Project Design Phase-II
Technology Stack (Architecture & Stack)

Date	16 April 2025
Team ID	SWTID1743315733
Project Name	Personal Finance Tracker
Maximum Marks	4 Marks

Technical Architecture:

The Personal Finance Tracker employs a standard 3-tier web application architecture:

1. **Client-Side (Presentation Tier): A React-based Single Page Application (SPA) running in the user's web browser. It handles user interactions, renders the UI, and communicates with the backend API.**
2. **Server-Side (Application/Logic Tier): A Node.js application using the Express.js framework. It acts as a RESTful API, handling business logic, processing requests from the frontend, managing user authentication, and interacting with the database.**
3. **Data Tier: A MongoDB database (likely hosted on MongoDB Atlas for cloud deployment or running locally for development) persists all application data, including user credentials, income/expense transactions, and budget information. Mongoose ODM is used for object data modeling.**

Architectural Diagram (Conceptual):

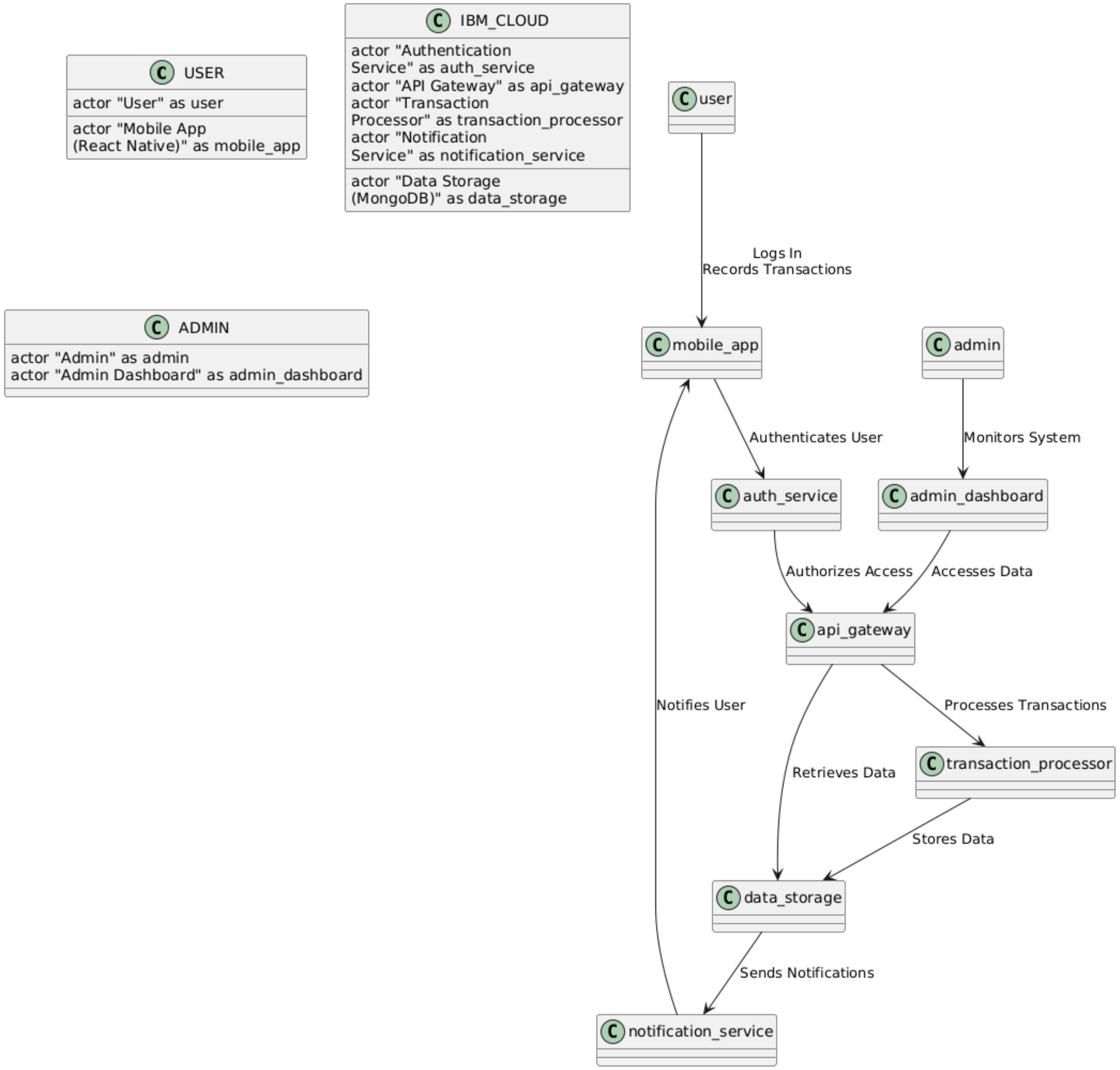

Table-1 : Components & Technologies:

S.No	Component	Description	Technology
1.	User Interface	Web UI allowing users to interact with the application.	React.js, HTML5, CSS3, JavaScript (ES6+)
2.	Application Logic-1	Backend REST API server handling requests, business logic.	Node.js, Express.js
3.	Application Logic-2	User authentication and authorization logic.	JWT (JSON Web Tokens), bcrypt (for password hashing)
4.	Application Logic-3	Data modeling and interaction with the database.	Mongoose ODM
5.	Database	NoSQL database storing user, income, expense, and budget data.	MongoDB
6.	Cloud Database	Cloud-hosted database service (Optional/Deployment Target)	MongoDB Atlas (or specify if using local instance)
7.	File Storage	Not applicable for core features.	N/A
8.	External API-1	Not applicable for core features.	N/A
9.	External API-2	Not applicable for core features.	N/A
10.	Machine Learning Model	Not applicable.	N/A
11.	Infrastructure	Application deployment environment.	Local Development Server (Node), Potential: Heroku, Vercel, AWS EC2, Docker

Table-2: Application Characteristics:

S.No	Characteristics	Description	Technology / Approach
1.	Open-Source Frameworks	Utilizes established open-source libraries and frameworks for development.	React.js, Node.js, Express.js, Mongoose, Axios, Chart.js
2.	Security Implementations	Implements standard security practices for web applications.	Password Hashing (bcrypt), JWT for session management, HTTPS (requires deployment configuration), Input validation, Environment variables for sensitive keys, CORS configuration.
3.	Scalable Architecture	Follows a 3-tier architecture separating concerns. The backend API is stateless, allowing horizontal scaling (running multiple instances behind a load balancer).	3-Tier Architecture, Stateless API design. MongoDB Atlas provides built-in scaling options.
4.	Availability	Dependent on deployment strategy. Cloud platforms offer high availability features.	Deployment on platforms like Heroku, Vercel, AWS with potential use of load balancers and multiple server instances. Database replication (feature of MongoDB Atlas).
5.	Performance	Design considerations include efficient data retrieval and rendering.	Database indexing (e.g., on userId, date), Pagination for transaction lists, Efficient state management in React (Context API), Asynchronous operations in Node.js.