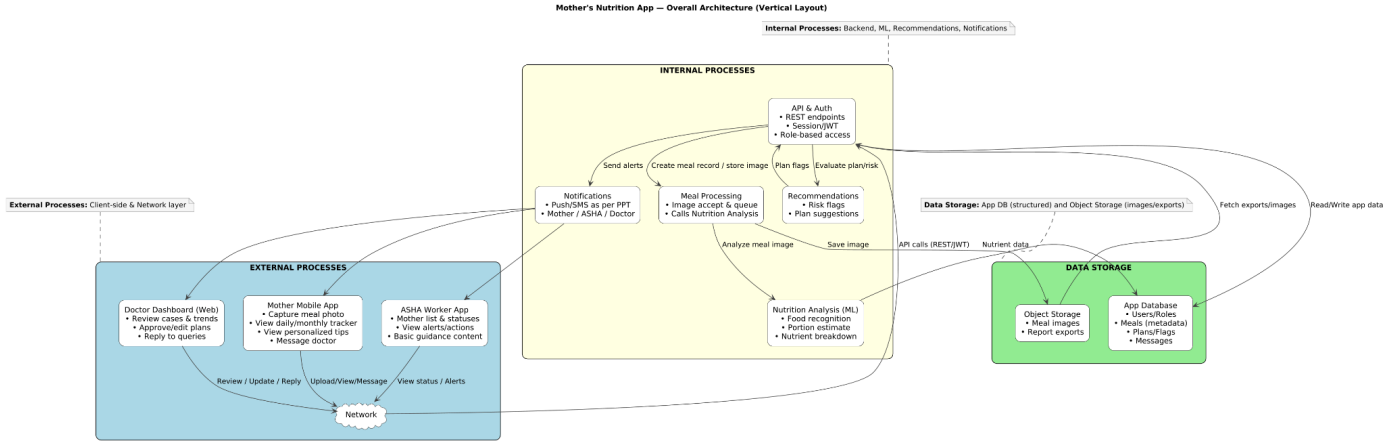
Design for social innovation
Project: Mother's nutrition app
Team: Saja Abracadabra
Final Report

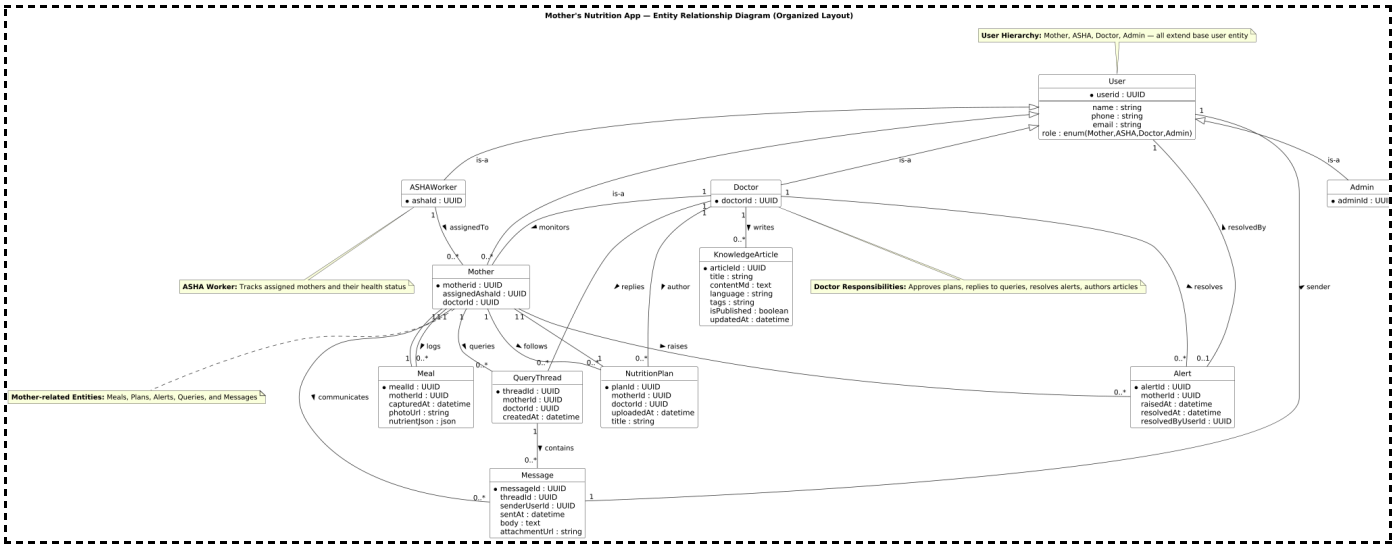
Architecture

Following is the architecture diagram to further describe the application: (link for further clarity)



ER Diagram

Following is the er diagram describing our database and application.(link for further clarity)



User flows

1. MOTHER FLOWS

1.1 Registration & Login

Go to /signup → fill details → select Mother → submit.
System creates account, assigns doctor + ASHA worker, logs user in → redirects to /mother.
Login: enter email, password, select Mother, authenticate → /mother.

1.2 Meal Upload & Tracking

On /mother, choose Upload Meal → select meal type, date, image.
System saves image → runs nutrient extraction (dummy) → compares with plan → creates alert if deficit.
Dashboard updates: meal history, nutrient summary, alerts.

1.3 View Nutrition Plan

Dashboard shows today's plan: required, consumed, remaining nutrients.
Color-coded: Green = complete, Red = deficit.
Doctor details visible (name, email).

1.4 Query Management

Click ? → submit query (subject, message).
System stores it under mother, marks as pending.
View history → see doctor responses → status updates to answered/resolved.

1.5 Alerts

Dashboard shows nutrient alerts if intake <60%.
Shows date, deficient nutrients, values and percentages.
Clears once fixed, resolved, or triaged.

2. DOCTOR FLOWS

2.1 Registration & Login

Signup → select Doctor → submit → redirected to /doctor.
Login with role Doctor → /doctor.

2.2 Patient Management

Dashboard lists all assigned mothers with alerts, profile buttons.
View profile to see details, meal history, alerts, queries, nutrition plan.

2.3 Create / Update Nutrition Plan

In patient profile → fill plan manually or load preset (trimester/lactation).
Save → plan becomes active, previous deactivated.

2.4 Query Response

View patient queries → open a pending one → type reply → send.
Status changes to answered, mother sees response.

2.5 Alerts Monitoring

Dashboard shows alert count per mother.
Open patient → view alert details → update plan or respond as needed.

3. ASHA WORKER FLOWS

3.1 Registration & Login

Signup → select ASHA → submit → /asha.
Login with role ASHA → /asha.

3.2 View Assigned Mothers

Dashboard lists assigned mothers, alerts, last visit date, and details button.

3.3 Record Field Visit

Select mother → "Record Visit" → fill date, type, observations, optional photo → submit.
Visit stored and visible in history.

3.4 Alert Triage

Dashboard shows all active alerts.
Open alert → choose action (monitor/escalate) → add notes → submit.
System marks alert as triaged and optionally escalates.

3.5 View Mother Details

View profile → shows demographics, nutrition plan, weekly meals, alerts, visit history.

4. COMMON FLOWS

4.1 Logout

Click Logout → session cleared → redirected to /login.

4.2 Session Management

After login: session[user_id], session[role] stored.
All routes verify session and role:

Mothers → /mother
Doctors → /doctor
ASHAs → /asha
Unauthorized access → 403.

Session expires on logout, timeout, or browser close**.**

APIs

1)Authentication and routing

Method	Route	Description	Auth	Body / Params	Returns
GET	/	Redirect to dashboard based on session role	Session	—	Redirect
GET	/login	Show login page	None	—	HTML
POST	/login	Login user	None	email, password, role	Redirect + session
GET	/signup	Show signup page	None	—	HTML
POST	/signup	Register user	None	User details (form)	Redirect + session
GET	/logout	Logout	Session	—	Redirect

2)Dashboard pages

Method	Route	Description	Auth	Body / Params
GET	/mother	Mother dashboard	mother	—
GET	/doctor	Doctor dashboard	doctor	—
GET	/asha	ASHA dashboard	asha	—
GET	/doctor/patient/<mother_id>	View mother info	doctor	—
POST	/doctor/patient/<mother_id>	Update nutrition plan	doctor	Nutrition plan form

3) Meals and nutrition

Method	Route	Description	Auth	Body / Params	Returns
POST	/api/meals/upload	Upload meal, analyze nutrients	Session	mealType, mealDate, image, motherId?	Meal + OCR + deficits + summary
GET	/api/meals/<meal_id>	Get single meal	Session	URL param	Meal object
GET	/api/meals/mother/<mother_id>	All meals for mother	Session	URL param	Meal list
POST	/api/nutrition-plans	Create/update plan	doctor	JSON plan	Plan object
GET	/api/nutrients/remaining/<mother_id>	Today's nutrient summary	Session	URL param	Required, consumed, remaining

4)Alerts API

Method	Route	Description	Auth	Params	Returns
GET	/api/alerts/<mother_id>	Get active alerts	Session	Mother ID	Alerts list

5) Queries API

Method	Route	Description	Auth	Body	Returns
POST	/api/queries/create	Mother creates query	mother	subject, message, category	Query object
GET	/api/queries/my-queries	List mother's queries	mother	status?, limit?	Query list
GET	/api/queries/all	All queries	doctor	Filters (status, category, limit)	Query list
POST	/api/queries/<query_id>/reply	Reply to query	doctor	message, updateStatus	Query reply

PUT	/api/queries/<query_id>/update-status	Update status/priority	doctor	status, priority	Query status
PUT	/api/queries/<query_id>/assign	Reassign query	doctor	doctorId	-
GET	/api/queries/<query_id>	Get query	Session	Query id	Query
GET	/api/queries	All queries	doctor		Query list
GET	/api/queries/mother/<mother_id>	Queries for mother	Session		Query List
POST	/query/<query_id>/respond	Reply to query (old schema)	doctor	Response	

6)Doctor APIs

Method	Route	Description	Auth	Returns
GET	/api/mothers/assigned/<doctor_id>	Mothers assigned to doctor	doctor	List
GET	/api/doctor/<doctor_id>	Doctor details	Session	Profile

7)Asha Worker API

Method	Route	Description	Auth	Body / Params	Returns
GET	/api/asha/assignments	ASHA assignments	asha	—	List
GET	/api/asha/<asha_id>	ASHA details	Session	_	Profile
GET	/api/asha/mothers/<asha_id>	Mothers under ASHA	asha	—	List

POST	/api/asha/visits	Create visit record	asha	motherId, date, type, observations, photo?	Visit object
GET	/api/asha/alerts	Alerts for ASHA mothers	asha	—	Alerts
POST	/api/asha/alerts/<alert_id>/triage	Triage alert	asha	action, notes, escalate?	Updated alert
GET	/api/asha/mother_details/<mother_id>	Full mother details	asha	—	Profile + weekly meals + alerts

Folder Structure

mothers_nutrition/latest_imp/

```
│
├── 📄 app.py                  Main Flask application
├── 📄 config.py               Configuration settings
├── 📄 models.py               Database models & functions
├── 📄 presets.py              RDA nutrition presets
├── 📄 requirements.txt        Python dependencies
│
├── 📁 routes/
│   ├── auth.py                Authentication routes
│   └── queries.py             Query/messaging API blueprint
│
├── 📁 templates/
│   ├── base.html              Base template
│   ├── login.html             Login page
│   ├── signup.html            Signup page
│   ├── mother.html            Mother dashboard
│   ├── doctor.html            Doctor dashboard
│   ├── doctor_profile.html    Patient profile page
│   ├── asha.html              ASHA worker dashboard
│   └── query.html             Legacy query page
│
├── 📁 static/
│   └── js/
```

```
│       └── doctor_form.js            Doctor form preset loader
│
├── 📁 utils/
│   ├── ocr_dummy.py                  Dummy OCR analysis
│   ├── nutrition_check.py            Nutrient comparison logic
│   └── nutrient_mapper.py            Nutrient mapping utilities
│
├── 📁 uploads/                       Meal photos storage
│
├── 📄 test_queries.py                Query API testing script
├── 📄 setup_query_db.py              Database setup script
```

Setup Instructions

This app uses Flask framework.The requirements in the requirements.txt doc can be installed. To run the app, the following command can be run:

python app.py

The database is online on MongoDB Atlas.