Technical Implementation

# Context: Domain Situation (10%)

The German and Bavarian Red Cross utilize the Verletzenanhängekarte (injury attachment card) to identify and record sick or injured people in the case of a large crisis or tragedy. It was adopted in 2002, to standardize the procedure across the nation. The sighting outlines treatment priorities and provides the first official evaluation of the affected person's health. Additionally, the initial actions conducted are noted, and depending on the circumstance, the transport priority is established at a later time. All of this improves the situational awareness of everyone involved, enabling them to track down service and deliver information (Deutsches Rotes Kreuz, 2004).

The domain situation here is defined as:

* Paramedics working at a mass emergency site sighting and registering patients are the target users
* The Verletzenanhängekarte by the German/Bavarian Red Cross should be abstracted and digitalized
* An interactive map and form should be used as visual and interaction idioms
* The algorithm and computation have to be as efficient as possible and compatible with tablets

Currently, The Verletzenanhängekarte is laminated, and it's filled out with a marker. With various colored stripes, the priorities are set. This approach is ineffective since the writing can be smudged and it takes too long to fill out the card's back. The card's digitization ought to solve these issues.

In the technical implementation the use cases of sighting, recording and finding the patient are prioritized and executed. After the patient was initially recorded, the paramedics have to find the highly prioritized patients for transport or more intense treatment. Therefore, the paramedics can look at the map, find the patients through the GPS of their QR Codes, and pull up their cards on their Android tablets. If needed they can edit the information provided.

# Translation (30%)

This section of the paper focuses on the explanation of the data and task abstraction in the Verletzenanhängekarte digitalization. As the users will be significantly reliant on the software to function, the design choices for visual encoding and interaction idioms underline the significance of a user-centered design. The design's selection of markers and channels is also discussed, with an emphasis on guaranteeing quick information access and exchange.

The digitalization focuses on the representation of sighted patients as QR codes in the application.

The conceptual model of the Verletzenanhängekarte project is based on a mental construction of the task of treating patients in a mass emergency situation. The dataset consists of these QR codes which serve as items with various attributes, that can include geographic position, prioritization level, and patient information. The attributes can have geometric value, such as position on the map, and sequential, ordinal value, such as patient priority through color coding in a flat table. The data model is a mathematical abstraction of the datasets with these attributes, which is transformed to support the paramedics' reasoning and treatment process. The flat table format makes it easy for the paramedics to view and sort the information, while the geometric and ordinal attributes help to contextualize the patient's situation and prioritize their treatment. The cardinality of the dataset in the Verletzenanhängekarte application is the number of sighted patients, as each patient is represented as a separate item in the dataset.

For each attribute, the cardinality depends on the type of data it represents. For categorical data, the cardinality is the level of priority in the attribute. For quantitative data, the cardinality is the range of values the attribute can take, such as the patient's GPS location.

When determining the cardinality of each attribute, it is important to consider the task at hand and the purpose of the attribute. For example, it may be necessary to transform the data, such as scaling the patient's location to fit within the range of the map. This would allow the map to accurately display the patient's location and provide a clear visualization of the task.

The data abstraction process in the Verletzenkarte project relies on the conceptual model to efficiently encode the information, making it accessible and easily understandable for the paramedics in high-pressure situations.

The task abstraction involves identifying the actions and targets for the paramedics using the system. The Verletzenanhängekarte project employs the following actions:

* Actions: Analyze
  + Consume: The paramedics using the digital Verletzenanhängekarte will consume information from the patient's form that has been filled out and stored in the database.
  + Produce: The paramedics will have the ability to annotate and record any additional information they gather while treating the patient. They can also derive new information from the patient's data.
* Actions: Search
  + Locate: The paramedics can locate the patients on the map by scanning the QR Code provided to the patient or by typing the code into the search area of the application.
* Actions: Query
  + Identify: The paramedics can identify the patient's information, including their prioritization, by looking at the markers on the map or by searching the patient's form.

The targets of these actions are the patients and their information stored in the system. By defining these actions and targets, the design of the digital Verletzenanhängekarte can be optimized to support the paramedics' needs and ensure that they have access to the information they need to make informed decisions and provide effective treatment.

The design of the digital Verletzenanhängekarte focuses on making the information accessible, clear, and concise for the paramedics through the use of color coding, markers, and shapes, and visual encoding and abstraction idioms. It was discussed that the paper based form should be preserved. That is why, the HTML form was created to resemble the original Verletzenanhängekarte as closely as possible and logically related data is separated through space, headings, and changing input fields. This design choice emphasizes a user-centered approach, ensuring that the paramedics have quick and easy access to the patient information they need to provide effective treatment in high-pressure situations. The user heavily interacts here with the application through change parameters with buttons, radio buttons, checkboxes, text fields, controls, and canvas.

Contrarily, the symbol map has more design freedom because it hasn't yet been established. The user and the sighted patients should be located on the map. Since each user is distinct on the map and thus their visualization is less significant, the Leaflet map's mark is used. On the map, the patient marks are more significant. Each mark serves as a visual representation of a patient and provides the paramedic with crucial information. The symbols used on the map to represent patients are geometric primitives that are made to stand out aesthetically. The markers' colors were taken from the official color scheme of the German Red Cross.

The channel properties of the marks, such as color and shape, have been chosen based on their effectiveness in accurately conveying information to the human perceptual system. The channel effectiveness of the marks is crucial for ensuring that paramedics have access to the information they need to make informed decisions and provide effective treatment in high-pressure situations.

The overall design of the website is kept simple and minimalistic, with little interaction, to prioritize the task and data and make it easy to use even in stressful situations. By carefully considering the design choices for visual encoding and interaction idioms, the Verletzenanhängekarte digitalization aims to provide an efficient and user-friendly solution for paramedics during mass emergencies.

This application is designed for the use on tablets.

# Implementation (50%)

The backend of the website is constructed with PHP, the frontend with HTML, and the interactions are built with JavaScript. The website is implemented on Apache and is accessible through XAMPP. The SQLite DB Browser was used to build the database, and SQLite3 was used to connect to the website. The database was created simply because the visualization aspect of the project was the main focus; otherwise, the data could have been organized into the various sections of the card and would have been FHIR-compliant.

The search page is the website's landing side. It is specifically implemented to be used for QR Code patient searches. So, the only element on this page is the search bar.

The upcoming forms can be accessible by either scanning or putting in the QR code (this option wasn't considered here because it couldn't be tested). The user will be redirected to the input form if the RGB color code for the QR code being searched is present in the "farbe" column; otherwise, they will be forwarded to the output form. As will be explained later, two forms were used. The 'farbe' column in the entire application controls whether or not the patient is regarded as recorded because it is only filled in when the algorithm has chosen the proper color with sufficient input data. This column in the input form is filled in manually because the algorithm is not yet setup.

The description of the following pages and functions are explained in the order of prioritization, but where implemented in opposing order due to the complexity of each function.

Graphical user interface, text, application

Description automatically generated

Figure 1: Input form

This form's objective is to record a paramedic's patient sighting information. The form is created as previously described. Logically related data is separated using breaks and paddings, headings, and switching types of input fields between multiple forms, as shown in figure 1 between "Diagnose" and "Transportinformation." Since headings are missing from the paper-based Verletzenanhängekarte, they were added separately. They offer a logical explanation of each form's purpose. This makes it easier for paramedics, even those who are not familiar with the Verletzenanhängekarte, to get the information they need immediately.

In order to reduce space on the website and improve efficiency, smaller subsections like "Transportinformation" and "Innnenliegenede Suchdienstkarte," as well as "Zustand/Uhrzeit" and "Erst-Therapie," were implemented to be next to each other.

Some HTML form fields feature prepared inputs to increase efficiency, allowing users to select from them rather than typing their own values. It becomes quicker as a result, and the receiver can understand it more easily. Based on the location of the emergency site and the resources available to the fictional paramedics team, the solutions were chosen.

The text fields are made to be as large as possible to accommodate all the required data, yet small enough to prevent unnecessary waste of space and prevent the user from scrolling endlessly.

A formality that can be automatically filled out to save time is the date field that is prefilled and unchangeable.

The image that is used in the original card is also implemented digitally and can be annotated with fabric.js in order to maintain the original card's integrity. Red was chosen as the brush color to provide the sharpest contrast. Fabric.js was used to annotate the image as it is compatible with touch.

None of the fields must be filled out because doing so will either not always be possible for the user (for example, if the patient is unconscious) or will not be necessary due to the sighting and recording process.

The explanation of the input form also applies to the output form.

Graphical user interface, text, application, email

Description automatically generated

Figure 2: Output form

This form's objective is to output patient sighting information, with the possibility for paramedics to modify and add information.

When looking up an QR code, the user is moved to a different form if the patient has already been recorded. There are a few minor variations between this form and the previous, but overall they are similar.

The "Änderungen speichern" button that appears next to each heading indicates that this form is actually made up of multiple smaller forms. Due to their compatibility with the smaller form's subdivisions, these more compact forms allow for more accurate alterations to be made to a unit. Another advantage is that mistakes committed unintentionally when stressing out or scrolling the page won't be accidentally preserved.

The main heading and the texts' various colors are another distinction from the first form. The color kept in the database, which is produced by an algorithm that establishes the priority, serves as the basis for these hues. The page's strong, colored heading draws attention to the patient's priority level. The user can clearly tell which priority they are dealing with thanks to the colored wording used across the page.

Finally, the previously entered information is retrieved from the database and shown in the appropriate input fields so the user can view the data required and also make changes to it more easily.

Map

Description automatically generated

Figure 3: Interactive symbol map

The patients who have already been recorded are displayed on the interactive map. The map was made with Leaflet, as was mentioned in the Translation section, because it allows interaction through touch. The markers were created using the LeafIcon class, and the database's values for color, longitude, and latitude are what define their position and color. The marker's popup information is constructed and shows the patient's name, a short diagnosis, and the respective QR code. The view is fixed to the user´s location and follows the user because the Locate Control displays the user's current location.

Each marker on this interactive map has a legend underneath it so the viewer can quickly understand what they mean.

As before, the locations are currently filled out in the input form because there aren't any real QR Codes with GPS.

The map fills the majority of the screen because it spans the entire width of the browser and has a height of 500 pixels. As a result, the map is less condensed while still displaying a lot of information.

The sole interaction method available on the map is selecting. The user can navigate the map and see and read the necessary information by clicking and zooming intuitively. As hovering is not possible on touchscreens, this option was not available.

The navigation bar of the website is also created simple and subtle, so it does not distract from the information displayed and is easy and intuitive to use. It Is also fixed on the top of the page, so quick navigation is possible.

Each PHP file contains the HTML code for the corresponding website. Every of these files contains the base code, which includes the header, libraries, and styling, for clarity's sake. For better overview and because the PHP files would have been too long otherwise, the HTML portion and PHP portion of each form were separated and the PHP portion was extracted. The data folder contains these files interacting with the database. The templates folder was organized with base.php and custom.css files. All this allows readability and quick understanding for third-parties that may want to extend the code.

# Testing (5%)

Unit testing was done manually while implementing each unit in the code manually.

Runtime and memory testing were conducted with Xdebug and the VSCode extension “PHP tools”. The following figure shows the results sorted by best to worst:

Graphical user interface

Description automatically generated

Figure 4: Runtime test functions Map

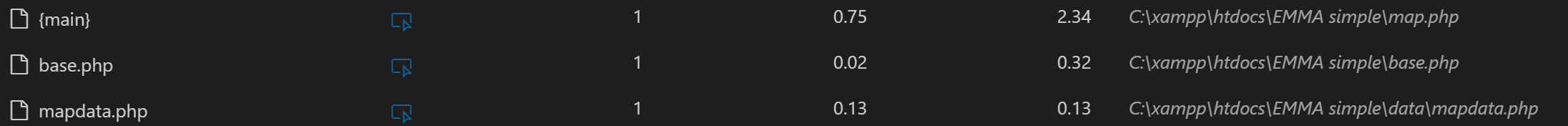


Figure 5: Runtime test files Map

Text

Description automatically generated

Figure 6: Memory test Map

The results for the interactive map show, that it neither uses much load time nor memory time. The profiling also showed no hot paths in the code. The query, as in all the other functions, uses the most load time. This is due to the fact that the function retrieves data from the database.

Text

Description automatically generated

Figure 10: Memory test Search

A picture containing graphical user interface

Description automatically generated

Figure 11: : Runtime test functions Search

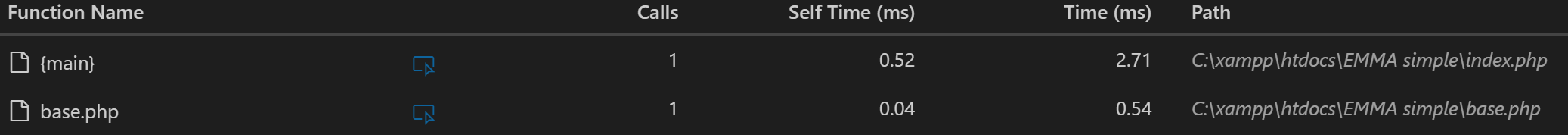


Figure 12: Runtime test files Search

The results for the search function show low memory use and loading time. The hot path here is the query, which is also resembled in figure 11. It is not understandable why the search function has worse results than the map function, as the map is more complex.

Text

Description automatically generated

Figure 13: Memory test Input form

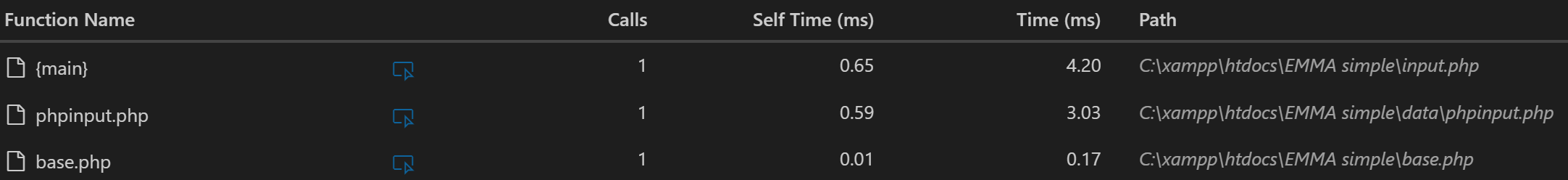


Figure 14: Runtime test files Output form

A picture containing text

Description automatically generated

Figure 15: Runtime test functions Insert form

The input form has high numbers as it updates 41 instances at once to the database. This is also the reason why the update queries (prepare and execute) are the hot paths. Originally, there was also a function to check if the QR Code exists in the database. As the results showed this to be a hot path, the check was moved into the search page, which makes logically more sense as well.

Text

Description automatically generated

Figure 7: Memory test Output file

A screenshot of a computer

Description automatically generated with medium confidence

Figure 8: Runtime test functions Output file

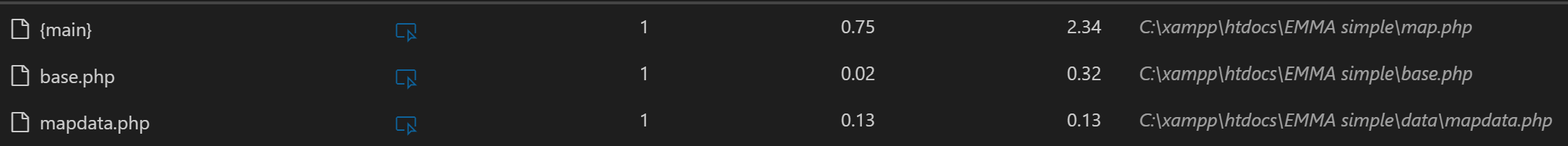


Figure 9: Runtime test files Output file

The output form has the worst results resulting from the purpose of this form. It retrieves all data and uploads data to the database. This makes the load time and memory storage rise.

In all the main files in each test the hot paths where including different files that were outsourced and closing the database.

Overall, the results are satisfying and determine the application as fast and slim.

# Roadmap (5%)

Even though the input form was highly prioritized, the map and the output form were implemented first. This happened due to PHP errors of undefined arrays which could not be resolved in the long input form. So the output form was created first, as correcting and debugging was easier with multiple small forms. The main issue with the forms was that the fields are not required to be filled and having an image to store and retrieve. The storing of the image could not be solved in this project and needs to be solved later on. In the end two functioning, straightforward forms were achieved.

Overall, this application succeeds in being efficient, extensible, and usable.

Usability is attained as a result of the UI elements' simple and intuitive design. As handwriting is eliminated and additional helpful features are added, sighting and recording can be completed more quickly than the conventional method. The website is also easily accessible because to consideration given to the touch feature, which is crucial because paramedics utilize tablets.

A solid folder organization, commenting, and idiomatically sound code all contribute to extensibility.

As demonstrated by the test results, efficiency is attained.

The following phases involve fixing the image storing issue, creating a more organized database, securing website access, and utilizing actual data. Additionally, the issue of whether and when to delete the data must be addressed.

# References

Deutsches Rotes Kreuz. (2004). *Merkblätter: Hinweise zum Gebrauch der Anhängekarte für Verletzte/Kranke mit beiliegendem Formularsatz Suchdienstkarte für Verletzte/Kranke*. Deutsches Rotes Kreuz Kreisverband Alsfeld e.V. Retrieved February 5, 2023, from http://www.drk-alsfeld.de/infothek/drk-merkblaetter-hinweise/anhaengekarte-fuer-verletztekranke.html