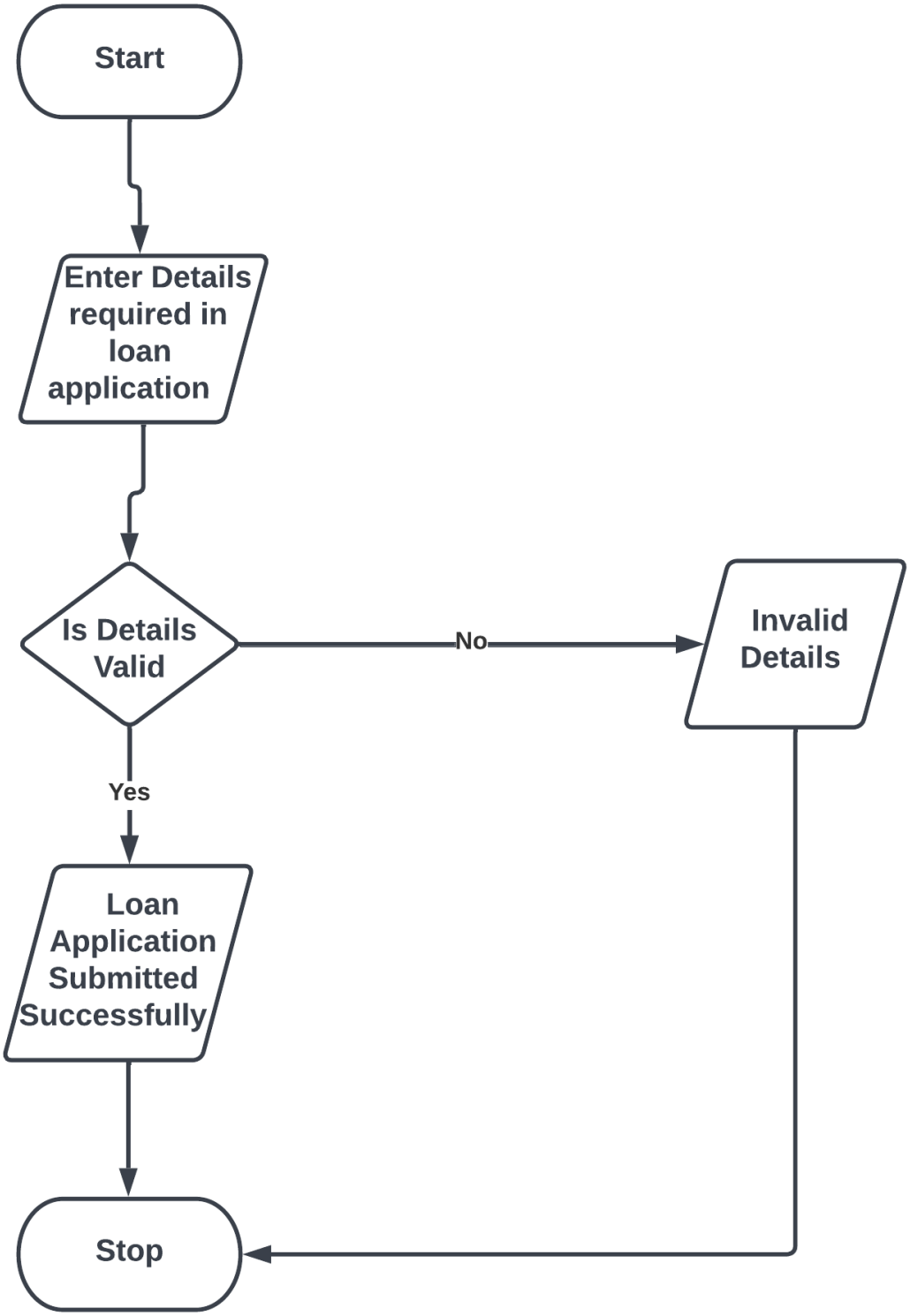
Flow Chart

A flowchart is a picture of the separate steps of a process in sequential order. It is a generic tool that can be adapted for a wide variety of purposes, and can be used to describe various processes, such as a manufacturing process, an administrative or service process, or a project plan. It's a common process analysis tool and one of the seven basic quality tools.

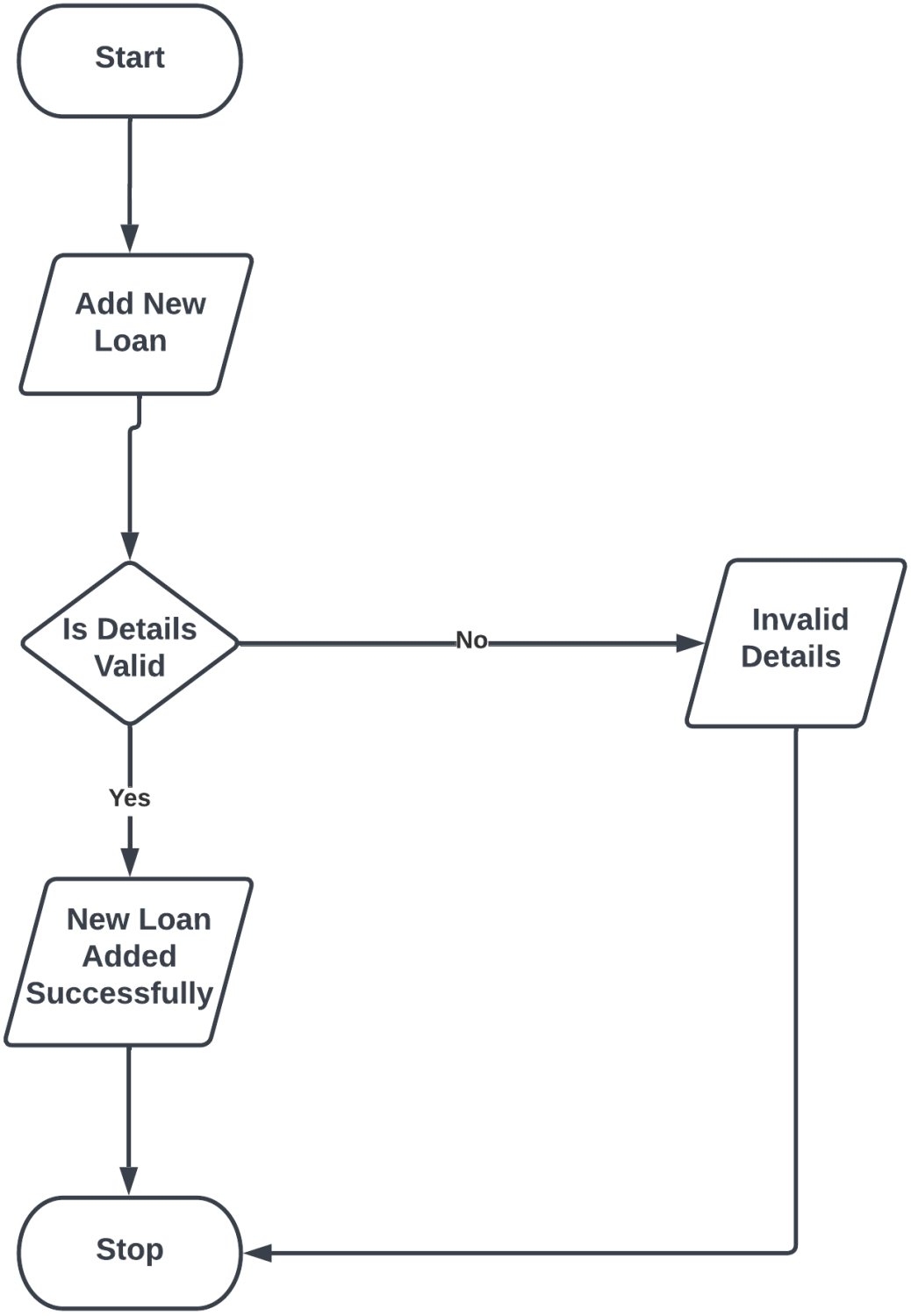
Elements that may be included in a flowchart are a sequence of actions, materials or services entering or leaving the process (inputs and outputs), decisions that must be made, people who become involved, time involved at each step, and/or process measurements.



Login Flow Chart



New Loan Application Flow Chart



New Loan Type Flow Chart

Input/Output Modules of the Project

Input Modules:

- Customer Registration
- New Loan Application
- Loan Repayment
- New Loan Type
- EMI Calculator

Output Modules:

- Loan Application Status
- Loan Types
- Payment Status
- Loan Applications

BIBLIOGRAPHY

For IDE Installation:

https://code.visualstudio.com/

For XAMPP Installation:

https://www.apachefriends.org/download.html

Documentation Reference:

https://docs.djangoproject.com/en/4.0/

https://www.w3schools.com/python/

https://www.w3schools.com/mySQl/default.asp