(Day-1)

Requirements Gathering from Restaurant (BRD-Business Requirement Document)

Process-1:

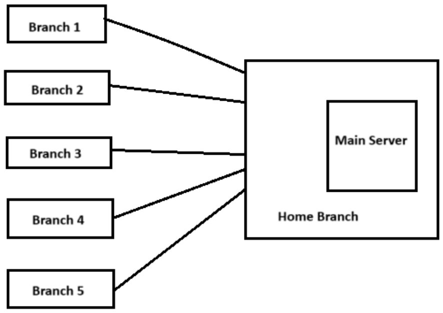
Business Analyst:

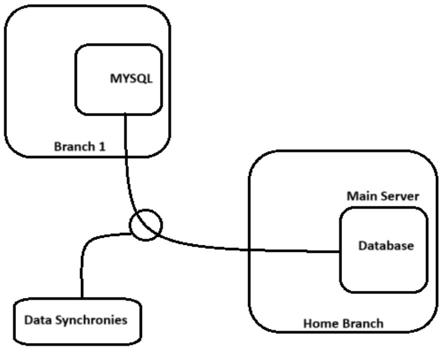
After the Business Requirements Document (BRD), the Business Analyst is providing the functional requirements to the Platform Architect.

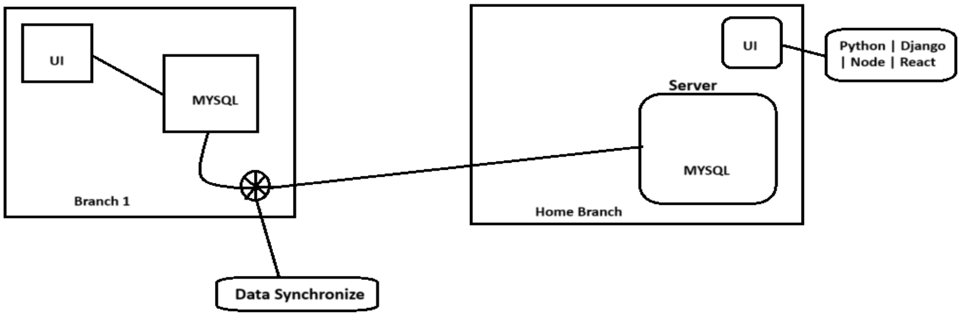
Process-2:

Platform Architect:

After the Business Analyst provides the functional requirements, the Platform Architect uses them to design the system architecture.







Branch's (Offline Branch's)

- **UI**: The user interface is hosted locally at the branch, allowing users to interact with the system even without internet connectivity.
- **MySQL**: The local database where all data entries and transactions are stored.
- **Status**: This branch operates **offline**, meaning it functions independently without an active internet connection.
- **Function**: Users input data via the UI, and the data is stored in the local MySQL database.

Data Synchronize Component

- **Role**: Acts as a middle layer that handles the **synchronization of data** from offline branches to the central home branch once an internet connection becomes available.
- **Mechanism**:
  - Monitors for internet availability.
  - When online, pulls data from the local MySQL.
  - Pushes/syncs the data to the **Home Branch Server**.
- **Use Case**: Ideal for remote or rural locations where continuous internet is not available.

Home Branch (Central Server)

- **Server**:

  - Hosts the **main MySQL database**, which acts as the central repository.
  - Receives and stores synchronized data from all branches.
- **UI**: This is the centralized user interface accessible via internet.
- **Tech Stack**: Powered by **Python, Django, Node.js, and React**, indicating:
  - **Python/Django** for backend APIs and server-side logic.
  - **Node.js** possibly for middleware, services, or real-time functions.
  - **React** for a dynamic and responsive frontend.

Data Syncretization:

Data synchronization is the process of ensuring that data in two or more locations remains consistent, accurate, and up-to-date. It involves the continuous or scheduled alignment of data between systems, devices, or databases, so that any change made in one location is reflected in the others.

Example:
If a user updates their contact information on a mobile app, data synchronization ensures that the same update appears on the web application and the backend database.

(Day -2)

Process-3:

After discussing the architecture, the manager will consult with team members and assign separate tasks to each team member.

Engineer team:

(Menu,Ordering,Billing,Log)

1. **Responsibilities:**

- Choose methodology: Agile (2-week sprints)
- Create roadmap:
  - Sprint 1: UI mockups
  - Sprint 2: Backend billing engine

    - Sprint 3: Payment integration
    - Sprint 4: Reporting & inventory sync
- Manage:
    - JIRA/Trello for tasks
    - Daily standups
    - Sprint planning & retrospectives
- Risk Management
- Timeline, Budget Tracking

Frontend Developers

- Build user-friendly POS interface
- Real-time order updates
- Responsive layout (tablet/kiosk support)

Backend Developers

- Create REST APIs
- Implement billing logic (menu price * qty + tax - discounts)
- Database schema: Tables: Users, Orders, Items, Transactions, Discounts, Taxes, Reports

Integration with:

- Inventory system
- Payment gateway
- Notification systems (SMS/Email for bills)

Process – 4:

Data Engineer

(Data Warehouse)

Responsibilities:

- Create ETL pipelines
- Clean & store transactional data in data warehouse

- Integrate with real-time systems (Kafka, Airflow optional)

Process – 5:

Data Analyst

Responsibilities:

- Build dashboards for:
    - Daily Sales
    - Peak Hour Analysis
    - Top-selling Items
    - Inventory Depletion Rate
- Tools: Power BI.

Basic Visual studio code

Python Basics:

1.What is python?

2.Complier and Interpreter

3.Clauses (If,Ifelse,else)

4.Loops (For,while)

5.Definition (def)

6.Class and Object

7.File Handling

8.Exception Handling.