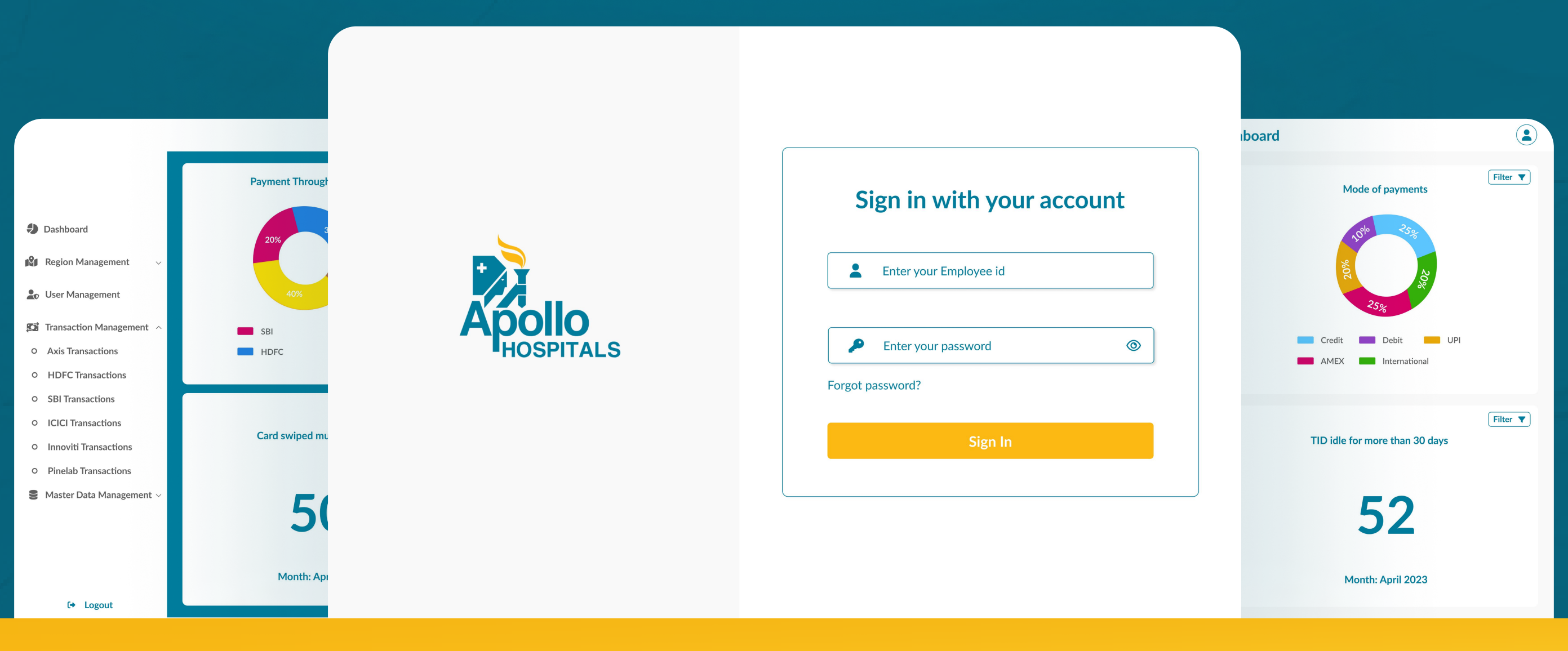
Introis technologies

2023

Case Study
MDR Tracking System for Apollo Hospital


Dashboard
Region Management
User Management
Transaction Management
Axis Transactions
HDFC Transactions
SBI Transactions
ICICI Transactions
Innoviti Transactions
Pinelab Transactions
Master Data Management
Logout
Payment Through Banks
SBI
HDFC
Card swiped multiple times
50
Month: April 2023
Apollo HOSPITALS
Sign in with your account
Enter your Employee id
Enter your password
Forgot password?
Sign In
Mode of payments
Filter
Credit
Debit
UPI
AMEX
International
TID idle for more than 30 days
52
Month: April 2023

“About Client.

Apollo Hospitals, a leader in healthcare, required a reliable, automated solution to track Merchant Discount Rates (MDR) for card-based payments across various banks. The goal was to centralize and streamline the monitoring of MDR rates, reduce financial discrepancies, and ensure regional data access control, catering to the specific needs of its officials and administrators.

What are our Goals.

- Centralize MDR rate tracking for accurate monitoring across multiple banks.
- Highlight high MDR transactions to reduce discrepancies and optimize costs.
- Enable role-based and regional access to ensure secure data access.

The project challenges included handling large daily data inflows, setting up secure access controls, and building an intuitive user interface.

DEVLOPEMENT PHASE.

1. REQUIRMENTS

The development team conducted a thorough analysis to understand Apollo Hospitals' needs. This phase involved

Stakeholder Meetings

Engaging with Apollo’s finance and IT departments to document requirements for MDR tracking, data access, and security protocols

Data Flow Mapping

Outlining the data flow between bank sources (SBI, Axis, ICICI, HDFC), Pinelab/Innoviti TIDs, and Apollo’s internal systems.

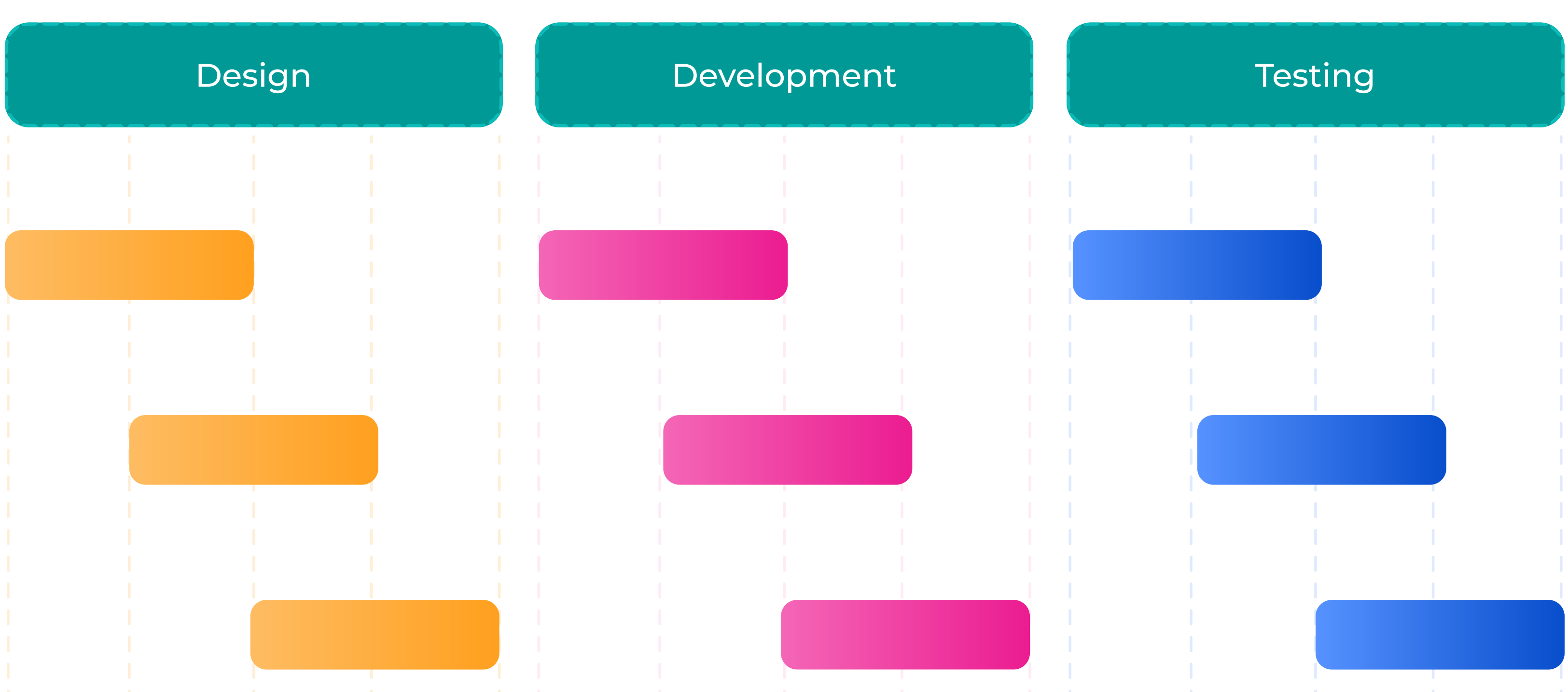
Requirement Documentation

Finalizing key functional requirements, such as high MDR flagging, TID idle time monitoring, and region-based access controls.

DEVLOPEMENT PHASE.

2. PLANNING

With a clear understanding of the requirements, the development team devised a plan that outlined


Design
Development
Testing

Project Milestones

Defining timelines for design, development, testing, and deployment phases.

Technology Stack Selection

Choosing technologies that supported scalability, such as secure database management for handling sensitive transaction data.

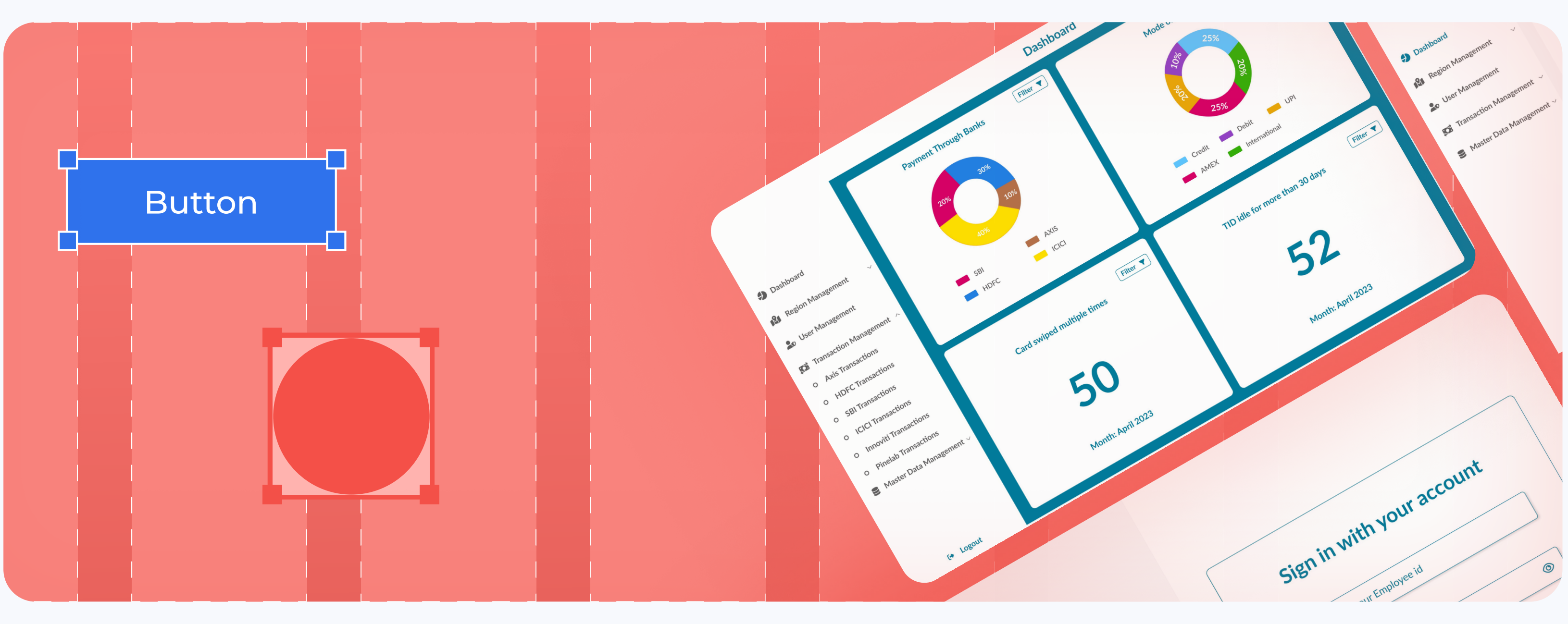
Task Breakdown

Assigning tasks to developers for backend, frontend, and database development.

DEVLOPEMENT PHASE.

3. DESIGN

The design phase focused on creating an intuitive, secure, and functional system. Key deliverables included


Button
Dashboard
Payment Through Banks
Card swiped multiple times
50
Month: April 2023
Mode of payments
TID idle for more than 30 days
52
Month: April 2023
Sign in with your account
Enter your Employee id

UI/UX Design

Developing a user-friendly dashboard with clear data displays for MDR rate, transaction type, and filtering options.

Database Design

Structuring a secure database to store and retrieve MDR data by region, bank, and card type.

System Architecture

Creating a modular architecture to support automatic email data imports, error flagging, and manual data uploads.

DEVLOPEMENT PHASE.

4. DEVELOPMENT

In this phase, developers implemented the features as per design specifications

TECHOLOGY STACKS ARE USED

ReactJS | Node.js | MySQL | GitHub | Express.js | Nginx

Backend Development

Coding the backend to automate data imports from banks, store transactional data, and perform MDR comparisons between Pinelab/Innoviti TIDs and bank statements.

Frontend Development

Building a dynamic dashboard that allowed users to view MDR details, filter data, and export reports.

Access Control Setup

Implementing role-based access, enabling officials to view regional data and granting full access to admins for broader control.

DEVLOPEMENT PHASE.

5. TESTING

Testing was conducted to ensure the system met Apollo’s requirements in terms of functionality, accuracy, and security

Unit testing | Integration Testing | User Acceptance Testing (UAT)

Unit Testing

Testing individual modules, such as data imports, filtering, and exporting, to confirm accuracy.

Integration Testing

Ensuring that the backend, database, and frontend worked seamlessly together.

User Acceptance Testing (UAT)

Engaging Apollo’s representatives in testing to validate that the system met expectations for ease of use and accuracy.

DEVLOPEMENT PHASE.

6. DEPLOYMENT AND SUPPORT

The team successfully deployed the MDR Tracking System within Apollo’s intranet. Ongoing support was established to address any issues and to facilitate future updates based on user feedback.

TECHNICAL STACKS

Frontend

ReactJS: Used for creating an interactive and responsive dashboard interface for MDR tracking.

Backend

Node.js: Serves as the primary backend framework, handling REST API requests for data management and access control.
Express.js (Node API): Used for building and structuring the API endpoints.
Nginx: Configured as a web server and proxy server to handle HTTPS requests securely.
pm2: For Node.js process management, ensuring reliability and load balancing.

Database

MySQL: A relational database for storing MDR data, user roles, access permissions, and transaction details across regions and banks

Version Control and Deployment

GitHub: Version control for codebase management, with pull requests managing updates to both frontend and backend.
DNS/Nameserver and SSL Certificates: For secure domain management and data transmission.

Results

- Real-time tracking of MDR rates across banks, with high-rate alerts for quick review.
- Role-based access ensured data security and compliance
- Intuitive dashboard for easy data filtering, analysis, and export, saving reporting time.
- User-friendly interface led to widespread use, improving financial transparency and efficiency across regional offices.

THANK YOU