

Project design phase

Solution Architecture

Date	30/10/2025
Time id	NM2025TMID05890
Project name	Calculating Family Expenses using Service Now
Maximum mark	2 marks

1. Goals

The primary goal of the Family Expense Calculator System is to create a smart, automated, and user-friendly platform that helps households monitor, categorize, and analyze their financial activities.

Specific goals include:

- 1. Expense Tracking:** Record and categorize all family expenses automatically or manually.
- 2. Budget Management:** Set monthly or annual budgets and receive alerts when spending exceeds thresholds.
- 3. Data Visualization:** Provide visual reports (charts, tables, and summaries) for better financial insight.
- 4. Smart Recommendations:** Suggest saving opportunities based on previous expense patterns.
- 5. Multi-User Access:** Enable different family members to view, add, or monitor expenses collaboratively.

2. Key Components

Component	Description
User Interface (UI)	Interactive dashboard allowing users to input expenses, view reports, and set budgets.
Expense Database	Centralized data store for expense records, categories, and user profiles.
Budget Engine	Core logic to calculate total spending, compare it to budgets, and generate alerts.
Analytics Module	Processes transaction data and provides trend analysis, graphs, and summaries.
Authentication System	Provides secure login for each user/family member.
Notification System	Sends alerts for overspending, due bills, or savings tips.
Data Backup & Security	Ensures data integrity, encryption, and regular backup to prevent loss.

3. Development Phases

Phase	Activities	Outcome
1. Requirement Analysis	Identify family needs, types of expenses, data inputs, and reporting preferences.	Functional and non-functional requirements documented.
2. System Design	Create database schema, user interface layouts, and architecture diagrams.	Detailed design documentation ready for development.
3. Development	Build frontend (web/app UI), backend logic, and database integration.	Functional modules implemented.
4. Integration & Testing	Combine all modules; perform unit, integration, and user testing.	Stable and verified expense management system.
5. Deployment	Deploy application on cloud or local server for user access.	Live, fully operational system.
6. Monitoring & Maintenance	Monitor usage, collect feedback, fix bugs, and add new features.	Continuous improvement and sustainability.

4. Solution Architecture Design

1. Architecture Overview

The system follows a three-tier architecture:

- Presentation Layer (Frontend):
 - Developed using HTML, CSS, JavaScript (or React).
 - Displays dashboards, charts, and forms for expense input and visualization.
- Application Layer (Backend):
 - Developed using Flask or Node.js.
 - Handles expense calculations, budget validation, and analytics logic.
- Data Layer (Database):
 - Uses MySQL or SQLite for storing user data, expense records, and budget limits.

2. Workflow

1. User logs in and enters expenses (manual input or upload).
2. System categorizes and stores data in the database.
3. Budget engine calculates totals and compares with predefined budgets.
4. Analytics module generates visual charts and summaries.
5. Notifications alert users when expenses exceed limits or unusual spending is detected.
6. Data is backed up periodically for safety.

3. Tools & Technologies

- Frontend: HTML5, CSS3, JavaScript (React optional)
- Backend: Python (Flask) or Node.js
- Database: MySQL / SQLite
- Analytics & Charts: Chart.js or Power BI integration
- Hosting: Render / Vercel / AWS
- Security: JWT Authentication, SSL, Data Encryption

Example :

Solution Architecture

