Design Document for NutritionTracker

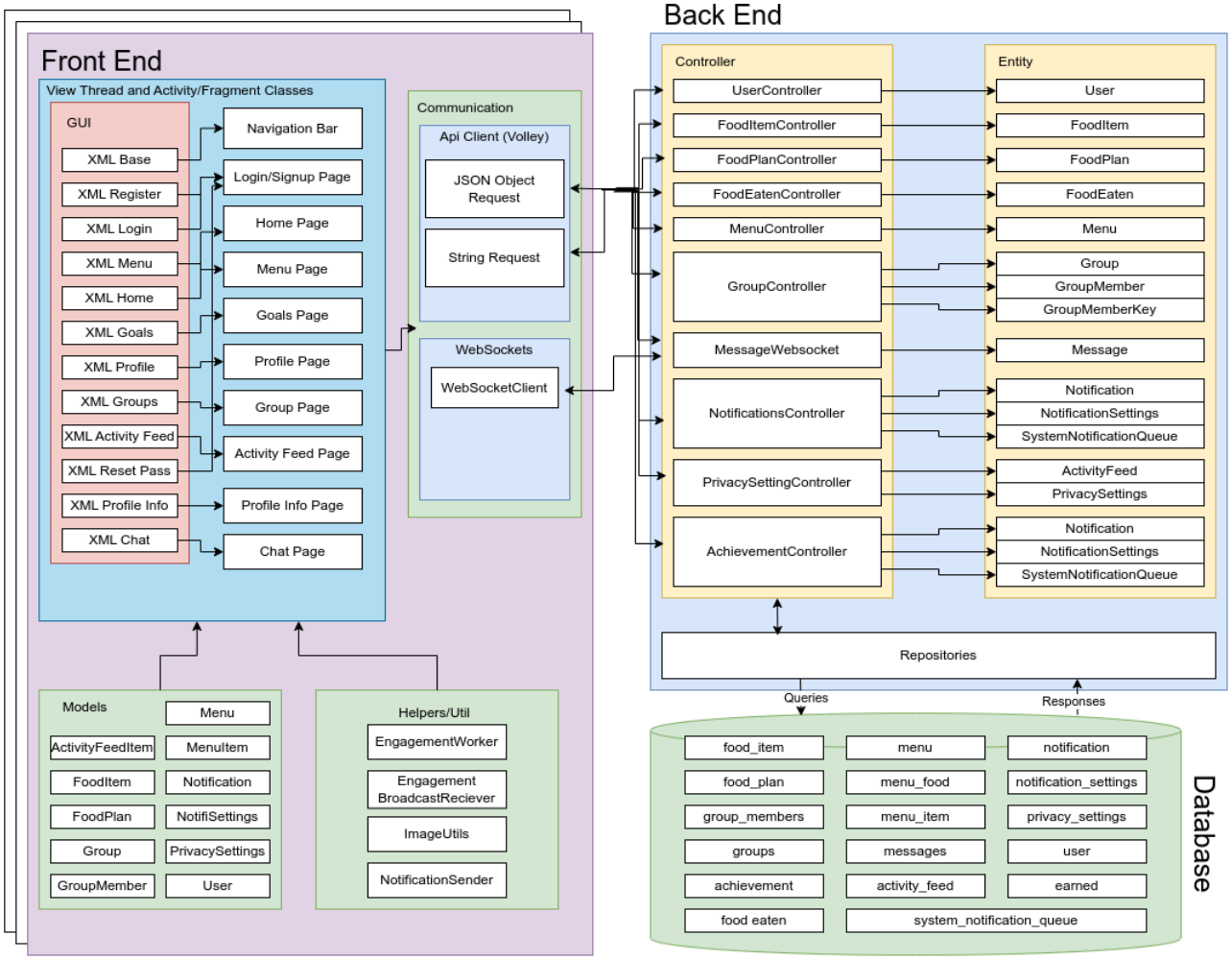
Group: 1_jubair_6

Alexander Svobodny: 30% contribution

Caleb Sanchez: 20% contribution

Connor Shepherd: 25% contribution

Molly Heston: 25% contribution


Front End
View Thread and Activity/Fragment Classes
GUI
XML Base
XML Register
XML Login
XML Menu
XML Home
XML Goals
XML Profile
XML Groups
XML Activity Feed
XML Reset Pass
XML Profile Info
XML Chat
Navigation Bar
Login/Signup Page
Home Page
Menu Page
Goals Page
Profile Page
Group Page
Activity Feed Page
Profile Info Page
Chat Page
Communication
Api Client (Volley)
JSON Object Request
String Request
WebSockets
WebSocketClient
Models
Menu
ActivityFeedItem
MenuItem
FoodItem
Notification
FoodPlan
NotifiSettings
Group
PrivacySettings
GroupMember
User
Helpers/Util
EngagementWorker
Engagement BroadcastReciever
ImageUtils
NotificationSender
Back End
Controller
UserController
FoodItemController
FoodPlanController
FoodEatenController
MenuController
GroupController
MessageWebsocket
NotificationsController
PrivacySettingController
AchievementController
Entity
User
FoodItem
FoodPlan
FoodEaten
Menu
Group
GroupMember
GroupMemberKey
Message
Notification
NotificationSettings
SystemNotificationQueue
ActivityFeed
PrivacySettings
Notification
NotificationSettings
SystemNotificationQueue
Repositories
Queries
Responses
food_item
menu
notification
food_plan
menu_food
notification_settings
group_members
menu_item
privacy_settings
groups
messages
user
achievement
activity_feed
earned
food eaten
system_notification_queue
Database

Front End

The front end is built using native Android with Java, following a modular approach that separates concerns for maintainability and efficiency. The main components are organized as follows:

View/Fragment Classes:

The UI layer comprises Activities and Fragments, each dedicated to a specific feature like Home, Menu, Goals, Profile, etc. Views are defined in XML layouts for consistent and structured UI.

Navigation:

A bottom navigation bar enables seamless switching between features, with the Android Navigation component managing Fragment transitions and back stack behavior.

Communication:

The app uses the Volley library for REST API communication, and a custom WebSocketClient for real-time updates like chat and activity feeds.

Models:

Core objects like Menu, FoodItem, Group, and User are represented by model classes to maintain data consistency and type safety throughout the application.

Helpers/Util:

Utility classes handle tasks such as image processing, data formatting, authentication, profile management, and notifications.

XML Resources:

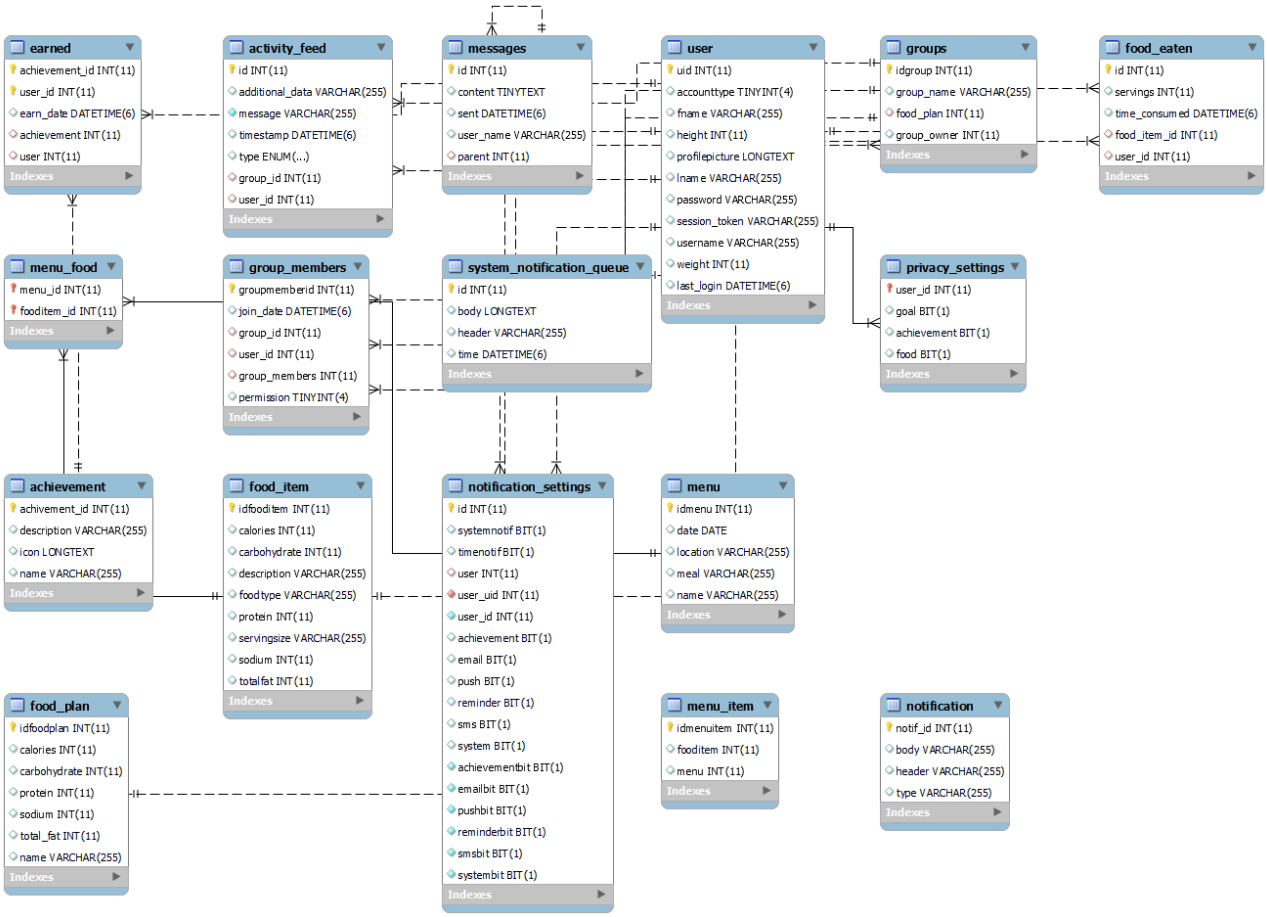
Layouts are organized by feature, including base components, navigation, menus, profiles, and settings, ensuring maintainability and clean structure.

Back End:

The back end is built with Spring Boot, hosted on a remote server provided by Iowa State University. Each major function has at least one controller and one entity. Controllers act as intermediaries between the front end and back end, using repositories to connect to the SQL database. Some repositories include optimized queries for efficient database searches.

Database:

All saved information is stored in an SQL database. This information contains user data and settings, food items, menus of food items, groups and their members, messages, achievements, and activity feed items. In the database, foreign keys are used to connect elements that are related to each other.


earned
achievement_id INT(11)
user_id INT(11)
earn_date DATETIME(6)
achievement INT(11)
user INT(11)
Indexes
activity_feed
id INT(11)
additional_data VARCHAR(255)
message VARCHAR(255)
timestamp DATETIME(6)
type ENUM(...)
group_id INT(11)
user_id INT(11)
Indexes
messages
id INT(11)
content TINYTEXT
sent DATETIME(6)
user_name VARCHAR(255)
parent INT(11)
Indexes
user
uid INT(11)
accounttype TINYINT(4)
fname VARCHAR(255)
height INT(11)
profilepicture LONGTEXT
lname VARCHAR(255)
password VARCHAR(255)
session_token VARCHAR(255)
username VARCHAR(255)
weight INT(11)
last_login DATETIME(6)
Indexes
groups
idgroup INT(11)
group_name VARCHAR(255)
food_plan INT(11)
group_owner INT(11)
Indexes
food_eaten
id INT(11)
servings INT(11)
time_consumed DATETIME(6)
food_item_id INT(11)
user_id INT(11)
Indexes
menu_food
menu_id INT(11)
fooditem_id INT(11)
Indexes
group_members
groupmemberid INT(11)
join_date DATETIME(6)
group_id INT(11)
user_id INT(11)
group_members INT(11)
permission TINYINT(4)
Indexes
system_notification_queue
id INT(11)
body LONGTEXT
header VARCHAR(255)
time DATETIME(6)
Indexes
privacy_settings
user_id INT(11)
goal BIT(1)
achievement BIT(1)
food BIT(1)
Indexes
achievement
achivement_id INT(11)
description VARCHAR(255)
icon LONGTEXT
name VARCHAR(255)
Indexes
food_item
idfooditem INT(11)
calories INT(11)
carbohydrate INT(11)
description VARCHAR(255)
foodtype VARCHAR(255)
protein INT(11)
servingsize VARCHAR(255)
sodium INT(11)
totalfat INT(11)
Indexes
notification_settings
id INT(11)
systemnotif BIT(1)
timenotifBIT(1)
user INT(11)
user_uid INT(11)
user_id INT(11)
achievement BIT(1)
email BIT(1)
push BIT(1)
reminder BIT(1)
sms BIT(1)
system BIT(1)
achievementbit BIT(1)
emailbit BIT(1)
pushbit BIT(1)
reminderbit BIT(1)
smsbit BIT(1)
systembit BIT(1)
Indexes
menu
idmenu INT(11)
date DATE
location VARCHAR(255)
meal VARCHAR(255)
name VARCHAR(255)
Indexes
food_plan
idfoodplan INT(11)
calories INT(11)
carbohydrate INT(11)
protein INT(11)
sodium INT(11)
total_fat INT(11)
name VARCHAR(255)
Indexes
menu_item
idmenuitem INT(11)
fooditem INT(11)
menu INT(11)
Indexes
notification
notif_id INT(11)
body VARCHAR(255)
header VARCHAR(255)
type VARCHAR(255)
Indexes