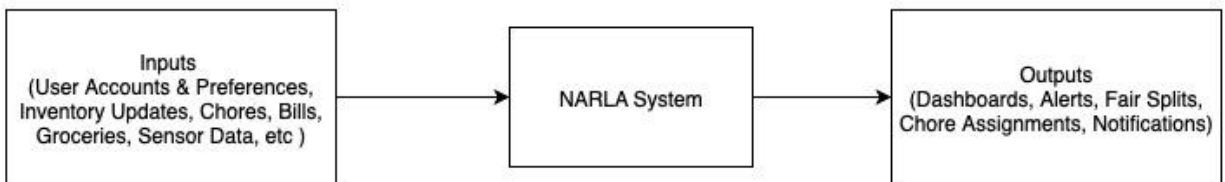


## ASSIGNMENT – 4

- **Project:** NARLA - AI-Driven Apartment Co-Pilot
- **Team:** Neha Ross Lenin, Aditya Anand
- **Goal Statement:** Help roommates and students fairly manage chores, groceries, and shared bills with smart automation, clear accountability, and gentle reminders.

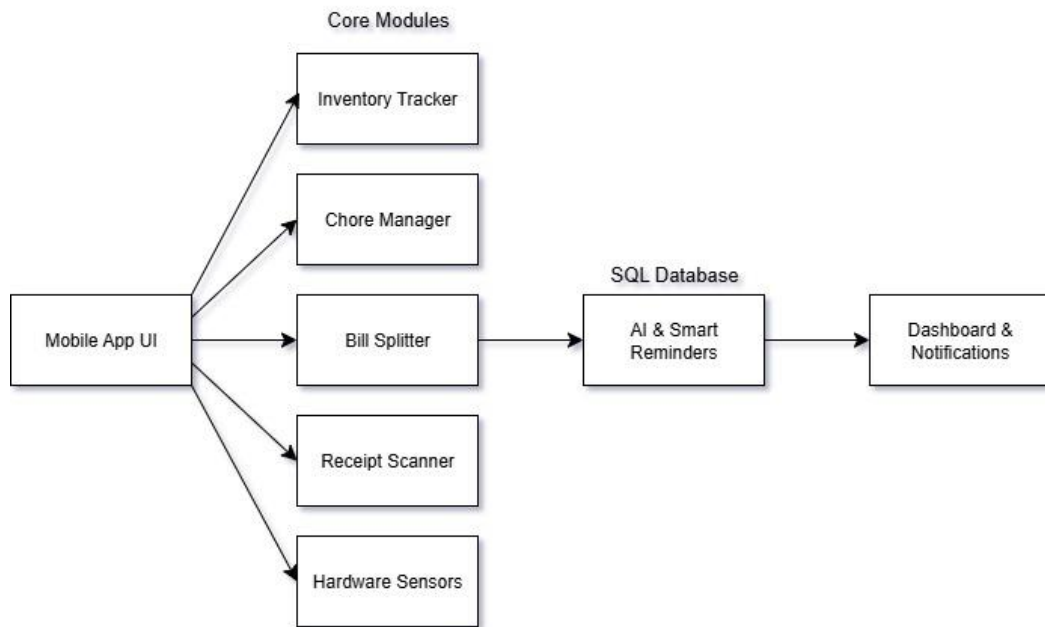
### Design D0:

In this diagram, each box represents a major element of the system at the highest level: Inputs, the NARLA System itself, and Outputs. The arrows/lines represent the flow of data into and out of the system. This graphical depiction shows what our project will do without going into internal details. It emphasizes the basic inputs (such as user accounts, receipts, chores, and sensor data) and the main outputs (dashboards, alerts, fair splits, chore assignments, and notifications). The focus is on illustrating how the NARLA system transforms user and environmental data into meaningful outputs for roommates.



## Design D1:

In this diagram, each box represents one of the main modules or subsystems of NARLA. The Mobile App UI is the entry point where users interact with the system. From there, inputs flow into the Core Modules (Inventory Tracker, Chore Manager, Bill Splitter, Receipt Scanner, and Hardware Sensors), which handle the essential apartment management functions. The SQL Database serves as the central storage hub, allowing all modules to read and write persistent data. The AI & Smart Reminders module uses stored data and module outputs to generate predictions and proactive nudges. Finally, arrows between the boxes represent the flow of data between modules, showing how user input is transformed into meaningful outputs displayed in the Dashboard & Notifications for roommates. This depiction highlights the modular breakdown of the system while maintaining a focus on inputs and outputs at the subsystem level.



## Design D2:

This diagram provides the most detailed view of the NARLA system. Each box represents a specific subsystem, technology, or interface, while the arrows/lines show the flow of data and interactions between them. The User interacts with the system through the Mobile App UI (React Native/Flutter), which connects to the Backend Server (Python). The backend coordinates all modules, including the Chore Manager, Bill Splitter, Inventory Tracker, and AI Prediction Engine, while maintaining persistent records in the SQL Database (users, inventory, bills, chores). Additional components such as OCR Receipt Scanning and Weight Sensors (Raspberry Pi) feed data into the Inventory Tracker, enabling automatic updates through AI and sensor integration. Outputs are delivered back to users via the Dashboard View and Alerts & Reminders (Push Notifications), ensuring transparency, accountability, and proactive support. This diagram emphasizes not only the flow of inputs and outputs but also how the backend orchestrates modules and integrates advanced features for a seamless user experience.

