Microservices Architecture & Feature Specification (Split/Deng System)

> **Note on terminology:** The Persian word **"دنگ" (Deng)** in this document refers to a **shared expense / split** entry.
> We keep the domain term **Deng** for clarity.

High-Level Overview
- **Microservices:** Three core services
  1) **SSO/User Service** (authentication, profile, OTP)
  2) **Deng (Split) Service** (groups, expenses, shares, settlements)
  3) **Financial Management Service** (personal P&L, categories, SMS import)
- **Inter-service messaging:** **Kafka** for asynchronous events between services.
- **OTP transport:** **RabbitMQ** for OTP message queueing and delivery.
- **AuthN/AuthZ:** **JWT** (access/refresh) for API security and user sessions.
- **Export:** PDF/CSV export for Deng and Financial reports (with options).

SSO/User Service
Authentication & Authorization
- **JWT**-based tokens for users (access & refresh).
- **OTP**-based phone verification **only** for:
  - **Sign up**
  - **Forgot password** (password reset)
- **RabbitMQ** used to queue and dispatch OTP messages (e.g., SMS provider).

User Profile
- Fields:
  - **Name** (optional)
  - **Phone number** (required; primary identifier)
  - **Avatar** (optional; if empty, **auto-pick** a random avatar from default images)
  - **Bank card number** (optional)
  - **Cardholder name** (optional)
- SSO issues tokens; downstream services verify JWT.

Activity Logging
- **Audit logs** for all critical auth/profile events across services.

Deng (Split) Service
Core Concepts
- **Group**: a container for shared expenses (Dengs).
  - **Group name** and **group image** (both optional).
  - **Rounding toggle** at group level (round up/down for better readability).
  - **Currency support**: IRR (Rial), Toman (configurable/optional per Deng).
  - **Single-person groups** allowed (for hypothetical or personal-only tracking).
  - **Subgroups** supported; ability to **split between subgroups**.
- **Deng (Expense Entry)** properties:
  - **Amount**
  - **Who paid**
  - **What for (description)**
  - **Receipt image** (optional, can be empty)
  - **Date**: auto-generated on creation; **editable** if needed
  - **Per-user share**:
    - **Automatic split**
    - **Manual split** by **percentage** or **absolute amounts**
      (sum **must equal** the Deng total)

Group Membership & Permissions
- **Create group**: any user can create a group.
- **Add members**:
  - By **phone number** or
  - From **users in other groups** the creator belongs to
- **Edit/Delete permissions**:
  - A user may **edit/delete only the Dengs they created**.
  - **Admin (group creator)** can **edit all Dengs** in the group.
  - Non-admin users can **view** other users' Dengs but **cannot edit** them.

Debt Simplification & Settlement Optimization
- For debts like **A owes B**, **B owes C**, system **nets** and **removes intermediaries** where possible.
- **Objective:** minimize the **number of payments** so users make as few transfers as possible.
- Provide a **graph-based settlement algorithm** that:
  - Computes **net balances** per user within the group.
  - Produces a **minimal (or near-minimal)** set of transfers to settle.
  - Avoids requiring one person to pay multiple people **unless necessary**.
- **Manual settlements**:
  - Users record manual settlements (amount, date, optional note/receipt).
  - In the future, support **payment gateway** for **automatic settlement** with **tracking number** and **timestamp**.

Deng Lists, Details & Summaries
- **Group Deng list view** shows:
  - **Total amount** of that group
  - For each Deng: **payer**, **date**, **title/description**
  - For the viewing user: **you owe / you are owed / settled** amount **for that Deng**
  - A button for **"You are not a debtor"** (ack/confirm if needed)
- **Deng details view** shows:
  - **Full breakdown of shares** per user
  - **Name, date, receipt (if any)**
  - If the viewing user is **not a participant** in the Deng, show **"You are not a participant in this Deng"**, but still display it in the group list.
- **Per-user analytics (in-group)** for the viewing user:
  - **Monthly**, **yearly**, **weekly** expense history
  - **Total expenses contributed** (payer perspective) in the group
  - **Total of all personal spending** the user has logged in that group
- **Group-level toggles**:
  - **Rounding up/down** of displayed Deng amounts (visual display only; raw values preserved).

UX Interactions
- **Tap/Click on a member**: show **card number** and **cardholder name** (if available).
- **Group details**: show **member list**, **sum of all expenses this month**, and **all-time total**.
- **Expense aggregation**:
  - Show **monthly totals per Deng type/label** and **overall totals**.
- **In-app notifications** per group for **every action** (create/update/delete Deng, settlements, member changes, etc.).

Exports
- **PDF/CSV** export:
  - Choose **which group(s)** and **which Deng(s)** to include.
  - Include **summaries** and/or **detailed line items**.
  - Include **attachments list** (receipt references).

Financial Management Service (Separate Service; SSO-enabled)
Functionality
- **Transactions**: log **expenses** and **income** with:
  - **Category**, **amount**, **date**, **description**
- **Home dashboard**:
  - Show **sum of expenses** and **sum of income** for the **current month**
- **Aggregations/Reports**:
  - Totals by **category** and **time period** (monthly, yearly)
- **Input methods**:
  - **Manual entry**
  - **Automatic import from SMS** (parse incoming messages to create transactions)
- **Activity logs**: audit all actions (create, edit, delete, import).
- **Financial profile**: summarized insights of the user's overall **spending/income behavior** across the system.

Exports
- **PDF/CSV** reports:
  - Select **groups**, **Dengs**, and **report scopes** (month, year, custom range)
  - Include **filters** by category, payer, participant, and presence of attachments

---

Messaging & Events
- **Kafka** topics for service-to-service events:
  - User created/updated, group created/updated, Deng created/updated/deleted, settlement recorded, notification events, export requested, etc.
- **RabbitMQ** for OTP workflow:
  - Enqueue OTP jobs; workers deliver via SMS provider; track TTL, rate limits, and retries.

Security & Compliance
- **JWT** validation at each service boundary.
- **Role-based access control** within groups (admin vs. member).
- **Data protection** for receipts and PII (avatar, phone, bank card).
- **Comprehensive audit logging** across all services.
- **Rate limiting** on OTP and auth endpoints.

Data Model (Indicative)
- **User**: id, phone, name?, avatar?, card_number?, cardholder_name?
- **Group**: id, name?, image?, admin_user_id, rounding_mode?, currency?
- **GroupMember**: group_id, user_id, role, joined_at
- **Deng**: id, group_id, payer_user_id, amount, currency, title, note?, date, receipt_url?
- **DengShare**: deng_id, user_id, share_amount (or share_percent), is_manual
- **Settlement**: id, group_id, from_user_id, to_user_id, amount, date, method, note?, receipt_url?
- **Notification**: id, group_id?, user_id, type, payload, created_at, read_at?
- **Transaction (Finance)**: id, user_id, type(expense|income), category, amount, date, description, source(manual|sms), raw_sms?

APIs (Sketch)
- **SSO**: /auth/signup (OTP), /auth/login, /auth/refresh, /auth/forgot (OTP), /users/me, /users/{id}
- **Groups**: /groups (CRUD), /groups/{id}/members (add/remove), /groups/{id}/summary

- **Dengs**: /groups/{id}/dengs (list/create), /dengs/{id} (get/update/delete), /dengs/{id}/shares
- **Settlements**: /groups/{id}/settlements, /settlements/{id}
- **Exports**: /exports/pdf, /exports/csv with filters (groups, dengs, date range)
- **Finance**: /finance/transactions (CRUD), /finance/summary, /finance/reports

---

Future Work
- **Online payment gateway** integration for automatic settlements; store **tracking number** and **timestamp**.
- **Advanced optimization** for debt settlements (approx-minimum transfer count).
- **More currencies** and **multi-currency groups** with FX.
- **External notifications** (email/push) in addition to in-app.