FitNesse App

Analysis and Design Document

Student: Oancea Eduard

**Group: 30431**

Revision History

|  |  |  |  |
| --- | --- | --- | --- |
| **Date** | **Version** | **Description** | **Author** |
| 18/Mar/20 | 1.0 | Initial analysis and design | Oancea Eduard |
| 30/Apr/20 | 1.1 | Add Component and Deployment diagrams, as well as Data Model and Design Model | Oancea Eduard |
|  |  |  |  |
|  |  |  |  |

Table of Contents

I. Project Specification 4

II. Elaboration – Iteration 1.1 4

1. Domain Model 4

2. Architectural Design 4

2.1 Conceptual Architecture 4

2.2 Component and Deployment Diagrams 5

III. Elaboration – Iteration 1.2 5

1. Design Model 5

1.1 Dynamic Behavior 5

1.2 Class Design 5

2. Data Model 5

3. Unit Testing 5

IV. Elaboration – Iteration 2 5

1. Architectural Design Refinement 5

2. Design Model Refinement 5

V. Construction and Transition 5

1. System Testing 5

2. Future improvements 5

VI. Bibliography 5

# Project Specification

Design and implement a desktop application for a fitness tracking application. The application should have 3 types of users(a regular user, a super user and an anonymous user).

The regular user can perform the following operations:

* CRUD on tracking entries
* CRUD on goals
* CRUD on the food database
* View certain statistics

The anonymous user can perform the following operations:

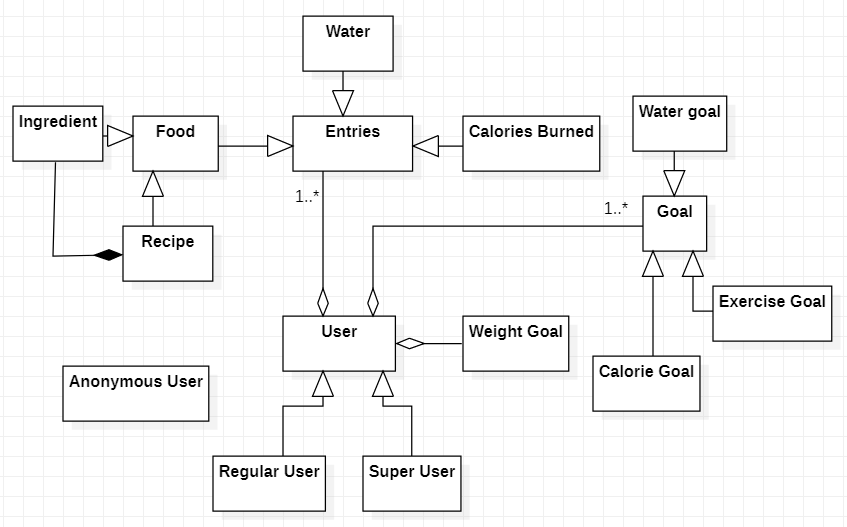
* Sign in
* Sign up

The super user can perform the following operations:

* CRUD on users
* CRUD on the food database

# Elaboration – Iteration 1.1

# Domain Model

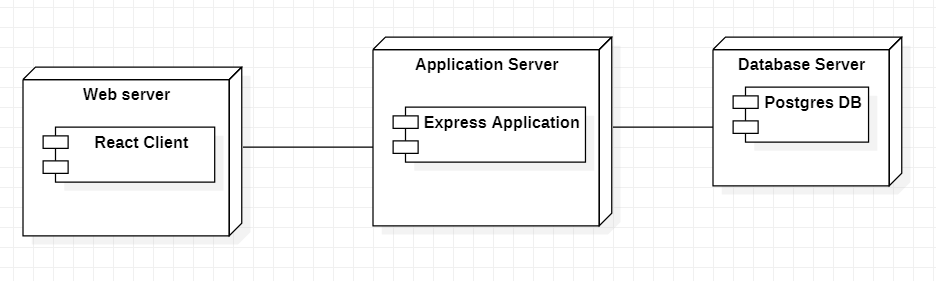


# Architectural Design

## Conceptual Architecture

This project is designed using the layered architecture pattern, as it is a client-server application. It’s built with Javascript with React as the client-side framework and Express as the server-side framework. The data will be persisted in a Postgres relational database. The architectural style of choice is Volatility Based Decomposition. This is especially easy to do with React, as it promotes a component-based design.

## Component and Deployment Diagrams

**

# Elaboration – Iteration 1.2

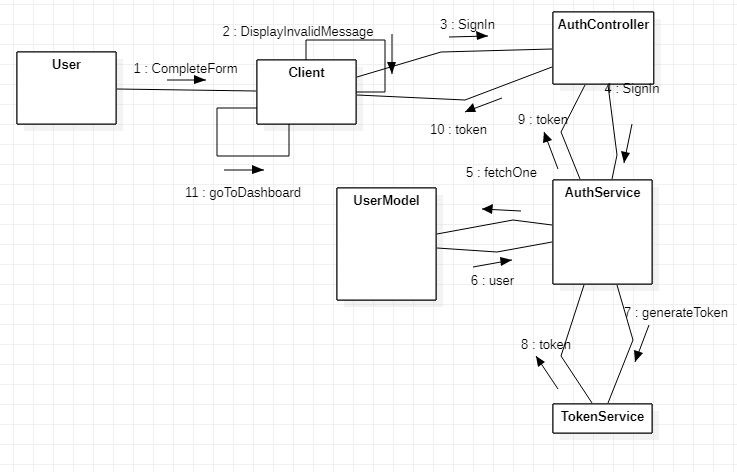
# Design Model

## Dynamic Behavior

Sequence diagram for the authentication process.

*A screenshot of a cell phone

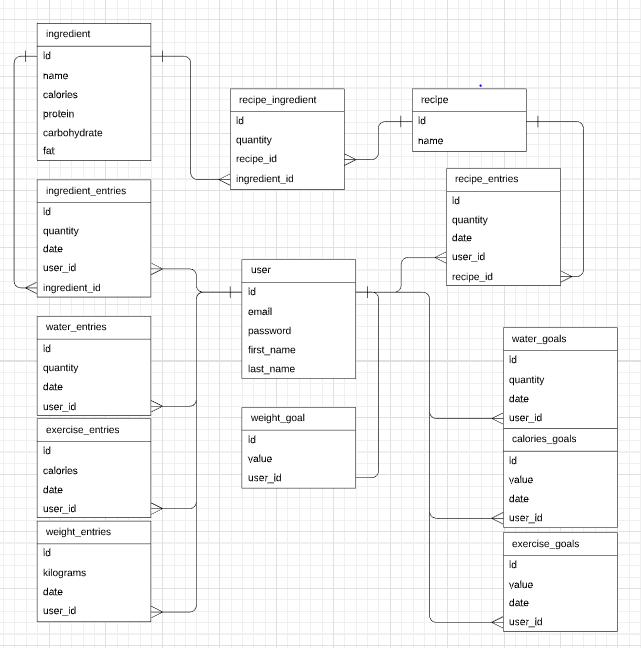
Description automatically generated*

**

## Class Design

*[Create the UML class diagram; apply GoF patterns and motivate your choice]*

# Data Model

**

# Unit Testing

*[Present the used testing methods and the associated test case scenarios.]*

# Elaboration – Iteration 2

# Architectural Design Refinement

*[Refine the architectural design: conceptual architecture, package design (consider package design principles), component and deployment diagrams. Motivate the changes that have been made.]*

# Design Model Refinement

## *[Refine the UML class diagram by applying class design principles and GRASP; motivate your choices. Deliver the updated class diagrams.]*

# Construction and Transition

# System Testing

*[Describe how you applied integration testing and present the associated test case scenarios.]*

# Future improvements

*[Present future improvements for the system]*

# Bibliography