|  |
| --- |
| Mindfull Memory  Team: LST3 |
| Analysis and  design report |
| Yussef Dalton,  Katoo Roofthooft,  Mikolaj Jodlowski,  Nariste Abdykadyrova,  Jana Vande Velde,  Thijs Lintermans |

Contents

[1. Introduction 3](#_Toc154058741)

[2. Assignment description 4](#_Toc154058742)

[2.1. Background information 4](#_Toc154058743)

[2.2. Objectives and target groups 4](#_Toc154058744)

[3. Requirements analysis 5](#_Toc154058745)

[3.1. Functional requirements 6](#_Toc154058746)

[3.1.1. Use Case Diagram 6](#_Toc154058747)

[3.1.2. Use Case Descriptions with prototypes 7](#_Toc154058748)

[3.2. Non-functional requirements 36](#_Toc154058749)

[4. Priority by functionality 37](#_Toc154058750)

# Introduction

Welcome to our report, where we'll introduce you to a special application designed to help young people with memory problems, particularly those dealing with young-dementia and other memory-related issues.

This report will provide you with some background information about the application, a use case diagram, and detailed use case descriptions. Besides that we are also going to explore the bigger picture, why we're doing this, and how it can make a real difference in people's lives.

As you dive deeper into this report, you'll discover more about how the app works, including detailed descriptions of how different parts of it will help our users. Together, we hope to create a solution that doesn't just make life easier but also brings comfort and support to those who need it most.

First, we will give you some background information and give you some insights into how things currently work and how it could work better with the app.

Then we will explain a little bit more about our target audience. We will explain who is going to use the app and what they can do with it. So you get a better understanding of the functionalities that the app will provide.

Afterwards, we will discuss the functional requirements with a use case diagram and descriptions. In the use case diagram, you can see who are the persons that are going to interact with the system (=actors), and what they can do (use cases). Besides that, we describe everything more in detail in the use case description. In these descriptions, we will explain the normal flow and the possible alternatives that might be necessary for the app.

# Assignment description

In this section, you can find more information about what we are doing. You can also find out who we are doing it for and why we are doing it.

## Background information

The application we are going to develop is designed to help young people with memory problems, particularly those dealing with young dementia and other memory-related issues.

Our client, for whom this analysis and design are being carried out for, is seeking a new system due to the limitations of the current manual methods of scheduling and tracking tasks and appointments.

Currently, our client has to rely on tools like Google Calendar, notepad, ToDo-apps,... All these applications are very good, but they all are separate apps and don’t function together. For example, they cannot easily check off completed tasks or appointments, and there is no mechanism for storing and reviewing questionnaire responses related to medication side effects before a doctor's appointment. It is also not possible to make sure only the people who need to see certain things, can see them.

## Objectives and target groups

The target audience of this application are people (=patients) who have a degenerative disease, such as dementia, but are much younger than the typical age group, such as 18- to 50-year-olds.

In addition, these patients live with friends or family who need to take care of these individuals, who are referred to as “informal carers”. As well as those, these patients might also have a doctor with whom a patient will have regular visits. The doctors prescribe medication for these patients, and it is critical that negative side effects of the medication be reported, so that the doctor can adjust the dosage or change the medication. If the doctor does not know this information, the consequences could be disastrous.

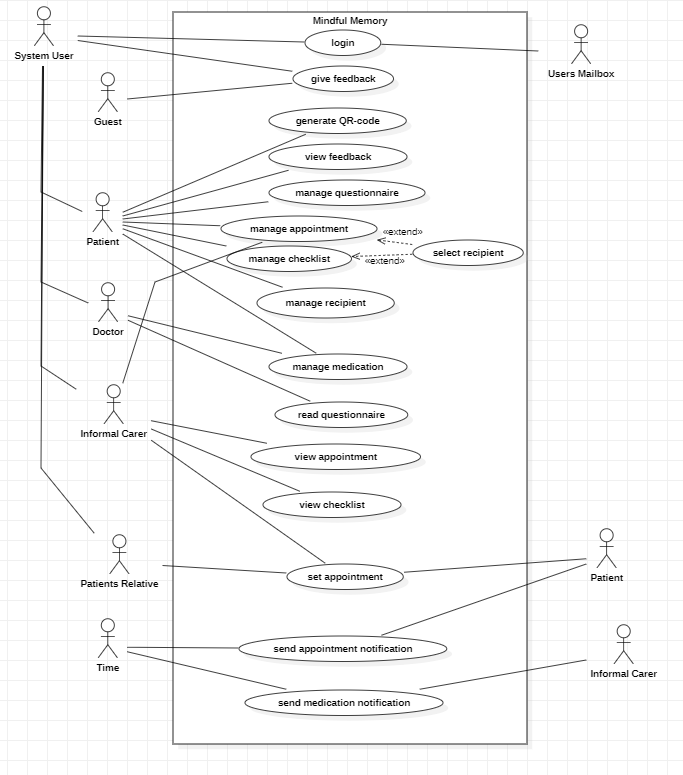
The primary objective of this application is to significantly improve the quality of these individuals with memory problems. We aim to provide a solution that streamlines their daily routines, helps manage appointments, tasks, and medication-related activities, and facilitates communication with healthcare professionals.

# Requirements analysis

In this section we will explain the functional requirements for our application. First we will show you the Use Case Diagram, here you can see who are the persons interacting with the system (actors) and what they can do (use cases). Besides that we are also going to explain the use cases more in detail, we will do this with the use case descriptions. In the use case description you can read more about the normal flow/alternatives of a use case.

## Functional requirements

### Use Case Diagram



### Use Case Descriptions with prototypes

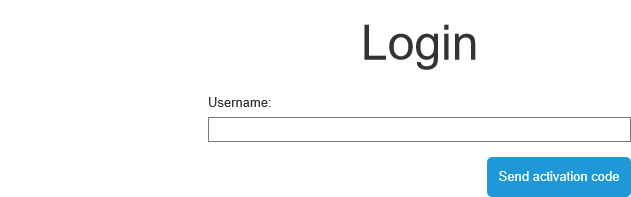
#### Login

Functionality: As a user, I can login.

Normal flow: The system shows the login page. The actor enters in his username. The system sends an activation code to the patient's mailbox. The actor enters the activation code and logs in.

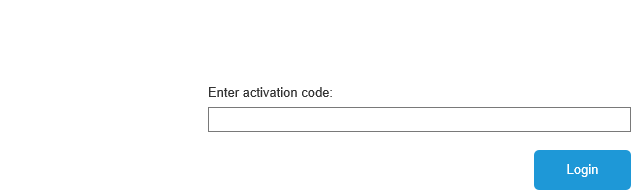
Screens:

1\_UC Login



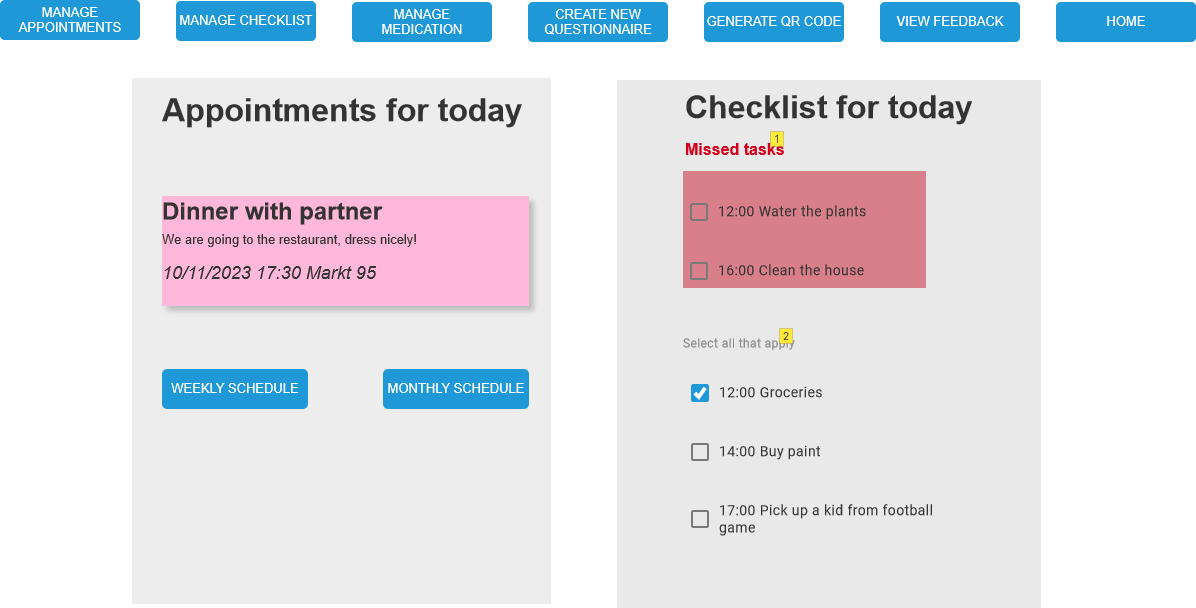
|  |  |
| --- | --- |
| **Action** | **Explanation** |
| Send activation code | Sends the activation code to the user’s mailbox, depending on if it is a patient, patient’s relative, Informal carer, doctor, or guest. |

1.1\_UC login



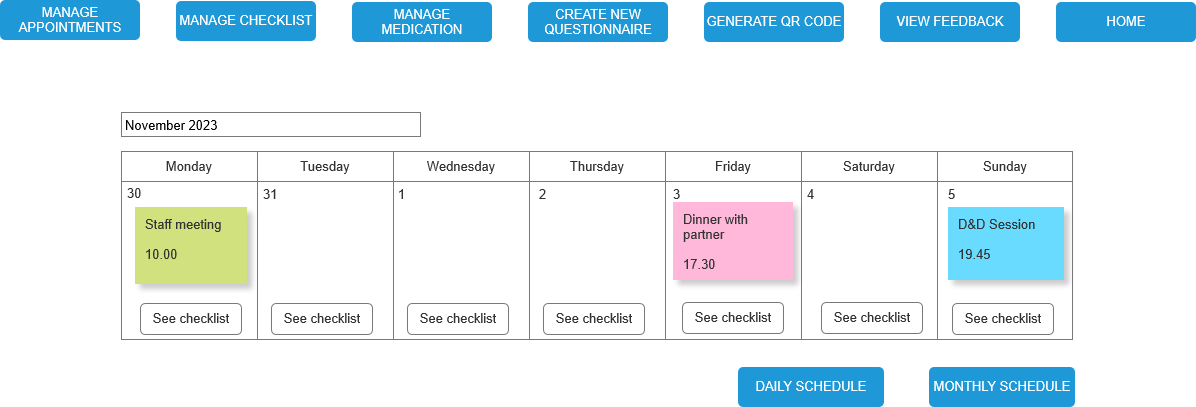
|  |  |
| --- | --- |
| **Action** | **Explanation** |
| Login | Redirects to one of the several homepages, depending on if it a patient, patient’s relative, informal carer, doctor, or guest. |

0\_Homepage patient daily



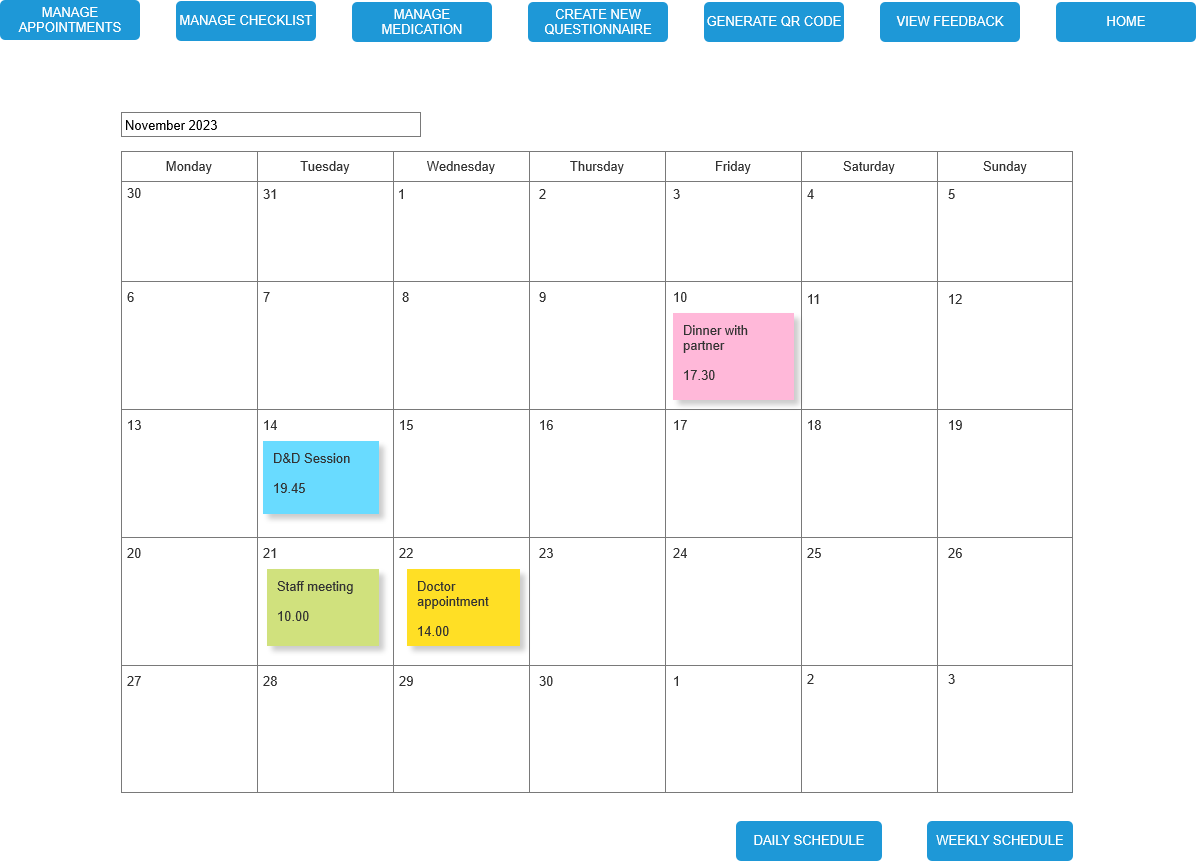
|  |  |
| --- | --- |
| **Action** | **Explanation** |
| Manage appointments | To 6\_UC Manage appointments |
| Manage checklist | To 7\_UC Manage Checklist |
| Manage medication | To 9.2\_UC Patient Manage Medication |
| Create new questionnaire | To 5\_UC Fill in questionnaire |
| Generate QR code | To 3\_UC Generate QR-code |
| View feedback | To 4\_UC view feedback |
| Home | To 0\_Homepage\_Patient Daily |
| Weekly schedule | To 0.0.1\_Homepage Patient Weekly |
| Monthly schedule | To 0.0.2\_Homepage Patient Monthly |
| 1 | Check missed task |
| 2 | Check or uncheck task for current day |

0.0.1\_Homepage Patient Weekly



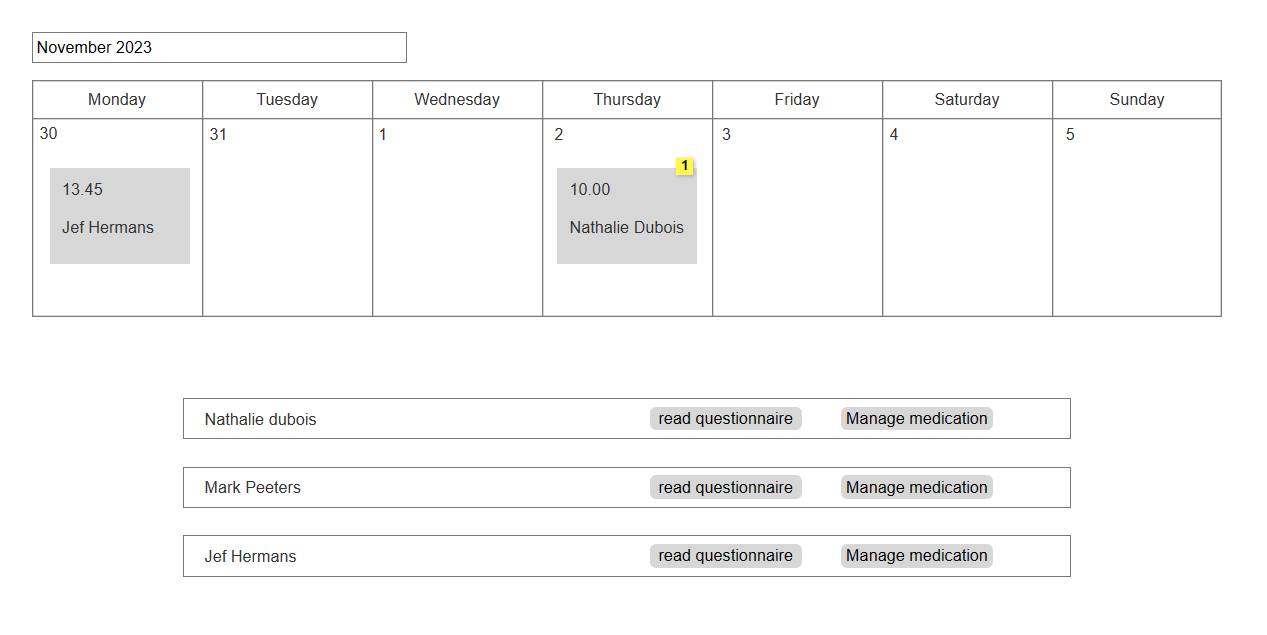
|  |  |
| --- | --- |
| **Action** | **Explanation** |
| Daily schedule | To 0\_Homepage Patient Weekly |
| Monthly schedule | To 0.0.2\_Homepage Patient Monthly |

0.0.2\_Homepage Patient Monthly



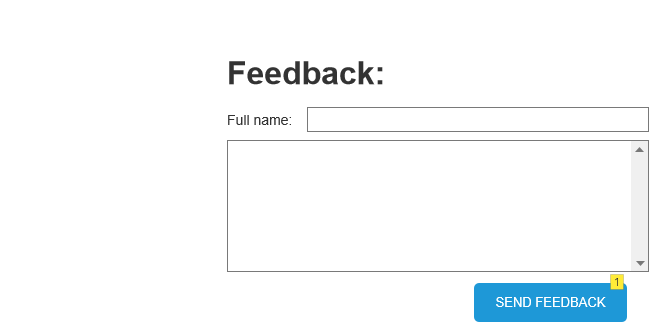
|  |  |
| --- | --- |
| **Action** | **Explanation** |
| Daily schedule | To 0\_Homepage\_Patient Daily |
| Weekly schedule | To 0.0.1\_Homepage Patient Weekly |

0.1\_Homepage Doctor



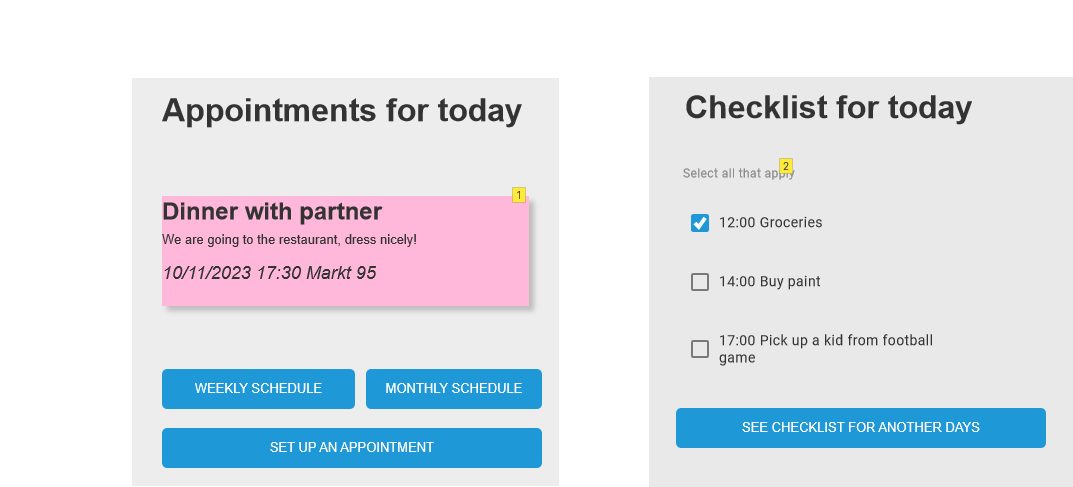
|  |  |
| --- | --- |
| **Action** | **Explanaition** |
| 1 | Appointment overview |
| Read questionnaire | To 10\_UC read questionnaire |
| Manage medication | To 9.1\_UC Doctor Manage medication |

0.2\_Homepage Guest

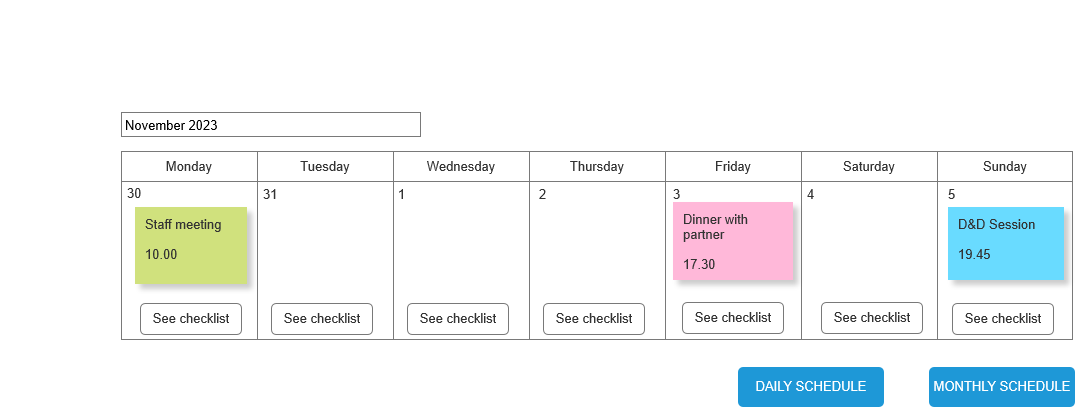


|  |  |
| --- | --- |
| **Action** | **Explanation** |
| 1 | This will send the feedback to the patient's view feedback page. |
| Send Feedback | Shows that your feedback has been sent. |

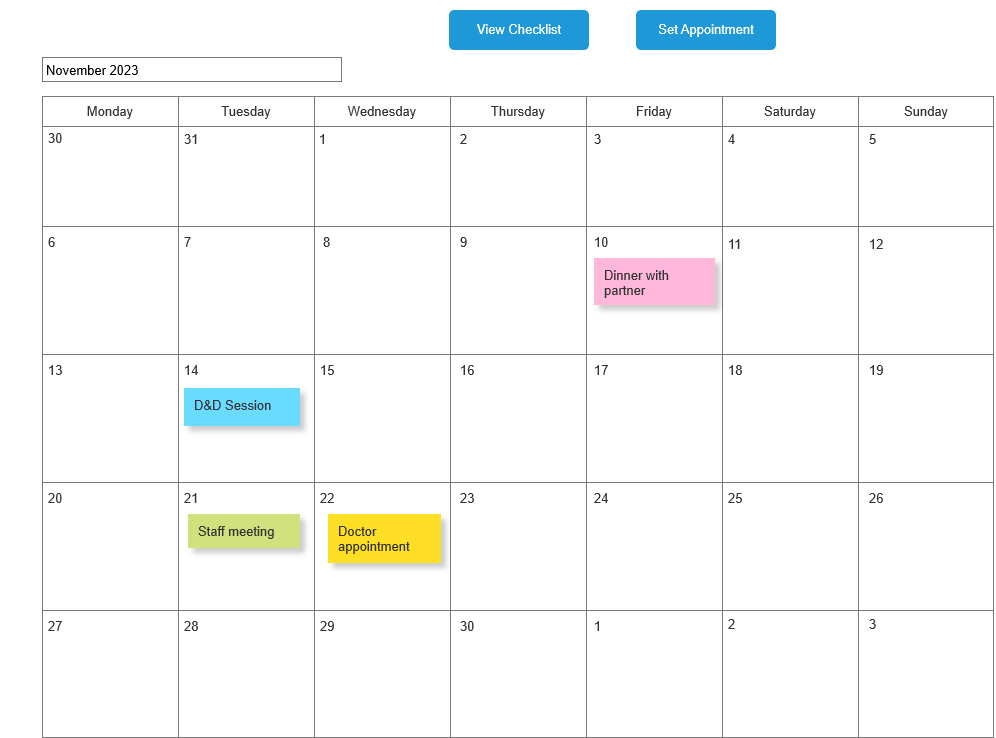
0.3\_Homepage Informal Carer Daily



|  |  |
| --- | --- |
| **Action** | **Explanation** |
| Weekly schedule | To 0.3.1\_Homepage Informal Carer Weekly |
| Monthly schedule | To 0.3.2\_Homepage Informal Carer Monthly |
| Set up an appointment | To 6.5\_UC Manage appointment |
| See checklist for another days | To 7\_UC Manage checklist |
| 1 | See appointment for the current day |
| 2 | Check or uncheck tasks for the current day |

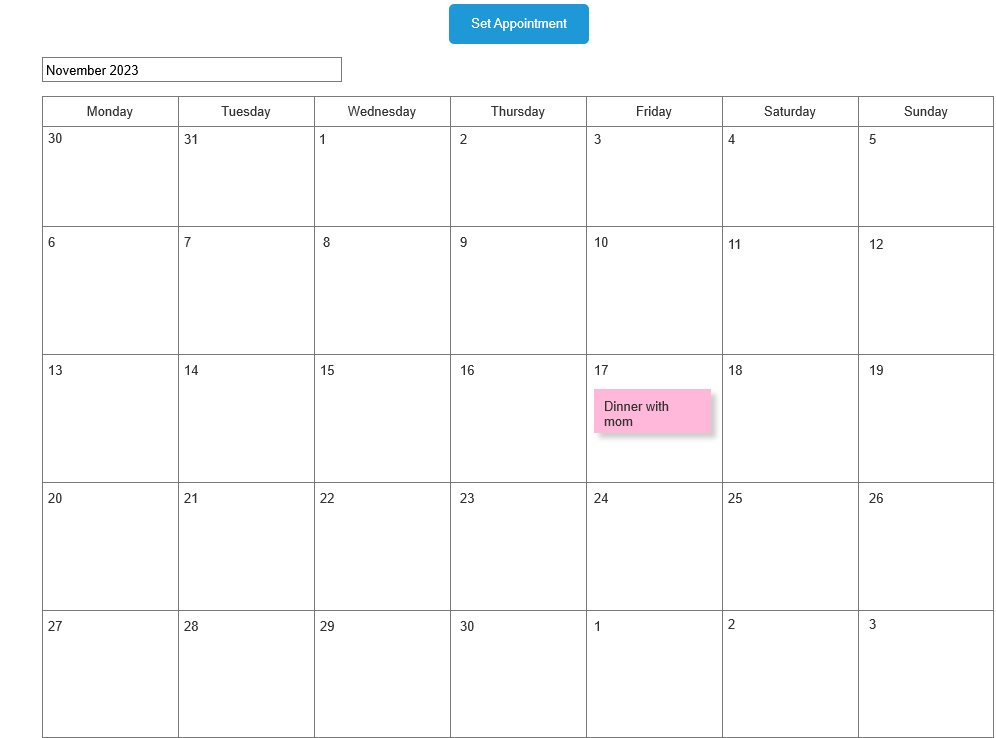
0.3\_Homepage Informal Carer Weekly

|  |  |
| --- | --- |
| **Action** | **Explanation** |
| See checklist | To 12.1\_UC View Checklist |
| Daily schedule | To 0.3\_Homepage Informal Carer Daily |
| Monthly schedule | To 0.3.2\_Homepage Informal Carer Monthly |

0.3\_Homepage Informal Carer Monthly

|  |  |
| --- | --- |
| **Action** | **Explanation** |
| View Checklist | To 12\_View Checklist |
| Set Appointment | To 13\_Set Appointment |

0.4\_Homepage Patient Relative



|  |  |
| --- | --- |
| **Action** | **Explanation** |
| Set Appointment | To 13\_Set Appointment |

#### Give feedback

Functionality: As a user, I can give feedback.

As a guest, I can give feedback

Precondition: The actor must be logged in.

Normal flow: The system shows an area where the actor can fill in their feedback. The actor writes his feedback and submits it.

Screens:

2\_UC Give feedback



|  |  |
| --- | --- |
| **Action** | **Explanation** |
| Send Feedback | Shows that your feedback has been sent. |

#### Generate QR-code

Functionality: As a patient, I can generate a QR-code.

Precondition: The actor must be logged in

Normal flow: The system shows a screen with a button to generate the QR code. The actor presses the button generate. The system will show the generated QR code.

Screens:

1.1\_UC login

A screenshot of a computer

Description automatically generatedA qr code on a white background

Description automatically generated

|  |  |
| --- | --- |
| **Action** | **Explanation** |
| Generate | The generated QR code will be shown in the grey area. |

#### View feedback

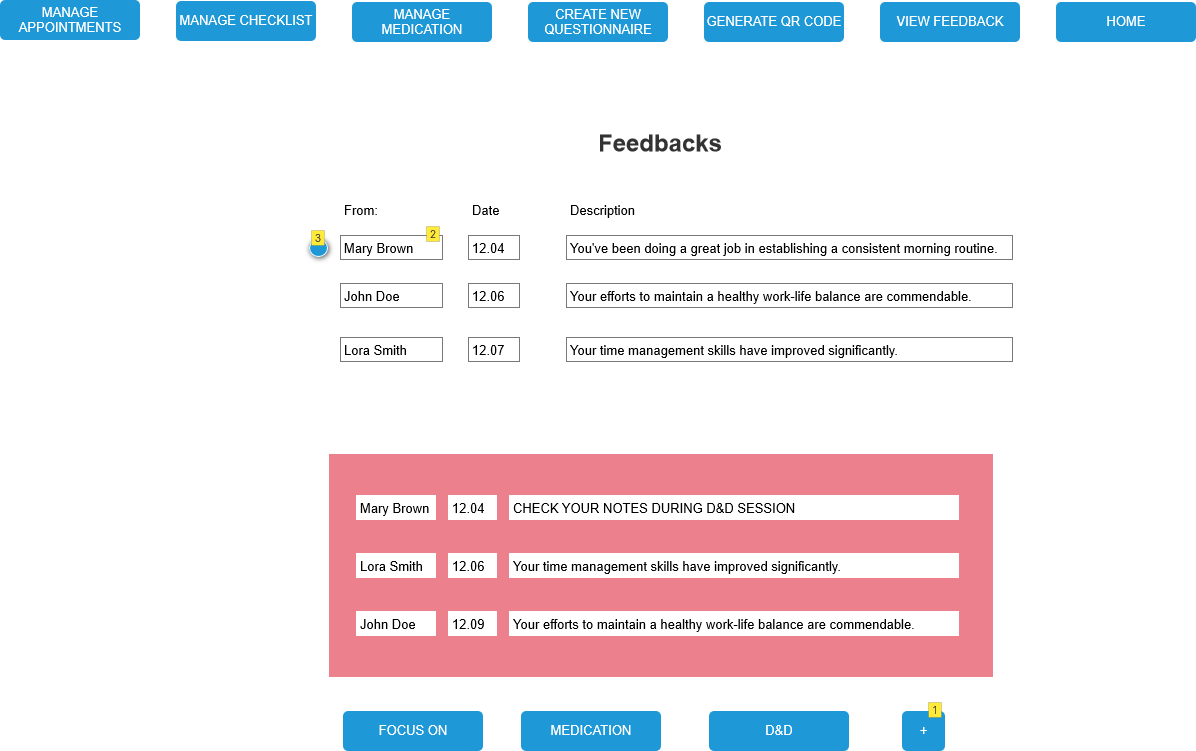
Functionality: As a patient, I can view feedback.

Precondition: The actor must be logged in.

Normal flow: The system shows al the feedback given with the name of who wrote the feedback and what date they have written it and categorise them.

Screens:

4\_UC View feedback



|  |  |
| --- | --- |
| **Action** | **Explanation** |
| Feedback (2) | User can drag and drop them in the box below |
| 3 | Indicates new feedback |
| 1 | Add new category |

#### Fill in questionnaire

Functionality: As a patient, I can fill in a questionnaire.

Precondition: The actor must be logged in.

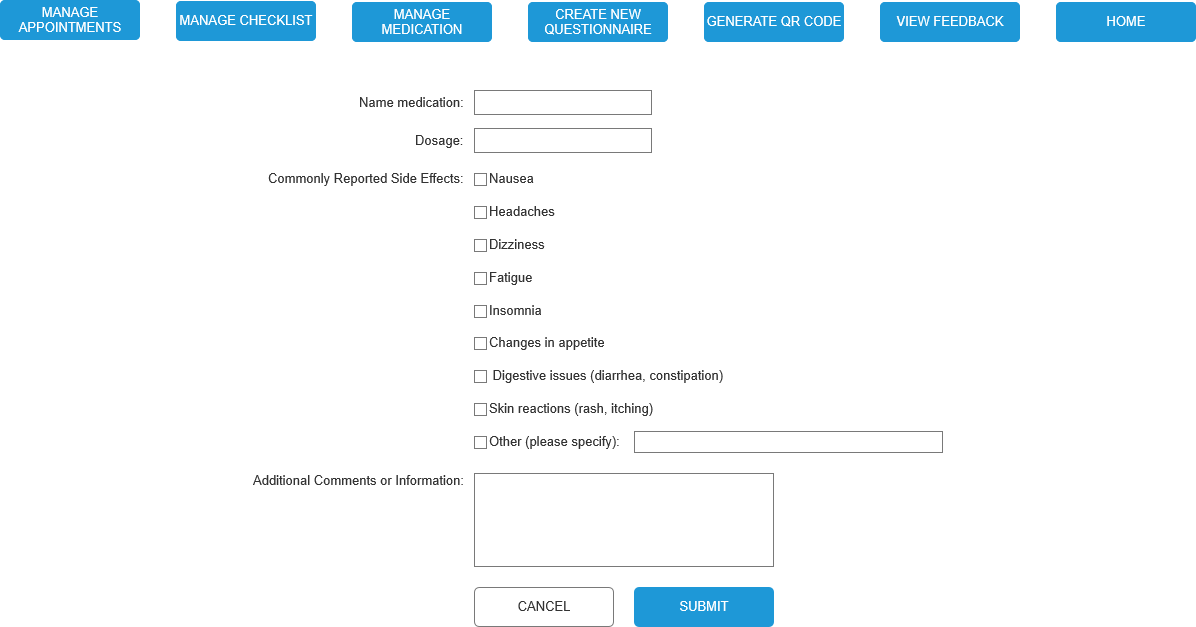
Normal flow: The system shows a list of questions to be answered (=questionnaire). The actor answers the questions and submits the questionnaire.

Alternatives:

* Change questionnaire : The actor chooses a questionnaire. The system shows the filled-in questionnaire. The actor changes the details of the questionnaire and confirms.

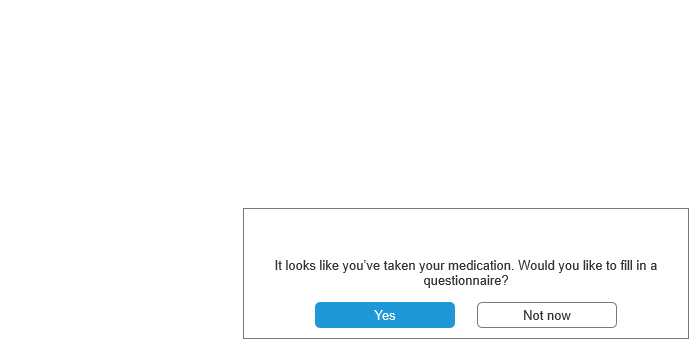
Screens:

5\_UC Fill in questionnaire



|  |  |
| --- | --- |
| **Action** | **Explanation** |
| Submit | Save questionnaire and return to 0\_Homepage patient daily |
| Cancel | to 0\_Homepage patient daily |

16\_UC Questionnaire notification



|  |  |
| --- | --- |
| **Action** | **Explanation** |
| Yes | Goes to 5\_UC Fill in questionnaire |

#### Manage appointments

Functionality: As a patient, I can manage appointments.

Precondition: The actor must be logged in.

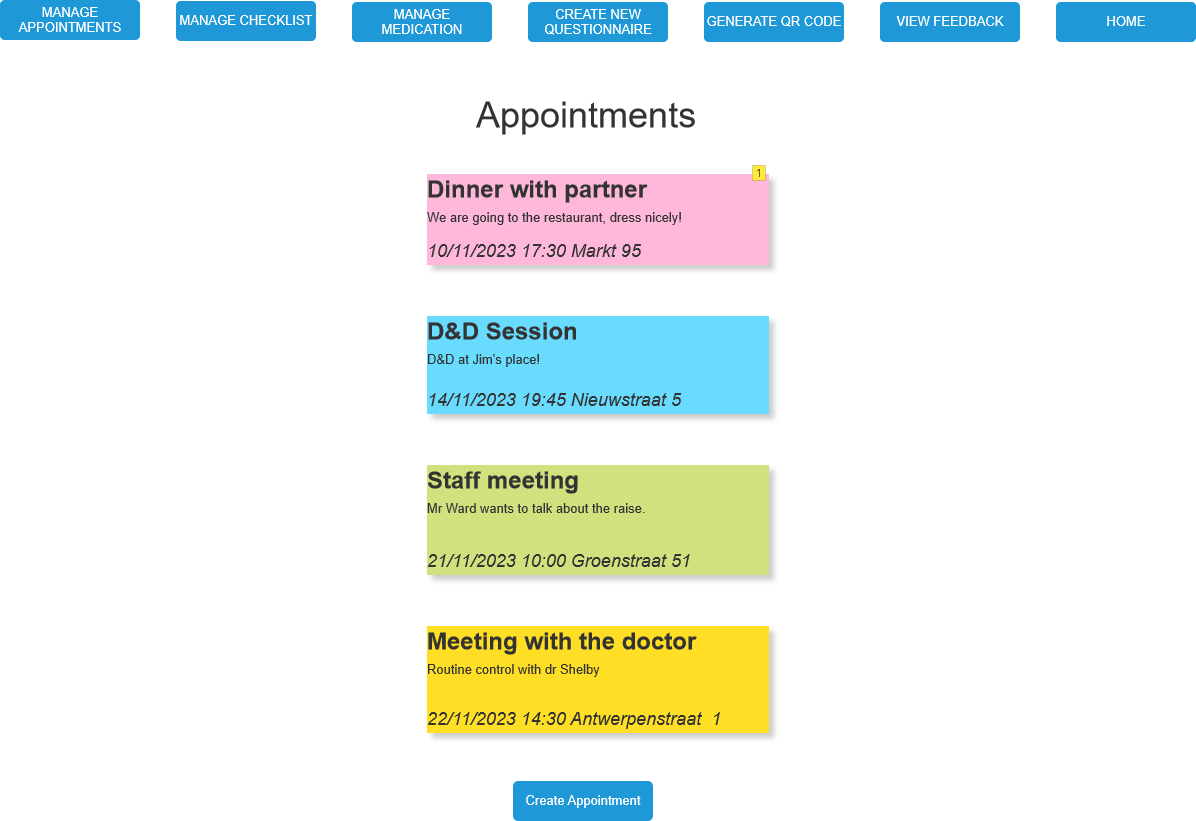
Normal flow: The system shows a list about the settings of the appointment. The actor fills in the settings and confirms. The system shows the appointment in the calendar.

Alternatives:

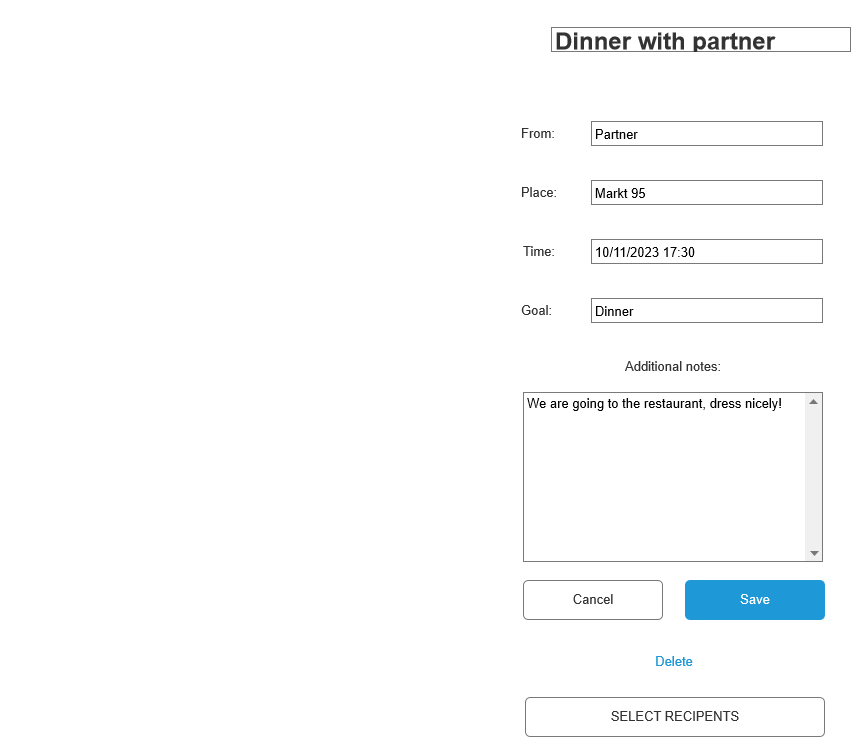
* Delete an appointment: The actor can select an appointment and the system will ask the actor to confirm the appointment that will be deleted.
* Changing an appointment: The actor can select an appointment and the system will show the settings of the appointment. The actor can change the settings and confirm.
* Select recipients: An actor can choose who has a view into his list of appointments. The system allows chosen people to see the list of appointments.

Screens:

6\_UC Manage appointment



|  |  |
| --- | --- |
| **Action** | **Explanation** |
| 1 | The system opens the settings of this appointment. |
| Dinner with partner | To 6.1\_UC Manage appointment page |
| D&D Session | To 6.2\_UC Manage appointment page |
| Staff meeting | To 6.3\_UC Manage appointment page |
| Meeting with the doctor | To 6.4\_UC Manage appointment page |
| Create Appointment | To 6.5\_UC Manage appointments page |

6.1\_UC Manage appointment

|  |  |
| --- | --- |
| **Action** | **Explanation** |
| Cancel | To 6\_UC Manage appointments page |
| Save | To 6\_UC Manage appointments page |
| Delete | To 6\_UC Manage appointments page |
| Select Recipients | To 8.1\_UC Select recipient page |

#### Manage checklist

Functionality: As a patient, I can manage my checklist.

Precondition: The actor must be logged in.

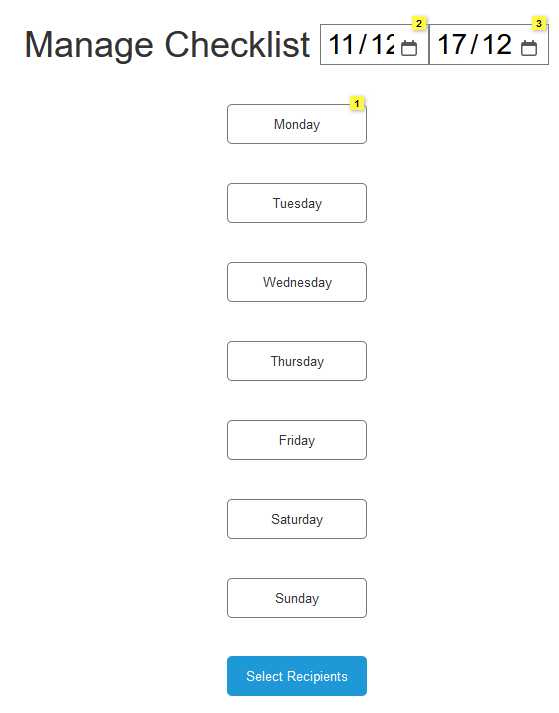
Normal flow: The system shows a checklist. The actor checks all the items from the checklist. The system deletes the pop-ups from the screen.

Alternatives:

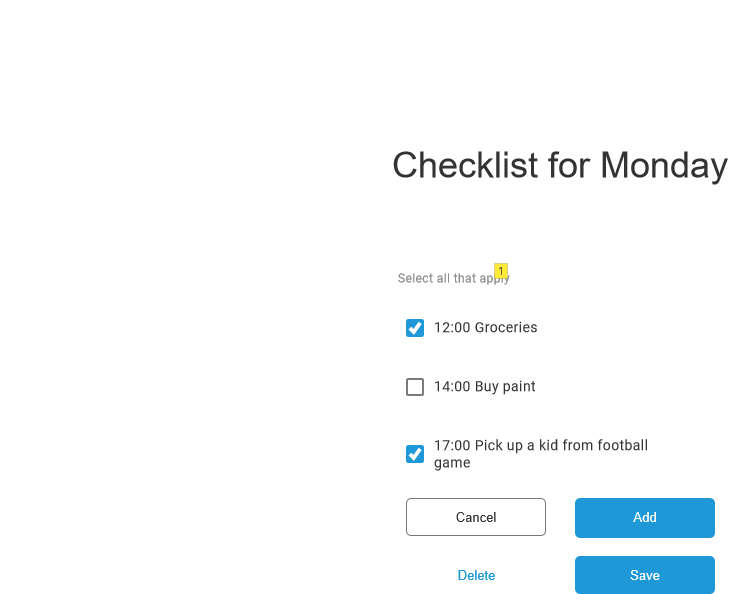
* Select recipients: An actor can choose who has a view into his checklist. The system allows chosen people to see the checklist.

Screens:

7\_UC Manage checklist



|  |  |
| --- | --- |
| **Action** | **Explanation** |
| 1 | When you press on one of the days it shows you the checklist for that day. |
| 2 | Choose a starting date |
| 3 | Choose an ending date |
| Select Recipients | To 8.1\_UC Select recipient |

7.1\_UC Manage checklist

|  |  |
| --- | --- |
| **Action** | **Explanation** |
| 1 | A list with all the things to do on this particular day. |
| Cancel | To 7\_UC Manage checklist |
| Save | To 7\_UC Manage checklist |
| Delete | Deletes an item from the checklist |
| Add | Add an item to the checklist |

#### Select recipients

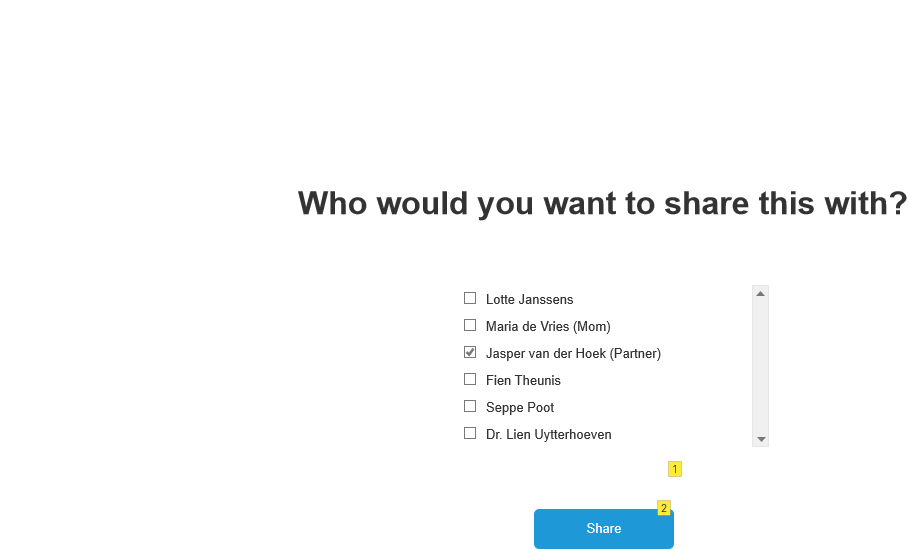
Functionality: As a patient, I can select recipients.

Precondition: The actor must be logged in.

Normal flow: System provides a list of recipients that the actor can share with. Actor selects and submits choice.

Screens:

8.1\_UC Select recipient



|  |  |
| --- | --- |
| **Action** | **Explanation** |
| 2 | Shares the activity (e.g. appointment) with the selected recipients. Redirects to the homepage. |

#### Manage recipients

Functionality: As a patient, I can manage recipients.

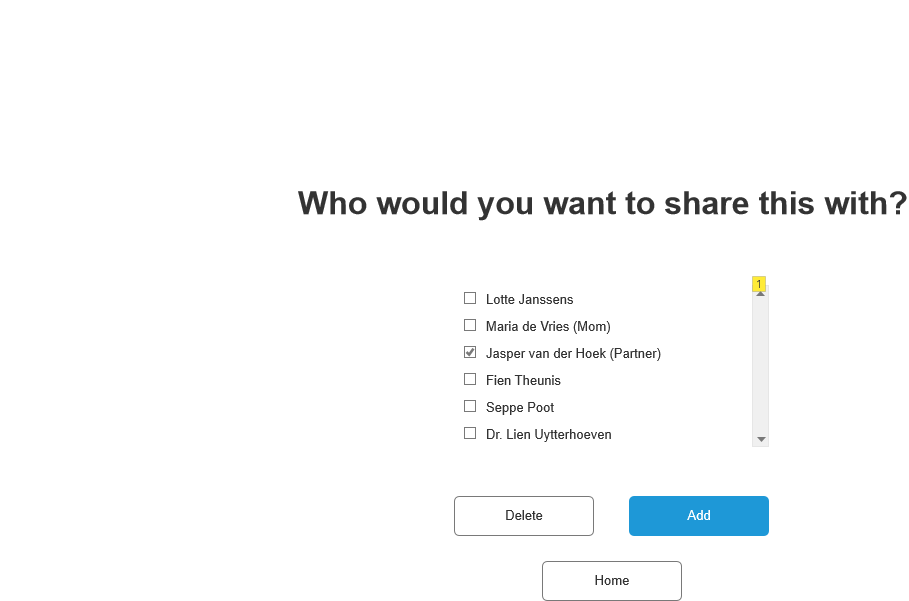
Precondition: The actor must be logged in.

Normal flow: System provides a list of recipients. The actor creates a recipient.

Alternatives:

* Delete recipient: System provides a list of recipients. The actor chooses a recipient they want to delete. The system deletes the recipient.

Screens:

8.0\_UC Manage recipient

|  |  |
| --- | --- |
| **Action** | **Explanation** |
| 1 | List with all the recipients. |
| Home | To 8.1\_UC Select recipient |
| Delete | Delete a person from the list. |
| Add | Add a person to the list. |

#### Manage medication

Functionality: As a doctor, I can manage medication.

As a patient, I can manage medication.

Precondition: The actor must be logged in.

Normal flow: The system displays a list of medication. The actor chooses the medication according to patient’s health status.

Alternatives:

* Change medication: The system shows the list of medication. The actor chooses the medication that has to be changed. The system shows the details of the medication. The actor changes the details.

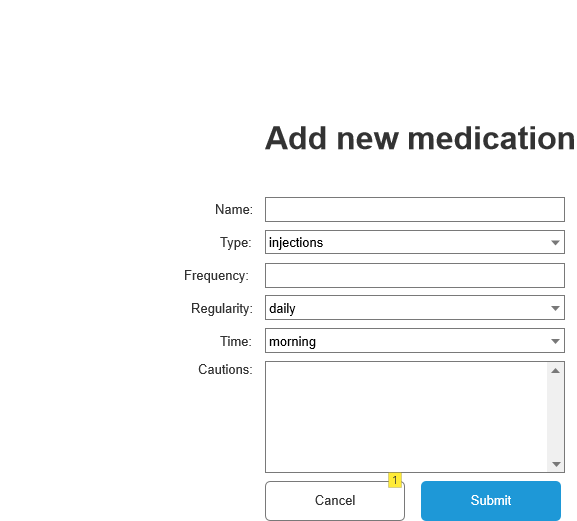
Screens:

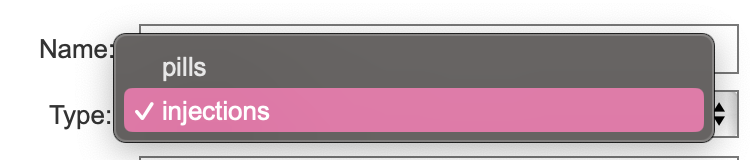
9.1\_UC Doctor Manage medication

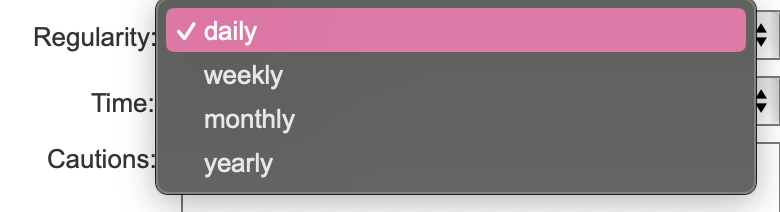


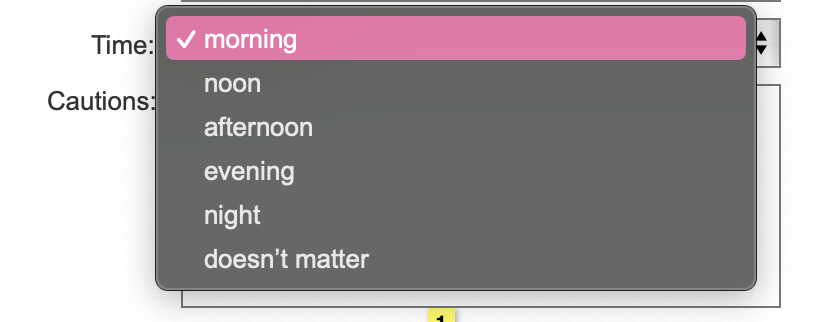
|  |  |
| --- | --- |
| **Action** | **Explanation** |
| 1 | This shows the cautions that the doctor has laid out for taking this specific medication, e.g. that you shouldn't take this piece of medication while drinking alcohol, for example. |
| 2 | Saves changes to the medication and goes to the homepage |
| 3 | Goes to 9.1.1\_UC Doctor Manage medication |

9.1.1\_UC Doctor Manage medication



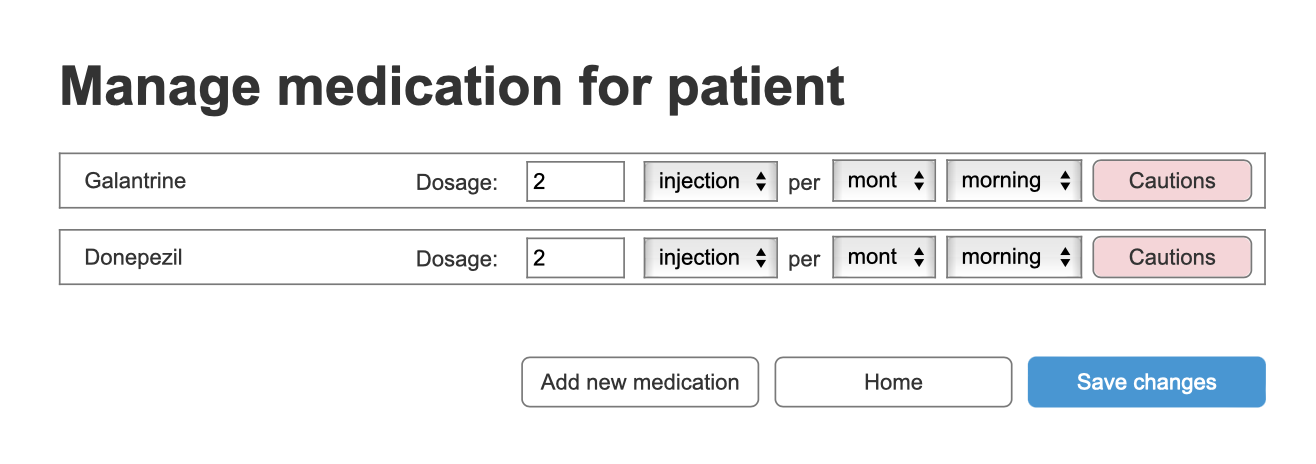




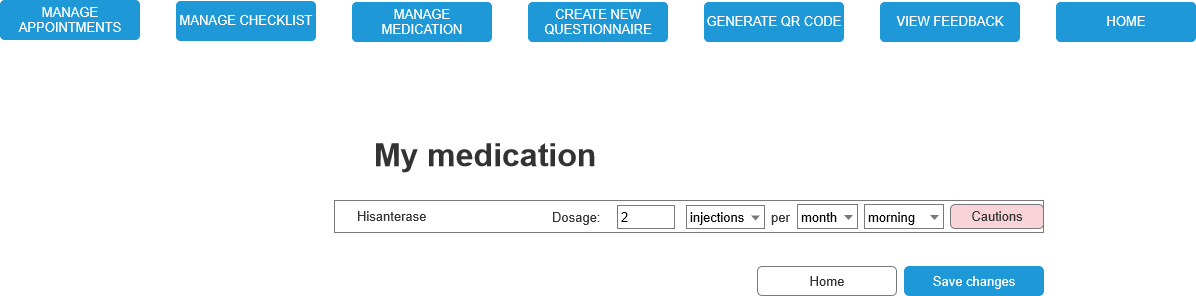


|  |  |
| --- | --- |
| **Action** | **Explanation** |
| 1 | Goes back to 9.1\_UC Doctor Manage medication |
| Submit | Adds new medication for patients. System notifies the patient via email |

9.1.2\_UC Doctor Manage medication



9.2\_UC Patient Manage medication



#### Read questionnaire

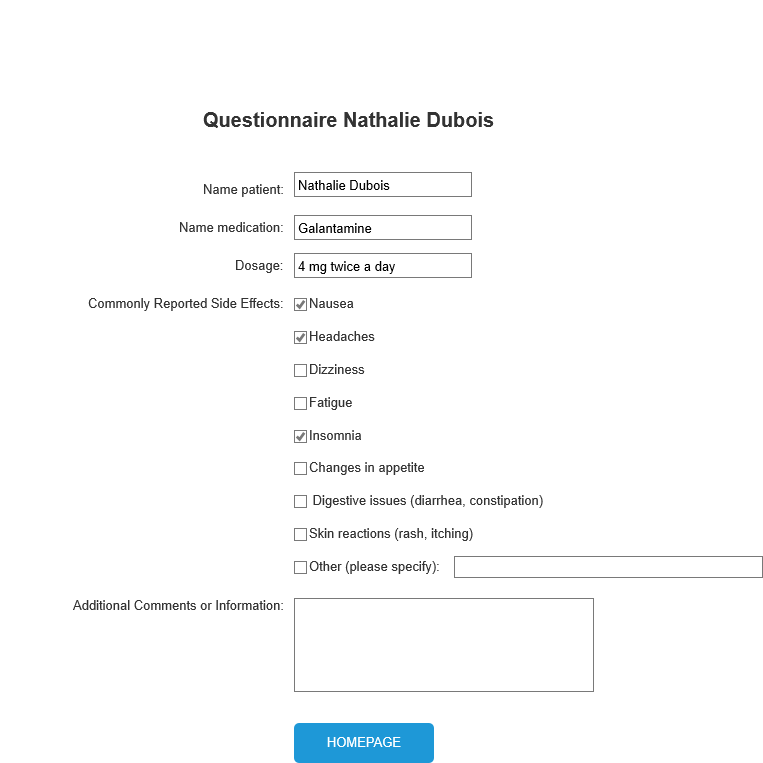
Functionality: As a doctor, I can read questionnaires.

Precondition: The actor must be logged in. The patient has submitted the questionnaire.

Normal flow: The system shows the patient’s previous questionnaires.

Screens:

10\_UC read questionnaire



|  |  |
| --- | --- |
| **Action** | **Explanation** |
| Homepage | To homepage doctor |

#### Send medication notification

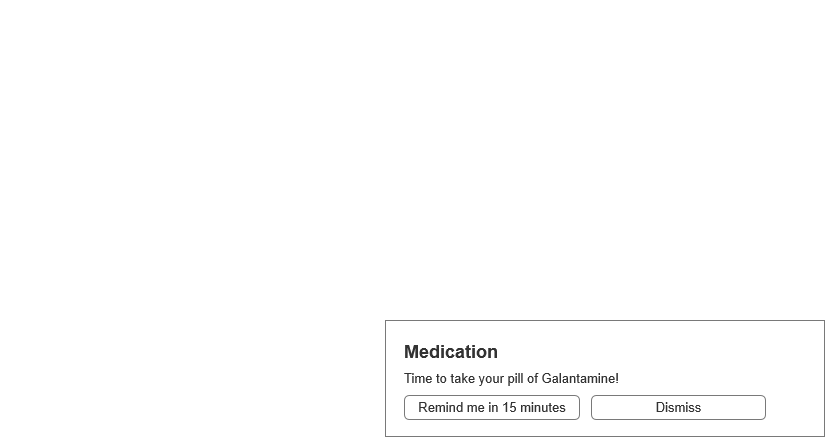
Functionality: As a time, I can send medication notifications.

Precondition: The actor must be logged in.

Normal flow: The system gives the actor a notification that the medication must be taken in. The actor forgets to take the medication and to delete the notification. The system sends a notification to the informal carer after one hour of not taking the medication.

Screens:

15.1\_UC Send medication notification



|  |  |
| --- | --- |
| **Action** | **Explanation** |
| Dismiss | Dismisses the notification |
| Remind me in 15 minutes | This same message will reappear in 15 minutes. |

#### Send appointment notification

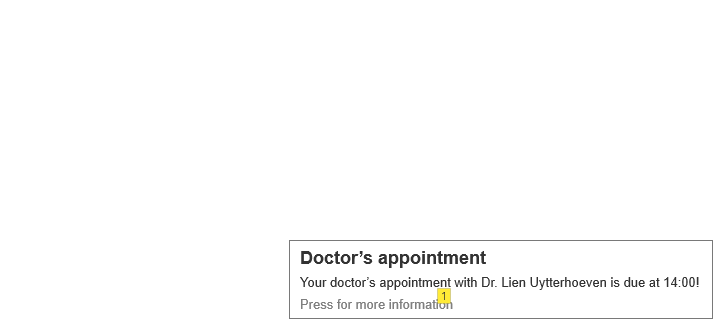
Functionality: As a time, I can send appointment notifications.

Precondition: The actor must be logged in.

Normal flow: The system gives the actor a notification that there is an appointment in one hour.

Screens:

14\_Send appointment notification



|  |  |
| --- | --- |
| **Action** | **Explanation** |
| 1 | To 14.1\_UC Send appointment notification |

#### 14.1\_Send appointment notification



#### View appointments

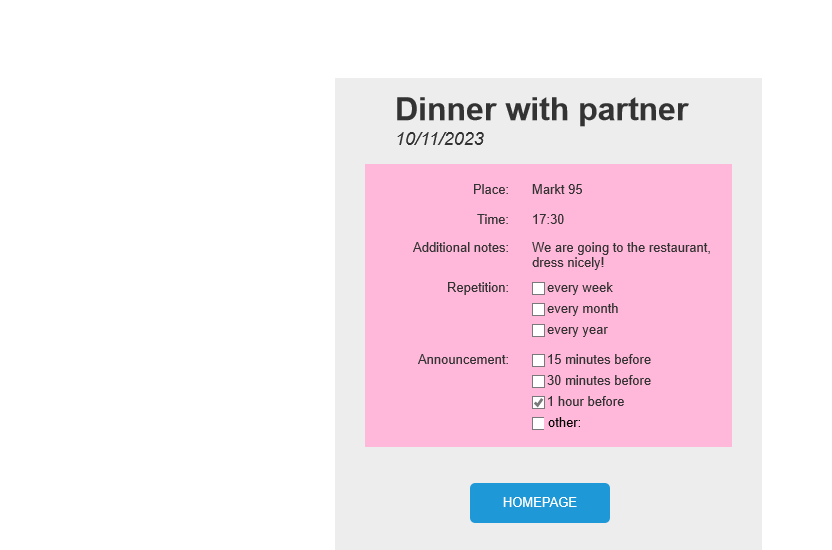
Functionality: As an informal carer, I can view appointments.

Precondition: The actor must be logged in.

Normal flow: The system shows a list of patient appointments. The actor presses on an appointment. The system shows the pressed appointment.

Screens:

11\_UC View appointment



|  |  |
| --- | --- |
| **Action** | **Explanation** |
| Homepage | To homepage informal carer |

#### View checklist

Functionality: As an informal carer, I can supervise checklists.

Precondition: The actor must be logged in.

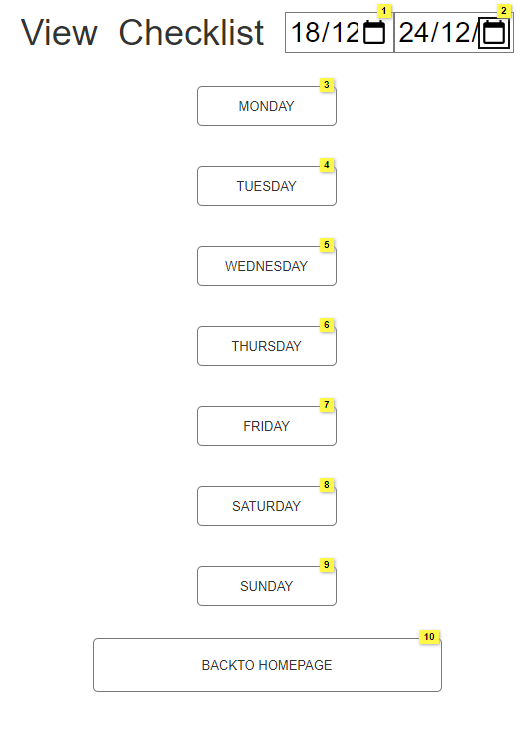
Normal flow: The system displays the patient’s checklist. The actor checks if everything is checked off and submits the checklist.

Alternatives:

* Send notification: The system displays the patient’s checklist. The actor sends a notification to the patient if something is not checked off.

Screens:

12\_UC View checklist



|  |  |
| --- | --- |
| **Action** | **Explanation** |
| 1 | Choose a starting date (Monday) |
| 2 | Choose an ending date (Sunday) |
| 3 | View checklist for Monday |
| 4 | View checklist for Tuesday |
| 5 | View checklist for Wednesday |
| 6 | View checklist for Thursday |
| 7 | View checklist for Friday |
| 8 | View checklist for Saturday |
| 9 | View checklist for Sunday |
| 10 | Goes to 0.3.1\_Homepage Informal Carer Weekly |

12.1\_UC View checklist

|  |  |
| --- | --- |
| **Action** | **Explanation** |
| 1 | Check or uncheck specific task on the checklist |
| 2 | Go back to 12\_UC View checklist |
| 3 | Go back to 0.3.1\_Homepage Informal Carer Weekly |

#### Set appointments

Functionality: As an informal carer, I can set an appointment.

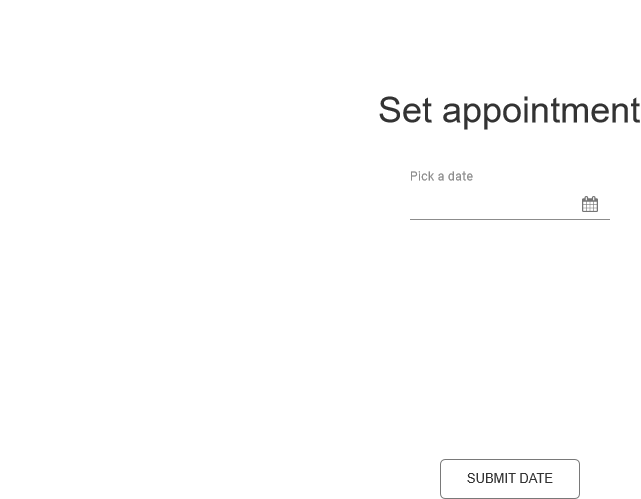
As a patient’s relative, I can set an appointment

Precondition: The actor must be logged in.

Normal flow: The system shows a list of the settings. The actor fills in the right settings. The system sends a notification to the patient that an appointment has been made.

Screens:

13\_UC Set appointment



|  |  |
| --- | --- |
| **Action** | **Explanation** |
| Submit date | To 13.1\_UC Set appointment |

A screenshot of a computer

Description automatically generated13.1\_UC Set appointment

|  |  |
| --- | --- |
| **Action** | **Explanation** |
| Submit | To 0.3\_Homepage Informal Carer Daily |
| Cancel | To 13\_UC Set appointment |

## Non-functional requirements

#### Implementation

This application will be implemented in the PHP programming language using the Laravel framework together with Livewire. We will also be using the MySQL relational database management system. For authentication and access control, we will be using Jetstream which works alongside the Laravel framework. For the user facing portion of the application, we will be using Tailwind components as well for Alpine.js for interactivity.

#### Performance

**Response time:**

The system's responsiveness to specific events is a critical aspect of its performance. Our goal is to achieve optimal response times, defining the acceptable duration for the system to react to particular events.

**Scalability:**

Scalability is a critical aspect of our application. We're designing the system to handle a defined number of simultaneous actions, including users scheduling appointments, filling in medication questionnaires, and managing task lists concurrently.

In cases of unexpected surges, such as a sudden increase in appointment requests or users concurrently engaging with medication questionnaires, the application is built to gracefully scale resources dynamically. This ensures optimal performance without compromising the user experience.

The scalability considerations extend to seamless handling of concurrent interactions with task lists and questionnaires. The system efficiently manages increased loads during peak periods, guaranteeing a responsive and reliable experience. This proactive scalability approach aims to maintain optimal performance and user satisfaction even during peak usage.

#### Quality requirements

The app's design focuses on user-friendliness, featuring a simple and intuitive interface for easy navigation. Users can personalize reminders and notifications according to their preferences, providing a customized experience. Security is a top priority, with a strong emphasis on encryption and robust measures to safeguard sensitive user information, particularly concerning medication and personal data. Moreover, the app ensures accessibility even without an internet connection, allowing users to access essential features seamlessly in offline mode.

# Priority by functionality

In our development process, we have prioritized functionalities using the MoSCoW framework.

Must-haves, including login, medication, checklists for patients, and appointments, form the core and essential components for our client's minimum requirements.

Moving to should-haves, we are focusing on elements like notifications, questionnaires, and recipients. They are not critical, but they bring a level of enhancement to the user experience that we find valuable.

Exploring the could-haves, we have considered integrating QR code and feedback functionalities – think of them as the final changes that bring extra flair. These features add a touch of versatility and perfectly complement the core functionality without stealing the spotlight. Our goal is to make sure that the software not only covers the essentials, but also elevates the user experience and goes the extra mile to exceed expectations. These features are desirable but not critical.

Lastly, in the would-have category, we have considered the possibility of having a mobile application that works alongside the web application, but it would require writing effectively two different applications.