Software Requirements Specification

for

<TEDIAS Use-case 2 neurogical amnamesis dasboard>

Version 1.0 approved

Prepared by Mario Bijeljanin

<organization>

<01.06.2022>

Table of Contents

1. Introduction [1](#__RefHeading___Toc441230972)

2. Overall Description [3](#__RefHeading___Toc441230978)

3. External Interface Requirements [7](#__RefHeading___Toc441230986)

4. System Features [18](#__RefHeading___Toc441230991)

5. Other Non-functional Requirements [20](#__RefHeading___Toc441230994)

6. Other Requirements [22](#__RefHeading___Toc441231000)

7. Risks [23](#__RefHeading___Toc3219_2379971635)

Revision History

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Date** | **Reason For Changes** | **Version** |
|  |  |  |  |
|  |  |  |  |

# Introduction

## Purpose

The purpose of the dashboard is to optimize the efficiency of a neurological clinic and to help the whole medical staff in managing patient prioritization by filtering patient information by importance. This will assist the staff in having a good general idea which patients to prioritize as well as create the best custom treatment plan accordingly. In turn, patient wait times will be minimal, as well as the costs to the hospital and insurance provider

The improvement would be facilitated by taking out the guess work from the nurses job, simply allowing the software to do the legwork for them. This holds especially true for neurological patients who, due to their condition, are often unable to best communicate. The current patient waiting time will be displayed on the dashboard. This will help medical personell to better keep track of patient waiting times in relation to the urgency of symptoms.

Abuse of the system is flagged through appropriately coded questions designed to recognize if the patient is trying to shorten their wait time via dishonestly over-exaggerating the urgency of their predicament.

The aim of the dashboard is to improve the patient triage process by reducing patient total wait time and enabling staff to focus on high priority cases.

## Document Conventions

This Document was created based on the IEEE template for System Requirement Specification Documents with the featured user patient journey as implemented by the IREB template.

This document is organized according to the screens of the software. I.e, homepage, patient list, patient panel

These were mainly gathering information through interviews, documenting requirements and merging similar requirements by the standards of the well-known International Requirements Engineering Board (IREB)

IREB defines three different requirements: functional requirement, quality requirement, and limitations,

## Intended Audience and Reading Suggestions

This document’s target audience is the software designer commissioned to execute the final product. The document itself serves as a guide to better understand how to implement the dashboard and understand its design. The functional requirements are specifications that must be implemented in order to meet user requirements and comply with all data protection guidelines and regulatory requirements.

The functional requirements are specifications that must be implemented in order to meet user requirements and comply with all data protection/privacy guidelines and regulatory requirements.

The complete user patient journeys are provided and may be used as a guideline for possible future feature updates. To achieve this, multiple interviews with relevant stakeholders where planned, conducted and analyzed. From these interviews, requirements were identified and combined to give a more cohesive overview about the missing functionality.

The dashboard is intended for medical staff working in patient triage at the Mannheim neurological A&E (Accidents & Emergency) department.

## Product Scope

The project and this project report were done during the summer semester 2022 at the University of Heidelberg /Mannheim. The end goal of this project is to provide a functional software that will be used primarily in the healthcare system.

The scope of this project covers the design of the dashboard and its functionality. It shows the layout and design of each screen in the software, and which features are to be implemented in each. All data that was collected for this dashboard will be shown and explained for potential expansion by other developers or engineers.

## References

The chapter "Anforderungen ermitteln"from the book "Business Analysis und Requirements Engineering" written by Peter Hruschka

The book "Basiswissen Requirements Engineering" written by Klaus Pohl and Chris Rupp

Certified Professional for Requirements Engineering" (CPRE) from "International Requirements Engineering Boards" (IREB)

The book “Data science” written by John D.Kelleher and Brendan Tierney

Kodey S. Crandall; (2020). Risk Assessments: A Weighted Score Approach to Improving Risk Management Decisions . 2020 Intermountain Engineering, Technology and Computing (IETC)

# Overall Description

## Product Perspective

The software for the dashboard was developed for the use in medical facilitates specially for neurological clinics. The need for this kind of software arose from the lack of suitable software, aimed at assisting neurological patients and to serve as the next stepping stone digitization of the medical field. At the moment doctors and the medical stuff working there use a mixture of software and paper to record the patients that vising the clinic with the hope that their neurological problems will be effectively treated. The software that is normally used organizes patients in a first came -first serve manner which, in turn, leads to the possibility of a patient in critical condition having to wait**.** The key dashboard functionality is the integrated questionnaire and ranking system, which has the goal of best assessing the patient’s status and needs as well as ranking them by priority.

## Product Functions

The dashboard will allow users to access and view information about patients in the hospital ward where the dashboard is implemented. The patients will be able to describe all their symptoms and the doctor and the medical staff will have a good overview of the patients conditions, the goal is to optimize the patient treatment and to avoid unnecessary long waiting time from the patients

## User Classes and Characteristics

The users of this software are and the patients and the medical staff each with their own intended terminal it is used to maximize the efficiency in the intended field.

## Operating Environment

The application consists of a back-end and a front-end service (see architecture diagram). Both services are containerised as part of a microservices architecture and run in a Docker container. The Docker base image is a Linux operating system, preferentially Ubuntu. The back-end service is registered to the. NET Core Middleware, through which all REST communication is handled. The back-end is Python based whilst the front-end is based on a Node.Js/Vue.Js Web Tech-Stack.

## Design and Implementation Constraints

Long-term software and server maintenance/support will be decided upon at the completion of the TEDIAS project scope and the resultant situation/needs at that time. The implementation constraints also fall under the scope of that project.

## User Documentation

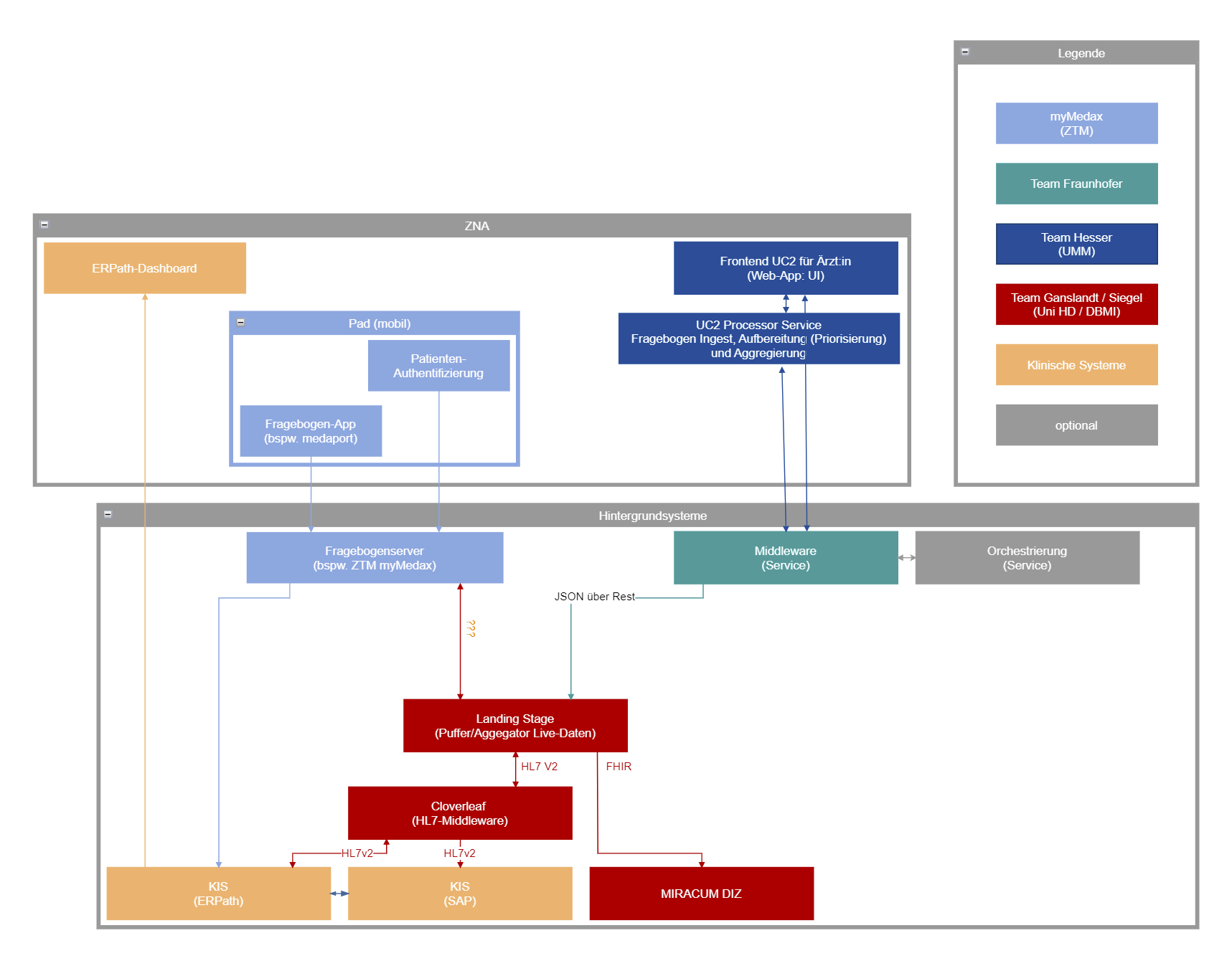
Role: Member of the neurological clinic

Full list of the Requirements basis with the stakeholders involved and their requests

|  |  |
| --- | --- |
| Issue Type | Summary |
| User Role | R: Patient |
| User Task | UT: Describe symptoms |
| Epic | ST: Providing my health card |
| Story | US: as a patient I want to be able to scan my health card so that I can use the dashboard |
| Story | US: As a patient I would like to able to fill out a consent form in order to agree to being part of a study and to being filmed, respectfully. |
| System Function | SF: Give information and consent |
| Epic | ST: Answer questionnaire |
| Story | US: As a patient I would like to be able to use the "HELP" button so that I can help me in case of ambiguity or problem the medical staff |
| Story | US: As a patient I would like to be able to answer the questionnaire so that the doctor gets a good overview of my symptoms and I don't have to wait a long time |
| System Function | SF: Adequately fill out the questionnaire |
| Story | US: As a patient I would like to be able to finish the questionnaire so that the results are transformed into the system. |
| System Function | SF: Closing of the questionnaire |
| User Role | R: Doctor |
| User Task | UT: Diagnosing |
| Story | US: As a doctor, I would like to be able to document my final diagnosis from the patient so that it is recorded in the system and the colleagues can help the patient in the best possible way |
| Epic | ST: Read pager notification |
| Story | US: As a doctor I would like to receive a notification if there is an emergency so that I can help the person concerned as soon as possible. |
| Story | US: As a doctor I would like to be able to receive a notification if a patient has specifically answered the questionnaire and his answer needs a possible quick treatment |
| Story | US: As a doctor, I would like to be able to see if a patient has answered the questions in such a way that he is not a case for the emergency room so that I ask the triage force to perform a triage in a timely manner, so that the patient can be referred to “MVZ or ÄBD” as soon as possible, so as not to occupy the emergency room unnecessarily. |
| Epic | ST: Evaluation of the questionnaire |
| Story | US: As a doctor I want to be able to click on the ICON and see the patient's results so that I can improve my diagnosis |
| System Function | SF: Open ICON |
| Epic | ST: Grading of the questionnaire |
| System Function | SF: Review the questionnaire |
| Story | US: As a doctor I want to be able to give my grade to the ranking of the patient so that we can check the efficient of the system and possibly the system needs an update |
| System Function | SF: Identify Problems |
| Story | US: As a doctor I would like to store my grade in the system and the storage so that we get a long-term study to evaluate the system |
| Story | US: As a doctor I would like to get a warning if my note and those of the system are much apart so that we may update the system |
| User role | R: Nurse |
| User Task | UT: Treat patient |
| Epic | ST: Deploying the tablet |
| Story | US: As a nurse I would like to point out in the patients that we are under data collection law and we are under silence |
| Story | US: As a nurse I would like to point out to the patients that we need a commitment so that we can film the patient and the patient becomes part of a study |
| Epic | ST: Lead patients to the waiting area |
| Story | US: As a nurse I want to be able to see if the patient has pressed the "HELP" button so that I can help the patient if necessary |
| System Function | SF: Get the notification |
| Story | US: As a nurse I want to be able to see if the patient finishes the questionnaire so that I can pick up the tablet and there can be the next patient |
| Role | R: IT Admin |
| User story | UT: IT-system construction |
| Epic | ST: Maintaining the IT-system |
| Story | US: As an IT admin I would like to be able to update the system if necessary |
| Story | US: As an IT admin I would like to be able to connect the devices on the secured WiFi of the clinic |

## Assumptions and Dependencies

All IT-dependencies are described in the Architecture diagram (see appendix). The Use-Case 2 Dashboard is integrated into the rest of the TEDIAS landscape via the .NET Core Middleware.

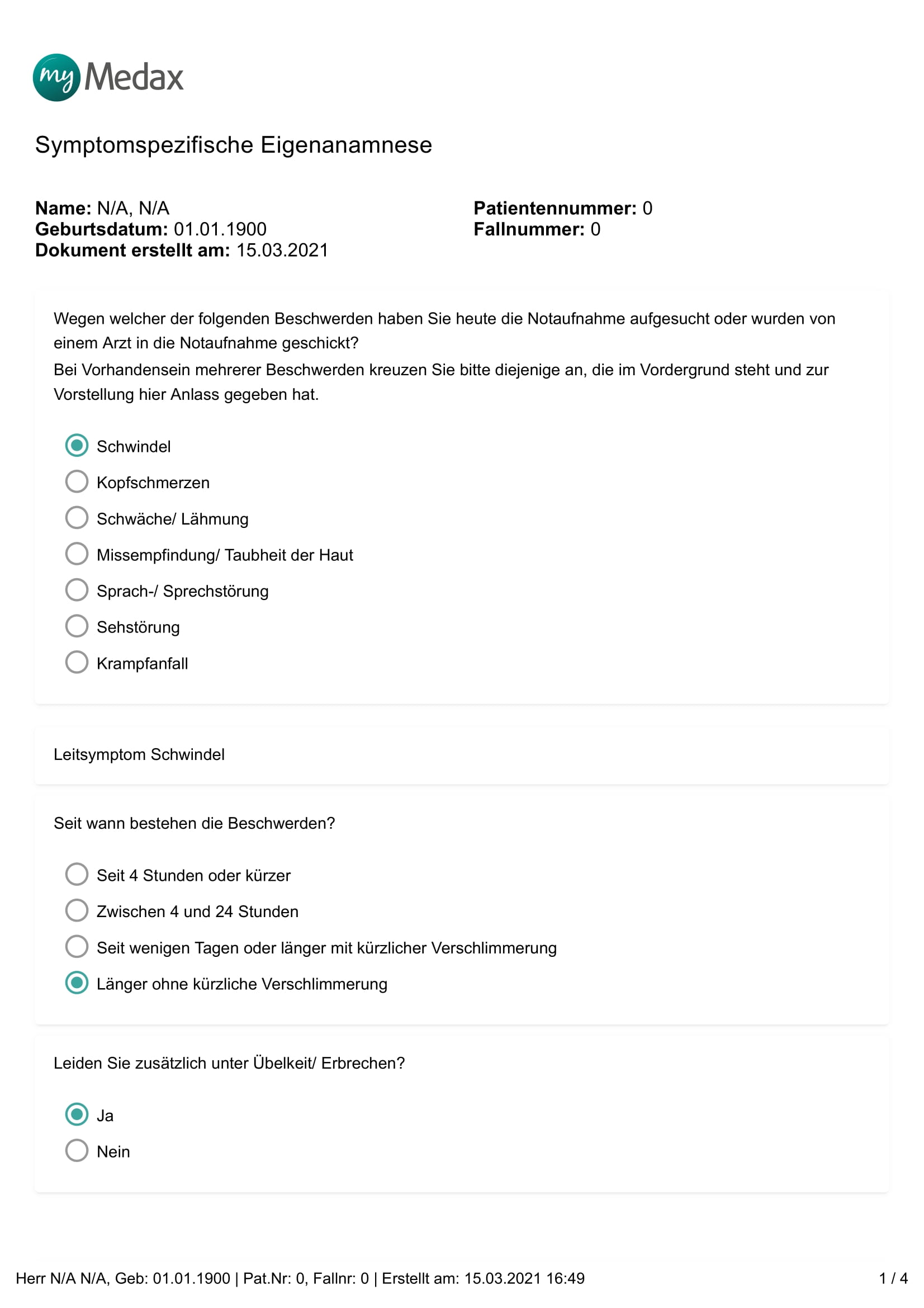


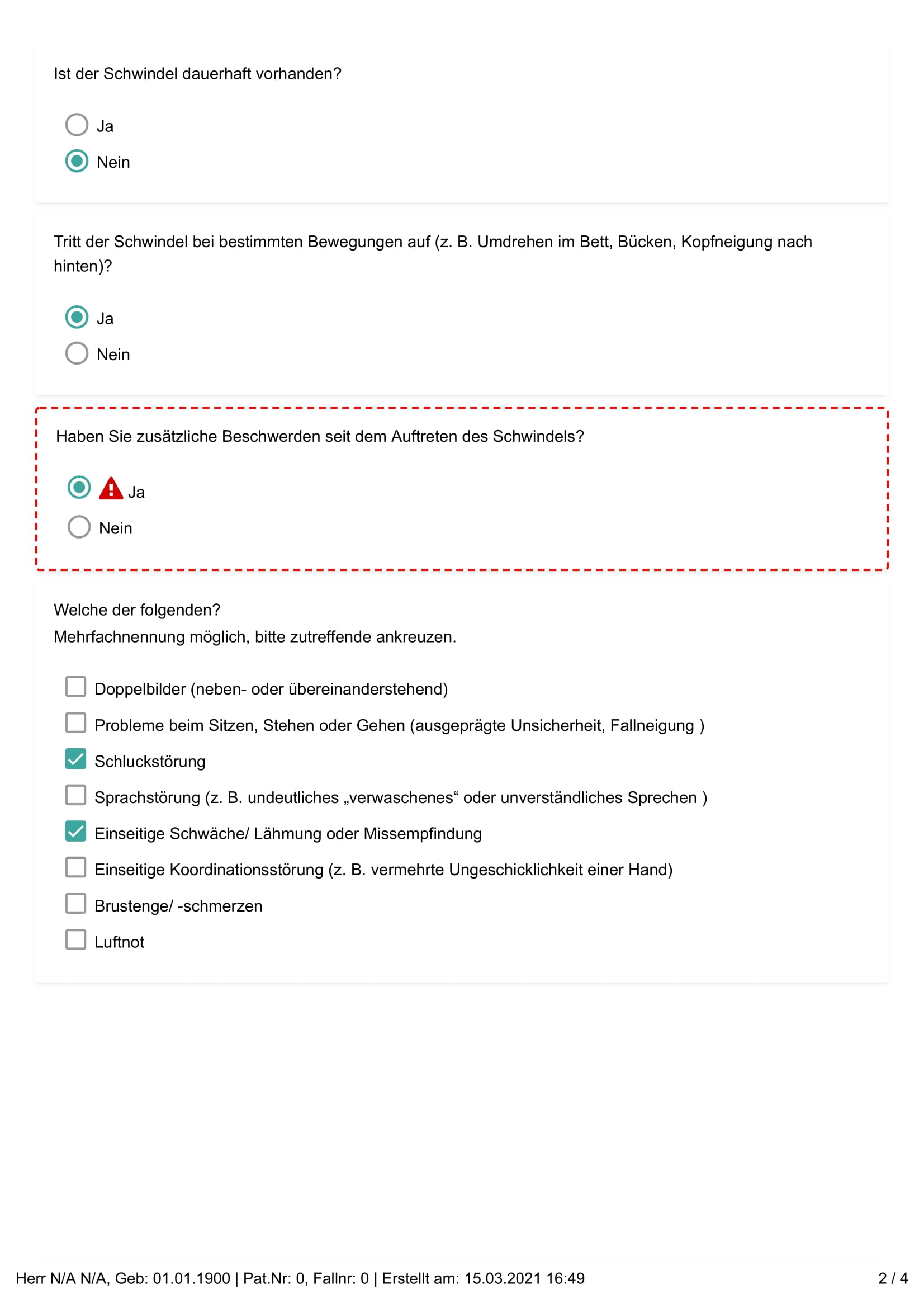
# External Interface Requirements

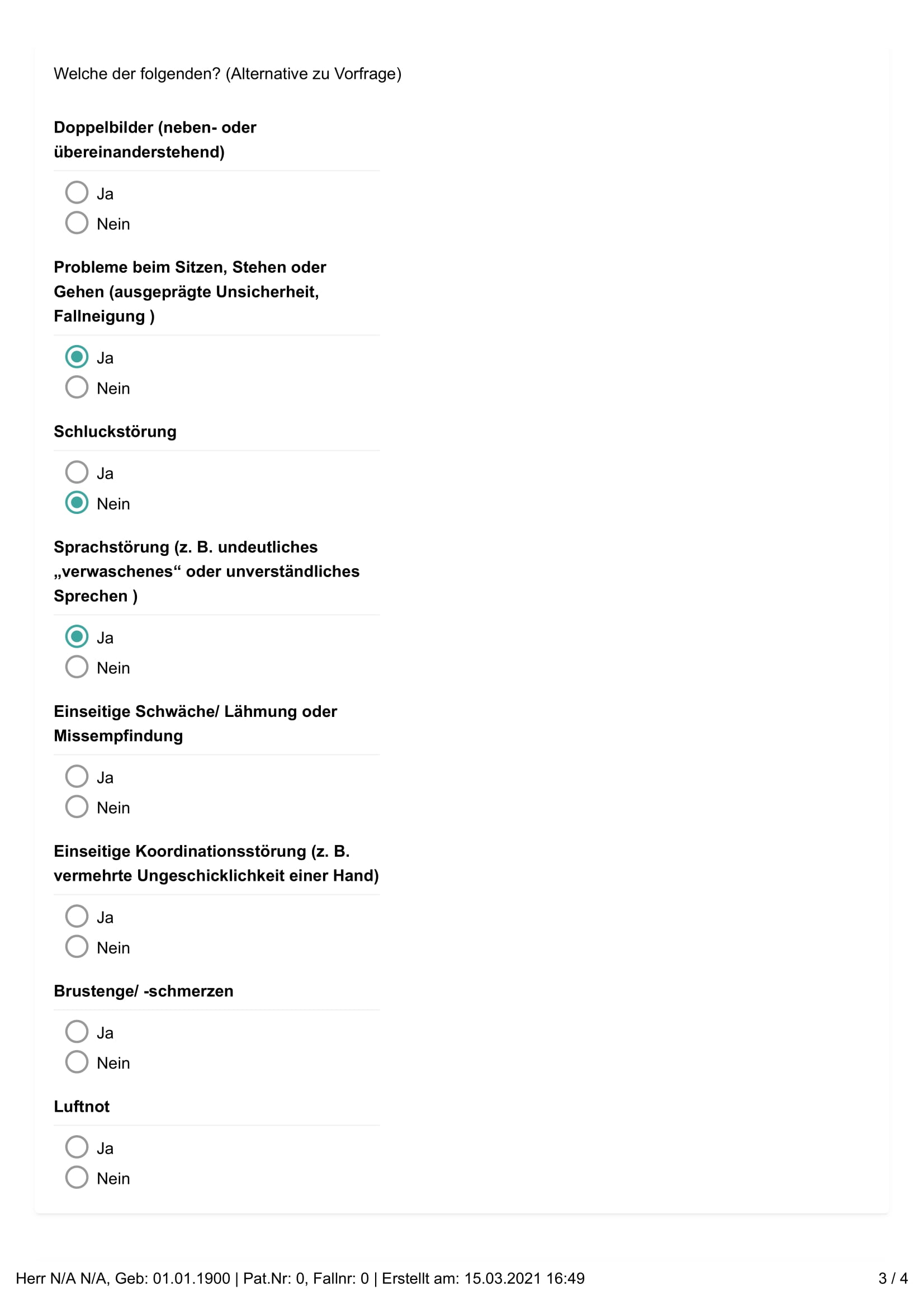
## User Interfaces

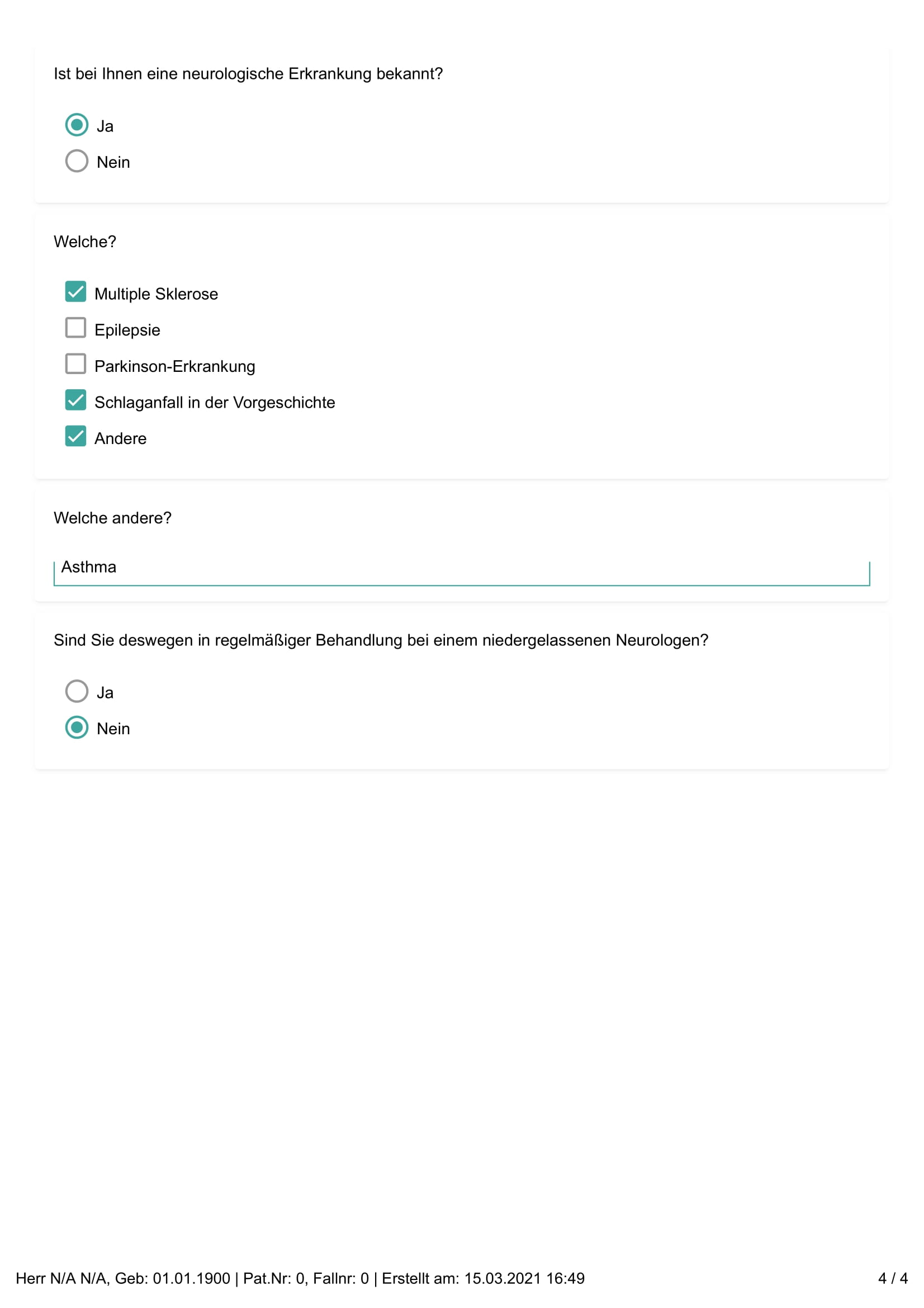
There are two versions of the user interface one for the patient and the other one for the doctor and medical worker

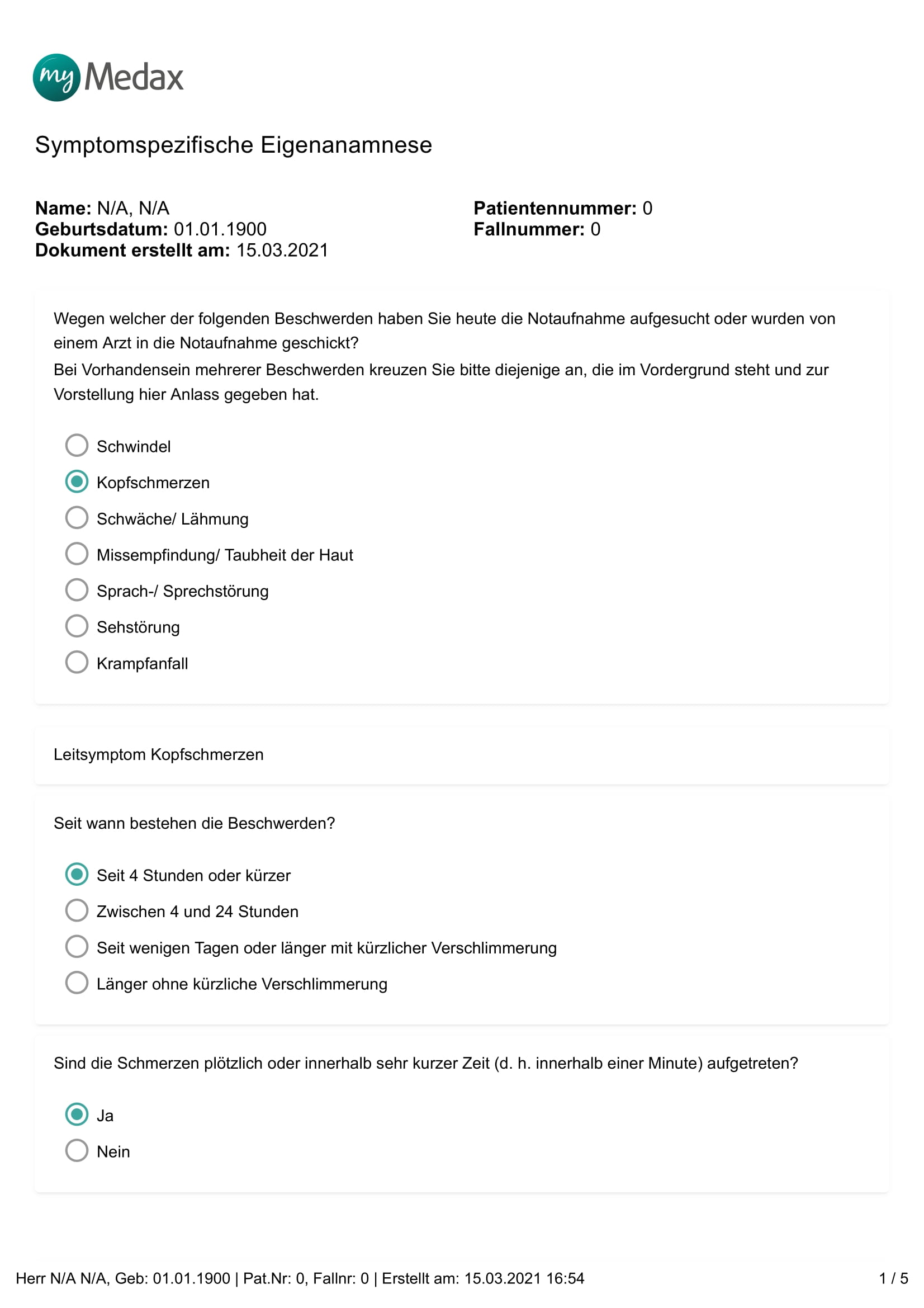
The user interface changes for the patient on the behave of the main symptom here are a few examples

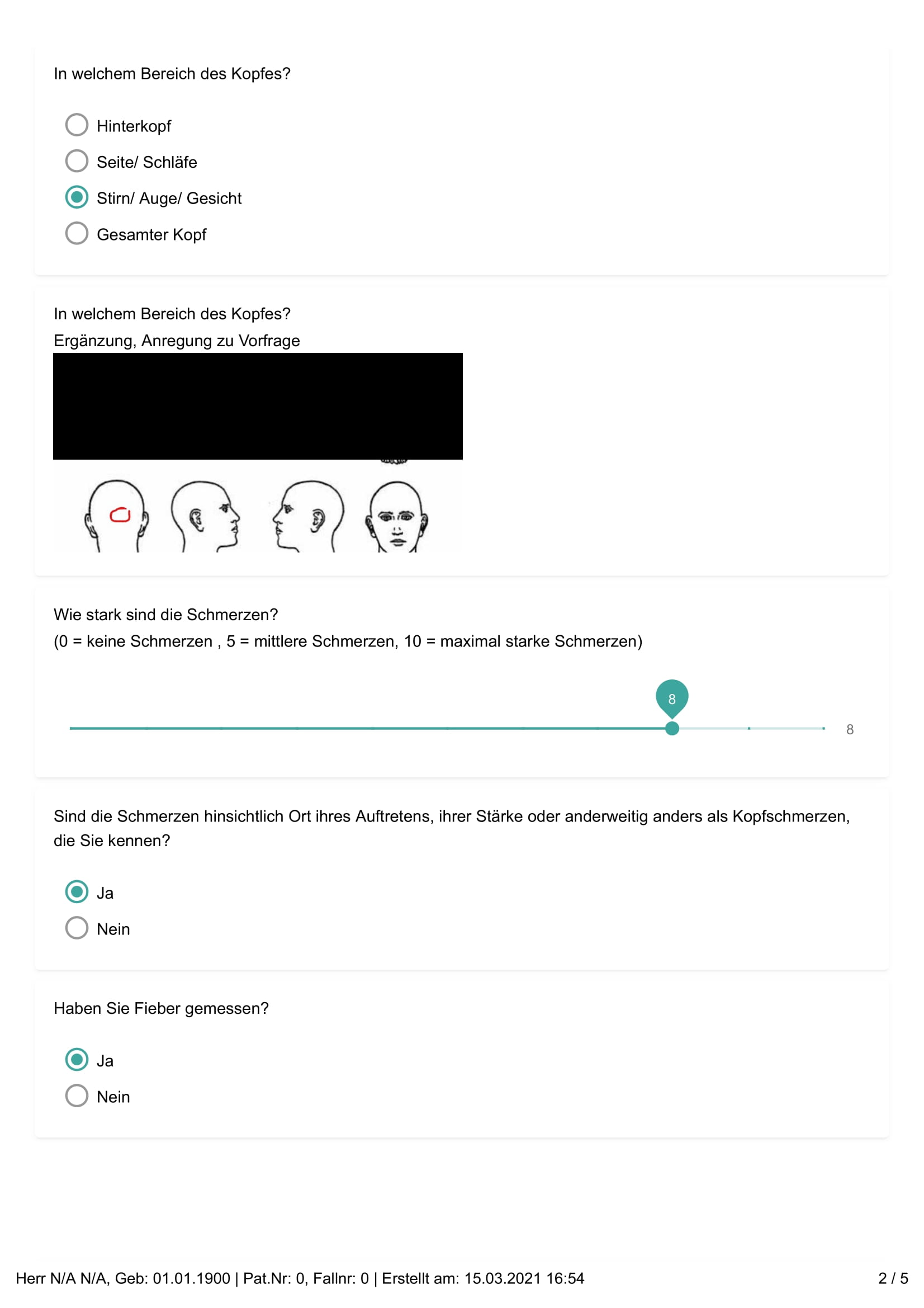


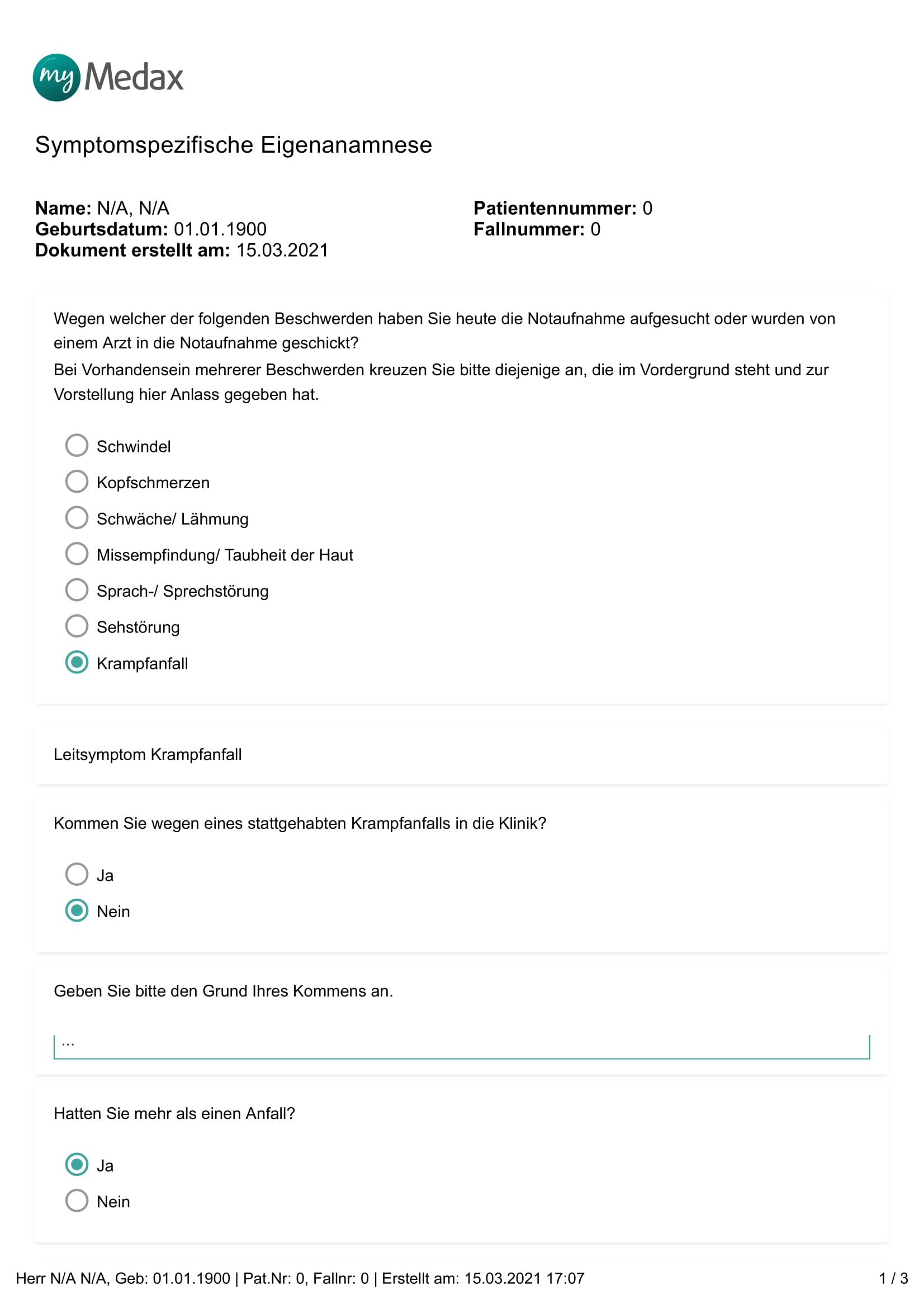


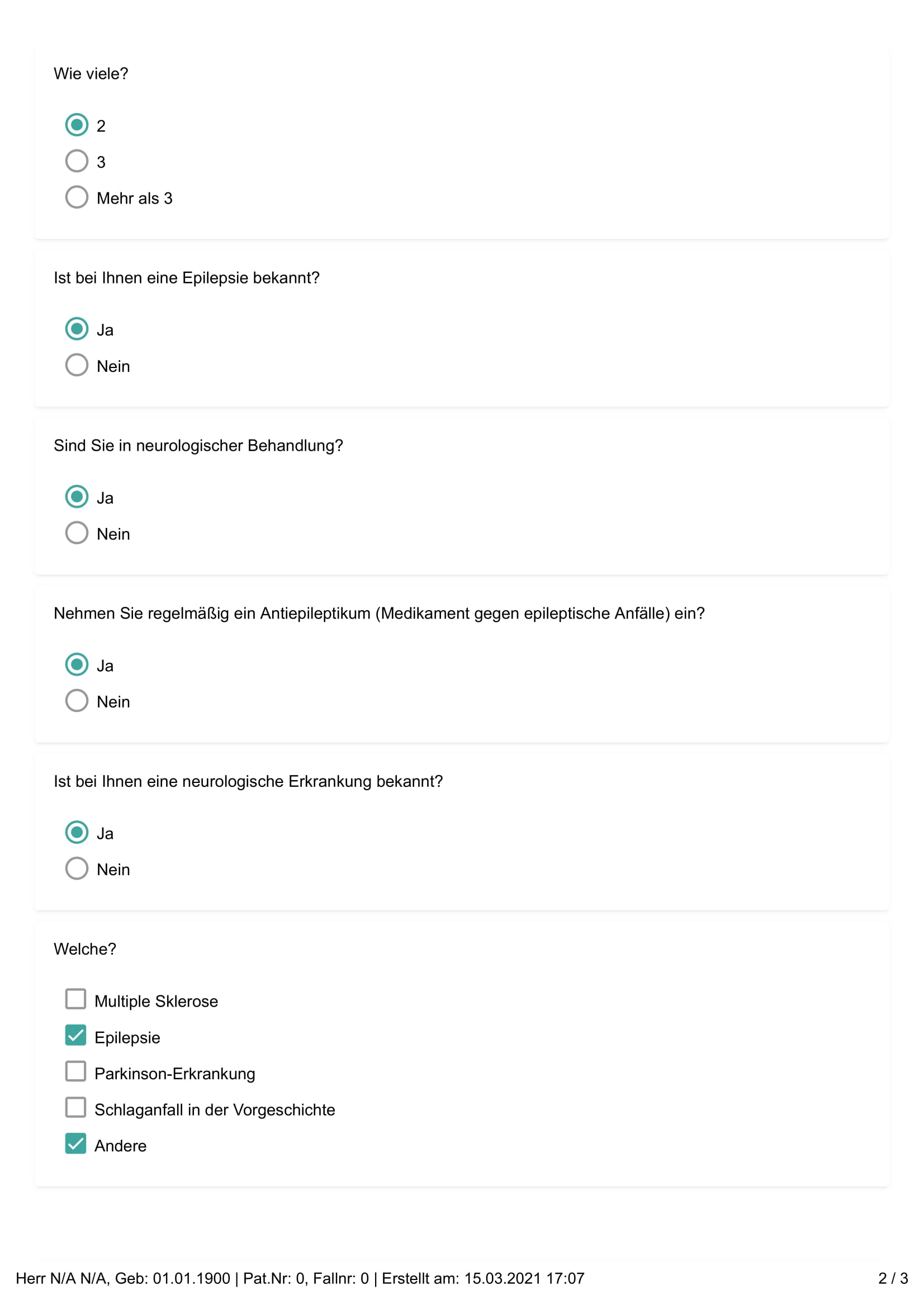


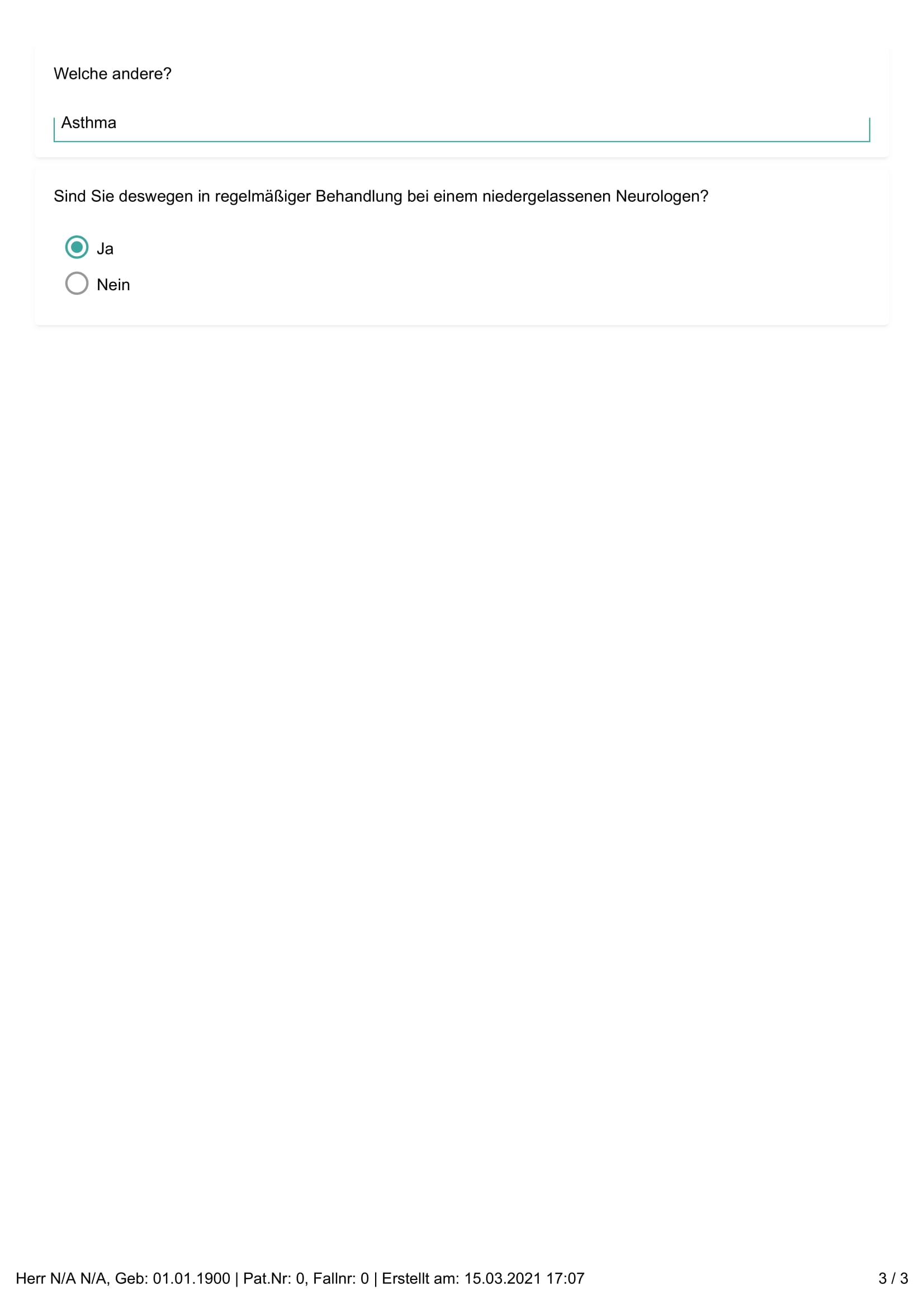






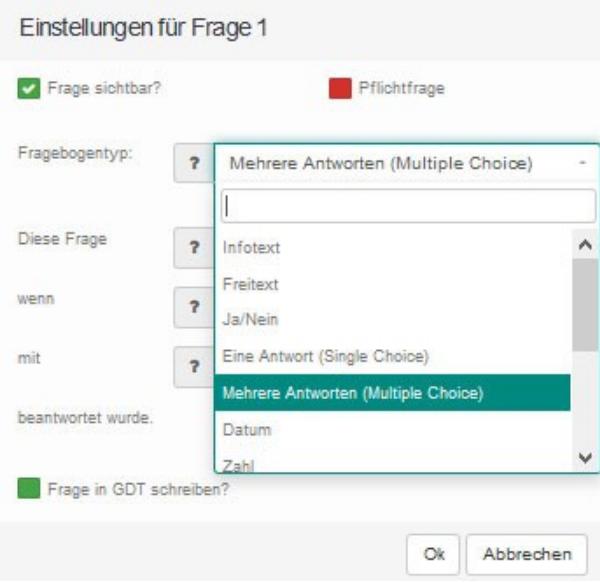


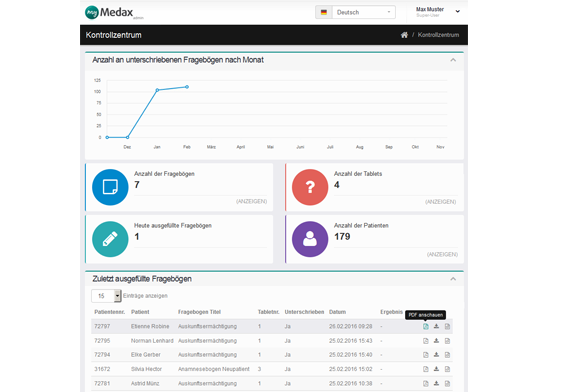




The questions are optimized based on the patients main symptom in this version there is one main symptom, there are also some questions that are universal for every main symptom, and there are multiple questions where the patient can select more answers

The medical staff and the doctors have a different user interface view. Doctor have the ability to implement new questions in the questionnaire and to modify the existing questions





The following are available for questionnaire design

Question types available:

Yes/No Questions

Single Choice Questions

Multiple Choice Questions

Date queries with calendar sheet display

Info texts

Querying Size

Query of weight

Drawings

Insert image files

Free text input

Signature fields

and the questionnaire is available in different languages

## Hardware Interfaces

This is out of scope for this project.

## Software Interfaces

The software will be containerised and deployed as docker images. Communication out of the docker containers are managed via the standard docker interfaces,

The use-case 2 Dashboard interfaces(see architecture diagram) with the TEDIAS Middleware(this runs on the Docnet CORE 5 Framework) as well as the Landing Stage and Orchestrator Services.

There are all managed via the Middleware.

## Communications Interfaces

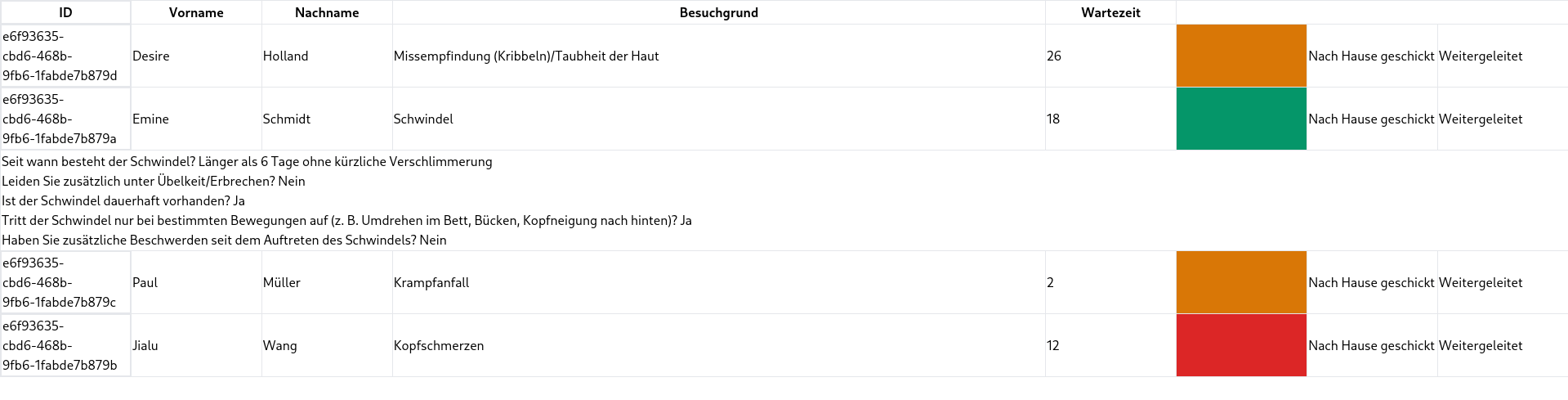
The backend and dashboard frontend will commnicate via the REST (Representational state transfer) standard.

# System Features

This section illustrates the full design of the dashboard according to pages and tabs in the software. Each page or tab will have its contents described by functional requirements. Functional Requirements are labeled F-REQ. Nonfunctional Requirements are labeled NF-REQ. The numbering scheme is continuous when considering functional and nonfunctional requirements for each part of the dashboard. For example, the homepage may have F-REQ-1.1 and NF-REQ-1.2 but not both F-REQ-1.1 and NF-REQ-1.1. The first number in the numbering scheme is always 4, since all requirements are listed here in section 4. The second number indicates to which part of the software is being referred. 4.1 is the Homepage.

## Homepage

The Homepage is the default screen when the Dashboard is opened. Figure 1 shows the homepage, with labeled red boxes indicating areas of reference for the functional requirements.



4.1.1 Description and Priority

High priority table updating after a short amount of time.

4.1.2 Stimulus/Response Sequences

Patients are added to the dashboard once they have completed the questionnaire. Once they have left the A&E stage, the medical staff clicks on the "Bearbeitet/Processed" button to remove a patient from the list.

4.1.3 Functional Requirements

* F REQ-1:Table of waiting patients
* NF REQ 1: Update of the table will be done every 5 seconds or if a new patient has been imported/added into the system
* NF REQ 1.2: The system function without any pop-up windows so it’s easy to use and it limits the possibility of confusing medical staff
* F REQ 1.3:The card shows the basic patient information that the doctor can use to identify the patient
  + F REQ 1.3.1: Patient ID
  + F REQ 1.3.2: First name
  + F REQ 1.3.3: Last name
  + F REQ 1.3.4: Main symptom
  + F REQ 1.3.5: Waiting time how long is the patient waiting
  + F REQ 1.3.6: Alert System(patient priority status ): This is a system on the homepage which shows the user all “alerts” which are currently active for all the patients in the ward. It summarizes the important updates and tasks which should be viewed or performed by the user, regarding each patient. All alerts for all patients on the ward are displayed here.
    - F REQ 1.3.6.1: These are divided into 3 groups of alerts: red, yellow, and green alerts
  + F REQ 1.3.7: Clicking on the patient row containing the patient’s name opens up a field containing the questions and answers from the questionnaire that were mainly responsible for the patient’s priority status.
    - NF REQ 1.3.7.1: The resulting addition should be displayed in a time not longer then 200ms
    - F REQ 1.3.7.2: Clicking on an other patient in the table will result in the closing of the side menu of the previous patient
    - NF REQ 1.3.7.3: Clicking on an empty spot to dismiss the alert will hide it until it has changed (e.g. new patients have been added)
* F REQ 1.4: Deleting a patient should result in a disappearance of the patient from the system
  + NF REQ 1.4.1: The deleting process should be done in a time not longer then 200ms
  + F REQ 1.4.2: Patients remain in the patient list until they are addressed or dismissed by a doctor.
  + Patient IDs are not reused within a given work day.

## System Feature 2 (and so on)

# Other Non-functional Requirements

## Performance Requirements

After turning on the device and after scanning the health card, the questionnaire should be ready to use. After a user clicks a button a response should be displayed within 200ms in order to avoid a feeling of insufficiency and incompetence on part of the medical staff, a feeling which can come about as a consequence of slow software. While the information given by the patients is being processed a loading bar should appear, signaling to the patient that the system is working on their info. Loading times should not take longer than 1 or 2 seconds. For repeat patients pre-loading data that is already in the system would reduce patient prep and wait time. Updates that are made while the dashboard is running should be fully loaded before being displayed to the user. The source of the request for the attributes comes from the interview and it is defined as pre requirements specification traceability.

## Safety Requirements

It is of the utmost importance that the safety of the dashboard be guaranteed as it translates directly into patient safety and data privacy. Seeing as that the dashboard is the only point of entry available for a patient to give patient information it is vital that it’s stability be protected as to prevent possible therapy errors a failure on part of the dashboard or the servers behind it would result in doctors not being unable to look up patient information and patient miss-ranking in the ranking system the end result would be that a patient has to wait a long time and that could worsen his situation. This includes regular nightly backups and offline data storage so as to be available even in case of an error occurring. The offline data format must have a distinct watermark as to ensure the doctor reading the information is made aware from when the information is and that it is out of date.

## Security Requirements

Dashboard data privacy must be a priority at all times. The IT team must diligently work to ensure the protection of the patient data displayed within the Dashboard. To ensure this, a sleep timer of 5 minutes is to be set up which would require the patient to re-login to access it again and should only be accessible from the Hospital network or Hospital VPN. All network traffic should be routed via the internal network. At no stage should any traffic be exposed to the public internet.

Access to the dashboard should not be available to any user within the university clinic network. Access should be restricted via a non-public username and password

Documentation for legal purposes. In certain projects or industries (for example, in a safety-critical environment such as medicine, as in this paper) it is necessary to comply with certain documentation regulations.

## Software Quality Attributes

The software has to be highly adaptable and flexible. Adapting to the environmental changes in the form of information provided by the user(s).One of the software key features is its continues availability and a state of readiness. Being robust, as defined in the computer science field, represents the system’s ability to cope with erroneous input and un-erroneous input Portability in is the usability of the same software in different environments for the software this is not necessary because it is specifically build for a hospital. Usability defined as time saved to be put to better use, i.e work with patients. Scalability is out of scope for this document. High availability is not a requirement at this stage.

## Business Rules

Mostly used for object-oriented software systems. Client creates business model, which contains artifacts from the business world. This provides the basis for the further requirements of the system. The contractor uses the Software Requirements Specification (SRS) to document the entirety of the software requirements. These are based on the ISO/IEC/IEEE standard.

This product is intended for use within a clinical setting. As such, the legal requirements are regulated by the "Medizinproduktegesetz" within the Federal republic of Germany as well as the EU General Data Protection Regulation (GDPR). Adherence to these regulations is essential. Further regulatory requirements are imposed by the Mannheim Universitätsklinikum GmbH Ethics Commission. Further regulatory requirements are imposed by the Mannheim Universitätsklinikum GmbH data protection officer. Medical confidentiality (German: Verschwiegenheitspflicht) is to be respected at all times.

# Other Requirements

ADD/REMOVE button for the medical stuff if a patient leaves the clinic UNDO button if a medical worker accidentally deletes a person from the waiting list.

If a patient needs to be escorted the data should be transferred to the next station so that the medical staff in the other clinic get the data in time. The data should be organized so that the medical staff can see how the patient’s info is categorized.

If a patient needs assistance to be transferred to the next station within the hospital.

# Risks

Risks are evaluated according to the method proposed by Crandall . Risks are computed on a

scale from 0 to 10, with 10 being the highest risk. The impact of Data Privacy risks is associated

with what kind of patient information is at risk. The impact of software function risks is correlated to medical errors which could arise if the software malfunctions. Mitigation strategies are presented

as recommendations and are not functional requirements of the dashboard.

The risks are evaluated and computed, using the method proposed by Crandall, on a scale from 0 to 10, with 10 being the highest risk. The impact of Data Privacy risks is associated

with what kind of patient information is at risk. The impact of software function risks is correlated to medical errors which could arise if the software malfunctions.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Risk** | **Category** | **Description** | **Overall Risk Rank (ORR)  = WRTS + WDIS** | **Mitigation Strategy** |
| Implementation | Software funktionalität | Red Flag Classification and Ranking System – Red flag stacking and/or severity (displayed in red color intensity) improves waiting list ranking. | 32,4 | In the next version, a better red flag classification should be inserted. |
| Data privacy | Data privacy | The user leaves the dashboard open and leaves patient information on the screen. | 26,4 | Blocking after a set of delay (i.e. 5 minutes) of no activity after which the user needs their password. |
| Data privacy | Daten Display | Patient data is displayed incorrectly. | 21,6 | Data is written in PyMySQL where it is possible to change the |
| Patients are swapped | Daten Display | Patient data is displayed or stored incorrectly | 19,8 | UNDO button in the next version of the software. However, if there is an incorrect input in the program, there is any possibility, reverse the option(UNDO) or delete the errors without deleting the patient from the queue. These are the ones that should be taken to fix an incorrect patient input. |
| Dashboard speichert nicht richtig | Data storage | The dashboard allows you to edit and add information to a patient record. As a result, there is a risk that data can be lost if the dashboard is closed or crashed and has not been saved properly | 21 | The dashboard should automatically save every 60 seconds or when information is entered somewhere and store a backup of the data on the protected hospital server. |
| Input error | Software funktionalität | If an input error is difficult to correct, this risk increases. | 15,4 | Any input in the program should have a way to undo or delete the error. |
| Implementation | Software funktionalität | Urgent pop-up windows for important information such as allergies or other pre-existing conditions | 13 | In the next version, an option should be inserted as soon as the doctor presses on a patient should important pop-up windows appear |
| Data protection risk due to theft | Data privacy | If the laptop or dashboard on which the software resides is stolen. | 18 | 1)Make sure that the dashboard program is properly password protected in addition to the laptop itself. The dashboard is browser-based, so an additional password should be included to access the hospital's VPN. 2) Stick a sticker on the back with the inscription this is the property of the clinic please return it. |
| Privacy risk through subtlety | Data privacy | This includes the ability for someone to look over the user's shoulder and collect patient information that they shouldn't have access to. | 15,5 | 1)Avoid large displays of patient information with identifiers on the same screen. If possible, avoid using patient images. 2)The patient should be made aware that other patients may be able to read his results, so the patient should be careful when filling in. |
| Data protection risk due to the storage of patient Ids | Data privacy | This includes storing patient identifiers in the software and how easy it is for an unauthorized user to search patient information. | 20 | Searching for patient information must be easy for the user. It is not advisable to compromise the usability of the dashboard to mitigate this risk. |
| Dashboard crashing | Software funktionalität | A high priority is also given to Internet and server connection problems. The dashboard is designed so that doctors can be updated information about patients at all times. If the Internet or backend servers are down, some information cannot be updated, and a doctor might make decisions in healthcare based on outdated information. | 11 | To avoid the risks, we can insert the following steps to ensure the normal functionality of the hospital A warning on the homepage that shows that the dashboard is currently offline and a date/time when it was last updated. So that hospital staff can get an overview again as easily as possible and the patients are cared for in the best possible way. |
| Privacy policy | Data storage | Patients are not willing to share their personal data | 2,5 | In case of data protection concerns, a doctor can have a personal conversation with the patient about the topic. |

Appendix A: Glossary

|  |  |
| --- | --- |
| Role |  |
| Requirement Engineering | Requirements engineering is a systematic and disciplined approach to specification and management of requirements with the following objectives:  1) To know the relevant requirements, to establish consensus among the stakeholders on the requirements, to document the requirements in accordance with specified standards and to manage the requirements systematically.  2) To understand and document the wishes and needs of the stakeholders as well as to specify and manage the requirements in order to minimize the risk that the system does not meet the wishes and needs of the stakeholders. |
| Questionnaire | With open and/or closed questions (i.e multiple-choice questions), stakeholder requirements can also be determined become. Questionnaires can be used to obtain a great deal of information in a short time and at low cost. |
| Use-case diagram | Easy-to-understand model for the overview-like documentation of functionalities of the system under consideration, their relationships with each other and the relationships of the system to its environment from a usage point of view |
| Use Case | A list of actions or event steps typically defining the interactions between a role (known in the Unified Modeling Language(UML) as an actor) and a system to achieve a goal |
| Actor | Persons or systems outside the system boundaries that interact with the system under consideration |
| ISO/IEC/IEEE standard | Provides exemplary outlines for software requirements documentation. Chapters can be divided into five thematically definable blocks:  Introductory information and general description of the software  List of all documents referenced in the specification  Specific requirements  Planned verification measures  Appendixes |
| Software Requirements Specification (SRS) | The software requirements specification lays out functional and non-functional requirements, and it may include a set of use cases that describe user interactions that the software must provide to the user for perfect interaction.  Software requirements specification establishes the basis for an agreement between customers and contractors or suppliers on how the software product should function (in a market-driven project, these roles may be played by the marketing and development divisions). |
| Pre-Requirements-Specification(RS)-Traceability | Traceability relationships are subsumed to those artifacts that precede the requirement in the course of the project |
| Post-Requirements-Specification(RS)-Traceability | Traceability information of requirements for artifacts that are downstream of the requirements during the course of the project |
| Requirements basis | Set of all requirements of a system |
| Stakeholder | individuals and organizations, who are actively involved in a project or whose interests as a result of project implementation or of the project completion positive or negative can be influenced. Stakeholders can also the project and its results influence. |
| Stakeholder list | Stakeholder lists are tools for identifying stakeholders. These are checklists with which relevant stakeholders are specifically and systematically determined  can be. |
| Request | 1) A condition or skill required by a user (person or system) to solve a problem or achieve a goal.  2) A condition or capability that a system or subsystem must meet or possess in order to comply with a contract, standard, specification or other formally prescribed documents.  3) A documented representation of a condition or property according to (1) & (2) |
| Functional requirement | A functional requirement is a requirement regarding the result of a behavior to be provided by a function of the system. |
| Quality requirement | A quality requirement is a requirement that refers to a quality feature that is not covered by functional requirements |
| Risk factors | Critical framework conditions of the project. These usually originate from human, organizational or content-related technical influences, most frequently human influences |
| Human influences | Social, group dynamic and cognitive skills of the stakeholders have on the selection of a suitable investigative technique  high influence. Whether the knowledge that is determined affects the individual  is explicitly conscious or rather self-evident and implicit  (i.e. hidden), also influences the choice of investigative technique.  Communication between requirements engineer and stakeholder  the quality of communication depends on the experience of both sides, as well as, for example, social skills, such as leading a conversation |
| Change of perspective | Stakeholders are thus encouraged to adopt a new perspective. This technique is excellently suited if stakeholders can only formulate their knowledge very one-sidedly due to a familiar, strongly narrowed point of view, but not if the requirements require a deep level of detail, otherwise the technology is very complex. |
| RUP | Mostly used for object-oriented software systems. Client creates business model, which contains artifacts from the business world. This provides the basis for the further requirements of the system. The contractor uses the Software Requirements Specification (SRS) to document the entirety of the software requirements. These are based on the ISO/IEC/IEEE standard. |
| Acceptance criteria | Acceptance criteria are conditions that a software product must meet in order to be accepted by a user, customer or – in the case of system-level functions – the user system.  Acceptance criteria consist of a compilation of information, each of which contains a clear result of passing or failing. The information specifies both functional and non-functional requirements and can be applied at the level of epics, features and the story itself. Acceptance criteria are also part of the "Definition of Done" |
| Attribute | Each instance in a data set is described by a number of attributes(also known as feature or variables).An attribute captures one piece of information relating to an instance |
| Clustering | Identifying groups of similar instances in a data set |
| Correlation | The strength of association between two attributes |
| Data Analysis | Any process for extracting useful information from data. Types of data analysis include data visualization, summary statistics, correlation analysis, and modeling using machine learning |
| Database | A central repository of data. The most common database structure is a relational database, which stores data in tables with a structure of one row per instance and one column per attribute. This representation is ideal for storing data with a clear structure that can be decomposed into natural attributes |
| Data set | A collection of data regarding to a set of instances, with each described in terms of a set of attributes. In its most basic form, a data set is organized in an n\*m matrix, where n ist the number of instances(rows) and m is the number of attributes(columns) |
| Decision tree | A type of prediction model that encodes if-then-else rules in a tree structure. Each node in the tree defines one attribute to test, and a path from the root node to a terminating leaf node defines a sequence of tests that an instance must pass for the label of the terminating node to be predicted for that instance |
|  |  |

The structure of a role should be the same.

* Naming convention R: <<Name>>
* A role is an abstract summary of needs, interests, expectations, behavior and responsibilities.
* A role is described by
  + her name,
  + their tasks (UT)
  + Success criteria (i.e. when did the role do its job well)
  + Communication partners in the execution of tasks
  + Degree of innovation (i.e. the role already existed before)
  + Rough user profile, i.e. knowledge/experience/skills
    - regarding tasks
    - regarding software system

The structure of a user task should be the same.

* Naming convention UT: <<Name>>
* The name of a subtitle must comply with the grammar: <Object> <Verb >suffice
* A UT is a responsibility of a role that produces a coherent outcome, i.e result = recreational athlete has way for her ride. It is independent of IT support (task level)
* The description can explain this UT in more detail

The structure of a sub task should be the same.

* Naming convention ST: <<Name>>
* The name must satisfy the grammar: <Verb > <Object>
* An ST is assigned to a UT. It corresponds to a completed sub-goal of the UT. It provides a partial result and is not good to interrupt during implementation.
* The ST is independent of IT support
* All STs that are important for the UT are described, even if they are not to be supported by the software.

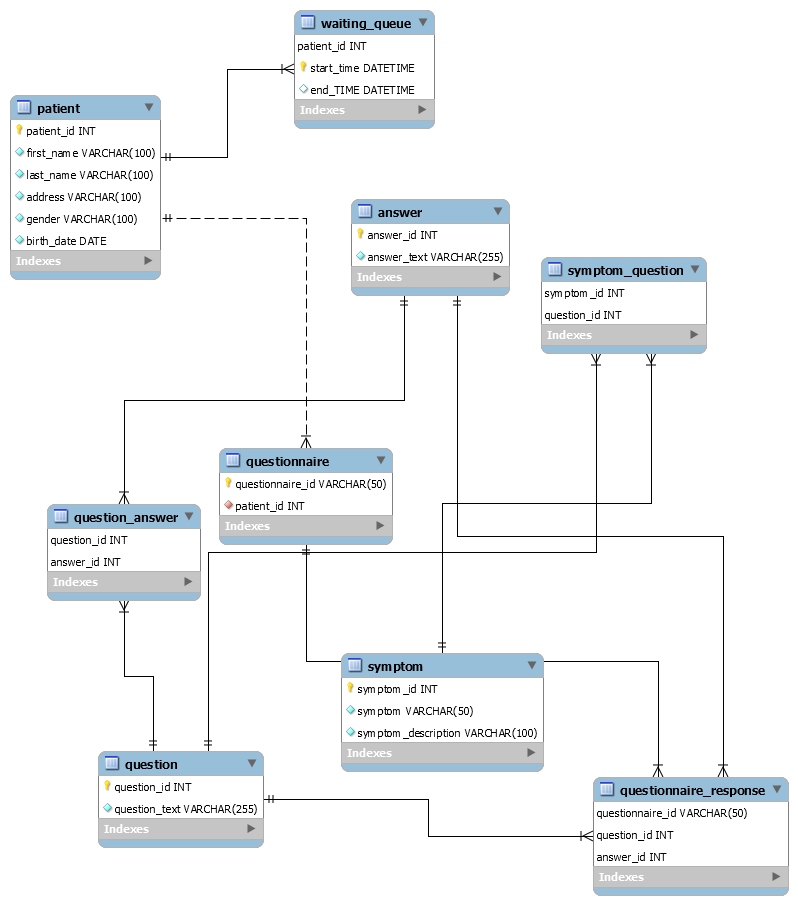
The structure of a user story should be the same.

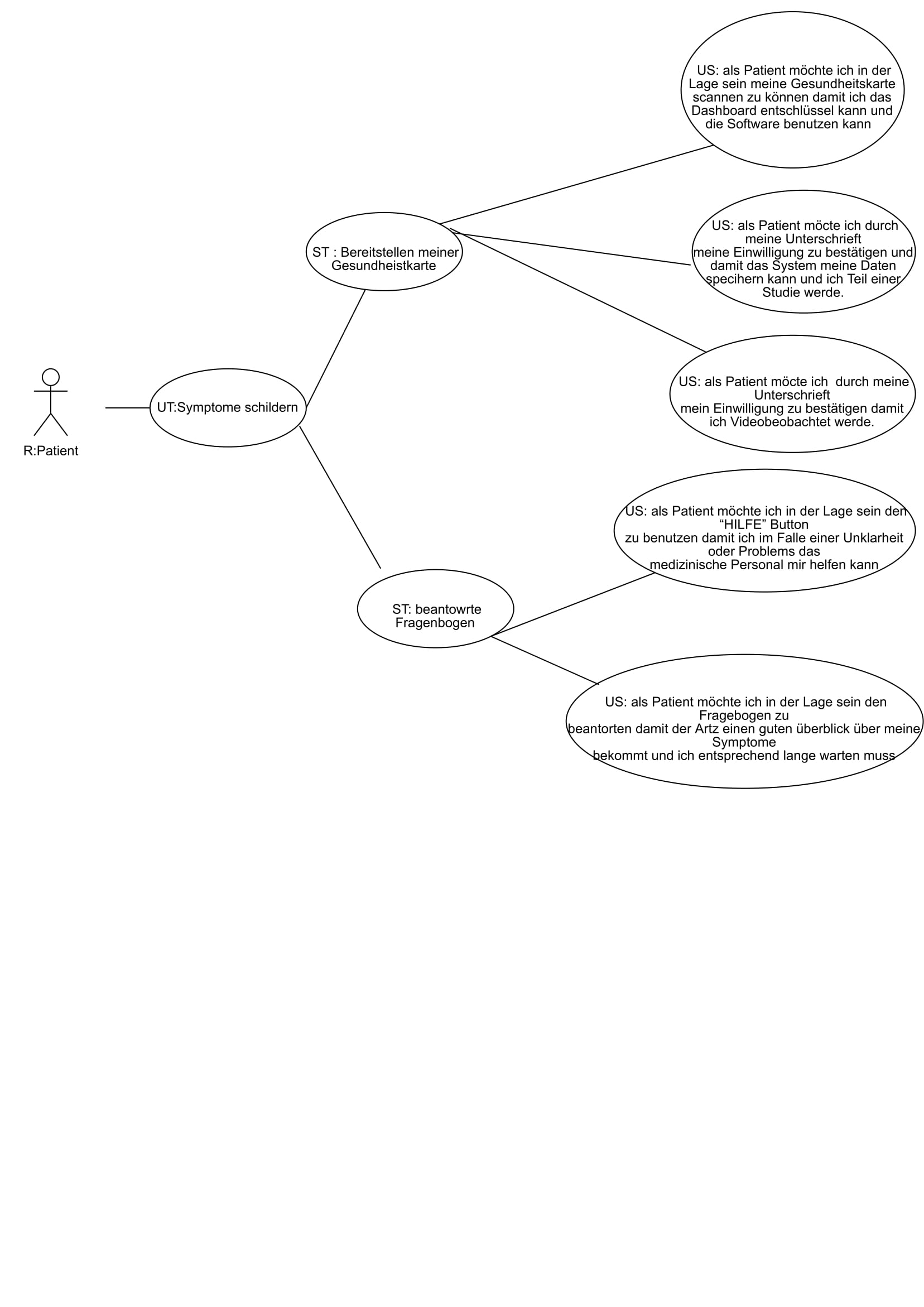
* Naming convention US: <<Name>>
* The name must comply with the grammar As<Role> if I want to do <activity = something> so that I <foundation = achieve advantages>
* US describe ONE functionality of the software. It is therefore IT-dependent.
* US are problem-oriented, not solution-oriented (no GUI details)
* Justification of a US is motivated by
  + A need or frustration of a persona
  + Using the result in another US, Epic, or User Task
* The description must be explained in more detail by the US, e.B. by specifying required and generated data, pre- or post-condition or by specifying acceptance criteria.
* US meet the criteria for good requirements
  + Clear terminology
  + Estimable (can be planned and prioritized)
  + As a set without explicit dependencies
  + Complete as a quantity
  + As a set conflict-free
  + As a set without redundancies

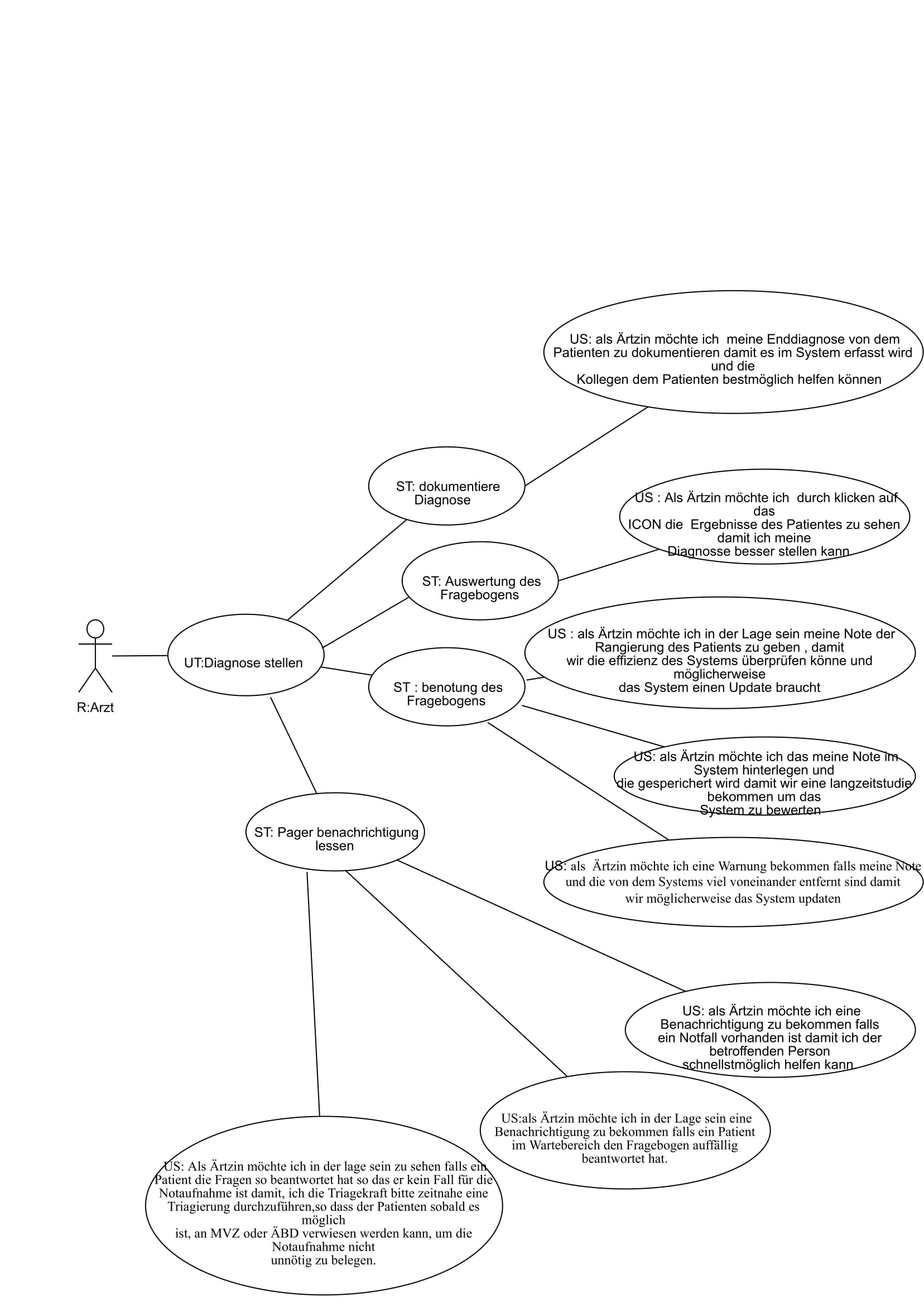
.

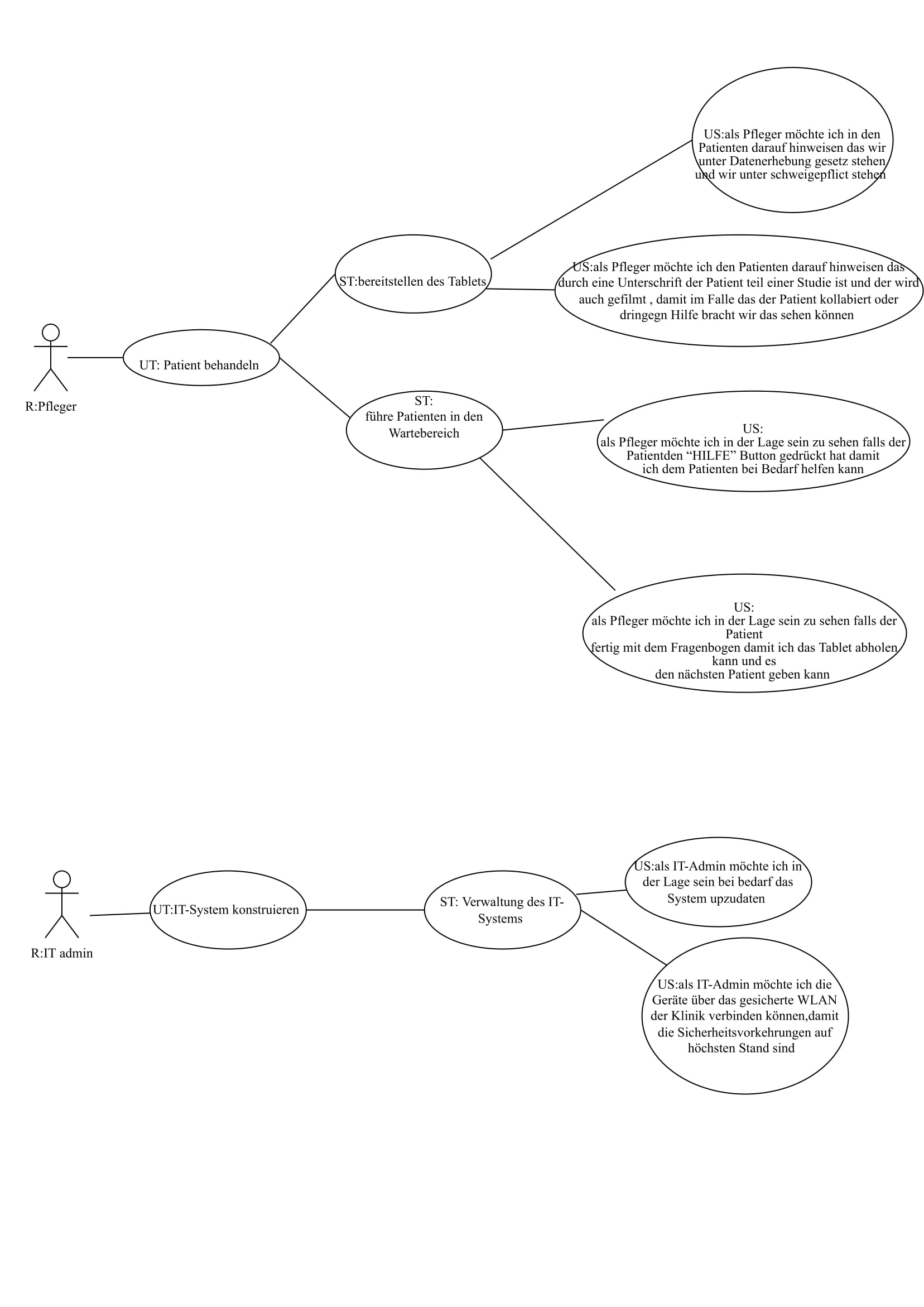
Appendix B: Analysis Models

Verbal or direct face-to-face communication is an important tool in requirements engineering because of interactivity. in practice, however, there are various situations that prevent or complicate verbal communication. In addition, pieces of information are sometimes so complex that verbal communication is inefficient. a written text or different diagrams can be read again and again. The principles of UML and RUP are important to showcase some connections between user cases in the project









Appendix C: To Be Determined List

TBD-1: List of symptom keywords for the TEDIAS Avatar to flag and display during the patient discussion in the Cabin.

Appendix D: Possible future features

Future feature suggestions:

Red flag classification and ranking system – red flag stacking and/or severity (displayed in red color intensity) boosts wait list ranking.

Ranking ascension via time spent waiting – a time threshold is set so that if a patient has spent a significant amount of time waiting, his ranking is boosted. For example, a green flagged patient gets boosted into the orange category.

Summary view – Complete patient statistics displayed in an orderly manner, all in one place. Category, age, symptoms, treatment if administered and planned. All compared with the system approximated wait and treatment time. Average daily wait time should be recorded, on workdays as well as weekends and holidays, and compared accordingly. With the addiction of those statistical features in a future version of the software the process of data analysis would be much easier and important decisions would be pack up with numbers and so it allows doctors to maybe reorganize the working times to see on which days there are more patients for example weekend days so that on those days there should be more doctors on that station.

A statistic based on the waiting time of patients, grouped by their main symptom, could be helpful in reorganizing the questionnaire if necessary. To see if a main symptom group’s wait time is on average disproportionately longer than the norm.

Statistics – Long term statistics displayed retroactively in order to asses treatment success and efficiency, this could help with the long-term data analysis, with maybe clustering some data.

A better sorting feature: Enhanced sorting feature, e.g. group by priority color, waiting time, symptoms, etc. History of the patient should be somehow involved in the ranking procedure.

Patient consent issues needs to be considered (this is dealt in other parts of the TEDIAS projects) as people are not keen to share some of their experience.

Urgent info pop ups - bubbles displaying vital, urgent information (i.e penicillin allergy) which can be closed at will.

Digital patient record for better information sharing with other medical staff within the hospital visit.

An internal communication system for doctors and medical staff which would ensure accurate info is being passed on immediately with the addition of TO DO lists (i.e check blood pressure) with feedback slots for the staff to fill out.

As soon as a patient has entered hazard variables such as allergies in the system, these are also displayed every time the patient data is called up.

RISK ALT

The system is limited to one mayor symptom per patient per visit, greatly reducing its diagnosing capabilities which can, in turn, cause needles patient stress.

In pregnancy, the symptoms would call for a false ranking placement thus would lead to inefficiency. In the decision tree that is implemented in the algorithm that woman would be in a disadvantage. In the case of patient pregnancy, misclassification may occur due to inappropriate question answering.

People with some psychical problems could be under more stress if they need to agree on being filmed and being part of a study. The system would be better to separate those two requirements.

There is no alarm that the device left the building if a patient, intentionally or unintentionally, takes the device with him. Especially older patients tend to forget to bring it back to the medical staff.

There is no red flag for patients that answered “krämpfe/cramps “as their main symptom, cramps could be an important indicator that there is a more serious reason behind it or in pregnancy where it is possible to have more symptoms that overlap.

Patients that have a specific chronic diseases like Multiple sclerosis would find it difficult to choose for a main symptom and the lack of a “patient history” field could rank that person falsely.

The lack of red flag at the field of “Are you aware of a neurological disease?” question and the available answers.

If the main symptom is “Weakness/paralysis/discomfort?” in the following question “Where are your complaints?” the possible answer should be split up into two possible answers instead of one “Head/face – one-sided/double-sided”.

When the user would leave the dashboard open, leaving the patient’s information on screen.

Were the Dashboard to crash and not be able to restart, the user would not be able to access the patient’s information or enter new items into the Patient File in that period.

The Dashboard itself is designed so that users can access updated information on patients at all times. Should the intranet or backend servers crash, some information may not be updated, and a user could make healthcare decisions without new, relevant info and thus make a mistake.