universität wien

MASTERARBEIT | MASTER'S THESIS

Titel | Title

CARPI - Utilizing a Chatbot to improve Accessibility for Route Planning Interfaces

verfasst von | submitted by

Henry Tom Eretge B.Sc.

angestrebter akademischer Grad | in partial fulfilment of the requirements for the degree of

Master of Science (MSc)

Wien | Vienna, 2025

Studienkennzahl lt. Studienblatt | Degree programme code as it appears on the student record sheet:	UA 066 977
Studienrichtung lt. Studienblatt | Degree programme as it appears on the student record sheet:	Masterstudium Business Analytics
Betreut von | Supervisor:	Univ.-Prof. Torsten Möller PhD

I. Abstract (English)

Current literature shows a major gap in research regarding the usability of route-planning platforms. This thesis investigates the impacts of replacing traditional input fields of a route-planning application with a chatbot interface.

Applying the user-centered design approach, a new route-planning tool fitted to users' expectations is developed. Users are interviewed to gather requirements. Based on these requirements, a set of prototypes is created. Potential users are asked for their favorite prototype parts. These parts are implemented as an application, including the chatbot. The application is evaluated and compared to two other route-planning tools.

We compared our newly developed tool to Google Maps and the route planning tool of the Austrian railway company ÖBB. The overall usability of the newly created tool is comparable with existing solutions. The time needed to complete route planning is longer. However, integrating demands and preferences in the new application is improved compared to traditional route planning applications, such as Google Maps. The accessibility is also comparable to existing solutions. For users with less experience with route planning, simplicity, integration of demands and preferences, and accessibility to laypeople are rated higher than the compared solutions. Most users positively noted having only one input field instead of filling in the information for the searched route in different fields.

The application being able to compete with solutions developed by large companies is a success and shows the possibilities of simplifying user interfaces by adding a chatbot. More research regarding conversational interfaces is needed. Performance improvements, additional features, and more user tests to increase the significance and validity are the possibilities for continuing this study.

II. Abstract (Deutsch)

In der aktuellen Literatur zeigt sich eine Lücke zur Forschung der Benutzbarkeit von Routenplanungsanwendungen. Diese Masterarbeit untersucht die Auswirkungen des Ersetzens von konventionelle Eingabefeldern bei Routenplanungsanwendungen durch eine Chatbot Oberfläche. Die Anwendung des aufstrebenden Themengebiets der Chatbots, wie z.B. ChatGPT, auf ein im Alltag viel benutztes System eröffnet viele neue Möglichkeiten.

Mit dem Prinzip des benutzerzentrierten Designs wird eine neue Routenplanungsanwendung entwickelt. Potenzielle Nutzer werden interviewt, um Anforderungen zu identifizieren. Aus diesen Anforderungen werden eine Anzahl von Prototypen entwickelt, die von weiteren potenziellen Nutzern bewertet werden. Schließlich wird eine Anwendung inklusive des Chatbots entwickelt, die ausgewertet und mit zwei etablierten Routenplanungsanwendungen verglichen wird.

Die Benutzerfreundlichkeit der neuen Anwendung ist vergleichbar mit den vorhandenen Systemen. Die benötigte Dauer für Routenplanung ist bei dem System mit dem integrierten Chatbot länger. Ein Aspekt der Benutzerfreundlichkeit ist positiv auffällig, die Integration von Anforderungen und Präferenzen. Die Zugänglichkeit ist ebenfalls vergleichbar mit den etablierten Anwendungen. Für Nutzer mit weniger Routenplanungserfahrung fallen die Bewertungen von der Einfachheit, der Integration von Anforderungen und Präferenzen und der Zugänglichkeit von Laien dagegen überwiegend positiver aus. Die Verminderung auf ein einzelnes Eingabefeld ist von den meisten Nutzer positiv angemerkt worden.

Insgesamt kann die Anwendung mit den von großen Konzernen entwickelten Systemen konkurrieren. Dieser Erfolg zeigt die Möglichkeiten Nutzeroberflächen durch die Ergänzung eines Chatbots zu vereinfachen und weist die Richtung für weitere Forschung zu Benutzerschnittstellen mit Chatbots. Mehr Leistung, zusätzliche Funktionalitäten und mehr Nutzertests, um die Aussagekraft der Ergebnisse zu erhöhen sind beispielhafte Möglichkeiten diese Studie fortzusetzen.

III. Table of Contents

IV. List of Figures

V. List of Tables

VI. Index of abbreviations

API ..... Application Programming Interface
AWS EC2 ..... Amazon Web Services Elastic Compute Cloud
BBB ..... BusBahnBim
DB-N ..... Deutsche Bahn Navigator
GM ..... Google Maps
GTFS ..... General Transit Feed Specification
JSON ..... JavaScript Object Notation
LLM ..... Large Language Model
OSM ..... OpenStreetMaps
OTP2 ..... OpenTripPlaner 2
UCD ..... User-Centered Design
UI ..... User Interface

1 Motivation

Route planning is a part of most people's day-to-day life: going to work, finding the coffee place your friend recommended, or going to the next vacation destination. Before modern smartphones, the timetables of buses and trains were part of every household, and the vacation was booked with a travel agency. With the availability of the Internet, different solutions have become available, e.g., timetables are always accessible, including live information. With increasing possibilities, the complexity of route planning interfaces has also increased. Routes differ, e.g., by the amount of time one has to make a connection, impacting one's chance of arriving at the destination on time. Additional transportation modes like car- and bike-sharing increase the possible routes even more.

Modern interfaces like the well-known Google Maps try to tackle this problem by integrating most forms of transportation. But what happens if my preference is not the fastest route or I am too scared to use the clumsy e-scooters? Adjusting preferences until a route fitting for personal desires is found can be difficult, and there is no "one size fits all" in public transportation systems [1]. While some people will take the car instead, and others will optimize their route on the go, there might be a better third option. Introducing a conversational interface for route planning where all preferences, demands, and constraints can be said or typed might reduce the complexity while improving accessibility.

This thesis develops a route-planning application with a conversational interface and analyzes its advantages and disadvantages.

1.1. Research Gap

In our section on "Previous Work", relevant work for route-planning accessibility and chatbots for information retrieval is presented. **However, the current literature does not provide solutions to enhance usability and accessibility for route planning by integrating a chatbot.** A comparable solution for travel planning was found only with outdated nongenerative chatbots and just for monomodal traveling [2]. Creating a new open-domain[1] and generative[2] chatbot expands the possibilities (e.g., follow-up questions and more flexible decision spaces) while contributing to useability and accessibility.

1.2. Contributions of this Thesis

This thesis contributes to the current research by gathering insights into how adding a chatbot to a route-planning application impacts accessibility and usability. Furthermore, a list of requirements for route-planning applications is developed. Developing a functional application where the most important requirements are implemented is a practical contribution to this thesis. This application gathers user experiences and compares them with existing solutions. An analysis of these user experiences, also dependent on demographics, shows the advantages and disadvantages of the new route-planning application with an integrated chatbot.

[1] "open-domain chatbots can engage in conversation in any topic" [3]

[2] new machine learning models for the generation of information in the form of, e.g., text [4]

1.3. Research Question

The described research gap leads to the following research question:

Can a **generative chatbot simplify and shorten** the process of finding travel itineraries with **equally good or better results than existing systems,** given a layperson's **demands and preferences**?

To answer this question, four steps are taken:

1. What requirements should the interface fulfill to make it accessible to a broad range of users?
2. What is an effective user interface enabling the requirements?
3. Is it technically reasonable to integrate a chatbot in a route-planning application while considering the users' requirements?
4. What are the advantages and disadvantages of adding a chatbot to itinerary planning software?

To conclude, this thesis aims to combine the existing technologies and research of chatbots, as well as travel itinerary planning and visualization, to create a tool that is more usable and accessible for laypersons than existing solutions.

2 Structure and Methodology of the Thesis

This thesis will first describe previous works to introduce concepts and topics applied. Afterward, the thesis is structured typically for projects researching new interaction and visualization techniques.

This thesis will follow the idea of user-centered design (UCD) to achieve usability based on the user's needs. "UCD is herein considered, in a broad sense, the practice of the following principles, the active involvement of users for a clear understanding of user and task requirements, iterative design and evaluation, and a multi-disciplinary approach." [5] To achieve this involvement of users while creating a new application, the stages of the core phase of the design study methodology are applied with the following definition of a design study: "A design study is a project in which visualization researchers analyze a specific real-world problem faced by domain experts, design a visualization system that supports solving this problem, validate the design, and reflect about lessons learned to refine visualization design guidelines." [6] This definition fits the goal of this thesis: find a visualization solution for an interface to project onto an existing optimization algorithm for route-planning. Technical solutions supplement the visualization solution.

The core phase of the design study methodology consists of four stages [6]:

1. Discover → Requirements Analysis
2. Design → Prototyping a chatbot application
3. Implement → Application Development
4. Deploy → Evaluation

A description of the stages follows:

1. The phase starts with '**discovery**,' what would be called requirements analysis in software engineering [7]. Problems identified with current design solutions and requirements should be found from several sources, such as user interviews or domain literature. The potential incompleteness,

especially in interviews, should be considered during the whole process as users typically cannot express all their needs [8]. The first interviews are conducted. Discussing design solutions rather than requirements with the interviewees should be prevented and is seen as one of the potential pitfalls. However, not only the problematic parts of the workflow should be considered. The working parts should also be identified, written down, adapted, and reused. This also complements the potential incompleteness of the interviews. An abstraction of the mentioned requirements will follow afterward. The more detailed methodology for this step is explained in Chapter 4.1.

2. The following **design** step is based on the previous discovery step. The shared understanding with the experts can now be translated into designing visualization solutions. Visualization solutions include data abstractions, visual encodings, interaction mechanisms, etc. Several solutions can be designed and considered here. Narrowing down the considerations to a few solutions is the next step. The narrowed-down proposals should again be discussed with domain experts to verify the solutions. Here, possible users are approached again to get input to the solutions. For this step, prototyping and its methodology are discussed in more detail in Chapter 5.
3. The **implementation** is the next step. Agreed-on solutions can be realized immediately. Feedback loops and testing during the process help to gather problems not discovered in the design process. A focus on usability should always be part of this step to avoid the need for training to use the application.
4. The last step of the core phase is '**Deploy'**. Deploying itself is relatively straightforward, but gathering feedback is also considered part of this step. Validation can, for example, be achieved by experts and other users doing tasks faster or more correctly. A case study with actual data and real users is considered optimal, which happens as part of a comparison to existing systems in this thesis, again with potential users. This step is further discussed in Chapter 7.

The information systems community introduced similar frameworks, also called design research. The steps of identifying a problem and its solution before developing and finally finishing with an evaluation and a conclusion are comparable [9].

3 Previous Works

In the following chapter, a range of previous works regarding route planning accessibility and chatbots for information retrieval is presented to show the current level and the research gaps.

3.1. Route Planning Accessibility

To increase route planning accessibility, research in the direction of user-centered design for public transportation systems has been done. Not only have layouts, colors, and buttons been adjusted, but the user has also accompanied the process of the interface's development [10]. Needs to have all the information gathered in one place have been identified, especially regarding timetable updates, fare updates, a navigation system, trip personalization and delay information [11]. This needed information overlaps with what personnel of transportation agencies are often asked [12]. A needed confirmation for already gathered information is also mentioned in this paper.

Furthermore, an easy-to-understand search mask is requested by users as people feel that "asking a human being is the most comfortable way to gather the desired data" [12]. Reducing the complexity and

increasing trust in the results using modern technologies is one of the solutions [1]. These modern solutions need to be packed into some kind of interface. User interfaces (UI) are how humans interact with most computer systems. The better the UI, the easier it is to use the system. The UI is the most significant factor in improving a system's usability [13]. Unfortunately, research on non-driving transportation systems is lacking. The review paper, identifying 1179 papers in the research direction of human-centered design and public transportation by Lengkong et al. (2024) even speaks of "a significant research gap […] concerning the application of the HCD principle in public transportation services for a diverse target group" [14]. Other papers cover topics such as accessibility to impaired people (e.g., [15], [16]), autonomous vehicles in transportation (e.g., [17], [18]), or public kiosks design of transit agencies (e.g., [19], [20]).

3.2. Chatbots for Information Retrieval

Public transportation systems are the "most ubiquitous and complex large-scale systems found in modern society" [1]. New ways are necessary to navigate these systems, especially the databases they are based on. Chatbots can be used to navigate the vast field of information. For example, FAQs can be replaced by chatbots that give only the answer needed without reading through all the questions asked [21], or chatbots could be used to engage with election panel data [22]. The chatbot needs to generate a query to navigate more complicated, larger databases. This leads to issues like understanding which attribute the user refers to with their query [23]. It can be said that the hardest step is to translate the user's query into a database query. Vakulenko and Savenkov (2017) used an approach where part of a table is used to train a model for table lookups. This model can look up data based on the user's query and answer with natural language. It is limited to one lookup at a time, can only access table-based data, and cannot handle inadequate user queries [24]. Information retrieval also improved with the recent improvements in Large Language Models (LLMs). An LLM is a kind of Artificial Intelligence (AI) that can understand and process human language [25]. The development of LLMs happens so fast that in 2017, chatbots were already forecasted to become the primary way of interacting with computer applications [26]. Li et al. (2024) have, therefore, much improved on text-to-SQL being able to perform it on "large and dirty databases," but even with the newest powerful models for query generation, a lack of robustness is observed [27]. Little research is done on non-SQL databases such as GraphQL. Ni et al. (2022) took the challenge to create GraphQL's tree-based queries from natural language by combining the knowledge of the entire LLMs and information about the database schema [28]. In general, scepsis still remains towards LLMs and chatbots. Therefore, trust should be established. Explainability, especially when something does not work, helps the user gain trust [29], for example, for the before-stated confirmation of already gathered information.

4 Requirements Analysis

After introducing some concepts, the core phase of the design study methodology starts with the discovery step.

4.1. Methodology

Domain problems and requirements are identified from two different sources: literature and interviews designed for this thesis. In the following section, a guideline for the interviews is introduced.

The following interview concept is based on Helfferich (2022) [30]. The concept of the guideline-based interview is chosen. This kind of interview, often also called a semi-structured interview, is recommended by different authors, e.g., Adams, W.C. (2015), Cohen D., Crabtree B. (2006), and Cachia, M., Millward, L. (2011), because they offer the benefits of both structured and unstructured interviews. The structure allows for easier preparation and evaluation of the interview while questions can be open-ended and follow-up questions can still be asked [31], [32], [33]. Semi-structured interviews are also well-suited for (video) calls [33]. Usually, the guideline consists of a mix of open questions and narrative requests. The guideline must not be followed word by word, but the intention of the questions should not change between interviewees. Similar questions between interviews facilitate the evaluation afterward. Three steps in the interview for creating the guideline are recommended in the literature:

1. The interviewee can speak as openly as possible based on the request to narrate.
2. Pre-defined points missed in the first step are asked again.
3. Structured and preformulated questions are asked last.

This way of starting with openness and concluding with structured questions helps to fulfill the requirements for a guideline [30]:

1. Openness: Allow the interviewee to express all his thoughts.
2. Clarity: Allow the interviewer to keep an overview of the plan and narrative.
3. Continuity: Do not interrupt any chain of thought of the interviewee.

4.2. Interview Design

The guideline-based interview will consist of six questions. The goal of the questions is to understand the interviewee's experiences, problems, and ideas with travel planning while also proposing the idea of a chatbot for itinerary planning and getting his or her thoughts about it. The guideline questions will be as follows:

1. Open questions to start the conversation

   What are your experiences with itinerary planning? Please walk me through the last time planning a travel itinerary. If you do not remember, imagine you want to go to Tarragona on the 15th of June. How would you approach planning?

2. Expected characteristics of platforms

   What platform did you use, and why did you choose it?

3. Possible difficulties

   Did you encounter any difficulties / challenges? How did you overcome them?

4. Requirements/ design solutions

   Would you improve anything on the platforms you use? If so, what and how?

5. Considered preferences

What were the most important factors for you when choosing a particular itinerary? Consider factors such as price, time of departure/arrival, duration, type of transportation, number of changes, etc.

6. Attitude towards a chatbot

What do you think about introducing a chatbot to process all your requests directly?

4.3. Selection of Interview Partners

The selection of the six interviewees tries to cover a range of ages, travel-booking experience, and digital competencies as broadly as possible. Older generations might be more used to booking all kinds of transportation itineraries with actual persons (travel agencies or even train tickets bought on board). In comparison, younger generations are used to booking everything online, comparing different web pages, and not getting any apparent support. Age, usual booking method, trips planned in the past year, digital competency and experience with user interfaces, and frequency of chatbot usage are recorded for all interviewees to allow an evaluation afterward. All interviews are performed in German as it is the first language of all interviewees and the author of this thesis. They were conducted via an online video call, except for the fourth interview, which took place in person.

ID	Age	Gender	Usual Booking Method	Frequency trip planning per year	Digital competency	Frequency chatbot usage per year	Duration
1	55	Female	DB-Navigator	1	Weekly	0	15:50
2	25	Male	Omio, Check24	3	Daily	5	16:06
3	56	Male	Booking.com, Check24, DB-Navigator	11	2 times per week	0	14:56
4	30	Male	Google-Maps, BusBahnBim, CheckFelix	3	Daily	1	10:30
5	24	Male	DB-Navigator	5	2 times per week	4	13:47
6	25	Female	DB-Navigator	25	Daily	0	18:33

Table 1: Demographics of interviewees

The above table shows diversity for all parameters. The interviews aimed to get some inputs and ideas and not to get representable data. This is the reason for having only six respondents.

- The age is distributed from 24 to 56 with an average of 35,8.
- The gender of the interviewees was two-thirds male and one-third female.
- The number of trips planned last year ranged from 1 to 25, with an average of 8.
- The frequency of chatbot usage ranged from 0 to 5, with an average of 1.66.

- The average interview took around 15 minutes, with the longest interview lasting 18:33 minutes and the shortest lasting 10:30 minutes.

4.4. Domain Problems

The main sources for the abstracted problems are the interviews as well as Meske, Bunde, and Ehmke (2020) [34]. Much other literature covers prerequisites of route planning applications, such as data being available or users being open to new applications. Requirements for the system itself are hardly addressed [35], [36].

The following table shows the abstracted domain problems with the sources in the right part of the table, literature, or interviews. The number in the respective columns shows either the interview ID or the challenge number identified by Meske, Bunde, and Ehmke (2020).

Domain Problems	Sources	
	Literature	Interviews
1. Too many input fields, which may also differ, are considered impractical and overwhelming	1	2
2. Number and quality of available routes depending on different filters, e.g., max number of changes, are not predictable	1, 5	
3. Too few filter options, especially for special user groups (families, impairments…)	2	
4. Assistance through the different filters is missing	3	
5. Preferences are not integrated properly leading to a large solution space	4	
6. More attractive options, that are slightly outside the applied filter range, are not shown	6	
7. Transparency for recommendations is missing	7	
8. Combination of different transport modes must be looked at individually		2, 3
9. Best price could be on a different platform		2, 3
10. Best in one-dimension options (e.g. fastest, least transfers, least walking) are not shown		2
11. Request cannot be processed, and no alternative is offered		2
12. Criticism/ skepticism towards AI-Application		1
13. Comparison portals do not always show the best price		3, 4

14. Lacking price transparency for included services		3
15. Booking process too complicated		6

Table 2: Domain Problems

The challenges build one of the pillars for the following requirements.

4.5. Requirements

The requirements are deducted from three different sources. In the source's columns of the table, the source can be found:

1. Requirements mentioned in the interviews.

   The number in the following table refers to the number of the interview.

2. Domain problems collected in the previous chapter.

   The number in the following table refers to the number of the domain problem.

3. Features of existing travel-planning solutions.

   The App by Deutsche Bahn, the DB-Navigator (DB-N), was referenced in several Interviews (I1, I2, I3, I6). It was also described as exemplary for just booking train journeys. Therefore, the characteristics will be analyzed and added to the list of requirements. Google Maps (GM) was also mentioned several times (I4, I6), as well as Bus-Bahn-Bim (BBB) (I4). The references in the following table show which of the requirements can be found in the respective existing solution.

The requirements are directly grouped into different types and priorities to allow for their use in the prototyping step. The different types are "functionality" and "usability". "Functionality" refers to tasks where the user can interact. "Usability" describes complementary additional characteristics.

We categorize the priority into four parts. "One" is the most important, while "Four" is likely out of the scope of this thesis. They are classified by their necessity and added value. The application would not work without the tasks from priority one, and the tasks from priority four add less value than those from priority two.

The sources follow the same structure as the table of the domain problems; the column shows the source, while the cell content specifies where the requirement comes from in the respective source.

Lastly, the requirements are categorized into different design goals corresponding to the different parts of the prototypes to create. This is solely done to understand the connection between the requirements and the prototype designs. These design goals will be referenced in the next chapter.

The table is sorted by priority first and design goal second.

Requirements	Type	Priority	Sources	

			Inter-views	Existing solutions	Domain Problems	Design Goals
1. Input of departure and destination	Functionality	1	2	DB-N, GM, BBB		Input
2. Input of date and time	Functionality	1		DB-N, GM, BBB		Input
3. Show a manageable number of options	Functionality	1	4		5	Output
4. Show current search criteria	Functionality	2		DB-N, GM, BBB		Information regarding the search
5. Allow several (filter) options	Functionality	2			3	Input
6. Integrate all preferences	Functionality	2			5	Input
7. Manageable amount of input fields	Usability	2	6		1	Input
8. Include different types of transportation options	Usability	2	1, 2	DB-N, GM, BBB	8	Output
9. Show good results with not much input	Usability	2	2			Output
10. Itineraries should be structured	Usability	2	3	DB-N, GM, BBB		Output
11. Possibility to change the sorting of the itineraries	Functionality	3		DB-N, GM		Input
12. Give some guidance through the process	Functionality	3			4	Information regarding the search
13. Redirect to some form of help if request cannot be processed	Functionality	3			11	Information regarding the search
14. Being able to save shown itineraries	Functionality	3	1			Saving routes

15. Add link to operator's platform	Functionality	3			13	Output
16. Show effects of changing filter options	Usability	3			2	Information regarding the search
17. Make reason for recommendations visible	Usability	3			7, 12	Information regarding the search
18. Reduce wait times	Usability	3	2			Output
19. Information where to enter and exit the train (-station)	Usability	3	6	GM		Output
20. Input of traveler's information	Functionality	4		DB-N		Input
21. Quick and easy booking	Functionality	4		DB-N	15	Input
22. Good solutions slightly out of the selected filters are shown and marked	Usability	4	2		6	Output
23. Show best price	Usability	4			9	Output
24. Show as many as possible Pareto-optional options	Usability	4			10	Output
25. Show included services	Usability	4			14	Output
26. Information regarding the ticket booking process	Usability	4	4			Output
27. Information regarding services provided at train stations	Usability	4	4			Output
28. Live Information and location	Usability	4	6	DB-N, GM, BBB		Output

Table 3: Requirements

This step allows us answering the first additional research question: 'What are the requirements the interface should fulfill to make it accessible to a broad range of users?'

5 Prototyping a chatbot application

In the following chapter, the prototyping process will first be described methodologically, followed by preparing the prototyping before designing a prototype and going through a few steps of iterative improvement. This represents the design stage of the design study methodology.

5.1. Methodology of Prototyping

Prototyping is a complex process essential for developing a user-friendly tool. Therefore, the methodology will be discussed in more detail in this chapter. This whole chapter follows 'Effective Prototyping for Software Makers' [37]. However, prototyping represents just a small part of this thesis. Therefore, some details from the literature are neglected as the methodology was initially conceptualized for more complex applications created by whole teams.

The process is divided into four phases: (1) Plan, (2) Specification, (3) Design, and (4) Results.

1. Plan

The prototyping starts by verifying requirements and potentially listing them in tabular form. A classification in type and priority is preferable. In the next step, a task flow is derived to provide an overview of the most important user actions and their dependencies. Afterward, a prototype design and fidelity are defined. This is unimportant for this thesis as a low-fidelity prototype as a sketch is the only type achievable in the given time.

2. Specification

Characteristics such as the target audience and medium are defined first. A method and, finally, the tool used for the prototyping is decided upon based on these characteristics.

3. Design

The next phase is the most relevant one. In the first part of this phase, design criteria are specified, which will be followed in the second part of the design creation. Starting with the most complex part of the prototype, every page should be divided block-wise into functional regions, e.g., the top part being navigation, underneath an input field, and so on. Detailing the blocks by urgency slowly completes the picture. The blocks maintain the possibility to rearrange items to comply with the design principles chosen beforehand. The prototype design is ready for the next step as soon as all the design principles have been fulfilled.

4. Results

The last phase concerns reviewing, validating, and finally deploying the prototype. The review and validation are supposed to happen internally as well as externally. For example, internal members are team members, while external validation happens with potential end users. It is not only about showing the result but also about understanding the process behind the creation. External validation happens by showing different options to potential users and letting them decide which ones they like the most.

5.2. Prototyping Preparation

As low-fidelity is the only type of prototype achievable, the second phase, the specification, can be described next. The prototype is developed for potential end-users. Here, the design is decided, which is the basis for the implementation. The prototype consists of sketched parts of the application that are not interactive. These sketches are described in a narrative style where the audience can ask questions. This method is called wireframe prototyping [37].

The requirements have already been identified in the previous chapter. Types and priorities have been classified at the same time. Therefore, it can be continued with the next step.

From the "Essential" and "Functional"-type tasks, a task flow is created at first. The flow is from top to bottom. The indented tasks on the right are part of the input process. The two tasks on the left parallel all the other tasks.

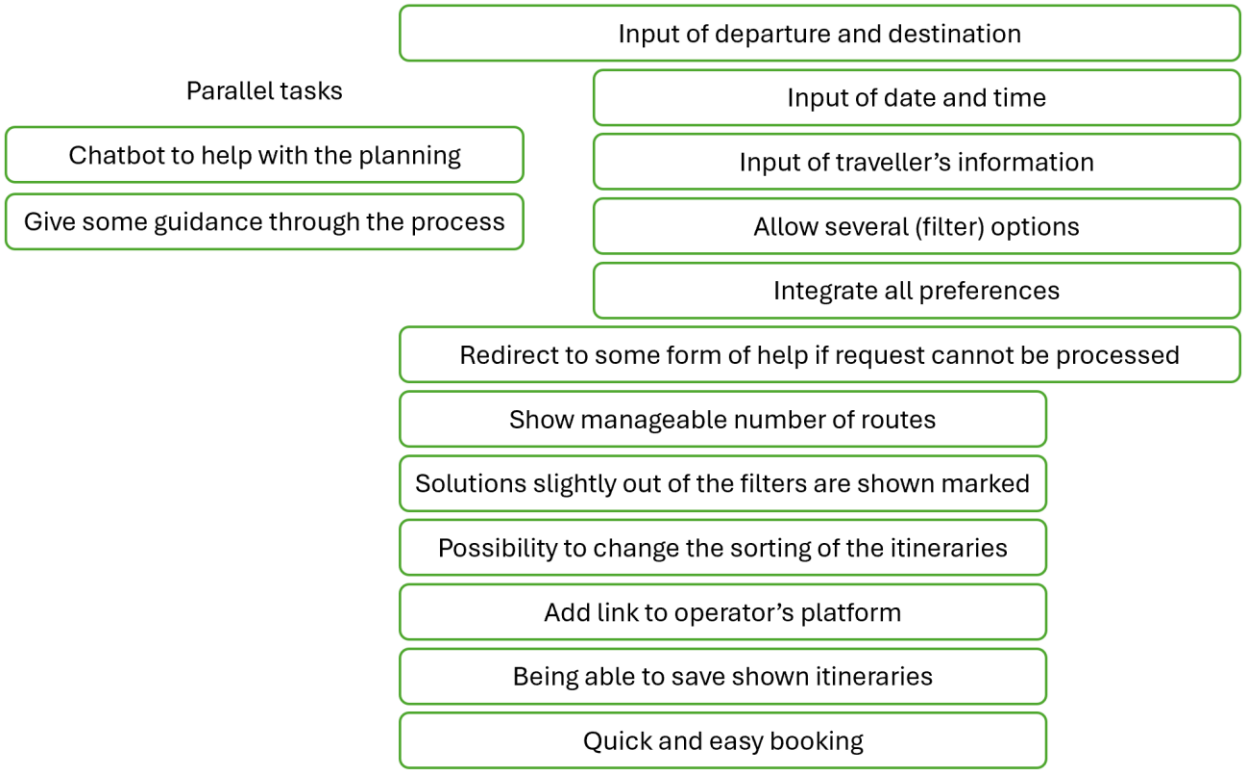


Figure 1: Task flow

A few design principles to follow for the next step are defined. They are chosen from the list in Arnowitz et al. (2007) [37]. The first design principle is a grid-based organization, meaning ordering design elements in an imaginary grid to make the page less messy. The second design principle is universal accessibility, which is a big part of the objective of this thesis. The last design principle is unity and variety. This is a challenge for the design step as the goal is to have uniformity in the overall design concept while highlighting important functionalities by applying variety to the underlying design component.

5.3. First Prototype Design

Following the literature, a block-wise division is made and filled out. The original first result is the following sketch:

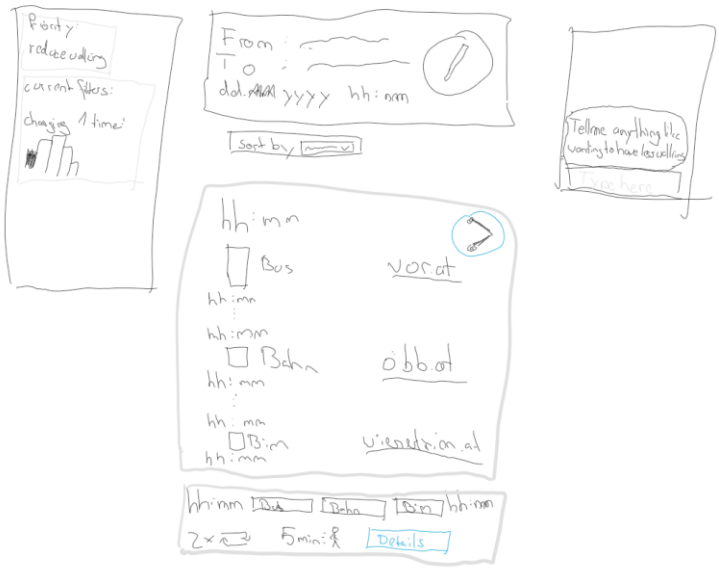


Figure 2: First prototype sketch

This first set of prototypes, like the one shown, is quickly discarded as one of the most important requirements to have a manageable amount of input fields is not satisfied with having the chatbot as well as the input options for the destination, departure, and the date and time. Many of the thoughts put into these prototypes were considered anyway.

We identified some other problems with the first prototype. They are listed to be improved for the next step:

- The chatbot should be the main or only possibility to interact with to have a manageable amount of input fields.
- A chat history is not necessary, as all relevant information is part of the information view on the left side.
- Schedules proposed earlier should have the possibility to be saved and to be accessible later on so that one can continue the search with different parameters.
- When schedules are shown, they should be named to allow interaction with them.
- The reason for showing each itinerary should be visible, e.g., when more than one priority is given.

Additionally, some takeaways to facilitate the next prototyping steps were:

- Familiar designs are good but not entirely necessary.
- Different kinds of information displays should be proposed for itineraries, like a tabular or a map view, to allow potential users in the next step to choose what they like the most.
- Different approaches for the other single components of the prototypes are also needed to give a choice in the next prototyping step.

These problems and insights are taken and incorporated into the next step. Based on the design goals from the requirements Table 3, a second set of prototypes is again done as a block-wise division consisting of three blocks from left to right:

1. Information about the underlying criteria for the route search (all requirements for the design goal of 'Information regarding the search')
2. A chatbot window where itineraries are also shown (all requirements for the design goal of 'input and output')
3. A block where favored itineraries can be saved to (all requirements for the design goal of 'saving routes')

Each of these blocks has different possible designs which can be combined in any way. This contributes to the usual grid-based organization with main content in the middle of the page and additional information to the left and right as in Figure 3 can be seen. The final design options for the single blocks can be seen after the next chapter of the internal prototype validation.

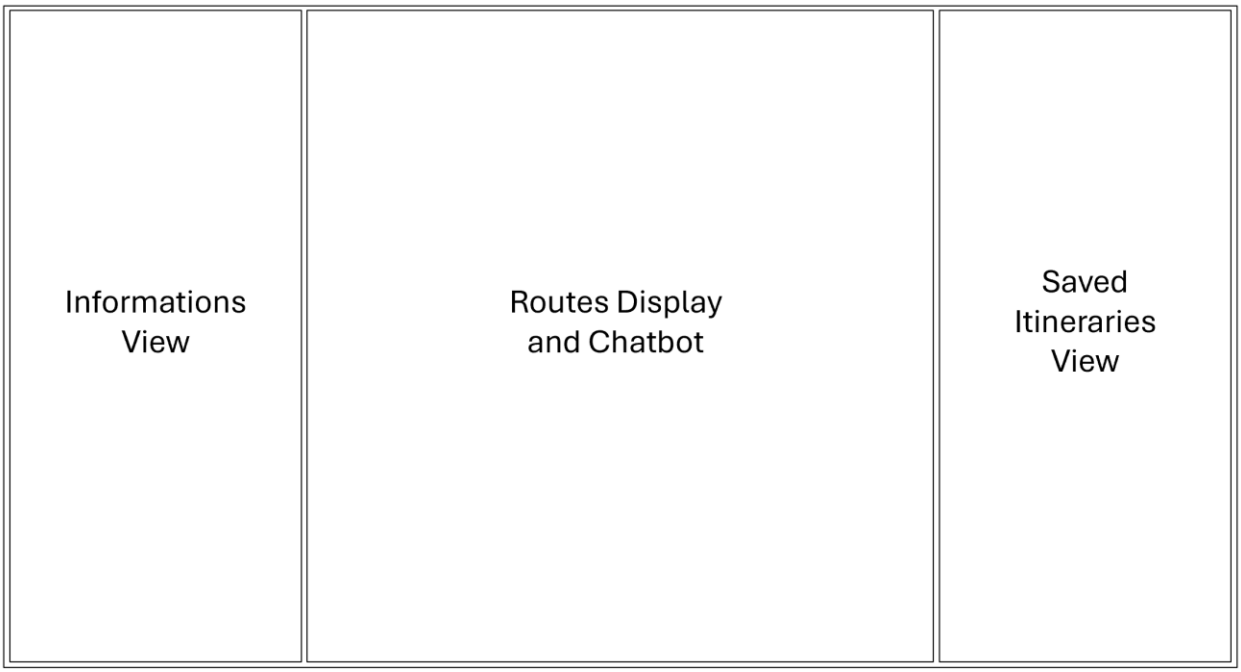


Figure 3: Grid-based application organization

5.4. Internal prototype validation

After implementing the set of design options, validation needs to be done. First, an internal review session with different members of a supporting research group and a supporting professor is done.

ID	Position	Field of research
1	Professor	Visualization and Data Science
2	Postdoctoral Researcher	Visualization and Data Science
3	Scientific Researcher	Visualization and Data Science

4	PhD Student	Visualization and Data Science
5	Post-Doctoral Researcher	Astronomy and Data Science
6	Professor	Business Analytics

Table 4: Interviewed research staff

The set of design options like the one presented in the next chapter is shown to the respondents. New ideas, critical thoughts and improvements are stated. This internal review allows for some additional insights from a scientific viewpoint. The most important insights and ideas are given before the final design options are shown:

- Allow vocal input
- Order the shown routes by time of the day
- Label parts of the application clearly
- Onboarding: Small tutorial on how the page works when the application is started
- The borders of map show the geographical scope of the application
- When no route is found for too many applied filters, show one slightly out of the applied filters, but clearly indicate it being outside the applied filters
- Adding colors for the saved itineraries' view, how 'good' certain values are, to allow comparability at first glance

These ideas are now investigated and if possible integrated.

5.5. Prototype design options

Finally, the prototype parts are designed, having the design principles in mind. They are divided into the components and design options. For the components the goal is to fulfill the requirements with as many priority levels as possible. Different design options fulfilling the requirements in different ways are displayed. The components consist of two alternatives for the chatbot window, four alternatives on how to show possible routes, five alternatives for the displayed information and three alternatives for the saved itineraries view. The possible designs are shown in the following chapters.

5.5.1. Chatbot Window

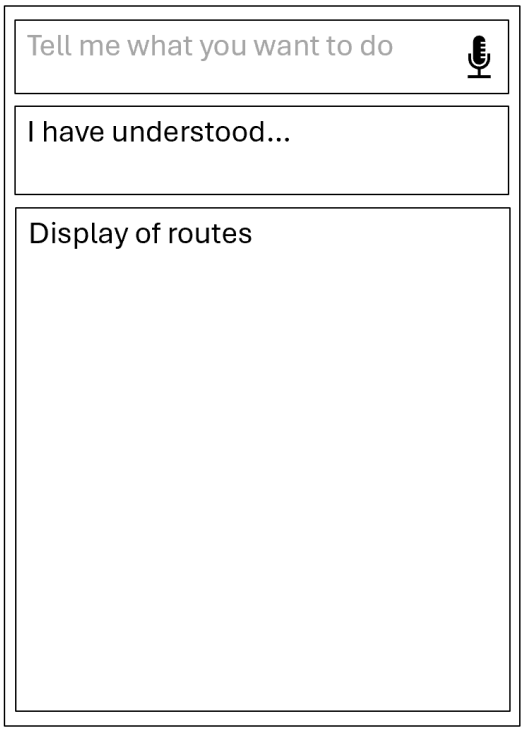


Figure 4: Chatbot window: input at the top

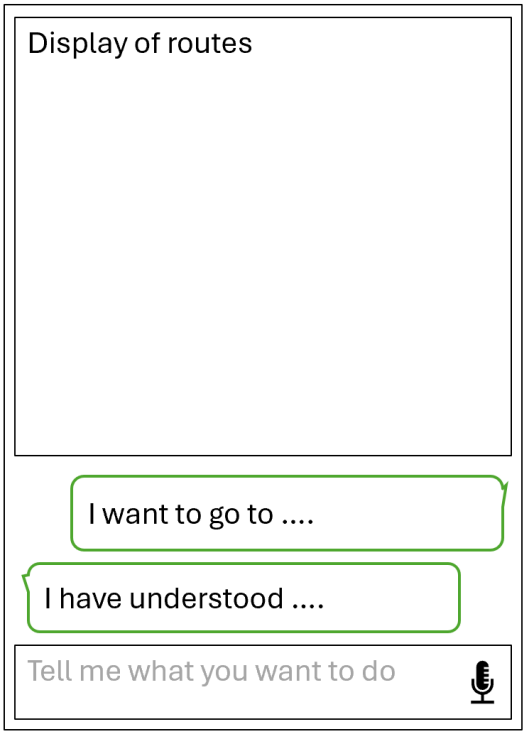


Figure 5: Chatbot window: input at the bottom

Two options are offered for the chat window. The first figure shows the input field on the top, while the second figure shows the input field on the bottom, as is common in many chat applications like WhatsApp or ChatGPT. The second option additionally shows the last input message. This part corresponds to the design goal “Input” from chapter 4.5 Requirements. All requirements from priority one and two are fulfilled as all requests can be typed into the one input field:

1. Input of departure and destination
2. Input of date and time
3. Allow several filter functions
4. Integrate all preferences
5. Manageable amount of input fields

5.5.2. Display of Routes

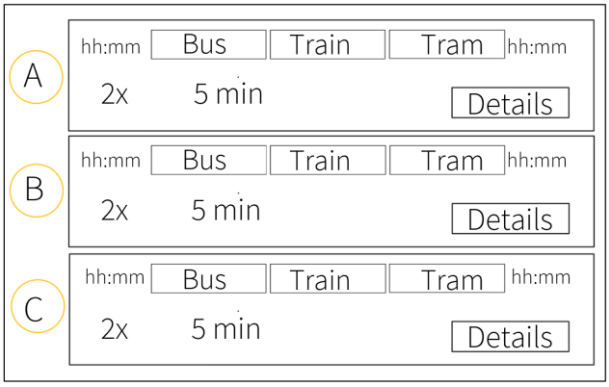


Figure 6: Display of routes: conventional view

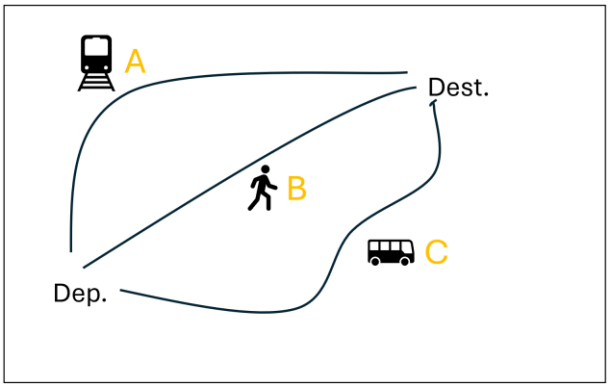


Figure 7: Display of routes: map

A		B		C	
HH:MM	Dep	HH:MM	Dep	HH:MM	Dep
HH:MM	Inter1	HH:MM	Inter1	HH:MM	Inter1
HH:MM	Inter1	HH:MM	Inter1	HH:MM	Inter1
HH:MM	Inter2	HH:MM	Inter2	HH:MM	Inter2
HH:MM	Inter2	HH:MM	Inter2	HH:MM	Inter2
HH:MM	Dest	HH:MM	Dest	HH:MM	Dest

Figure 8: Display of routes: intermediate stops

	Dep	Dest	Duration	Changes	Mode
A	HH:MM	HH:MM	00:15	1	Tram
B	HH:MM	HH:MM	00:23	2	Bus
C	HH:MM	HH:MM	00:45	0	Walk

Figure 9: Display of routes: key facts

The display of the routes has four alternatives:

1. The first one is inspired by platforms mentioned in the interviews, such as Google Maps and DB-Navigator. It shows the time of departure and arrival, the transportation mode as well as the number of changes, and the duration of the route.
2. The second option consists of a map with the routes sketched. A small icon marks the mode of transportation.
3. A table marks the third option. The focus is on intermediate stops and their timings, as little more information is shown.
4. The fourth option consists of a table. Key facts such as departure time, destination time, duration, number of changes, and the mode of transportation are shown. A color scheme allows for easy comparison between the fields. The color scheme consists of three colors: green, yellow, and red. Each color represents a third from the best to the worst value in the respective column. Green represents the best values, yellow represents values around the median, and red represents the worst values.

Each route in each view is noted by an orange letter to be able to index that route to the chatbot. Each option visualizes the requirement for the design goal of the output. Again, all requirements of priorities one and two have been fulfilled at all solutions:

1. Show a manageable number of options

2. Include different types of transportation options
3. Show good results with not much input
4. Itineraries should be structured

The fourth requirement for the map view can be debated. The routes are not structured in a table but geographically.

5.5.3. Information Tab

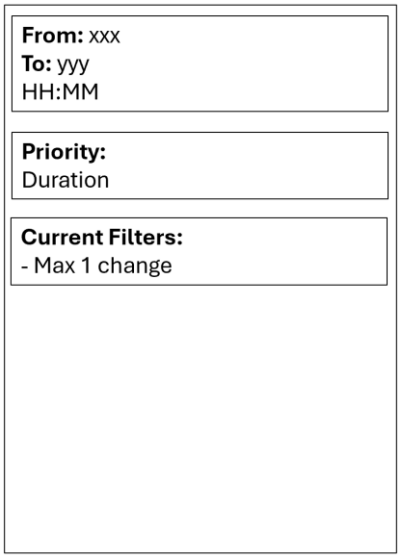


Figure 10: Information tab: show dynamically

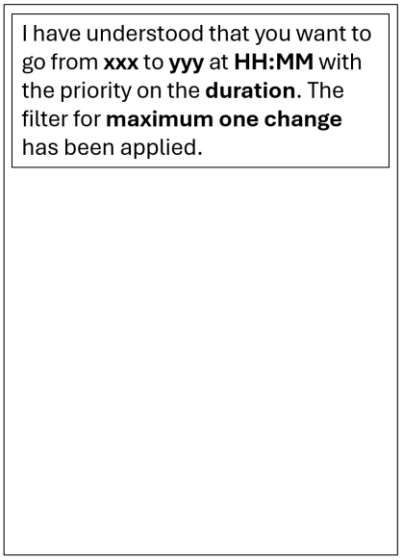


Figure 11: Information tab: running text

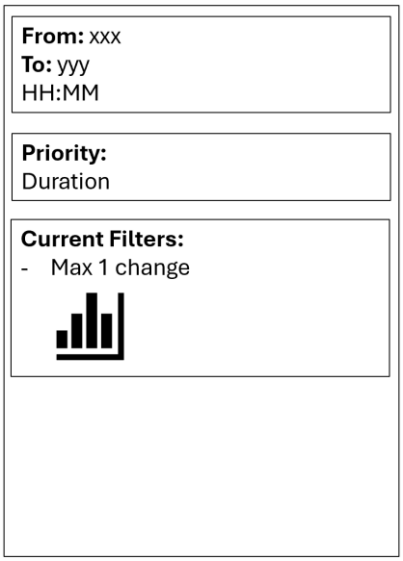


Figure 12: Information tab: graphics added

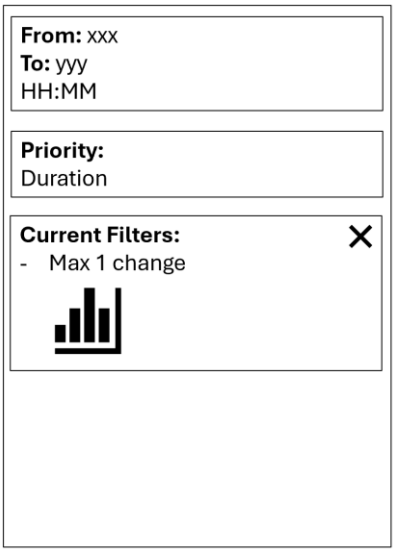


Figure 13: Information tab: allow interactivity

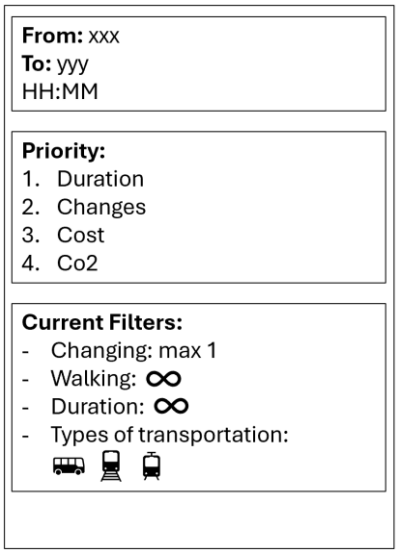


Figure 14: Information tab: show everything

Five options are available for the information tab, corresponding to the design goal of 'Information regarding the search':

1. The first option shows the simplest one. Departure and destination, as well as the departure time, are always shown. Everything else, like filters, is only displayed when selected by the user.

2. The second, most different, option gives all the information the model is using in a running text.
3. The third view behaves similarly to the first one, but a small graphic is also shown for all applied filters. The graphic shows how the number of possible routes would change depending on the filter.
4. For the fourth option, interactivity with the information tab is allowed. The filter can be deleted by clicking the cross associated with the respective filter.
5. The last figure always shows all the possibilities of the underlying model. The priorities are ranked, and the filters are shown as infinite if nothing else is specified.

The only requirement of priority one and two for the design goal has been fulfilled:

1. Show current search criteria.

5.5.4. Saved Routes View

	Dep	Dest	Duration	Changes	Mode
S1	HH:MM	HH:MM	00:15	1	Tram
S2	HH:MM	HH:MM	00:23	2	Bus
S3	HH:MM	HH:MM	00:45	0	Walk

Figure 15: Saved routes: values

	Time	Duration	Changes	Mode
S1		00:15	1	Tram
S2		00:23	2	Bus
S3		00:45	0	Walk

Figure 16: Saved routes: graphics

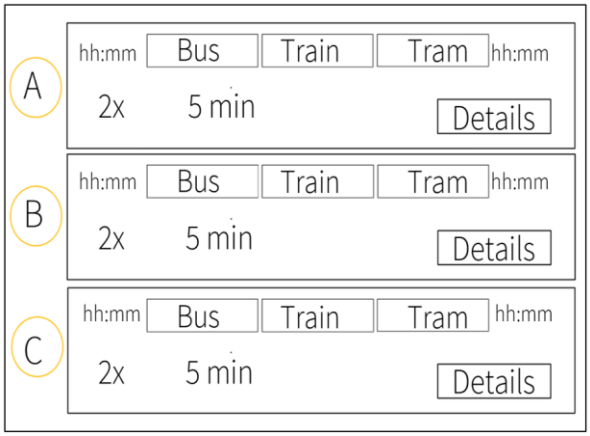


Figure 17: Saved routes: conventional view

The three design options of the saved routes view are similar to the ones from the display of routes. The focus is to make routes comparable to each other to decide the final route summarized by the design goal of ‘saving routes’.

1. The first option is built like the fourth display of routes option with the key facts and the colors, making the comparison possible at first glance.
2. The second alternative replaces departure and destination time with a clock.
3. The third option is again similar to the first display of the routes view, which is similar to existing platforms.

There are no requirements for the design goal of saving routes with a priority of one or two. The only requirement for this design goal is of priority three and has been fulfilled with the design options given:

1. Being able to save shown itineraries

5.6. External prototype design choices

In the next step, another small survey is conducted. Several potential users who have not yet been part of the development process are shown the different possible designs for all the components. An overview of the questioned users is given as follows:

ID	Age	Gender	Usual Route-Planning Method
1	25	Male	Google Maps
2	34	Male	ÖBB Scotty
3	24	Male	DB-Navigator
4	23	Female	DB-Navigator/ Google Maps
5	24	Male	ÖBB Scotty
6	25	Male	Google Maps
7	25	Female	ÖBB Scotty
8	55	Female	DB-Navigator
9	23	Female	Google Maps
10	23	Male	Google Maps
11	53	Male	DB-Navigator

Table 5: Overview of questioned potential users

An explanation and the reason for overall design choices are also given to the respondents. Everybody chooses their favorite and second favorite design for each part. A table with the answers and a small analysis is given as follows. The tables are sorted by the sum of the first and second votes.

Design option	# of first votes	# of second votes	Place
Input at the bottom 	11	0	#1
Input at the top 	0	0	#2

Table 6: Choices for the chatbot window

The respondents unanimously favored having the chat input at the bottom. A second vote was not noted as it would be redundant.

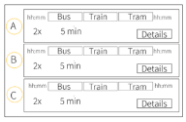
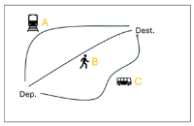
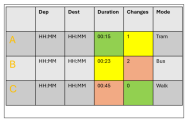
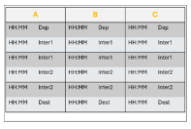
Design option	# of first votes	# of second votes	Place
Conventional view 	6	4	#1
Map 	1	5	#2
Key facts 	4	1	#3
Tabular 	0	0	#4

Table 7: Choices for the display of routes

The votes for the display of routes were split between a conventional view like the German DB-Navigator or Google Maps, a table with key facts, and the display of the routes on a map. Nearly everybody chose the conventional view as either their first or second choice. The map was often chosen as a good optional addition and got the second most combined first and second votes. The table with key facts was, if chosen, mostly the first choice. Noticeably, this view was mainly liked by people who are used to working with a

lot of information at work. As not everybody does this, interpreting the data could be more challenging for others. Therefore, this table will not be in the final prototype and placed third. The main view will be the conventional view with an option to show the map.

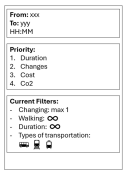
Design option	# of first votes	# of second votes	Place
Dynamically 	4	4	#1
Interactivity 	5	1	#2
Show everything 	1	4	#3
Graphics 	1	2	#4
Running text 	0	1	#5

Table 8: Choices for the information's tab

Preferences on the tab to show information were more varied. Except for showing a running text, every option got at least one of both first and second choices. The second and third most chosen views allow interactivity and show all different filters, etc. The most chosen option to dynamically show only current factors for the route search was chosen by over 70% of the respondents and is therefore selected for the final prototype.

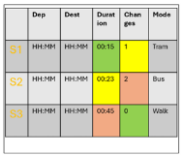
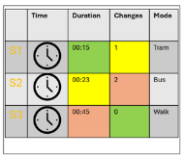
Design option	# of first votes	# of second votes	Place
Values 	5	4	#1
Graphics 	4	4	#2
Conventional view 	2	1	#3

Table 9: Choices for the saved routes view

Decisions regarding the saved routes were mixed again. The favored was the table with key values, followed by the table with mixed graphics and values. The same view as the conventional display for the saved routes was only chosen 3 out of 11 times (first and second choices). Therefore, a table with values will be part of the final prototype.

5.7. Final prototype

Based on the analysis above, a final prototype has been decided upon. A sketch can be seen in Figure 18. Interaction is only possible with the chat field at the bottom of the middle. A new routing request can be entered, the existing request can be continued, or one of the routes shown can be saved to the right part of the window.

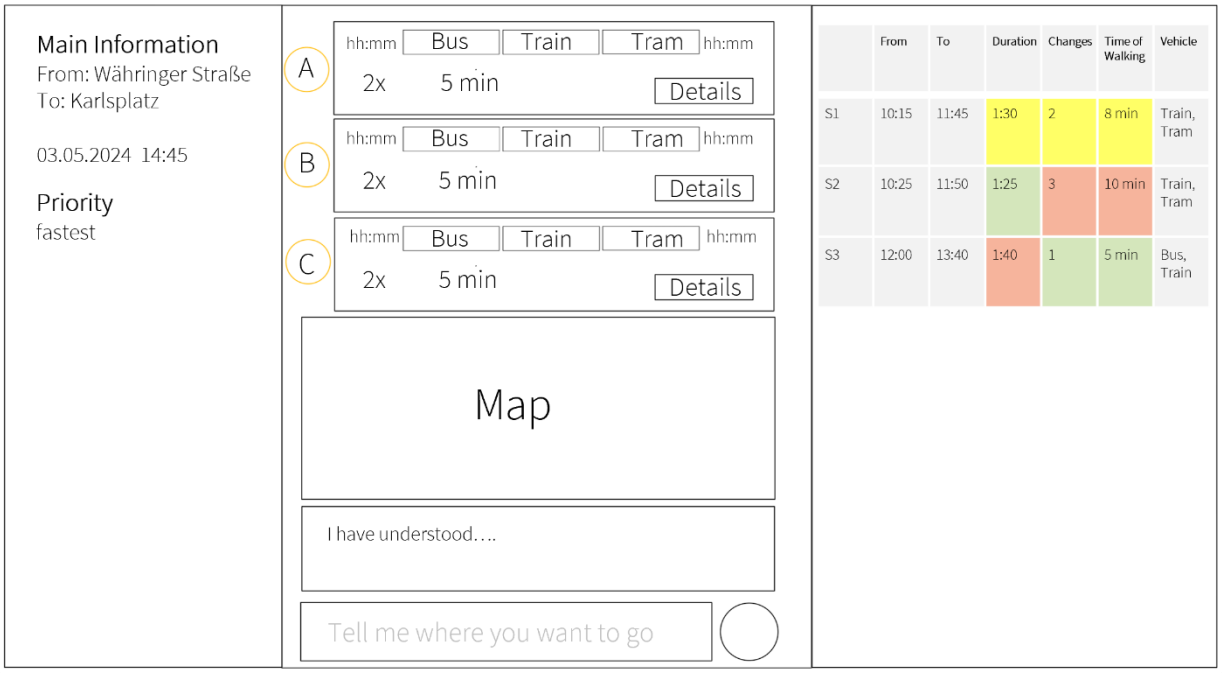


Figure 18: Final prototype

The final prototype answers the second additional research question: 'What is an effective user interface enabling the requirements?' The effectiveness will be tested in the last step.

6 Application Development

In the following section, we describe the high-fidelity prototype based on the low-fidelity prototype introduced in the implementation phase of the design study methodology. Firstly, all databases and tools used are introduced. Afterward, the functionality of the backend is described in detail. The fulfillment of the characteristics described in Chapter 4.5 will also be checked. The code is published in the author's GitHub repository[3].

6.1. Data Basis, OTP2

The data is based on an open-source tool called OpenTripPlaner 2 (OTP2) [38]. This tool combines data from OpenStreetMap (OSM) with timetable data in GTFS format, which is made available by most transportation agencies. The tool creates a graph based on this data when it is started the first time. This can take some time. For example, building the graph for Austria took around 25 minutes on an AWS EC2 r5a.xlarge instance. The integrated route-searching algorithm allows interaction with this graph. OTP2 can run on any device, i.e., calls to the interface do not incur any costs. Hosting is possible on any local or virtual machine. In this case, a virtual machine, the AWS EC2 instance, is used. OTP2s API is a GraphQL API that takes a JSON format as an input and delivers its output similarly. The API offers much customization,

[3] https://github.com/Henryi896/route-planning-chatbot

from choosing the mode of transportation to specifying if a rental bike used in the route must be returned before reaching the destination. This allows us to handle many specific user requests.

6.2. ClaudeAI

The chosen AI Assistant for this thesis is ClaudeAI because of costs, capability, and availability. ClaudeAI is an AI Assistant by Anthropic. The Claude 3 model family was released on the 4th of March and was made available to Europe 10 days later. The best model outperforms the well-known ChatGPT-4, and even the cheapest model outperforms the ChatGPT-3.5 model. Anthropic's website praises accuracy, extended context of 200,000 tokens, and near-perfect recall [39], [40]. The model family consists of three models with different capabilities and different pricing. The cheapest model costs 25ct per million input tokens, and the most capable model is already 3$ for the samfie input. Output tokens cost five times as much as the respective input tokens [41]. ChatGPT by OpenAI's pricing ranges from 50ct to 5$ for slightly less-performing models [42]. Better performance, lower price, and easy accessibility make ClaudeAI an attractive AI assistant for many applications.

6.3. Tools Used for the Interface

The tool is built in JavaScript using D3 and the vue.js framework [43], [44]. The tailwind framework is additionally used for the styling [45]. Vue allows reactivity while keeping the application modular. D3 helps with dynamic manipulation of parts of the interface. The map is supported by leaflet. The interface is also hosted on the EC2 instance described in Chapter 6.1 Data Basis, OTP2 and can be accessed through any web browser.

6.4. From user input to routes

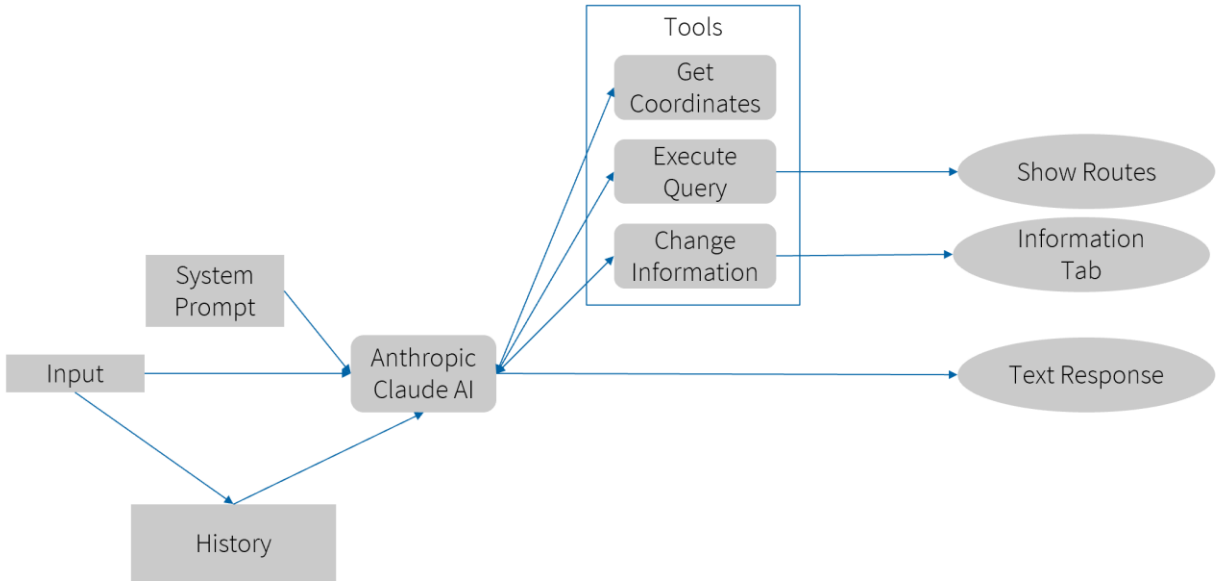


Figure 19: Backend process from user input to display of routes and information

Developing a process to understand the user's input and converting it into the routes shown needed the most work in this thesis. Figure 19 shows the final backend process of the developed chatbot process. Once the application is initialized, the user can enter her input. This input is then processed by the chosen AI assistant ClaudeAI. The system prompt and the current chat history are considered for this processing.

The algorithm itself chooses whether it needs to use a tool or directly give a text response. If all the necessary information to create a query is gathered, the query is sent to the 'Execute Query'-Tool. Usually, the answer should contain up to three routes. In case either no routes or an error is returned, ClaudeAI tries to correct the query and executes the tool again. When routes are found, they are shown to the user, the information view is updated, and a text answer is generated by ClaudeAI. Afterward, the user can follow up on these routes or start a new inquiry.

6.5. Final application

Finally, the data basis, all tools, and the process from the input to the routes is put together to develop the high-fidelity prototype, including the chatbot. The interface, developed by the author of this thesis, can be seen in Figure 20. The similarity to the final low-fidelity prototype is obvious. The biggest difference is the map that changed position to the saved routes view, as the space between the routes and the chat window got too small with longer inquiries and longer answers from the chatbot.

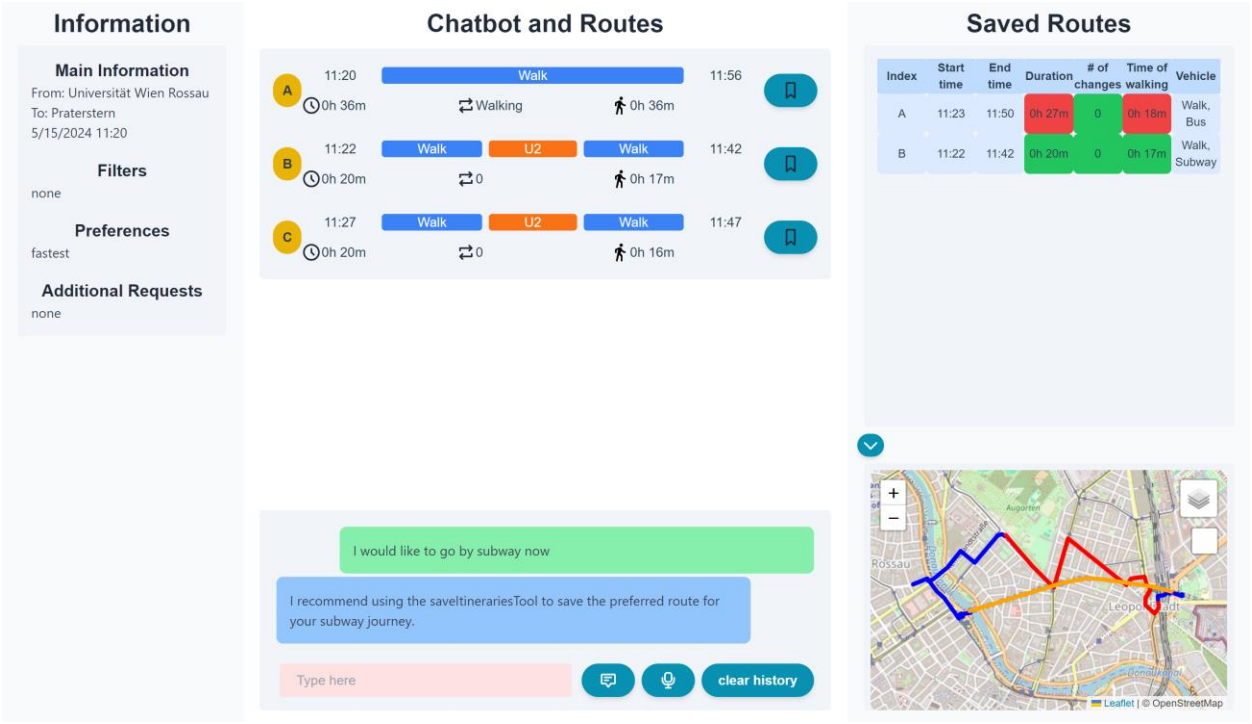


Figure 20: Final application interface

A comparison to the requirements from Chapter 4.5 should be made. All requirements with a priority of two or higher have been achieved. Additionally, some guidance is given, routes can be saved, and even if no routes for the current filters are found, some routes outside of the applied filters are shown and marked (e.g. showing a train route, if a bus route was requested, but does not exist). These were requirements with respective priorities three, three, and four. The backlog of additional requirements from chapter 4.5 is unfortunately too big for the scope of this thesis and leaves room for improvement and future research.

Having the final functional application affirms the third additional research question: Is it technically reasonable to integrate a chatbot in a route-planning application while considering the user requirements?

6.6. Encountered challenges

While developing the application, we encountered a few challenges.

1. Prompt engineering

Utilizing an AI API such as ClaudeAI needs customization to have the AI respond according to expectations. Customization happens through the system prompt. Getting this prompt right or as good as possible takes a lot of instinctive feeling. Sometimes, the model behaves immediately as you want, and sometimes, many specific instructions are needed to receive your wanted answer. This is probably one of the biggest downsides of using a non-deterministic black box.

2. Developments in the current chat models

Publicly available AI assistants are a very dynamic environment. New models and new providers pop up frequently. For example, ClaudeAI by Anthropic just became available when most of the application development was finished. A switch from the former used ChatGPT model was relatively easy due to the similar structure of the two models. They still behave differently, and the system prompt needed some adjustments. Furthermore, it is not only about different providers or models; the models seem to change over time. Unfortunately, this cannot be confirmed as the providers keep the models as black boxes.

3. Context limitations

The context one can give an AI assistant is limited by the number of tokens (something similar to syllables). This context cannot be arbitrarily large. Therefore, the AI assistant does not simply take all the documentation of the API, and the length of the considered chat history cannot be unlimited. Different solutions like vector stores or summaries are possible but behave much differently. This requires some trial and error.

4. Cost of AI

The cost of AI adds to the previous point. The longer the context, the more expensive each API call is. There are more and less capable models with different prices as well. Some optimization and consideration are needed to get the right value for the money.

5. Faulty GTFS-files

GTFS files contain all of the information about public transportation. Remarkably, most traffic enterprises offer and maintain them for free. But sometimes, the maintenance is not flawless. While trying to use this thesis' data basis, the starting process of OTP2 failed a few times because of faulty GTFS files. For example, route IDs were duplicates. Manual editing allowed for solving these errors.

6. Duration of each request

Each request to the API takes some time and cannot be influenced. Every time one of the algorithm's tools, introduced in Chapter 6.4, is utilized by the API, it behaves as another request to the API. Combining these two facts can lead to long waiting times for every input. Different factors can reduce this duration for each request:

- Where are the tools used?

- Can tools be combined into one?
- Can tools be substituted, e.g., by providing additional information to the AI?

More optimization is needed to reduce calls to the API and, therefore, waiting times and costs.

7 Evaluation

A test for the main research question will be designed using the guideline 'How to Design and Report Experiments' [46]. A proper cause and effect should first be abstracted from the research statement. The independent variable (the cause) is manipulated in the next step. Here, it is important to control for confounding variables. The individual's experiences within the given conditions must be comparable, and all the conditions must contain the same information. The provided materials in the scenario can be chosen using common sense. The number of test participants should not be too low to allow proper evaluation. A repeated measures design where each participant tests the different conditions reduces the needed number of participants and the sensitivity as the noise produced no longer depends on the tester. A proper experimental design, where a difference in the effect can directly be deduced from the changing cause, is the way to go, if possible.

The next step is measuring the dependent variable(s). In the best-case scenario, a parametric measurement is chosen to allow using parametric statistics. Examples of parametric measurements can be time to complete the task and self-report questionnaires. When using self-report measures, the validity must be considered, e.g., does the metric measure what it is supposed to? Additionally, the reliability of the measure should be considered. High reliability is achieved when the measure produces the same results under the same conditions. Lastly, measurement errors can always happen, even if everything is done to prevent them. The obtained score usually consists of four parts:

1. "A 'true score' for the thing we hope we are measuring
2. A 'score for other things' that we are measuring inadvertently
3. Systematic (non-random) bias - which isn't too bad as long as it affects all participants in the study equally, and not just those of some groups more than others
4. Random (non-systematic) error, which should cancel out over large numbers of observations." [46]

The primary goal is to maximize the part of the 'true score' rather than preventing all interferences, which is impossible. Additionally, the score should be designed so that not too many participants fall into the same category.

7.1. System Usability Scale (SUS)

The System Usability Scale is a standardized tool to measure the usability of a system. It was developed by Brooke in 1996 [47]. It offers a good basis for the questionnaire as its validity has been verified, and it is used widely in usability research as 18,311 citations on Google Scholar show. It allows for comparability to other studies but also between systems with a different interface technology [48]. The SUS consists of ten questions alternating between the expected response for a good application being strong agreement and the common response being strong disagreement. The questions will be answered after the system is used on a five-point Likert scale, from 'Strongly disagree' to 'Strongly agree' [47]. The SUS is described as one of the questionnaires needing the smallest sample size to be significant. A sample size of twelve is

considered enough in the often cited paper of Tulis and Stetson [49]. Even for small sample sizes of five, the mean is close to one of the larger sample sizes most of the time [50].

Translations to several languages have been researched. As this thesis is written in the German-speaking world, a translation by B. Rummel [51] is used, which is suggested to be a good compromise between wording and methodically clean development [52].

7.2. Test Protocol

Based on the methodology described above, a test protocol is developed to answer the research question stated at the beginning of this thesis:

Can a **generative chatbot simplify and shorten** the process of finding travel itineraries with **equally good or better results than existing systems,** given a layperson's **demands and preferences**?

The main research question is adding the chatbot to a route planning tool. The main effects are simplicity, duration, and usability of the process, with an expected positive correlation for simplicity and usability and an expected negative correlation for duration. Other metrics to consider in answering our research question are the quality of results and integration of the user's demands and preferences. The material is our novel interface utilizing a chatbot and two publicly available comparable systems (ÖBB Scotty and Google Maps).

The system is not shown to the attendees beforehand. Each participant will test all three systems. For this, they are assigned to one of three groups. Each group will see a different system at first to minimize learning effects.

Each participant receives a task she must complete with all three systems. The task will be to find a route from the address 'Hannovergasse 37, 1020 Wien' to 'Schladming' on 29.06.2024 at 10:00 with at most two transfers.

After getting to know the systems, the user must finish a self-imposed task. The goal of the self-imposed task is to evaluate the systems outside of the controllable variables. The user should define the self-imposed task before using any system. This reduces dependency on the before-used system and allows integration and testing of the user's preferences in each system.

In the next step, a measure is developed to observe the effect on the dependent variable. Duration is the easiest effect as it can be measured for the assigned task. To get results for the further effects, each participant is given a survey based on the standardized SUS test. This test covers simplicity and usability. The metrics of the research question that have not yet been answered are the quality of the results and the integration of demand and preferences. Therefore, the SUS test is extended by two statements. The additional statements are kept in the SUS style, alternating between a positive and a negative outcome. Finally, the following questions are to be answered by each participant:

1. I think that I would like to use this system frequently.
2. I found the system unnecessarily complex.
3. I thought the system was easy to use.
4. I think that I would need the support of a technical person to be able to use this system.
5. I found the various functions in this system were well integrated.

6. I thought there was too much inconsistency in this system.
7. I would imagine that most people would learn to use this system very quickly.
8. I found the system very cumbersome to use.
9. I felt very confident using the system.
10. I needed to learn a lot of things before I could get going with this system.
11. I thought that the results of the system were satisfying.
12. I would have liked to integrate my own preferences better.

From now on, these questions will be denoted by 'SUS-X' in the thesis. The SUS is answered after using one of the systems and before continuing to use another system.

After completing the tasks and answering the questions for all three systems, three open questions must be answered to allow some qualitative analysis of the results. The three questions are:

1. Which feature had the biggest impact on the usability of one of the systems?
2. Have problems or challenges occurred when using the chatbot system that have not occurred for the other systems?
3. Do you have any proposals for improving the chatbot system?

8 Results

In total, 12 respondents tested the three systems as this is the recommended smallest sample size [49]. At first, the demographics and the experience with chatbots and route-plannings are displayed. Afterward, the times needed for the tasks and the SUS scores are shown. In the end, correlations and interaction of the SUS answers based on two independent variables are calculated. In section 9, Discussion, these results are used to answer the research questions.

8.1. Demographics

A small overview of the demographics can be seen:

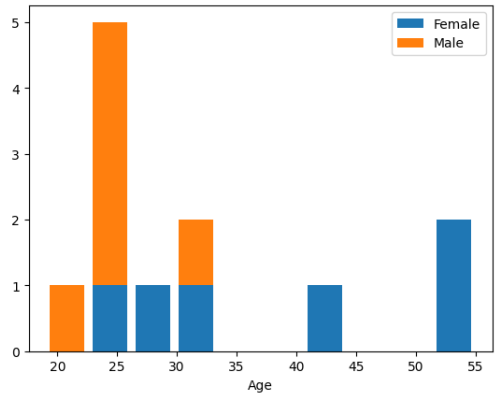


Figure 21: Age of respondents

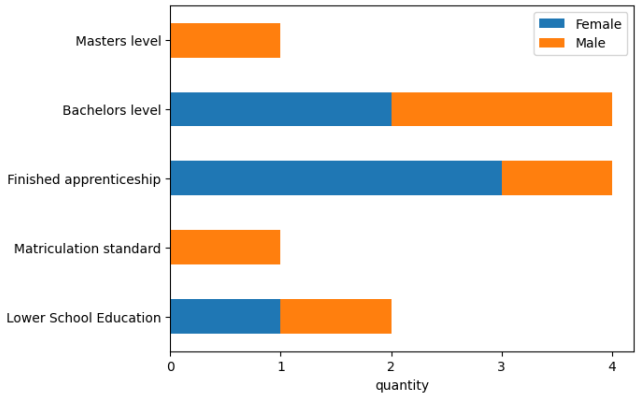


Figure 22: Education of respondents

Figure 23: Gender of respondents

The respondents' age ranges from 19 to 55 with half of the respondents being between 20 and 29 years old and a mean of 31.91. The level of education is more equally distributed with two thirds having either a finished apprenticeship or a bachelor's degree. The gender is evenly distributed between males and females.

8.2. Experience of respondents

A further overview of experience with chatbots and route planning is stated:

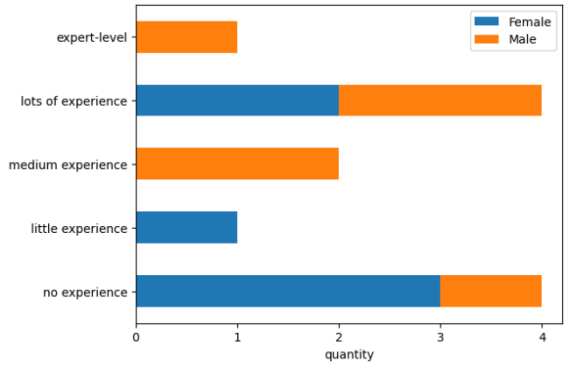


Figure 24: Chatbot experience of respondents

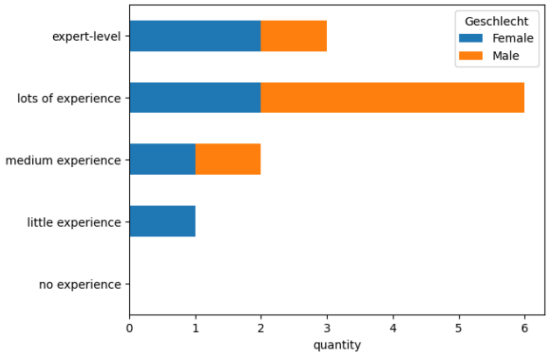


Figure 25: Route-planning experience of respondents

The chatbot experience shows two spikes at no experience and a lot of experience. The experience with route planning is distributed around lots of experience and no experience was not chosen at all, showing that experience with route-planning is more common than experience with chatbots.

8.3. Times needed for Tasks

The times for the given and the self-imposed task can be seen below:

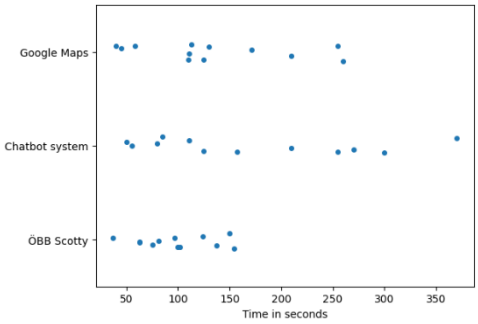


Figure 26: Times for the given task dependent on the system

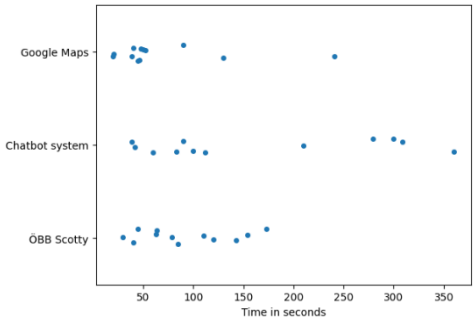


Figure 27: Times for the self-imposed task dependent on the system

The chatbot system takes on average, the longest for both tasks, while ÖBB is the fastest for the given task, and Google Maps is the fastest for the self-imposed task. The significance of the differences can be tested with the Wilcoxon test, which bases the results on ranks that are more appropriate to the non-parametric data. A significance level of 0.05 or below is considered significant, but the small sample size restricts the representativity.

- For the given task, the significance level compared to Google Maps is 0.42 and to the ÖBB 0.064. Therefore, neither differences are significant; the one with ÖBB is close and should be investigated further.
- For the self-imposed task, the significance levels are 0.034 and 0.016, respectively; therefore, the differences are significant.

8.4. System Usability Score

The following section will present the answers to the SUS questions per system. A Wilcoxon test is performed for each task, comparing the chatbot system to each of the other systems. Wilcoxon was chosen as the data is non-parametric, and there are always two systems to compare. The comparison between Google Maps and the ÖBB system is irrelevant to this thesis. Only questions with a significant result (SUS1, SUS5, SUS7, SUS12) are displayed. The other results can be found in the appendix. The significance (p) shows that the differences between the systems are significant for the respective outcome, while the test-statistic (W) shows the impact. The color is adjusted to the alternating nature of the questions, with green denoting positive answers to the questions:

Figure 28: I think that I would like to use this system frequently

The first SUS question shows a significance level of 0.06 to Google Maps. Google Maps was rated much better than the chatbot system for the question if the user would like to use the system frequently (W=9.5).

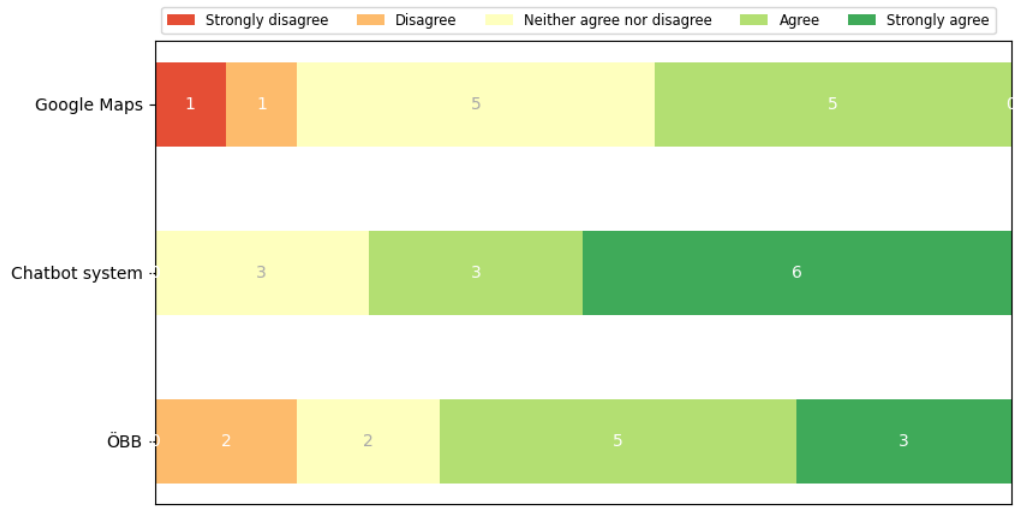


Figure 29: I found the various functions in this system were well-integrated

The fifth SUS question shows a high significance level of 0.007 to Google Maps. The chatbot system has gotten a much better rating of whether the functions were adequately integrated into the system (W=4).

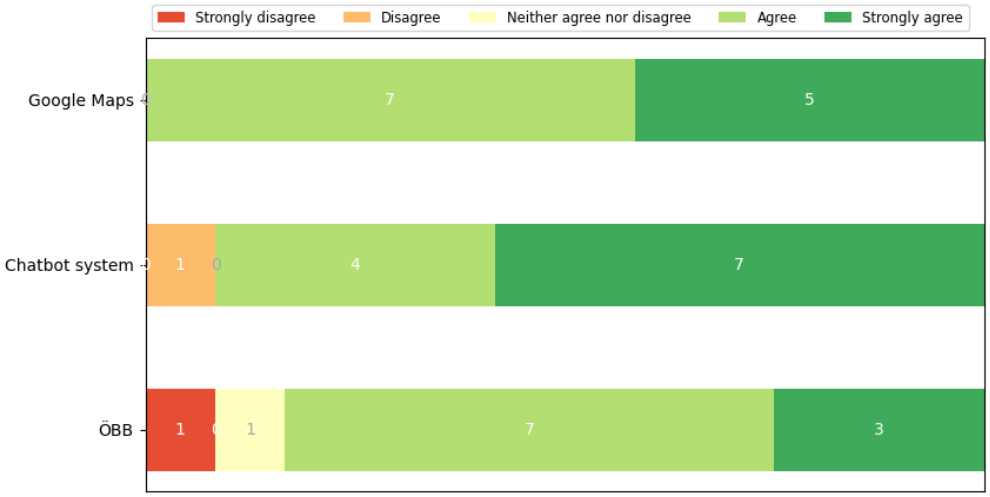


Figure 30: I would imagine that most people would learn to use this system very quickly

The seventh SUS question shows a significance level of 0.014 compared to ÖBB. Because of the unlikely test-statistic of 0, another one-sided Wilcoxon test is done. The new test shows that the Chatbot system performs much better (test-statistic of 21 and significance of 0.007). This can also be seen in Figure 30.

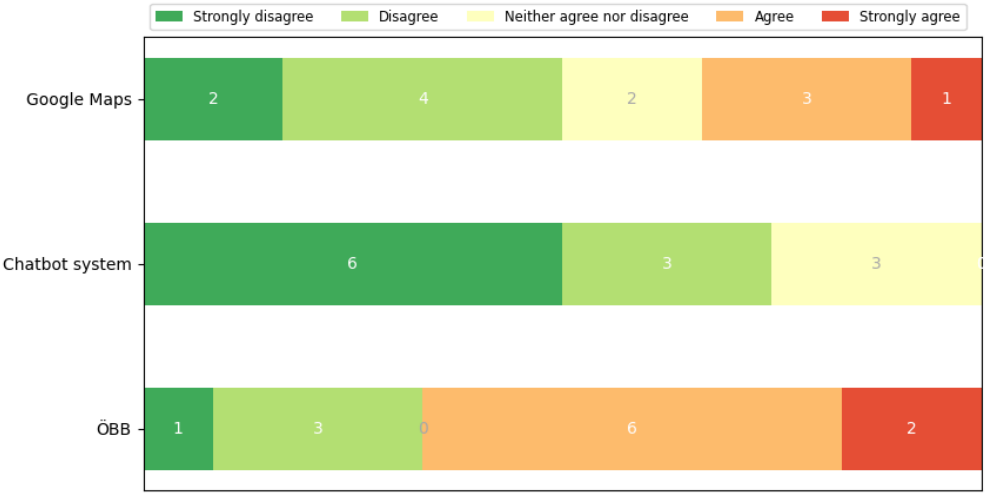


Figure 31: I would have liked to integrate my own preferences better

For SUS question twelve, the significance for the comparison to Google Maps is 0.06, and to the ÖBB system, it is 0.015. Test statistics of 5 and 6, respectively, show that users would have liked to integrate the preferences more in comparison to the chatbot system.

The SUS scores for each system can be calculated from the first ten questions by SUS's own system [47].

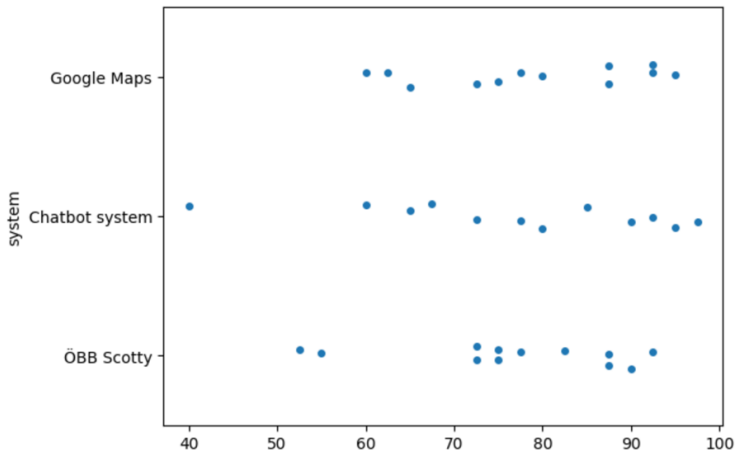


Figure 32: SUS scores of the systems

Quite noticeable is the one score of 40. The open questions showed that this respondent had major technical problems with the chatbot system such as the address not being recognized.

System	Mean	Median	Standard deviation	Max	Min
Chatbot	76.88	78.25	16.89	97.5	40
Google Maps	78.96	78.25	12.27	95	60
ÖBB	76.67	76.25	12.76	92.5	52.5

Table 10: Summary statistics for the SUS score

The Wilcoxon significance levels for the SUS Score are 0.85 and 0.9 and, therefore, far from significant. The mean is the highest for Google Maps, while the highest median is shared between Google Maps and the chatbot system. ÖBB Scotty occupies the last place for both. The standard deviation for the chatbot system is higher with the lowest min and the largest max value.

8.5. Correlations

After seeing some results and the significance of differences between the results, dependencies between demographics and experiences to the test results are calculated. The test is an experimental design with repeated measures. The results are not normally distributed and, therefore, non-parametric. Non-parametric data ranking of the scores gives more reliable results. That's why Spearman's Rho is used to calculate the correlations [46].

Correlations with a p-value below 0.05 are shown. Spearman's ρ shows how strong the correlation is and if it is positive or negative.

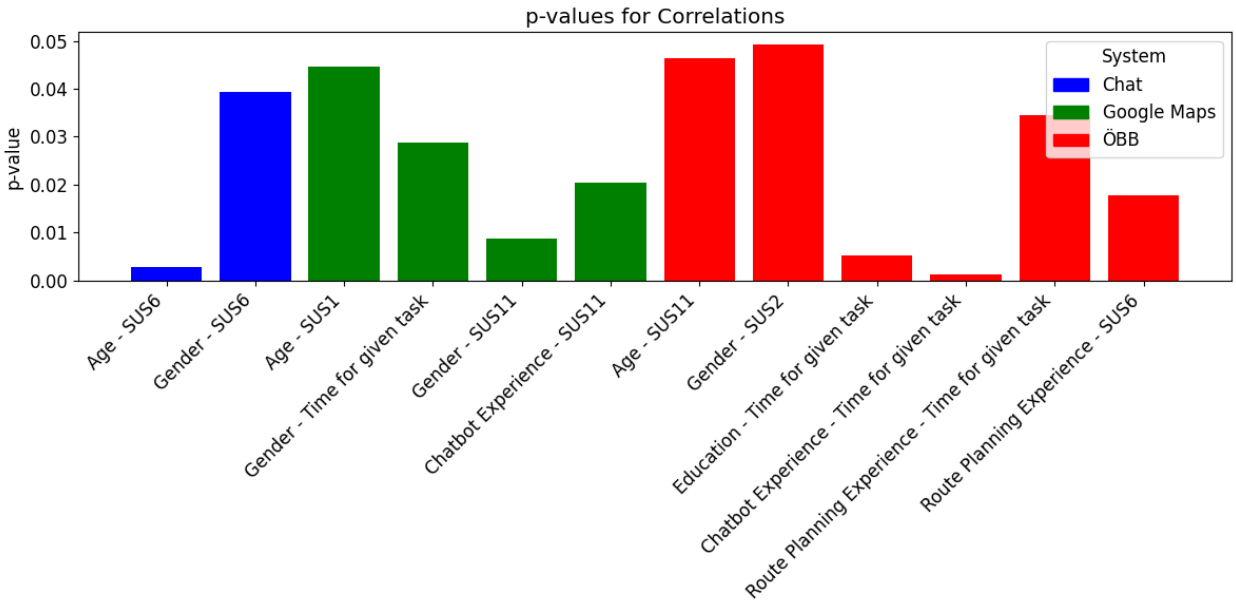


Figure 33: p-value of significant correlations

A reminder of the questions relevant for these graphics:

1. I think that I would like to use this system frequently.
2. I found the system unnecessarily complex.
6. I thought there was too much inconsistency in this system.
11. I thought that the results of the system were satisfying.

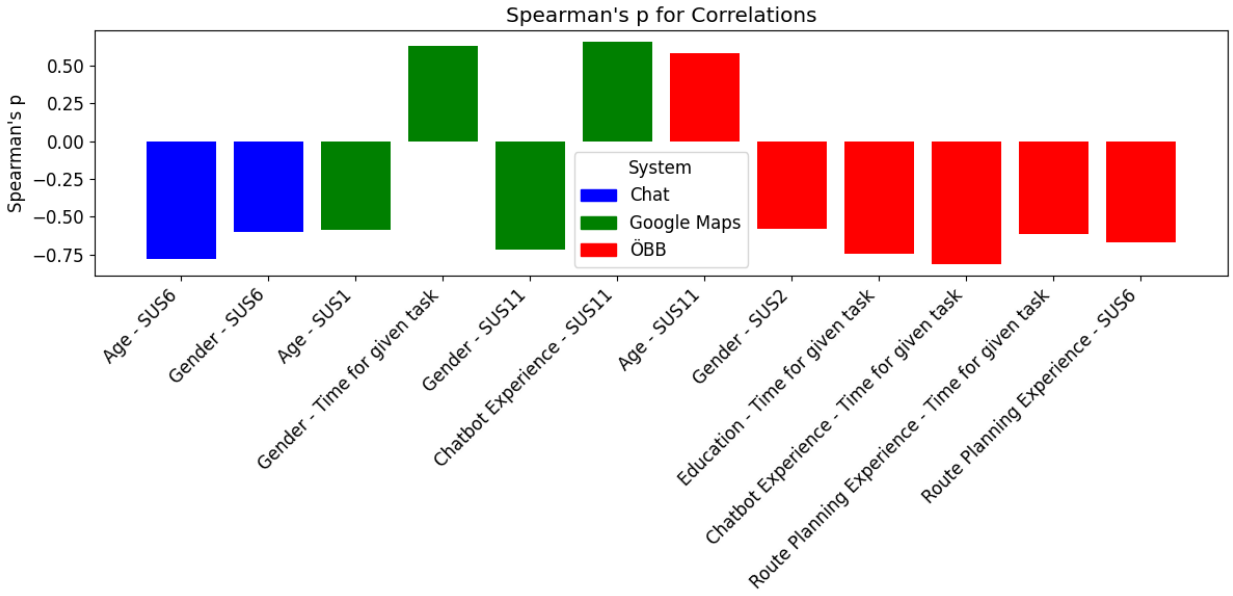


Figure 34: Spearman's ρ of significant correlations

These results are further discussed at Chapter 9.1.

8.6. Interaction Terms with two independent variables

Calculating effects based on two independent variables, like system and age, on the SUS score is more complicated. An ANOVA analysis should not be done as the data is non-parametric. To get some results and still account for the non-parametric attribute, the scores are ranked, and based on this, a mixed effects model is performed. The chatbot system is the system to which the two other systems are compared. Therefore, the system is always the first independent variable. The mixed effects model is imperfect as the underlying dataset is too small, and convergence warnings arise. The significant results are still presented and analyzed, but for verification, a test with more respondents should be conducted. For each result, the scatter plot for all three systems is given with a suggested trendline (also if the correlation is significant for only one system). Five stands for 'strongly agree' while one means 'strongly disagree'. The wording of the SUS questions can be found under the figures.

1. Age is positively correlated with the chatbot systems' SUS score, SUS 1, and SUS 11. For Google Maps, age has a negative impact on these ratings.

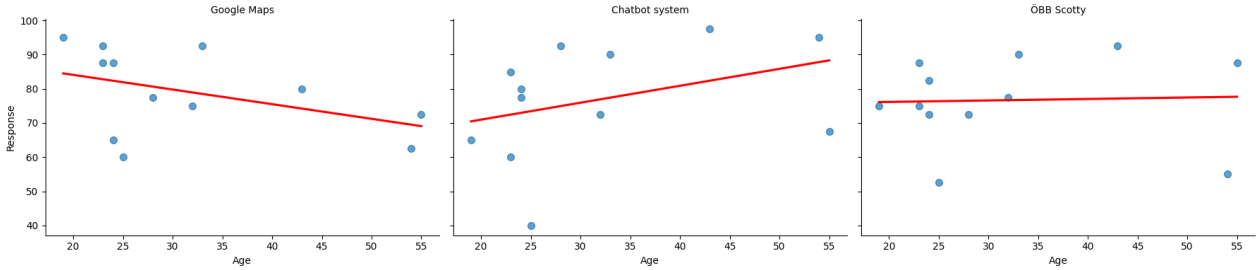


Figure 35: Overall SUS score dependent on age and system

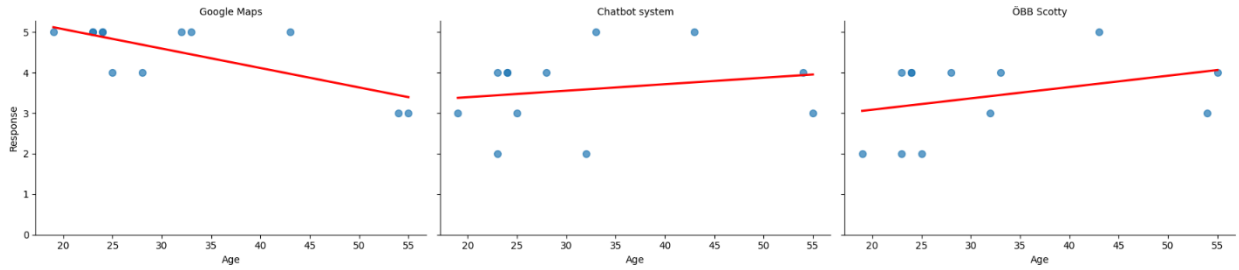


Figure 36: SUS1: "I think that I would like to use this system frequently." dependent on age and system

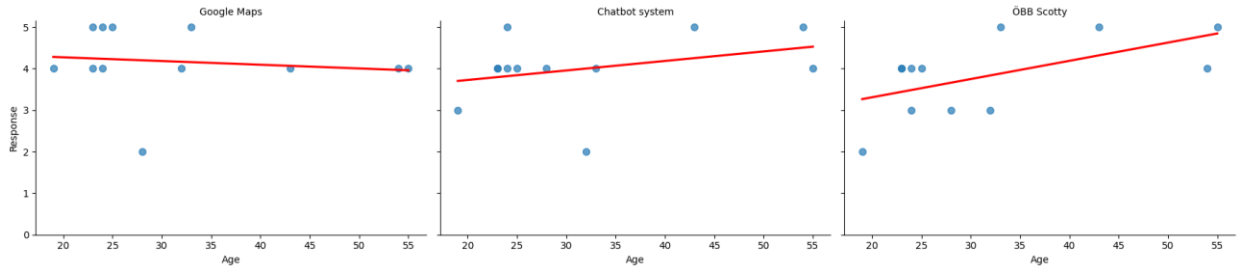


Figure 37: SUS11: "I thought that the results of the system were satisfying." dependent on age and system

2. The age is negatively correlated with SUS 2 and SUS 6 for the chatbot system, while the impact is positive for Google Maps.

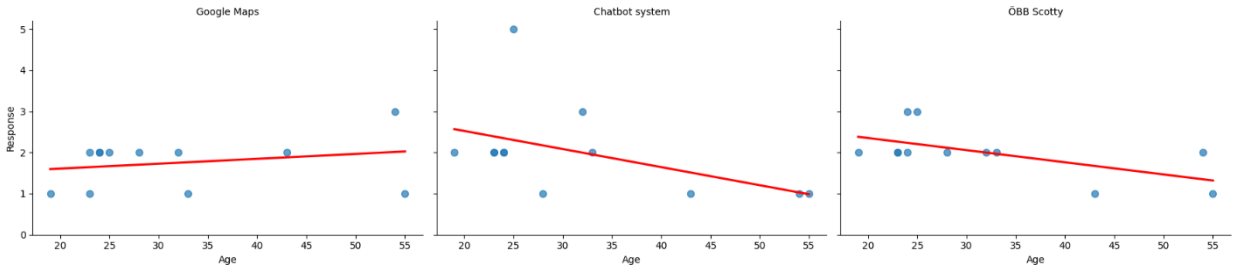


Figure 38: SUS2: "I found the system unnecessarily complex." dependent on age and system

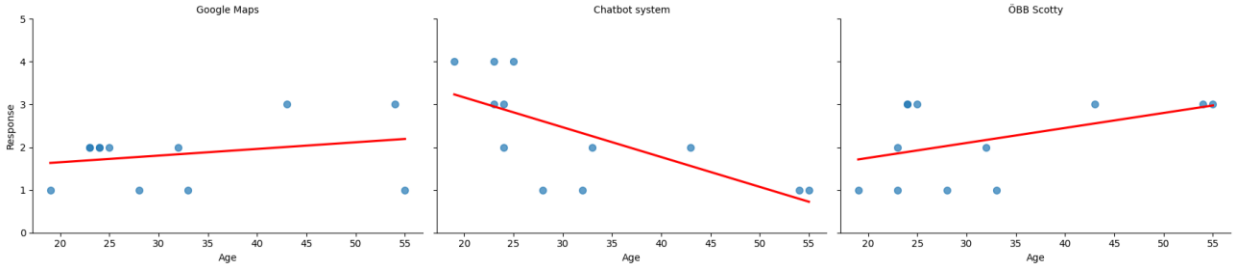


Figure 39: SUS6: "I thought there was too much inconsistency in this system." dependent on age and system

3. The answers to SUS 2 and SUS 6 are lower rated by women for the chatbot system. For Google Maps and ÖBB at SUS 6, they are higher rated than by men.

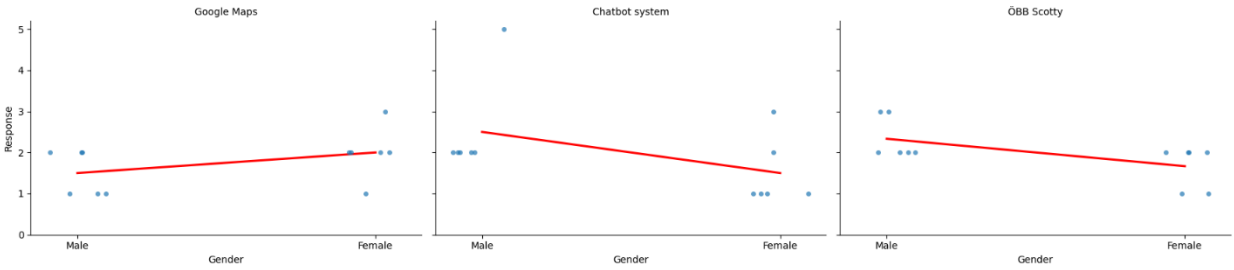


Figure 40: SUS2: "I found the system unnecessarily complex." dependent on gender and system

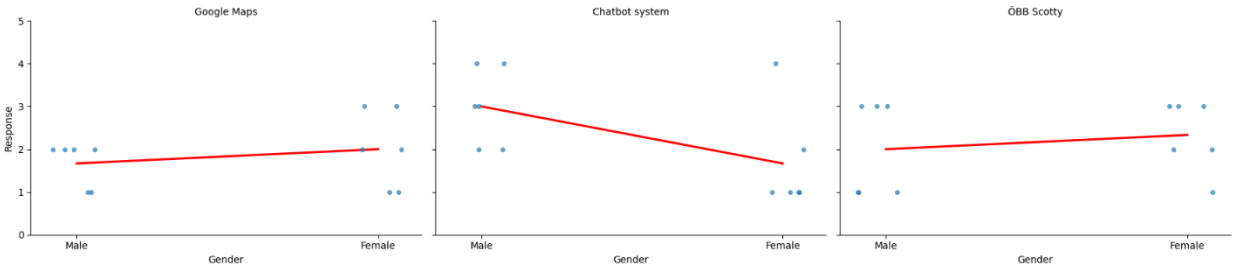


Figure 41: SUS6: "I thought there was too much inconsistency in this system." dependent on gender and system

4. SUS11 is higher or equally rated by women in the chatbot and ÖBB systems and in Google Maps it is higher rated by men.

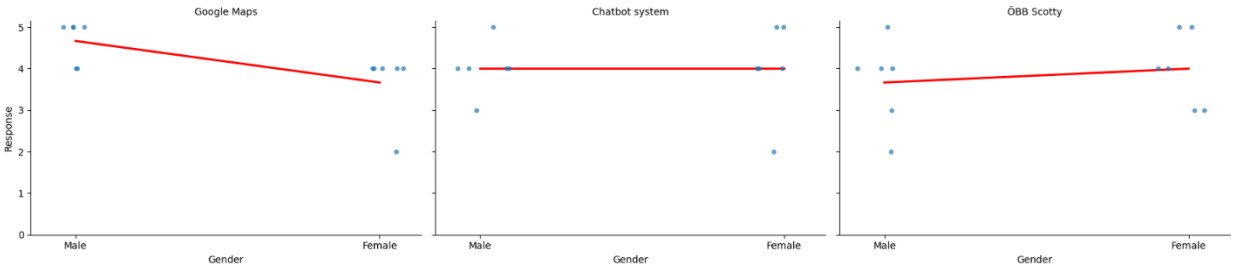


Figure 42: SUS11: "I thought that the results of the system were satisfying." dependent on gender and system

5. The scores for SUS 2 and SUS 6 are increasing with increasing chatbot experience for the chatbot system. On the contrary, for SUS 2 at Google Maps and for SUS 6 at the ÖBB, the chatbot experience has a negative impact.

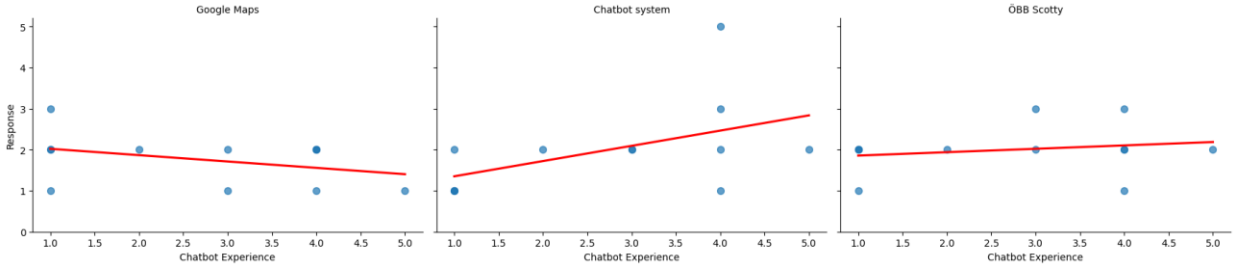


Figure 43: SUS2: "I found the system unnecessarily complex." dependent on chatbot experience and system

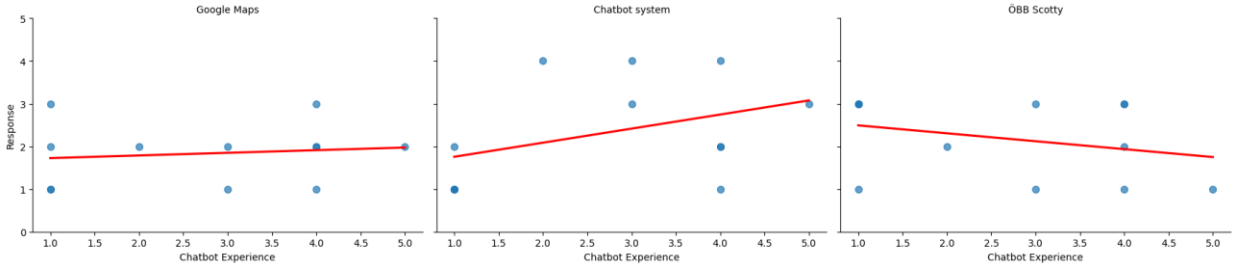


Figure 44: SUS6: "I thought there was too much inconsistency in this system." dependent on chatbot experience and system

6. The answer to SUS 11 is less good for the chatbot system with a higher chatbot experience. For Google Maps, the answers to SUS 11 are higher with more chatbot experience.

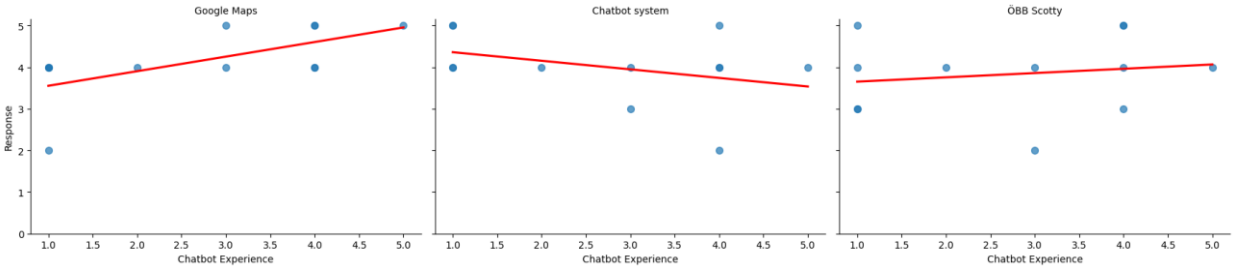


Figure 45: SUS11: "I thought that the results of the system were satisfying." dependent on chatbot experience and system

7. Higher experience with route planning has a negative impact on the answer to SUS1 for the chatbot system, while the effect is positive for Google Maps.

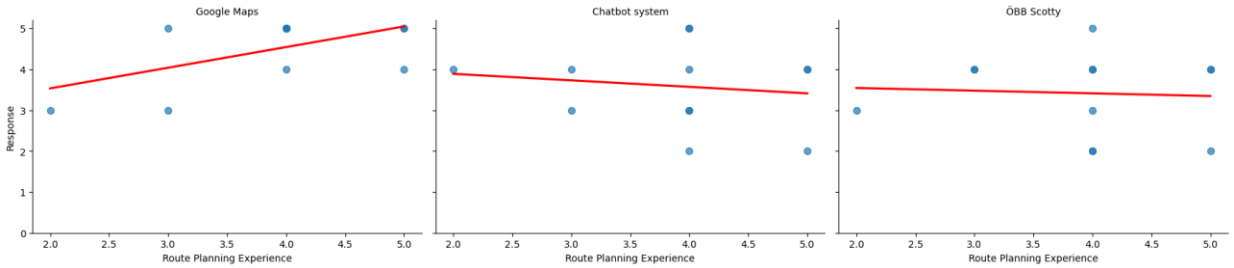


Figure 46: SUS1: "I think that I would like to use this system frequently." dependent on route-planning experience and system

8. More route-planning experience leads to higher scores for SUS 6 for the chatbot system. For both ÖBB and Google Maps, the route experience decreases the ratings of SUS 6.

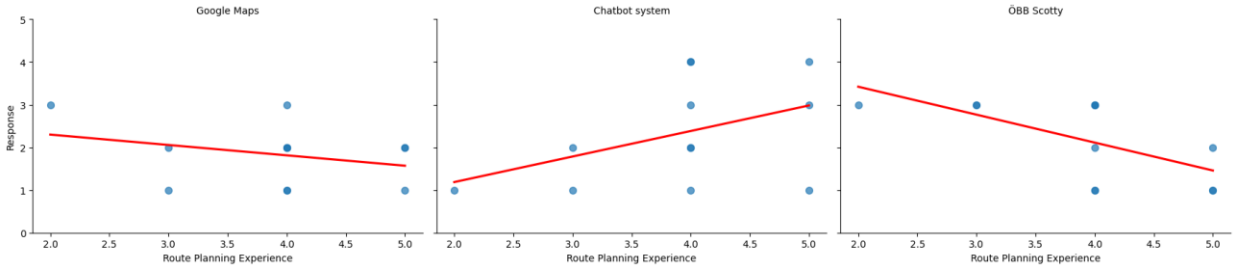


Figure 47: SUS6: "I thought there was too much inconsistency in this system." dependent on route-planning experience and system

8.7. Answers to Open Questions

A small summary of the answers to the open questions is given as follows:

1. Which feature had the most significant impact on the usability of one of the systems?
    a. Most respondents mentioned that they liked having only one input field to enter all the information into the chatbot system (seven out of twelve respondents).
    b. Some respondents mentioned familiarity with Google Maps and, therefore, better usability (three out of twelve respondents).
    c. Other mentions were filter functions of Google Maps and the ÖBB system, verification of the entered places with Google Maps, and the chatbot system's compare function.
2. Have problems or challenges occurred when using the chatbot system that have not occurred for the other systems?
    a. Four out of twelve respondents mentioned that the address was not being found or that no address was proposed for the input.
    b. Two of the twelve respondents had problems that the current date was incorrectly acquired.
    c. Further mentioned problems were unfamiliarity, unclear symbols, and specifications needed to get the right results.
3. Do you have any proposals for improving the chatbot system?
    a. Some mentioned the stated problems from the second question or the duration of the requests (five out of twelve respondents).
    b. Others gave the interface design as possible improvement (two out of twelve respondents).

c. Two out of twelve respondents requested more details for the routes.

9 Discussion

The discussion starts by answering the research question. Afterward, other aspects of usability, as well as some limitations, are touched upon.

9.1. Research question

The main research question was:

Can a **generative chatbot simplify and shorten** the process of finding travel itineraries with **equally good or better results than existing systems,** given a layperson's **demands and preferences**?

The parts are discussed individually as follows. Some graphs already shown in the results section (chapter 8) are displayed again for better readability.

1. Regarding **time**, the durations needed to finish the tasks are substantially higher for the chatbot application than for both other systems, as seen in Figure 26 and Figure 27. The time for the self-imposed task is statistically significantly different from the others. Several possible reasons were found in the answers to the open questions.
    1. The chatbot's loading times still need to be optimized.
    2. The chatbot differs from the systems users typically use, so some onboarding is needed.
    3. The functionality is not as robust as other systems. (e.g., there were problems with the current date and places not being recognized.)
2. Regarding **usability**, the chatbot's mean SUS score is between the scores of Google Maps and ÖBB, while the median shares the first place with Google Maps. Notable is the high standard deviation of the chatbot system's SUS score. The means of the system scores lie between 76.6 and 79, which is considered between good and excellent [48]. The open questions also reveal the idea's capability to reduce the input into one field, as seven out of twelve respondents positively noted this. From this, it can be derived that the chatbot system's other functionality is not as good as that of the other systems.
3. The third SUS question denotes the research question of **simplicity**: I thought the system was easy to use. Here, the chatbot system outperforms both other systems, but not statistically significant. Additional test participants are required to solidify this result.
4. Continuing with the **quality of the search results**, denoted by the eleventh question ("I thought that the results of the system were satisfying"), the chatbot system scores again between Google Maps and the ÖBB, with Google Maps retaking first place. The differences are again not statistically significant.
5. Lastly, integrating **demands and preferences** is another aspect where the chatbot system outperforms both systems according to the Wilcoxon test (significance of 0.015 to the ÖBB and 0.06 to Google Maps). Nine people disagreed, and six of them strongly, with the statement that they would have liked to integrate their own preferences, confirming the stated thesis. Figure 29 is showing this phenomenon.
6. **Accessibility for laypeople** is the last aspect of the research question. This has not been denoted by a simple question. Therefore, the differences between demographics for the different systems

are analyzed. These results must be taken with a grain of salt as the small number of participants is not representative, and correlations and trendlines might have a high error rate. The age is analyzed first.

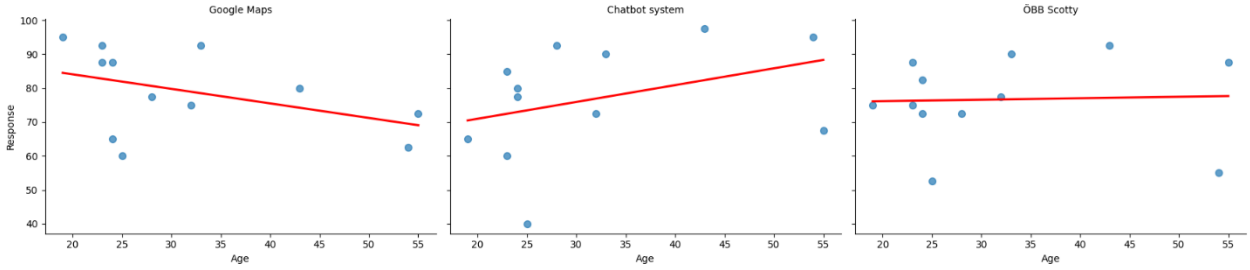


Figure 48: SUS score depending on age and system

A positive correlation between age and the SUS score for the chatbot system, a negative correlation for Google Maps, and no correlation for ÖBB can be seen in Figure 48. Some clear trends of single SUS questions are analyzed before interpreting the results.

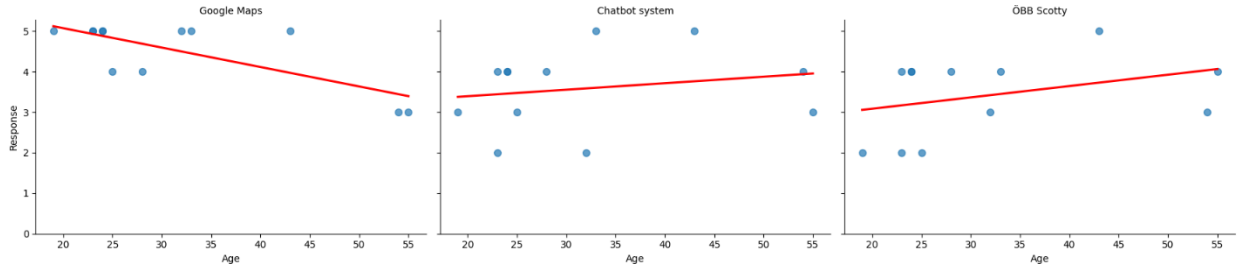


Figure 49: SUS1:"I think that I would like to use this system frequently." dependent on age and system

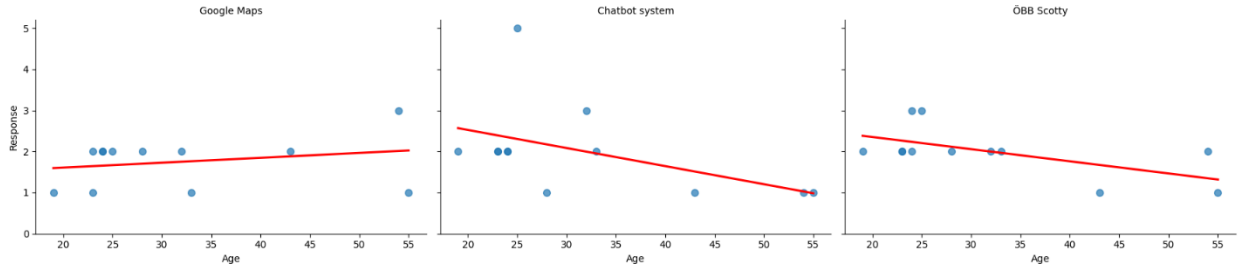


Figure 50: SUS2: " I found the system unnecessarily complex." dependent on age and system

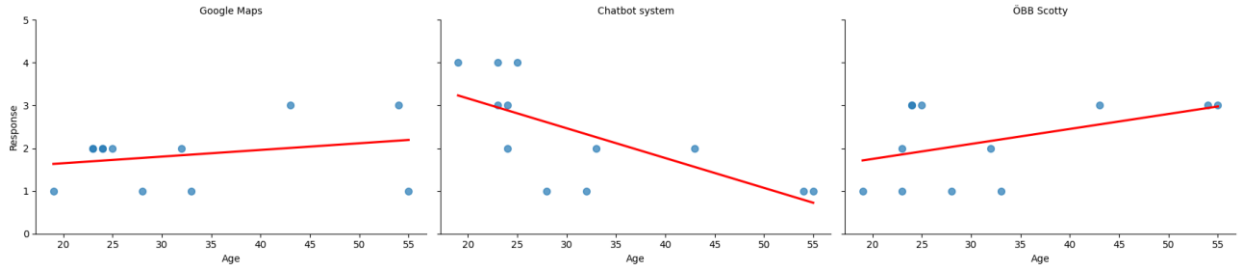


Figure 51: SUS6:"I thought there was too much inconsistency in this system." dependent on age and system

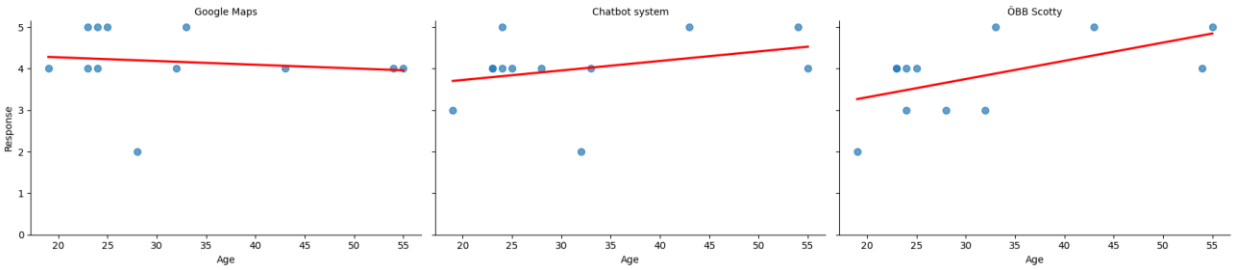


Figure 52: SUS11:"I thought that the results of the system were satisfying." dependent on age and system

With increasing age, the answers for the single SUS questions one, two, six, and eleven were less positive or even worse (already considering the alternating nature of the SUS questions) for using Google Maps compared to the chatbot system. Similar dependencies were partly seen for the independent variables of chatbot experience and route experience, especially with the inconsistency in Google Maps and the ÖBB as well, but also for complexity and satisfaction with the results. This suggests that older age groups who are usually less used to modern systems manage the chatbot much better than the other systems, especially regarding complexity, inconsistency, likelihood to use the system again, and satisfaction with the results. The fact that more route-planning experience has a significantly different effect on the likelihood of using Google Maps frequently underlines this thesis as displayed in Figure 53. The test was performed only with Germans and showed a significant Wilcoxon score. That is why the Scotty application by the Austrian railway company ÖBB can be considered a not well-known system.

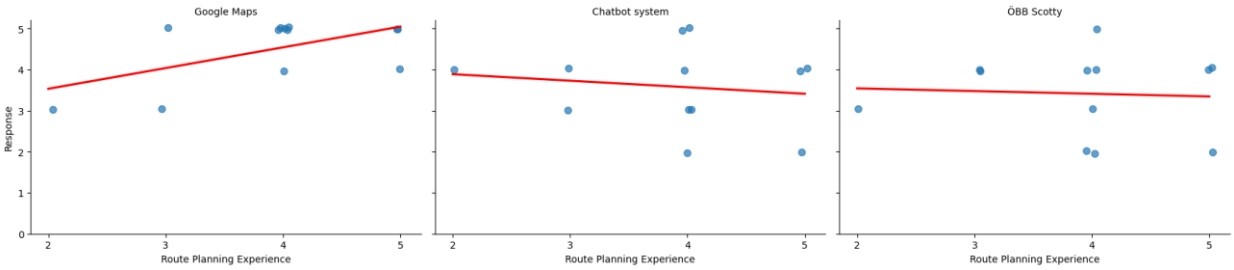


Figure 53: SUS1:" I think that I would like to use this system frequently." dependent on route-planning experience and system

Being female had nearly only significant impacts on Google Maps. The complexity and inconsistency were rated worse than for men, but still, the results were liked better in total, compared to the other systems. Other aspects must have made up for these negligences. Women also had more significant problems with the inconsistency of the ÖBB system, as shown in Figure 54. Maybe the two systems were mainly tested by men in the application development, or the small sample size resulted in unexpected biases. Also, the female respondents were, on average, older, which could impact the results.

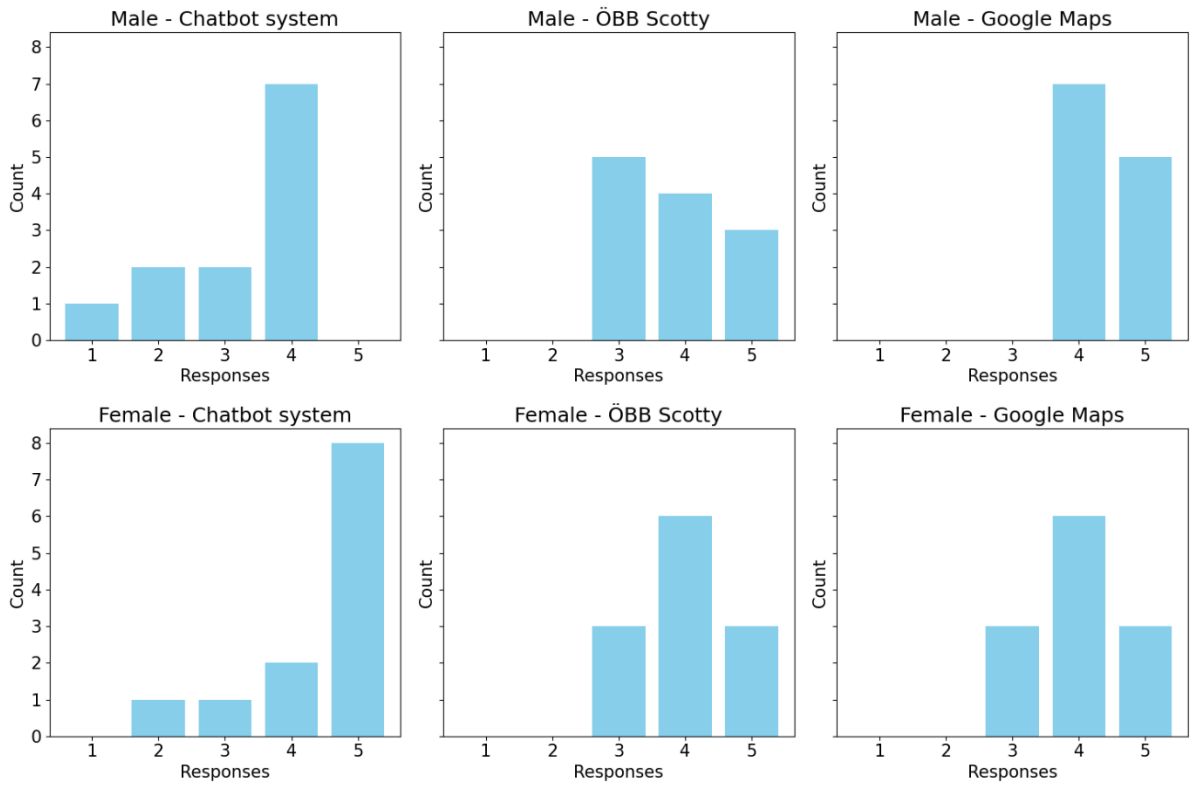


Figure 54: Rating of inconsistency dependent on gender and system (5 = consistent)

In total, the chatbot system is slower but has similar usability. It outperforms other systems in the integration of demand and preferences. A light advantage regarding simplicity can also be seen. The chatbot system performs better with less experience (novices) and higher age. This can be tied to the simplicity and accessibility. Also, most respondents saw the idea of chatting about route planning and having only one input field as positive.

9.2. Aspects of Usability

The fifth SUS question is analyzed now as it is only one of three questions (in addition to the first and twelfth questions, which have been analyzed before) with a statistically significant difference between the systems and a non-zero test statistic regarding the Wilcoxon score.

5. I found the various functions in this system were well-integrated

In this case, the chatbot system is the best-performing system. It is mostly first place (nine out of twelve times), as displayed in Figure 55. Having only one input field reduces the need to integrate many functions on their own. Seven out of twelve respondents stated the convenience of the one input field in the open questions.

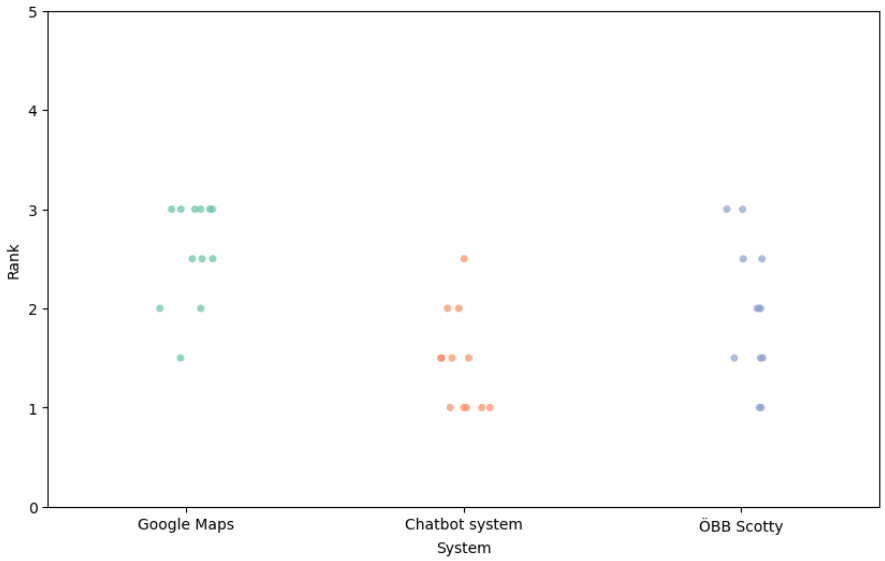


Figure 55: Ranked answer to integrating functions per system

A rank of x.5 means that a respondent gave the same score for two systems, and they share the place, therefore.

9.3. Limitations

One limitation is the number of respondents. For the SUS score, the number of 12 respondents can be seen as representative, as described earlier. Regarding dependencies on demographics, the results with more respondents would have been more representative. Digging deeper into the reason for the results and differences could be supported by an additional think-aloud protocol, where users verbalize their thoughts while performing tasks [53].

A larger number of comparable systems could be interesting as well. For example, systems from national railway companies could be interesting comparisons. The German DB Navigator was mentioned in many interviews.

The current state of LLMs is another limitation. Flawlees Interpretation of the input text, maximum size of input data (such as documentation for the API) and well adapted answers to the user remain challenges with todays systems. With the current rapid development these limitations may solve themselves.

More time to develop and test the chatbot system could have made a big difference in the results. The durations for the tasks may have decreased, and with less waiting and additionally more robust functionalities, the question scores may have increased as well.

10 Conclusion

In this thesis, a new route-planning application was programmed to compare to existing systems. The new application's main difference is the addition of a chatbot and, therefore, reducing the number of input fields to one. To develop a comparable application, six potential users with different background and experience have been interviewed for requirements. A prototype has been derived from these requirements with the help of 17 people, consisting of research staff and potential users. The prototype was

implemented using JavaScript with the open-source tool OpenTripPlaner 2 as the data basis. The new system has been compared to Google Maps and the ÖBB system with the help of twelve test users. Durations for tasks, the SUS questions, two additional questions in SUS style, and three open questions have been asked to the users for each system.

10.1. Key findings

For the key findings, the last additional research question is answered: What are the quantitative and qualitative advantages and disadvantages of adding a chatbot to itinerary planning software?

A quantitative disadvantage at this stage of development is the time needed to complete the tasks. According to the test results, qualitative advantages are better simplicity, improved consistency, and much better integration of demands and preferences. Also, accessibility can be positively highlighted as the overall score improves with age and the experience of fewer route plannings. This can also be seen as people used to Google Maps would like to use Google Maps more frequently. The answers for the novel chatbot and ÖBB systems show no correlation with route-planning experience.

The overall SUS score of the chatbot system performs similarly to that of the other systems. This can also be seen as a success as tremendous resources have been poured into developing the other systems, which millions of people use daily.

10.2. Future Development of the application

More time can be spent developing the chatbot system to reach and exceed the functionality and robustness levels of the compared systems. First, all disruptions and errors stated in the answers to the open questions need to be tackled. The waiting time for each request should be reduced. This way, the current functionality is not touched, but the usability increases.

The requirements not yet fulfilled can be implemented to improve functionality as these represent the user's desires for such applications. Some examples are:

1. Possibility to change the sorting of the itineraries
2. Make the reason for the recommendation visible
3. Quick and easy booking
4. Show included services

Solutions that are not part of other systems are also mentioned here because potential users have been asked about problems and desires regarding currently used systems. Some examples are:

1. Show the effects of changing filter options
2. Good solutions slightly out of the selected filters are shown and marked
3. Show as many Pareto optimal solutions as possible

Finally, the application scope can be extended. Adding countries in addition to Austria, as well as flight data, makes comparing routes more challenging. The feature of saving and comparing routes has not been used much by the test users yet and could be another positively salient feature of the chatbot application in the bigger scope.

Therefore, this thesis has started the research on introducing conversational interfaces into route-planning applications while gathering the first interesting insights and paving the path for more research.

VII. Bibliography

[1] S. Carmien, M. Dawe, G. Fischer, A. Gorman, A. Kintsch, and J. F. Sullivan, 'Socio-technical environments supporting people with cognitive disabilities using public transportation', *ACM Trans. Comput.-Hum. Interact.*, vol. 12, no. 2, pp. 233–262, Jun. 2005, doi: 10.1145/1067860.1067865.

[2] Lino, Pedro Manuel Dias Braga, 'Travel Booking Chatbot'. Jun. 2018.

[3] D. Adiwardana *et al.*, 'Towards a Human-like Open-Domain Chatbot', 2020, *arXiv*. doi: 10.48550/ARXIV.2001.09977.

[4] E. Hermann and S. Puntoni, 'Artificial intelligence and consumer behavior: From predictive to generative AI', *J. Bus. Res.*, vol. 180, p. 114720, Jul. 2024, doi: 10.1016/j.jbusres.2024.114720.

[5] K. Vredenburg, J.-Y. Mao, P. W. Smith, and T. Carey, 'A survey of user-centered design practice', in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, Minneapolis Minnesota USA: ACM, Apr. 2002, pp. 471–478. doi: 10.1145/503376.503460.

[6] M. Sedlmair, M. Meyer, and T. Munzner, 'Design Study Methodology: Reflections from the Trenches and the Stacks', *IEEE Trans. Vis. Comput. Graph.*, vol. 18, no. 12, pp. 2431–2440, Dec. 2012, doi: 10.1109/TVCG.2012.213.

[7] B. L. Kovitz, *Practical software requirements: a manual of content and style*. Greenwich, Conn: Manning, 1999.

[8] H. Sharp, *Interaction design 5e*. Indianapolis, IN: John Wiley and Sons, 2019.

[9] A. Hevner and S. Chatterjee, *Design Research in Information Systems: Theory and Practice*, vol. 22. in Integrated Series in Information Systems, vol. 22. Boston, MA: Springer US, 2010. doi: 10.1007/978-1-4419-5653-8.

[10] W. V. Siricharoen, 'Experiencing User-Centered Design (UCD) Practice (Case Study: Interactive Route Navigation Map of Bangkok Underground and Sky Train)', in *Human-Computer Interaction*, P. Forbrig, F. Paternó, and A. Mark Pejtersen, Eds., Berlin, Heidelberg: Springer, 2010, pp. 70–79. doi: 10.1007/978-3-642-15231-3_8.

[11] N. Vittayaphorn *et al.*, 'Design and Development of a User-Centered Mobile Application for Intermodal Public Transit in Bangkok: A Design Thinking Approach', *Infocommunications J.*, vol. 15, no. Special Issue, pp. 41–52, 2023, doi: 10.36244/ICJ.2023.SI-IODCR.7.

[12] S. Beul-Leusmann, E.-M. Jakobs, and M. Ziefle, 'User-centered design of passenger information systems', in *IEEE International Professonal Communication 2013 Conference*, Jul. 2013, pp. 1–8. doi: 10.1109/IPCC.2013.6623931.

[13] D. Stone, *User Interface Design and Evaluation*. in Interactive Technologies Series. San Diego: Elsevier Science & Technology, 2005.

[14] C. R. Lengkong, C. Mayas, H. Krömker, and M. Hirth, 'The Development of Human-Centered Design in Public Transportation: A Literature Review', in *HCI in Mobility, Transport, and Automotive Systems*, vol. 14733, H. Krömker, Ed., in Lecture Notes in Computer Science, vol. 14733. , Cham: Springer Nature Switzerland, 2024, pp. 40–62. doi: 10.1007/978-3-031-60480-5_3.

[15] H. Detjen, S. Schneegass, S. Geisler, A. Kun, and V. Sundar, 'An Emergent Design Framework for Accessible and Inclusive Future Mobility', in *Proceedings of the 14th International Conference on Automotive User Interfaces and Interactive Vehicular Applications*, Seoul Republic of Korea: ACM, Sep. 2022, pp. 1–12. doi: 10.1145/3543174.3546087.

[16] J.-E. Kim, M. Bessho, N. Koshizuka, and K. Sakamura, 'Enhancing Public Transit Accessibility for the Visually Impaired Using IoT and Open Data Infrastructures', in *Proceedings of the The First International Conference on IoT in Urban Space*, Rome, Italy: ICST, 2014. doi: 10.4108/icst.urbiot.2014.257263.

[17] L. Frizziero, G. Donnici, G. Galiè, G. Pala, M. Pilla, and E. Zamagna, 'QFD and SDE Methods Applied to Autonomous Minibus Redesign and an Innovative Mobile Charging System (MBS)', *Inventions*, vol. 8, no. 1, p. 1, Dec. 2022, doi: 10.3390/inventions8010001.

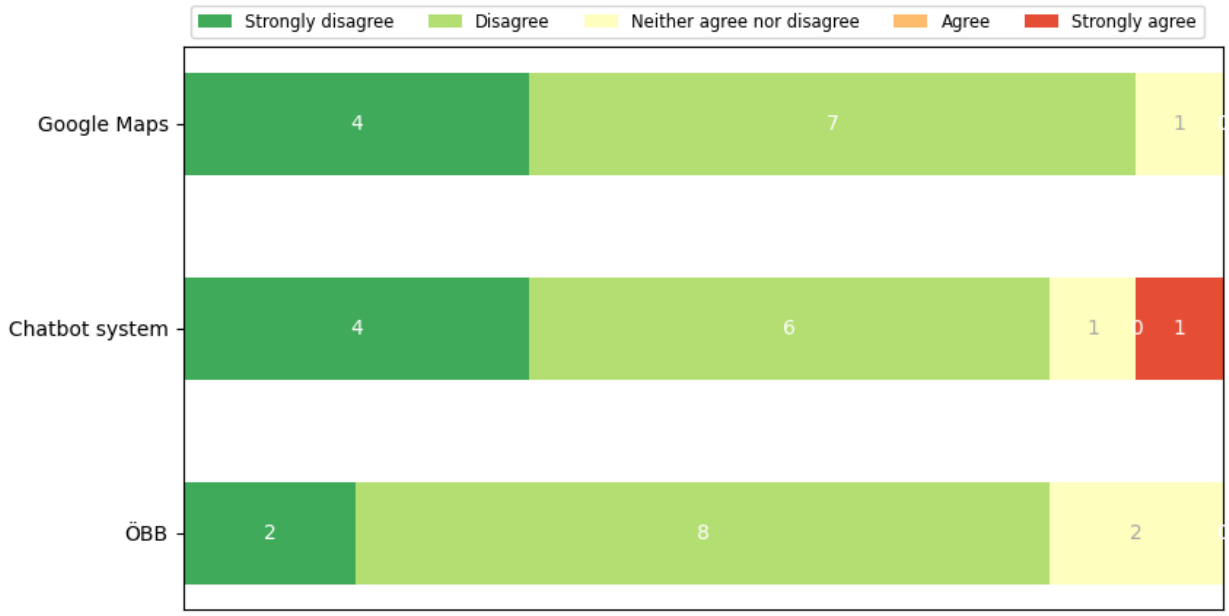
[18] H. Cornet, S. Stadler, P. Kong, G. Marinkovic, F. Frenkler, and P. M. Sathikh, 'User-centred design of autonomous mobility for public transportation in Singapore', *Transp. Res. Procedia*, vol. 41, pp. 191–203, Jan. 2019, doi: 10.1016/j.trpro.2019.09.038.
[19] M. Ekşioğlu, 'User Experience Design of a Prototype Kiosk: A Case for the İstanbul Public Transportation System', *Int. J. Human–Computer Interact.*, vol. 32, no. 10, pp. 802–813, Oct. 2016, doi: 10.1080/10447318.2016.1199179.
[20] Y. Kang, S. Park, K. Kim, and H. Kim, 'Development of interactive map-based ui for subway ticketing kiosk system', in *Proceedings of IADIS International Conference Interfaces and Human Computer Interaction*, 2010, pp. 319–322.
[21] B. A. Shawar and E. Atwell, 'Chatbots: are they really useful?', *J. Lang. Technol. Comput. Linguist.*, vol. 22, no. 1, pp. 29–49, 2007.
[22] B. Jordan, L. Koesten, and T. Möller, 'Chatting About Data - Interacting with Voice Interfaces to Engage with Election Panel Data', in *Proceedings of the 5th International Conference on Conversational User Interfaces*, Eindhoven Netherlands: ACM, Jul. 2023, pp. 1–12. doi: 10.1145/3571884.3597126.
[23] N. Castaldo, F. Daniel, M. Matera, and V. Zaccaria, 'Conversational Data Exploration', in *Web Engineering*, Springer, Cham, 2019, pp. 490–497. doi: 10.1007/978-3-030-19274-7_34.
[24] S. Vakulenko and V. Savenkov, 'TableQA: Question Answering on Tabular Data', 2017, *arXiv*. doi: 10.48550/ARXIV.1705.06504.
[25] Y. Yao, J. Duan, K. Xu, Y. Cai, Z. Sun, and Y. Zhang, 'A Survey on Large Language Model (LLM) Security and Privacy: The Good, The Bad, and The Ugly', *High-Confid. Comput.*, vol. 4, no. 2, p. 100211, Jun. 2024, doi: 10.1016/j.hcc.2024.100211.
[26] A. Følstad and P. B. Brandtzæg, 'Chatbots and the new world of HCI', *Interactions*, vol. 24, no. 4, pp. 38–42, Jun. 2017, doi: 10.1145/3085558.
[27] J. Li *et al.*, 'Can LLM Already Serve as A Database Interface? A BIg Bench for Large-Scale Database Grounded Text-to-SQLs', 2023.
[28] P. Ni, R. Okhrati, S. Guan, and V. Chang, 'Knowledge Graph and Deep Learning-based Text-to-GraphQL Model for Intelligent Medical Consultation Chatbot', *Inf. Syst. Front.*, vol. 26, no. 1, pp. 137–156, Feb. 2024, doi: 10.1007/s10796-022-10295-0.
[29] N. Burkart and M. F. Huber, 'A Survey on the Explainability of Supervised Machine Learning', *J. Artif. Intell. Res.*, vol. 70, pp. 245–317, Jan. 2021, doi: 10.1613/jair.1.12228.
[30] C. Helfferich, 'Leitfaden- und Experteninterviews', in *Handbuch Methoden der empirischen Sozialforschung*, N. Baur and J. Blasius, Eds., Wiesbaden: Springer Fachmedien Wiesbaden, 2022, pp. 875–892. doi: 10.1007/978-3-658-37985-8_55.
[31] W. C. Adams, 'Conducting Semi-Structured Interviews', in *Handbook of Practical Program Evaluation*, 1st ed., K. E. Newcomer, H. P. Hatry, and J. S. Wholey, Eds., Wiley, 2015, pp. 492–505. doi: 10.1002/9781119171386.ch19.
[32] D. Cohen, B. Crabtree, 'Semi-structured Interviews'. Jul. 01, 2006.
[33] M. Cachia and L. Millward, 'The telephone medium and semi-structured interviews: a complementary fit', *Qual. Res. Organ. Manag. Int. J.*, vol. 6, no. 3, pp. 265–277, Jan. 2011, doi: 10.1108/17465641111188420.
[34] C. Meske, E. Bunde, and J. F. Ehmke, 'Improving Customers' Decision Making on Blackboxed Multimodal Mobility Platforms – A Design Science Approach', 2020.
[35] M. V. Corazza and G. Carassiti, 'Investigating Maturity Requirements to Operate Mobility as a Service: The Rome Case', *Sustainability*, vol. 13, no. 15, p. 8367, Jul. 2021, doi: 10.3390/su13158367.
[36] P. Jittrapirom, V. Marchau, R. Van Der Heijden, and H. Meurs, 'Future implementation of mobility as a service (MaaS): Results of an international Delphi study', *Travel Behav. Soc.*, vol. 21, pp. 281–294, Oct. 2020, doi: 10.1016/j.tbs.2018.12.004.
[37] J. Arnowitz, M. Arent, and N. Berger, *Effective prototyping for software makers*. in The Morgan Kaufmann series in interactive technologies. Amsterdam Boston Heidelberg: Elsevier Morgan Kaufmann, 2007.

[38] ‘OpenTripPlanner 2’. Accessed: Jul. 10, 2024. [Online]. Available: https://docs.opentripplanner.org/en/latest/
[39] Anthropic, ‘Introducing the next generation of Claude’, anthropic.com. Accessed: Jul. 02, 2024. [Online]. Available: https://www.anthropic.com/news/claude-3-family
[40] Antrophic, ‘Claude is now available in the EU \ Anthropic’, anthropic.com. Accessed: Jul. 02, 2024. [Online]. Available: https://www.anthropic.com/news/claude-europe
[41] Antrophic, ‘Pricing’, anthropic.com. Accessed: Jul. 02, 2024. [Online]. Available: https://www.anthropic.com/pricing#anthropic-api
[42] OpenAI, ‘Pricing OpenAI’, openai.com. Accessed: Jul. 02, 2024. [Online]. Available: https://openai.com/api/pricing/
[43] ‘Vue.js - The Progressive JavaScript Framework | Vue.js’. Accessed: Jul. 10, 2024. [Online]. Available: https://vuejs.org/
[44] ‘D3 by Observable | The JavaScript library for bespoke data visualization’. Accessed: Jul. 10, 2024. [Online]. Available: https://d3js.org/
[45] ‘Tailwind CSS - Rapidly build modern websites without ever leaving your HTML.’ Accessed: Jul. 10, 2024. [Online]. Available: https://tailwindcss.com/
[46] A. P. Field and G. Hole, *How to design and report experiments*. London ; Thousand Oaks, Calif: Sage publications Ltd, 2003.
[47] J. Brooke, ‘SUS - A quick and dirty usability scale’, 1996.
[48] A. Bangor, P. T. Kortum, and J. T. Miller, ‘An Empirical Evaluation of the System Usability Scale’, *Int. J. Hum.-Comput. Interact.*, vol. 24, no. 6, pp. 574–594, Jul. 2008, doi: 10.1080/10447310802205776.
[49] T. S. Tullis and J. N. Stetson, ‘A Comparison of Questionnaires for Assessing Website Usability’, 2004.
[50] J. Sauro, ‘10 Things to Know About the System Usability Scale (SUS)’, MeasuringU. Accessed: Nov. 18, 2024. [Online]. Available: https://measuringu.com/10-things-sus/
[51] B. Rummel, ‘System Usability Scale (Translated into German)’, Apr. 2013.
[52] T. J. Brix, A. Janssen, M. Storck, and J. Varghese, ‘Comparison of German Translations of the System Usability Scale – Which to Take?’, in *German Medical Data Sciences 2023 — Science. Close to People.*, R. Röhrig, N. Grabe, M. Haag, U. Hübner, U. Sax, C. Oliver Schmidt, M. Sedlmayr, and A. Zapf, Eds., IOS Press, 2023. doi: 10.3233/SHTI230699.
[53] Jääskeläinen, Ritta, ‘Think-aloud protocol’, in *Handbook of Translation Studies*, Y. Gambier and L. van Doorslaer, Eds., Amsterdam: John Benjamins Publishing Company, 2010. doi: 10.1075/hts.1.

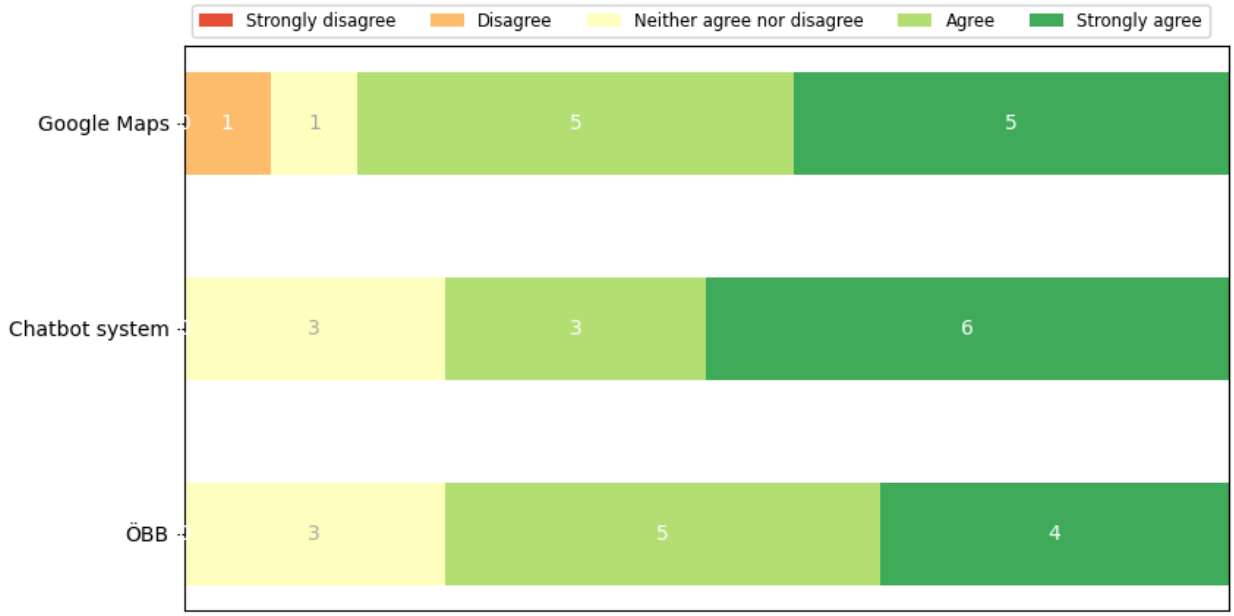
VIII. Appendix

Answers to the SUS questions with no significant results according to the Wilcoxon test:

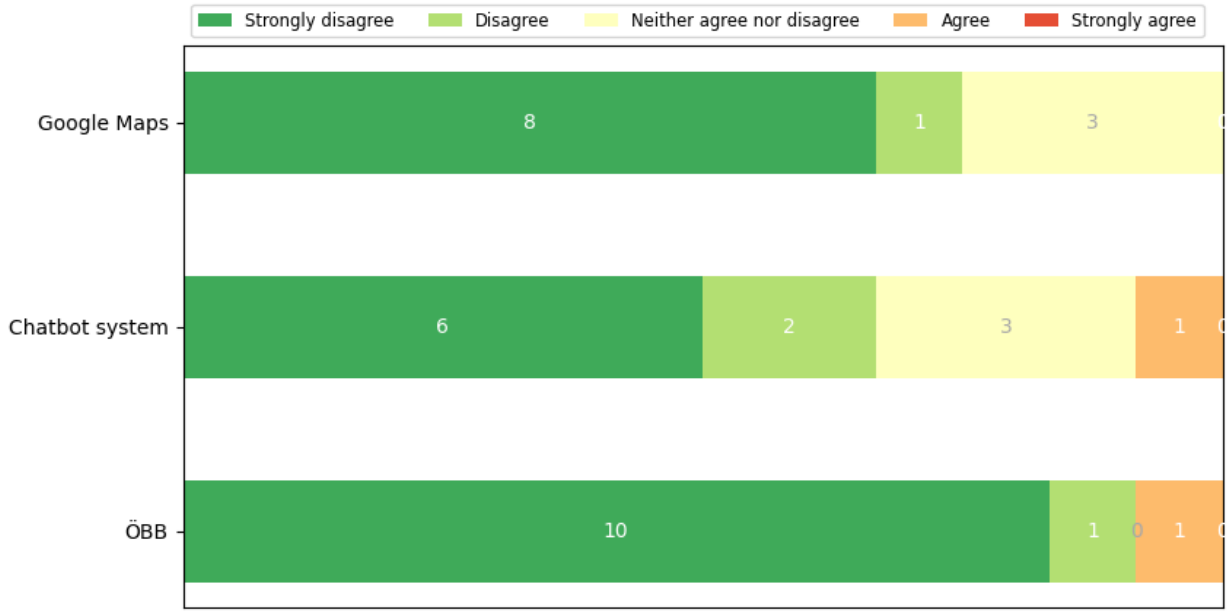
I found the system unnecessarily complex:



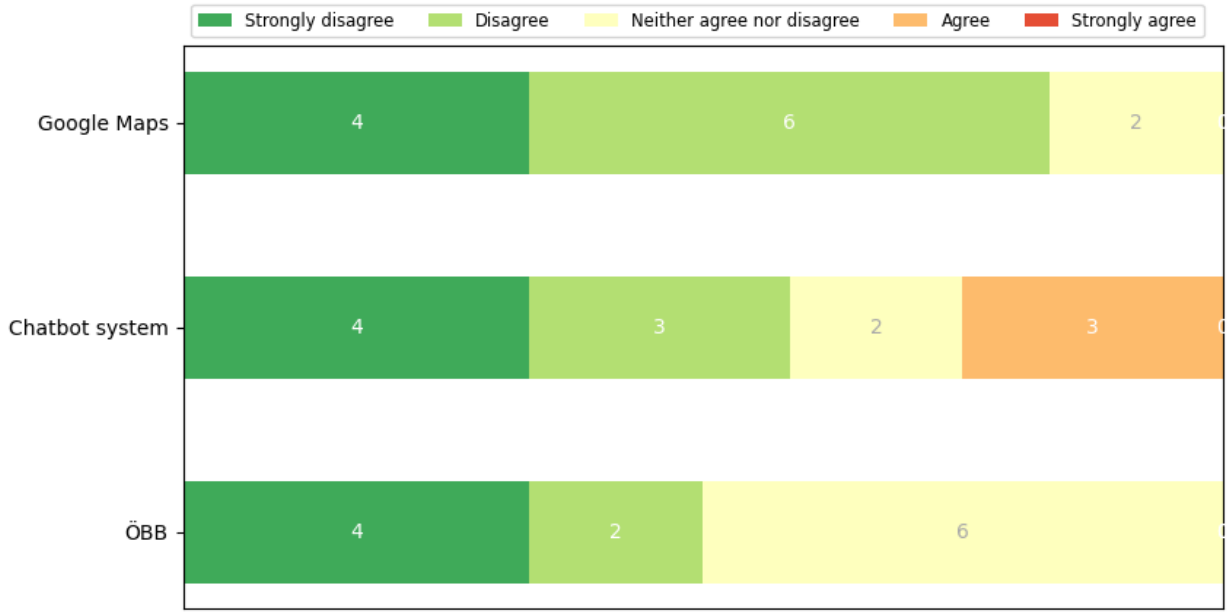
I thought the system was easy to use:



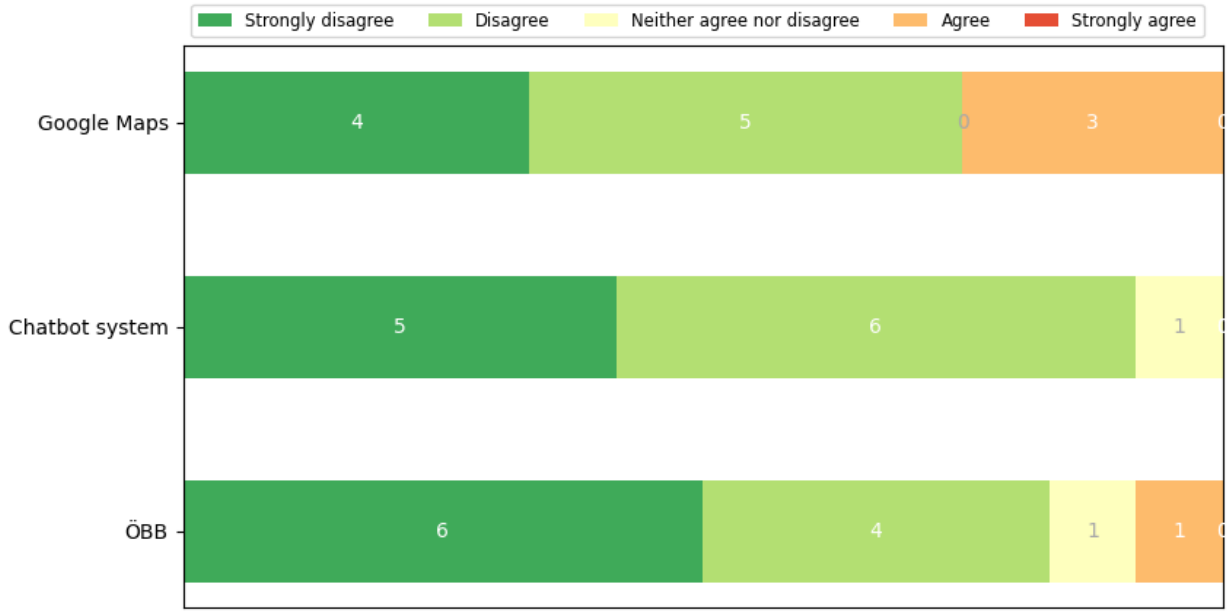
I think that I would need the support of a technical person to be able to use this system:



I thought there was too much inconsistency in this system:

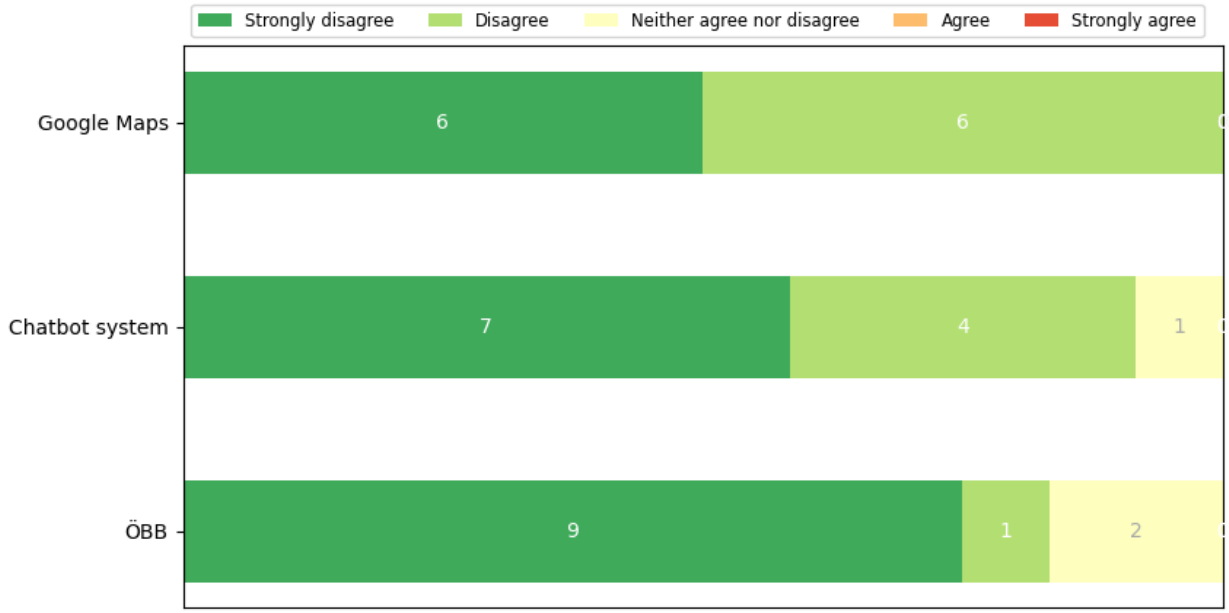


I found the system very cumbersome to use:



I felt very confident using the system:

I needed to learn a lot of things before I could get going with this system:



I thought that the results of the system were satisfying: