Design Specification Document

Health Tracker

Can Tuksavul

2016

Contents

[Introduction 3](#_Toc469933623)

[Architecture 3](#_Toc469933624)

[Client Side Architecture 4](#_Toc469933625)

[Components 4](#_Toc469933627)

[Client Components 4](#_Toc469933628)

[API 5](#_Toc469933629)

[Test 5](#_Toc469933630)

[Login 5](#_Toc469933631)

[Register 5](#_Toc469933632)

[Update User Details 5](#_Toc469933633)

[Get the User Details 5](#_Toc469933634)

[Get Activities 5](#_Toc469933635)

[New Activity 5](#_Toc469933636)

[Search NDB API 5](#_Toc469933637)

[Get NDB Food 5](#_Toc469933638)

[Create Consumption 5](#_Toc469933639)

[Get All Foods in a Given Date 6](#_Toc469933640)

[Get Calorie Intake Between Given Dates 6](#_Toc469933641)

[Get Total Nutrition Values on a given Date 6](#_Toc469933642)

[Create/Update Weight on a Given Date 6](#_Toc469933643)

[Get Weight on a Given Date 6](#_Toc469933644)

[Get Weights between a given date interval 6](#_Toc469933645)

[References 6](#_Toc469933646)

# Introduction

This document triesr to capture design elements of the Health Tracker software produced for the Software Development Practice course in Bogazici University Software Engineering Department.

Main focus will be on transferring design idea through visual representations and their supporting textual descriptions.

Highest level representations will be the following: Deployment Diagram, Component Diagram

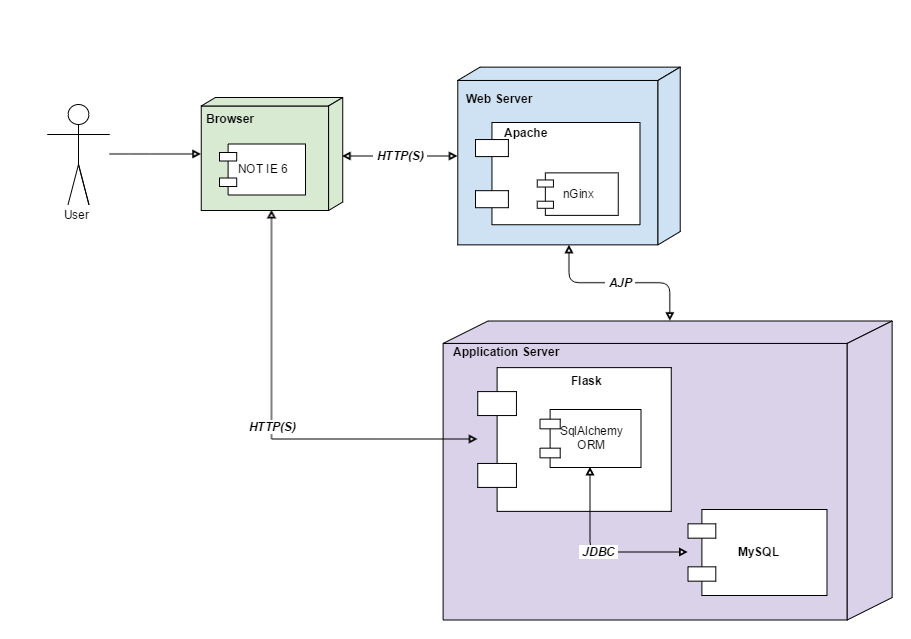
ER Diagram is used for showing the application’s main assets.

Rest of the representations focus on providing implementation design view by using the following: Sample Class Diagram, Sample Sequence Diagram.

Server Backend API will be provided as it is sort of the backbone of the application and it also shows parallelism to use cases as well as it can be used for providing entry points to both backend and frontend applications.

# Architecture

Server Client architecture is used. Client uses MVC architecture. Server is designed to be a RESTful service. Datastore communications are handled through Object Relational Mapping.



## Client Side Architecture

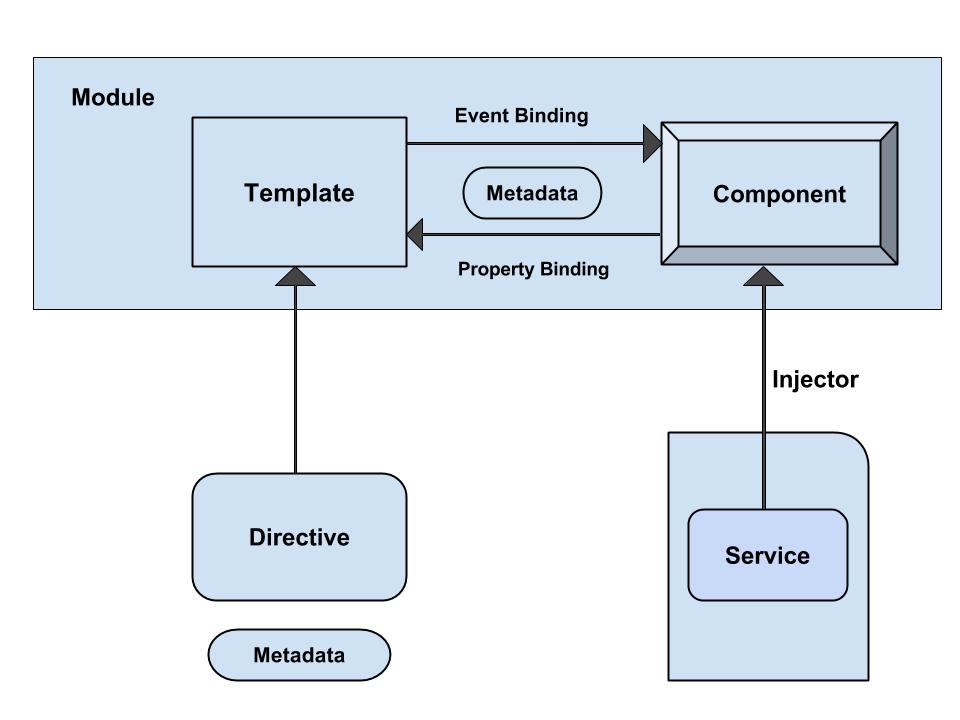
Following is one of the best descriptions of the Angular 2 architecture. Reference: (tutorialspoint)

Angular applications consists of modules and components. Modules bundles components, pipes, services and handles dependency injection.

Components are generally act as views and controllers together of the traditional Model View Controller pattern.

Services act similar to Models but I also chose to add abstract classes of the backend models in a separate folder to increase readability.

# 



# Components

Components is shown in two sections.

## Client Components

# API

## Test

Simple test url. Calling this should return an empty template with “Hello World” text.

url: ht.cantuksavul.com:5000

## Login

Logs in with the UserCredential model credentials.

url: ht.cantuksavul.com:5000/api/auth', methods=['POST']

## Register

Registers a new user with the UserCredential model credentials.

url: ht.cantuksavul.com:5000/api/auth/new', methods=['POST']

## Update User Details

Updates user details with the UserDetails model.

url: ht.cantuksavul.com:5000/api/user/<int:user\_id>', methods=['PUT']

## Get the User Details

Returns UserDetail with the given Id.

url: ht.cantuksavul.com:5000/api/user/<int:user\_id>', methods=['GET']

## Get Activities

Returns Activity[] model for the given date

url: ht.cantuksavul.com:5000/api/user/<int:user\_id>/activity/<date>', methods=['GET']

## New Activity

Creates new activity for the given date.

url: ht.cantuksavul.com:5000/api/user/<int:user\_id>/activity/new', methods=['POST']

## Search NDB API

Searches chars from NDB API.

url: ht.cantuksavul.com:5000/api/food/<chars>', methods=['GET']

## Get NDB Food

Returns the Food model of the food with the given ndbno.

url: ht.cantuksavul.com:5000/api/food/<ndbno>/measures', methods=['GET']

## Create Consumption

Creates a consumption with the Food model.

url: ht.cantuksavul.com:5000/api/user/<int:user\_id>/food/new', methods=['POST']

## Get All Foods in a Given Date

url: ht.cantuksavul.com:5000/api/user/<int:user\_id>/food/<date>', methods=['GET']

## Get Calorie Intake Between Given Dates

url: ht.cantuksavul.com:5000/api/user/<int:user\_id>/consumption/energy/<start\_date>/<end\_date>', methods=['GET']

## Get Total Nutrition Values on a given Date

url: ht.cantuksavul.com:5000/api/user/<int:user\_id>/consumption/<date>', methods=['GET']

## Create/Update Weight on a Given Date

Using Weight model

url: ht.cantuksavul.com:5000/api/user/<int:user\_id>/weight/<selected\_date>', methods=['PUT']

## Get Weight on a Given Date

Using Weight model

url: ht.cantuksavul.com:5000/api/user/<int:user\_id>/weight/<selected\_date>', methods=['GET']

## Get Weights between a given date interval

Using weight[] model.

url: ht.cantuksavul.com:5000/api/user/<int:user\_id>/weight/<start\_date>/<end\_date>', methods=['GET']

# References

*tutorialspoint.* (n.d.). Retrieved from https://www.tutorialspoint.com/angular2/angular2\_architecture.htm