

NourishNexus

Anton Marius Breinholt & Adrian Bay Dorph

antbr@itu.dk & addo@itu.dk

BIBAPRO1PE

Bachelorprojekt

Advisor: Elda Paja

Submitted: 15. May 2023

IT UNIVERSITY OF COPENHAGEN

Contents

1	Abstract	1
2	Preface and Introduction	2
3	Analysis of Existing Meal Planning Applications and Proposed Solution	3
3.1	State of the Art Analysis	3
3.2	Proposed Solution	4
3.2.1	Functional Requirements	4
3.2.2	Non-Functional Requirements	5
3.3	Methodology	5
4	User's guide and examples	7
4.1	Introduction	7
4.2	User Guide	7
4.2.1	Registering	7
4.2.2	Setting Nutrient Targets	8
4.2.3	Creating Recipes	10
4.2.4	Saving Recipes from the Community	11
4.2.5	Generating Meal Plans	12
4.2.6	Tracking Your Food Intake on the Day Page	12
4.2.7	Logging in as a Returning User	12
4.2.8	Additional Features of the System	12
5	Technical description of the program	14
5.1	Architecture	14
5.1.1	Backend Architecture	14
5.1.2	Frontend Architecture	16
5.1.3	Database Architecture	17
5.2	Implementation	18
5.2.1	Nutrition Target Algorithm	18

Contents	iii
5.2.2 Meal Planning Algorithm	21
5.3 Unit Testing	26
5.4 Populating the Database	27
5.5 Running the Program	28
6 Test	29
6.1 Introduction	29
6.2 Usability Testing Approach	29
6.3 Test Scenarios	30
6.4 Test Results	31
6.5 Changes Based on Test Results	32
7 Limitations of Our Work	34
8 Conclusion	35
Bibliography	38
9 Appendices	40
9.1 NourishNexus Wireframe	40
9.2 NourishNexus Mockup	41
9.3 User Tests	42

1 Abstract

Studies have shown meal planning as an important part of achieving a healthy lifestyle [1], but it can be a difficult task to create and follow meal plans that meet all of one's nutritional needs. In this bachelor project, we developed a meal-planning web application that helps users plan meals based on their nutritional targets and discover new recipes.

To develop the application, we followed an Agile work method with the Azure DevOps project management framework. We conducted user interviews and think-aloud usability testing to design the user interface.

The system requirements were defined based on the Problem analysis, State of the Art analysis, and what we as developers wanted to include. We implemented the application using Blazor and C# in ASP.NET and deployed it with Digital Ocean. We tested the application using various types of testing such as unit testing and usability testing to ensure its quality.

The final web application allows users to find and explore recipes, generate weekly meal plans that suit their nutritional targets, and customize their daily nutritional targets, either by getting a personalized recommendation for nutrient targets or by setting them manually.

The application also provides nutrition information for each recipe and meal plan, making it easier for users to maintain a balanced diet. Based on our usability testing, the web app was easy to navigate and use and provided the users with the necessary features, though long-term testing would need to be conducted to monitor whether following NourishNexus' meal plans would improve the user's physical health.

2 Preface and Introduction

In this report, we will discuss the development of NourishNexus, a personalized meal-planning web app. This bachelor project was conducted at the IT University of Copenhagen, under the guidance of our supervisor Elda Paja. We're grateful to our test users for their valuable feedback on the app.

In our modern, busy world, the problem of meal planning has grown larger [1]. This problem is further compounded by the limited repertoire of recipes that many people possess. For instance, a study conducted in the UK revealed that 25% of citizens have only three recipes in their repertoire [2]. Similarly, in Denmark, 47% of people decide what to eat on the same day, leading to unvaried meals and thus unhealthy dietary practices [3].

Unhealthy diets are a leading cause of chronic diseases such as heart disease, diabetes, and cancer [4]. In fact, unhealthy diets cause more deaths globally than smoking [5]. Studies show that meal planning has been associated with a healthier diet and less obesity [1]. By planning their meals, individuals can ensure that they are consuming both a balanced and nutritious diet, which can reduce the risk of developing chronic illnesses such as obesity. As meal planning can be a challenging task, we researched the existing options available (see Chapter 3).

Using our research we have developed a C# web application to provide a solution to the issue of meal planning. Our web app offers a personalized weekly meal plan that is tailored to the user's nutritional intake targets and the possibility of expanding the user's repertoire by enabling users to create and share their recipes with each other.

The report will address existing meal planning applications and our proposed solution, the user's guide and examples, a technical description of the program, our testing strategy and results, the limitations of the project, a conclusion, and future works.

3 Analysis of Existing Meal Planning Applications and Proposed Solution

3.1 State of the Art Analysis

In our research, we found that there were numerous options available[6], ranging from basic meal planners to more complex systems that incorporate advanced features such as food tracking and recipe recommendations. However, most of these apps specialize in select functions and do not consider all of the factors that our application intends to address, including how recently a recipe was used and nutritional targets.

We identified a close competitor to our intended application in PlateJoy[7], which offers custom meal plans based on a variety of data points, such as dietary preferences, food allergies, and weight loss goals. The app also provides a grocery list feature, which allows users to easily purchase the ingredients needed for their planned meals. However, PlateJoy does not offer meal recommendations based on the recency of dishes or nutritional balance, which are key features of our intended application.

Another competitor we identified was MealPrepPro[8], which recommends meals based on a weekly calorie intake goal and provides a recipe database where users can search for new recipes. The app also allows users to add food to their meals in the system and track their progress toward their goals. MealPrepPro does track, calories, and macronutrients, but not micronutrients, like our intended web app.

Lifesum was also among the competitors we identified [9]. This platform provides the user with the ability to find a diet that suits their lifestyle and food preferences. Then a 7-21 days meal plan based on that diet can be chosen. This application is fairly close to our intended one, but it focuses on meal plans adhering to diets, rather than to nutritional targets.

Another competitor we identified was Paprika.[10] Paprika enables the users to create their recipes and insert them into a manually created meal plan. Paprika also offers a feature for grocery lists.

We also discovered that Yummly was a competing company.[11] Yummly offers meal planning features with recipe suggestions based on personal preferences, such as dislikes, allergies, and diets. Yummly also allows users to create their own recipes and has a shopping list feature.

It is worth noting that meal planning is a highly competitive field with lots of solutions, but the above are what we found to be the closest to our intended web application.

3.2 Proposed Solution

NourishNexus is our proposed solution to the challenges of meal planning and healthy eating. There are multiple benefits of using the web app. Firstly, NourishNexus provides users with a convenient way to expand their recipe repertoire, with a community page for recipe sharing, and the ability to customize meal plans based on personal preferences and dietary restrictions. Secondly, our app offers guidance on healthy eating by tailoring meal plans to the recommended nutritional intake of each user.

The requirements for the web application (listed below), were defined based on our research during the Problem analysis, State of the Art analysis, and what we as developers wanted to include in the application.

3.2.1 Functional Requirements

- FR1** The app should allow users to create an account and log in.
- FR2** The web app will allow users to easily set their daily nutritional preferences.
- FR3** The app must be able to generate a personalized weekly meal plan for the user based on their nutritional preferences.
- FR4** The web app should allow users to search and browse a library of community-made recipes.

FR5 The app must allow users to create and share their recipes with the community.

FR6 The app should allow users to switch out recipes and food items in their generated meal plan

3.2.2 Non-Functional Requirements

NFR1 The application will be responsive and should load all pages within 3 seconds on desktop and mobile devices.

NFR2 The web app must be secure and protect user data with encryption and authentication.

NFR3 The app shall be scalable to handle a large number of users and recipes.

NFR4 The app must be maintainable and upgradable to support new features.

NFR5 The app should be containerized with Docker.

NFR6 The web app should be deployed on DigitalOcean.

NFR7 The app should be designed as a Mobile-First application.

3.3 Methodology

To manage our project progress, we used Azure DevOps, a project management tool that is based on Scrum. Using the tool, we organized our development work into sprints, managing a backlog of features and issues, and tracking progress through burndown charts and sprint velocity.

We also utilized a range of other tools and technologies to support our development process. For example, we used drawio to create the diagrams in this report. We also used Figma to create wireframes (see Appendix section 9.1) and mockups (see Appendix section 9.2) of our web app's UI, which helped us when developing the frontend.

Our backend was developed in C# using Microsoft's .NET framework, while the frontend was developed using Blazor WebAssembly. For our database, we used a Microsoft SQL Docker image, which we ran in a Docker container.

To manage our codebase and version control, we used GitHub. We also utilized GitHub Actions to automate parts of our CI/CD pipeline. This enabled us to execute our tests whenever code was pushed to the main branch, ensuring that our code was always thoroughly tested and of high quality.

We hosted our application on a Droplet from DigitalOcean that was running Ubuntu 18.04. To run our application, we first installed .NET and Docker on the Droplet. Then we followed the instructions described in our GitHub repository, but we used the nohup command to ensure that the command would not be canceled when exiting the terminal.

We also purchased a domain name, nourishnexus.dk, and routed it to our IP. To provide secure browsing for our users, we set up HTTPS using Certbot, Nginx, and Let's Encrypt. Finally, we configured a firewall to enforce the HTTPS policy.

4 User's guide and examples

4.1 Introduction

In this chapter, we will guide you through the pages of our application including their functionality and features. You will for instance be shown how to create recipes, set your nutrient targets, and generate a meal plan.

4.2 User Guide

4.2.1 Registering

To register for NourishNexus, go to our website at <https://nourishnexus.dk>. Click on the text that says "Click here to register!", which will redirect you to the Register page. (Figure 4.1)

Enter your email, username, password, and confirm your password and click on the "Register" button to sign up. This will log you into the system and redirect you to our "Nutrient Targets" page. (Figure 4.2)

Register

Email

Nickname

Password

Confirm Password

Register

Already signed up? [Click here to sign in!](#)

Figure 4.1: Register page

Welcome!

Get started by following the steps below

1. Set your daily nutritional targets on this page with the button below. Afterward you can edit the individual fields, if you have a different preference than the recommended.
2. [Create recipes](#) or go to the [community page](#) to save all the recipes you want to see in your meal plans!
3. Once you have saved or created enough recipes and set your nutritional targets you can generate your meal plan in the [week page!](#)

Set to recommended

Nutrient	Amount

Figure 4.2: NutrientTargets page

4.2.2 Setting Nutrient Targets

You are now on the "Nutrient Targets" page. When you first visit the page there will be a quick start guide for creating a meal plan. The guide consists of the following steps.

1. Set your daily nutritional targets on this page with the button below. Afterward, you can edit the individual fields if you have a different preference than the recommended one.

2. Create recipes or go to the community page to save all the recipes you want to see in your meal plans.
3. Once you have saved enough recipes and set your nutritional targets, you can generate your meal plan on the "Week" page!

We recommend most users press the button, which will open up a modal (Figure 4.3) where you can input your age, gender, height, weight, physical activity level, and weight goal. Clicking "Submit" will close the modal and generate your recommended values.

10.41

nourishnexus.dk/

Calculate nutritional targets

Age: 42

Gender: Male

Weight (kg): 74

Height (cm): 179

Physical Activity Level: Lightly Active

Weight Goal: Maintain weight

Submit Cancel

Figure 4.3: NutrientTargets modal

10.41

nourishnexus.dk/

NourishNexus

Create Recipe

Title

Categories

Asian Children meal
 easy Fish
 Fjerkrae Fruit
 Italian Japanese
 Meat Pasta Pork
 Vegan Vegetarian

Add new

Figure 4.4: CreateRecipe page

4.2.3 Creating Recipes

To create a recipe, navigate to the "Create Recipe" page (Figure 4.4) and fill out the following information:

1. Recipe name
2. Relevant categories
3. Description
4. Ingredients using the "Add Ingredient" button
5. Method
6. Relevant meal types such as "Breakfast" or "Dinner"
7. Recipe visibility (public or private)

The "Add Ingredient" button will take you to a page where you can search for food items (Figure 4.5). From there, click on a food item to add it along with the amount in grams (Figure 4.6). When you are done, click the "Create" button. This will send you to the "Saved Recipes" page. (Figure 4.7)

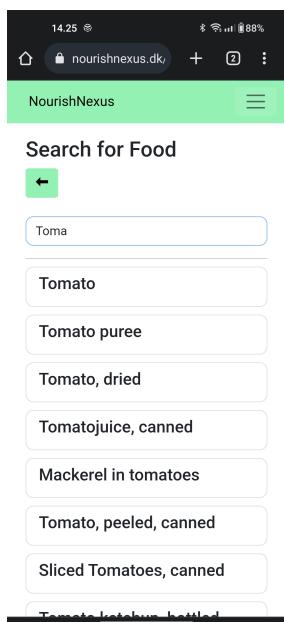


Figure 4.5: Search for food page

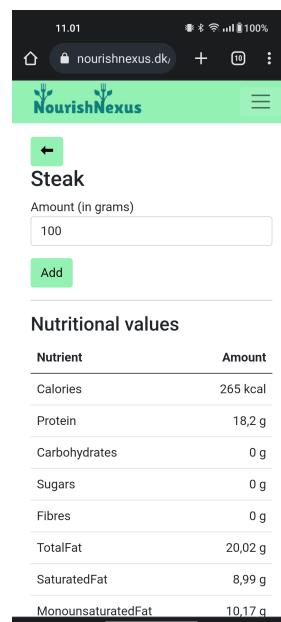


Figure 4.6: Add food page

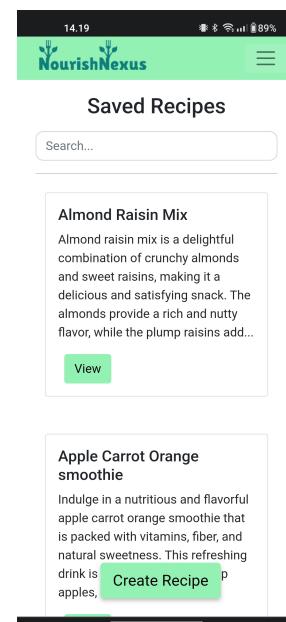


Figure 4.7: Saved recipes page

4.2.4 Saving Recipes from the Community

To save a recipe from the community, go to the "Community Recipes" page (Figure 4.8) and find a recipe that you'd like to save. Click on the "Save recipe" button. It will now appear on the "Saved Recipes" page and can be used to generate meal plans.

Community Recipes

Save all official recipes

Search...

Almond Raisin Mix
Almond raisin mix is a delightful combination of crunchy almonds and sweet raisins, making it a delicious and satisfying snack. The almonds provide a rich and nutty flavor, while the plump raisins add...

View **Save recipe**

Apple Carrot Orange smoothie
Indulge in a nutritious and flavorful apple carrot orange smoothie that is packed with vitamins, fiber, and natural sweetness. This refreshing...

Figure 4.8: Community recipes page

11.21 ⓘ 4G 53%

nourishnexus.dk/ + 13 :

NourishNexus

Week 17 17/4 - 23/4

Generate meal plan

Daily calorie goal 2539,25 kcal

Total calories 2315,25 kcal

Breakfast	Eggs and Bacon
Lunch	Fish fillets
Dinner	Pasta Putanesca
Snacks	Apple Carrot Orange smoo...

Monday ➔

Total calories 2315,25 kcal

Breakfast	Fruit bowl
Lunch	Sushi: Maki and Nigiri🍣
Dinner	Torskeladen
Snacks	Strawberry Banana Smoot...

Tuesday ➔

Figure 4.9: Week page

4.2.5 Generating Meal Plans

Once you have set your nutrient targets and saved or created enough recipes, you can generate your meal plan on the "Week" page (Figure 4.9). Simply select the week that you'd like to generate a plan for, and then click on the "Generate Meal Plan" button. After a brief loading period, the meal plan for the week will be displayed, and you can click on the individual days to see the recommended meals.

4.2.6 Tracking Your Food Intake on the Day Page

Now that you have generated your meal plan, it is time to head to the "Day Page" (Figure 4.10) to view the recipes that the meal plan has selected for you (Figure 4.11). If a particular recommendation does not suit you, you can swap it out for a recipe or food item that you would like to eat instead.

4.2.7 Logging in as a Returning User

When returning to the website, you may find that you have been logged out. In that case, you will be sent to the login screen (Figure 4.12), where you can log in with the email and password you used when registering.

4.2.8 Additional Features of the System

Additional features include updating your credentials on the Account page(Figure 4.14), as well as updating, and deleting your recipes from the "Saved Recipes" page(Figure 4.13). This functionality will not be described in detail here, as it is not part of the core user flow.

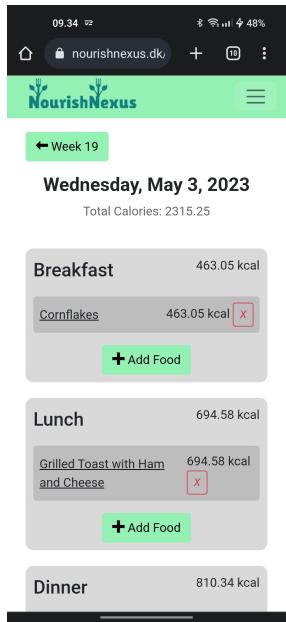


Figure 4.10: Day page

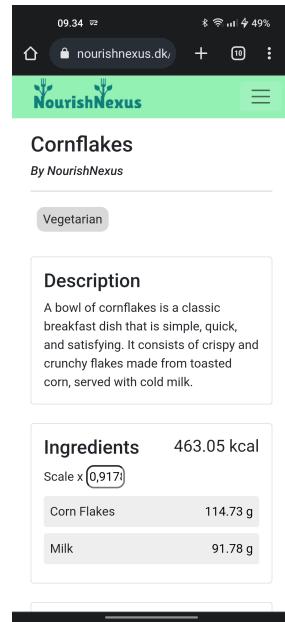


Figure 4.11: View Recipe page

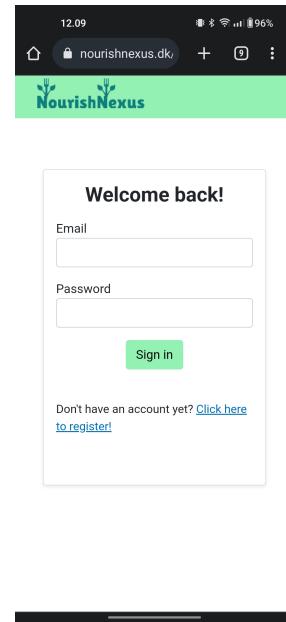


Figure 4.12: Login page

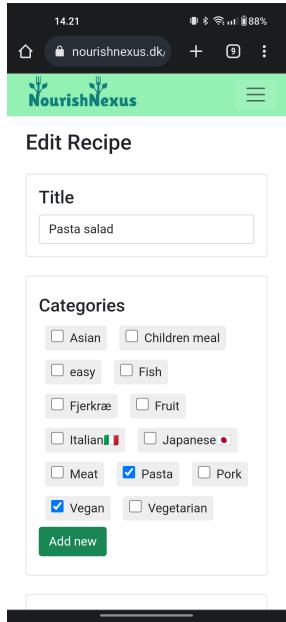


Figure 4.13: Edit recipe page

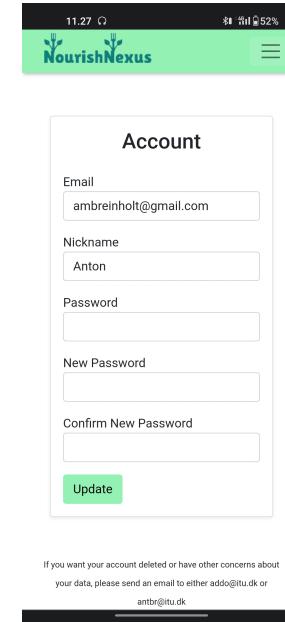


Figure 4.14: Account page

5 Technical description of the program

5.1 Architecture

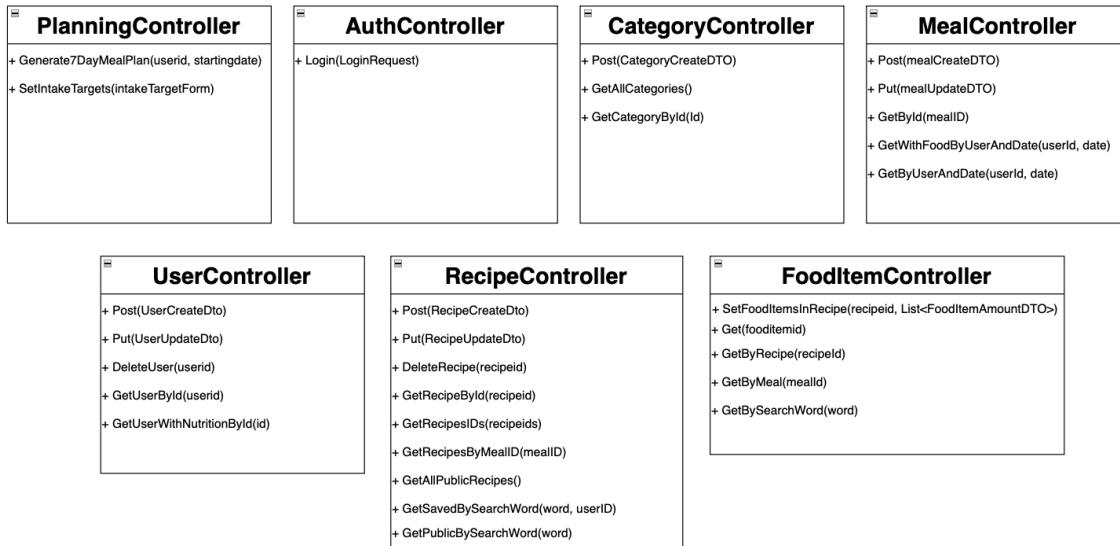
The system is comprised of a frontend, a backend, and a database.

5.1.1 Backend Architecture

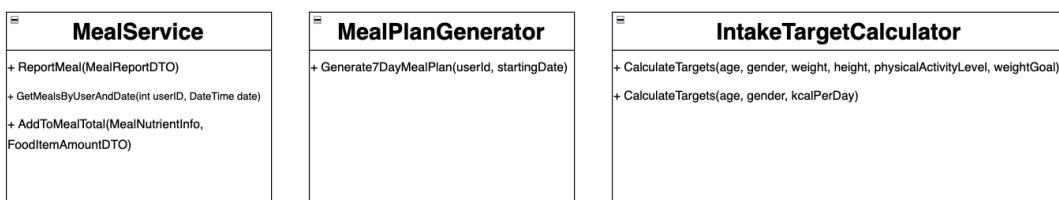
Our backend architecture follows the repository pattern, which centralizes data access code in repositories with CRUD methods to avoid duplication in controllers. See Figure 5.1 (Repositories). This decouples application and infrastructure layers, enabling independent changes for better scalability and maintenance. The pattern provides a level of abstraction that simplifies data manipulation and enhances testability through in-memory mocks instead of database queries.

Another benefit of the pattern is that it allows us to utilize interfaces for our repositories, hiding the implementation details, and enabling the use of dependency injection to inject repository interfaces into our application layer (See Figure 5.1 Services, Controllers). The interfaces act as contracts for the classes implementing them, which makes it more manageable to implement the repository, and additionally allows for easier swapping of the implementation. When reflecting upon the drawbacks of the pattern, the many layers of abstraction can require a lot of time and effort during development, but as we have implemented it before in other projects, it was not something that was of great concern to us.

Presentation Layer (Controllers)



Business Logic Layer (Services)



Data Access Layer (Repositories)

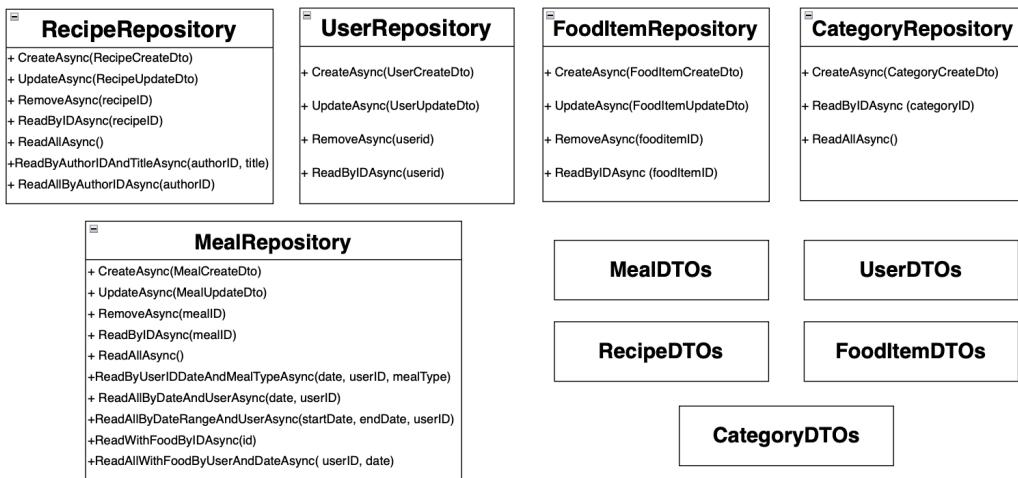


Figure 5.1: System Architecture Diagram of the Main Components of the System

5.1.2 Frontend Architecture

Our frontend architecture follows the Razor pages pattern. Each Razor page includes markup syntax and C# code. To communicate with our server-side API, we have defined client services to make HTTP calls to our controllers. We then use dependency injection to inject these services, through their respective interfaces, into our Razor pages. We have also injected Data Transfer Object (DTO)'s into our client project, as we need them to store data we utilize when calling our service methods for creating and updating entities.

To provide an example, let's take a look at our RecipeService. The CreateRecipe method uses a RecipeCreateDTO object to make a POST request to the Recipe controller endpoint on the server. As we can see in the CreateRecipe.razor code, the IRecipeService as well as the DTO namespace gets injected, then the DTOs' fields get bound to the user input. When creating the recipe, the CreateRecipe method gets called, with the DTO.

Here's the code snippet from the IRecipeService.cs file:

```

1 .....
2 Task<HttpResponseMessage> CreateRecipe(RecipeCreateDTO recipe);
3 .....

```

Here's the corresponding code from the RecipeService.cs file:

```

1 .....
2 public async Task<HttpResponseMessage> CreateRecipe(RecipeCreateDTO
    recipe)
    => await _http.PostAsJsonAsync("api/Recipe/", recipe);
4 .....

```

And here we have a code snippet from the CreateRecipe.razor file.

```

1 @using server.Core.EF.DTO;
2 @inject IRecipeService RecipeService
3 .....
4
5 <InputTextArea class="form-control" id="method" @bind-Value="@recipe
    .Method" rows="8" />
6 .....
7 private RecipeCreateDTO recipe = new();
8 .....
9 async Task HandleCreateRecipe()
10 {
    .....

```

```

12     var response = await RecipeService.CreateRecipe(recipe);
13     .....

```

5.1.3 Database Architecture

The purpose of our SQL database is to store and manage information related to the 5 entities of our system.

The User entity contains information about registered users of the system, including their username, email address, passwordhash, passwordsalt, and all the user's nutritional targets.

The Recipe entity contains information about individual recipes, including the recipe name, ingredients, etc.

The Meal entity contains information about meals, including the meal name, the recipes and food items included in the meal, and the meal type (i.e. breakfast, lunch, dinner, snack).

The FoodItem entity contains information about individual food items, including their name, and nutritional information.

The Category entity contains a category name.

To help illustrate the relationships between these entities, we have included an Entity Relation (ER) diagram (Figure 5.1). The diagram shows that a User can create many Recipes, a Meal can include many Recipes, and a Recipe can be used in many Meals. Additionally, a FoodItem can be used in many Recipes, and a Recipe can include many FoodItems. Furthermore, a Recipe can belong to many Categories, and a Category can have many Recipes.

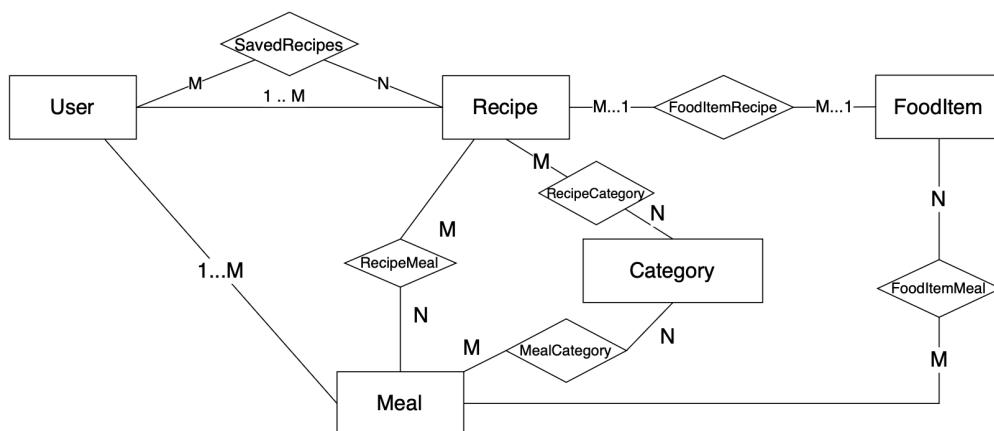


Figure 5.2: ER Diagram of the database

5.2 Implementation

5.2.1 Nutrition Target Algorithm

To enable the user, to quickly set their daily nutritional targets, we have created an algorithm that can give the recommended daily nutritional intake targets, adhering to the Nordic Nutrition Recommendations (NNR) [12]. In the system, there are 3 variables for each nutrient, the Lower Bound (LB), Ideal Intake (II), and Upper Bound (UB). The LB is the minimum amount of a given nutrient that the user can eat per day on average. The II is the ideal amount that the user should eat of the given nutrient per day on average. Finally, the UB is the maximum amount that the user can eat per day on average for the given nutrient.

To be able to determine what the nutritional recommendations are for the user, 3 parameters are needed. That is gender, age, and daily caloric intake. However, not everyone is aware of the appropriate daily caloric intake for their individual needs and therefore, we decided to implement the algorithm such that it also calculates the recommended daily caloric intake, based on age, gender, weight, height, physical activity level, and the user's weight goals.

- Lose weight fast
1000 kcal deficit per day, corresponding to losing 1 kg per week. [13]
- Lose weight
500 kcal deficit per day, corresponding to losing 0.5 kg per week. [13]
- Maintain weight
No deficit or surplus.
- Gain weight
500 kcal surplus per day, corresponding to gaining 0.5 kg per week. [14]
- Gain weight fast
1000 kcal surplus per day, corresponding to gaining 1 kg per week. [14]

How Does it Work?

The algorithm starts by calculating the recommended daily caloric intake. To do that, the Basal Metabolic Rate (BMR) is calculated using the following formulae [15]:

- For females:
$$\text{BMR (kcal / day)} = 10 * \text{weight (kg)} + 6.25 * \text{height (cm)} - 5 * \text{age (y)} - 161$$

- For males:

$$\text{BMR (kcal / day)} = 10 * \text{weight (kg)} + 6.25 * \text{height (cm)} - 5 * \text{age (y)} + 5$$

Once the BMR has been calculated, the Total Daily Energy Expenditure (TDEE) is calculated by multiplying the BMR by the physical activity level. Lastly, the TDEE is adjusted with regard to the weight goal, and the result is the recommended daily caloric intake.

Now that the daily caloric intake has been calculated, it can be used to determine the LB, II, and UB for all of the different nutrients adhering to the NNR [12]. Each nutrients LB, II, and UB are calculated using their function as can be seen in the example with thiamin in Figure 5.3.

Thiamin mg/d	Women	Men	Children		
			2-5 y	6-9 y	10-13 y girls/boys
Recommended intake	RI	1.1	1.4	0.6	0.9
Average requirement	AR	0.9	1.2		
Lower intake level	LI	0.5*	0.6*		
Upper intake level	UL	- **	- **		

* 0.8 mg at energy intakes <8 MJ/d and 1.0 mg/d for elderly.

** Not established.

Figure 5.3: Nordic Nutrition Recommendations' Thiamin Recommendations [12, p. 407]

```

1 public (float, float, float) CalculateThiamin(int age, Gender gender
    , float kcalPerDay)
2 {
3     float LB;
4     float II;
5     float UB = float.MaxValue;
6
7     if (age <= 5)
8     {
9         LB = 0f;
10        II = 0.6f;
11    }
12    else if (age >= 6 && age <= 9)
13    {

```

```

14         LB = 0f;
15         II = 0.9f;
16     }
17     else if (age >= 10 && age <= 13)
18     {
19         LB = 0f;
20         II = gender == Gender.Female ? 1.0f : 1.2f;
21     }
22 else
23 {
24     II = gender == Gender.Female ? 1.1f : 1.4f;
25     if (age >= 65)
26     {
27         LB = 1.0f;
28     }
29     else if (kcalPerDay < (8 / MJPerKcal))
30     {
31         LB = 0.8f;
32     }
33     else
34     {
35         LB = gender == Gender.Female ? 0.5f : 0.6f;
36     }
37 }
38
39     return (LB, II, UB);
40 }
```

For all nutrients, the II is always set to the recommended II, but when there are no recommendations for UB or LB, then they are set to the highest possible float value and 0 respectively.

With this algorithm, we have enabled the user to quickly set their personalized nutritional targets for 29 different nutrients and also their daily calorie intake goal, based on recommendations from the NNR [12], by simply providing their, age, gender, weight, height, physical activity level and their weight goals.

Time Complexity

The time complexity for calculating nutrient targets with this algorithm is O(1), meaning that it is a constant time algorithm. This is because regardless of the inputs, the algorithm only performs comparisons and arithmetic operations which are of constant time complexity.

5.2.2 Meal Planning Algorithm

To be able to make a meal plan for the users, we needed an algorithm for just that. The algorithm needed to be able to make a plan that fits the user's nutritional needs, that is to stay within the LB and UB for each nutrient target on a weekly basis, while trying to meet their ideal intake target. Furthermore, the algorithm needed to avoid suggesting the same recipes over and over again.

At first, we discovered the simplex algorithm, but we realized that it would be too hard to try to implement ourselves and completely understand how it works in the rather short time that we had. Therefore we looked for other options and we found a paper about an algorithm that could give a food plan for a single day. We took some inspiration from an activity diagram of the algorithm, which we developed our algorithm around.[16, p. 36]

Specifically, we were inspired by their way of ensuring that the nutrient intake would be within an acceptable range, by inserting an item from a list of food items into the plan and then checking if the plan fits the user's needs. If it does fit the user's needs, then the plan is successful otherwise the food with the most nutrients that do not suit the user's needs is replaced. This continues until all the nutrients are within the acceptable range.

With this approach, we developed our algorithm that makes a 7-day meal plan with 4 meals each day: Breakfast, Lunch, Dinner, and Snacks. An Activity diagram of our algorithm can be seen in Figure 5.4.

Checking for User Prerequisites (Steps 1 and 2)

The algorithm takes 2 parameters; the user id and the starting date. First off, the user is found and it checks if the user has the prerequisites to be able to generate a meal plan. These prerequisites include having saved enough recipes since the algorithm utilizes the user's saved recipes to generate a meal plan. This is to make sure that the user has accepted the recipes that are in the meal plan, and therefore the user will not get a meal plan that contains recipes that they do not like or cannot tolerate e.g. due to allergies. The amount of recipes the user needs to have saved for the algorithm to try to make a plan is 28 (7 for breakfast, 7 for lunch, 7 for dinner, 7 for snacks).

The second prerequisite is that the user needs to set their nutrient targets. If the prerequisites are not met, the algorithm will cancel the planning. Otherwise, it continues by calculating the ideal weekly intakes for all the nutrients.

Calculating the Ideal Intake Based on Previous Weeks (Step 3)

The user has already set their ideal intake, but the algorithm here takes into account what the user ate in the past 2 weeks, to even out the intake of the different nutrients. E.g. if they ate more thiamin distributed over the last 2 weeks than their ideal intake goal, the algorithm will try to balance this out in the current week's meal plan.

This is done by taking the ideal intake for 3 weeks and subtracting the sum of the past 2 weeks' intakes. This leaves the current week's ideal intake, but we also ensure that this amount lies within the LB and UB with a simple if statement. This is done for each of the 29 nutrients.

Ordering the List of Recipes (Step 4)

The next step is to arrange a list of the user's saved recipes. Since the algorithm later will take the recipes from the top of the list, we decided to order them by recency to make sure that the plan will try to use recipes that the user did not eat recently. It starts from the day before the starting date and goes through the meals of the day and if it finds a recipe, it will be added to the list. This continues 30 days back in time. All the remaining saved recipes that did not occur in the past 30 days are then added to the list after which the list gets reversed, such that the most recently cooked recipe is the last in the list.

Shuffling the First Half of the Recipe List (Step 5)

At this point, we have what we need to create an initial meal plan. To add an element of randomness to the meal plan we decided to shuffle the first half of the list of recipes, in order to avoid repeating the same pattern of meal plans over a longer time.

Creating an Initial Meal Plan (Steps 6 and 7)

The meal plan is then generated by creating an initial meal plan. Here it firstly adds all the food items and recipes that the user has already registered in the week where they are generating a meal plan. These meals are marked as locked and there will not be added anything to those meals, however, the contents of the meals will still be used to calculate the sum of nutrients. This enables the user to choose for themselves what they want to eat in a specific meal on a specific day and the meal plan will include this.

After inserting what the user already chose to eat, the algorithm continues by inserting the first recipes in the list of recipes in the plan using a method called *Fin-*

dAndRemoveRecipe which finds the first recipe in the list that is allowed to be used in a meal with the same MealType as the meal it is being inserted into. E.g. if the method is used with a meal that is breakfast, the method will find the first recipe that is allowed to be used in breakfast. That first recipe gets returned after being removed from the list of recipes. After that, the recipes that were found and removed from the recipe list get added to the meal. If the meals in the plan cannot be filled out with the recipes in the list, a plan cannot be made and it cancels the plan.

Adjusting the Plan to Hit the User's Nutritional Targets (Steps 8 - 14)

Now that an initial meal plan has been made, it needs to adhere to all the users' nutrient targets. To do this, the plan gets adjusted by replacing one breakfast recipe, one lunch recipe, one dinner recipe, and one snack at a time.

First, the plan is adjusted to meet or exceed the ideal intake for at least all nutrients except 3. This can be seen in steps 8 - 10 in the activity diagram in Figure 5.4. We decided to let it continue to step 11, while still having 3 nutrients that do not fully meet the ideal intake because it can be difficult to achieve the ideal intake target for all nutrients simultaneously, but all nutrients will be within the LB and UB in the end if the plan is successful. If this condition is not met, it replaces recipes until it is met. The recipes that get replaced are the recipes that have the most nutrients that subceeds the ideal intake for that single meal.

Once the meal plan meets the condition, it goes on to do the same thing in steps 11 - 14, but this time all the nutrients have to subceed their UB, and after one iteration of recipes that have the most nutrients that exceed the UB have been replaced, it checks that the lower bounds are still exceeded. If not it replaces the recipes that have the most nutrients that subceeds the lower bound. Once all the nutrients are in the range between the LB and UB the plan is successful and it is added to the user's meals.

Trying Again if the Plan Fails to Have All Nutrients Within Their Upper and Lower Bounds (Step 15)

There is a possibility that the list of recipes can be used up before the plan could be adjusted to completely suit all of the user's nutrient targets. In this case, the algorithm tries again by going back to the point where the first half of the recipes were shuffled, which enables the possibility that a successful plan can still be generated although the previous one did not succeed fully. This could go on for 50 times before it gives up and uses the plan anyway while telling that the plan has some nutrient targets that it doesn't live up to.

Inserting the Generated Meal Plan

When the plan has been generated it gets inserted into the user's meals on the week where the user decided to generate their meal plan.

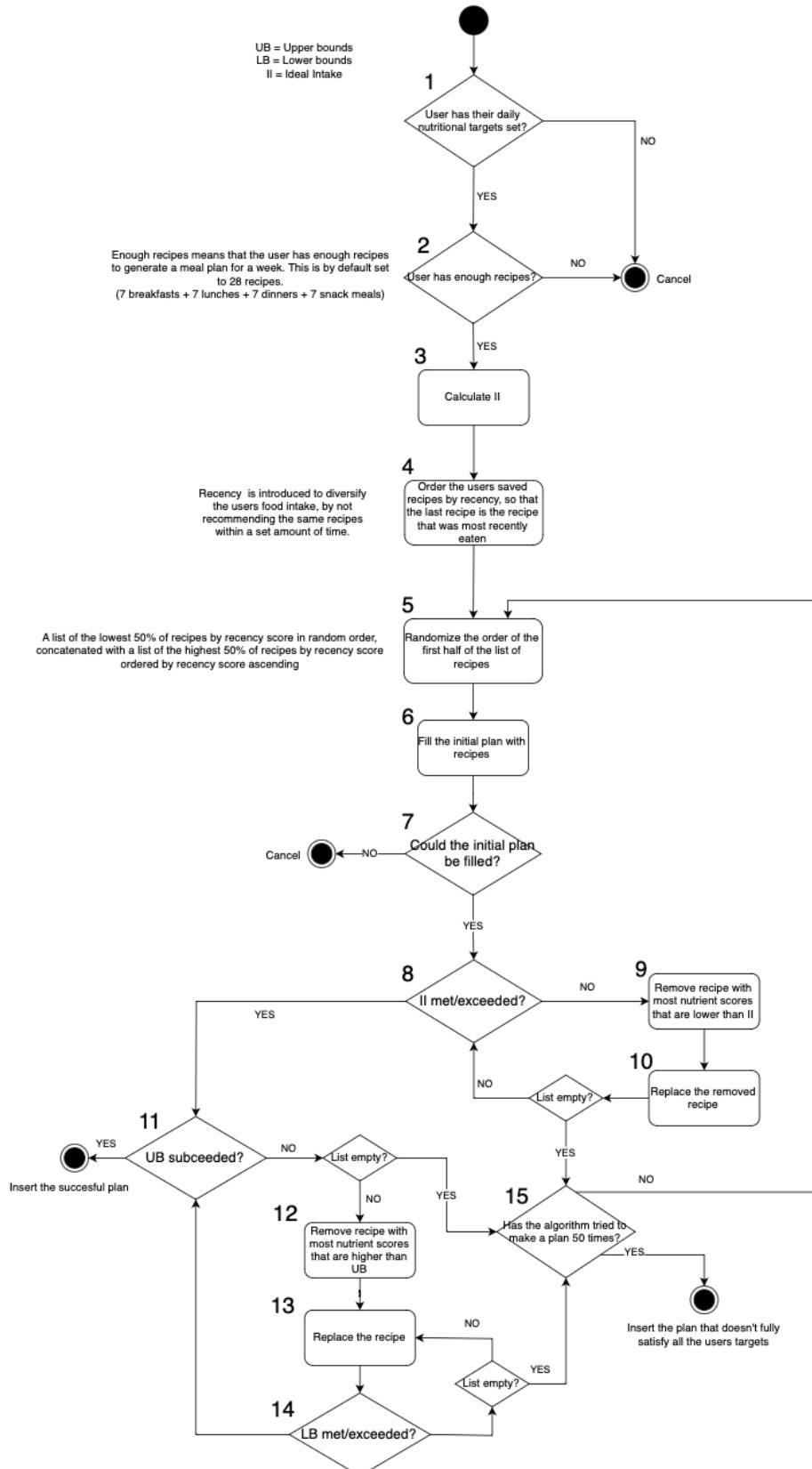


Figure 5.4: Activity Diagram of the Meal Planning Algorithm

Time Complexity

The time complexity for our implementation of the meal planning algorithm is $O(N^2)$ in the theoretical worst case, where N is the number of recipes that the user has saved. This is due to the fact that the algorithm, from steps 11 to 14 in the activity diagram in Figure 5.4, has a list of the user's saved recipes, where every time it has to replace recipes in the meal plan, which in the worst case is N times, it uses the RemoveAt, method to remove the recipes it has used to replace recipes in the meal plan, and RemoveAt runs in $O(N)$, thus the algorithm's time complexity is quadratic in the number of recipes the user has saved, in the worst case.

However, one could argue that a user would most likely not save more than 400 recipes for example. The number of FoodItems and Recipes in the system would be significantly higher in the case of scaling the system, where the number of saved recipes per user would remain the same. So in practice, if we consider the number of recipes that the user has saved would remain constant, the algorithm's time complexity is $O(N + M)$ according to our calculations, where N is the number of Recipes in the system, and M is the number of FoodItems, in the system. When we look at the Activity Diagram, the amount of FoodItems and Recipes in the system does not seem to affect the time complexity, but the way that we have mapped FoodItems to Recipes and FoodItems and Recipes to Meals means that every time we try to get the FoodItems or Recipes in a meal it takes $O(N)$ where N is the amount of FoodItems or Recipes in the system, respectively.

5.3 Unit Testing

Throughout the development of the web app, we have utilized unit testing in the backend part of the system, to ensure the quality of our code. During the project, we have mostly been testing in a Test Driven Development (TDD) fashion, by writing the tests first and then making them pass, although with a few exceptions.

When developing the nutrition target algorithm, TDD was utilized thoroughly, using Xunit's Theory andInlineData feature which allowed us to quickly test with multiple different parameter cases, which can be seen in the following code:

```

1 [Theory]
2 [InlineData(25, Gender.Male, 2650, 0.6f, 1.4f, float.MaxValue)]
3 [InlineData(25, Gender.Female, 2650, 0.5f, 1.1f, float.MaxValue)]
4 [InlineData(3, Gender.Male, 1500, 0f, 0.6f, float.MaxValue)]
5 [InlineData(7, Gender.Male, 1700, 0f, 0.9f, float.MaxValue)]
6 [InlineData(11, Gender.Male, 1800, 0f, 1.2f, float.MaxValue)]
7 [InlineData(11, Gender.Female, 1800, 0f, 1.0f, float.MaxValue)]
8 [InlineData(75, Gender.Male, 2000, 1.0f, 1.4f, float.MaxValue)]
9 [InlineData(25, Gender.Male, 1600, 0.8f, 1.4f, float.MaxValue)]
10 void CalculateThiamin(int age, Gender gender, float kcalPerDay,
11     float expectedLB, float expectedII, float expectedUB)
12 {
13     //Act
14     var actual = _intakeTargetCalculator.CalculateThiamin(age,
15         gender, kcalPerDay);
16     //Assert
17     Assert.Equal(expectedLB, actual.Item1, 0.0001);
18     Assert.Equal(expectedII, actual.Item2, 0.0001);
19     Assert.Equal(expectedUB, actual.Item3, 0.0001);
}

```

Here we cover all the cases, for determining the thiamin targets, following the NNR's recommendation which can be seen in Figure 5.3. The same testing practice was used for all the nutrients.

Code Coverage

During the implementation, we mostly followed TDD, but due to time constraints, there are some parts of the program that we decided to test after implementation, to increase the chance of completing the minimum viable product in time, but as a side effect this meant risking a lower the quality of our code. With 298 unit tests, the overall code coverage for the backend part of the system, ended up at 81.81%, with a line coverage of 81.44%, a branch coverage of 69.89%, and a method coverage of 94.11%. This is excluding the auto-generated code in the Migrations folder.

5.4 Populating the Database

We have populated the database with food items using DTU's Frida food item database [17]. We decided to use this specific database for food items, because it has a lot of food items with many different nutrient attributes, which we found to be perfect for our use case. We have also made default/official recipes authored by the user called

"NourishNexus". We have used ChatGPT [18] to generate descriptions and methods for the "NourishNexus" recipes.

5.5 Running the Program

We have deployed the web app on a DigitalOcean droplet with the domain name <https://nourishnexus.dk>. If you would like to run the app locally instead, please refer to the README section "How to run" of our public GitHub repository.

6 Test

6.1 Introduction

After finishing the initial development of NourishNexus, we wanted to make sure that any usability issues, lacking features, or bugs were identified and addressed quickly and effectively. To achieve this, we chose to conduct some usability testing.

6.2 Usability Testing Approach

We employed the think-aloud method to test the app's usability and recruited six participants between the ages of 22 and 54. Our objective was to evaluate the app across a broad age range, and research suggests that think-aloud testing is effective with 5 or more participants.[19] During the testing process, we documented any vocal observations made by the participants as well as any noteworthy observations we made ourselves.

An example of an observation we made is that some users did not realize that creating a category did not automatically check-mark it, which made us realize that there was a disconnect between what the users expected, and the behavior of the application. Therefore, we made the fix to automatically check-mark created categories.

Due to the mobile-first nature of the web app, four of the six tests were performed on mobile devices, while the remaining two were conducted on desktop computers.

6.3 Test Scenarios

We developed tasks for the participants to perform during the tests to cover a range of scenarios that users may face when using the web app. The tasks were given by asking the test participant how they would perform a certain task. The participants were given the following tasks, during the tests:

1. Register as a user
2. Set nutritional targets
3. Create a recipe
4. Save recipes from the community
5. Generate a meal plan
6. Register what you have eaten today

The success criterion was defined as having all test participants successfully perform all of the above tasks. We also asked the participants some interview questions afterwards. (Listed below)

- Would you consider using the application in its current state? If not, what improvements would need to be introduced to get you onboard?
- What was your overall impression of the meal planning web application? Was it easy to use and navigate?
- Can you talk about the features of the web application that you found most useful? Were there any features that you wish were different or more robust?
- How did the web application compare to other meal-planning tools or resources you have used in the past?
- Were there any specific challenges or difficulties you encountered during the tasks? If so, how did you handle them?
- Were there any moments during the think-aloud test where you felt unsure or hesitant about your approach? If so, why?
- Were there any moments during the think-aloud test where you felt the web application did not fully meet your expectations or needs in terms of meal planning and tracking? If so, please elaborate on what was lacking or could be improved.

6.4 Test Results

The results of the usability testing were analyzed based on the participants' verbal feedback and observed behavior during the think-aloud sessions. Overall, the participants were able to complete the tasks assigned to them with minimal difficulty and therefore passed our success criteria. However, several areas for improvement were identified. For brevity, only the problems that were noted by multiple users are listed in the table below, but the full results can be found in Appendix section 9.3

Table 6.1: Issues Identified During Usability Testing

Issue	Amount of participants
NourishNexus was hard to spell when typing in the website URL	6/6
Was annoyed with the process of saving recipes and/or the uninformative error message "You have to save more recipes" when trying to generate a meal plan	6/6
The food items only being in grams was noted as an annoyance	3/6
The "Set to recommended" button on the "Nutrient targets" page was hard to find	2/6
The calorie information on the week page was not as informative as a recipe name would be	2/6
Discovered a bug in the system, where an exception occurs if you try to generate a meal plan for the same week again	2/6

6.5 Changes Based on Test Results

We found the results of the usability testing to be very insightful. We were delighted that the users were able to perform the tasks, but at the same time discovered some lacking and missing features and even a bug in the system.

Based on the feedback from the participants, we have made the following changes to the system:

Create/Edit Recipe Page

- Capitalize category names when adding a new category.
- Automatically check-mark added categories.

Search Food Item Page

- Populate the page with food items before searching, so it does not appear empty.

Week Page

- Fixing the bug when generating a meal plan when all meals in the week are already filled out.
- Replacing kcal for the meals with the names of the first recipes that are added to the meal. If there is no recipe in the meal it will still show kcal.
- Changing the error message when generating the meal plan to be more informative.

Community Page

- Adding a feature to save all the recipes created by NourishNexus at once.
- Changing "To top" to only show up when there are enough recipes that it makes sense.

Saved Recipes Page

- Changing "To top" to only show up when there are enough recipes that it makes sense.

Nutrient Targets Page

- Adding a drop shadow to the "Set to recommended" button to make it more clear that it is a button.

- Make it more explicit that your nutritional values are saved when entering them.
- Add a "Show more" button to hide most of the nutrients, to prevent the user from getting overwhelmed.

Navbar

- Changing the positioning of the "Account" page.

7 Limitations of Our Work

It is worth noting that certain limitations and constraints impacted our ability to fully achieve our project objectives.

Due to the high development workload, and an assumption that performing usability tests on a deployed application would yield more valuable feedback, we decided to conduct usability testing on the application toward the end of the development cycle. Although the testing provided useful feedback, it was orchestrated late in the process, meaning that we only had sufficient time to address a few of the issues discovered in the usability tests.

Furthermore, we recognize that our use of Scrum in Azure DevOps, while helpful in managing our project, may not have been fully optimized for a team of two. Although we utilized features such as planning poker, story points, sprint planning, and backlog management, we did not make use of other features such as sprint retrospective and daily scrum, due to the fact that we were pair programming most of the time and therefore communicating.

In addition, we had intended to containerize the application with Docker. However, due to time constraints, this goal could not be fully achieved.

8 Conclusion

In conclusion, our project aimed to address the problem of meal planning and unhealthy eating habits through the development of NourishNexus, a personalized meal planning web app. Our research identified existing meal planning applications and evaluated their strengths and weaknesses, leading us to develop a new solution that aims to provide users with tailored meal plans to meet their nutritional needs and dietary preferences.

We leveraged technologies such as Blazor and C# in ASP.NET to develop NourishNexus, including the construction of two algorithms designed for setting nutritional targets and generating meal plans that suits the user's nutritional targets. We made use of tools like Git and DigitalOcean for version control and deployment. While we were unable to containerize the application, every other Functional and Non-Functional requirement for the application was fulfilled. Furthermore, the success criterion in the usability testing was fulfilled, as all users performed the tasks successfully.

When it comes to the problem of expanding people's recipe repertoire and promoting healthy eating, we can conclude that the system provides the user with features that address these issues, such as the community page, the personalized nutritional targets feature, and meal planning features. However we have not been able to perform long-term tests of how following a meal plan generated by NourishNexus, affects the user's health and food habits.

Future works

In future iterations of the program, we would like to incorporate the following features, sorted by priority (descending).

- Infrastructure as Code with Docker
- Implement more features based on testing feedback

- Conduct a second round of usability testing
- Perform long-term user tests like previously described
- Implement Categories on meals (recipe needs to have a specified category in order to be able to be added to a specific meal)
- Develop a feature for recurring meals e.g. if you want to eat the same breakfast on some select days of the week
- Develop a "Grocery List" feature

Abbreviations

BMR Basal Metabolic Rate. 18, 19

DTO Data Transfer Object. 16

ER Entity Relation. 17

II Ideal Intake. 18–20

LB Lower Bound. 18–23

NNR Nordic Nutrition Recommendations. 18–20, 27

TDD Test Driven Development. 26, 27

TDEE Total Daily Energy Expenditure. 19

UB Upper Bound. 18–23

Bibliography

- [1] P. Ducrot, C. Mejean, C. Julia, E. Kesse-Guyot, M. Touvier, L. Fezeu, E. Tavéra-García, S. Hercberg, and S. Péneau, "Meal planning is associated with food variety, diet quality and body weight status in a large sample of french adults," *Int. J. Behav. Nutr. Phys. Act.*, vol. 14, no. 1, p. 12, Feb. 2017.
- [2] A. Hall, "One in four british people can only cook three recipes, survey finds," 2018-08-23. [Online]. Available: <https://www.independent.co.uk/extras/lifestyle/british-people-cooking-recipes-kitchen-skills-food-dishes-survey-a8409586.html>
- [3] M. Brandsen, "Danskere går i køkkenet uden opskrifter: Vi kører på rutinen," 2018-10-13. [Online]. Available: <https://www.bt.dk/forbrug/danskere-gaar-i-koekkenet-uden-opskrifter-vi-koerer-paa-rutinen>
- [4] R. Micha, J. Peñalvo, F. Cudhea, F. Imamura, C. Rehm, and D. Mozaffarian, "Association between dietary factors and mortality from heart disease, stroke, and type 2 diabetes in the united states," *JAMA*, vol. 317, no. 9, pp. 912–924, 2017.
- [5] A. Afshin and et al., "Health effects of dietary risks in 195 countries, 1990–2017: a systematic analysis for the global burden of disease study 2017," *The Lancet*, vol. 393, no. 10184, pp. 1958–1972, May 2019. [Online]. Available: [https://doi.org/10.1016/S0140-6736\(19\)30041-8](https://doi.org/10.1016/S0140-6736(19)30041-8)
- [6] D. Centoni, "Best meal-planning apps of 2023," 2022. [Online]. Available: <https://www.thespruceeats.com/best-meal-planning-apps-4766812>
- [7] "Platejoy," <https://platejoy.com>, [Accessed: Apr. 28, 2023].
- [8] MealPrepPro, "Mealpreppro: Meal planning made simple," <https://www.mealpreppro.com/>, accessed: Apr. 28, 2023.

- [9] Lifesum, "Lifesum - health app for weight loss and food tracking," <https://lifesum.com/da/>, [Accessed May 5, 2023].
- [10] P. R. Manager, "Paprika recipe manager - app for recipes, menu planning, and grocery shopping," <https://www.paprikaapp.com/>, [Accessed May 5, 2023].
- [11] Yummly, "Yummly - recipes, grocery shopping, meal planning," <https://www.yummly.com/>, [Accessed May 5, 2023].
- [12] Nordic Council of Ministers, "Nordic nutrition recommendations 2012: Integrating nutrition and physical activity," Nordic Council of Ministers Secretariat, Report Nord 2014:002, 2014, pp. 217, 249, 272, 281, 335, 349, 385, 407, 413, 419, 423, 435, 449, 465, 475, 501, 509, 515, 535, 543, 573, 583, 591 and 601. [Online]. Available: <http://dx.doi.org/10.6027/Nord2014-002>
- [13] P. Singh and Łucja Zaborowska, "Calorie deficit calculator," 2023-02-02. [Online]. Available: <https://www.omnicalculator.com/health/calorie-deficit>
- [14] J. Michałowska, "Weight gain calculator," 2023-02-02. [Online]. Available: <https://www.omnicalculator.com/health/weight-gain>
- [15] A. Zajac and J. Michałowska, "Maintenance calorie calculator," 2023-02-09. [Online]. Available: <https://www.omnicalculator.com/health/maintenance-calorie>
- [16] T. Pikes and R. Adams, "Computational nutrition: An algorithm to generate a diet plan to meet specific nutritional requirements," p. 36, 2016-05-27. [Online]. Available: https://scholarworks.gvsu.edu/cgi/viewcontent.cgi?article=1068&context=oapsf_articles
- [17] DTU Food Institute, "Frida: Danish Food Composition Database," <https://frida.fooddata.dk/?lang=en>, Accessed 2023, accessed on May 15, 2023.
- [18] OpenAI, "Chatgpt: A large language model trained by openai," <https://chat.openai.com/>, 2021, accessed on May 15, 2023.
- [19] J. Nielsen, "How many test users in a usability study?" *Nielsen Norman Group*, 2006, accessed: May 2, 2023. [Online]. Available: <https://www.nngroup.com/articles/how-many-test-users/>

9 Appendices

9.1 NourishNexus Wireframe

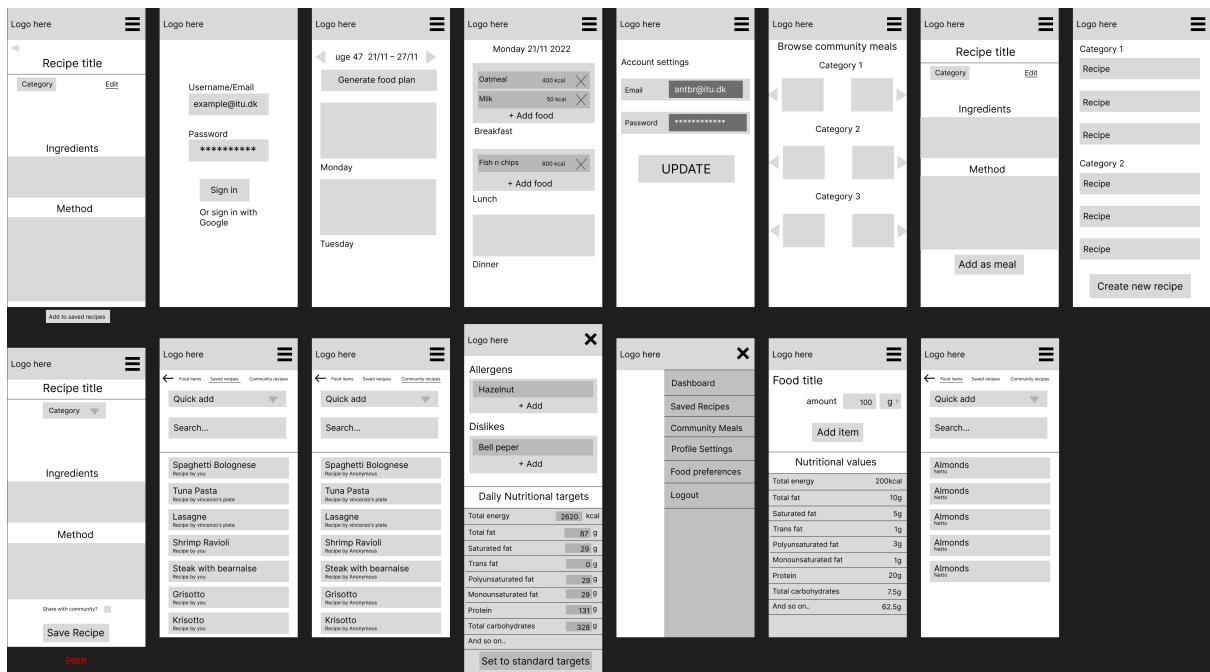


Figure 9.1: Example Image

9.2 NourishNexus Mockup

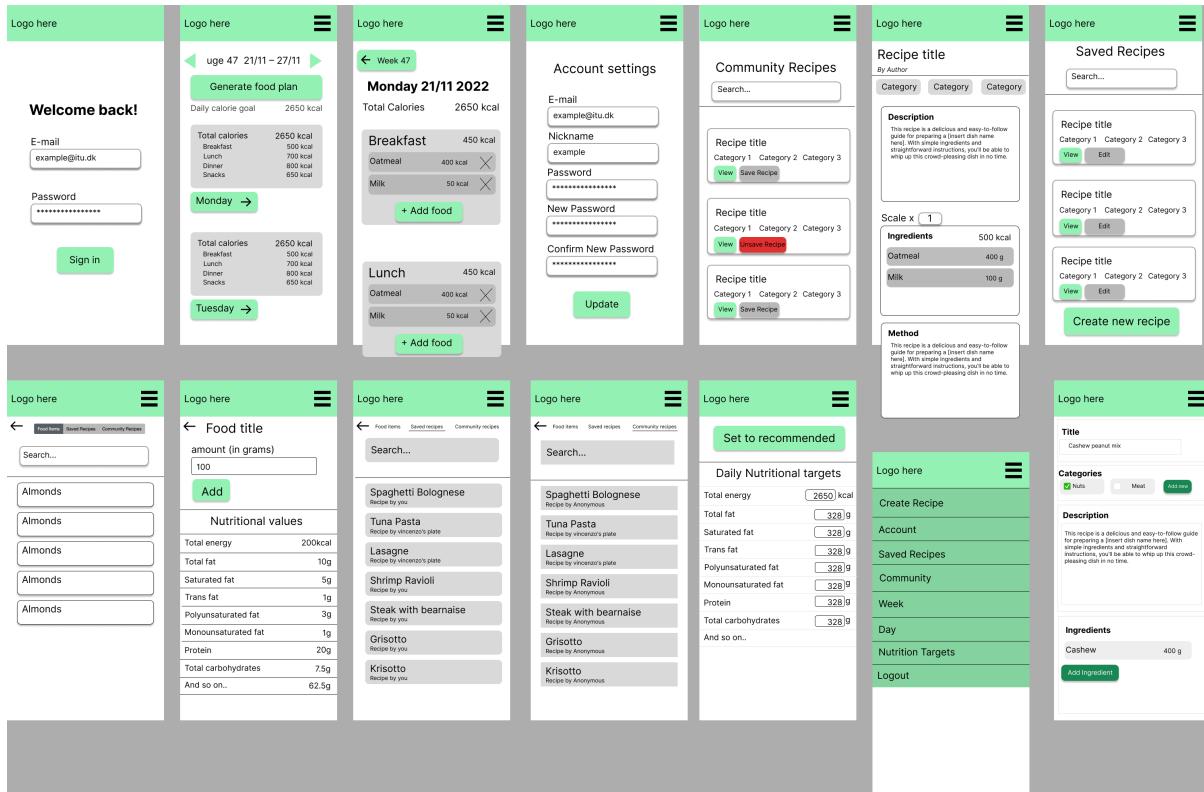


Figure 9.2: Example Image

9.3 User Tests

Success criteria:

All of the users must be able to...

- ...register as a user
- ...login
- ...set their nutrient targets
- ...save recipes from the community page
- ...create recipes
- ...generate a meal plan on the week page
- ...view recipes on the day page
- ...add and remove food items and recipes on the day page
- ...Logout

Information to test users

- What is the purpose of the app?
 - This is a webapp that can help you in planning your meals and also is able to track a lot of different nutrients.
 - The meal plan generator will try to make a plan for you which will give you the right amount of a lot of different nutrients within a week. It will also take into account what you ate in the previous weeks, so that if you got more than needed of a certain nutrient in the previous week, the plan will try to give you less and vice versa, to balance the intake.
 - You can specify the amounts yourself or use the recommended amounts based on age, gender, weight, height, physical activity and your weight goals.
 - The meal plan it makes takes into account what you have eaten in the previous weeks and tries to plan meals that will give you all the recommended amount of all nutrients distributed over a week.
 - You can get inspired by other users' recipes and share your own or keep them for yourself.

Tasks:

How would you...

- ...register as a user?
- ...set your nutrient targets?
- ...create your own recipes?
- ...save a recipe from another user?
- ...generate your meal plan?
- ...register what you have eaten?

Interview Questions

- Would you consider using the application in its current state? If not, what improvements would need to be introduced to get you onboard?
- What was your overall impression of the meal planning web application? Was it easy to use and navigate?

- Can you talk about the features of the web application that you found most useful? Were there any features that you wish were different or more robust?
- How did the web application compare to other meal planning tools or resources you have used in the past?
- Were there any specific challenges or difficulties you encountered during the tasks? If so, how did you handle them?
- Were there any moments during the think-aloud test where you felt unsure or hesitant about your approach? If so, why?
- Were there any moments during the think-aloud test where you felt the web application did not fully meet your expectations or needs in terms of meal planning and tracking? If so, please elaborate on what was lacking or could be improved.

Test sessions

Man, 52 years old, Anders Breinholt

The participant was able to perform all the tasks

The feedback given during the test was

- The test person had difficulty spelling the NourishNexus name
- Could, for some reason, not see the button to generate the recommended nutrient targets.
 - We probably need to update the text from “on this page” to “with the click of the button below” or something.
- Note that for some nutrients, their upper bound value was not shown after going through the generator.
 - This is probably due to some upper bounds being set to maximum infinity.
- Noted that it was a lot of work to create recipes.
- Found that it was not clear that saved recipes contained “Your saved recipes” instead of like “Saved recipes in the system”
- He did not like that he could not see his own recipe on the community page.
- He found it confusing that the “to top” button was there, when you were scrolled up
- He found it a lot of work to select so many different recipes to be able to generate the plan
- He did not find the description of the recipe helpful in the community. There was too much text, and he would rather just see it in the “View recipe” page.
- He found the week page to be frustrating, because it shows the same value every day. All the calorie values are the same. He would rather have the name of the recipes for the different meals on the day.
- He found the fact that everything was in grams to be annoying

Interview

- Would you consider using the application in its current state? If not, what improvements would need to be introduced to get you onboard?
 - No, there would need to be a better system of generating the meal plan, so you don't have to save so many recipes. Also the unit in grams needed to be updated to have other units available for certain items.
- What was your overall impression of the meal planning web application? Was it easy to use and navigate?
 - It was easy to use and navigate
- Can you talk about the features of the web application that you found most useful? Were there any features that you wish were different or more robust?
 - The nutritional recommendation calculator
 - The meal plan generator
 - Wished that the meal plan week page, displayed names of recipes instead of the calorie count
- How did the web application compare to other meal planning tools or resources you have used in the past?
 - Have not used anything like this before
- Were there any specific challenges or difficulties you encountered during the tasks? If so, how did you handle them?
 - Could not find the "Set to recommended" button on the nutrient page. I had to help him after 2 minutes of not finding it.
- Were there any moments during the think-aloud test where you felt unsure or hesitant about your approach? If so, why?
 - Not apart from the incident mentioned above
- Were there any moments during the think-aloud test where you felt the web application did not fully meet your expectations or needs in terms of meal planning and tracking? If so, please elaborate on what was lacking or could be improved.
 - The fact you had to save so many recipes, and the calorie information instead of recipe names. Otherwise no.

Woman, 54 years old, Helle Jensen

The participant was able to perform all the tasks

The feedback given during the test was

- The test person had difficulty spelling the NourishNexus name
- Started trying to login, but found out she needed to register first
- Found the “Set to recommended” button to be confusing, would rather have something like “Change nutritional targets” or something else.
- Noted that it seemed very professional that the generated values got calculated so fast
- Thought that “Community recipes” was confusing as she did not know if community implied a group of people within the system, or just the entire user base
- Noted that the navbar structure was off. She thought we needed to have the most important stuff at the top, so she was confused as to why account was in second position
- She thought that the titles “Day” and “Week” were bad, because she did not know if it was a weekly or daily meal planner, and that needed to be more obvious.
- She thought that the ordering on the community recipes were confusing. She would have liked to have it grouped after mealtype.
- She would have liked more info on when she had enough recipes to generate the plan. It was frustrating saving recipes, then going back to find out she had not saved enough, them going back and forth again and so on.
- She thought it was ok that it took time to save the recipes, since the users of the app are probably pretty serious about their meals.
- She also noted that the week page was lacking, and like the previous test person, she wanted the name of the recipes/meal she was having instead of the calorie count.
- She thought it a bit annoying to be redirected to the login screen upon logging out, would rather have seen a “Thank you, and come back another time” screen or something.
- She thought that all the values in grams was annoying

Interview

- Would you consider using the application in its current state? If not, what improvements would need to be introduced to get you onboard?
 - If she was in the target demographic yes, but she did not feel the need to use a meal plan
- What was your overall impression of the meal planning web application? Was it easy to use and navigate?
 - It was easy to use and navigate
- Can you talk about the features of the web application that you found most useful? Were there any features that you wish were different or more robust?

- The nutritional recommendation calculator
 - The meal plan generator
 - Wished that the meal plan week page, displayed names of recipes instead of the calorie count
- How did the web application compare to other meal planning tools or resources you have used in the past?
 - Have not used anything like this before
- Were there any specific challenges or difficulties you encountered during the tasks? If so, how did you handle them?
 - Had a problem with registering, tried to login at the start, but found she needed to register first
 - At one point in time, she clicked on the “arrow back” icon on her phone, and got taken back to the register page, which caused some confusion. She used arrow forward to get back again.
- Were there any moments during the think-aloud test where you felt unsure or hesitant about your approach? If so, why?
 - She felt unsure of where to go after generating the nutritional recommendation values. At that point, the guide had disappeared, because she accidentally exited the page, so she was very much in doubt about where to go, “Create Recipe”, “Saved Recipes” or “Community Recipes”.
- Were there any moments during the think-aloud test where you felt the web application did not fully meet your expectations or needs in terms of meal planning and tracking? If so, please elaborate on what was lacking or could be improved.
 - Recipe names instead of calorie info, otherwise no.

Male 54 years old, Henrik Dorph

Completed all the tasks successfully

Henrik:

- Problemer med at stave nourishnexus
- registrerer sig selv
- Overser set to recommended knap, men finder den senere
- finder selv ud af at ændre på sukker, fordi han har sukkersyge
- Tror man skal trykke på en ok eller submit knap i set nutrition
- Finder create recipe med det samme
- Finder nemt fødevarer til kylling i karry
- Lidt overvældet af hvor mange opskrifter man skal have og spørger efter om der er lister at opskrifter hvor man kan sige check på dem alle på en gang.
- Laver plan, men bliver ikke succes fordi han ikke tilføjede nok, men kunne godt lave en plan.

Interview

- Would you consider using the application in its current state? If not, what improvements would need to be introduced to get you onboard?
 - ja why not
- What was your overall impression of the meal planning web application? Was it easy to use and navigate?
 - ja very intuitive
- Can you talk about the features of the web application that you found most useful? Were there any features that you wish were different or more robust?
 - planning meal for at week very useful
 - brugeren ved ikke at man bare kan tilføje ting og at madplanen så tager det med.
- How did the web application compare to other meal planning tools or resources you have used in the past?
 - har ikke brugt andre værktøjer (paprika, men kan ikke gøre andet end opskrifter og indkøbsliste)
- Were there any specific challenges or difficulties you encountered during the tasks? If so, how did you handle them?
 - nej, men have lidt problemer med registrering af mad på dagen efter at have lavet en madplan
- Were there any moments during the think-aloud test where you felt unsure or hesitant about your approach? If so, why?
 - nej, men have lidt problemer med registrering af mad på dagen efter at have lavet en madplan
- Were there any moments during the think-aloud test where you felt the web application did not fully meet your expectations or needs in terms of meal planning and tracking? If so, please elaborate on what was lacking or could be improved.
 - nope

lidt feedback:

lidt underligt med rå kylling man kan tilføje.

Måske filterer lidt af fødevarerne.

kan nemt finde ud af at tilføje sin egen opskrift til dagen, med flere portioner på en gang.

Female, 52 years old, Karin Bay Dorph

Completed all the tasks successfully.

- She had problems spelling nourishnexus.dk
- She could easily find the register page and she easily registers herself as a new user.
- She is a little overwhelmed by the amount of different nutrients there are and she does not know the amounts for each nutrient, but she finds the Set to recommended button.
- She thinks that there are too many decimals in the number fields for each nutrient, but she thinks it is okay when it is in milligrams or micrograms.
- She does not realize that she needs to add the category to the recipe she is creating after having created a new category.
- She thinks it is hard to judge the amounts of some ingredients, because she only knows the amounts for vegetables per piece, but she still does a good job adding the ingredients in grams.
- Cannot find sweet peas.
- She thinks the meal types should be unchecked to start with.
- Initially tries to find other peoples shared recipes in saved recipes, but finds out that she needs to go to the community page.
- She thinks it is weird that the save button turns red when saving a recipe.
- She presses the generate meal plan button and it cancels because she does not have enough recipes saved.
- She thinks it is because she did not save her daily nutrient targets.
- Siger at hun ikke ved om hun har gempt sine goals når hun trykker på set to recommended
- She is unsure if her nutrient goals have been saved. (They save automatically)
- She finds out that the nutrient goals have been saved, but is unsure what causes the generated meal plan to cancel.
- She still thinks the problem has to do with her nutrient goals, and does not realize it is because she does not have enough recipes saved.
- She says she would prefer that there are recipes saved in advance.
- She thinks that she needs to create all the recipes, but realized shortly after that she can go to the community page and save other people's recipes from there.
- She saves enough recipes and goes to the week page, and generates a meal plan.

Interview

- Would you consider using the application in its current state? If not, what improvements would need to be introduced to get you onboard?
 - It seems a bit too complex and she needs to do a lot of work to get going.
 - She thinks it is very nice that it can generate recommendations for nutrient targets.
 - She thinks it would be better and that she would use the web app if it told the user how many recipes they need.

- What was your overall impression of the meal planning web application? Was it easy to use and navigate?
 - Ambitiøst, tror den vil hun vil kunne bruge den med hendes tilpasningsforslag. Smart at den finder ud af næringsstofferne i hendes egne opskrifter.
- Can you talk about the features of the web application that you found most useful? Were there any features that you wish were different or more robust?
 - Set recommended, var lige ud af landevejen.
- How did the web application compare to other meal planning tools or resources you have used in the past?
 - paprika kan hente opskrifter på nettet. Går ind på Valdemarsro fx og tilføjer den til paprika.
 - Problemets er, at hun ikke er alene. Ved ikke hvordan man skulle tilpasses til flere personer på en gang.
- Were there any specific challenges or difficulties you encountered during the tasks? If so, how did you handle them?
 - Hvordan hun fik genereret hendes targets, men kunne finde det efter noget tid med at scrollle op og ned på siden.
- Were there any moments during the think-aloud test where you felt unsure or hesitant about your approach? If so, why?
 - ja
- Were there any moments during the think-aloud test where you felt unsure or hesitant about your approach? If so, why?
 - 1 gang da hun ikke havde fået lavet planen da den fejlede. Ville hellere have at den kun viste hvad fejlen præcist var. Så ikke at der både stod set nutrient targets og save more recipes på en gang.
- Were there any moments during the think-aloud test where you felt the web application did not fully meet your expectations or needs in terms of meal planning and tracking? If so, please elaborate on what was lacking or could be improved.
 - Den er mere intuitiv end paprika til at tilføje mad til madplanen. Ikke tydeligt hvad navigationsmenuens knapper betyder i paprika, men det er det i nourishnexus. I paprika ved hun stadig ikke efter lang tids brug af appen hvad note og tilføj menu, gør eller betyder.

Bugs found:

There is an error when the meal plan has already been made, and is being generated again.

Male 23 years old, Alexander Frederiksen

Finder register hurtigt
Kan godt lide loading cirklen
Lidt overvældet da han kom ind på set nutrient targets.
Finder set recommended
Exemplar på aktivitetsniveau
Irriterer ham lidt at boxene ikke aligner i nutrient targets
Finder nemt create recipe
Automatisk stort forbogstav på categories
Overser at han skal tjekke category efter han har tilføjet en ny category
Finder nemt ingredienser og tilføjer dem til
to top skal poppe op når man er rullet lidt ned
vil gerne kunne tilføje noget til andres opskrifter, både som mulig
filtrerings mulighed på community page og saved recipes
sig hvor mange
Lidt forvirret over at der ikke var noget til at starte med
Når planen ikke opfylder alle næringsstoffer, skal den vise hvilke næringsstoffer den ikke
opfylder.
Kan godt lide de små ikoner

Interview

Would you consider using the application in its current state? If not, what improvements would need to be introduced to get you onboard?

- a. Jeg ved ikke jeg vil bruge den som nu, jeg kan se den har potentielle, så hvis den lige får en ekstra kærlig hånd, så vil jeg helt sikkert bruge den.

What was your overall impression of the meal planning web application? Was it easy to use and navigate?

- b. Jeg ved ikke om jeg har noget mere specifikt end de ting jeg allerede har sagt, men jeg kan overordnet meget godt lide systemet, med at man kan se community opskrifter, måske herude hvor der også stod morgenmad, frokost, aftensmad og snack kategorier.
- c. Det var ekstremt nemt at navigere, det eneste jeg havde problemer var når jeg søgte for food og det var tomt

Can you talk about the features of the web application that you found most useful?

Were there any features that you wish were different or more robust?

- d. Kunne rigtig godt lide at man fik mange forskellige opskrifter og at man kan tilføje sine egne opskrifter. Kan godt lide at han selv kan vælge hvad der skal være med i sin madplan.

How did the web application compare to other meal planning tools or resources you have used in the past?

- e. Kan godt lide at det er gratis. Bliver skræmt væk på andre meal planning muligheder fordi han skal give kortinformationer, og han vil ikke betale.

Were there any specific challenges or difficulties you encountered during the tasks? If so, how did you handle them?

f. -

Were there any moments during the think-aloud test where you felt unsure or hesitant about your approach? If so, why?

g. Han var lidt usikker, da der ikke kunne genereres en succesfuld måltidsplan,
ift. hvad han skulle gøre for at den kunne lave en succesfuld måltidsplan.

Were there any moments during the think-aloud test where you felt the web application did not fully meet your expectations or needs in terms of meal planning and tracking? If so, please elaborate on what was lacking or could be improved.

h. Nej, kunne godt lide at alt var så nemt. Nemt at forstå hvilke retter der var.
Næsten uden at læse teksten på knapperne kunne han forstå hvad knappen gjorde pga farverne.

Ekstra:

Jeg synes det ville være rart hvis jeg kunne justere så jeg kan spise en mindre morgenmad og en større frokost

Male 22 years old, Laust Jensen

The test person did not have any issues performing the tasks.

Interview

1. Would you consider using the application in its current state? If not, what improvements would need to be introduced to get you onboard?
 - i. He would not use it currently, there were a bit too many issues like the fact that you could not just choose all recipes to save. He got frustrated with going back and forth selecting recipes, only to find out he did not have enough.
2. What was your overall impression of the meal planning web application? Was it easy to use and navigate?
 - a. Yes it was very easy to use and navigate
3. Can you talk about the features of the web application that you found most useful?
 - a. The meal planner was very useful, when it was not prompting for more recipes to be saved.
4. How did the web application compare to other meal planning tools or resources you have used in the past?
 - a. He had not used any meal planning tools in the past
5. Were there any specific challenges or difficulties you encountered during the tasks? If so, how did you handle them?
 - a. Was frustrated and challenged by the fact that there was no clear indication as to how many recipes he had to save, to make the meal plan

6. Were there any moments during the think-aloud test where you felt unsure or hesitant about your approach? If so, why?
 - a. Not other than the frustrating process of meal planning
7. Were there any moments during the think-aloud test where you felt the web application did not fully meet your expectations or needs in terms of meal planning and tracking?
 - a. Not other than what has previously been discussed