**(**Meals-schedule**)**

# Software Design Document

### INTRODUCTION

## Purpose

The purpose of this document is to give more information about our project. To dive in to the specific components that are in our project. While making this document we learnt and got knowledge of how the components are going be connected, how we are going to build our tables in the database and more.

## Scope

The main goal is to help restaurants, companies, nutritionists, individuals and more to achieve better schedule in their meal plans. Our project will have comprehensive functionality which will be beneficial for the users because they will be able to gain all the information they need for a meal plan (such as the cost, the ingredients, the nutrition values, instructions of how to store specific ingredients, and more (is detailed in the document " Software Requirements Specification" under project-scope)).

### SYSTEM OVERVIEW

General description – The application will be able to perform several functionalities that their main goal is to give the users all the information they need to schedule a meal. Starting from choosing the right schedule, adding, or creating a recipe and its ingredient, how to store and how to make the recipe and most importantly the cost (what is the cost for each ingredient, for the whole recipe and for the entire schedule). For each functionality there will be an algorithm that will be connected to the database, writing, or reading from it using php language. Several algorithms will be a result from the user action, and some will be not a directed outcome from the user actions (For example while one user uploads a recipe and choosing to share, another user as a result sees it in the share recipes. The main function was writing to the database and allowing the recipe to be shared, but while doing that – another user went to the shared recipes and saw a new recipe that was pulled from the recipes table in the database).

Design – we tried to come up with the right design so that when we will reach the step of writing the code it will be the easiest for us. We built our tables in the database in a way that it will be easy to access and write queries. We designed the user-interface in a way we believe it will be the easiest for the user to understand how to operate our application. We came up with all the functionalities after performing market survey and observing what each project had missing. We combined everything we believe is necessary and added even more.

### SYSTEM ARCHITECTURE

## Architectural Design –

In the PowerPoint "Tables in database " we specified the tables we need in order to store all the data for the application.

In the PowerPoint "User Interface " we specified the gui generally will be. What functionality the user can perform and what information the user can receive according to their actions.  
In the build-in database (recipes, ingredients) we will need external source that is able to provide us the necessary stuff.

To make the application well reliable, we will need some sort of image processing, to make sure input validity. Therefore, we need to search for a build-in algorithm.

Communication with the database will be with php. We will need to write each GET / POST according to the necessity that will rise from the user actions. That action will be received in the code, in our code we will know to trace which php file we need to send to the database to receive the right information for the user.

## Decomposition Description

Listed in the PowerPoint "Tables in database".

### DATA DESIGN

## Data Description

For this project we will used database of MySQL and php to communicate with it.  
This kind of database stores its information in tables. Each component will have its own table, linking it to small tables to store information of a kind (the linking will help us to extract information more easily (by foreign key)).  
In order to use the information from the database, we will pull the information and create an object. It will help us to use it in the code, we will be able to perform other functions on this object according to the user request from the application.  
User table – will have user details to store id, email, and other necessary information  
(for example, by the user id, we can extract user self-ingredient/recipes and display it in the target place).   
Ingredient table – will store all the ingredients in the system  
Recipes table - will store all the recipes in the system.

## Data Dictionary

We are building the database tables in a way that the design will help us access the data later-on in an easy way. We tried to design our next steps in the code, think how we can build the tables to be in order for to make it most beneficial. In the efficiency way and in the easier way, we took them both under consideration.

For the tables general look you can open the PowerPoint "Tables in database".

### COMPONENT DESIGN

1. . In order to insert information, we will access the database using php language and write each query to match its algorithm.
2. All the information we will receive from the user and pull it to our code, then send it to the database.
3. We split the tables in order to create database that is easier to access and extract information, also to make normalized tables.
4. There is a page for the user to insert all the information in order to create a recipe/ingredient/ schedule a specific meal (daily, weekly,..,event etc)/create profile/and more .We will fill this information in the specific table and create a unique id for it.
5. User table: we save the user information as we create the profile.
6. "Ingredient table "-

This table is connected to 3 more tables that holds further information about that recipe – "Share-ingredient table" , "Ingredient-cost", "Nutritious table" .

1. Recipe table :

This table is also connected to 3 different tables that contains more information about the recipe – "Share- recipe"," Instruction-recipe-text table","Recipe-ingredient table".

1. Daily table :

This table is also connected to other table called – "Daily schedule". By creating a daily meal schedule first we will create a unique key for that schedule and add it to the table with the userID. In the second table we will write the information we need in order to make that schedule, such as the number of meals in the day, the recipes for that meal, the quantity we want from it. We know that for more information we will access it from the recipe table that we receive from the recipes we want for that schedule.

1. Weekly table :

This table is also connected to "Weekly-Daily-meals table". In the first table we save the unique id we create and the userID that created that schedule. In the second table we save the further information for the schedule, such as the number of day, and which daily we rely on (Saved as the dailyID that we can access its information from "Daily table"). Same goes for "Monthly table", "Yearly table" but instead of the number of day it will be the number of week/ number of month accordingly.

1. Upcoming schedule table – will be presented by daily :

Once the user decides to schedule a date for a certain meal-plan we will extract its dailyID and represent it by that. (For example if we want to schedule a date for a certain monthly schedule that we already created, we will go in that table and have its weekly information. From weekly information table we will go deeper to its daily information and extract each daily. So by that we will represent the upcoming daily the user have, Accordingly we will do for the other schedule). We will insert the information into that table by duplicating the information from the specific schedule the user made, we will not work on the same data as the information from the previous tables.

1. Event Information table: This table automatically has a date set for when the user decides, unlike the other schedule that we can create without scheduling a date. In this table we save the userId, the name of the event , the date and the unique schedule we set for it. The further information we save in the second table that saves all the recipes we want (by the recipeID) the quantity we want from that recipe and for which event-schedule it connected to.

1. **REQUIREMENTS MATRIX**

|  |  |
| --- | --- |
| Functionality | Algorithm |
| Daily, weekly, monthly and yearly cost for a recipes that schedule for each of them. | We will retrieve all the needed ingredient and the quantity needed for a specific recipe and calculate the cost per ingredient with its quantity, and at the end sum all up and that will be the cost for the recipe. daily – we will sum all the recipe schedule for a specific day.  weekly – we will sum all the recipes schedule for (1-7 days of the week, can be in this range, if for example we schedule 5 days only, then summing all the 5 days recipes cost)  Monthly, yearly – in the same way. |
| Daily, weekly, monthly and yearly amount of ingredients(in grams) for a recipe. | We will go over each recipe and extract the ingredients needed. For each ingredient we will look for how much gram of it we need for the recipe and add to the list. Eventually we will have all the ingredients with the quantity to represent the user for all the recipes for the day/week/month/etc |
| Provide the users the option to add new recipe that was created by them | In the recipes page the user will be able to be directed to "create recipe" page. In that page we will give the user the option to fill up the needed information in order to complete the new recipe. For example the instruction for how to make it with an options to upload a picture for every step, name of the recipe , and more. At the end we will save that data in our database and present it to the user in the recipes page. |
| Allowing the user to mark recipe as "favorite" and presenting them in different page. | For each recipe that is presented in the recipes page, there is the ability to mark it as favorite. Once the user mark it, in the database we extract the recipe by the recipeId and copy the entire data to a new row with a new recipeID and new ownerID (because we don’t want that if one changes it, then the other will receive the changes as as well). After we copy it, because it received the new ownerID as our userID it will presented in the user recipe page. |
| Present the way to store large quantities of food that the user buys (ways to store for a long time) | For each ingredient that is uploaded by the system or by the user, is given the option to fill up how we are supposed to store large quantities of this product. |
| Present recipes of food that the ingredients are in season | For each ingredient there is a label for which season we can consume this product. We can filter in the recipes page to show only the recipes according to a selected season. We will go over a recipe's ingredient and see if there is a specific ingredient that we cannot buy at a certain type, and therefore we will know how to classify this recipe and present the right recipes. |
| Daily, weekly, monthly and yearly shopping list for all meals (Note that the app will only provide the groceries list, but the user won't be able to purchase the items). | When the user decides to add to the cart a certain choice. All the data will be stored in the database for the products the user chose. After that we can go over each ID and see the ingredients/recipes/ the user choice and calculate accordingly. For the ingredient list we will go over all the recipes and quantities, double it accordingly and provide mutual groceries list for everything. Note that the user will be given a list only and cannot purchase from the application |
| For each recipe, presenting the calories intake, amount of protein/fat/carbs | For each recipe there is Nutritious table that contains the data such as protein, fat, carbs more. when the user chooses to upload a recipe there is an option to fill up this kind of information, and from the build-in ones it will be provided as well. |
| Search for recipe in the data base using key words and presenting the matching results | There is an option to search for a recipe using key words. Note that this kind of search is by the recipe name. we will extract all the recipes that contains a certain word. |
| Provide the option to share recipe between users. (There will be a page name "Shared recipe" that will present all the recipes that were shared by the users and not the default recipes provided by the app). | In the database when we upload ingredient or a recipe, there is a column OwnerID (if it was uploaded by the system or by a user(and than we store the userID)), the user is able to share the ingredient/recipe using a checkbox accordingly (wants/ doesn’t want). If the user chooses she/he wants than we check another column in the table as true(ShareRecipe/Ingredient) and we are able to filter for other users according to that column, if its false we wont present, if it's true we will. |
| Provide the option to create an event and add meals for further organization (for example upcoming birthday) | There is an option to "Create Event", we will direct the user to a page where the user can list recipes/create recipes for that current event. We will store all the information in the database , the ID of the recipe and how much the user want from it. |
| For each ingredient that is in the app, there will be an option to receive all the recipes that include that ingredient | There will be an option to filter up recipes in the recipes page according to ingredient, we will go over each recipe, if that ingredient is in their list we will present it to the user, if not, we wouldn’t. |
| For each ingredient that is in the app, there will be an option to receive all the information about it, cost, how to store it (optional – carbs, protein etc.) | There is an information page for each ingredient. In the ingredient page there will be an option to click "information" and we will trace that ingredient using the ingredientID and extract the needed information in order to present it to the user. |

### HUMAN INTERFACE DESIGN

## Overview of User Interface

Listed in the PowerPoint "User interface"- For each action there is a comment that if you click it there is a description.

## Screen Images

Listed in the PowerPoint "User interface".

## Screen Objects and Actions

Listed in the PowerPoint "User interface"- For each action there is a comment that if you click it there is a description.