מש – מבשלים עם מה שיש

# Software Design Document(SDD)

**Name: Amit Tzumi**

**Date: 15/12/2021**

**TABLE OF CONTENTS**

1. [INTRODUCTION 2](#_bookmark0)
   1. [Purpose **2**](#_bookmark1)
   2. [Scope **2**](#_bookmark2)
   3. [Overview **2**](#_bookmark3)
2. [SYSTEM OVERVIEW 2](#_bookmark6)
3. [SYSTEM ARCHITECTURE 3-4](#_bookmark7)
   1. [Architectural Design **3**](#_bookmark8)
   2. [Decomposition Description **4**](#_bookmark9)
   3. [Design Rationale **4**](#_bookmark10)
4. [DATA DESIGN 5](#_bookmark11)
   1. [Data Description **5**](#_bookmark12)
   2. [Data Dictionary **5**](#_bookmark13)
5. [COMPONENT DESIGN 6](#_bookmark14)
6. [HUMAN INTERFACE DESIGN 7-8](#_bookmark15)
   1. [Overview of User Interface **7**](#_bookmark16)
   2. [Screen Images **7**](#_bookmark17)
   3. [Screen Objects and Actions **8**](#_bookmark18)

### INTRODUCTION

## Purpose

This software design document describes the architecture and system design of my project - Mesh - cook with what you have. This document is intended for my facilitator - Eran Shmuel.

## Scope

In the "Mesh - cook with what you have" project, we will present a unique online recipe site that includes a variety of recipes in a variety of categories. The site offers its customers to choose recipes according to the ingredients they have at home. Each user will enter at the entrance to the site the groceries he has at home and / or in the refrigerator, thus adjusting recipes accordingly, without having to leave the house for shopping.

## Overview

In this document, I would like to present the system and its decomposition into subsystems, of what each subsystem is composed in great detail. In addition, I will also show how and where I store my data. Finally, I will show you how the UI – user interface looks like.

### SYSTEM OVERVIEW

When you log in to the site, you will send an HTTP request to the web server and receive the site's login page from the server.

The application will contain a system for user management, so when browsing the site, the user will be asked to register for the site or to perform a login process, in this process, a check will be made against the database that the user is actually registered on the site.

The application will be divided into two main modules:

Client side - where all the UI part, the visual part, this part will be divided into screens, each screen will include HTML, CSS, JAVASCRIPT file. The screen will contain all the relevant visual objects (grids, buttons, scroller bar, textfiels and more)

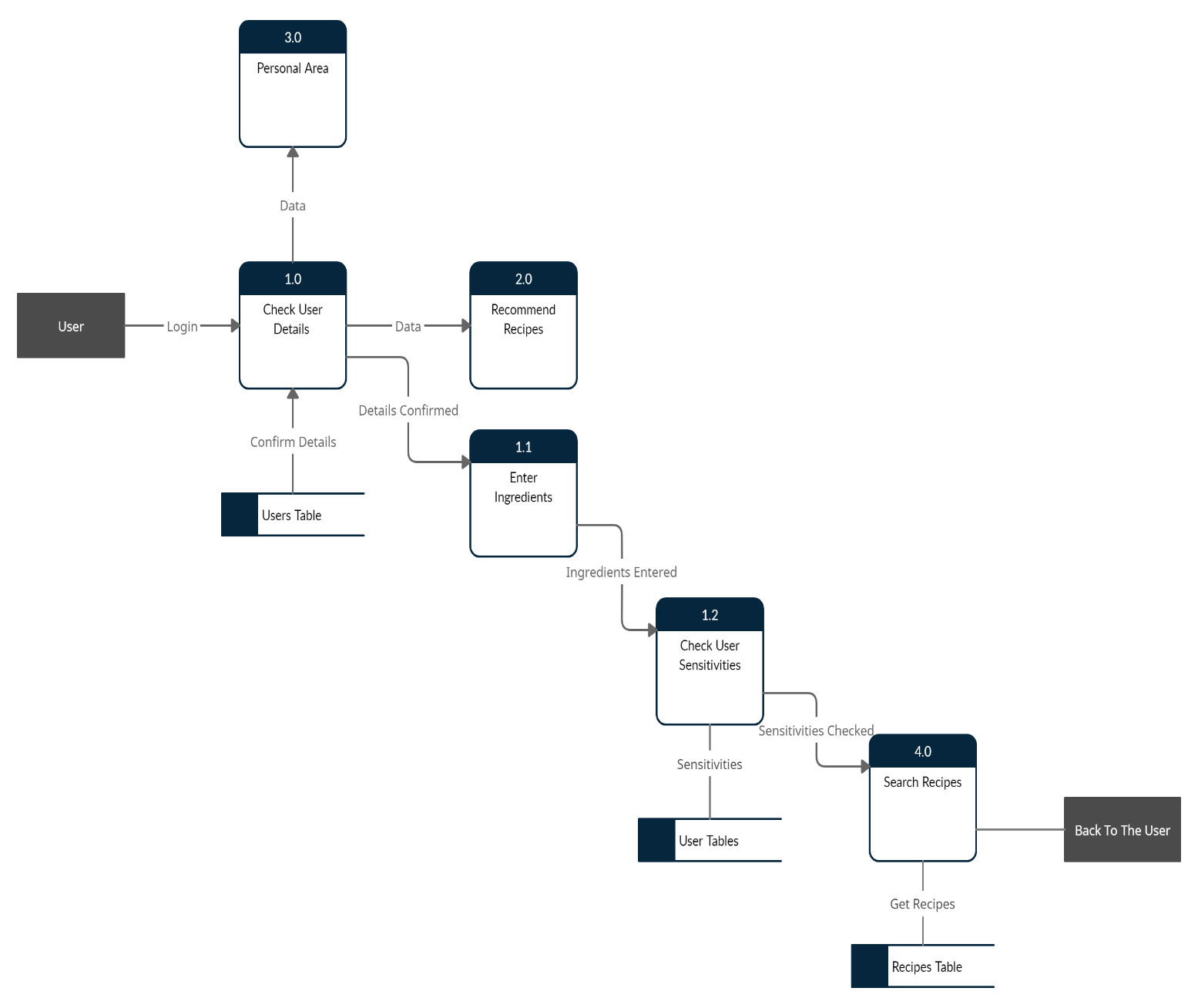
Server Side - on this side will be kept the database, classes and relevant functions.

### SYSTEM ARCHITECTURE

## Architectural DesignC:\Users\Amitt\AppData\Local\Microsoft\Windows\INetCache\Content.Word\Final 3.1 11.png

When the user enters the app he must log in to get to the home screen. On the home screen the user has three options: **1.** Finding recipes according to specific types of food / ingredients - the user will enter specific food ingredients and the system will search the database for selected recipes according to the user's choice and depending on his sensitivities in the personal area. **2.** Personal area - a user who is connected to the system will be able to enter the personal area, where data about the user profile will appear. In this window, the user has the option to set his sensibilities and see his favorite recipes. **3.** Recommended recipes - the user has the option to see the recipes with the highest rating on the site. In order to display the data to the user we need to send an HTTP request to the server so that he can bring us the data or update it according to the user's choice.

## Decomposition Description



## Design Rationale

I chose this architecture because I looked from the user's point of view and so it seems to me the friendliest, and if the user wants to access and see any data then he should ask permission from the server.

### DATA DESIGN

## Data Description

When we want to access data from the server we send an HTTP request and then the server returns us a JSON file, after we get the JSON file, we convert it to suitable data structures. In our case in the get\_recipes function, we convert our JSON file to a list of type Recipe, each Recipe has the attributes: recipe\_id, name, ingredients, categories, link, image\_link, description. In the get\_recommended function we also convert the JSON file to a list of type Recipe like in the get\_recipes function, because we are also need to return recipes. In the set\_sensitive function we need to change the user's sensitivities so we convert the JSON file to a list of type String that will show us the ingredients to which the user is sensitive. We intend to use a Data Scrapper that will take data from recipe sites and put it into our database. Our database will contain 2 main tables: Users, Recipes.

## Data Dictionary

|  |  |
| --- | --- |
| **Users** | |
| PK | id int NOT NULL |
| key | name char NOT NULL |
| key | email char NOT NULL |
| key | password char NOT NULL |
| key | sensitives list |

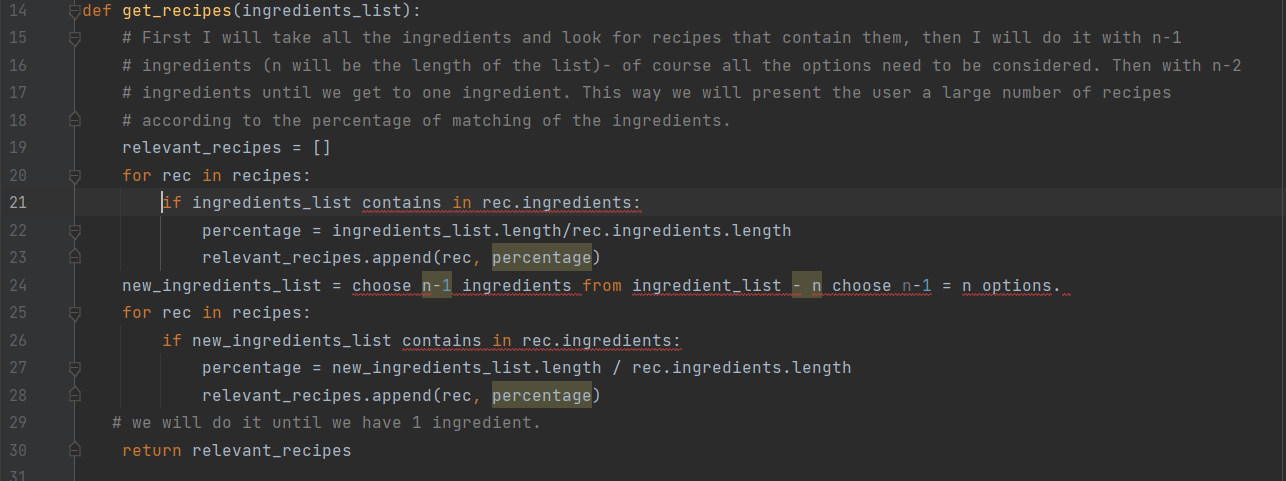
|  |  |
| --- | --- |
| **Recipes** | |
| PK | id int NOT NULL |
| key | name char NOT NULL |
| key | ingredients list NOT NULL |
| key | categories char NOT NULL |
| key | link char NOT NULL |
| key | image\_link char NOT NULL |
| key | description char NOT NULL |
| key | likes int |

set\_sensitive(List of ingredients) - This function receives a list of ingredients and updates the sensitivity field in the Users table.

get\_recommended() – This function returns the highest rated recipes from Recipes table.

get\_recipes(List of ingredients) - This function receives a list of ingredients that the user has selected and returns recipes that contain these ingredients.

### COMPONENT DESIGN



### HUMAN INTERFACE DESIGN

## Overview of User Interface

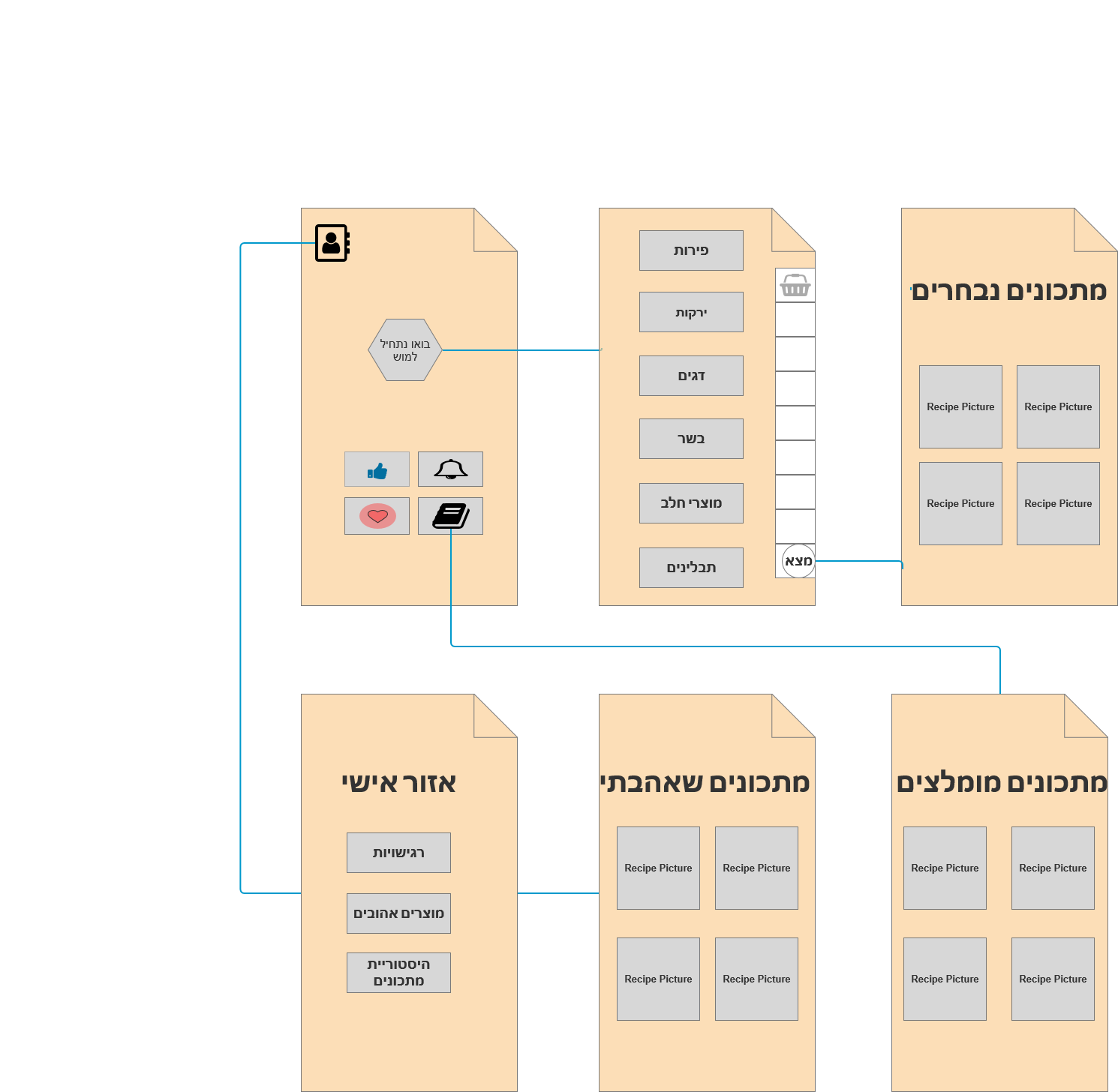
1. Login / register to the system- registration to the system includes analysis of the user profile, the user will enter ingredients that we do not want to use when preparing the recipe, in addition, characterizing the customer such as: vegetarian, vegan, gluten free, etc.

2. Finding recipes according to specific types of food / ingredients - the user will enter specific food ingredients and the system will search the database for selected recipes according to the user's choice.

3. Personal area - a user who is connected to the system will be able to enter the personal area, where data about the user profile will appear. The user can see his favorite recipes, can set his sensitivities.

4. Recommended recipes – a user can see the highest rated recipes in the app.

## Screen Images



## Screen Objects and Actions

The home screen has 3 buttons - the personal area, "bo nathil lamush", recommended recipes. If the user clicks on the personal area - he will be able to edit his sensitivities, see his favorite recipes and his recipe history. If the user clicks on "bo nathil lamush" then he will be taken to the ingredients selection page, where he will enter the ingredients he has at home to get recipes according to his products. Once the user has entered the products, he will be taken to a page which will show him the recipes relevant to him. If the user clicks on the recommended recipes - he will be taken to the page where the 20 recipes with the highest rating in the app are displayed.