Universitatea Politehnica din Timișoara

CALORY - FOOD TRACKER MOBILE APPLICATION

Candidate:
Bogdan Tatu

Scientific coordinator:
Conf. Dr. Ing. Dan Pescaru

INTRODUCTION

1

INTRODUCTION

- full-stack mobile application
- user-friendly interface
- curated food data

WHAT?

- track food
- see daily stats (calories, macros, vitamins)
- track your progress
- add custom food
- change goals

WHY?

TECHNICAL BACKGROUND

2

BACKGROUND

1 Code Management

Git and GitHub

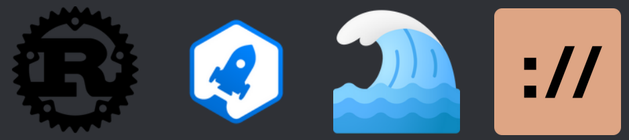
2 Database

PostgreSQL on Railway

3 Backend

Rust

SQLx, Tide, Surf, Serde



4 Frontend

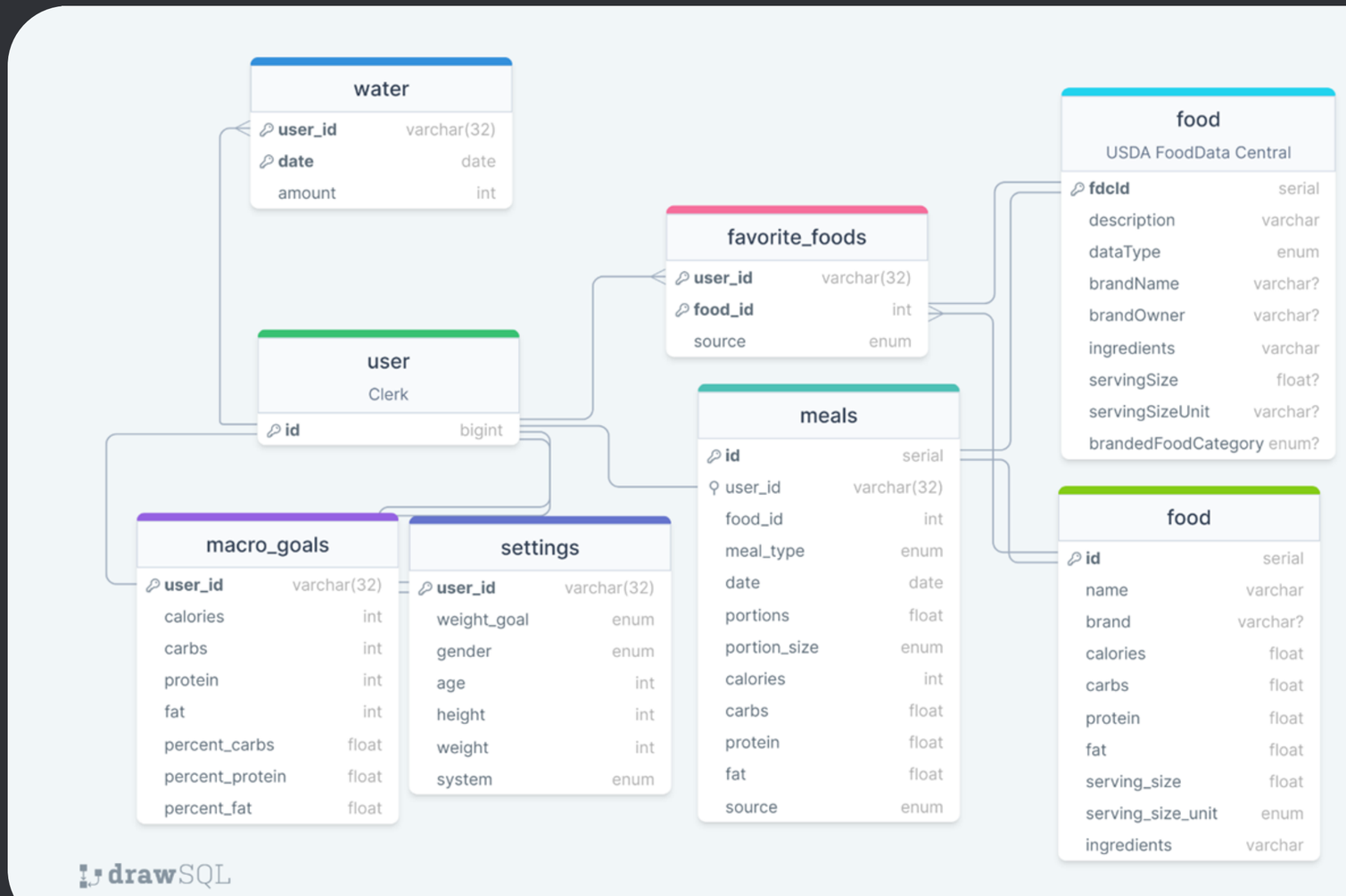
React Native (Typescript)

Expo, Tailwind, TanStack Query

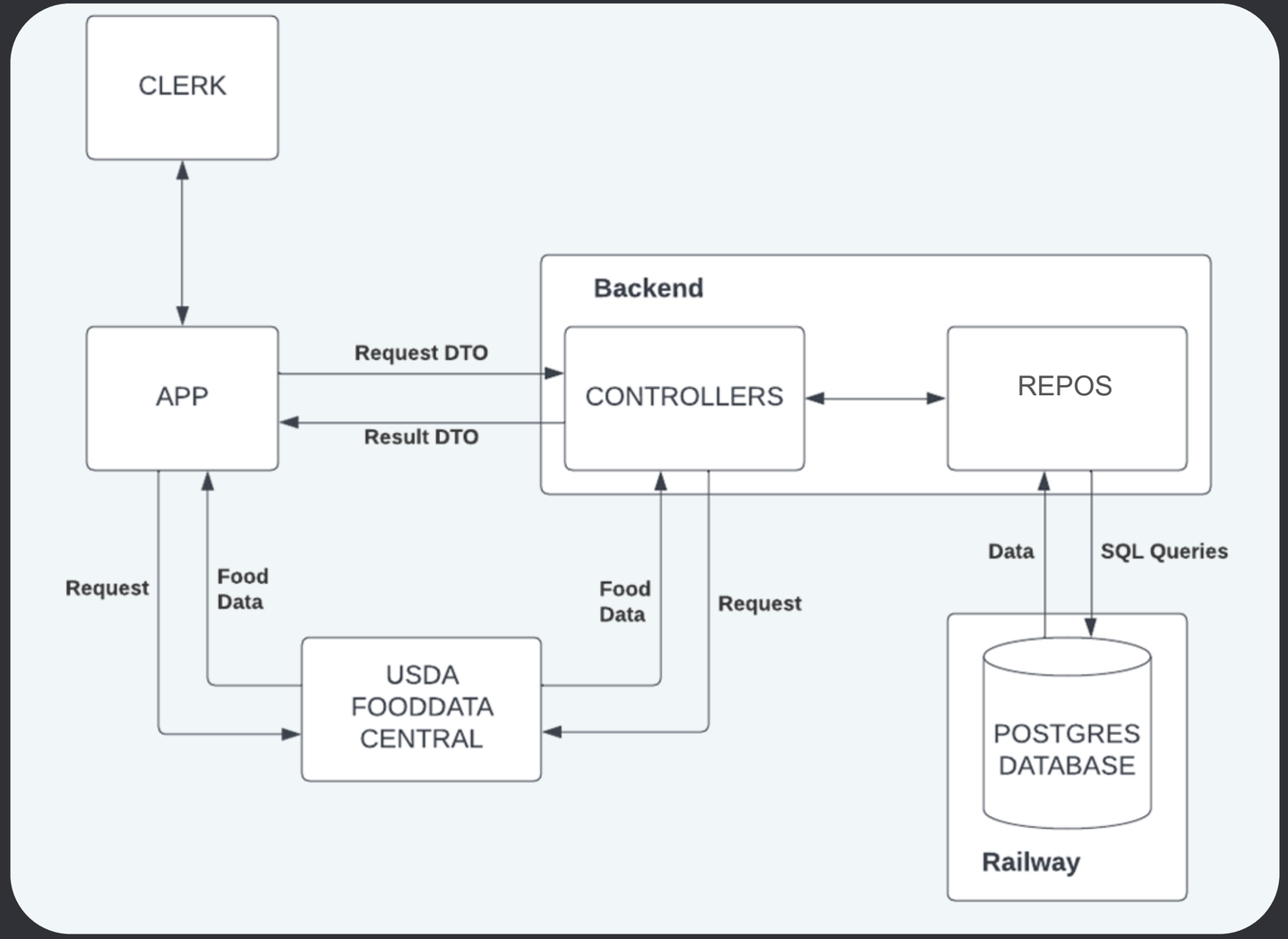
DESIGN

3

DATABASE


water
user_id varchar(32)
date date
amount int
favorite_foods
user_id varchar(32)
food_id int
source enum
food
USDA FoodData Central
fdcId serial
description varchar
dataType enum
brandName varchar?
brandOwner varchar?
ingredients varchar
servingSize float?
servingSizeUnit varchar?
brandedFoodCategory enum?
user
Clerk
id bigint
meals
id serial
user_id varchar(32)
food_id int
meal_type enum
date date
portions float
portion_size enum
calories int
carbs float
protein float
fat float
source enum
food
id serial
name varchar
brand varchar?
calories float
carbs float
protein float
fat float
serving_size float
serving_size_unit enum
ingredients varchar
macro_goals
user_id varchar(32)
calories int
carbs int
protein int
fat int
percent_carbs float
percent_protein float
percent_fat float
settings
user_id varchar(32)
weight_goal enum
gender enum
age int
height int
weight int
system enum
drawSQL

BLOCK DIAGRAM



IMPLEMENTATION 4

GET MEALS

"/api/v2/user/:uid/meals/:date"

```
pub async fn get_meals(req: Request<PgPool>) -> Result
```

```
let user_id = req.param("uid")?;
let date = req.param("date")?;

let connection = req.state();

let Ok(date) = NaiveDate::parse_ymd(date) else {
    return Ok(error_message!(
        tide::StatusCode::BadRequest,
        "invalid-date-format",
        "Invalid date format. Format must be YYYY-MM-DD"
    ));
};
```

GET MEALS

```rust
let mut usda_meals = meal_repo::get_by_user_date_and_source(
    connection, user_id, date, &Source::Usda
).await.map_err_to_server_error()?;

let mut user_meals = meal_repo::get_by_user_date_and_source(
    connection, user_id, date, &Source::User
).await.map_err_to_server_error()?;

if usda_meals.is_empty() && user_meals.is_empty() {
    return Ok(response!(StatusCode::Ok, Vec::<MealDto>::new()));
}
let usda_ids: Vec<i32> = usda_meals.iter()
    .map(|m| m.food_id).collect();
let user_ids: Vec<i32> = user_meals.iter()
    .map(|m| m.food_id).collect();
```

GET MEALS

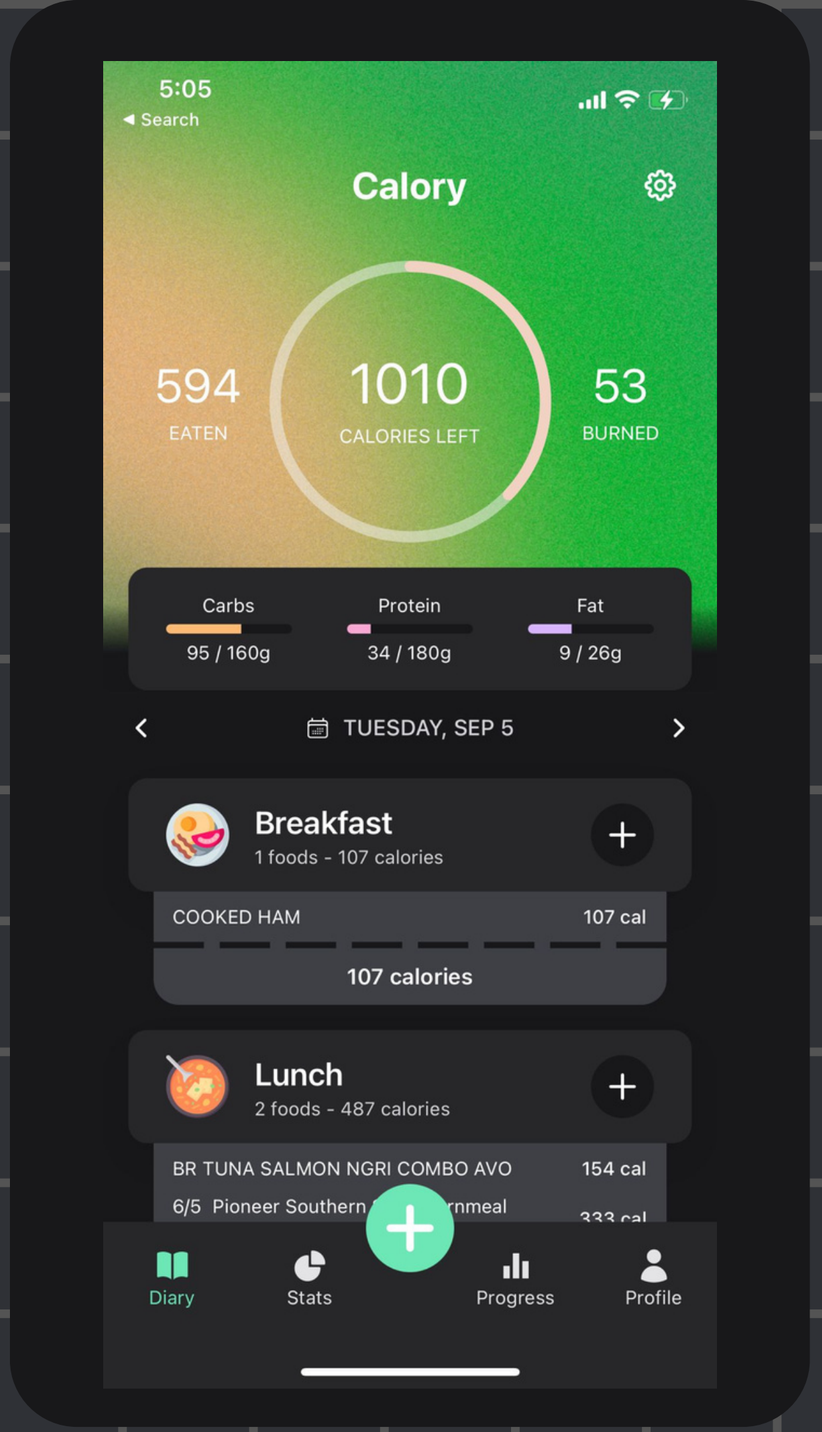
```rust
let mut usda_foods = usda_food::get_usda_foods_by_ids(usda_ids).await?;
let mut user_foods = food_repo::get_by_ids(connection, &user_ids)
    .await.map_err_to_server_error()?
    .into_iter().map(|f| f.into()).collect::<Vec<_>>();

usda_meals.append(&mut user_meals);
usda_foods.append(&mut user_foods);

let meals = std::iter::zip(usda_meals, usda_foods)
    .map(|(meal, food)| MealDto::from_meal(meal, food))
    .collect::<Vec<_>>();

Ok(response!(StatusCode::Ok, meals))
```

RESULT


5:05
Search
Calory
594
EATEN
1010
CALORIES LEFT
53
BURNED
Carbs
95 / 160g
Protein
34 / 180g
Fat
9 / 26g
TUESDAY, SEP 5
Breakfast
1 foods - 107 calories
COOKED HAM
107 cal
107 calories
Lunch
2 foods - 487 calories
BR TUNA SALMON NGRI COMBO AVO
154 cal
Diary
Stats
Progress
Profile

CONCLUSION

5

FUTURE WORK

FEATURES

- ACHIEVEMENTS
- SMART WATCH
- AI FOOD RECOGNITION

DATA

- DATABASES
- USERS

THANK YOU!