

Design Doc

Introduction

Scope

A social platform for sharing recipes with each other. The project's goal is to provide ideas for varied and creative meals and make choosing what to cook easy and accessible.

Overview

In this document I'll describe 3 user's journeys.

Terminology

- React - frontend language
- Node.js - backend language
- MongoDB - Database

Software Design Description

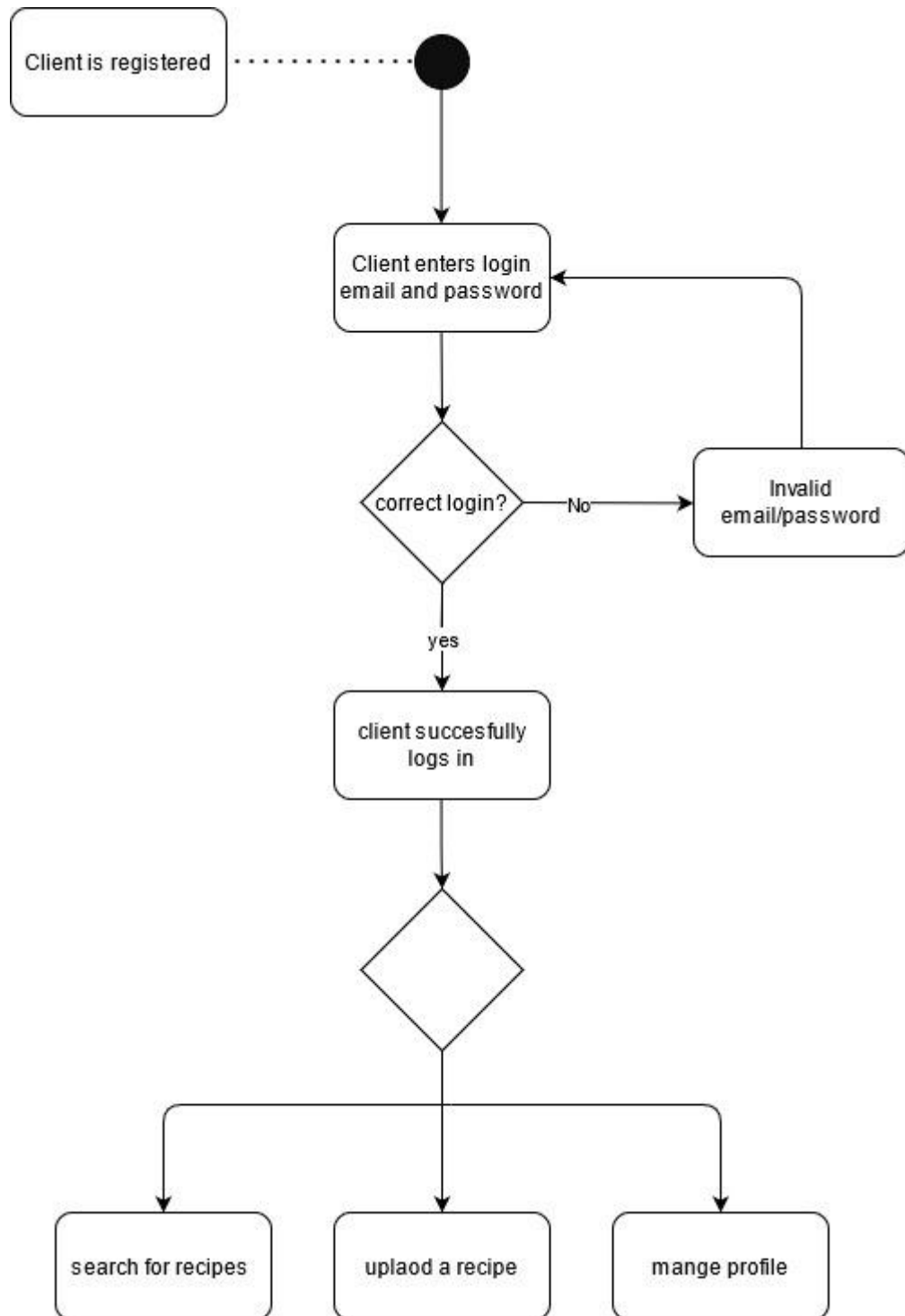
First, to use the website, each user needs to create an account.

On the website, users can search for recipes by category, by the meal's type (breakfast/lunch/dinner), or to view all the recipes that are in the website. Also users can upload and share their own recipes. In the profile management page, each user can edit his personal info and manage his posts.

Software Architecture

- Login form
- Sign Up form
- Homepage with all the recipes
- Upload a new recipe
- User profile management

User Interface



- Login: a form with name, email and password fields
- Signup: a form with email and password fields
- Homepage: a web page with all the recipes on the website. users can search for recipes by category or meal's type
- Upload a recipe: a form with title, ingredients, time it takes, preparation instructions, recipe's categories, recipe meal's type.
- Manage your profile: update personal info, manage your recipes (edit, delete).

Data Handling

The data that will be saved:

- User's info (when creating an account - name, email, password)
- User's recipes, likes and comments.
- Logs (see in test and monitor section)

I'm planning to use MongoDB. When a user fills and submits a form the data will save in the relevant table in the DB.

Integration with other systems

The frontend side will integrate with MongoDB and will save, read and display the relevant data. The backend side will also integrate with MongoDB and will use data for its calculations (It will calculate a rank for each user).

Tests And Monitor

Tests

- Invalid inputs:
 - A user submits an empty form.
 - A user fills invalid email address.
-

Logs

- User creates an account
- User login
- User uploads a recipe
- User searches for a recipe (?)

Alerts

- When there is busy traffic on the website.
- When there is left 95% available storage in the DB.
- When a recipe is deleted not by the user who uploaded it.