SplitSmart

Sprint 2

Team:
Mohammad Areeb
Prabhkrit Singh
Samarjeet Singh

Table of Contents

CRC Cards:

User	
Responsibilities: Store user email, password hash, name, join date; Validate email format; Manage authentication state	Collaborators: AuthController, Group, UserRepository

AuthController	
Responsibilities: Handle registration requests; Handle login requests; Generate JWT tokens; Validate credentials	Collaborators: User, UserRepository, PasswordUtil, JWTUtil

UserRepository	
Responsibilities: Save user to database; Retrieve user by email; Retrieve user by ID; Check email uniqueness; Fetch user profile	Collaborators: User, MongoDB

PasswordUtil	
Responsibilities: Hash passwords with bcrypt; Compare password with hash; Validate password strength	Collaborators: User

JWTUtil	
Responsibilities: Generate JWT token; Sign token with secret; Verify token; Extract user ID from token	Collaborators: AuthController

Group	
Responsibilities: Store group name, description, members; Track group admin; Store creation date; Manage member list	Collaborators: User, GroupController, GroupRepository

GroupController	
Responsibilities: Handle create group requests; Handle list groups requests; Validate group data; Set admin on creation	Collaborators: Group, GroupRepository, User

GroupRepository	
Responsibilities: Save group to database; Retrieve groups by user; Retrieve group by ID; Add member to group; Get member count	Collaborators: Group, MongoDB

Expense	
Responsibilities: Store expense amount, description, date. Track who paid (payer). Store category. Link to group. Maintain expense splits array. Calculate if expense is fully settled. Get total splits amount.	Collaborators: User, Group, ExpenseController, ExpenseRepository

ExpenseSplit	
Responsibilities: Store individual split amount per user. Track payment status. Link expense to specific user. Calculate individual owed amounts.	Collaborators: Expense, User

ExpenseController	
Responsibilities: Handle create expense requests. Validate split amounts equal total. Handle list expenses by group. Handle update/delete expense. Calculate group balances. Mark splits as paid.	Collaborators: Expense, ExpenseRepository, Group

ExpenseRepository	
Responsibilities: Save expense to database. Retrieve expenses by group. Retrieve expense by ID. Update expense details. Delete expense. Populate payer and split user details. Query expenses with filters.	Collaborators: Expense, MongoDB

BalanceCalculator	
Responsibilities: Calculate net balance per user in group. Track what each user paid. Track what each user owes. Generate balance summary. Determine who owes whom. Aggregate all group expenses.	Collaborators: ExpenseRepository, Group

AuthMiddleware	
Responsibilities: Check JWT token validity; Extract user from token; Protect authenticated routes; Return 401 on invalid token	Collaborators: JWTUtil, User

RegistrationForm (Frontend)	
Responsibilities: Render email, password, confirm password fields; Validate form inputs client-side; Provide error feedback; Submit to API	Collaborators: AuthService, User

LoginForm (Frontend)	
Responsibilities: Render email and password fields; Validate inputs; Submit credentials to API; Store token on success	Collaborators: AuthService, User, TokenStorage

GroupForm (Frontend)	
Responsibilities: Call create group API; Call list groups API; Manage group state; Filter groups by user	Collaborators: GroupForm, GroupList, API

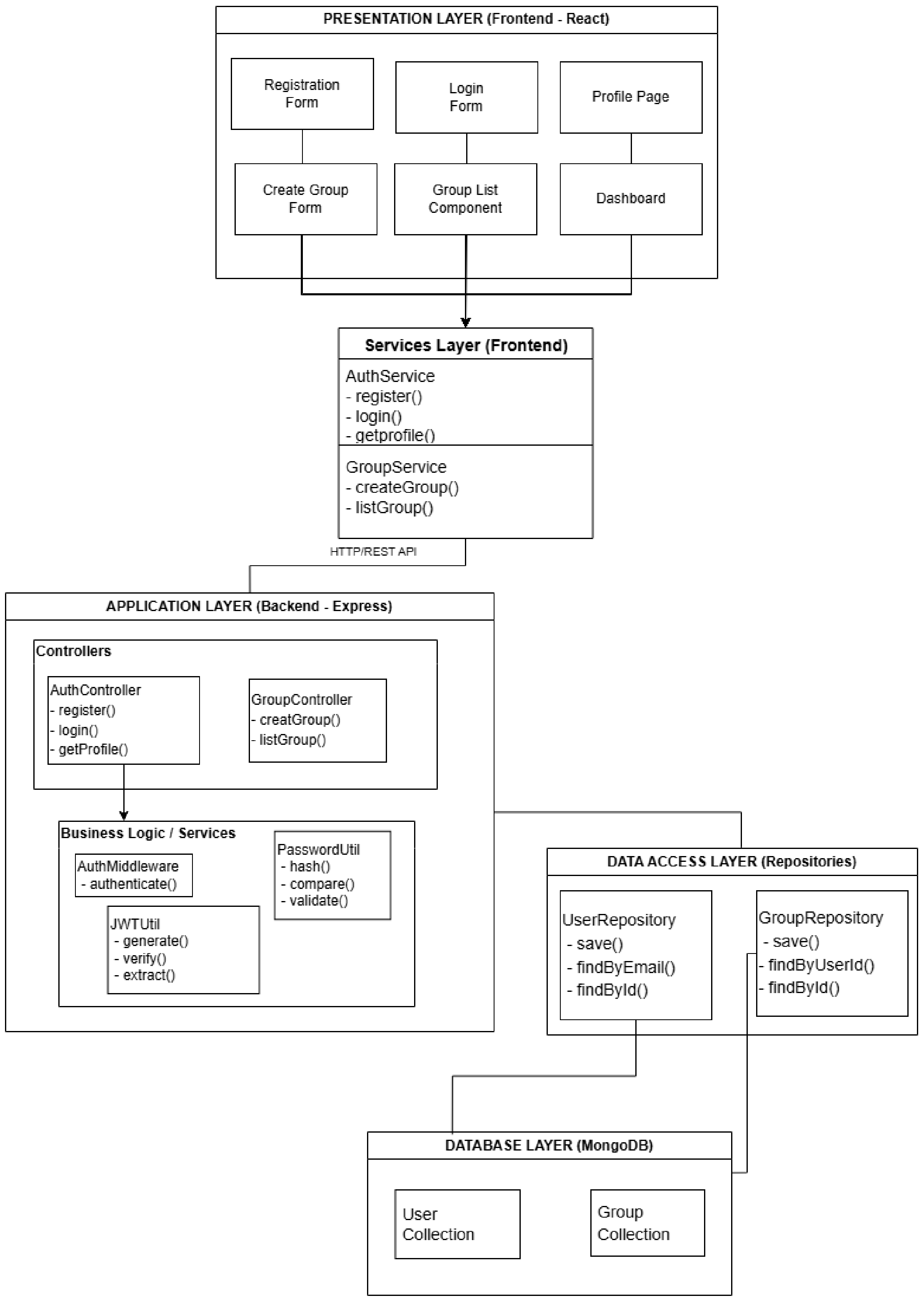
ExpenseForm (Frontend)	
Responsibilities: Render expense creation form. Input amount, description, category. Configure splits between members. Validate split amounts. Submit expense to API. Display validation errors.	Collaborators: ExpenseService, API

ExpenseList (Frontend)	
Responsibilities: Display expenses by group. Show expense details (payer, amount, splits). Filter/sort expenses. Edit/delete actions with authorization.	Collaborators: ExpenseService, API

BalanceDashboard (Frontend)	
Responsibilities: Display balance summary for group. Show who owes whom. Display paid vs owed amounts. Color-code positive/negative balances.	Collaborators: BalanceService, API

TokenStorage (Frontend)	
Responsibilities: Store JWT in localStorage/sessionStorage; Retrieve token; Clear token on logout; Check token expiry	Collaborators: AuthService

Software Architecture Diagram:



System Interaction with Environment

Dependencies:
- Node.js v18+, Express 4.18, MongoDB (Mongoose 8.20)
- bcryptjs 2.4 (password hashing), jsonwebtoken 9.0 (JWT auth)
- React 18, Vite (frontend build tool)
- cors, dotenv

External Systems:
- MongoDB Atlas (cloud database)
- JWT signing library
- bcryptjs for password hashing
- CORS for cross-origin requests
- Environment variables for config

Assumptions:
- Users have internet connection
- MongoDB connection always available
- Server runs on localhost:5000, Frontend on localhost:3000
- Modern browsers support ES6+

System Decomposition

Component Breakdown:

Component	Responsibility	Dependencies
AuthController	Handle registration & login requests	UserRepository, PasswordUtil, JWTUtil
GroupController	Handle group creation & listing	GroupRepository, User
UserRepository	CRUD operations for User	MongoDB
GroupRepository	CRUD operations for Group	MongoDB
AuthMiddleware	Protect routes, validate JWT	JWTUtil
PasswordUtil	Secure password handling	bcryptjs
JWTUtil	Token management	jsonwebtoken
Registration/Login Forms	User input capture	AuthService
Groups Dashboard	Display user's groups	GroupService
AuthService (Frontend)	API communication for auth	HTTP Client
GroupService (Frontend)	API communication for groups	HTTP Client
ExpenseController	Expense CRUD, splits, balance calculation	Expense, ExpenseRepository, Group
ExpenseRepository	Expense data access via Mongoose	Expense, MongoDB
BalanceCalculator	Calculate group balances	ExpenseRepository
ExpenseForm	Expense creation UI	ExpenseService
ExpenseList	Display group expenses	ExpenseService
BalanceDashboard	Display balances	BalanceService
ExpenseService (Frontend)	API calls for expenses	HTTP Client

Error Handling Strategy

Backend Errors:

- 400 Bad Request: Missing fields, invalid format, split amounts mismatch
- 401 Unauthorized: Invalid credentials, expired/missing JWT token
- 403 Forbidden: Not group member, not admin, not expense payer
- 404 Not Found: User/Group/Expense not found
- 409 Conflict: Email already exists, duplicate member
- 500 Internal Server Error: Database errors, unexpected failures

Frontend Errors:

- **Network Errors:** Display "Connection failed, please try again"
- **Invalid Input:** Show field-level error messages
- **API Errors:** Display user-friendly error toast messages
- **Auth Errors:** Redirect to login on 401, show "Session expired"
- **Form Validation:** Real-time validation with error icons

Validation Rules:

- **User:** Email (valid, unique), Password (min 8 chars), Full Name (required)
- **Group:** Name (3-100 chars), Members (valid emails)
- **Expense:** Amount (positive, max 2 decimals), Description (3-200 chars), Splits (sum = total)