Höhere Technische Bundeslehranstalt Kaindorf an der Sulm

Abteilung Informatik

Diplomarbeit

im Rahmen der Reife- und Diplomprüfung

Königskarte

Leon Edlinger
Paul Gigler
Andreas Weissl

5BHIF
2024/2025

Betreuer: Prof. DI Johannes Loibner, BSc
Projektpartner: Prof. DI Robert Müllerferli
Datum: MISSING DATE

Statutory declaration

I declare under oath that I have written the present diploma thesis independently and without outside help, have not used sources and aids other than those indicated and have identified the passages taken from the sources used literally and in terms of content as such.

Ort, Datum	Leon Edlinger
Ort, Datum	Paul Gigler
Ort, Datum	Andreas Weissl

Abstract

Abstract in English

Kurzfassung

Kurzfassung in Deutsch

Thanks

It would not have been possible to carry out this thesis to this extent without the active support of a number of people. We would therefore like to thank everyone who supported us in the implementation of this thesis.
...

Table of Contents

1 Introduction

TODO: Is halt die frage ob ma den anfang einfach so schreiben, war ja eigentlich net ganz so xD

Mobile apps are utilized for virtually all aspects of daily life in the modern world. So after we noticed that there is no application that allows the efficient planning of campaigns like the "Sternsinger-Aktion" we asked ourselves why, and furthermore, how hard it is to create an App with intuitive usability with the main purpose of simplifying the process of managing such a campaign and gaining a general overview of the progress made by the groups.

The app needs to comply with specific criteria we defined in cooperation with Prof. DI Robert Müllerferli. He is the main organizer of the campaign in the parish of Lieboch and helped us to work out the key aspects our project should implement. In the finished product, every user should be able to scan a QR-Code, through which the area of this group gets assigned to the device. These areas must be dynamically adjustable, so an admin can coordinate the workload of each area more efficiently. The areas also need to be clearly visible by an outline which gets drawn through "Border" addresses. These border addresses get calculated by an algorithm implemented by us. It should be visible at a glance if there is an "specification", which can be assigned by admins, set for an address. This should be realized through the use of different icons instead of the default icon. Apart from the app itself, we also implemented a web-portal through which administrators can manage and supervise the campaign.

TODO: vielleicht noch was rein bezüglich der borders und dann unten nurmehr drauf referenzieren?

The research part of this thesis will be dedicated to how components should act and look, so that new users can use this tool without requiring a long "onboarding" phase. It should feel familiar to interact with elements and the borders of what users can and can not do need to be clearly defined. Because our application also needs a reliable data source to guarantee the consistency and accuracy of marked addresses, we researched ways to keep our database up-to-date, without the need of much manual intervention. After defining the project requirements, we noticed that we need to calculate which addresses are border addresses. So we decided to take a look into different algorithms for this task and compare them concerning their efficiency, decide on one of them and implement it.

This thesis contains an in-depth description of our thought and development process, as well as any other steps we took to achieve our goal of a functional mobile application that can be used by volunteers in course of the "Sternsinger-Aktion 2025" taking place in the parish of Lieboch.

1.1 Team

This thesis was created by three Students attending the BHIF20 at the HTBLA Kaindorf Computer Science Department.

TODO: andis bild anpassen

Leon Edlinger

Database, Admin-Panel

Paul Gigler

Deployment, Mobile App

Andreas Weissl

Backend

2 Technologies

Development would not have been possible without making use of many tools, frameworks and environments. In this chapter each tool used in the creation of our software will be described briefly.

2.1 LaTeX

Hier kommt eine Beschreibung zu Latex hin

2.2 Frontend

2.2.1 Dart

Dart is a programming language initially designed for web development, with the goal, of replacing JavaScript, in mind. Today it gets used in a variety of software products, mainly because of the flutter framework. It can be compiled for many platforms and architectures (ARM, x64, RISC-V, JavaScript or WebAssembly) and is loved for its combination of High-Level Features, with practical language features like Garbage collection and optional Type annotation. It was developed by Google and is now an open-source project.

[]

2.2.2 Flutter

Flutter is an Open-Source software development framework. It allows programmers to compile their application for different platforms including Web, macOS, IOS as well as Windows and any type of Linux-based systems, all from one code-base, written in Dart. This allows for more efficient and faster cross-platform development. Another benefit of Google's toolkit are the highly customizable predefined UI components. Developers can mix and match these components however needed which makes them an applicable choice.

We chose flutter mainly for these reasons, but also because of our previous experience with Java to which Dart is quite similar. Through it, we were able to get started quickly, learn what we need along the way. Having a design through the components was also very helpful and saved us some time.

[25] [Dag19]

2.3 Backend

2.3.1 Java Spring

Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua. At vero eos et accusam et justo duo dolores et ea rebum. Stet clita kasd gubergren, no sea takimata sanctus est Lorem ipsum dolor sit amet. Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua. At vero eos et accusam et justo duo dolores et ea rebum. Stet clita kasd gubergren, no sea takimata sanctus est Lorem ipsum dolor sit amet. Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua. At vero eos et accusam et justo duo dolores et ea rebum. Stet clita kasd gubergren, no sea takimata sanctus est Lorem ipsum dolor sit amet.

Duis autem vel eum iriure dolor in hendrerit in vulputate velit esse molestie consequat, vel illum dolore eu feugiat nulla facilisis at vero eros et accumsan et iusto odio dignissim qui blandit praesent luptatum zzril delenit augue duis dolore te feugait nulla facilisi. Lorem ipsum dolor sit amet,

2.3.2 PostgreSQL

Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua. At vero eos et accusam et justo duo dolores et ea rebum. Stet clita kasd gubergren, no sea takimata sanctus est Lorem ipsum dolor sit amet. Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua. At vero eos et accusam et justo duo dolores et ea rebum. Stet clita kasd gubergren, no sea takimata sanctus est Lorem ipsum dolor sit amet. Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua. At vero eos et accusam et justo duo dolores et ea rebum. Stet clita kasd gubergren, no sea takimata sanctus est Lorem ipsum dolor sit amet.

Duis autem vel eum iriure dolor in hendrerit in vulputate velit esse molestie consequat, vel illum dolore eu feugiat nulla facilisis at vero eros et accumsan et iusto odio dignissim qui blandit praesent luptatum zzril delenit augue duis dolore te feugait nulla facilisi. Lorem ipsum dolor sit amet,

2.4 Version Control

2.4.1 Git

Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua. At vero eos et accusam et justo duo dolores et ea rebum. Stet clita kasd gubergren, no sea takimata sanctus est Lorem ipsum dolor sit amet. Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua. At vero eos et accusam et justo duo dolores et ea rebum. Stet clita kasd gubergren, no sea takimata sanctus est Lorem ipsum dolor sit amet. Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua. At vero eos et accusam et justo duo dolores et ea rebum. Stet clita kasd gubergren, no sea takimata sanctus est Lorem ipsum dolor sit amet.

Duis autem vel eum iriure dolor in hendrerit in vulputate velit esse molestie consequat, vel illum dolore eu feugiat nulla facilisis at vero eros et accumsan et iusto odio dignissim qui blandit praesent luptatum zzril delenit augue duis dolore te feugait nulla facilisi. Lorem ipsum dolor sit amet,

2.4.2 GitHub

Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua. At vero eos et accusam et justo duo dolores et ea rebum. Stet clita kasd gubergren, no sea takimata sanctus est Lorem ipsum dolor sit amet. Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua. At vero eos et accusam et justo duo dolores et ea rebum. Stet clita kasd gubergren, no sea takimata sanctus est Lorem ipsum dolor sit amet. Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua. At vero eos et accusam et justo duo dolores et ea rebum. Stet clita kasd gubergren, no sea takimata sanctus est Lorem ipsum dolor sit amet.

Duis autem vel eum iriure dolor in hendrerit in vulputate velit esse molestie consequat, vel illum dolore eu feugiat nulla facilisis at vero eros et accumsan et iusto odio dignissim qui blandit praesent luptatum zzril delenit augue duis dolore te feugait nulla facilisi. Lorem ipsum dolor sit amet,

2.5 Map Data

2.5.1 OpenStreetMap

Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua. At vero eos et accusam et justo duo dolores et ea rebum. Stet clita kasd gubergren, no sea takimata sanctus est Lorem ipsum dolor sit amet. Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua. At vero eos et accusam et justo duo dolores et ea rebum. Stet clita kasd gubergren, no sea takimata sanctus est Lorem ipsum dolor sit amet. Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua. At vero eos et accusam et justo duo dolores et ea rebum. Stet clita kasd gubergren, no sea takimata sanctus est Lorem ipsum dolor sit amet.

Duis autem vel eum iriure dolor in hendrerit in vulputate velit esse molestie consequat, vel illum dolore eu feugiat nulla facilisis at vero eros et accumsan et iusto odio dignissim qui blandit praesent luptatum zzril delenit augue duis dolore te feugait nulla facilisi. Lorem ipsum dolor sit amet,

2.5.2 Graphhopper

Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua. At vero eos et accusam et justo duo dolores et ea rebum. Stet clita kasd gubergren, no sea takimata sanctus est Lorem ipsum dolor sit amet. Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua. At vero eos et accusam et justo duo dolores et ea rebum. Stet clita kasd gubergren, no sea takimata sanctus est Lorem ipsum dolor sit amet. Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua. At vero eos et accusam et justo duo dolores et ea rebum. Stet clita kasd gubergren, no sea takimata sanctus est Lorem ipsum dolor sit amet.

Duis autem vel eum iriure dolor in hendrerit in vulputate velit esse molestie consequat, vel illum dolore eu feugiat nulla facilisis at vero eros et accumsan et iusto odio dignissim qui blandit praesent luptatum zzril delenit augue duis dolore te feugait nulla facilisi. Lorem ipsum dolor sit amet,

2.6 Development Tools

2.6.1 VS Code

Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua. At vero eos et accusam et justo duo dolores et ea rebum. Stet clita kasd gubergren, no sea takimata sanctus est Lorem ipsum dolor sit amet.

2.6.2 IntelliJ

Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua. At vero eos et accusam et justo duo dolores et ea rebum. Stet clita kasd gubergren, no sea takimata sanctus est Lorem ipsum dolor sit amet.

2.6.3 Android Studio

Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua. At vero eos et accusam et justo duo dolores et ea rebum. Stet clita kasd gubergren, no sea takimata sanctus est Lorem ipsum dolor sit amet.

2.6.4 Postman

Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua. At vero eos et accusam et justo duo dolores et ea rebum. Stet clita kasd gubergren, no sea takimata sanctus est Lorem ipsum dolor sit amet.

2.6.5 Figma

Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua. At vero eos et accusam et justo duo dolores et ea rebum. Stet clita kasd gubergren, no sea takimata sanctus est Lorem ipsum dolor sit amet.

2.7 Deployment

2.7.1 Docker

2.7.2 Uberspace

2.7.3 Webmin

3 Research Questions

3.1 Leon Edlinger

3.2 Paul Gigler

3.3 Andreas Weissl

4 Spring Framework

The backend leverages the **Spring Framework**, a comprehensive framework for enterprise Java development. This section explores its key components and advantages.

4.1 Spring Boot

Spring Boot simplifies configuration and deployment with embedded servers and opinionated setups. This reduces boilerplate code and accelerates development.

4.2 Spring Data JPA

Spring Data JPA provides abstractions for database interactions, streamlining CRUD operations and custom query creation.

4.3 Lombok

Lombok reduces boilerplate code by generating getters, setters, and other methods at compile time, improving code readability and maintainability.

4.4 Advantages

Using Spring enhances productivity, reduces setup complexity, and ensures scalability, making it ideal for this project.

5 Area Borders

The area borders feature addresses the research question by implementing computational geometry algorithms for precise geographical boundary calculations.

5.1 Purpose of Area Borders in the App

Accurate area borders are essential for defining regions based on user input, supporting the app's mapping functionality.

5.2 Overview of the Convex Hull Algorithm

The convex hull algorithm identifies the smallest convex polygon enclosing a set of points, making it a suitable choice for this project.

5.3 Use Cases of the Convex Hull in Industry

Applications of convex hulls in mapping, computer graphics, and robotics highlight their importance in solving real-world problems.

5.4 Alternate Methods for Area Border Calculation

Alternative methods like Voronoi diagrams and alpha shapes were considered but found less suitable due to complexity or computational demands.

5.5 Rationale for Choosing the Convex Hull Method

The convex hull algorithm offers a balance of simplicity, efficiency, and accuracy, aligning with the project's requirements.

5.6 Integration of the Algorithm into the Backend

The algorithm is implemented in the service layer, ensuring smooth integration with other backend components.

5.7 Challenges and Adjustments

Challenges included handling edge cases like collinear points, which were resolved through specific algorithm adjustments.

6 Structure of the Backend

The backend follows a layered architecture to promote separation of concerns, scalability, and maintainability. This section outlines the roles of each layer.

6.1 Controller Layer

The controller layer acts as the interface for incoming HTTP requests, delegating them to appropriate service methods.

6.2 Service Layer

The service layer contains business logic, validating data and coordinating interactions between controllers and repositories.

6.3 Repository Layer

Repositories abstract database operations, allowing the backend to interact with the database without explicit SQL queries.

6.4 Persistence Layer (Entity Classes)

Entity classes define the data model and its mapping to the relational database, ensuring a consistent schema.

6.5 Applied Design Principles (DTOs)

Data Transfer Objects (DTOs) enhance encapsulation and optimize data transfer between layers and external clients.

7 Defining usability

Since my research question "How can user-experience principals add to an intuitive map displayment for nonprofit activities in which people of different technical know-how levels collaborate?" is all about usability, I want to introduce you to its basic concepts and challenges but also provide some examples on how usability can impact a software's revenue and perception.

Usability is a critical aspect of software and interface design, ensuring that users can efficiently and effectively interact with a product or system. Its job is to provide clear feedback and "experiences" to the user, so interactions between software and human feel smooth and straight forward. Because each human being is different in its emotional experiences, it is difficult to design a kind of "one size fits all" solution. Due to this circumstance, many studies and experiments were conducted. [Nie24]

7.1 Why it is important

Usability ensures that users can accomplish their goals with minimal frustration and maximum efficiency. With the increasing reliance on digital tools, usability plays a key role, not only, in shaping user experiences but also accessibility of software for diverse user groups. A well-designed and thought-out usability concept can go a long way from refining a once tedious and complicated to use product, to one that can be operated even by non-familiar users or disabled people. This plays a big part in the inclusion of all age and knowledge groups as well as the general market share through mass adoption because of the easiness.

7.2 Components of Usability

According to Jakob Nielsen, usability consists of five core components. To achieve the best possible usability, each of factors must be taken into account and be improved to its maximum.

- Learnability

  How **easy** it is to accomplish basic tasks the first time

- Efficiency

  How **quickly** task can be accomplished after an initial learning period

- Memorability

  How **memorable** actions are to users so, after an extended period of not using a software

- Error handling

  How **many** errors users make while using the design and how **sever** they are

- Satisfaction

  How **pleasant** the overall experience of using the product is

[Nie24]

Now that we are aware of these key points, what measures can we take to reach the goal of great usability? According to Nasrullah Hamidli, human-computer-interaction relies on consistency, visibility, feedback, and simplicity. Consistency ensures users do not need to learn new interactions for each task. For example, buttons should look alike and be in a similar location. This makes for a more natural navigation across the product and an overall familiar feel. Simplicity connects directly to this. Its goal is to minimize clutter and make user interfaces easy to understand and provide one, clear way to accomplish a task, not many possible, but complicated and unintuitive ways. It also aims to reduce distractions. Visibility allows users to clearly understand their options at any given moment, this is most often achieved through visual cues, like, grayed out buttons. This goes hand in hand with the feedback aspect, which provides immediate confirmation of actions. Loading indicators, color-changes and alike get used most often.

Another important part of designing a good UI are typography and colors. These can act as parameters for the attention and emotions of users, as well as establish visual hierarchies, which intern, contribute again to a simpler to navigate interface. [Ham23]

7.3 Fundamental concepts

In this section we will look further into these basic concepts and examine what designers can concretely do to improve usability. To make it more descriptive I will provide two examples each, one that represents an interface that doesn't follow usability guidelines, and one that improves a specific aspect of it.

Visual Hierarchy

Starting of with Visual hierarchy. It is the most basic principle in UI/UX design and dictates, how users perceive and navigate content. A strong visual hierarchy contributes to simplicity and ensures that important elements are more visually prominent, guiding users toward essential actions and information. Factors include the text size, weight and spacing.

- In this example, visitors of our website are not able to clearly and promptly differentiate between headings and actual text. Because the items in the navigation bar have a different font-weight, it is distinguishable but not as clear as it could be.

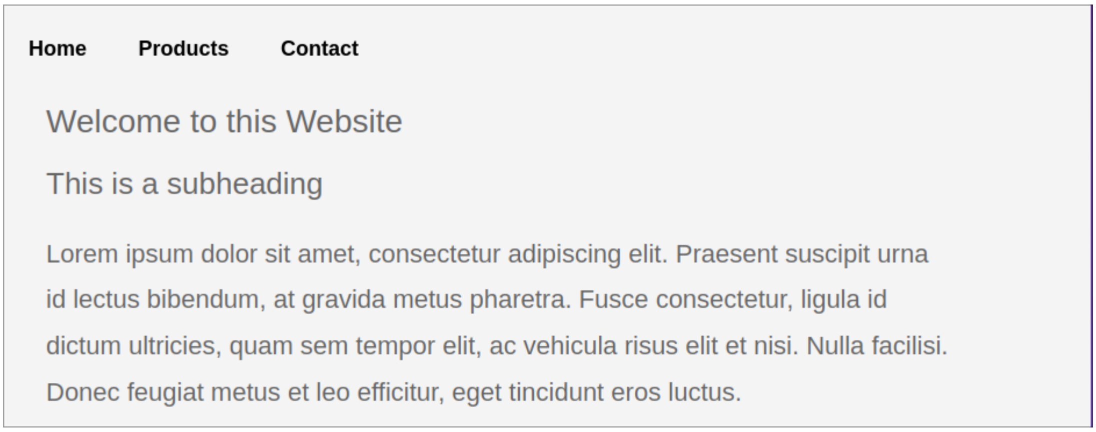

Abb. 1: Example of website with weak visual hierarchy

- If the font size and text color gets just tuned a little, it becomes suddenly much easier to distinguish primary content from secondary information. Headings are bold and larger, while body text is appropriately sized for readability. Proper spacing and alignment create a structured and pleasant reading experience.

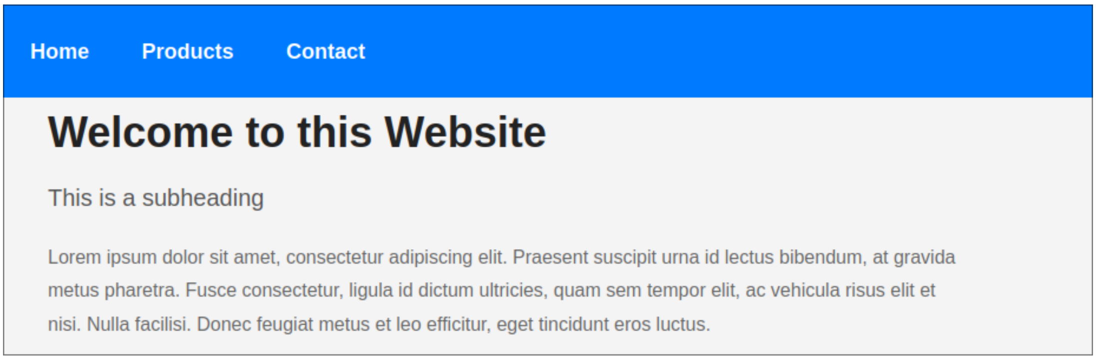

Abb. 2: Example of website with strong visual hierarchy

Source: FMD University - UX Visual Hierarchy

Color Theory & Contrast

Color selection plays a critical role in usability and aesthetics. High contrast improves readability, while poor color choices can make content inaccessible, especially for users with visual impairments. To help designer with their color choices, the color wheel was invented.

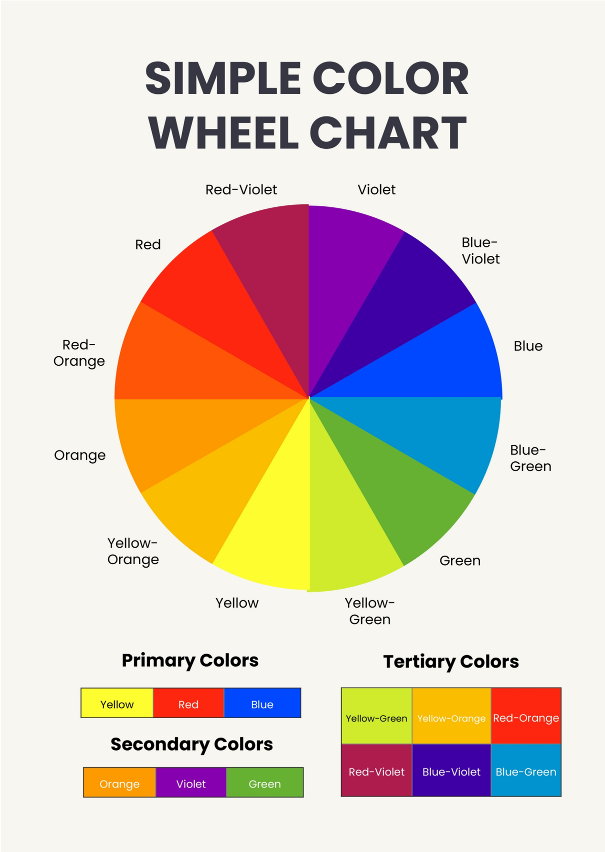

Abb. 3: Color wheel with primary, secondary and tertiary colors sorted (Source: `https://www.template.net/editable/114132/simple-color-wheel-chart`)

Relationship Type	Description	Use case
Complementary	Colors opposite each other	Creating bold contrast
Analogous	Colors next to each other	Harmonious, natural looks
Triadic	Three evenly spaced colors	Balanced, vibrant designs
Split Complementary	One color + two adjacent to its complement	Softer contrast than complementary

Table 1: Key color combinations with effects

TODO: Quelle für den table einfügen

Bad example: A bad design example features low-contrast elements, such as light gray text on a white background, making it difficult to read, especially for users with low vision. But also too many, or wrongly placed eye-catching colors can confuse and overwhelm the user.

Good example: A good design example ensures high contrast between text and background. Colors are chosen deliberately to create visual harmony while maintaining functionality and accessibility. Well-chosen colors draw the attention to important elements. But this can also be used in reverse, for example, a button to delete something important may be colored red, to indicate a fatal action, but could also be gray, so the user must consciously decide to press it. This acts as a safety measure.

Source: Atmos - Color Theory in UI Design

Consistent Layout & Interaction Patterns
Consistency in design enhances user experience by ensuring that UI elements and interactions are predictable across different states of the application.

Bad example: A inconsistent navigation bar where the position and style change across different screens, leading to confusion and frustration. But also purely aesthetic inconsistency impacts the overall perception of the application.

Good example: A good design example maintains a uniform layout with clearly defined navigation menus, button styles, and form elements. This consistency improves usability and allows users to develop mental models of how the system operates.

Source: Maze - UI Design Principles

Button Placement & Behavior
Buttons are a critical and one of the most common interactive elements in UI design. Their placement, size and responsiveness affect usability and efficiency.

Bad example: A bad design example places primary actions (e.g., "Submit") in less prominent or unintuitive areas, while secondary actions (e.g., "Cancel") are emphasized, leading to potential user errors. Moving forward has been related to an action on the right side of the screen since the very first user interfaces, so, putting a "Forward" button on the left side of the screen would be very unintuitive and feel weird.

Good example: A good design example ensures that primary actions are clearly distinguishable with appropriate sizing and positioning. Visual and tactile feedback, such as hover effects or color changes on click, confirm user interactions.

Source: Dribbble - Button UI Design

Feedback Providing immediate and clear feedback enhances user confidence and prevents frustration.

Bad example: A bad design example lacks feedback for user actions, such as clicking a button without visual confirmation, leaving users uncertain if their action was registered.

Good example: A good design example includes real-time feedback mechanisms like loading indicators, success messages, and animations to confirm user interactions.

Source: Dribbble - Feedback UI Design

Form Design Best Practices
Forms are a crucial aspect of digital interfaces. Their usability impacts user satisfaction and data collection efficiency.

Bad example: A bad design example relies solely on placeholder text without persistent labels, making it difficult for users to recall input requirements once they start typing.

Good example: A good design example includes clearly visible labels, real-time validation messages, and appropriately spaced input fields, reducing user errors and improving accessibility.
Source: Medium - Form Design Best Practices

7.4 Challenges in designing for a broad user spectrum

Designing for a diverse user base requires the addressing of varying levels of experience, prior knowledge, cognitive abilities, and accessibility needs. Failure to account for these differences can lead to usability issues, preventing certain groups from effectively using a system. Designers must implement features such as adjustable text sizes, screen reader compatibility, and intuitive navigation to ensure accessibility for all users.

TODO: wenn zu wenig inhalt dann könnte man auch noch was schreiben wieso des so is und auf welchen psychologischen dingen des basiert Interfaces should always be tailored to the needs and expectations of the end-user. This leads to challenges when the user-group is not clearly defined or consists of people with widely different backgrounds. For example elderly people often need further guidance when interacting with digital solutions than members of younger generations. This leads back to the core components of usability-design, interfaces need to be simple and unmistakable in their functionality. Failing to provide these core concepts will sooner or later result in a frustrated and shrinking user base. [Ham23]

8 Usability in context of maps

In this section we will further inspect the usability of mapping solutions like Google Maps. We will identify some flaws of Googles design choices and how they could influence specific user groups.

8.1 Basic Analysis of the Google Maps Interface

Google Maps is one of the most widely used mapping and navigation applications globally. Its feature set includes real-time traffic updates, route planning, and location discovery. Generally speaking, it is quite difficult to design a simultaneously user-friendly and functional mapping application. Maps get overloaded and confusing quiet easily. They are bloated with information like street names, house numbers, borders and geographical features like rivers or lakes. Due to this fact, maps are not easy to design according to the principles of usability. But Google developed a very good and intuitive concept on which we now will take a look.

The interface of Google Maps consists of a map, search bar and a menu for additional settings. The search function is prominently displayed as it is the most common tool used. This is a good use of *visual hierarchy* as the initial focus when opening the app immediately gets drawn to the bigger hint text in the search bar on top. Below it, there is a list of buttons, so users can quickly search for local places that match a specific category, like restaurants, cafés or gas stations. Notably there are no buttons for movement actions such as zooming or panning, all this is controlled through swipe and pinch gestures directly on the map.

To start navigating to a specific area, you can search for the street name and house number, or the name of a company or other details. Maps gives you recommendations and tries to provide auto-suggestions for your target. When the user selects a destination, the navigation can be started through a big blue button. This is an example of applied *color theory*. The route gets marked by an again bright blue line, which creates a good contrast to the other colors used on the map and captures the attention of the user. This line also has multiple purposes other than displaying the route, for example, if a part is orange, that means the traffic at this point is beginning to jam. If then there is a full stop traffic jam, the line turns red at this section. Also, icons for speed cameras or accidents that other users reported get displayed along it.

One of the core strengths of Google Maps is its interactive and responsive design. Users can zoom in and out using intuitive pinch gestures on mobile devices or scroll actions on desktops. The transition between zoom levels is smooth, preserving context and avoiding disorientation through to big scaling steps. Additionally, the map changes, depending on the zoom level. Street names and buildings get displayed only, if the user has zoomed in enough. The same happens with markers for businesses and other map data. Through this concept, Google ensures that users do not get overwhelmed.

8.2 Identifying Flaws in Google’s Design

While Maps is a well-polished product, it is not perfect. One significant flaw is the cognitive overload caused by excessive information. The inclusion of business listings, other suggested routes, live traffic data, and user-generated content can make it difficult for users to focus on their primary navigation tasks.

8.3 How Could Specific User Groups Struggle with This Design

Google Maps caters to a broad spectrum of users, but its design can pose difficulties for certain demographics:

Elderly Users: Many elderly individuals may find the interface overwhelming due to small text sizes, densely packed information, and complex menus. Their unfamiliarity with modern digital navigation tools may lead to confusion, especially when trying to search for locations or adjust route preferences. A lack of prominent, simplified navigation options tailored to this group amplifies the issue.
Users with Low Digital Literacy: People who are not well-versed in digital technology could struggle with Google Maps’ multitude of features. They may have difficulty understanding icons, switching between different map modes, or using advanced functionalities like saved locations and street view. A more guided and *simplified* mode could enhance their experience.
Users with Disabilities: Visually impaired users may struggle with *insufficient contrast*, small icons, and the *lack of tactile feedback*. While screen readers can assist, Google Maps does not always provide clear, structured data for these tools. Additionally, users with motor impairments may find it hard to navigate menus and interact with small buttons, particularly on touch screens.

By addressing these usability concerns, Google Maps could enhance its interface to be more intuitive and accessible for diverse user groups.

9 Adaptive algorithms and real-time data integration

9.1 Theoretical Framework

9.1.1 Traditional Methods for Address Database Management

9.1.2 Adaptive Algorithms: Concepts and Applications

9.1.3 Real-Time Data Integration Frameworks

9.2 Technical Framework

9.2.1 Data Sources

9.2.1.1 GPS Data

9.2.1.2 External APIs

9.2.1.3 User Inputs

9.2.2 Adaptive Algorithms

9.2.2.1 Fuzzy Matching

9.2.2.2 Machine Learning Model

9.2.2.3 Rule-Based Filters

9.2.2.4 Dynamic Duplicate Resolution

9.2.2.5 Real-Time Address Normalization

9.2.3 Evaluation Metrics

9.2.3.1 Accuracy

9.2.3.2 Latency

10 Traditional Methods for Address Database Management

11 Adaptive Algorithms: Concepts and Applications

12 Real-Time Data Integration Frameworks

13 Implementation of the Backend

The backend implementation combines theoretical concepts with practical solutions to ensure functionality and scalability.

13.1 Config of Spring Boot (application.properties)

The `application.properties` file configures essential settings, including database connections, logging, and server parameters.

13.2 Entity Classes (Structure/Purpose)

Entity classes define the application's data model, using annotations to map fields to database tables.

13.3 JPA-Repositories (DB Access and CRUD Operations)

Repositories simplify database access by providing methods for CRUD operations and enabling custom queries.

13.4 Service Classes

Service classes encapsulate business logic, coordinating data flow between controllers and repositories.

13.5 Rest Controller (API Endpoints and their Functions)

REST controllers define API endpoints, processing requests and returning responses to ensure seamless interaction with the frontend.

14 GraphHopper Setup

14.1 Why use GraphHopper?

14.2 Configuration

14.3 Local hosting

15 Working out the Wireframes

15.1 Map View

15.2 List View

15.3 Possible improvements for future versions

16 Functional implementation behind the application

16.1 Address-Provider

16.2 HTTP-Requests

16.3 Implementation of the Flutter Map Component

17 The app in use

17.1 Introducing new users

17.2 The app in operation

17.3 User Feedback

18 Final Thoughts

18.1 Leon Edlinger

18.2 Paul Gigler

18.3 Andreas Weissl

19 Meetings

Protokolle der Meetings, vielleicht auch ein zeitplan wann immer und wie lang

20 Working Hours

Arbeitspaket-Nr.	Beschreibung	Dauer
1	Einführung und Einarbeitung	8 h
2	Grundkonzept erstellen	8 h
3	Struktur der App festlegen	6 h
5	Wifi-Socket in App implementieren	39 h
6	Write-Funktionalität in App implementieren	14 h
7	Read-Funktionalität in App implementieren	19 h
8	Trim-Funktionalität in App implementieren	10 h
9	Konfigurationsmöglichkeiten für Flug in App implementieren	16 h
10	Höhenregelung-Funktionalität in App implementieren	14 h
12	Graphische Darstellung der Flugdaten	18 h
14	App testen und debuggen	19 h
26	Gesamtkonzept testen und debuggen	16 h
Summe		**187 h**

Table 2: Arbeitszeitnachweis

21 Source code directory

Source Code directory, kein plan was des is

22 List of figures

23 List of tables

24 Bibliography

[] *Flutter for Beginners*. URL: `https://books.google.at/books?hl=de&lr=&id=pF6vDwAAQBAJ&oi=fnd&pg=PP1&dq=benefits+dart+language&ots=dZJWUGVs4x&sig=a196WqhXmQzuy23cmcKpEpIqn_k&redir_esc=y#v=onepage&q=benefits%20dart%20language&f=false`.

[25] *flutter/README.md at master · flutter/flutter*. [Online; accessed 23. Jan. 2025]. Jan. 2025. URL: `https://github.com/flutter/flutter/blob/master/README.md`.

[Dag19] Lukas Dagne. "Flutter for cross-platform App and SDK development". In: (2019).

[Ham23] Nasrullah Hamidli. "Introduction to UI/UX design: key concepts and principles". In: *Academia. URL: https://www.academia.edu/98036432/Introduction_to_UI_UX_Design_Key_Concepts_and_Principles [accessed 2024-04-27]* (2023).

[Nie24] Jakob Nielsen. "Usability 101: Introduction to Usability". In: *Nielsen Norman Group* (Jan. 2024). URL: `https://www.nngroup.com/articles/usability-101-introduction-to-usability`.

25 Abbreviation

ADC	Analog Digital Converter
API	Application Programming Interface
BLE	Bluetooth Low Energy
CPU	Central Processing Unit
DAC	Digital Analog Converter
DAVE	Digital Application Virtual Engineer
DSP	Digital Signal Processor
FPU	Floating Point Unit
FPV	First Person View, First Pilot View
GPIO	General Purpose Input/Output
GPS	Global Positioning System
GUI	Graphical User Interface
HDMI	High Definition Multimedia Interface
I^2C	Inter-Integrated Circuit
IDE	Integrated Development Environment
IP	Internet Protocol
RPI	Raspberry Pi
SD	Secure Digital
SPI	Serial Peripheral Interface
USB	Universal Serial Bus
TCP	Transmission Control Protocol
UART	Universal Asynchronous Receiver Transmitter
WLAN	Wireless Local Area Network
WPA	WiFi Protected Access
XML	Extensible Markup Language