2026-01-27

Matslumparen

Agnes Wilson

Degree Project Fed24D
Medieinstitutet

Table of Contents

Project idea summary 3

Application flow 3

Purpose 5

Target audience 6

Goal 7

Activity and time plan 8

Site map/function analysis/technical breakdown 10

User stories x3 12

- Persona 1: Experienced User / Primary User
- Persona 2: New User / Secondary User
- Persona 3: Seasonality- and Sustainability-Oriented User

Accessibility 15

Test plan 16

Tech stack 17

Reflections 18

Links 19

Screenshots / gif 20

Project idea summary

Matslumparen is a smart web application designed to solve the daily dilemma of ”what do I eat tonight?” It utilizes a user’s personal database of recipes and intelligently selects the most suitable dish by incorporating real-time weather data and seasonality. This logic enables the app to return contextually appropriate suggestions, such as a warming tomato soup on a cold summer day, or a crisp, fresh salad on a warm one. This approach encourages seasonal ingredient use and minimizes the daily stress of meal selection.

Application flow

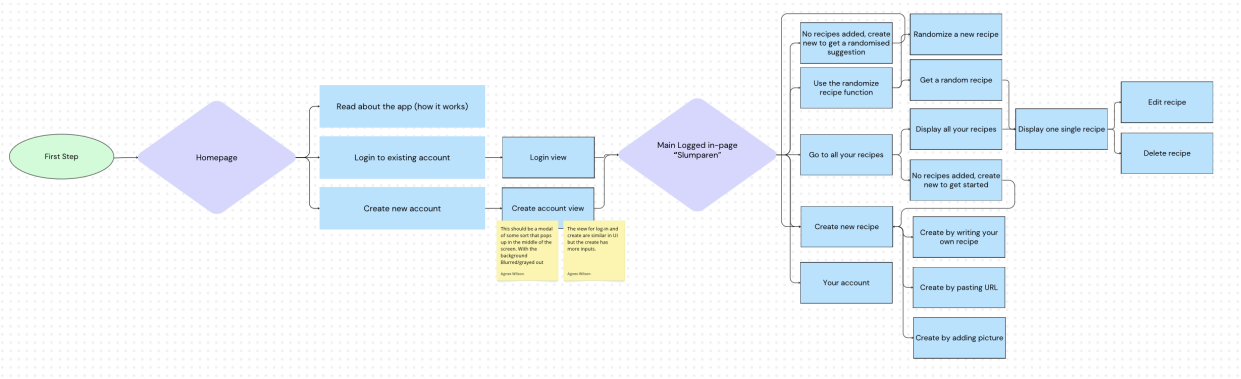
The visual flowchart and the first draft of the wireframes illustrate the intended user journey through the Matslumparen web application. The flow begins on the homepage, where users are introduced to the concept of the application. From here, they can choose to read more about how the app works (below the fold), log in to an existing account (accessible at the bottom of the page), or create a new account. This structure ensures that both new and returning users are clearly guided from the outset.

After logging in or creating an account, users are taken to the main logged-in view, *Slumparen*. This page functions as the central hub of the application and highlights the primary action: generating a randomized recipe suggestion based on the user’s existing recipe collection and contextual factors such as weather and current season. If no recipes have been added yet, the interface instead guides the user toward creating their first recipe, lowering the barrier to getting started and preventing a dead-end experience.

From the main view, users can navigate to related sections such as viewing all saved recipes, creating a new recipe, or accessing account settings. The intended interaction flow particularly supports experienced users by allowing them to log in and access the recipe randomization feature quickly, without the need to navigate through multiple pages or tabs. At the same time, the persistent and easily accessible tab navigation ensures that all other parts of the application remain available whenever the user wishes to explore or manage additional features.

The recipe creation flow is designed to be flexible, allowing users to add recipes in multiple ways, by writing a recipe manually, pasting a URL, or adding a recipe from an image. This accommodates different user behaviors and minimizes friction when building a personal recipe collection.

Overall, the flow emphasizes simplicity and low cognitive load. The navigation structure is intentionally shallow and centered around a small number of key actions, making it easy for users to move between discovering what to cook, managing their recipes, and expanding their collection. The web page clearly prioritizes the core use case, answering the question “What do I want to eat?” while still supporting secondary tasks without overwhelming the user.



Purpose

This idea originated from a personal need, but I strongly believe that the application has the potential to benefit far more people than just myself.

The core concept of the application can be described as standing on two main pillars. The first pillar is the user's personal recipe collection. I enjoy cooking and continuously discovering new recipes, which I then adapt over time as I learn new techniques and as my preferences evolve. For a long time, I have wanted a structured and convenient way to gather all my saved browser tabs, copied links, photographed recipes, and my own creations into one place.

In practice, I tend to rotate between a core set of approximately 15–20 dishes that I cook regularly. Over time, these dishes are replaced by new ones and often fall into oblivion, I simply forget that I ever used to cook them. I do not believe this is a problem unique to me. Most people have a mental cookbook of reliable dishes they know how to prepare, especially on a weekday evening after work. That moment is rarely ideal for browsing a large recipe application with thousands of options, which often leads to decision fatigue rather than inspiration.

This application does not aim to push users toward new recipes, nor does it contain a recipe database of its own. Its sole purpose is to help users collect and organize their own recipes in a way that suits their personal habits, and to assist in answering the recurring question: "What do I want to eat?"

The second pillar of the application is seasonality and weather awareness. I personally feel a strong connection between what I want to eat and external factors such as current season and weather. For example, I would find it difficult to enjoy a fresh tomato salad on a snowy day in November, just as a heavy stew feels out of place on a warm summer day. This preference is driven by two main factors.

The first is comfort and satisfaction. Food needs to feel appropriate for the context in order to be fulfilling. A light, fresh salad rarely feels satisfying as a winter dinner, whereas a salad with warm and seasonal elements, such as roasted brussels sprouts, orange, walnuts, and a honey-based dressing, can feel completely right. I believe many people experience this instinctively, even if they do not consciously reflect on it. An important aspect of this application is that it is entirely personal: there is no concept of right or wrong. If a user prefers to eat Caesar salad throughout December and January, they can simply assign that dish to those seasons.

The second factor is related to taste, availability, and cost. Seasonal produce is often more flavorful, more affordable, and more locally available. Fresh tomatoes tend to taste best during the Swedish summer months, while ingredients such as kale are cheaper and of higher quality in late autumn. By encouraging seasonal choices, the application can also support more sustainable consumption by reducing unnecessary transportation and promoting locally produced ingredients.

Ultimately, the goal of this application is to provide a meaningful and reliable answer to the recurring question “What do I want to eat?” by basing that answer on the user’s own preferences, habits, and taste. By combining a personal recipe collection with contextual factors such as seasonality and weather, the application makes it easier to rediscover dishes that may have been frequently cooked in the past but forgotten over time. At the same time, it lowers the threshold for building and maintaining a personal recipe collection, turning meal planning from a source of stress into a simple and intuitive process.

Target audience

The primary target audience for this application consists of individuals who regularly cook their own meals and have an existing interest in food, but who experience decision fatigue when planning everyday dinners. Although they may enjoy cooking, they often fall back on a small set of familiar dishes due to convenience and mental overload at the end of the day.

This group is likely to already save recipes in an unstructured manner such as through browser tabs, bookmarked links, screenshots, handwritten notes, or photos taken from cookbooks. Over time, these collections become fragmented and difficult to navigate. Rather than reducing mental effort, the scattered nature of these recipe collections often adds to the cognitive load, as users must remember where a specific recipe was saved and how to retrieve it. As a result, meal planning becomes a recurring source of stress rather than a creative or enjoyable activity. This group is also likely familiar with using various applications and websites to search for inspiration and discover new recipes.

The target audience is interested in this application because it addresses a familiar and recurring problem. Instead of introducing new recipes or external inspiration, the application builds on the user’s existing preferences and habits. The application reduces cognitive load and simplifies everyday decision-making. This makes it

particularly appealing to users who want structure and support in meal planning without sacrificing personal taste.

The secondary target audience includes individuals who value seasonality and sustainability in their food choices. These users may consciously or subconsciously adjust what they eat based on the time of year, ingredient availability, and price. They are interested in making food choices that feel appropriate, whether motivated by cost, environmental considerations, or a desire to support local producers.

This group is likely interested in the web application because it makes it easier to incorporate what they perceive as “the right choices” into everyday life. By reducing the effort required to identify seasonal ingredients and suitable dishes, the application allows users to eat seasonally without the distraction of non-seasonal alternatives commonly presented in other recipe platforms. It also minimizes the time spent considering what is currently in season and how to plan meals accordingly.

Goal

After three weeks, the project has resulted in a functional web application prototype that demonstrates the core concept of Matslumparen. Users can log in using a hard-coded user account, generate randomized meal suggestions and browse the integrated recipe collection. The generated suggestion is based on predefined recipes, seasonality and weather. The goal of this prototype was not to deliver a fully production-ready application, but to provide a stable and usable core experience that reflects the intended user flow and key design decisions.

The end product is expected to be used as an everyday tool that supports users in meal-planning scenarios. It can function both as a simple platform for collecting and managing personal recipes through the three available recipe input methods, and as a way to reduce the daily effort of deciding what to cook. The target audience is expected to interact with the application primarily during meal-planning situations, such as after work or when deciding on dinner. Users log in, generate a recipe suggestion based on their preferences, and either proceed with the suggested dish or request a new option. Over time, the application supports habitual use by helping users both decide what to eat in the moment and maintain a centralized collection of their personal recipes.

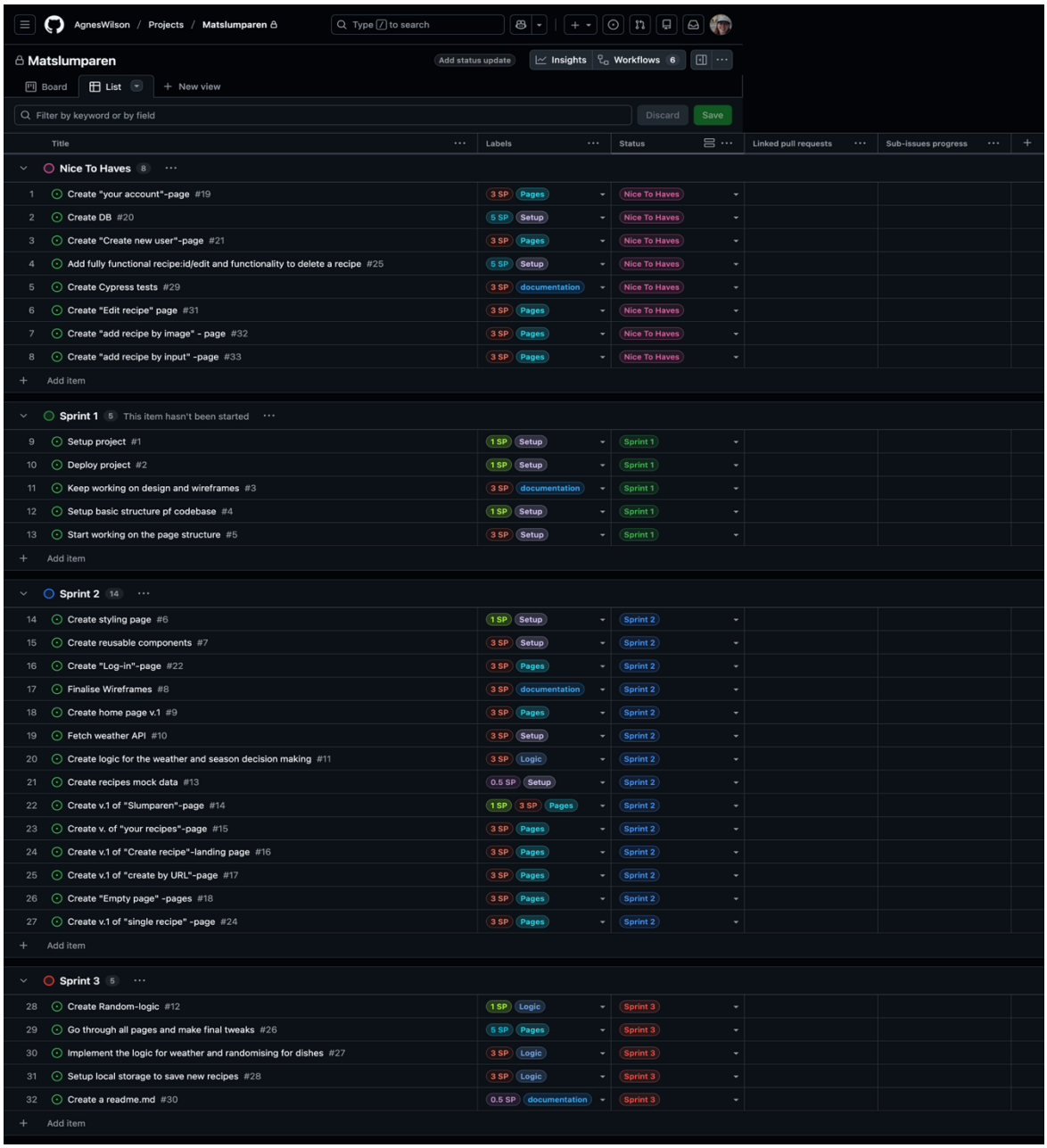
Activity and time plan

Most of the core development and functionality was produced during the second sprint, as planned. This allowed the third sprint to focus on refinements and ensuring a fully functional and cohesive product. By structuring the work in this way, the final sprint was dedicated to polishing and stabilizing existing features rather than implementing fundamental components.

I structured the planning for this project with significant buffer margins. Rather than trying to achieve as much as possible in a short window, I deliberately over-estimated the time required for core tasks. This strategic prioritization proved essential when an unforeseen personal situation temporarily limited my available development time. Because the 'must-have' features were prioritized early and the project scope was realistically defined, I was able to complete almost all planned functionality within the original timeframe. Notably, the final week remained dedicated solely to documentation and reporting as intended, without the need to reallocate that time to programming. Overall, the planning allowed for a successful delivery while maintaining a manageable workload even during unexpected circumstances.

Sprints / time for sprint	Objective
• Sprint 1 3 days (29/12, 30/12, 2/1)	• Set up the project in GitHub and get technical workflow, including deployment, fully functional • Keep working on design and wireframes • First user test on the wireframes to see if any changes need to be done to the design • Set up basic structure of the code base • (Start working on page structure i.e. routing, tabs…)
• sprint 2 2 weeks (5/1 – 16/1)	• Create v.1 of a majority of the pages • Finalize wireframes • Fetch weather API • Create mock data • Create logic for weather and

	seasonality decision making • Do user testing on v.1 of the existing pages • Get Slumparen working • Create random logic for Slumparen
• sprint 3 1 week (19/1 – 23/1)	• Final tweaks of all pages (version 2) • ~~Setup local storage~~



Site map/function analysis/technical breakdown

This section describes the technical structure of the Matslumparen web application, focusing on functionality, data handling, and system logic. While the *Application Flow* section presents the user journey, this section translates that flow into concrete technical components.

This breakdown specifically discusses the outcome of the prototype. Some pages, such as “Create Account,” are not included because they are outside the scope of the prototype. Similarly, a fully functional database with CRUD operations is not implemented in this project.

Site Map

The application is structured around a small number of clearly defined views, connected through client-side routing. The navigation hierarchy is intentionally shallow to ensure fast and intuitive access to the core functionality.

Home/Landing View

Introduces the concept and purpose of the application. Provides entry points for logging in and creating an account. Additional content below the fold explains how the application works.

Login View

Mockup of the login-flow using a hard-coded user account. Upon successful login, the user is redirected to the main logged-in view.

Slumparen (Main Logged-in View)

Serves as the central hub and primary entry point after login. The user can generate a single recipe suggestion from their collection based on factors such as the current season and weather. Navigation to the individual recipe detail view is available. If no recipes exist, the page guides the user to create their first recipe (empty page view). Recipe generation occurs when the user actively pushes a button.

Recipes Overview View

Displays a list of all saved recipes, the users can also navigate to individual recipe details by clicking the overview cards.

Recipe Detail View

Displays all information about a single recipe.

Create Recipe View

Explains to the user how to add recipes to their database. The app supports multiple input methods, including manual entry, pasting a URL, or adding a recipe from an image. In the prototype, no input method is functional.

Core Features

Recipe Storage and Management

In this prototype, the application uses a set of hard-coded recipes to demonstrate how the ”Slumparen” logic and filters work in practice. The focus has been on proving the core concept rather than building a backend. For a production-ready version, the next step would be to implement a real database with full CRUD functionality, so users can save and manage their own recipes.

Recipe Randomization (Slumparen)

The randomization function selects a single recipe from the hard-coded collection. Before selection, the recipe pool is filtered based on seasonality and weather. If the user generates a new suggestion, the filtered pool is stored in state so that only the randomization logic is repeated.

Seasonality Logic

Seasonality is handled through user-defined labels rather than fixed rules. The users decide which recipes belong to which seasons, giving full control over what is considered appropriate at different times of the year.

Weather Integration

The application fetches current weather data from an external API. For the prototype, the data is hard coded to Stockholm. Future improvements should include fetching localized weather data based on the user’s location. Weather conditions are mapped to simplified internal states (e.g., cold, mild, warm, sunny, rainy, cloudy), which influence which recipes are included in the randomization process.

Technical Breakdown

Data Management

A selection of initial recipes is hard coded into the prototype to provide example data. A mock user account is used to simulate logged-in behavior.

External Services

A weather API provides real-time weather data. API responses are processed and mapped to simplified weather categories, which are used by the recipe randomization logic.

User stories x 3

Persona 1: Experienced User / Primary User
Name: Johan Andersson
Age: 34
Occupation: UX Designer
Location: Stockholm
Technical proficiency: High

Background

Johan regularly cooks his own meals and has a strong interest in food and cooking. Over time, he has accumulated a large personal collection of recipes saved across multiple platforms, including browser bookmarks, notes, and screenshots. Despite this, he often experiences difficulty deciding what to cook after work due to mental fatigue and decision overload. This frequently leads him to rotate between the same five or six familiar dishes until he becomes tired of them and feels forced to search for new recipes.

Needs and Goals

- Reduce cognitive load when deciding what to cook on weekdays
- Reuse and rediscover previously saved and frequently cooked recipes
- Maintain enjoyment and motivation for cooking, even on weekday evenings

Behavior and Usage Pattern

Johan typically accesses the application in the evening after work. He logs in and immediately uses the recipe randomization feature Slumparen to receive a meal suggestion based on seasonality and current weather conditions. He expects the interaction to be fast and frictionless and does not want to manually browse or filter recipes. When he feels inspired, usually once or twice a month, often on weekends, he actively looks for new recipes in cookbooks and other recipe platforms and adds them to his personal collection in Matslumparen.

Relevance to the Project
This persona represents an experienced and frequent user who primarily interacts with the core functionality of Matslumparen. His behavior validates the importance of ease and efficiency through immediate access to the main feature, and a clear, predictable navigation structure.

Persona 2: New User / Secondary User
Name: Karin Lindholm
Age: 41
Occupation: Marketing Coordinator
Location: Malmö
Technical proficiency: Medium–High

Background
Karin enjoys cooking but often lacks the time and mental energy required for meal planning during busy periods. She saves recipes in an unstructured way, for example by photographing cookbook pages or bookmarking links. As a result, she frequently forgets which recipes she has saved and relies on a limited set of familiar meals.

Needs and Goals
- Store all personal recipes in one centralized location
- Minimize time and effort spent on everyday meal planning
- Receive guidance on what to cook without browsing large recipe databases

Behavior and Usage Pattern
Karin initially uses Matslumparen to add recipes using multiple input methods, such as pasting URLs or uploading images. She finds the low threshold for adding recipes, as well as the ability to comment on, edit, label, and group recipes by seasonality, particularly valuable. Once she has built a small collection, she uses the randomization feature sporadically during weekdays to help decide what to cook when she feels unmotivated or mentally tired.

Relevance to the Project
This persona highlights the importance of flexible recipe input methods and clear onboarding. This persona supports design decisions related to reducing barriers for new users and enabling personalization through editing, commenting, and labeling recipes. It also reinforces the application's focus on the user's own preferences rather than externally curated content.

Persona 3: Seasonality- and Sustainability-Oriented User
Name: Elin Persson
Age: 27
Occupation: Master's student
Location: Uppsala
Technical proficiency: Medium

Background
Elin values seasonal and sustainable food choices for both environmental and economic reasons. While she already adapts her cooking habits according to seasonality, she finds it time-consuming to continuously assess what is appropriate to eat throughout the year. She is often frustrated by other recipe platforms that recommend non-seasonal dishes, as this influences her to cook out of season, which conflicts with her values.

Needs and Goals
- Support seasonal and sustainable food choices
- Reduce effort when planning seasonally appropriate meals
- Maintain control over a curated and personal recipe collection

Behavior and Usage Pattern
Elin uses Matslumparen primarily as a decision-support tool rather than a source of inspiration. She assigns recipes to specific seasons and relies on the application to suggest dishes that are appropriate for the current time of year. She expects consistent, predictable behavior and minimal distractions from non-seasonal alternatives.

Relevance to the Project
This persona supports the project's second core pillar: seasonality awareness. It justifies the implementation of season-based labeling and filtering of recipes, allowing users to avoid distracting suggestions, such as summer-oriented salads in winter, that might otherwise influence them to act against their sustainability values.

Accessibility

Accessibility is an integral part of the Matslumparen web application and has been considered from the early design and wireframing stages. Since the application is intended for everyday use by a broad audience, it is essential that it is predictable, easy to navigate, and accessible regardless of the user's abilities, devices, or interaction methods.

From a visual and structural perspective, the wireframes are designed with a clear hierarchy and consistent grouping of content. Headings, sections, and interactive elements are visually separated and grouped where appropriate, making the interface easier to understand and scan. Color contrast will be tested to ensure sufficient readability, and color will never be the sole means of conveying information. For example, required form fields in the recipe creation flow will be clearly indicated using text labels and symbols in addition to color-based cues.

Semantic HTML is used throughout the application to ensure proper structure and meaning. Elements such as headers, navigation areas, main content sections, forms, and buttons will be implemented using appropriate HTML tags to support screen readers and other assistive technologies.

Keyboard accessibility is a key consideration. All core functionality, such as logging in, navigating between tabs, generating a recipe suggestion, and creating or editing recipes, is fully usable without a mouse. Focus order follows a logical sequence that mirrors the visual layout of the interface, and visible focus states will be implemented to ensure users always understand where they are within the application. Interactive elements such as buttons, links, and tabs are reachable and operable using standard keyboard interactions.

Predictability and consistency are central to the user experience of Matslumparen and contribute directly to accessibility. The main navigation remains consistent across all logged-in views. This reduces cognitive load and benefits users with cognitive or attention-related impairments, as well as users accessing the application in low-energy or time-constrained situations.

Accessibility will be treated as an ongoing consideration rather than a one-time checklist. Basic testing will be conducted using keyboard navigation and screen reader tools to identify potential issues early. By embedding accessibility into both design decisions and implementation practices, Matslumparen aims to provide an inclusive experience that aligns with diverse user needs and accessibility standards.

Test plan

User testing was the primary testing method used throughout this project. The main objective of these tests was to ensure that the core functionality and intended use of the application are clearly communicated from the design to the end user. User testing was conducted in three stages across the development process.

The first round of user testing took place early in the project when only wireframes were available. At this stage, the primary objective was to evaluate whether the overall design communicated the intended purpose and core principles of the application. The focus was on the tester's initial understanding of the application, its use case, and the overall impression of the interface, rather than on detailed interactions.

The second round of testing was conducted during the second sprint, when the tester was able to interact with an early functional version of the web application. The main goal at this stage was to assess whether the application behaves in a predictable and understandable way. Test participants were asked to perform basic tasks, such as navigating the interface and interacting with core features. This stage focused on identifying usability issues, potential confusion, or sources of frustration, as well as evaluating the overall user experience.

The third round of user testing took place during the third sprint, once most components and features were in place. The objective of this test was similar to the second round but with a stronger emphasis on the overall experience and flow of the application. This includes evaluating how intuitive the navigation feels, how cohesive the interface is, and how effectively the application supports its core use case.

User testing was conducted in a home environment with two test participants. One participant took part in all three testing stages, allowing for deeper insights into how the application evolves over time from the user's perspective. The second participant only took part in the second testing stage, providing feedback from a user who has not previously interacted with the application and could therefore offer fresh and unbiased insights.

Tech stack

The Matslumparen web application was developed using frameworks and tools that are familiar to me, such as React, TypeScript, ESLint, and Prettier. These technologies have

been used extensively in my previous projects, and by using them here, I can focus on building functionality while continuing to expand and deepen my existing knowledge.

For styling and design, Material-UI (MUI) was used. Although this was a new tool for me, I choose to explore it as an alternative design system, offering a different approach compared to the one used during the React course.

Programming language/framework

TypeScript and React was used for the frontend. MUI provided pre-built components for a consistent and accessible user interface.

Testing

Basic testing was performed manually through user testing sessions.

Deployment tools and pipelines

The application is deployed using GitHub Actions.

Configuration setups for development

Vite is used as the build tool. Prettier and ESLint were used to ensure a consistent code style and maintainable code.

Graphics

Wireframes are created using Figma. Icons and UI graphics were sourced from MUI.

Accessibility checks

Accessibility has been tested using tools such as Google Chrome Lighthouse, A11y Tools, and The A11Y Project checklist. These will help verify color contrast, keyboard navigation, and semantic structure.

Documentation

A complete *README.md* file is provided with clear instructions on cloning the repository, running the project locally, and using the product. Inline code comments have been added where necessary to explain functionality and logic.

Reflections

Planning and Time Management

The most significant takeaway from this project was the importance of realistic planning and "scope management." I chose to structure my timeline with significant

buffer margins, prioritizing a stable Minimum Viable Product (MVP) over a long list of complex features.

This strategic decision proved to be a lifesaver. Due to unforeseen personal circumstances that limited my development time, I was forced to re-evaluate my goals mid-project. Because I had planned for "air" in my schedule and prioritized the core "Slumparen" logic early on, I was able to complete almost all essential and planned functionality. The only major adjustment was the decision not to create a view where users could add and save their own recipes via localStorage, as originally planned.

This felt like a reasonable change since localStorage was never intended to be part of the final, production-ready product. For the eventual end product, I intend to implement a fully functional recipe database; therefore, skipping the "middle-ground" stage of local storage allowed me to focus on the core user experience.

Design and UI Challenges

My approach to design was iterative. I started with wireframes that were not 100% finalized, which I found surprisingly beneficial. It gave me the flexibility to adapt the UI as I discovered what worked best technically. Having a clear vision of the app's "look and feel" made it easy to refine the wireframes during the development phase. This approach also allowed me to test different layouts and placements based on feedback from user tests. Being able to pivot based on user insights rather than being locked into a static design made the final result much stronger.

However, the visual ambition of the project was a double-edged sword. I opted for a highly graphical layout, which turned out to be incredibly time-consuming. I underestimated the effort required for precise positioning and ensuring the design felt "right" across different screen sizes. In hindsight, I wish I had dedicated more time to mobile and tablet design before I started coding. Dealing with responsiveness for such a graphic-heavy site took a disproportionate amount of time. In future projects, I will aim for a more "mobile-first" approach and perhaps a more streamlined visual language to save time on CSS and positioning.

Conclusion

Overall, I am proud of the result. The project taught me that a well-thought-out plan isn't just about what you aim to build, but about how you handle the unexpected. By focusing on the core value and maintaining a flexible design process, I delivered a product that met the estimated outcome, and I was able to do so in a surprisingly collected and organized manner.

Links

Links added as a complement to the screenshots.

Github Project
https://github.com/users/AgnesWilson/projects/1/views/5

Live demo
https://agneswilson.github.io/Matslumparen/

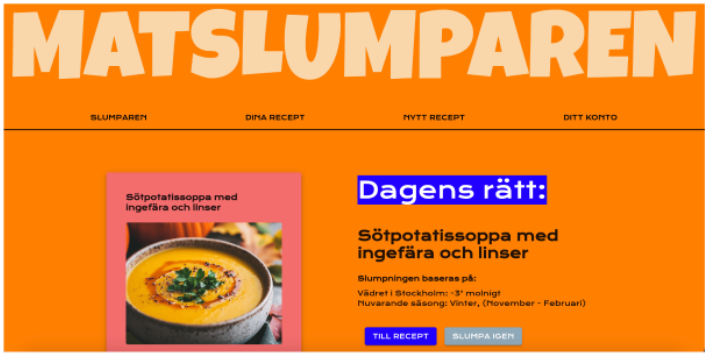
Wireframes
https://www.figma.com/design/dLATPW2Ci0ceR6BicNtzLu/Slutprojekt-planeringsapp?node-id=1-2&t=aEquxAZDYGpa34Lc-1

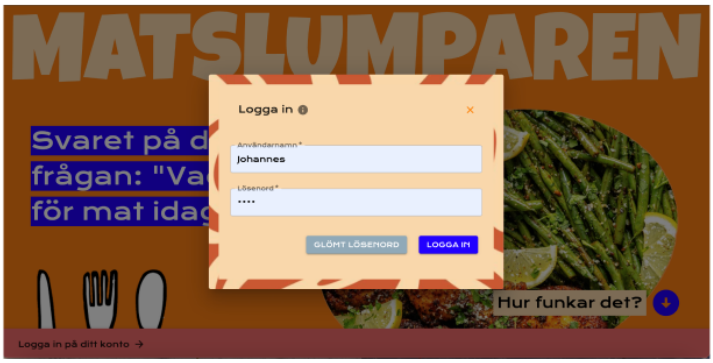
Flowchart
https://www.canva.com/design/DAG79o15r0s/HleoJbwb80kU1wxSbVGERw/edit

Some of the tools that will be used for testing accessibility:
https://a11y-tools.com
Google Chrome Lighthouse analysis
https://www.aremycolorsaccessible.com
https://www.a11yproject.com/checklist/
https://webaim.org/standards/wcag/WCAG2Checklist.pdf

Screenshots

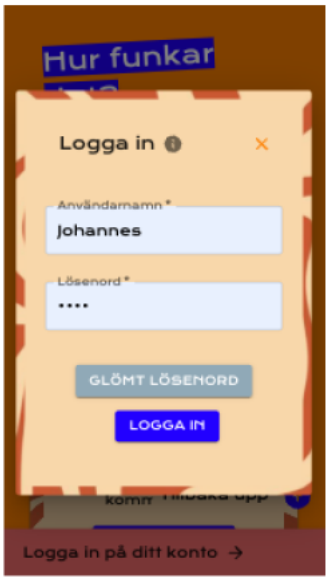
Desktop





Mobile

MATSLUMPAREN
Svaret på den eviga frågan: "Vad ska vi äta för mat idag"
Hur funkar det?
Logga in på ditt konto →


Hur funkar
Logga in
Användarnamn *
Johannes
Lösenord *
GLÖMT LÖSENORD
LOGGA IN
Logga in på ditt konto →

MATSLUMPAREN
Vad blir det för mat idag?
Stockholm: -3° molnigt
Vinter, (November - Februari)
SLUMPA!
Meny

MATSLUMPAREN
Belugabolognese
Säsong
Vinter, Höst, Vår
Väder
Snö, Molnigt, Regn, Vind
Temperatur
10° eller under
6 personer
45 min
Lika mustig som en vanlig bolognese men gjord på små svarta linser.
Dagens rätt:
Belugabolognese
Slumpningen baseras på:
Vädret i Stockholm: -3° molnigt
Nuvarande säsong: Vinter, (November - Februari)
TILL RECEPT
SLUMPA IGEN
Antal maträtter i din receptbank som uppfyller nuvarande väder och säsongkrav: 8
Se alla 8 recept
Meny

MATSLUMPAREN
Filtrera efter säsong
Vår | Sommar | Höst | Vinter | Alla
Filtrera efter väder
Soligt | Vind | Regn | Snö | Molnigt | Alla
Krämig Halloumigryta med linser
Meny

MATSLUMPAREN
Vad blir det för mat idag?
Stockholm: -3° molnigt
Vinter, (November - Februari)
Slumparen
Dina recept
Nytt recept
Ditt konto

MATSLUMPAREN
Johannes
Namn
Johannes Bäckelin
Ålder
30
E-post
johannes.backelin@google.com
← Logga ut
Meny

Matslumparen – Vad ska vi ät
localhost:5173/Matslumparen
MATSLUMPAREN
Svaret på den eviga frågan: "Vad ska vi äta för mat idag"
Hur funkar det?
Logga in på ditt konto →