



Low Level Design Document

Introduction

This Low Level Design (LLD) document outlines the architecture and core implementation details for **FinSight - Personal Finance Dashboard**. FinSight is a modern, interactive dashboard for visualizing spending, budgets, and financial goals, built with React, JavaScript, HTML, CSS, and Tailwind.

1. System Components

Component	Description / Responsibility
App	Root component, routing, context providers
Dashboard	Main layout, aggregates widgets
SpendingChart	Visualizes spending data (charts)
BudgetSummary	Displays budget status and breakdown
GoalsTracker	Shows progress towards financial goals
DataService	Fetches and manages financial data
ErrorBoundary	Catches and displays UI errors

2. Class/Interface Overview

Component/Class	Key Props/State/Methods	Relationships
App	-	Parent of all components
Dashboard	userData, onRefresh	Contains SpendingChart, BudgetSummary, GoalsTracker
SpendingChart	data	Receives data from Dashboard
BudgetSummary	data	Receives data from Dashboard
GoalsTracker	goals	Receives data from Dashboard
DataService	fetchUserData(), updateGoal()	Used by Dashboard

Key Methods Example:

```
// DataService.js
async function fetchUserData(userId): Promise<UserData>
async function updateGoal(goalId, newValue): Promise<Goal>
```



3. Data Structure Overview

Model	Fields
UserData	id, name, accounts: Account[], budgets: Budget[], goals: Goal[]
Account	id, type, balance, transactions: Transaction[]
Transaction	id, date, amount, category, description
Budget	id, category, limit, spent
Goal	id, name, targetAmount, currentAmount, dueDate

4. Algorithms / Logic

Data Fetch & Render Flow:

```
On Dashboard mount:
  Call DataService.fetchUserData(userId)
  If success:
    Set userData state
    Pass data to child components
  If error:
    Show error via ErrorBoundary
```

Spending Aggregation:

```
For each account in userData.accounts:
  For each transaction:
    Group by category, sum amounts
```

5. Error Handling

Scenario	Handling Approach
Data fetch failure	Show error message in UI
Invalid user input (e.g. goal)	Validate, show inline error
Chart rendering error	Fallback to placeholder, log error
Network timeout	Retry fetch, show notification
Unhandled JS error	ErrorBoundary displays fallback UI

End of Document