2023 - 2024

|  |
| --- |
| MINDSPARK  Team: LST2 |
| Analysis and  design report |
| Siebe Gios Julia Drygalska Abdul Salam Aldabik Myrthe Daniels Marcelina Siwka Ian Hoogstrate |

Table of contents

[1. Introduction 5](#_Toc1346497257)

[2. Assignment description 6](#_Toc1919351489)

[2.1. Background information 6](#_Toc1878297150)

[2.2. Objectives and target groups 6](#_Toc390715336)

[3. Requirements analysis 8](#_Toc110739458)

[3.1. Functional requirements 9](#_Toc1184709913)

[3.1.0. Use Case Diagram 9](#_Toc1293285352)

[3.1.1. Use Case Descriptions with prototypes 10](#_Toc1630295298)

[3.1.1.0. Login into account 10](#_Toc1259091106)

[3.1.1.1. Activate account 14](#_Toc1925884345)

[3.1.1.2. Change password 15](#_Toc55268793)

[3.1.1.3. Manage agenda. 17](#_Toc1400385385)

[3.1.1.4. Write a diary. 22](#_Toc1850023716)

[3.1.1.5. Read dairy 23](#_Toc1794796107)

[3.1.1.6. Read note. 25](#_Toc643324136)

[3.1.1.7. Register account 29](#_Toc1152799060)

[3.1.1.8. Fill out questionnaire. 30](#_Toc1013461781)

[3.1.1.9. Customize questionnaire. 33](#_Toc578240691)

[3.1.1.10. Give permissions. 34](#_Toc958701886)

[3.1.1.11. Add user 36](#_Toc12804171)

[3.1.1.12. Personalize home page 37](#_Toc1183875336)

[3.1.1.13. Manage to do list 38](#_Toc343356666)

[3.1.1.14. Manage routines 40](#_Toc313139928)

[3.1.1.15. Check location. 43](#_Toc383108449)

[3.1.1.16. Reminder medication 44](#_Toc317522949)

[3.1.1.17. Reminder appointments 45](#_Toc513446112)

[3.1.1.18. Add medication. 46](#_Toc2038331051)

[3.1.1.19. Show QRcode 42](#_Toc1996805820)

[3.1.1.20. Write note. 48](#_Toc973129489)

[3.1.1.21. Home page 49](#_Toc1632663312)

[3.1.1.22. Navigation 50](#_Toc1870594380)

[3.2. Non-functional requirements 52](#_Toc1673972972)

[3.2.0. Usability 52](#_Toc2013931371)

[3.2.1. Accessibility 52](#_Toc991267661)

[3.2.2. Performance 52](#_Toc1503369761)

[3.2.3. Reliability 52](#_Toc1410937257)

[3.2.4. Security 52](#_Toc369294030)

[3.2.5. Scalability 52](#_Toc991491453)

[3.2.6. Compatibility 52](#_Toc487286194)

[3.2.7. Adaptability 52](#_Toc1112570651)

[3.2.8. Documentation 53](#_Toc1513427493)

[4. Priority by functionality 54](#_Toc906287344)

[4.1. Login into account 54](#_Toc621374602)

[4.2. Activate account 54](#_Toc100843536)

[4.3. Change password 55](#_Toc1421737719)

[4.4. Manage agenda 55](#_Toc1087555888)

[4.5. Write a diary 56](#_Toc845048846)

[4.6. Read note 56](#_Toc1724965838)

[4.7. Register account 57](#_Toc1860864658)

[4.8. Fill out questionnaire 57](#_Toc897265675)

[4.9. Customize questionnaire 58](#_Toc1331311608)

[4.10. Give permissions 58](#_Toc1917659011)

[4.11. Add user 59](#_Toc676393958)

[4.12. Manage to do list 59](#_Toc1989097676)

[4.13. Add to do list 60](#_Toc1740816785)

[4.14. Check location 60](#_Toc1825024977)

[4.15. Reminder medication 61](#_Toc583481614)

[4.16. Reminder appointments 61](#_Toc1608453600)

[4.17. Add medication 62](#_Toc1915344171)

[4.18. Write note 62](#_Toc1910863164)

# Introduction

In an increasingly digital world, technology plays a pivotal role in enhancing the quality of life for individuals facing various challenges. One of these challenges is memory loss, which can significantly impact a person's ability to lead an independent and fulfilling life. So, we take on the challenge to develop an application for people with memory problems.

We are going to develop a web-based application designed to assist individuals with memory loss in managing their daily lives. This project is a testament to our commitment to leveraging technology for the betterment of society. We aim to provide an intuitive and user-friendly platform that aids those with memory problems in remembering crucial tasks, appointments, and cherished memories.

To ensure a clear understanding of the project's scope and functionality, this user requirements file has been created. Use case diagrams are an invaluable tool for visualizing and organizing the interactions between users and the system. In this document, we will explore the primary actors and their interactions with our system.

A core mission guides the project: to promote independence, reduce stress, and enhance the overall quality of life for individuals facing memory challenges. Through a combination of user-friendly interfaces, reminders, and memory-support features, our project aims to address the unique needs of this user group.

This user requirements specification serves as a foundational document to capture the essential functionalities and interactions about our project. It will help us, as well as client and people involved, better comprehend the application's functionality, anticipate user needs, and ensure the project aligns with our vision of delivering an inclusive and empathetic solution.

# Assignment description

In this section we will provide all the information we have gathered and heard about the assignment. The most crucial factor for us was the interview with the client. Although the information provided by our client through the interview was crucial, the insights we gained during our classes also significantly influenced our thinking. Within this section, we will provide the details both from the side of the client but also our own insight.

## Background information

tasks, meetings, and events. The first job she had ever done was a job at a charity dedicated to helping people, specifically with young dementia. That was her first encounter with this medical condition. Watching the daily struggles of the people she realized that there must be something to make it all easier for them. She also has memory problems but related to a different condition. During the interview she showed us how overwhelming it is for her to keep track of her daily activities. Sometimes it becomes hard because she must use a lot of apps to manage through the day.

Every application has its own, small purpose. For example, she has a different app for the calendar itself, another application for the checklist, daily/weekly planning, morning routines. For some people it might not sound so overwhelming but to people with memory problems it is a challenge. It is not convenient for the impacted person to check so many different apps. It would be much better to just have one app that interconnects everything together. During the interview we realized that it is an enormous problem that so many people must deal with daily, so taking into consideration their needs, suggestions, ideas is crucially important for our team.

## Objectives and target groups

An automated system would help the client and other people with memory problems to get through their day, without a lot of struggles. She mentioned to us that she easily gets stressed when she must use so many apps. Sometimes It can also happen that she just forgets to check one app which results in not having something done the way she wanted. So, our objective would be to create one calendar that combines everything she needs in one place.

Another significant improvement is that the doctor also has access to the application. His role is to control the health condition of the admin. He is supposed to decide when the admin needs new medication and easily prescribe it to them in an effortless way. So, the objective is to allow the doctor to create an account and easily convey the prescription to the admin. This way the doctor and the admin will be able to check everything with each other using the app, without real meetings Although, our client stated that it is especially important for her that he can only access the logs and questionnaires connected to her health and medication. She does not want everyone to be able to read her personal life’s matters. The application is supposed to also be kind of her “diary.” Reminding not only the important stuff but also some daily necessities that may not look that crucial at the first glance.

Our target group is everyone experiencing memory problems that can cause some distress during their daily life. With our application we would love to help the target group to feel at ease. Without proper organization and reminders, it is hard for them to keep a job, even though it is not actually dependent on them. That is why it sometimes becomes stressful and frustrating – even while trying their best at every step. They can still fail because of the condition that they have. There are a lot of people with young dementia but there is also a vast group of people that struggle with memory problems because of their pills and treatment. They should not be excluded from society and daily living. That is why this application would be a huge step towards normalizing their daily activities, appointments, and routines. Most of the target groups have an informal carer. The app will also be able to lessen some of the responsibilities for them.

The informal carers are often referred to as a family caregiver, so a person who provides unpaid care and support to a family member, friend or loved one who has, in our case, memory problems. Their role is to provide help to the admin, be sure that they do their tasks on time, that they do not forget important meetings. So, their objective is to see and help the admin to stay on track, which is why they would need access to the admin’s calendar. They also need a possibility to manage the calendar to help as well.

Our client also wants their friends to be able to leave a note for her. So, the main objective of the guest is to write down a log that later can be accessed by the admin themselves. The guest can be, for example: a friend from D&D group wanting to write a summary, that the admin will be able to read later.

So, we are creating an app for people experiencing memory problems that our client asked us to. We are listening to her advice and her life experiences, but the main point of the system is for everyone struggling with this spectrum.

# Requirements analysis

In this section we will finally focus on the functionalities of our project. After the first interview we already know the basic needs of our customer. According to everything we have researched and heard, it is time to specify those functionalities. First, we will provide our use case diagram we have created using starUML. We will also focus on functionalities, their core mission and meaning and why it is needed in the system we are creating. We will utilize case description to concentrate on every one of them and make it clear to understand.

## Functional requirements

### Use Case Diagram A diagram of a diagram Description automatically generated

### Use Case Descriptions with prototypes

Our functionalities names are bold and orange in the middle of the page. They tell something more about our use cases in the use case diagram above. The difference between admin and acquaintance is that we use the acquaintance to inherit all functionalities that the admin and informal can both do. The admin can still do extra functionalities on top of these. It can also be possible that with the configuration of the application, more acquaintances are made that can just do these basic functionalities, for example family members.

#### Login into account

Functionality: As an acquaintance, I can login into the account.

Normal flow: The actor enters their credentials to login into the system. The system shows the activities in the admin’s agenda.

Alternatives:

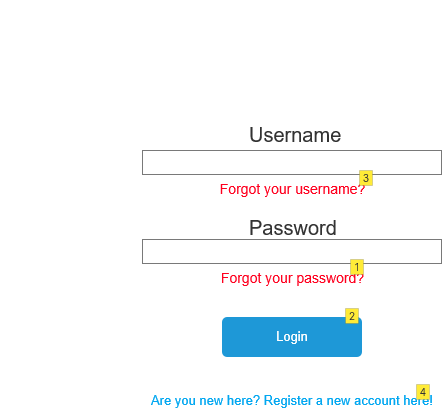
Wrong password: The system shows an error message. The actor can retry three times.

Email address does not exist: The system shows an error message. The actor chooses to register. The system starts UC register.

Reset password: The actor chooses to reset his password and fills in his email address. The system sends an email with a new activation link to the email address.

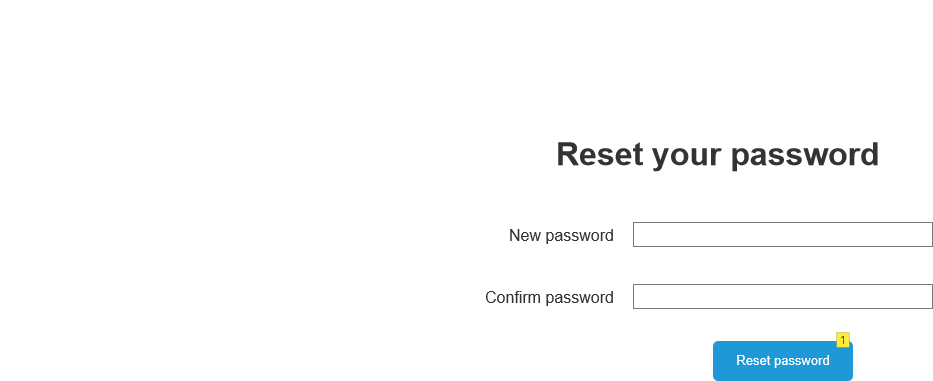
Forgot username: The actor enters the email address that is linked to their account. The system sends an email with the username linked to that email address.

UC1\_Login:



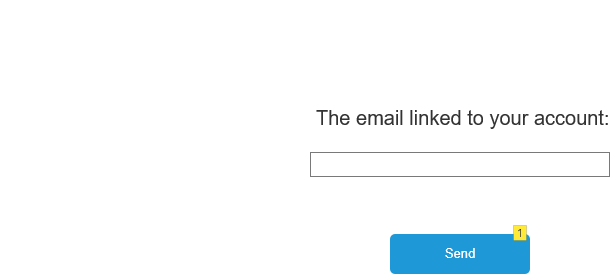
|  |  |
| --- | --- |
| Action | Explanation |
| 1 | This should only appear after someone tried to login and failed, when the link is clicked an email will be sent to authenticate, when the link in the email is clicked it will send you to the following page. |
| 2 | Upon clicking the button redirects to the main agenda page |
| 3 | This should only appear after someone tried to login and failed, when the link is clicked you will be sent to the UC1.1 page |
| 4 | After clicking it redirects the user to the UC18\_Register |
| Login | Go to UC15\_Homepage |

UC4.1\_ResetPassword:



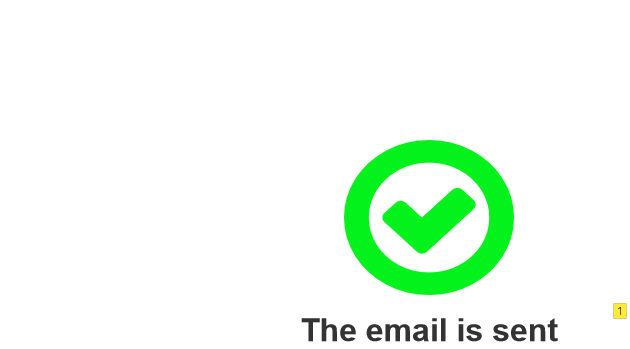
|  |  |
| --- | --- |
| Action | Explanation |
| 1 | After putting the new password and clicking the button the new password is set for the account associated with the email/username is changed |
| Reset password | Go to UC4.2\_PasswordResetConfirm |

UC1.1\_ForgotUsername:



|  |  |
| --- | --- |
| Action | Explanation |
| 1 | When you click the button, the system sends you an email with your username if the email is linked to an account and you are redirected to the UC1.2 page. |
| Send | Go to UC1.2\_ForgotUsernameSucces |

UC1.2\_ForgotUsernameSucces:



|  |  |
| --- | --- |
| Action | Explanation |
| 1 | You are automatically redirected to the login page. |

#### Activate account

Functionality: As an acquaintance, I can activate my account.

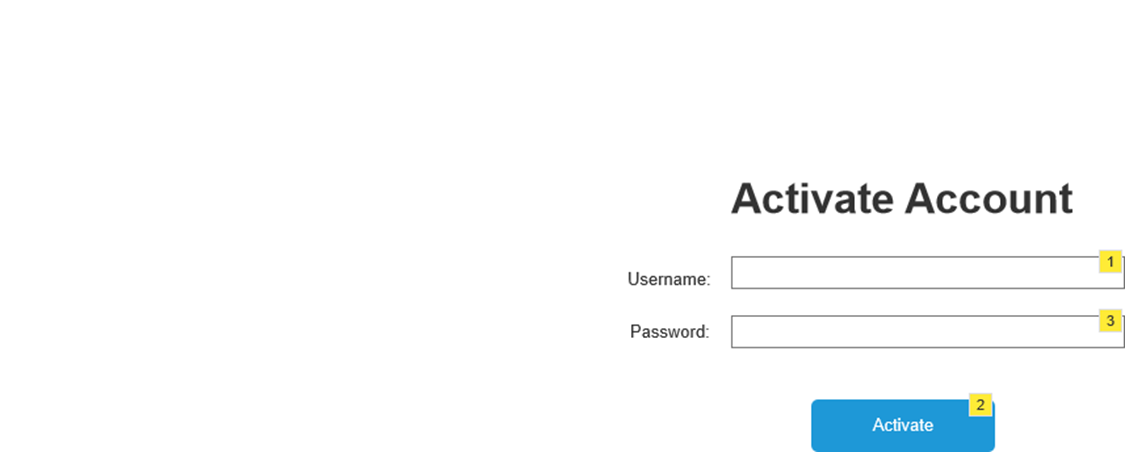
Preconditions: The acquaintance received a link to activate their account

Normal flow: The system asks for your new username and a password. The actor fills them in and saves their credentials. The system redirects to the login page. The actor can log in.

Alternatives:

Existing user: The system shows a message saying that there already is a acquaintance with the same name and asks them to change it. The actor can re-enter his credentials.

UC9.2\_Activate account:



|  |  |
| --- | --- |
| Action | Explanation |
| 1 | choose a username |
| 2 | your account is created |
| 3 | choose a password |
| Activate | Go to UC15\_Homepage |

#### Change password

Functionality: As an acquaintance, I can change my password

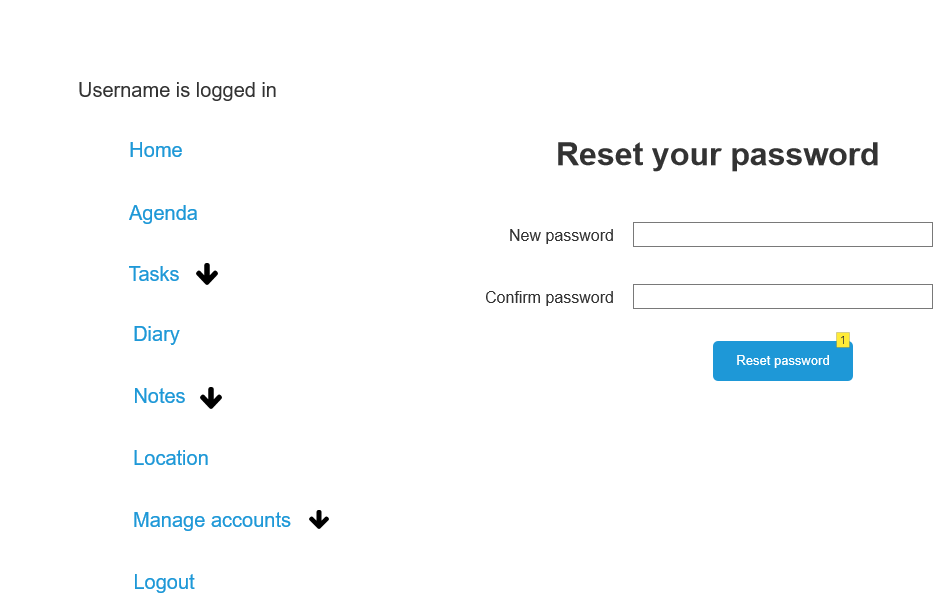
Preconditions: The acquaintance is logged in.

Normal flow: The system displays a screen that asks for the current password, the new password and to repeat the new password. The actor enters the passwords and saves his changes. The system shows a message that the password was successfully changed.

Alternatives:

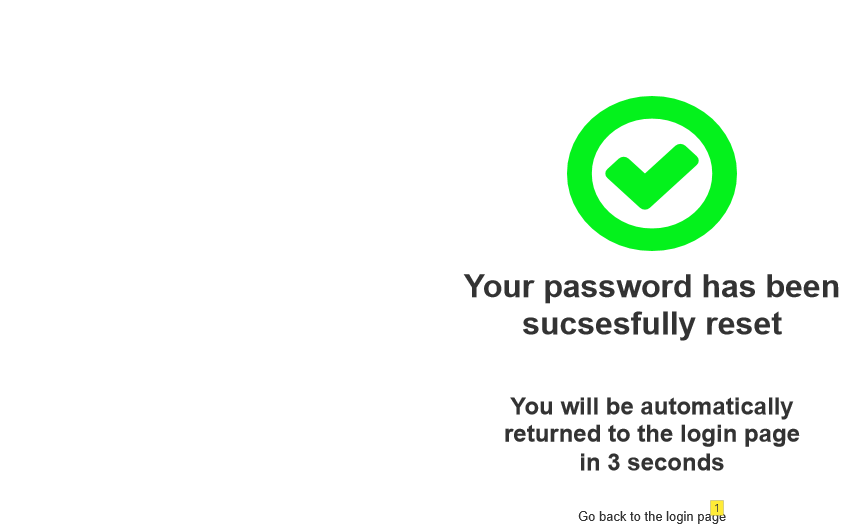
Same password: The system shows an error message that the new and old password are the same. The actor can fill in another password.

UC4.3\_ResetPasswordInsideOfTheSystem:



|  |  |
| --- | --- |
| Action | Explanation |
| 1 | After putting the new password and clicking the button the new password is set for the account associated with the email/username is changed |
| Reset password | Go to UC4.2\_PasswordResetConfirm |

UC4.2\_PasswordResetConfirm:



|  |  |
| --- | --- |
| Action | Explanation |
| 1 | Click on this link if you don't get redirected automatically to the login page |

#### Manage agenda.

Functionality: As an acquaintance, I can manage the agenda.

Preconditions: The acquaintance is logged in.

Normal flow: The system displays appointment settings. The actor changes or updates an activity in the calendar. The system saves the changes and shows them in the calendar. The admin and informal carer get a notification on their phone, which tells them something has changed about the activity.

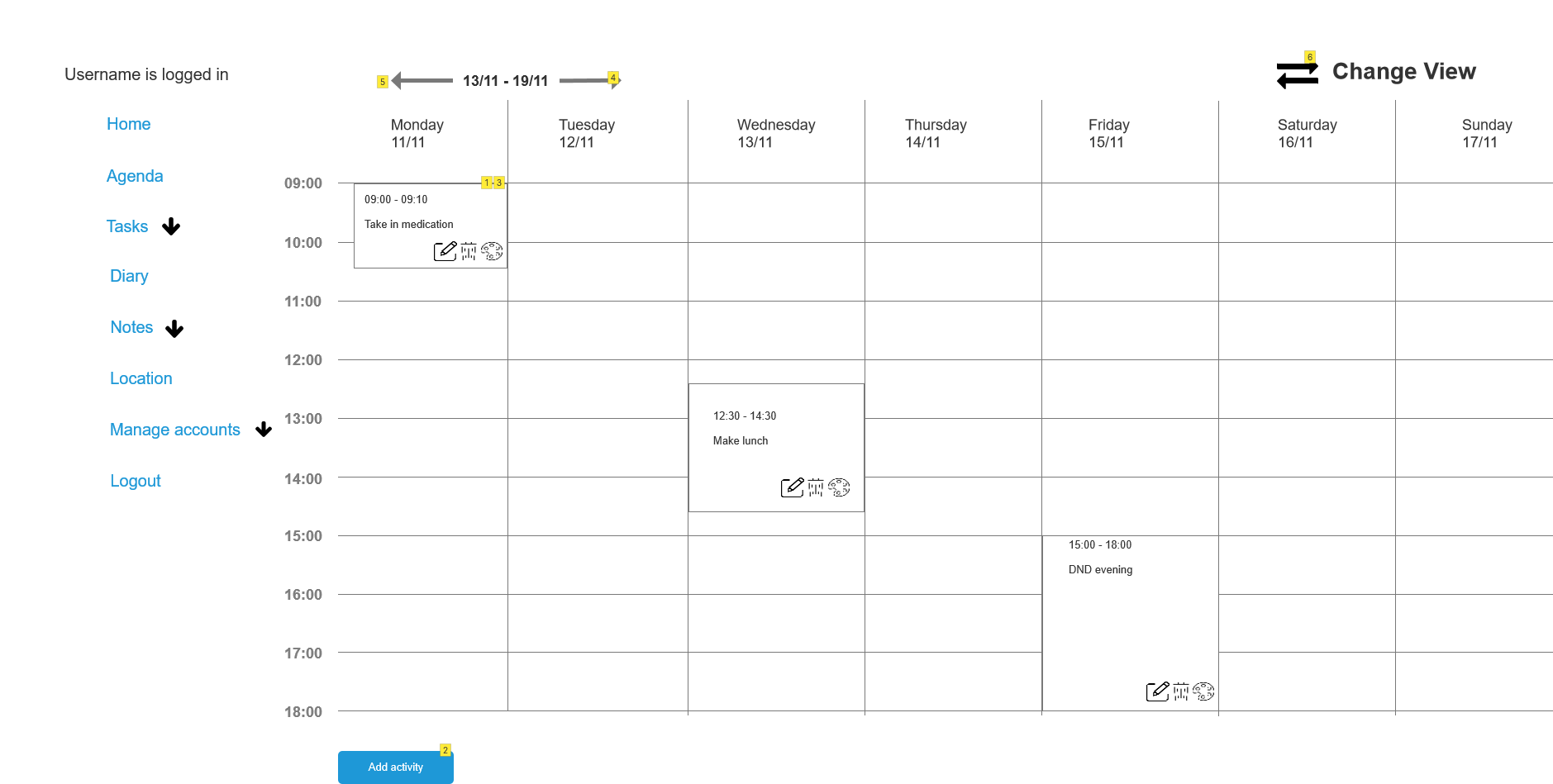
Alternatives:

Deleting activity: The actor chooses to delete the activity and confirms it. The admin and informal carer get a notification on their phone, which tells them something has changed about the activity.

Adding activity: The actor chooses to add an activity and fills in the name of the activity, the date, the start, and end time of the activity. He can also fill in some extra explanation about the activity. The system adds the activity to the calendar.

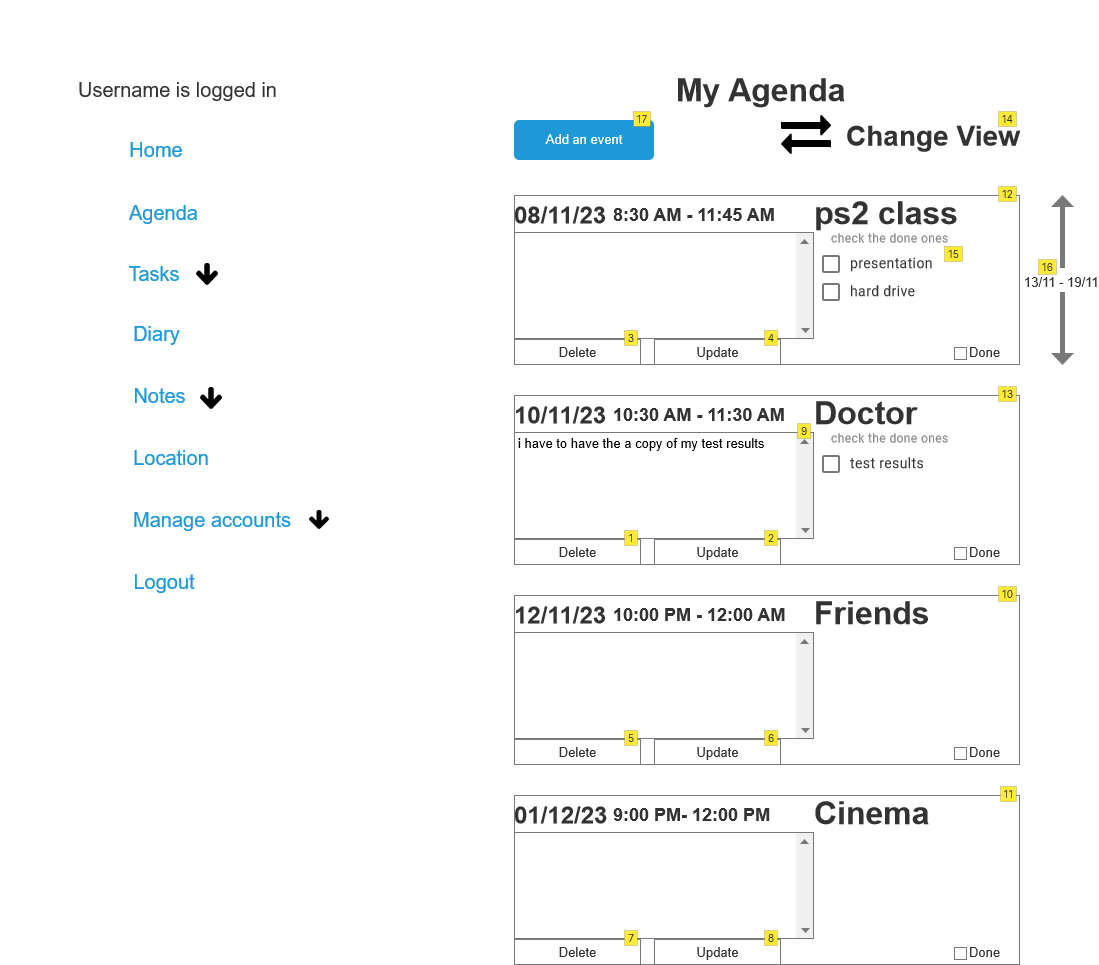
Adding a checklist: If the actor chooses to add a checklist, for example for a morning routine, to the activity, he can use the option to add a checklist. The actor fills in all the things that need to be checked off. The system adds the checklist to the activity and adds it to the calendar.

UC2\_Agenda:



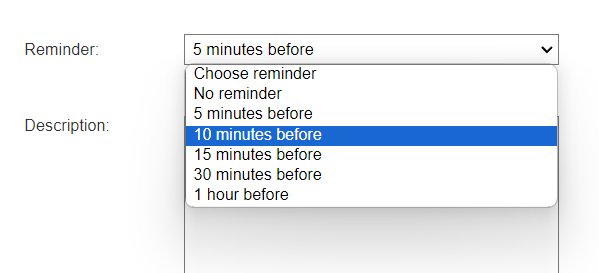
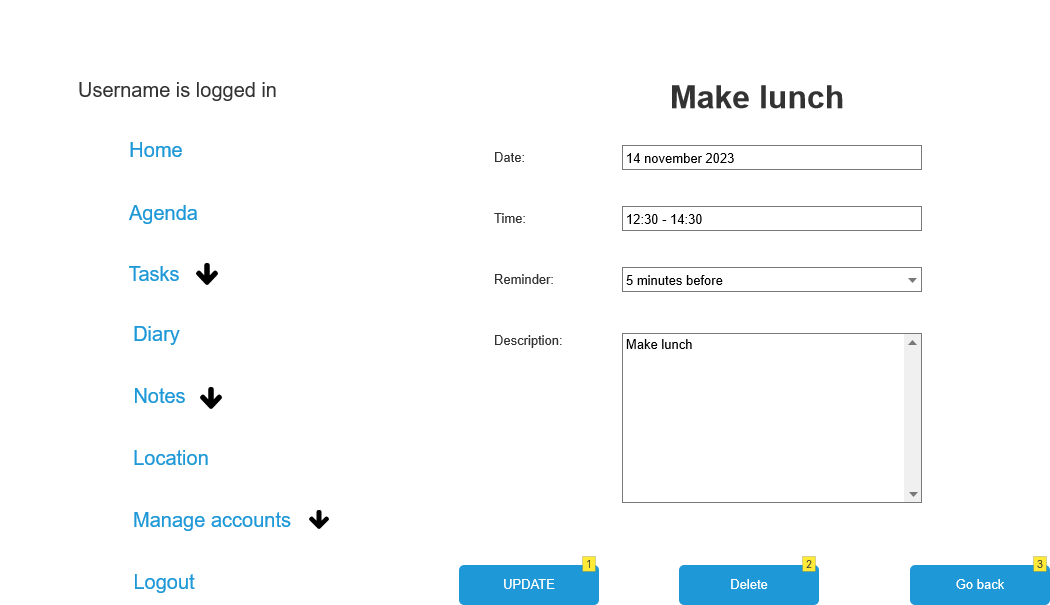
|  |  |
| --- | --- |
| Action | Explanation |
| 1 | Activities are added dynamically when you add an activity on the add activity page. |
| 2 | Goes to a new page where you can add an activity. |
| 3 | When you click on it, you see all details about the activity. |
| 4 | Go one week further |
| 5 | Go one week back |
| 6 | When you click the button, a different type of agenda is shown - the list view. It redirects to UC2.1 |
| Add activity | Go to UC3.1\_ManageAgenda |
| Home | Go to UC15\_Homepage |
| Agenda | Go to UC2\_Agenda |
| To do list | Go to UC16\_ToDoList |
| Routines | Go to UC19\_CustomizeMorningRoutine |
| Diary | Go to UC11.1\_ReadDiary |
| Notes | Go to UC7.1\_\_read\_note |
| Generate QR code | Go to UC13\_QRcode |
| Location | Go to UC14\_CheckLocation |
| Add account | Go to UC9.1\_AddUser |
| Permissions | Go to UC5\_AccountPermissions |
| Change your password | Go to UC4.3\_ResetPasswprdInsideOfTheSystem |
| Logout | Go to UC1\_Login |

UC2.1\_AgendaList:



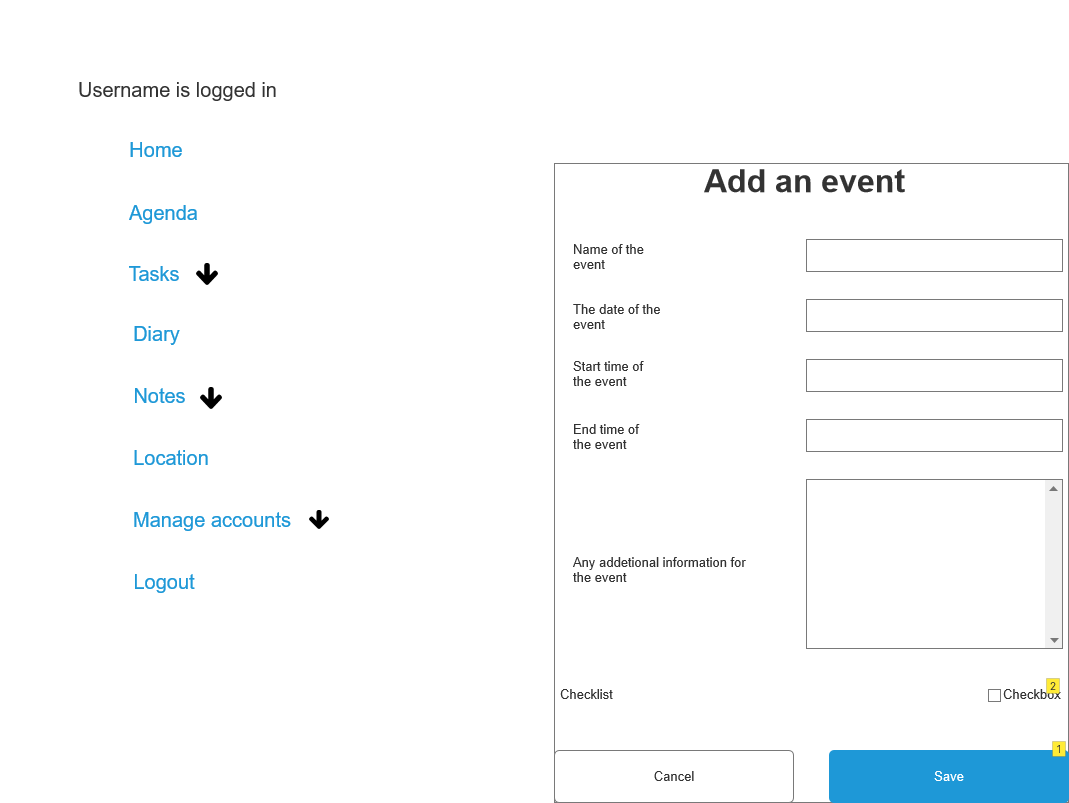
|  |  |
| --- | --- |
| Action | Explanation |
| 1,3,5,7 | when clicked the event will be deleted |
| 2,4,6,8 | when clicked you can change the header and the text |
| 9 | for any additional information regarding the event |
| 10,11 | this one does not have a checklist as in the add event the check box was not checked |
| 12,13 | this one has a checklist as in the add event the check box was checked |
| 14 | You can change the view back to the “calendar.” Redirects to UC2\_Agenda |
| 15 | You can click on the box if the activity is done, and the task is being marked as done |
| 16 | You can move the weeks down and up to access previous or future dates |
| 17 | You can add an event to the view |
| Add activity | Go to UC3.1\_ManageAgenda |

UC12\_ShowActivity:



|  |  |
| --- | --- |
| Action | Explanation |
| 1 | Updates the activity with new information and goes back to the home page. It first shows a message that it updated the activity successful. |
| 2 | It deletes the activity from your agenda and goes back to your agenda. |
| 3 | Goes back to the home page to show your agenda. |
| UPDATE | Go to UC2\_Agenda |
| Delete | Go to UC2\_Agenda |
| Go back | Go to UC2\_Agenda |

UC3.1\_ManageAgenda:



|  |  |
| --- | --- |
| Action | Explanation |
| 1 | after clicked it will save their event in the agenda |
| 2 | if clicked then you can make a check list |
| Cancel | Go to UC2\_Agenda |
| Save | Go to UC2\_Agenda |

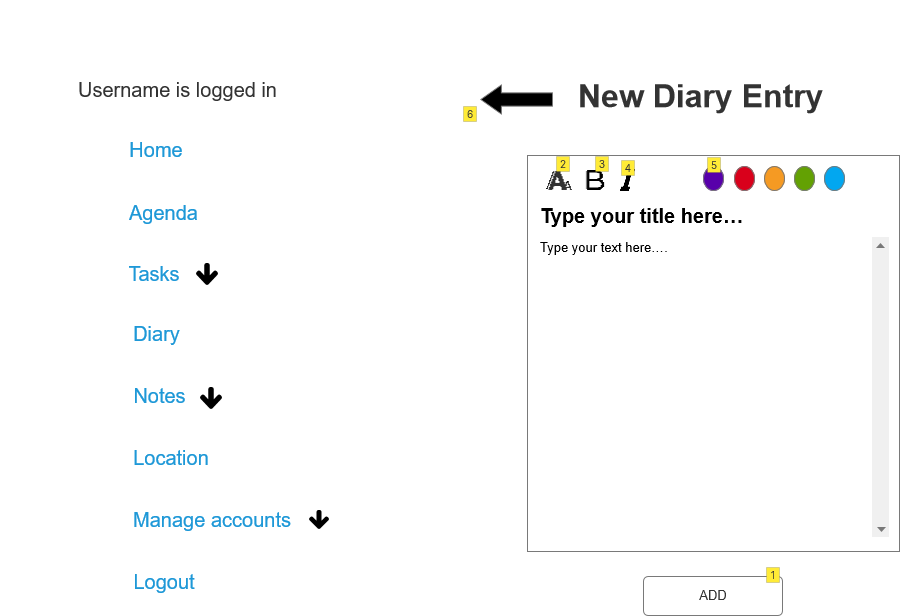
#### Write a diary.

Functionality: As an acquaintance, I can write a diary.

Precondition: The acquaintance should be logged in to navigate the diary.

Normal flow: The system shows the dairy. The actor writes a diary and all the necessary information about it. The system adds the dairy to the log and shows all the dairy entries including the new one.

UC11.2\_Write Diary:



|  |  |
| --- | --- |
| Action | Explanation |
| 1 | A new diary entry is added |
| 2 | The acquaintance can use which font they want to use to create the diary entry. |
| 3 | the acquaintance can pick this option to make the selected text bold |
| 4 | The acquaintance can select this option to make the text in cursive |
| 5 | The acquaintance can select in which color the entry will be displayed in the diary. |
| 6 | Takes acquaintance back to the diary view. |
| ADD | Go to UC11.1\_ReadDiary |

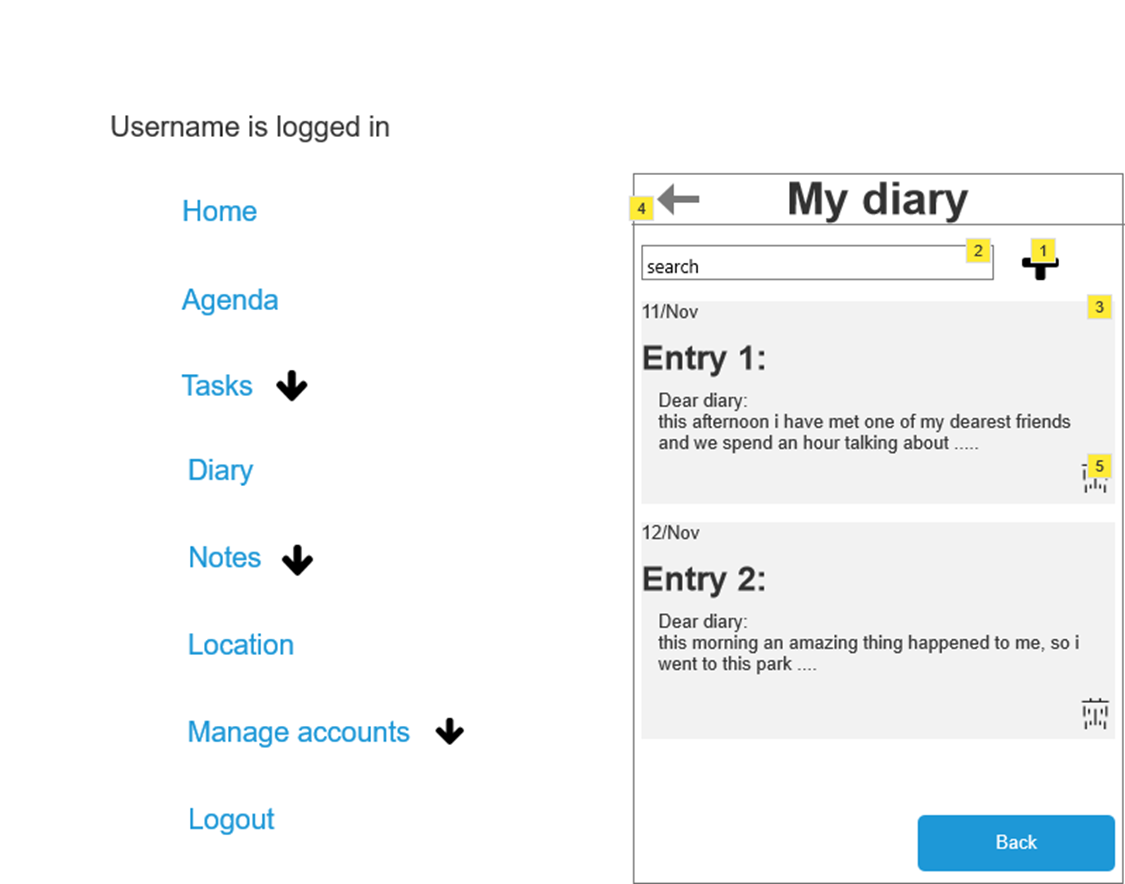
#### Read diary

Functionality: As an acquaintance, I can read dairy entries

Precondition: The acquaintance is logged in.

Normal flow: The system shows a list of all the dairy entries that have been written. The actor chooses one. The system shows the contents of the dairy entry and the date it was written.

UC11.1\_ReadDiary:



|  |  |
| --- | --- |
| Action | Explanation |
| 1 | When clicked the acquaintance is redirected to create a new diary entry |
| 2 | The user can look through the diary entries with specific key words. If the word exists in the diary entry, the diary entry is visible, if there are no diaries entries the screen is empty until the search is deleted |
| 3 | If you click on the specific diary entry, you can see the enlarged version of the diary entry and read it. |
| 4 | upon clicking you go back to the main menu where you can select other options again |
| 5 | archive the diary entry. You can find the archived diary entry through the button archive in UC7.1\_\_read\_note. |
| Back | Go to UC2\_Agenda |

#### Read note.

Functionality: As an acquaintance, I can read notes

Precondition: The acquaintance is logged in.

Normal flow: The system shows a list of all the notes that have been written by guests. The actor chooses one. The system shows the contents of the note and the date it was written.

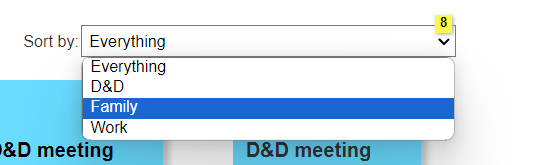
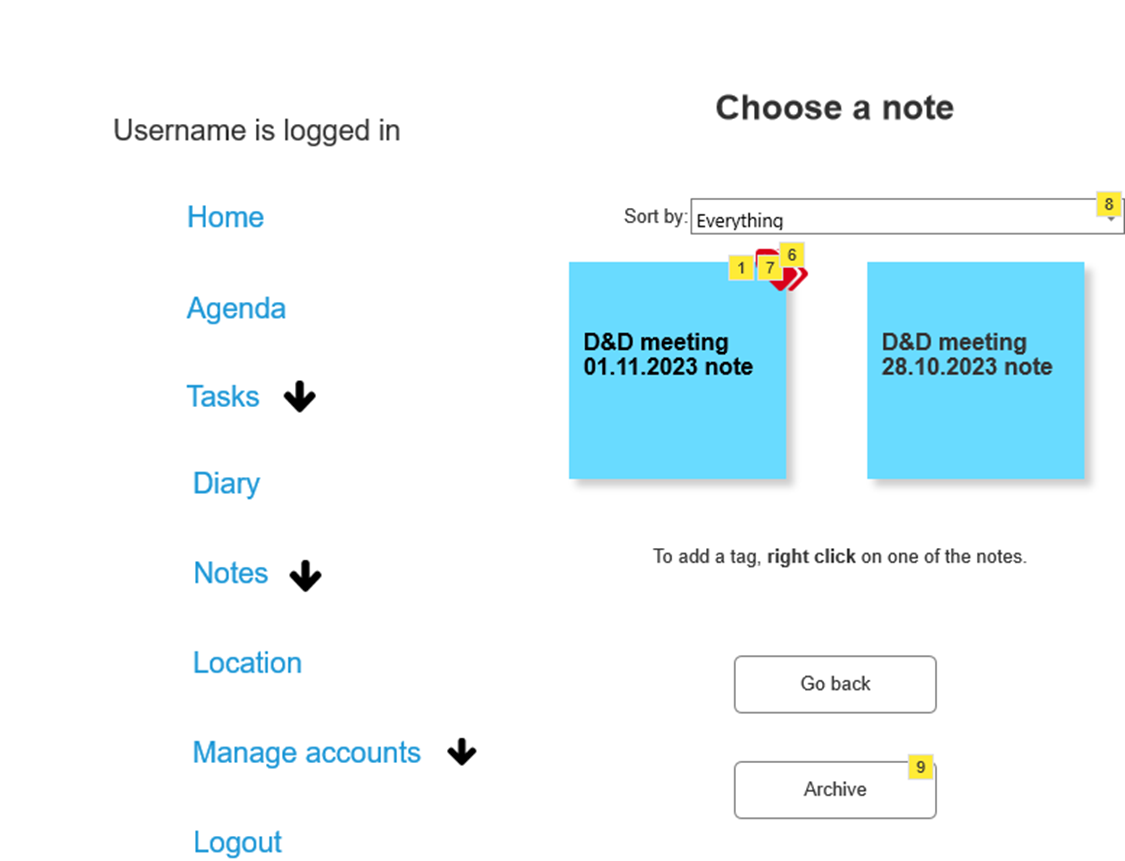
Alternative:

Archive note: If the actor does not need the note anymore, they can archive it. The note will be stored in the archive.

Filter notes: The actor can filter on a certain type of note to only show those notes. The system shows these specific notes and hides the rest.

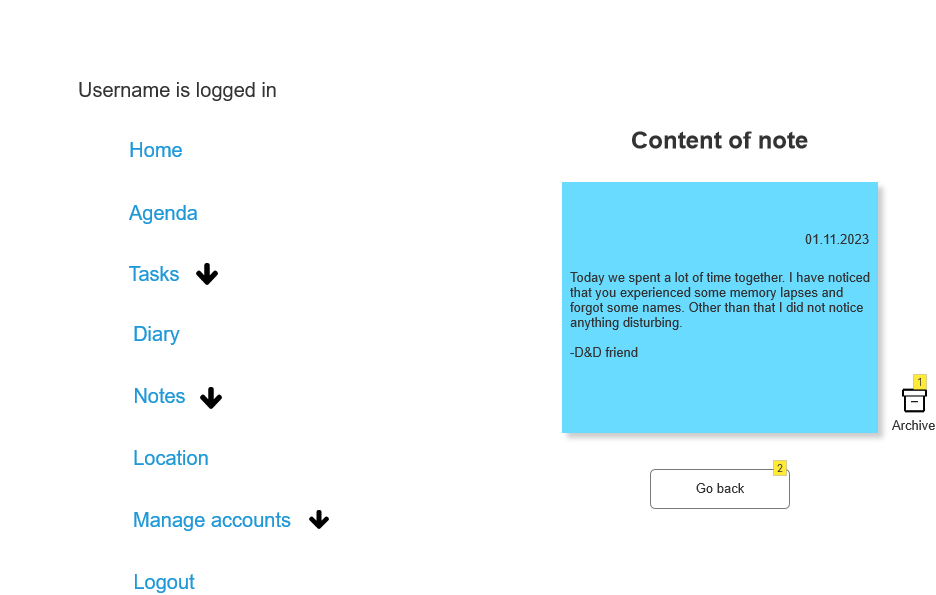
Give filter to note: The actor chooses a note and gives it a specific topic to filter it. The system saves the note with the topic.

UC7.1\_\_read\_note:

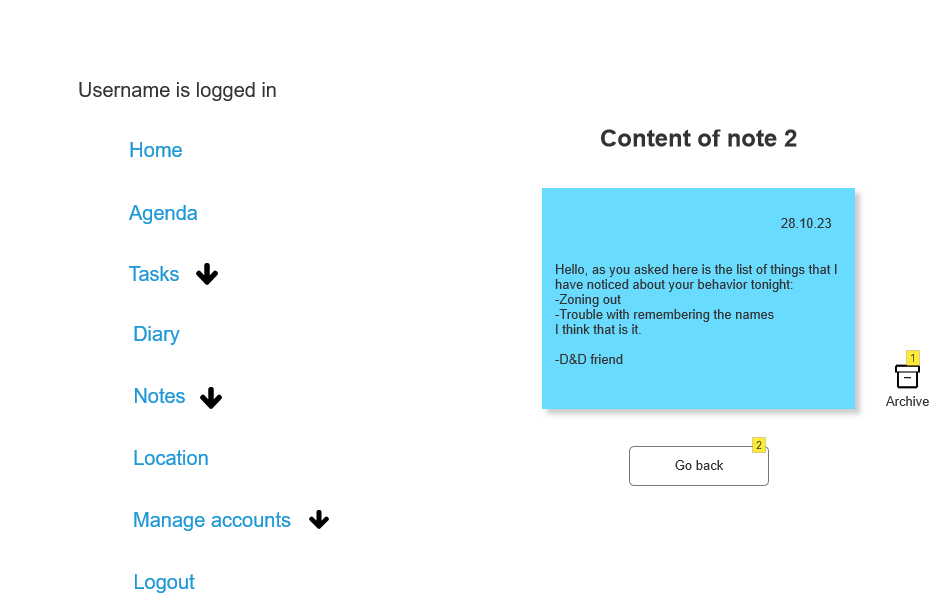


|  |  |
| --- | --- |
| Action | Explanation |
| 1 | When you click on the note, you can see the content of it |
| 6 | Tag assigned by a user; color is equivalent to the category assigned. |
| 7 | When the user right clicks on one of the notes, popup appears, on which they can assign tags or add new ones. |
| 8 | You can choose a tag in the drop-down list to filter the notes on |
| 9 | Go to the archive |
| Go back | Go to UC2\_Agenda |
| Archive | Go to UC7.4\_archive |

UC7.2\_read\_note:

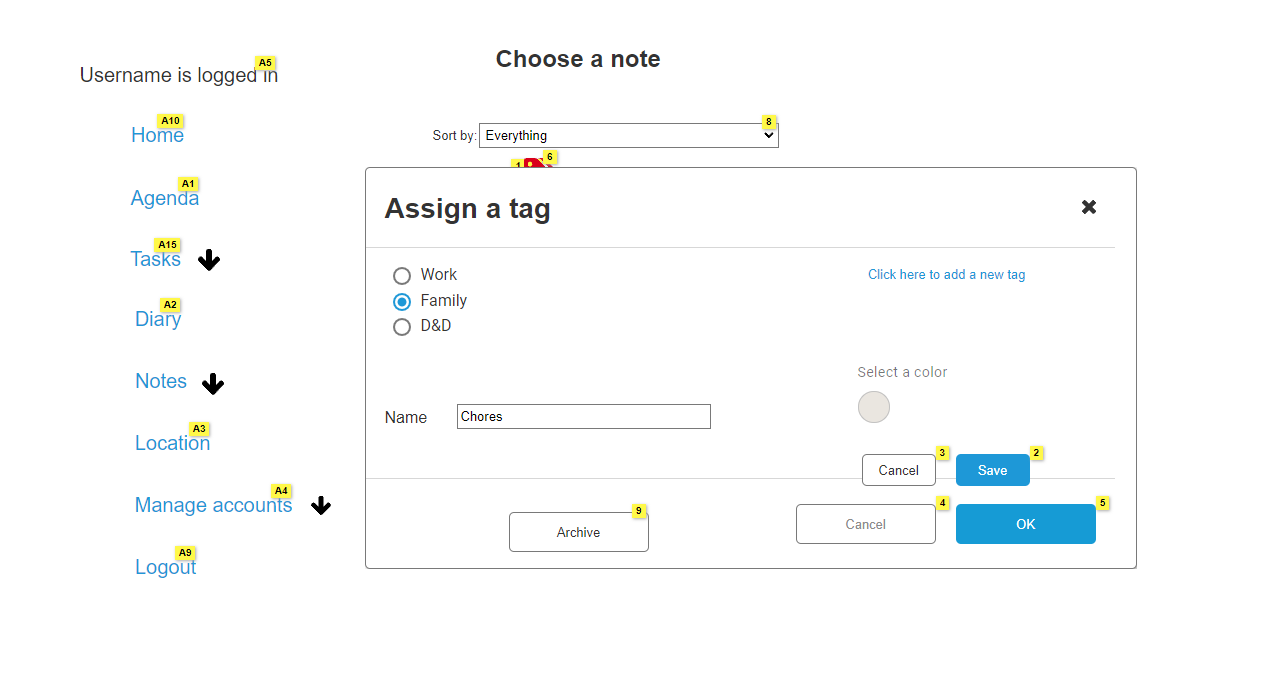


UC7.3\_read\_note:



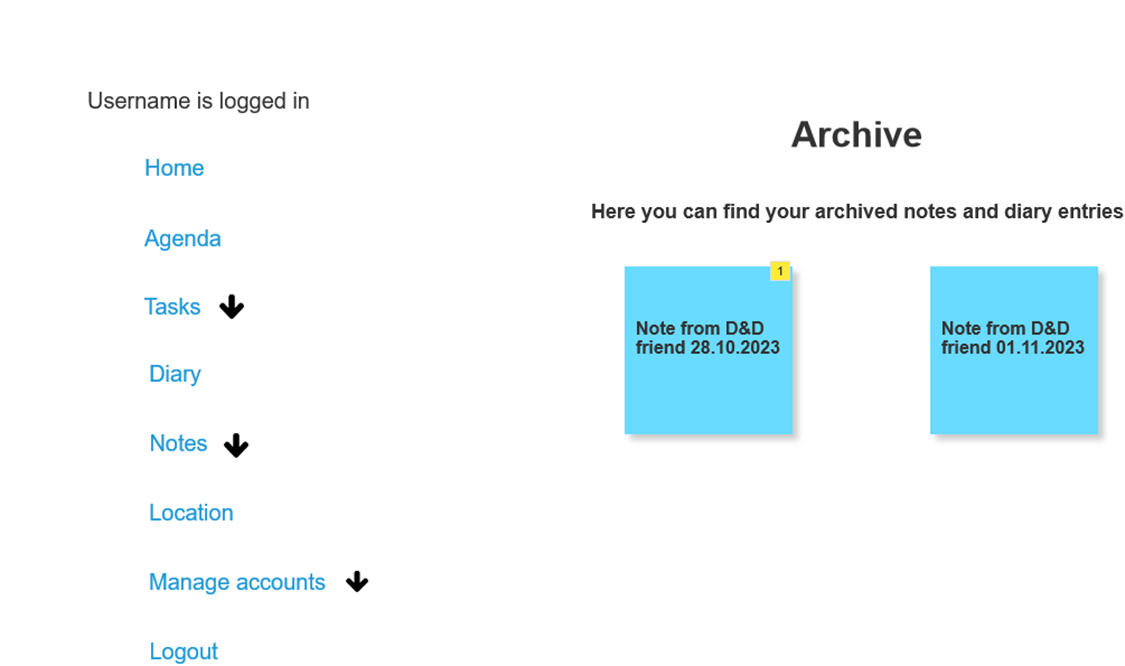
|  |  |
| --- | --- |
| Action | Explanation |
| 1 | When user decides to archive a note, the note also should disappear from the main page where you can choose the notes from |
| 2 | Takes you back to the main note view where you can see all the active notes |
| Go back | Go to UC7.1\_read\_note |

UC7.1\_\_read\_note pop-up:



|  |  |
| --- | --- |
| Action | Explanation |
| 2 | When the user saves the changes, new tag should appear on the list |
| 3 | When the user cancels the action, changes are not being saved. |
| 4 | Changes are not saved. |
| 5 | When user presses this button, system adds a new tag to the note that was previously selected. |

UC7.4\_archive:



|  |  |
| --- | --- |
| Action | Explanation |
| 1 | if you press the note or dairy entry you can see its content |

#### Register account

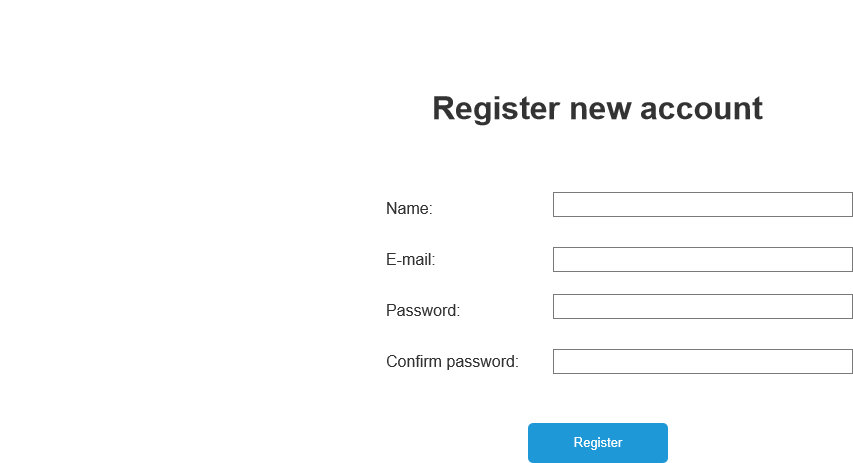
Functionality: As an acquaintance, I can register a new account.

Normal flow: The actor gives in his name, e-mail address and password. The system makes a new user and sends an e-mail to the user to tell him he succeeded in making a new account.

Alternatives:

E-mail already exists: The system gives a message to show that the e-mail already exists in the database. The actor gives in another e-mail address.

UC18\_Register:



|  |  |
| --- | --- |
| Action | Explanation |
| 1 | an email is sent to the given email address to confirm that the account is created |
| Register | Go to UC1\_Login |

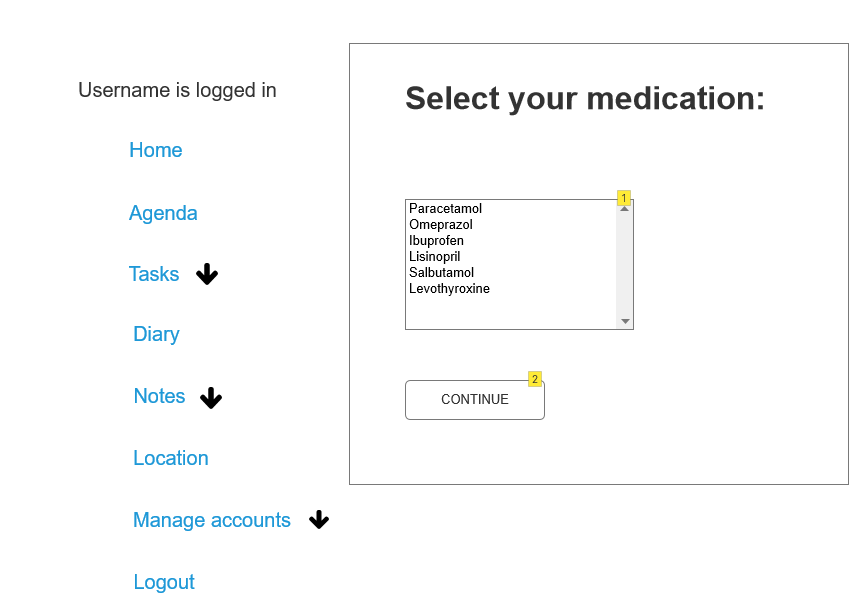
#### Fill out questionnaire.

Functionality: As an admin, I can fill out a questionnaire.

Precondition: The admin is logged in.

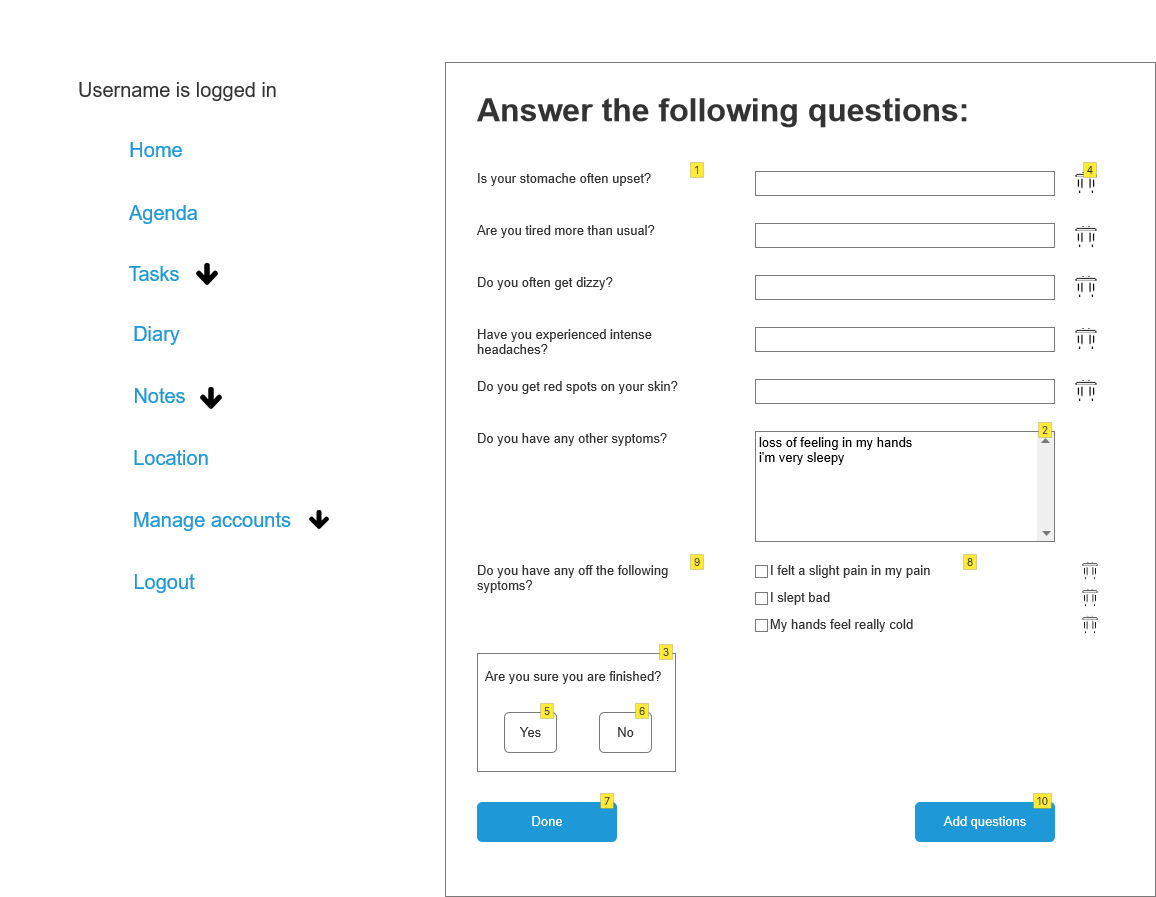
Normal flow: The system shows a list with the different medication. The actor selects the medication that was used. The system shows a premade questionnaire with different questions about the taken medication and a place to enter text about side effects that weren’t included in the questions or things that the doctor should be informed about. The actor fills in the questions. The system asks for confirmation. The actor confirms. The system sends the questionnaire to the doctors’ account. The system shows a success screen.

UC8\_FillOutQuestionnaire:



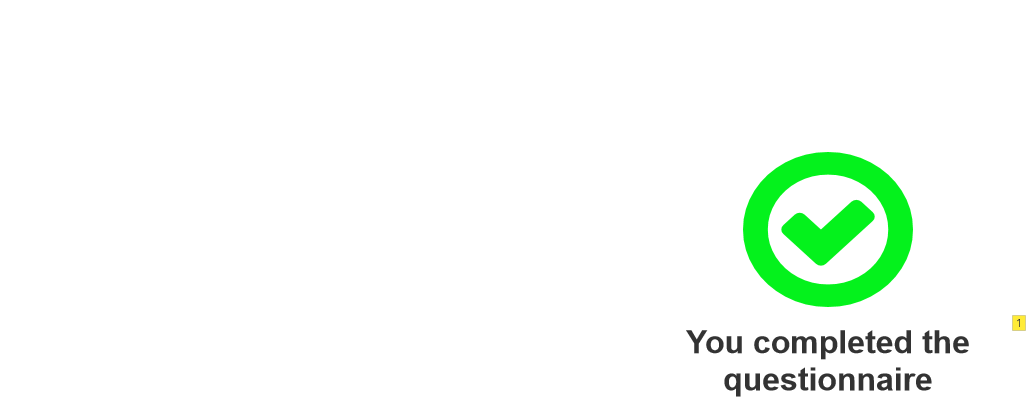
|  |  |
| --- | --- |
| Action | Explanation |
| 1 | You can select the medication of which you want to fill out the questionnaire. |
| 2 | You will be redirected to UC8.1\_FillOutQuestionnaire |
| CONTINUE | Go to UC8.1\_FillOutQuestionnaire |

UC8.1\_FillOutQuestionnaire:



|  |  |
| --- | --- |
| Action | Explanation |
| 1 | These questions are premade. You don't have to answer every question if you don't want to. |
| 2 | If you experienced any other symptoms, you could write them down here. The things that you write down here are added to the list box the next time you fill in the questionnaire. |
| 3 | This window appears when you press Done. If you're finished and press yes, you are redirected to the main page. If you aren't you can continue filling in the questionnaire. The pop-up will disappear if you press no. |
| 4 | If you click on this icon, you can delete the question |
| 5 | This button redirects to UC8.3\_SuccesScreen |
| 6 | This button hides the pop-up |
| 7 | When pressing this button, the pop-up appears |
| 8 | When you write something down in the list box above, it appears as a checkbox the next time that you fill in the questionnaire. |
| 9 | The first time you fill in the questionnaire, this question won't appear. If the admin doesn't fill in the text area before this question, it won't appear. Once the admin wrote something in the text area, the question will appear the next time you open the questionnaire. The things that the admin wrote in the text area will appear as checkboxes next to the question. |
| 10 | You will be redirected to the UC8.2 page to add a question |
| Add questions | Go to UC8.2\_CustomizeQuestionnaire |
| Yes | Go to UC8.3\_SuccesScreen |

UC8.3\_SuccesScreen:



|  |  |
| --- | --- |
| Action | Explanation |
| 1 | You will be redirected to the home page automatically |

#### Customize questionnaire.

Functionality: As an admin, I can customize a questionnaire.

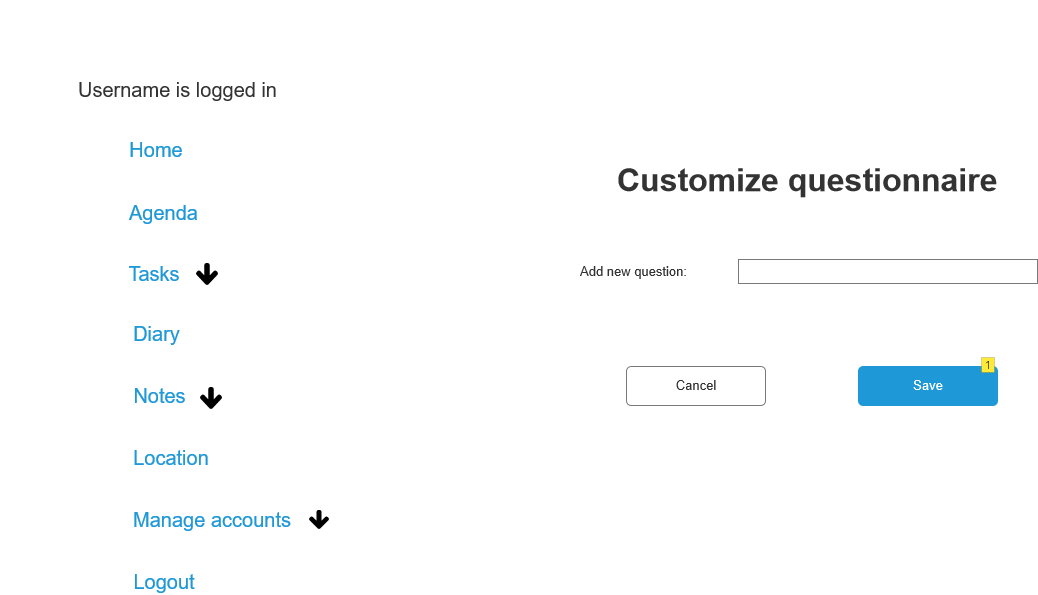
Precondition: The admin is logged in.

Normal flow: The system shows the questionnaire. The actor can add a new question to the list. The system shows the question and saves the answers the actor gives on them.

Alternatives:

Remove a question: The actor removes a question. The system removes the question from the questionnaire.

UC8.2\_CustomizeQuestionnaire:



|  |  |
| --- | --- |
| Action | Explanation |
| 1 | After adding new question, it should display in the questionnaire |
| Cancel | Go to UC8.1\_FillOutQuestionnaire |
| Save | Go to UC8.1\_FillOutQuestionnaire |

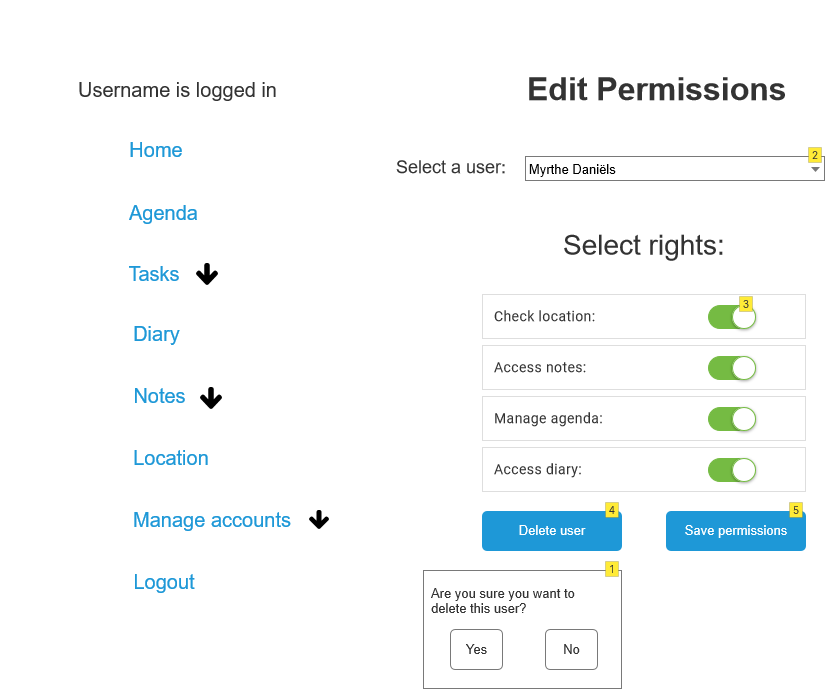
#### Give permissions.

Functionality: As an admin, I can give permissions.

Precondition: The admin is logged into the system.

Normal flow: The system displays all accounts. The actor chooses an account and selects what level of access it has in the application. System saves the settings.

UC5\_AccountPermissions:



|  |  |
| --- | --- |
| Action | Explanation |
| 1 | This window appears when you press Delete user. If you're sure and press yes, you are redirected to the main page. If you don't want to the pop-up will disappear. |
| 2 | Pick a user from the created user database to work on their permissions |
| 3 | Toggling the permissions puts it on or off for the specific picked account before |
| 4 | If you want to delete the user, it can be done via this button |
| 5 | Click this button to save and apply the changes to the picked users |
| Yes | Go to UC5.1\_AccountDeletedSucces |
| Save permissions | Go to UC5\_AccountPermissions |

UC5.1\_AccountDeletedSucces:



|  |  |
| --- | --- |
| Action | Explanation |
| 1 | After 3 seconds the page redirects to the previous page: Account permissions |

#### Add user

Functionality: As an admin, I can add users.

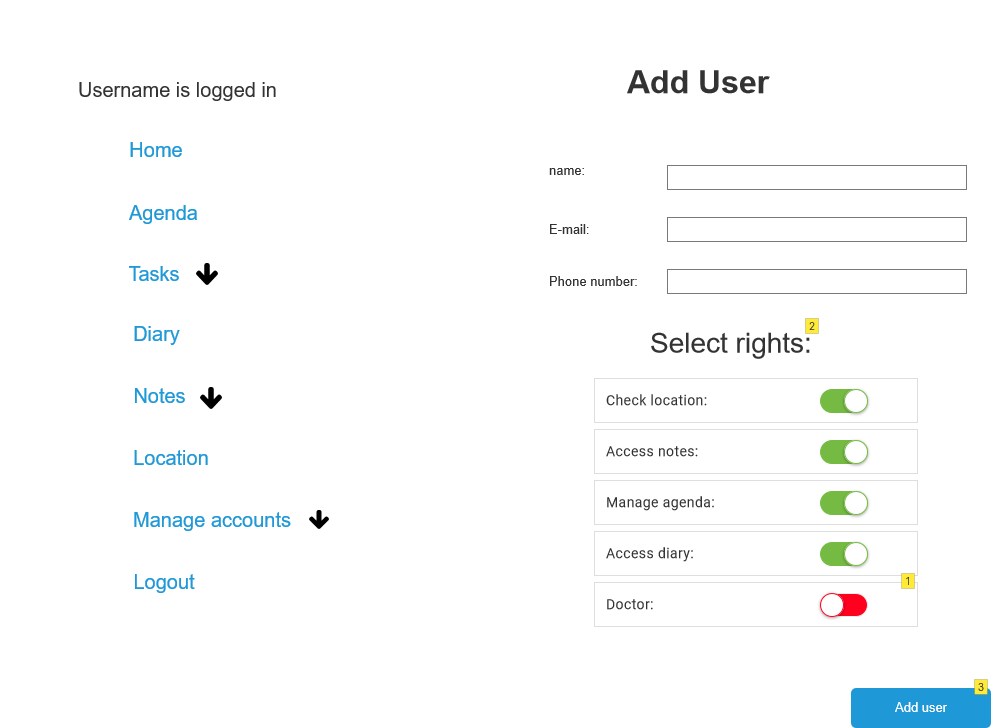
Precondition: The admin is logged into the system.

Normal flow: The system displays all users in the system. The actor fills in all information about the new user, like e-mail, permissions, name, and phone number. The system adds the user.

Alternative:

User already exists: The system gives a message that the username or e-mail is already in the system. The actor changes the username or e-mail to one that isn’t taken yet. The system adds the user.

UC9.1\_AddUser:



|  |  |
| --- | --- |
| Action | Explanation |
| 1 | says if the account is for a doctor or not |
| 2 | give permissions |
| 3 | the user is added |
| Add user | Go to UC9.2\_Activate account |

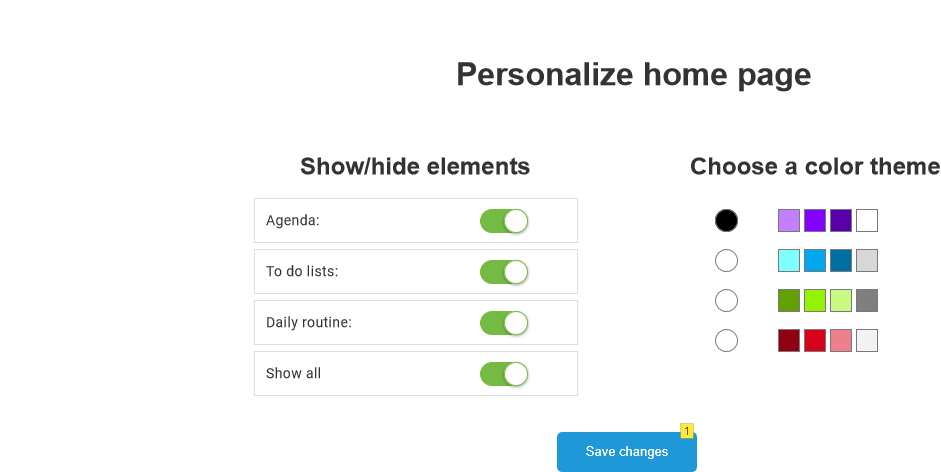
#### Personalize home page

Functionality: As an admin, I can personalize the home page.

Precondition: The admin is logged into the system.

Normal flow: The system shows the settings of the home page. The actor chooses what to show on the home page and what personalization to give. The system changes the home page to the personalization the user chooses.

UC17\_Personalize:



|  |  |
| --- | --- |
| Action | Explanation |
| 1 | Save the personalization’s and changes to the home page |
| Save changes | Go to UC15\_Homepage |

#### Manage to do list

Functionality: As an admin, I can manage a to do list.

Precondition: Admin is logged into the system.

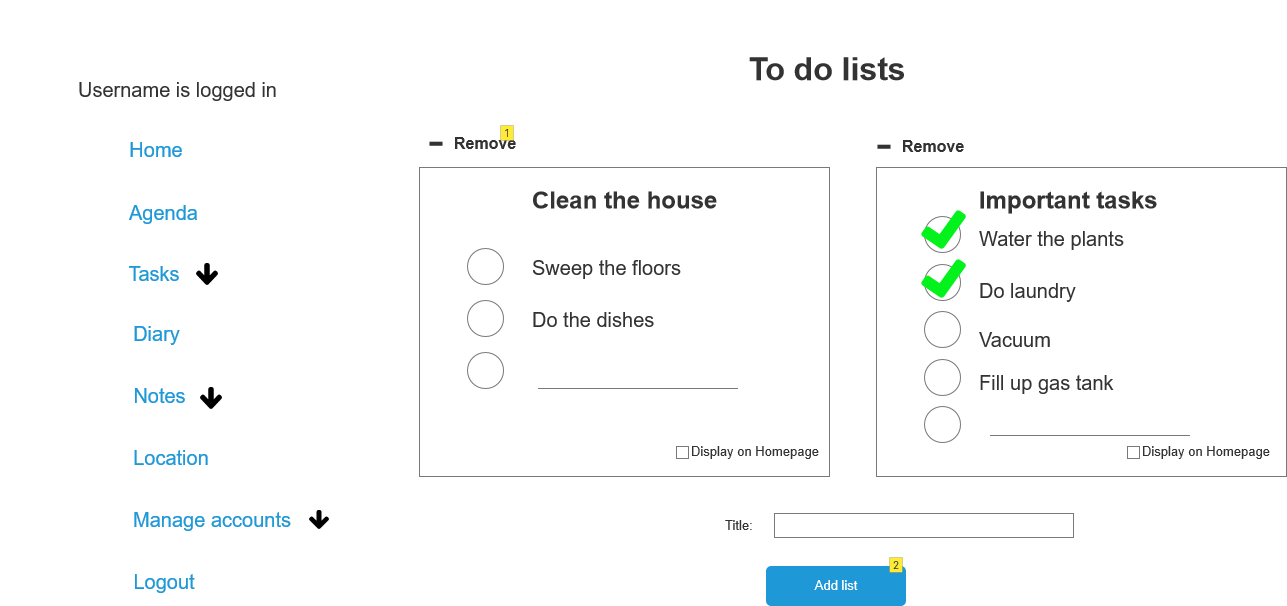
Normal flow: The systems show the existing to do lists. The actor adds new tasks to the to do list. The system adds the new tasks to the to do list.

Alternatives:

Remove task: The actor has done a task and removes it from the list. The system deletes the task from the do list.

Add to do list: The actor enters a title and adds the to do list. The system creates the to do list and saves the changes.

UC16\_ToDoList:



|  |  |
| --- | --- |
| Action | Explanation |
| 1 | Removes the to do list |
| 2 | Add the new to do list with the title that you choose |
| 3 | add tasks to the to do list. The task is added when you press enter |
| 4 | Tick the box if you want to display the to do list on the home page. You can only display one at a time |

#### Manage routines

Functionality: As an admin, I can manage routines.

Precondition: Admin is logged into the system.

Normal flow: The system displays the routines that already exist. The actor enters a title and creates a new routine. The system adds the routine to the system.

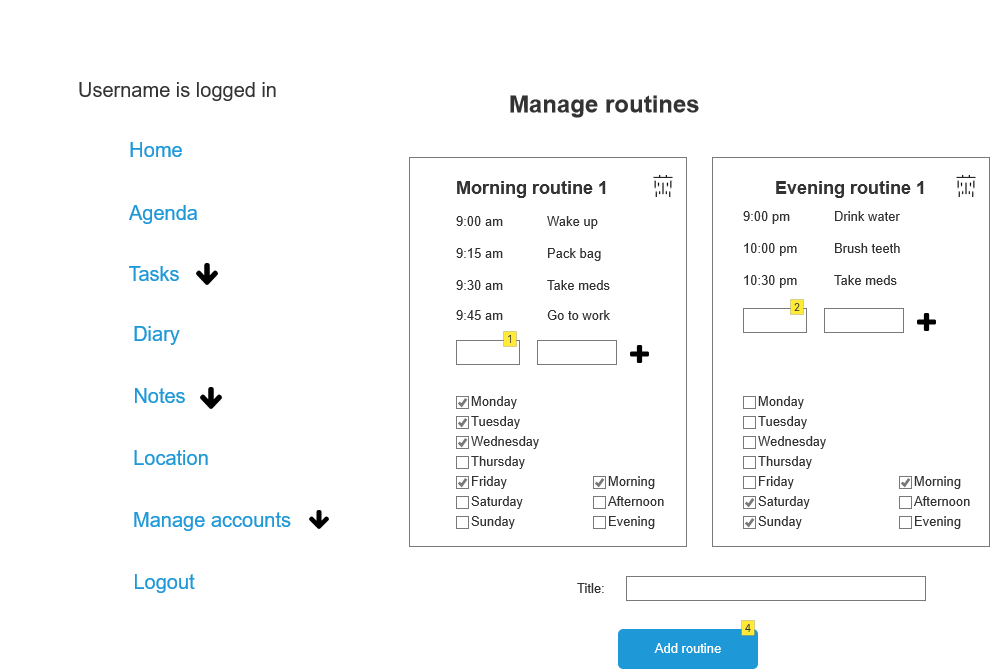
Alternatives:

Remove routine: The actor removes the routine. The system deletes the routine.

Add routine: The actor enters a title and adds the routine. The system creates the routine and saves the changes.

Add task: The actor adds a task to the routine. The system automatically places the tasks in chronological order and saves the changes.

UC19\_CustomizeRoutines:



|  |  |
| --- | --- |
| Action | Explanation |
| 1 | You write down a time in this box |
| 2 | Add a new routine with the name you entered |
| 3 | you write down a task in this box |
| 4 | you add the new task and time to the routine. The system automatically places the tasks in chronological order |
| 5 | with this button you delete the routine |

#### Show QRcode

Functionality: As an admin, I can show a QRcode

Precondition: Admin is logged into the system.

Normal flow: The system shows a QRcode, that a quest can scan.

UC13\_QRcode:

A qr code on a screen

Description automatically generated

|  |  |
| --- | --- |
| Action | Explanation |
| 1 | When you scan the QR code, friend can add note/feedback after the meeting. |
| Go Back | Go to UC15\_Homepage |

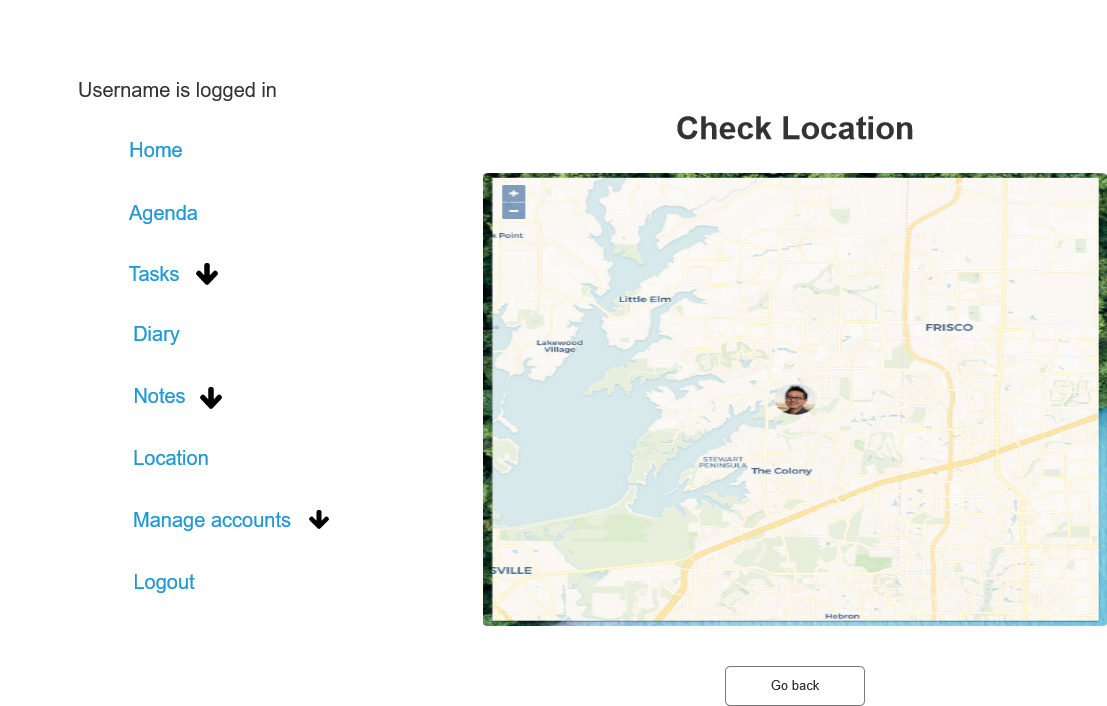
#### Check location.

Functionality: As an informal carer, I can check the location of the admin.

Preconditions: The informal carer is logged in.

Normal flow: The system displays the latest coordinates on a map that it received from the admins’ account.

UC14\_CheckLocation:



|  |  |
| --- | --- |
| Action | Explanation |
| 1 | check the location of the phone of the admin |
| Go back | Go to UC15\_Homepage |

#### Reminder medication

Functionality: As time, I can remind the admin and the informal carer about taking medication.

Normal flow: The actor sends a notification to the phone of the admin and the informal carer. This notification tells them that it is time to take medication.

Alternatives:

Missed medication: The actor sends a notification to the phone of the informal carer. This notification tells them that the admin has not taken their medication.

#### Reminder appointments

Functionality: As time, I can remind the admin and the informal carer about appointments.

Normal flow: The actor sends a notification to the phone of the admin and the informal carer. This notification tells them that an activity is going to start.

Alternatives:

A missed activity: The actor sends a notification to the phone of the admin and the informal carer. This notification tells them that an activity is not done.

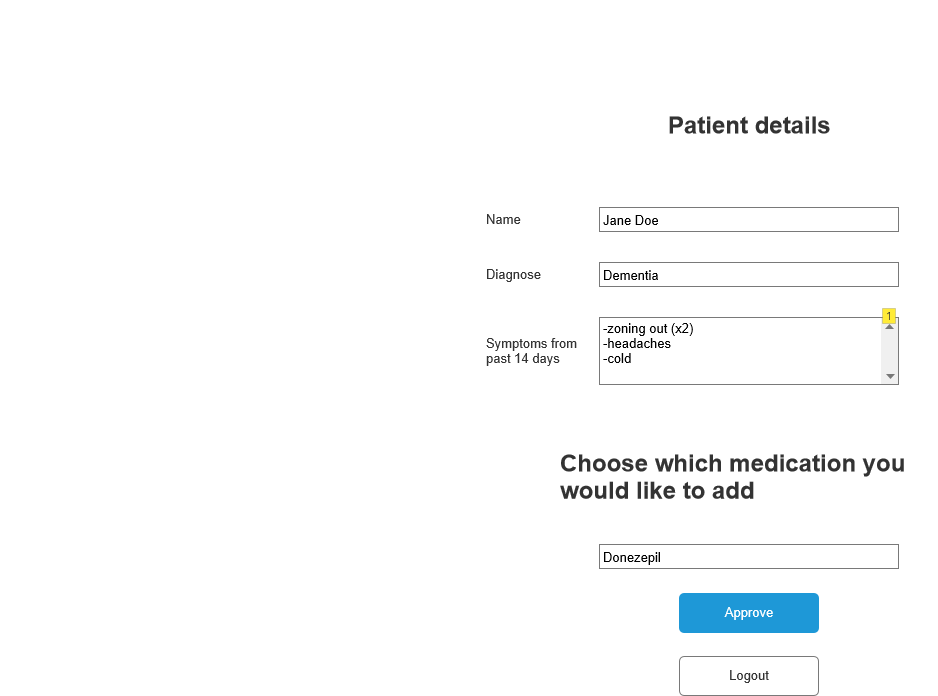
#### Add medication.

Functionality: As a doctor, I can add or delete medications.

Preconditions: The doctor is logged in.

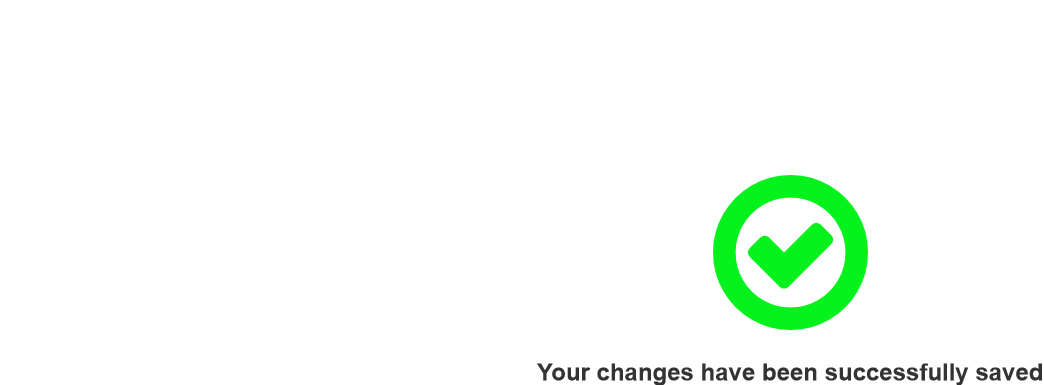
Normal flow: The system shows the information about the admin. The actor adds the medication details and saves the medication entry. The system saves the details.

UC10.1\_add\_medication:



|  |  |
| --- | --- |
| Action | Explanation |
| 1 | The symptoms that appear as checkboxes in the questionnaire are displayed here. The number of times that the checkboxes were clicked appears between brackets behind the symptom. |
| 2 | enter the medication that you want to add to the list |
| 3 | confirm the changes |
| Approve | Go to UC10.2\_add\_medication\_saved |
| Logout | Go to UC1\_Login |

UC10.2\_add\_medication\_saved:



|  |  |
| --- | --- |
| Action | Explanation |
| 1 | After 3 seconds the page redirects to the previous page UC10.1\_add\_medication |

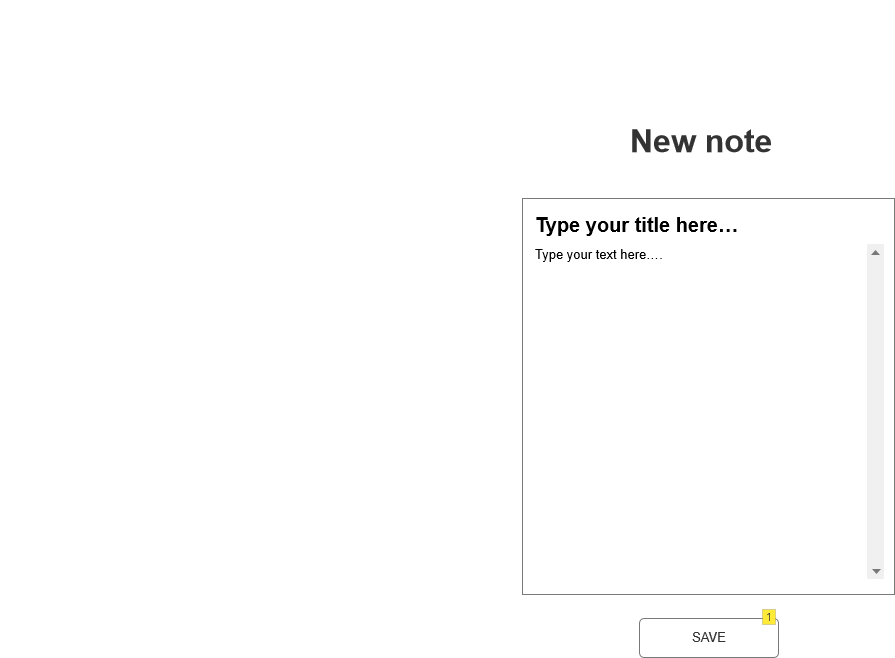
#### Write note.

Functionality: As a guest, I can write notes

Precondition: The guest has a temporary login by scanning a QR-code.

Normal flow: The system opens a page where the actor can write down their notes. The actor writes down all their remarks. The system saves the page between the other notes on the admin’s app.

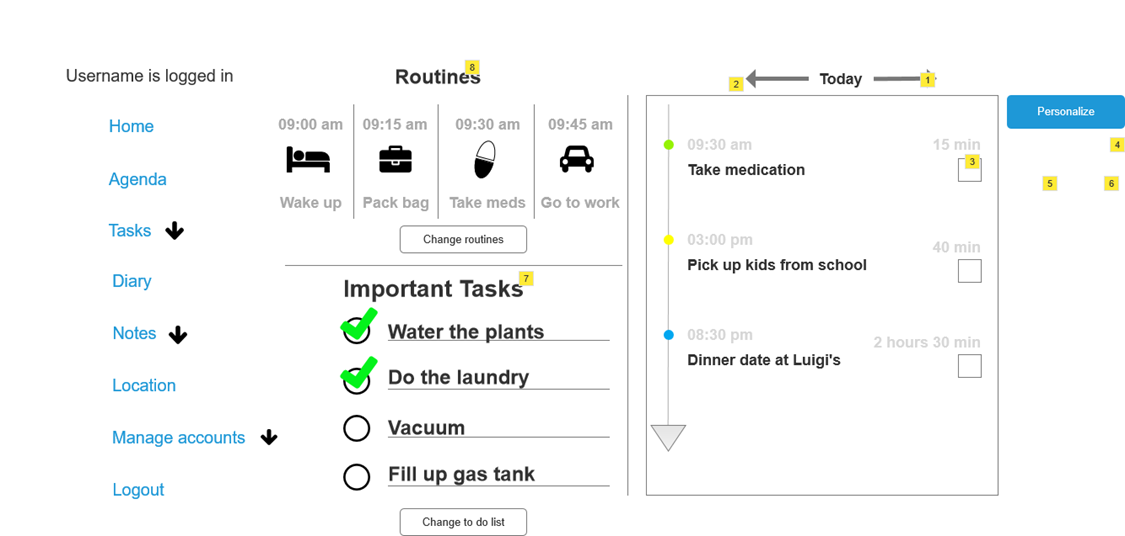
UC6\_Write note:



|  |  |
| --- | --- |
| Action | Explanation |
| 1 | After clicking save the note is being saved on the notes page. The page is being reloaded so that the guest leaving the note knows that it has been properly saved and sent. |
| SAVE | Go to UC6\_Write note |

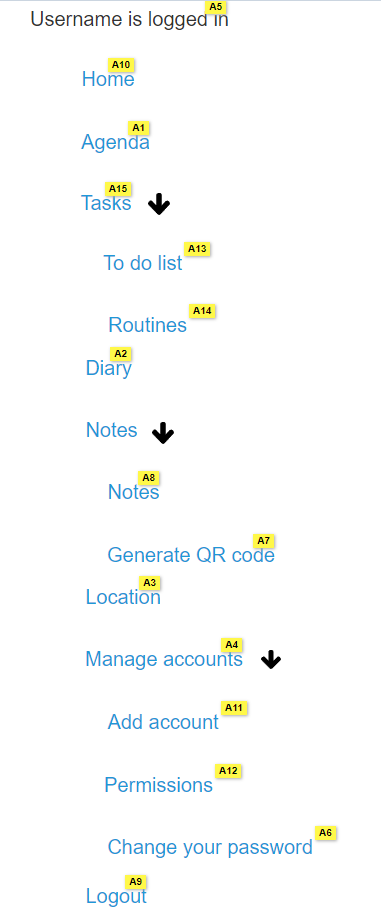
#### Home page

UC15\_Homepage:



|  |  |
| --- | --- |
| Action | Explanation |
| 1 | Go one day further |
| 2 | Go one day back |
| 3 | When you check of this box, the pop up will appear |
| 4 | This pop-up appears when you check off the check box |
| 5 | you are redirected to UC8\_FillOutQuestionnaire |
| 6 | This button hides the pop-up |
| 7 | There is one to do list at a time that is displayed, that was chosen in UC16\_ToDoList |
| 8 | There is one routine at a time that is displayed, depending on the settings that were decided in UC19\_CustomizeRoutines |
| Personalize | Go to UC17\_Personilaze |
| Change routines | Go to UC19\_CustomizeRoutines |
| Change to do list | Go to UC19\_ToDoList |

#### Navigation



|  |  |
| --- | --- |
| Action | Explanation |
| 1 | Goes to the home page, where you can see your agenda. |
| 2 | Goes to your diary, to show all your notes. |
| 3 | Goes to a page where you can check the location of a user. |
| 4 | Goes to a page where you can add, remove, and update users but also manage the permissions. |
| 5 | Shows the username of the logged in user. |
| 6 | Goes to a page where you can easily change your password |
| 7 | Goes to a page where qr code will be generated |
| 8 | Goes to your diary, to show all your notes. |
| 9 | Logs you out of the application. |
| 10 | Goes to the home page, where you can see your agenda. |
| 11 | Goes to a page where you can add, remove, and update users but also manage the permissions. |
| 12 | Goes to a page where you can add, remove, and update users but also manage the permissions. |
| 13 | Goes to the home page, where you can see your agenda. |
| 14 | The link goes to the UC19 page, where you can edit the morning routine |
| 15 | Goes to the home page, where you can see your agenda. |

## Non-functional requirements

### Usability

Keep the interface simple and uncluttered. Avoid unnecessary complexity in design and functionality. Make use of components and Livewire to work faster and maintain the layout over all pages.

### Accessibility

Ensure that text and interactive elements are large enough for easy readability and interaction. Also provide support for keyboard navigation to accommodate users who may have difficulty using a mouse. Ensure compatibility with screen readers to assist users with visual impairments. Do this by using the right CSS selectors like focus and by correct use of HTML tags.

### Performance

Optimize the application for quick loading times to enhance the user experience. Design the application to use minimal system resources to accommodate users with older devices or slower internet connections. You can check this with the Lighthouse tool in the Chrome Developer tools in your browser. During the design you need to think about clean code, storage of images and videos and meta tags.

### Reliability

Ensure the application is stable and robust to prevent crashes or unexpected behavior. Implement measures to maintain the integrity of user data, preventing loss or corruption. Do this by implementing the GDPR rules on your application.

### Security

Implement strong data protection measures to ensure the privacy and security of user data. Use secure authentication mechanisms to protect user accounts from unauthorized access. Do this using middleware like the implemented authentication middleware. Also use the hash function in Laravel for your passwords.

### Scalability

Design the application to manage potential growth in the number of users and data over time. Can do this by putting your application in the cloud using Kubernetes.

### Compatibility

Ensure the application works well across various web browsers to accommodate users with different preferences. Make the application responsive to different screen sizes and devices to enhance accessibility. This is the basic for all web designers, mobile first.

### Adaptability

Allow users to personalize the interface based on their preferences, such as choosing colors or adjusting font sizes. Design the application to be adaptable to the changing needs and abilities of individuals with dementia. Implement these functions using Livewire.

### Documentation

Provide clear and concise documentation to guide users in using the application. Offer accessible customer support channels for users and caregivers to address any issues or questions. An easy documentation page should be enough and clean links to your contact information.

# Priority by functionality

In this last part we will talk about the priorities for every functionality. We do this by using a MoSCoW analysis. We split our analysis in must have this, should have this, if possible, could have this and would like to have. Let us begin:

## Login into account

|  |  |  |
| --- | --- | --- |
| Letter | Statement | Description |
| M | Must have this | Successful login and redirection to the home page. Also, a secure password. |
| S | Should have this if possible | Secure for path traversal |
| C | Could have this if it does not affect anything else | / |
| W | Would like to have but won't have this time around | Login with social media or Google. |

## Activate account

|  |  |  |
| --- | --- | --- |
| Letter | Statement | Description |
| M | Must have this | Sending an e-mail with a link to make a new account. |
| S | Should have this if possible | Redirection to login page. |
| C | Could have this if it does not affect anything else | / |
| W | Would like to have but won't have this time around | / |

## Change password

|  |  |  |
| --- | --- | --- |
| Letter | Statement | Description |
| M | Must have this | Successful change of password that is also changed in the database. |
| S | Should have this if possible | Validation of input |
| C | Could have this if it does not affect anything else | Error message when password is the same as the previous password. |
| W | Would like to have but won't have this time around | Secure for XSS and SQL injections. |

## Manage agenda

|  |  |  |
| --- | --- | --- |
| Letter | Statement | Description |
| M | Must have this | See activities, add activities, change the information, and delete the activities. |
| S | Should have this if possible | Overlapping activities. |
| C | Could have this if it does not affect anything else | Adding a checklist to an activity. |
| W | Would like to have but won't have this time around | Importing agendas from other platforms like Microsoft. |

## Write a diary

|  |  |  |
| --- | --- | --- |
| Letter | Statement | Description |
| M | Must have this | Writing things and adding them to your diary. |
| S | Should have this if possible | Deleting things in the diary. |
| C | Could have this if it does not affect anything else | Give tags to your notes and make a filter. |
| W | Would like to have but won't have this time around | Personalization of the notes in for example different colors. |

## Read note

|  |  |  |
| --- | --- | --- |
| Letter | Statement | Description |
| M | Must have this | Showing all notes. |
| S | Should have this if possible | Showing notes only made by guests. |
| C | Could have this if it does not affect anything else | Remove notes after 30 days. |
| W | Would like to have but won't have this time around | Archive the notes you want to keep. |

## Register account

|  |  |  |
| --- | --- | --- |
| Letter | Statement | Description |
| M | Must have this | Safe all data of the client into his account and make a new account. Redirect to home page. |
| S | Should have this if possible | Validate input. |
| C | Could have this if it does not affect anything else | XSS security |
| W | Would like to have but won't have this time around | Tutorial of all the functionalities after redirection. |

## Fill out questionnaire

|  |  |  |
| --- | --- | --- |
| Letter | Statement | Description |
| M | Must have this | A questionnaire where you choose medication and get random questions about this medication. |
| S | Should have this if possible | / |
| C | Could have this if it does not affect anything else | Make it optional to fill it out and don’t send the questionnaire to the doctor every time it is fills in, but monthly. |
| W | Would like to have but won't have this time around | Add it to the doctor software. |

## 

## Customize questionnaire

|  |  |  |
| --- | --- | --- |
| Letter | Statement | Description |
| M | Must have this | Adding and deleting questions. |
| S | Should have this if possible | Remove and add new questions to a specific medication. |
| C | Could have this if it does not affect anything else | / |
| W | Would like to have but won't have this time around | / |

## Give permissions

|  |  |  |
| --- | --- | --- |
| Letter | Statement | Description |
| M | Must have this | Show all users and their permissions. Change the permissions and save it. |
| S | Should have this if possible | Don’t show sensitive data. |
| C | Could have this if it does not affect anything else | / |
| W | Would like to have but won't have this time around | / |

## Add user

|  |  |  |
| --- | --- | --- |
| Letter | Statement | Description |
| M | Must have this | Make it possible for the admin to add new users. |
| S | Should have this if possible | Only for the admin. |
| C | Could have this if it does not affect anything else | Let password be filled in by user itself. |
| W | Would like to have but won't have this time around | Strong passwords and no sensitive data. |

## Manage to do list

|  |  |  |
| --- | --- | --- |
| Letter | Statement | Description |
| M | Must have this | Delete and change to do lists. |
| S | Should have this if possible | Show to do lists on home page. |
| C | Could have this if it does not affect anything else | Choose what to do list to show on home page. |
| W | Would like to have but won't have this time around | Personalize the do list with colors. |

## Add to do list

|  |  |  |
| --- | --- | --- |
| Letter | Statement | Description |
| M | Must have this | Make new to do list. |
| S | Should have this if possible | Add, delete, and change activities |
| C | Could have this if it does not affect anything else | Validate input |
| W | Would like to have but won't have this time around | Personalize to do list with colors |

## Check location

|  |  |  |
| --- | --- | --- |
| Letter | Statement | Description |
| M | Must have this | Check the location of the admin for the close relative of the admin. |
| S | Should have this if possible | Link it to Google Maps to show directions. |
| C | Could have this if it does not affect anything else | Everybody can check the location if they have permissions. |
| W | Would like to have but won't have this time around | Show every user on the same map to give an overview for the admin. |

## Reminder medication

|  |  |  |
| --- | --- | --- |
| Letter | Statement | Description |
| M | Must have this | Show a notification via mail to take in medication. |
| S | Should have this if possible | Send notification to informal carer. |
| C | Could have this if it does not affect anything else | Add a personal sound to the message. |
| W | Would like to have but won't have this time around | Show notification on your phone |

## Reminder appointments

|  |  |  |
| --- | --- | --- |
| Letter | Statement | Description |
| M | Must have this | Show a notification via mail to reminder of appointment. |
| S | Should have this if possible | Send notification to informal carer or person related to the appointment. |
| C | Could have this if it does not affect anything else | Add a personal sound to the message. |
| W | Would like to have but won't have this time around | Show notification on your phone |

## Add medication

|  |  |  |
| --- | --- | --- |
| Letter | Statement | Description |
| M | Must have this | The doctor can add medication to certain clients. |
| S | Should have this if possible | Can’t see navigation, only add medication. |
| C | Could have this if it does not affect anything else | Security |
| W | Would like to have but won't have this time around | Add notes to the medication. |

## Write note

|  |  |  |
| --- | --- | --- |
| Letter | Statement | Description |
| M | Must have this | The guest can add notes. |
| S | Should have this if possible | Generate link for guest to fill in note. |
| C | Could have this if it does not affect anything else | Generate QR code to fill in note. |
| W | Would like to have but won't have this time around | / |