Project Deliveries 3
Project Title: PayHabProject
Team Name: Dream Team

1. Introduction

PayHab is a project designed to help students track their credit-based transactions with Habib university vendors such as Raheem Bhai,Tapal etc. Many students do not carry cash but still purchase food or other items from on-campus vendors, who allow them to pay later. However, students often forget their dues, leading to confusion and losses. Our solution provides a digital way to track these transactions and ensure timely payments.

2. Function and Non functional requirements

2.1 Functional Requirements (FRs):

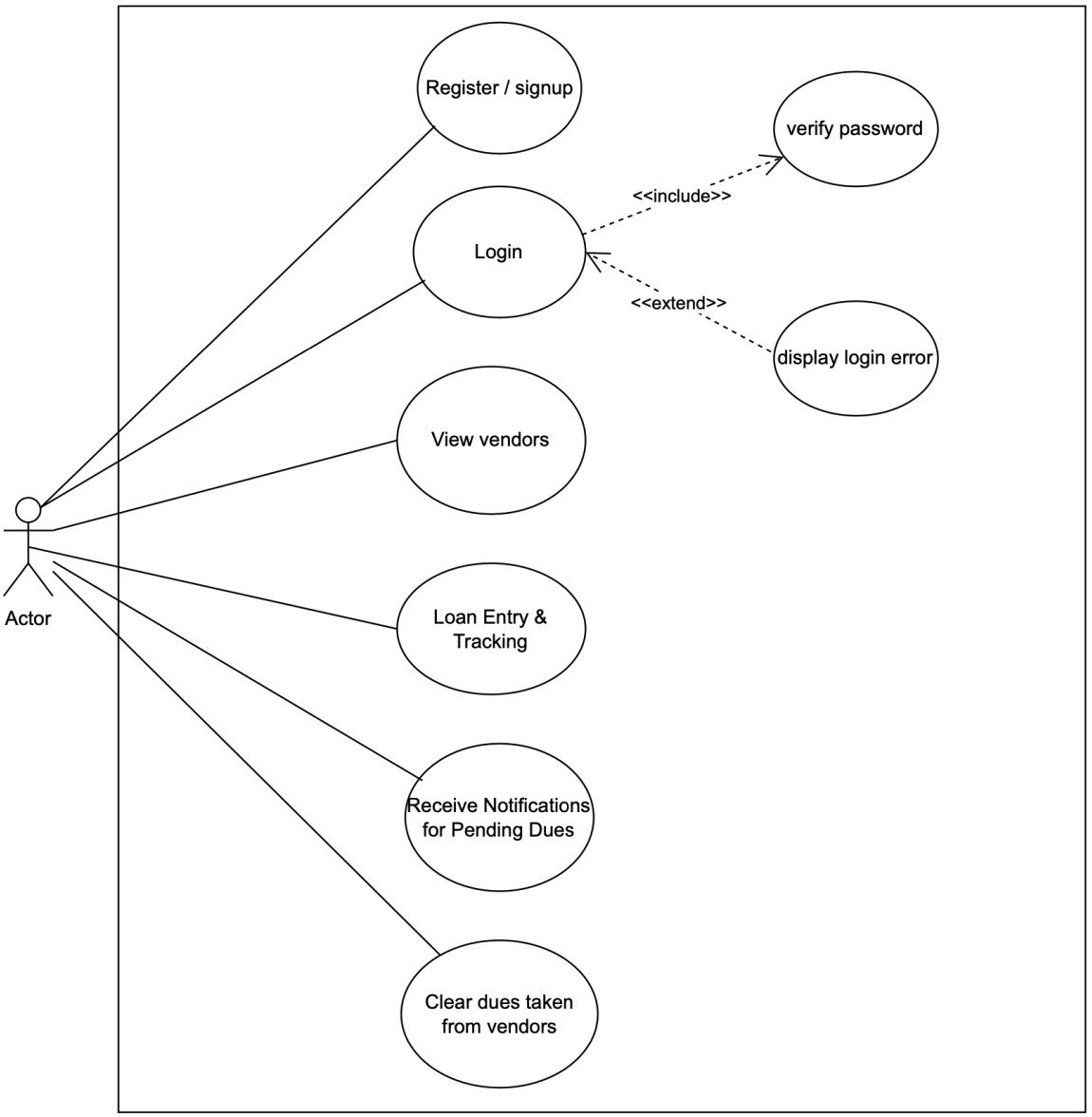
These define the core functionalities that the system must provide:
Registration / Signup – User can sign up putting ID and creating a new password
User Authentication – Secure login using student ID and password.
Dashboard – Displays vendors, user and loan amounts.
Loan Entry & Tracking – Users can enter loans taken from vendors, which appear as outstanding dues.
Loan Payment – Students can clear off their dues through the app.
Notifications – The system sends reminders to students about unpaid loans.
Real-time Data Sync – Firebase ensures real-time updates between users and vendors.
RESTful API Communication – The app interacts with the backend through APIs.

2.2 Non-Functional Requirements (NFRs)

These define system attributes such as performance, security, and usability:
Security – Firebase authentication and security rules ensure authorized access.
Scalability – The app should efficiently handle increasing users and transactions.
Performance – Loan tracking and payments should process in real time.
Usability – The UI (Flutter) should be user-friendly and intuitive.
Reliability – System should function without crashes or data loss.
Maintainability – Modular backend (Flask) for easy updates and debugging.
Cross-Platform Support – The Flutter frontend ensures compatibility with Android and iOS.
Data Integrity – Input validation mechanisms in Flask and Firebase prevent incorrect entries.

3. Use Case Diagram

Diagram here:



Explanation:

The Use Case Diagram represents how students interact with the PayHab system to track and clear their loans from university vendors.

3.1 Actor (User / Student)

- **The actor represents the student who uses the PayHab system.**

- **The student can perform several actions such as registering, logging in, tracking loans, and clearing the dues from their side.**

3.2 User Authentication (Register & Login)

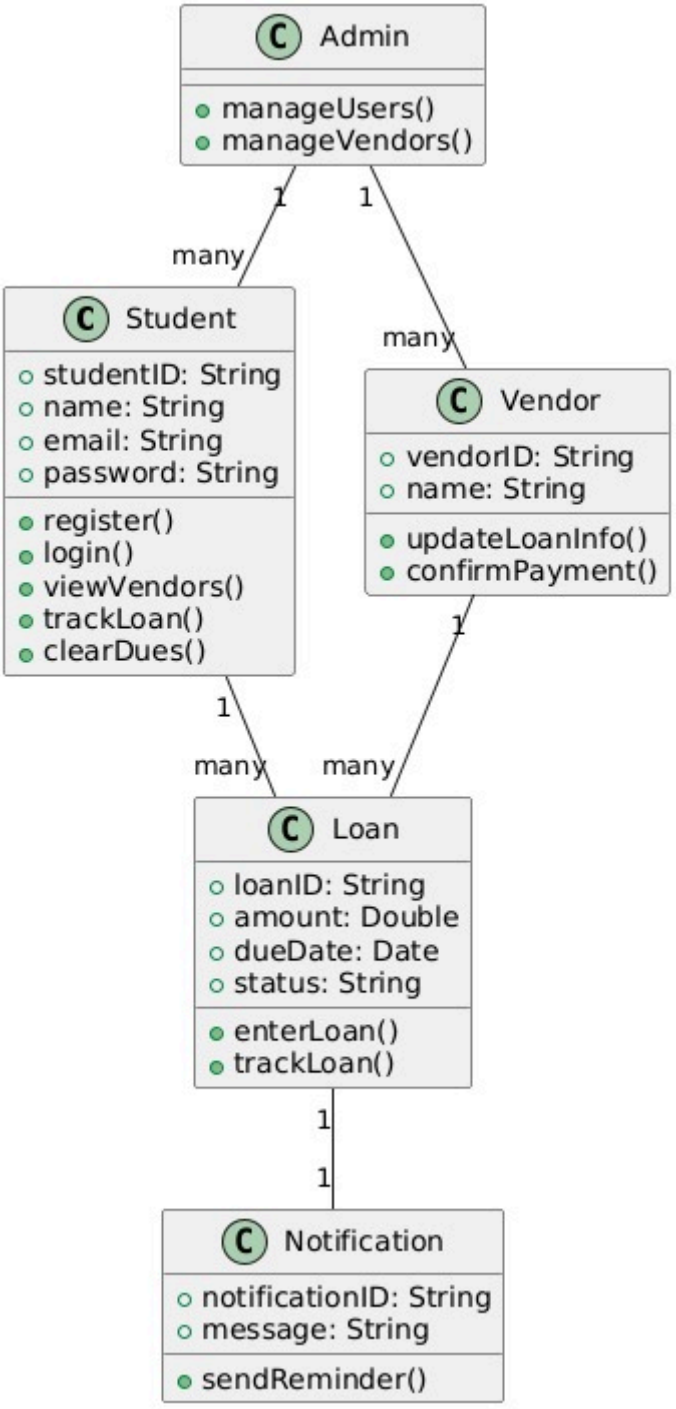
- **Register / Signup:**
  - **This is a separate use case that allows new users to create an account using their student ID and password.**
  - **It is directly connected to the actor with a solid line because it is an independent action.**
- **Login:**
  - **Once registered, the user can log in using their credentials.**
  - **There is no <<include>> relationship between register and login because a user does not register every time they log in.**
  - **The login process includes password verification and extends to a "display login error" case if authentication fails.**

3.3 Dashboard & Vendor Interactions

- **View Vendors: After logging in, users can see a list of vendors they owe money to.**
- **Loan Entry & Tracking: Users can record new loans taken from vendors, which will be tracked in the system.**
- **Clear Dues Taken from Vendors: Students can clear off their outstanding amounts using the app themselves if they have paid it off.**
- **Receive Notifications for Pending Dues: If a loan remains unpaid, the system sends reminders to students.**

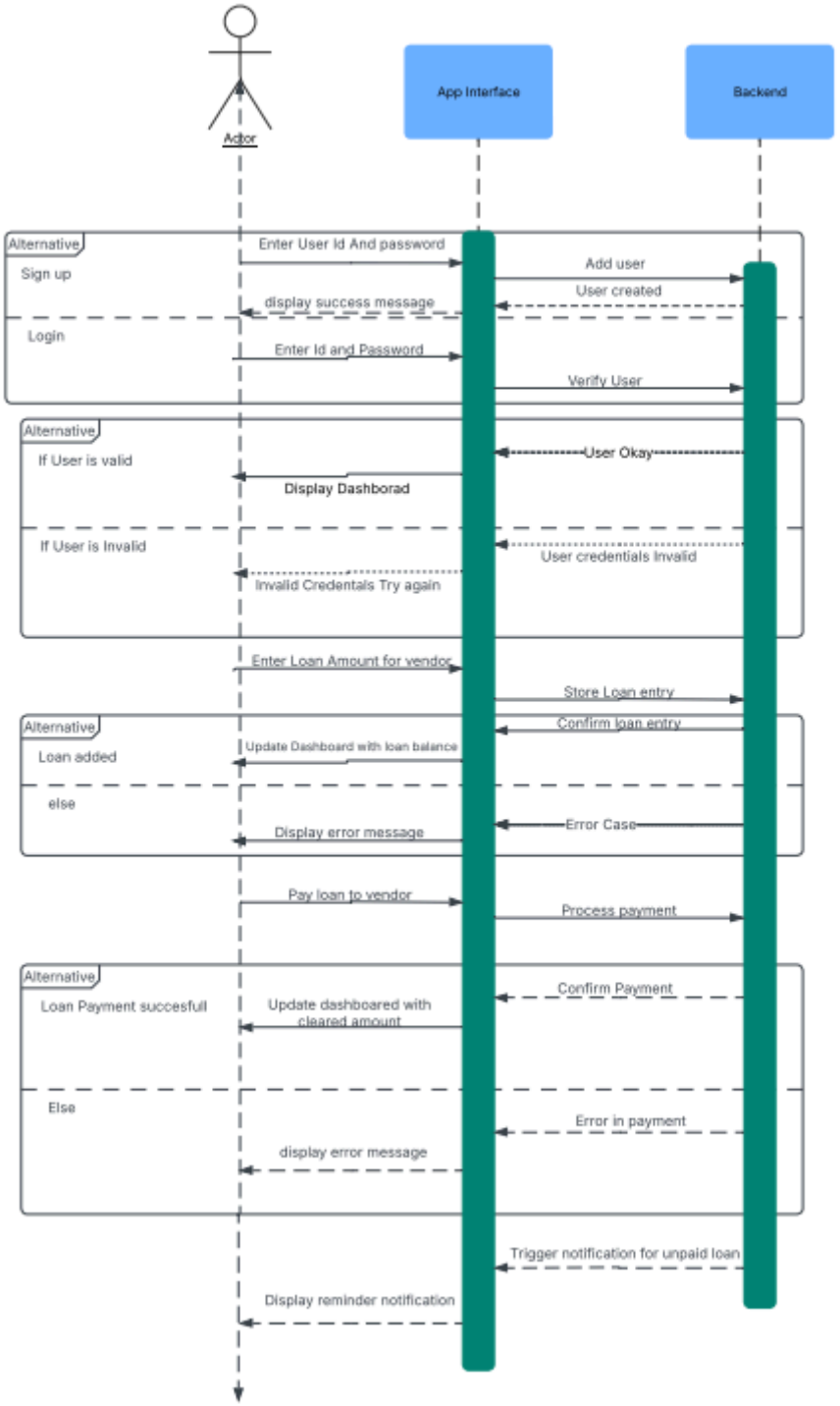
4. Class diagram

Diagram here:



5. Sequence Diagram

Diagram here:


Actor
App Interface
Backend
Alternative
Sign up
Enter User Id And password
Add user
User created
display success message
Login
Enter Id and Password
Verify User
Alternative
If User is valid
User Okay
Display Dashborad
If User is Invalid
User credentials Invalid
Invalid Credentals Try again
Enter Loan Amount for vendor
Store Loan entry
Alternative
Loan added
Confirm loan entry
Update Dashboard with loan balance
else
Error Case
Display error message
Pay loan to vendor
Process payment
Alternative
Loan Payment succesfull
Confirm Payment
Update dashboared with cleared amount
Else
Error in payment
display error message
Trigger notification for unpaid loan
Display reminder notification