FACULTY
OF INFORMATION
TECHNOLOGY
CTU IN PRAGUE

Assignment of master's thesis

Title:	WhereIS - user interface design
Student:	Bc. Illia Brylov
Supervisor:	Ing. David Bernhauer, Ph.D.
Study program:	Informatics
Branch / specialization:	Web Engineering
Department:	Department of Software Engineering
Validity:	until the end of summer semester 2024/2025

Instructions

As part of T. Heger's bachelor thesis, a prototype of the WhereIS application was created to make it easier to navigate not only in the school buildings but also help in other aspects of study. The following master thesis builds on this prototype, and its primary goal is to complete the application to a state that allows real use. The thesis is part of the team project WhereIS.

1. Collect and analyze user requirements.
2. Search for similar systems, focusing on the user interface.
3. Analyze an existing prototype and identify problems.
4. Based on the analysis, develop and document a new user interface design for the web application.
5. Implement the user interface in the React framework, which will be connected to the REST API.
6. Deploy the resulting user interface.
7. Perform usability testing.
8. Evaluate the results of the design, implementation, and testing.

Electronically approved by Ing. Jaroslav Kuchař, Ph.D. on 31 January 2024 in Prague.

FACULTY
OF INFORMATION
TECHNOLOGY
CTU IN PRAGUE

Master's thesis

WhereIS - user interface design

Bc. Illia Brylov

Department of Software Engineering
Supervisor: Ing. David Bernhauer, Ph.D.

May 9, 2024

Acknowledgements

I would like to thank first and foremost Ing. David Bernhauer, Ph.D., for his valuable advice not only on my work but also on life in general. Furthermore, my gratitude goes to my partner Bc. T. Heger, who has been not only a great colleague during work on this project but also provided me with psychological support in my personal life. I must not forget to thank my father for the invaluable life chance he has given me, and finally, the entire faculty for the opportunity to meet amazing people and gain a new perspective on things.

Declaration

In Prague on May 9, 2024 …………………

Citation of this thesis

Brylov, Illia. *WhereIS - user interface design.* Master's thesis. Czech Technical University in Prague, Faculty of Information Technology, 2024.

Abstrakt

Tato diplomová práce popisuje návrh, implementaci a hodnocení nového uživatelského rozhraní pro aplikaci WhereIS, která si klade za cíl zlepšit navigaci a vyhledáváni napříč různými systémy na Fakultě informačních technologií Českého vysokého učení technického v Praze. Využívající metodiku User-Centered designu, fáze návrhu zahrnovala analýzu stávajících systémů a sběr požadavků uživatelů prostřednictvím dotazníků, na základě prototypu vyvinutého Bc. T. Hegerem v rámci jeho bakalářské práce. Rozhraní je implementováno s využitím knihovny React a ověřeno prostřednictvím heuristické analýzy a uživatelského testování s koncovými uživateli.

Klíčová slova návrh uživatelského rozhraní, User-Centered Design, univerzitní vyhledávací platforma, informační systém, WhereIS, React

Abstract

This master thesis details the design, implementation, and evaluation of an new user interface for the WhereIS application, aiming to enhance navigation and search functionalities across various systems at the Faculty of Information Technologies of the Czech Technical University in Prague. Employing a User-Centered Design approach, the design phase involved analyzing existing systems and gathering user requirements through questionnaires, building upon the prototype developed by Bc. T. Heger in his bachelor thesis. The interface is implemented using the React library and evaluated through heuristic analysis and usability testing with end users.

Keywords user interface design, User-Centered Design, university search platform, information system, WhereIS, React

Contents

List of Figures

List of Tables

List of Listings

Introduction

The Czech Technical University (CTU) and its Faculty of Information Technology (FIT) operate several information systems that are separate and are utilized to gather information for various students' agendas. The multitude of information systems, which sometimes do not even have consistent information among each other, can pose a challenge for the faculty's new students. This thesis aims to research and implement the enhancement of an existing prototype of a system, which is addressing this problem.

The prototype for such a solution was originally developed by Bc. Tomáš Heger in his bachelor thesis [1], where he proposed an information system (web application) focused on facilitating access to information about courses, teachers, rooms – named Where Information System (WhereIS).

During the NI-NUR course[1], we participated in a team project that enhanced and expanded the functionality of the application's backend and completely reworked the application frontend. The division of responsibilities was as follows:

- Bc. Tomáš Heger – focused on backend development (adding features to find the canteens and check the menu, implementing a favourite canteens system, introducing a close events calendar, refactoring old code, developing a new REST API),
- Bc. Illia Brylov – dedicated to frontend development (designing the new user interface for the enhanced version of the application, implementing the frontend in the React framework, connecting the frontend to the new REST API),
- Bc. Jakub Jabůrek – engaged in both backend and frontend improvements (refactoring the user login process, managing session, implementing a return button on the frontend, orchestrating Docker deployment).

In this thesis we will describe the work which was done and will continue in implementing the frontend solution.

[1]NI-NUR is a FIT course covering the theoretical aspects of human-computer interaction and user interface design.

Thesis Goals

The primary goal of this thesis is to deliver a new, production-ready user interface frontend solution for the WhereIS application.

The analytical part of the thesis aims to design a new user interface for the WhereIS application. This design process will start by researching the methods of User-Centered Design (UCD) and futher application of these methods like: gathering and analysis of user requirements and ideas, provided through a student questionnaire, the analysis of the WhereIS application prototype developed by Bc. Tomáš Heger in his bachelor thesis, and a review of similar search systems.

The implementation phase is tasked with delivering a newly created frontend solution that adheres to the requirements outlined in the analytical part of this thesis. Next steps involve usability testing of the developed solution with students, complemented by a heuristic evaluation to assess its effectiveness and identify areas for improvement.

Motivation

The complexity and inconsistency of high amount of systems used by the faculty and university are widely recognized challenges among its students. Therefore, the thesis author is highly motivated to contribute to a project with the potential for immediate and meaningful impact. This initiative aims to significantly improve the experience of both current and future members of the Faculty of Information Technology (FIT).

Thesis Structure

The Chapter 1 will describe the reader a theoretical background on user-centered design principles, which will support the analysis and implementation in subsequent chapters. It will also outline the methodology for usability testing of the solution.

The Chapter 2 will introduce the existing faculty and university systems and external systems offering similar functionalities. Then it will analyze the user interface of the application prototype implemented by Bc. T. Heger in his bachelor thesis, ending with the analysis of student questionnaire responses.

The Chapter 3 will report the design process for the new user interface solution and will end with a low-fidelity prototype.

The Chapter 4 will focus on the implementation of the UI based on the low-fidelity prototype within the React framework. This implementation will connect to the new REST API developed by Bc. T. Heger in his master thesis and be deployed to a public domain address.

The Chapter 5 will assess the implementation through heuristic evaluation and in-person moderated usability testing.

Chapter 1

Overview

The problem is that in avoiding the paths that contain the tar, you may never reach any destination; in avoiding temptation, you remain pure, but irrelevant.

Donald A. Norman

In this chapter, we introduce the reader to User-Centered Design (UCD). We will employ this framework in the next chapters to assist in analyzing and designing user interface solution of WhereIS application. This chapter covers the methodology to be utilized throughout this thesis.

1.1 User-Centered Design

In this section we will tell about the origins of User-Centered Design and it's key principles.

1.1.1 Origins

The User-Centered Design (UCD) approach comes from studying how people interact with computers. It got a big boost from Donald A. Norman's important paper [2], "Design Principles for Human-Computer Interfaces", written in 1983. Norman pointed out problems with how systems were made back then, mainly that there wasn't a clear way to check if a design was working well or even finished. People making these systems really didn't have a set way to measure if a system was "good enough" or "complete".

Norman suggested four methods to add a needed approach, focusing on different parts of designing to make it more focused on users:

- recognizing that users have distinct needs, which even the best-intentioned designers might not meet with their designs,
- the introduction of quantitative methods and guidelines, which, however, may clash with the realities of development pressures,

- enforcing consistency through software tools for interface design, embedding principles directly into these tools,
- separating the interface design process from other programming tasks, pushing for a standard way to communicate and work on the interface that even non-programmers can use, possibly through software tools.

Norman's ideas, which are now well-known, helped shape the principles we generally use in design today. His work led to the User-Centered System Design (UCSD) project, which puts the user at the heart of the design process. This project highlighted five key ideas:

1. **Trade-offs are inherent:** Every design principle application has its strengths and weaknesses, with the goal being to make trade-offs explicit.
2. **There are no errors, only iterations:** Designers should view each user action as a step towards a goal, facilitating efficient progression to the desired outcome.
3. **The importance of low-level protocols:** Consistency in basic user *tasks* across systems standardizes the user experience significantly.
4. **Structured activities:** User tasks naturally group according to *goals*, suggesting that operations should be organized to reflect this perception.
5. **Dominance of information retrieval:** Importance of designing intuitive file and directory structures, focusing on making information and help resources easily accessible to users.

Norman's talk about trade-offs brought about a way to measure how happy users were with different parts of a system. He showed how people who don't use a system often and those who do, have different likes – people who use it less prefer menus, while regular users like commands. These choices affect how fast users can use the system, how easy it is to learn, how mistakes are fixed, and what users need to know beforehand.

This work started a bunch of other studies that gave the industry and academia clearer ways and rules for making and evaluating systems, all aiming to put the user's needs first.

1.1.2 Human-Centered Design

Originally published in 1988 as "The Psychology of Everyday Things", the book [3] by Donald A. Norman was reissued in 2013 as "The Design of Everyday Things", is very influential in both academia and practical fields.

In this book the author describes Human-Centered Design (HCD). HCD makes sure that when we design products, we think about what people *need*, *can do*, and *how they act*. This way, designs are made to be easy for users to understand and use. The main idea is to make designs that talk clearly to users, telling them what they can do, what's happening now, and what will happen next.

Norman highlights six fundamental concepts essential for HCD:

- Affordances

- Signifiers
- Constraints
- Mappings
- Feedback
- Conceptual Model

Among these, the conceptual model is really important because it helps people really understand how a system works, making it easier for them to use it smoothly.

This thesis will take a closer look at HCD's six fundamental principles to uncover the guides of design thinking that make products great for users.

Affordance. The idea of *affordance* [3, pp. 10–13] is about how a physical object can be used by someone or something, like an animal, human, or even a machine. It's all about the match between what an object can do and what a user can do with it. For example, a chair is made to support weight, which means it's made for sitting.

Even if you can't see an affordance, it's still there. But for people making things, it's important that these affordances are easy to see because they guide users on how to use objects.

Affordances are all about the opportunities in the world for interaction. While some of these opportunities are easy to notice, others might not be so obvious.

When people can easily tell what they can do with an object without needing a guide or label, that's thanks to perceived affordances. The term *signifiers* was introduced to describe the aspects of affordances that communicate where and how to interact with something.

Signifiers. *Signifier* [3, pp. 13–20] means any kind of mark or sound, anything you can notice that tells people how they should act. Signifiers are crucial for communicating with users, whether you meant to communicate or not.

People need clues to understand how to use products or services. They look for any hint that can help them figure things out—what it's for, what's happening, and what else they could do. It's the hint that matters, anything that could give useful information. Designers have to give these clues. What people need, and what designers have to give, are signifiers.

Signifiers are like signals. They can be signs, labels, or pictures out in the world, like the "push", "pull", or "exit" signs on doors, or arrows and diagrams that show what to do or where to go, or other kinds of guidance. Some signifiers are just things you see and understand right away, like a door handle or how a switch is built.

Mapping. *Mapping* [3, pp. 20–23] is crucial for designing controls and displays. It's about arranging controls in a way that matches what they control, making it easier for users to understand how to use them.

Using spatial analogies, or natural mapping, helps people get how things work right away. For instance, pushing a control up to move something up, or arranging light switches to match the layout of the lights. Some of these mappings are based on universal human experiences, like moving your hand up to mean "more" and down for "less", which is why we often use up and down to show levels of intensity.

But, it's important to remember that some mappings we think of as "natural" might actually be specific to certain cultures. **What feels intuitive in one culture might not in another.**

A gadget becomes user-friendly when its design makes the actions you can take obvious, and when its controls and what they show you make sense naturally.

Feedback. *Feedback* [3, pp. 23–25], the process of letting users know their action's outcome, is crucial in interaction design. Immediate feedback is essential; even slight delays can be unsettling, leading users to abandon their tasks.

However, excessive feedback can be as problematic as insufficient feedback, overwhelming users and causing them to ignore or disable notifications altogether. This risks missing out on important alerts.

Feedback must be carefully designed to confirm actions without being intrusive. It's vital to prioritize feedback so that less critical information is subtle, while crucial alerts are noticeable, ensuring they grab the user's attention without disrupting the overall user experience.

Conceptual Model. A *conceptual model* [3, pp. 25–30] simplifies how something works. It's a mental shortcut that doesn't need to be fully accurate to be helpful. For instance, computer users understand files and folders through desktop icons, creating a mental picture of how their computer organizes data.

Conceptual models range from complex versions in manuals to simpler, mental ones people use daily. These mental models, unique to each individual, help users grasp how things function. People might even have multiple, sometimes conflicting, models for a single object, based on its different aspects.

Clues to operation come from the system's design elements like signifiers, affordances, and mappings. These models are crucial for understanding how things work, predicting outcomes, and deciding what to do when unexpected issues arise.

Users develop their conceptual models through interactions with the product, instructions, and any available resources. While designers hope users' models will match the intended design, the reality often differs due to the indirect communication methods.

1.1.3 Conclusion

We've introduced two terms, which do sound quite similar. The difference is, that Human-Centered Design (HCD) looks at the big picture of designing things based on what people need, while User-Centered Design (UCD) focuses more on how users interact with a product. In simple terms, UCD is a subset of HCD, and both are about making designs that are easy and good for users. In this thesis we will stay focused on a narrower one – UCD.

1.2 UCD in Practice

In this section, we'll explore how UCD transitioned from the academic insights of Donald A. Norman into widespread industry use. We'll examine the implementation of UCD, highlighting its contributions to enhancing product development process.

1.2.1 IBM

The origin of modern UCD is widely attributed to International Business Machines Corporation (IBM) in the 1980s. At the Thomas J. Watson Research Center, Gould and Boises developed a methodology that emphasized four critical steps: focusing early on the characteristics and needs of the intended user population, inclusion of users as part of the design team, empirical and experimental measurement, and iterative practices. They believed that this principled approach was necessary for the progress towards systems that are significantly easier to learn and more useful [4].

IBM's Design Thinking [5] is simple – it's about making users the top priority for the whole team. This concept has been developed further by adding methods and actions to build a structure that applies design thinking to teams of any size, no matter if they are in the same place or spread out.

Karel Vredenburg in his article [6] states, that IBM's UCD process emphasizes designing the complete user experience, from when a potential customer first hears about a product or service to when they think about upgrading. This is different from older methods that mainly concentrated on the software's graphical user interface or hardware components like the computer keyboard. To carry out this process, IBM uses a team with diverse skills to design all parts of the user's interaction.

Percival Lucena et al. explain [7], that previously, a software's potential market share was mostly restricted by the supply chain and distribution ability. But now, a product's growth is based on how well it meets user needs. IBM Design Thinking includes initial analysis and user feedback in all stages, helping to better understand the problems to solve and the best solutions for users. A survey showed that 80 % of users were very satisfied with the projects delivered. The enhanced user experience led to increased productivity, saving time and resources, and expanding the user base for the services provided. Despite these positive qualitative results, more research is needed to accurately gauge these satisfaction levels and understand the limitations of the software framework developed.

Karel Vredenburg adds [6], that it's important to mention that while IBM set up a human factors organization more than 40 years ago and used different usability and human factors methods over time, it was only with the launch of an improved version of UCD in 1995 that a significant organizational change occurred.

1.2.2 UCD – Industry State

In this section, we'll explore how UCD was received by developers and users in real-world scenarios. We'll do this by reviewing the outcomes and various surveys and analyses.

Ji-Ye Mao et al. describe [8], that development of UCD has seen a range of views on its usefulness and use. At first, many UCD methods were criticized for being inefficient or impractical for various reasons. Nielsen pointed out a common feeling among developers, who found usability engineering methods intimidating because of their complexity, time use, and high costs. However, the rise of e-commerce has greatly highlighted the importance of usability and UCD. The ease with which users can change providers online means that badly designed websites could lose up to half their visitors. As a result, UCD is now widely suggested for creating user-friendly web designs, stressing the need to assess and improve UCD practices. The survey showed that UCD has a big positive effect on product development. 72 % of people say that UCD methods have greatly shaped their company's product development, scoring 5 or more on a scale of 7. Most people agree that products have become more useful (79 %) and user-friendly (82 %), showing a wide agreement on the advantages of UCD. In other survey, interestingly, more participants (44 %) said that User-Centered Design methods lowered product development costs, while 24 % thought costs went up. This contradicts the old view of UCD as costly, where UCD expenses often went over 10 % of the whole project budgets.

Another survey [9] discovered, that the team structure in projects often shows a dedication to User-Centered Design. On average, 3 out of 17 team members are focused on UCD tasks. This distribution highlights the significance given to UCD within project teams. Further analysis reveals that UCD methods are widely regarded as beneficial for enhancing product usefulness and usability. However, the adoption rate of UCD methodologies varies significantly across organizations, indicating room for a more unified approach.

Moreover, Zahid Hussain et al. state [10] that the integration of UCD with agile development practices is seen as a rising trend. Agile team members are more and more acknowledging the essential role of usability in software development. They note that this combination improves product quality, usability, and user satisfaction. It also brings value to the team and the development process, a feeling shared by most survey participants who have implemented agile UCD processes.

In the study by Olujimi Daniel Alao et al. [11], the University Information System was designed using the UCD and then the efficiency, effectiveness, user-friendliness, simplicity, and learnability of the proposed design were assessed. The design achieved a System Usability Scale (SUS) [2] score of 84, which is considered above average. This score indicates that participants found the platform enjoyable to use and were able to navigate and interact with it easily, despite their first-time exposure. The study findings suggest that when developing comprehensive platforms like University Management and Information Systems, which integrate various subsystems, it is crucial to prioritize usability and a positive user experience. This research highlights the importance of focusing on usability to enhance the functionality of such platforms.

[2]System Usability Scale – is a simple, ten-item attitude Likert scale[12] giving a global view of subjective assessments of usability.

1.3 Challenges and Critiques

Even though the UCD had positive outcomes in development and user feedback, some issues can still be found.

The study by Ji-Ye Mao et al. from 2001 [9] showed mixed views on whether User-Centered Design is cost-effective. About a third of those surveyed were unsure if UCD saves money in product development, giving a neutral score of 4 out of 7. Interestingly, more people (45 %) thought UCD methods cut down on product development time, while fewer (22 %) believed it raised costs. Over 20 % seeing UCD as increasing costs is a concern that needs more research. This also applies to views on product development time, indicating a need for more in-depth study. It also shows it's hard to measure how well UCD works and to agree on evaluation standards. People can't decide how to judge UCD success, often using personal judgement even though they'd prefer objective measures. This could slow down UCD's wider use. The study suggests we need standard ways to measure UCD effectiveness to help it grow and be more widely accepted.

1.4 Methods

This part is fully dedicated to describing the methodology of our work on the new WhereIS user interface solution, which is centered arround using UCD principles described in previous parts.

1.4.1 The Human-Centered Design Process

The Human-Centered Design Process was introduced in already mentioned best-selling book [3] by D. Norman – "The Design of Everyday Things" describes the two phases of design: finding the right problem and fulfilling human need.

There are four different activities in the Human-Centered Design Process:

1. **Observation**
2. **Idea generation (ideation)**
3. **Prototyping**
4. **Testing**

These four tasks are repeated; meaning, they are done again and again, with each round providing more understanding and bringing us nearer to the wanted solution. We will use Human-Centered Design Process as a general guideline for our UCD design process, since, how we've mentioned before, the UCD can be considered as a subset of HCD. Now, let's look at each activity individually.

Observation. When we look at users, it's key they actually reflect who we're designing for. Traditional details like age or income often don't tell us as much as what the user does. Even across different cultures, the kinds of tasks people do are surprisingly alike. So, our studies should zero in on these tasks and how they're accomplished, mindful of any local cultural or environmental tweaks. This is especially true for business products where the tasks stand out.

In design *research* the first phase is about figuring out the real problem, requiring deep knowledge of what users truly need. Once the design team knows the problem, finding the right solution again dives deep into understanding our target users – their tasks, skills, past experiences, and cultural specifics that might influence the solution.

Idea generation (ideation). After figuring out what are the goals, team starts generating potential solutions – this is called ideation. It happens twice: first, when we're figuring out exactly what the problem is, and then when coming up with ways to solve it. Ideation is where the design team can get to be really creative. There are lots of different methods how to ideate, commonly grouped under "brainstorming" [3].

Prototyping. To really check if an idea works, design team has to test it. This means creating a quick prototype for each solution. Early on, these can be as simple as drawings, models out of foam or cardboard, or even just pictures made with basic tools. Prototyping in the beginning is mainly to make sure everyone understands the problem correctly. If people are already using something similar to what the deisgn team is making, they can think of that as a kind of prototype too.

Testing. The design team should assemble a small group that closely matches the target user demographic. This group should use the prototypes in a manner that closely simulates real-world usage. If the device is typically used by one person, the team should conduct tests with individuals one at a time.

After the study, the design team should delve deeper into participants' thought processes by reviewing their actions with them. This involves reminding them of their actions and posing questions. It can be beneficial to show participants video recordings of their activities as reminders.

1.5 WhereIS Design Process

In this part, we'll outline the various aspects of the design process of the WhereIS user interface. These aspects are closely aligned with the four main activities of the Human-Centered Design Process. This design process will be used later in all subsequent chapters of this thesis.

1.5.1 Observation

1.5.1.1 Project Analysis

The design process begins by documenting the essential information about our project, which is creating the user interface for the WhereIS application. This step covers defining the product's purpose, the business needs it aims to meet, and any constraints that might affect its development.

[3]Brainstorming - a session where people come up with lots of new ideas [13].

Product Statement. Product vision, often known as a product statement, outlines the product's broad, long-term objective [14]. Vision statements are inspirational statements that explain briefly where the product wants to go and what it wants to accomplish in the long run. The statement should serve as a useful tool to all stakeholders engaged in creating a product about the common goal they're attempting to achieve with this product. Our vision statement should also address why we are developing a product and what our organization expects to achieve with it in the future.

Geoffrey Moore in his book "Crossing the Chasm" says [15] that product statement should clearly answer on these three questions:

1. **What is the product?**
2. **What is new about the product?**
3. **What the product is not?**

He also recommends, that product statement can be in a sentence format like: ***Product name*** *is a* ***what*** *that does* ***what*** *for* ***whom****.*

Business Requirements. Business requirements, crucial for delivering project value, typically focus on financial outcomes like boosting revenue or lowering costs. Yet, the value can also be found in intangibles such as improving the user experience [16].

The significance of precise requirements is highlighted by research showing that problematic requirements lead to a significant portion of project failures, contributing to a major share of rework costs and software issues [17, 18].

Additionally, a study [18] notes that revisions in project requirements are a frequent cause for project delays and budget overruns, particularly in government IT projects.

The business requirement could be, for example: "The system should maximize the user conversion rate[4]".

Constraints. Constraints in a project outline the boundaries and limitations the project must operate within. These can stem from various factors, both inside and outside the project's control. Clearly defining constraints helps set realistic boundaries for what the project can achieve. The constraint, for example, could be that the project is intended for the target user group from a specific country or region.

1.5.1.2 Existing Designs Research

The study M.L. Wong et al. state [19], that user interface designers should be aware of the market needs and to compare existing designs in the market, if any. The market positioning and comparison between proposed design and any existing designs can be made in each task, target user or interface.

[4]A conversion rate records the percentage of users who have completed a desired action.

Search Systems' User Interfaces. Researching the academic foundation of search systems and studying existing systems with search functionalities will inspire us and provide reference points. Through analyzing these systems, we can identify trends, best practices, and potential pitfalls to avoid.

CTU Systems Research. Investigating CTU systems with search capabilities will offer a deeper understanding of the specific context and requirements of the target user group. This research ensures that the design aligns closely with the needs and preferences of CTU users.

Prototype Research. We will examine the existing prototype by Bc. T. Heger [1] to understand its functionality and identify any design issues. This examination will allow us to assess what the app does, identify any missing features, and pinpoint areas where usability can be improved.

1.5.1.3 User Requirements Questionnaire

A questionnaire is a research instrument consisting of a series of questions with the aim of obtaining information and responses from participants. In the preliminary stages of system development, conducting a user questionnaire serves as a vital step to understand user perspectives and preferences before actual implementation. This is supported by the results of, where the new University Management Information System was designed to improve the user-experience with the app [11].

Questionnaire have also been shown to be valuable tool for improving user experience, as demonstrated by Jude Ejike and Ediz Edip Akcay in their work on enhancing the NHS app [20].

Distributing questionnaires to respondents is an essential aspect of understanding user needs in system development, aligning with the principles of User-Centered Design [21].

Translating insights gathered from user questionnaires into actionable requirements is crucial for system development. The results from the questionnaire can be employed in User-Centered Design techniques such as personas, use cases and scenarios. This approach ensures that the resulting system aligns with user expectations and needs [22].

The questionnaire will consists of Likert scale [12] rating questions as well as open-ended questions. Utilizing the Likert scale ranging from 1 to 5 in rating questions, the questionnaire assesses users' experiences, providing valuable insights into user satisfaction and areas for improvement. Open-ended questions within the questionnaire allow respondents to express themselves in detail [11].

1.5.1.4 Personas

Personas, introduced in the late 1990s, serve as a pivotal tool to maintain user-centric design practices. They represent synthesized views of target users, embodying common behavioral characteristics. By offering designers a fictional but holistic perspective of user groups, personas aid in conveying diverse user needs effectively. Empathy-driven narratives within personas enhance designers' understanding of user perspectives and needs. This approach ensures that design decisions are aligned with user requirements, ultimately resulting in superior designs. The primary advantage of personas lies in their ability to focus product design teams on the specific goals and needs of target customers. By bringing target consumers to life, personas integrate their needs and objectives into the design process. This integration bridges the gap between designers and consumers, enhancing the effectiveness of User-Centered Design (UCD) processes [23].

Designers, when empathizing with personas, can effectively address user needs despite limited knowledge about specific user requirements. This empathetic connection allows designers to infer user desires and guide the design process accordingly [24].

Furthermore, personas facilitate the integration of diverse user viewpoints in the design process, even when access to real users is limited. Contrary to distancing the team from users, personas ensure the inclusion of various user perspectives in the design of web-based e-services [25].

In software analysis and design, personas help focus on the features and goals of the product's end user, ensuring the relevance and effectiveness of the design process [26].

To better understand the interface's possible users, we will define three personas:

- **Persona A**: is a *primary* user. Will use all functionalities and the system must be implemented for them. We are designing the interface for this persona User Goals,
- **Persona B**: is an *occasional* user. We are not implementing the interface for them. But the solution should solve their User Goals,
- **Persona C**: is a *negative* user. The system is not designed for them. This user probably *never* uses this system.

1.5.2 Ideation

The next part of design process is determining what the intended users' goals are, and then devising an interface that can help people achieve those goals by completing a series of tasks. Goals in the domain of information access can range quite widely, from finding a plumber to keeping informed about a business competitor, from writing a publishable scholarly article to investigating an allegation of fraud. Information access tasks are used to achieve these goals. These tasks span the spectrum from asking specific questions to exhaustively researching a topic [27].

1.5.2.1 User Goals

User Goals serve as a general description of what the application user aims to achieve. Requirements elicitation means figuring out what the system needs to do. There are two main ways to do this: focusing on how people will use it or focusing on the product itself. When we use the first method, called 'usage-centric,' we pay a lot of attention to what users want to achieve, a this is called user goals. This helps us figure out what functions the system should have [26].

Goals give us a clear idea of what the system needs to do, especially in terms of its functions. But they can also include other important details like how fast the system needs to respond, how often certain goals come up, which users are a priority, their characteristics, the rules the system needs to follow, and how complex the goals are [28].

1.5.2.2 Use Cases

Use Cases serve as practical roadmaps guiding the system toward fulfilling the User Goals. They delineate the interactions between users or other systems and the system itself to accomplish specific objectives. For instance, if the User Goal is making a purchase, a Use Case might outline the steps involved in completing that purchase online.

The introduction of Use Cases aimed to refine how requirements are documented, replacing vague paragraphs with clear scenarios illustrating user-system interactions to achieve their goals. They provide a comprehensive view, focusing on what users want to achieve and how the system facilitates those objectives [28].

In simpler terms, a Use Case narrates what a user *expects* from the system and how the system responds to meet those needs. This understanding is crucial in human-computer interaction as it helps capture user needs from their standpoint [29].

Use Cases come in varied formats, but the best approach is to write them from the user's perspective, detailing their expectations or demands from the system [30].

Martin Fowler states [31]: "There is no standard way to write the content of a use case, and different formats work well in different cases."

He describes "a common style to use" as follows:

- **Title**: goal the use case is trying to satisfy,
- **Main Success *Scenario***: numbered list of **Steps**,
- **Step**: a simple statement of the interaction between the actor and a system,
- **Extensions**: separately numbered lists, one per **Extension**,
- **Extension**: a condition that results in different interactions from the **Main Success *Scenario***.

Each Use Case is a composite of scenarios linked by a shared goal, with actors (users or systems) playing distinct roles in achieving that goal [29].

1.5.3 Prototyping

1.5.3.1 Scenarios

Scenarios are sequences of interactions that occur under specific conditions to achieve the main actor's goal, resulting in a particular outcome. These interactions start with a trigger and continue until the goal is either achieved or abandoned, with the system fulfilling its responsibilities along the way [28].

The UI design often undergoes frequent changes, making it impractical to use scenarios as contractual or system requirements. Typically, UI design happens after defining goals and interactions. Therefore, most prefer to focus on semantic interaction levels, outlining what information needs to pass through the system without specifying the sequence or nature of the interaction [28].

1.5.3.2 Tasks and Task Groups

User tasks outline the specific steps users will perform to accomplish their goals with the product. These tasks should align with User Goals and corresponding Use Cases. Tasks serve as important components in designing user interactions with a new product. This phase focuses on defining the user's interaction process and the specific steps required to achieve their goals, **without considering the design of user interface elements** [32].

Defining user actions involves deriving tasks and subtasks from the system's functional requirements or User Goals. These actions are validated to ensure they activate system functions effectively [33].

Tasks can then be grouped into Task Groups based on whether they can be accessed from the corresponding user interface page, providing a structured approach to organizing user interactions with the system.

1.5.3.3 Lo-Fi Prototype

A low-fidelity (lo-fi) prototype serves as a simplified representation of design concepts, often created through sketches or basic interactive paper models. It allows for easy visualization of design ideas and facilitates comparison between different alternatives without requiring significant time and effort [34].

Lo-fi prototypes originated as a technique in software design, particularly in the era of desktop computer interfaces. One common method involves creating paper prototypes of desktop interfaces, with simulated interactivity provided by a human familiar with the interface functionality [35].

Creating prototypes also aids developers in refining application designs, as they can be easily modified based on development needs. Prototyping is expected to save resources, time, and funds while ensuring the efficiency and usefulness of the final applications for users [36].

In our case, the type of lo-fi prototype utilized will be the wireframe. Wireframes are initial designs of graphical user interfaces (GUIs), featuring simplified visual components and layouts selected based on user needs and software requirements. Designers iteratively refine wireframes by comparing them with existing online design examples [37].

Wireframes capture the type and layout of visual components without focusing on high-fidelity visual details. They can be rapidly prototyped and refined with minimal effort, allowing for the exploration of visually distinct but semantically relevant UI designs [37].

1.5.3.4 Implementation

The implementation phase involves developing the user interface solution, which will be a web client communicating with the backend server.

The choice of the web as a medium was deliberate due to its widespread usage and multiplatform compatibility, accessible from devices such as mobile phones, PCs, and tablets.

Recent advancements in JavaScript technologies have enabled HTML rendering to occur in the client. This architectural approach follows a thick-client concept with a simplified backend, where the client is a Single-page Web Application (SPA) emulating the routing process. The thick-client model avoids synchronous communication with the server, with the JavaScript application in the browser responsible for determining when to communicate with the server [38].

Communication between the client and server is asynchronous. In asynchronous HTTP requests, the client sends a request to the server without the browser blocking while waiting for the result. This typically occurs with an AJAX request. Upon receiving the result from the server, a browser event is triggered, typically handled by a JavaScript function that updates the DOM tree with the new data [38].

In traditional approaches, each user interaction with the page triggers a server request resulting in a new page, disrupting the user experience. With the modern approach, requests go directly to JavaScript event handlers, enabling the client to remain on the same page throughout the session without requiring page reloads [38].

The user interface implementation, i.e., the web client, will be developed using the React framework [39], as specified in the thesis description.

1.5.4 Testing

To evaluate the implementation both heuristic analysis and usability testing will be conducted. Among all techniques, user testing and heuristic evaluation are perhaps two of the most popular ones [40].

1.5.4.1 Heuristic Analysis

Heuristic evaluation is an informal method for finding usability problems in a user interface design by having a small set of evaluators examine the interface and judge its compliance with recognized usability principles (the "heuristics") [41, 42].

These are as follows [43]:

1. **Visibility of system status**
2. **Match between system and the real world**
3. **User control and freedom**
4. **Consistency and standards**
5. **Error prevention**
6. **Recognition rather than recall**
7. **Flexibility and efficiency of use**
8. **Aesthetic and minimalist design**

9. **Help users recognize, diagnose, and recover from errors**
10. **Help and documentation**

Heuristic evaluation was originally developed as a usability engineering method for evaluators who had some knowledge of usability principles but were not necessarily usability experts as such [41].

Heuristic evaluation thus falls into the general category of usability inspection methods together with methods like pluralistic usability walkthroughs, claims analysis, and cognitive walkthroughs, with the main difference being that it is less formal than the other methods and intended as a "discount usability engineering" method [41, 44]. In general, the goal is to find major usability problems in a design or a system without using the large set of resources typically required for usability testing [44].

In evaluation sessions, each evaluator inspect the interface alone and only find a small number of usability problems. Severity ratings can be collected by sending a questionnaire to the evaluators after all evaluation sessions, listing the complete set of usability problems that have been discovered, and asking them to rate the severity of each problem [40].

This fast and inexpensive tool [41, 45] that can be used to identify and address design issues early in a project. It's not meant to replace user tests, but rather add to the set of evaluation tools. Four experiments showed that individual evaluators were mostly quite bad at doing such heuristic evaluations and that they only found between 20 and 51 % of the usability problems in the interfaces they evaluated [42]. While heuristic evaluation can be applied early in a project, user tests are conducted later to find issues that could not be captured before [45].

It's true that the most reliable results are often obtained when analysis involves three or more evaluators. Unfortunately, due to the current circumstances and the scope of this thesis, achieving such a level of participation is unfeasible. For this reason, in our project, we'll use heuristic analysis done by the thesis author alongside user testing to ensure better evaluation results.

Evaluation will follow the structure outlined in the workbook made by Nielsen Norman Group [46]. We will progress through the same test scenarios that will be prepared for the usability testing sessions mentioned earlier. Each scenario walkthrough will be assessed based on the 10 specified heuristics. Issues identified for each scenario and heuristic will be documented, along with recommendations for resolving them. Further details on the judgment criteria will be presented in the Chapter 5 of this thesis.

1.5.4.2 Usability Testing

Usability testing is crucial for evaluating the ease of use of the designed interface, especially for the target user group [11]. It helps to:

1. identify flaws in the product's design,
2. highlight areas for improvement.

The study of how people search can vary significantly, making it challenging to directly compare results. Sometimes, time may not be the best metric for judging a search tool, especially if it helps users learn as they search. To address this, two popular methods are used: longitudinal studies, where users interact with a new interface over time, and large-scale A/B testing on busy websites, where users are unaware they are part of the test. These methods help assess how new features perform over time or immediately, based on user actions and feedback [27].

Implementing both of these methods can be challenging given current conditions and resource limitations. Therefore, a lighter version of the first method is proposed for the initial usability testing of the WhereIS user interface.

The large-scale usability testing described as the second method can be conducted once the system is deployed for an extended period and has a sufficient user base.

Usability testing of the WhereIS user interface will be conducted with students from FIT CTU in three steps, individually for each user.

1. **Introduction and Pre-test Survey:** Users will be briefed on the purpose and rules of the testing. They will also complete a Pre-test Survey, providing demographic information.

2. **Application Testing:** Users will interact with the application, attempting to navigate through prepared test scenarios. These scenarios are presented as stories, combining various Use Cases. The goal is to observe user behavior and problem-solving approaches, documented in the test protocol.

3. **Post-test Survey:** After testing, users will complete a Post-test Survey. This survey will include rating questions on a Likert scale (1 to 5) regarding their opinion of the tested application. Additionally, an open-ended question will seek feedback on the overall experience and suggestions for improvement.

The questions and scenarios chosen for these three steps will be outlined in the Chapter 5 of this thesis.

Chapter 2

Analysis

This section is linked to the Observation 1.4.1 design phase. We'll start by exploring the theoretical background of search user interfaces to understand their main components, best practices, and design recommendations. Then, we'll analyze some of the most well-known search user interfaces, including internal university and faculty systems, along with the application prototype by Bc. T. Heger [1]. The section will conclude with an analysis of the User Requirements Questionnaire, which will help us gauge the mood of FIT participants regarding the use of existing faculty systems and gather their ideas.

2.1 Search Systems' User Interfaces

To gain a comprehensive understanding of search system user interfaces, we'll explore Marti A. Hearst's chapter "User Interfaces for Search" in the book "Modern Information Retrieval: The Concepts and Technology behind Search Engines" [27]. Hearst break down the essence of search and its user interface components.

Search is broadly categorized into two main types: information lookup and exploratory search. The latter further divides into learning and investigating tasks, where the goal might involve thorough research or staying updated on a topic. The search process is seen as valuable not only for its end results but also for the learning that occurs along the way.

The traditional model of information seeking is depicted as a cycle with stages such as problem identification, articulating information needs, formulating queries, and evaluating results. Recent perspectives, however, view the search process as dynamic, recognizing that searchers' needs evolve as they gain new insights.

A key strategy in this evolving process is *berry picking* or *orienteering*, where users start with broad queries to navigate towards more specific information. This approach is supported by the fact that many web users prefer browsing through structured information rather than starting with a keyword search. The browsing experience can be less mentally demanding and more effective when information structures align well with users' needs.

An important aspect of search usability is the searcher's ability to maintain their "scent" towards their goal without getting lost. This underscores the importance of well-designed navigation and the value of allowing users to easily revisit previously accessed information, highlighting features like “favourites”.

Hearst also notes that in web search engines, users often focus on the top-ranked results, assuming that the top results are the most relevant simply because of their position.

This section provides a foundation for designing user-friendly search interfaces by outlining essential components and strategies that cater to users’ evolving information needs and search behaviors.

2.1.1 Query Specification Interfaces

Hearst describes, that for web search engines [5], users typically input their queries using a keyboard, often in the form of brief text, usually ranging from one to three words. This input is facilitated by a search box entry form, where users type their query and then activate the search either by pressing the return key on their keyboard or clicking on a search button associated with the form.

To assist users, search interfaces often integrate hints within the search box, using grayed-out text to suggest what type of information should be entered. For example, a search box might include prompts like “What are you looking for?” or “When (tonight, this weekend, ...)”. As soon as the user starts typing, this placeholder text disappears, allowing for the input of search terms.

An innovation that significantly enhances the process of specifying queries is the dynamic generation of query suggestions as the user types. Known variously as auto-complete, auto-suggest, or dynamic query suggestions, this feature provides real-time suggestions, streamlining the search process. Depending on the design, the query might execute immediately upon selecting a suggestion or might require the user to either press the Return key or click the Search button to initiate the search.

2.1.2 Retrieval Results Display

When presenting search results, it’s crucial to either show the full documents or provide a summary that represents their content effectively. This summary, known as a *document surrogate*, is essential for a successful search interface. The creation of document surrogates and the design of retrieval result displays are areas ripe for innovation and research.

Web searches typically display the page title prominently, accompanied by the URL and occasionally other metadata. For searches across specific information collections, metadata like the publication date and the author’s name are often included, though such metadata is less relevant to Web pages. The text summary, also referred to as an abstract, extract, excerpt, or snippet, which contains text from the document, is crucial for evaluating the search results.

The conventional approach to displaying results is a vertical list of textual summaries, sometimes known as the SERP (Search Engine Results Page).

[5] A web search engine is a software application that crawls the web to index it and provides information based on user search queries [47].

The traditional format of the "10 blue links" [6] has evolved into more enriched interfaces, which are often personalized based on the search query, the user, and other factors.

Summaries may be excerpts from the full text featuring the search terms. Alternatively, specialized metadata may be displayed alongside the standard textual results, employing a method known as blended results (also called universal search).

In general, users engaged in known-item searching tend to prefer short surrogates that clearly indicate the desired information.

Studies have found that features that provide direct answers improve user engagement on SERP, reduce user effort, and promote user satisfaction. Besides contributing to user satisfaction, these features also encourage user engagement [48].

2.1.3 Organizing Search Results

Two popular methods for grouping search results are *category systems*, particularly faceted categories, and *clustering*.

A category system is a set of meaningful labels organized in such a way as to reflect the concepts relevant to a domain. They are usually created manually. Good category systems have the characteristics of being coherent and (relatively) complete, and their structure is predictable and consistent across search results for an information collection. Among the commonly used category structures for organizing search results and presenting information collections are *flat*, *hierarchical*, and *faceted categories*.

Faceted metadata, an alternative representation, has emerged as the primary means of organizing website content and search results. Faceted metadata lies between flat categories and full knowledge representation in complexity. Properly designed faceted metadata is understood by users and preferred over other organizational methods. Instead of one large category hierarchy, faceted metadata consists of a set of categories, each corresponding to a different facet or feature type relevant to the collection to be navigated. Once the facets are designed, each item in the collection is assigned any number of labels from the facets. After the facets are designed, each item in the collection is assigned any number of labels from the facets. An example of facets or labels could include the author, publication year, publisher, and genre when searching for a book.

Usability studies have shown that users like and succeed in using faceted navigation if the interface is properly designed. Faceted interfaces are overwhelmingly preferred for collection search and browsing as an alternative to standard keyword-and-results listing interfaces. However, users tend to dislike disorderly groupings often produced by clustering. They prefer understandable hierarchies where categories are presented at uniform levels of granularity.

2.1.4 Best Practices

Preethy Ann Kochummen emphasizes in her web article that users are accustomed to specific layouts for search functions. Therefore, it's crucial for

[6]"10 blue links" refers to the ten organic search results that appear on a search engine results page (SERP) in response to a user's query. These links are typically listed in order of relevance to the query, as determined by the search engine's algorithm.

designs to adhere to search UI best practices as much as possible to prevent confusion [49].

When a website hosts various content types or products, it's beneficial to provide users with filters and sorting functions. These options should be displayed on the search results page to allow users to refine their search without feeling overwhelmed. Additionally, it's advisable to visually separate navigation from content to avoid confusion [49, 50].

Microcopy refers to short and helpful text placed throughout the site to assist users in navigating and using the interface. It enhances site usability by clarifying the purpose of various search components. Birchbox [7], for example, utilizes microcopy in its search bar to inform users that they can search for both brands and products. Additionally, the site employs microcopy for search suggestions just below the search bar. This approach can be strategic for companies aiming to achieve business objectives by prominently featuring target brands where customers are likely to notice them [50].

Many websites choose to use a magnifying glass icon for their search instead of displaying the word "Search" directly. Others go for a more traditional approach, providing users with a submit button along with descriptive text. The choice between these options depends on factors like the type of business, the target audience, and the website's design [49].

Additionally, incorporating auto-complete or query suggestions, which provide real-time search recommendations as users type, can enhance the search experience [50].

2.2 External Systems Research

In this section, we will examine a well-known implementation of web search engine, which is used worldwide – Google Search.

2.2.1 Google

Google Search [8] is a web search engine developed by Google and is one of the most widely used search engines globally. Google Search enables users to search for information on the internet by entering keywords or phrases into the search bar. The search engine then returns a list of relevant websites, documents, images, videos, and other content from its vast index.

On Google Search's main page, the central feature of the system is the search query text input field, which occupies the majority of the screen space. As users begin typing, query suggestions dynamically appear and change with each keystroke. The search process is initiated only when the user presses "Enter" on the keyboard or selects one of the suggestions. Subsequently, users are directed to the Search Engine Results Page (SERP).

Google Search's SERP page consists of several parts:

1. **Search Bar**: At the top of the page, there's a search bar where users can see the current query or input the new one.

[7]Birchbox is a New York City-based online monthly subscription service that sends its subscribers a box of four to five selected samples beauty-related products.

[8]Available at: `https://www.google.com/`

2. **Search Results**: The main section of the SERP displays the search results relevant to the user's query. These results typically include a mix of organic search results [9], paid advertisements, and specialized search features such as featured snippets, knowledge panels, and related questions.

   The search results are *surrogates* as discussed in section 2.1.2. Typically, each search result comprises a clickable title representing the web page's title and a URL displaying the web address. Positioned beneath the title and URL, there's typically a concise excerpt providing a hint into the content available on the web page.

3. **Knowledge Panels**: Google may display knowledge panels or knowledge cards alongside search results for certain queries that typically involve well-defined entities and topics for which Google has reliable, structured data. These panels provide quick access to relevant information sourced from Google's Knowledge Graph [10], such as basic facts, summaries, images, and related topics.

4. **Featured Snippets**: Featured snippets are special search results that appear at the top of the SERP in a prominent box. They provide concise answers to specific questions or queries, often extracted directly from web pages and displayed in a way that aims to answer the user's question at a glance.

The analysis of the UI structure of Google Search is presented in Figure 2.1.

Bruno Oliveira and Carla Teixeira Lopes conducted a systematic analysis in their paper titled "The Evolution of Web Search User Interfaces – An Archaeological Analysis of Google Search Engine Result Page" [48]. This study focuses on tracing the trends in the evolution of search engine user interfaces and web design, with a specific emphasis on Google Search's web search engine.

According to their work, significant changes have occurred in the visibility of the search box and the current query for searchers since 2019. Previously, the query was always available in the search box, but it disappeared if users scrolled down. However, as of 2019, the current query is always visible, even when users scroll down and the header of the SERP with search box is no longer visible.

They noted several key features of the SERP layout. There is typically space for sign-in and user account information on the right side of the screen. At the bottom of the page, two noticeable areas are present: pagination, aligned to the center of the results container, and the footer, which spans the entire width of the page.

Users tend to focus on the first few search results and often reformulate their query if they cannot find promising results at the top of the list. Additionally, aggregated search, which involves aggregating results from heterogeneous sources (verticals) and presenting them in a single interface, has become a standard practice.

[9]Organic search results are the Web page listings that most closely match the user's search query based on relevance, and not affected by advertisers [51].

[10]The Google Knowledge Graph is a knowledge base from which Google serves relevant information in an infobox beside its search results [52].

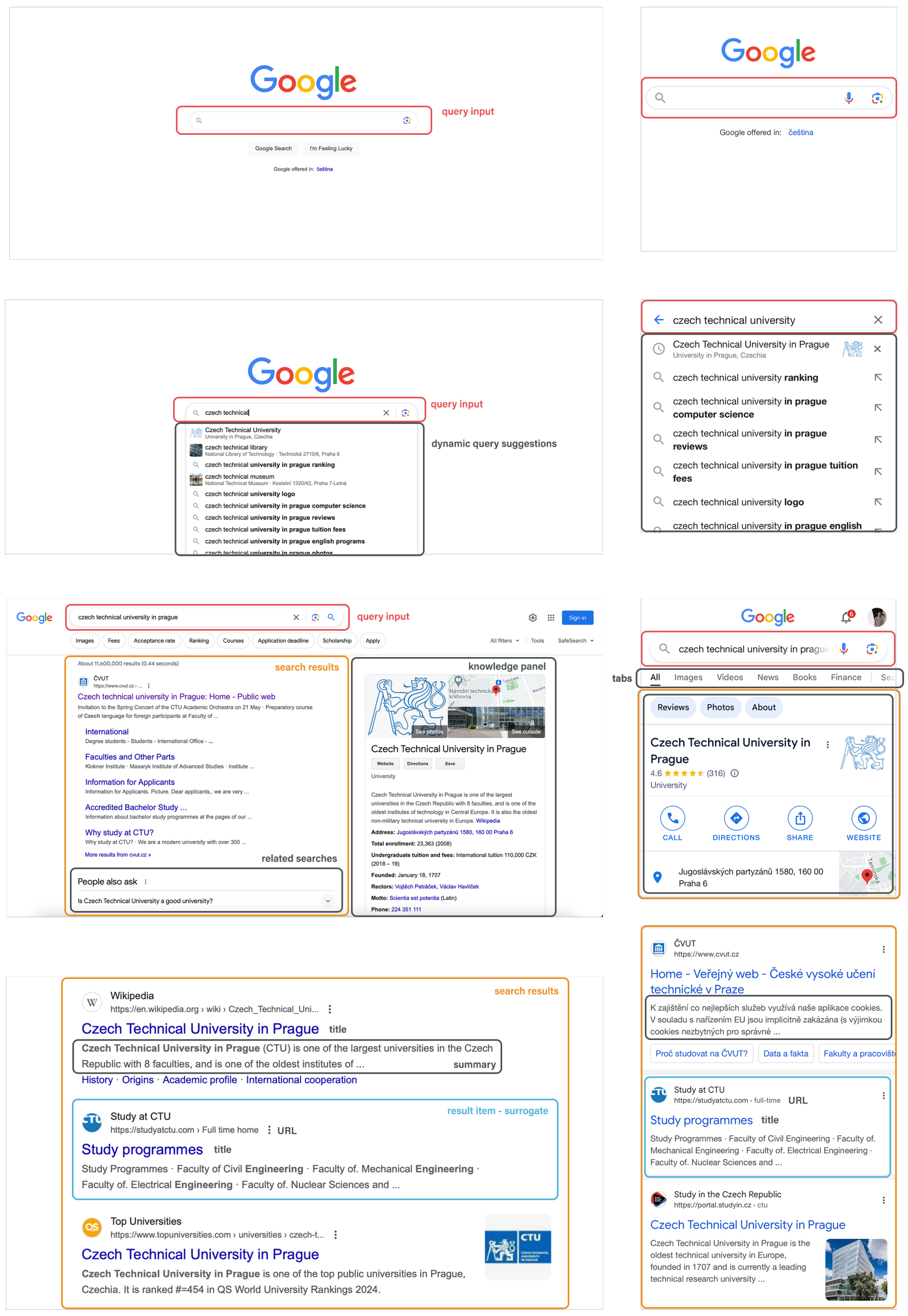


Figure 2.1: Google Search – UI structure

The Category Hierarchy, which was commonly used in the early years, has been discontinued, possibly due to the decline in popularity of web directories.

Featured Snippets, introduced in 2016, are answer boxes that provide responses to question-related queries based on information from web pages. They consistently appear at the top of the results container and differ from enriched results in the type and position of extra content.

The Knowledge Panel, is perhaps the highlight of SERP features. It is a dynamic feature that provides direct information in various formats within the same panel, pointing to related content. The contents range from text to images, ratings, social profiles, factual information, and similar search topics, helping the user to understand a particular subject quickly and facilitating a more in-depth search. This element appeared for the first time in 2014, lasting until now.

Related Searches, a common element on SERP pages from an early age, offers suggestions for related searches based on search log data. These suggestions can support exploration or provide alternative query statements to express information needs differently, with each link leading to the respective SERP.

2.3 CTU Systems Research

Students at FIT often find it frustrating to juggle multiple systems to keep up with their academic and non-academic needs. With new systems popping up nearly every year, the challenge keeps getting bigger. In this review, we'll explore some of the most commonly used systems, each serving different purposes but sometimes overlapping in functionality. We'll look at them from the student's viewpoint, focusing on how easy they are to use and how well they work.

2.3.1 FIT CTU Courses

Description. FIT CTU Courses [53] are the cornerstone of our faculty's system, and for good reason: nearly every course at FIT CTU has its own dedicated page in this system. These pages offer students comprehensive information about each course for a given semester. This information includes course introductions, classification rules, lists of teachers, and most importantly, study materials such as lecture slides, lab materials, and homework assignments.

Functionalities. Central to the Courses system is the search bar, which enables users to find courses by their code or name. The search functionality operates in real-time, providing results as users type their queries.

The search results display relevant information for each course, including its code, name, tags, and a link to its page. Clicking on a result redirects the user to either the course page on the Courses system itself or to the Moodle page [11]. If a link is absent, it indicates that either the course does not have a page or that the page has not been set up yet. Tags provide additional details such as the semesters in which the course is offered, the program level (bachelor, master, or doctoral) and the faculty department. The course code can be considered the tag as well.

[11]Alternative system, which was used before the Courses system was introduced.

For users who prefer not to use interactive search, they can directly access a course page by entering the exact course code in the URL: `https://courses.fit.cvut.cz/{code}`.

A valuable addition to the Courses system is a personalized homepage for logged-in students. Here, they can find upcoming events from the calendar and a list of courses they are currently enrolled in for the semester. Over time, course classifications have also been included next to each course in this list.

Interface. The system keeps things simple with a palette of gray shades, giving it a clean and minimalist look, visible in Figure 2.2. The layout is well-organized, making it easy to find what you need without any clutter. Both the mobile and desktop versions are user-friendly, with clear navigation links highlighted in light blue and underlined when hovered.

It's crucial for users to easily access personalized information. Highlighting upcoming events related to courses – like event type, timing, and location – is valuable, offering users a clear overview of what's ahead. Incorporating such a feature into the WhereIS user interface, and potentially mirroring the event layout seen in Figure 2.3 already familiar to students, could enhance user experience.

Students appreciate the Courses system for its straightforward design and easy access to essential information.

2.3.2 UserMap

Description. UserMap [54] is a CTU system designed to facilitate searching for individuals by their name, surname, and/or username. It primarily serves students looking to find teachers and non-academic staff of the university. Access to search for students and other university partners is restricted to university employees.

Functionalities. The central feature of the system is its search bar, where users enter their query and then click the search button to retrieve results. The results list includes the person's name, last name, academic degrees, and the CTU entity/faculty they belong to. Users can choose how many results to display, with the default value set to 15. To view additional results, users navigate to the next page.

Clicking on a person's name redirects the user to a new page displaying detailed information about that individual. This information typically includes a photo, a list of CTU roles and departments, contact information (room, phone, email, and web page), and an ORCID ID number [12].

However, the search process can be somewhat challenging as specific queries are required to find individuals. For example, to find Ing. Ladislav Vagner, Ph.D., users need to enter "Vagner" or "Ladislav Vagner". Queries like "Ladislav", "xvagner", or "ladislav.vagner" will not yield results.

[12]The ORCID is a non-proprietary alphanumeric code to uniquely identify authors and contributors of scholarly communication.

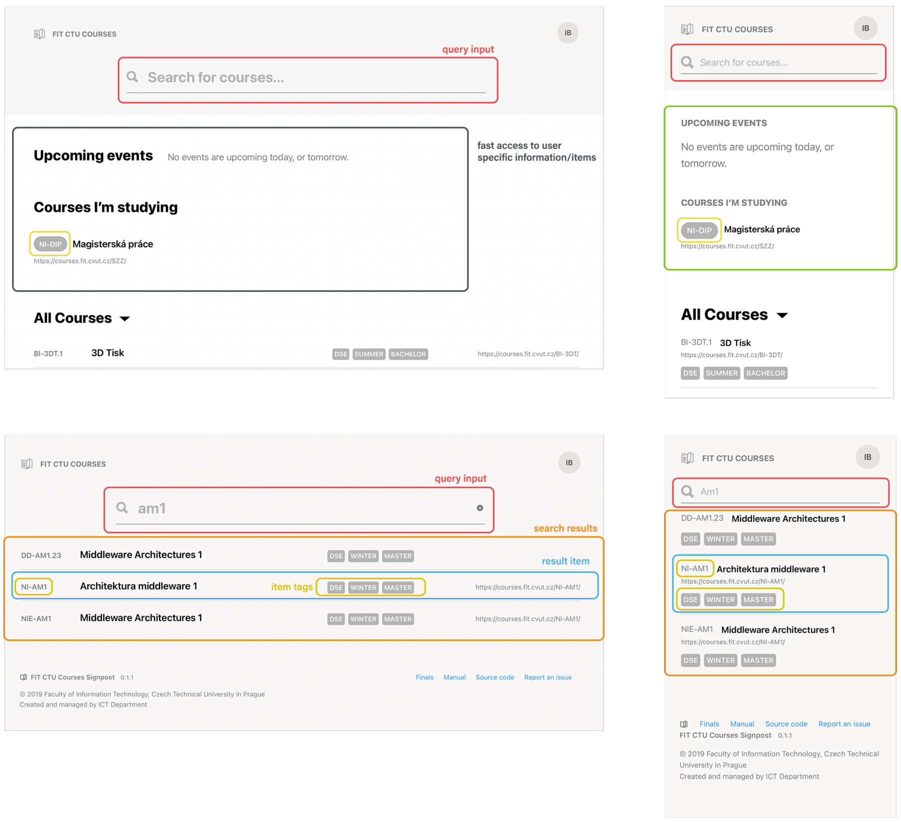

FIT CTU COURSES
query input
Search for courses...
fast access to user specific information/items
Upcoming events
No events are upcoming today, or tomorrow.
Courses I'm studying
NI-DIP Magisterská práce
All Courses
BI-3DT.1 3D Tisk
DSE SUMMER BACHELOR
am1
search results
result item
item tags
DD-AM1.23 Middleware Architectures 1
NI-AM1 Architektura middleware 1
NIE-AM1 Middleware Architectures 1
DSE WINTER MASTER
Finals Manual Source code Report an issue
FIT CTU Courses Signpost 0.1.1

Figure 2.2: Courses – UI structure

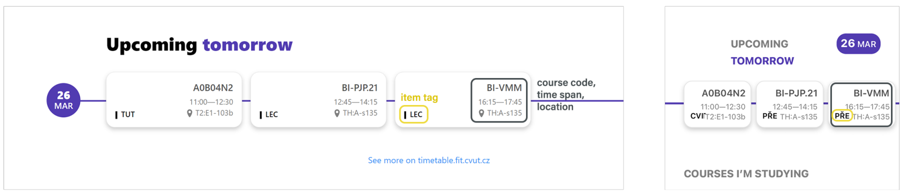
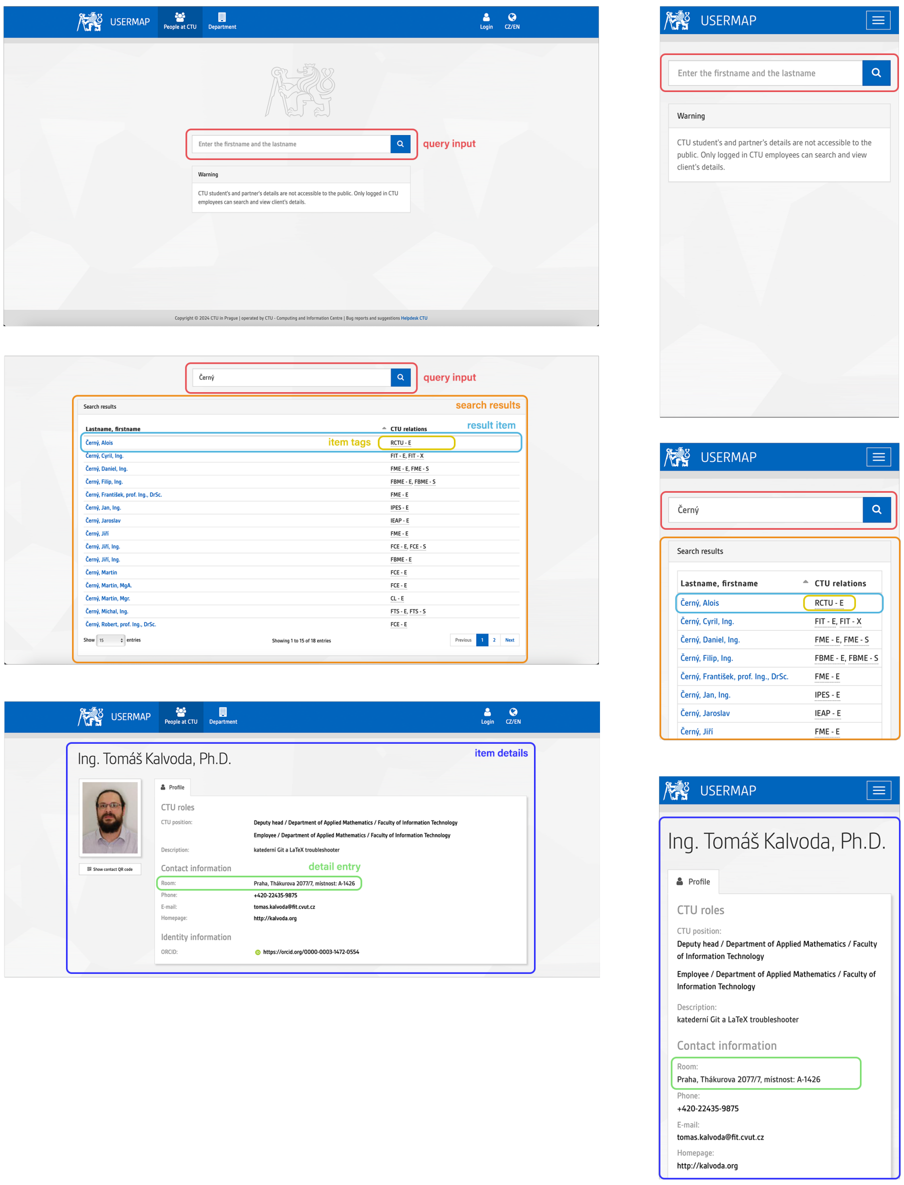

Upcoming tomorrow
26 MAR
A0B04N2 11:00—12:30 T2:E1-103b TUT
BI-PJP.21 12:45—14:15 TH:A-s135 LEC
item tag
BI-VMM 16:15—17:45 TH:A-s135
course code, time span, location
See more on timetable.fit.cvut.cz
UPCOMING TOMORROW
COURSES I'M STUDYING

Figure 2.3: Courses – calendar UI

Interface. The interface adopts a simplistic design, using the blue color from the CTU Brand Style Guide [13]. While the layout is well-structured, the absence of an interactive search system is a notable drawback. Additionally, users must navigate the system by redirecting to new pages, which can hinder the overall user experience. Query input area uses both magnifying icon and microcopy to

[13]CTU blue color or CTU blue, is a blue color from the CTU Brand Style Guide [55], which is recommended to use in most of the cases to support uniform and consistent visual style of university systems. The color is close to #0065bd.

help user better distinguish and use the search functionality. The UI elements of both desktop and mobile versions are analysed in Figure 2.4.



Warning

CTU student's and partner's details are not accessible to the public. Only logged in CTU employees can search and view client's details.

Copyright © 2024 CTU in Prague | operated by CTU - Computing and Information Centre | Bug reports and suggestions Helpdesk CTU

Černý

Search results

Lastname, firstname	CTU relations
Černý, Alois	RCTU - E
Černý, Cyril, Ing.	FIT - E, FIT - X
Černý, Daniel, Ing.	FME - E, FME - S
Černý, Filip, Ing.	FBME - E, FBME - S
Černý, František, prof. Ing., DrSc.	FME - E
Černý, Jan, Ing.	IPES - E
Černý, Jaroslav	IEAP - E
Černý, Jiří	FME - E
Černý, Jiří, Ing.	FCE - E, FCE - S
Černý, Jiří, Ing.	FBME - E
Černý, Martin	FCE - E
Černý, Martin, MgA.	FCE - E
Černý, Martin, Mgr.	CL - E
Černý, Michal, Ing.	FTS - E, FTS - S
Černý, Robert, prof. Ing., DrSc.	FCE - E

Show 15 entries — Showing 1 to 15 of 18 entries — Previous 1 2 Next

Ing. Tomáš Kalvoda, Ph.D.

CTU roles

CTU position:	Deputy head / Department of Applied Mathematics / Faculty of Information Technology
	Employee / Department of Applied Mathematics / Faculty of Information Technology
Description:	katederní Git a LaTeX troubleshooter

Contact information

Room:	Praha, Thákurova 2077/7, místnost: A-1426
Phone:	+420-22435-9875
E-mail:	tomas.kalvoda@fit.cvut.cz
Homepage:	http://kalvoda.org

Identity information

ORCID:	https://orcid.org/0000-0003-1472-0554

Figure 2.4: UserMap – UI structure

2.3.3 KOS

Description. KOS [56], the main university information system, serves as a central hub for managing various aspects of student life. It's the one of the systems shared across different faculties of the university, and its primary aim is to streamline the study agenda.

Functionalities. KOS offers a wide range of functions, with some of the most important being:

- Course enrollment
- Timetable creation
- Registration for tests and exams

Students can also view their study results for each semester and subject, update their contact information, and check their study-related payments. While the system is robust, many features are intended for use by teachers and employees, so they might not be readily visible to students.

KOS also provides an overview of all courses offered at CTU. This is helpful when students are selecting non-mandatory courses to enroll in. To access this overview, students need to navigate to the course enrollment section, where they can filter courses by semester, faculty, and optionally, department. From there, they can view detailed information about each course, including language, credit count, completion type, and instructors. However, it lacks direct links to study materials, such as those found on the Courses page.

Interface. The system's primary color is CTU blue, and while it has undergone redesigns in the past, remnants of the old layout still persist. The search process for available courses is not straightforward, as users must navigate through filters to find relevant results. Sometimes, it's easier to browse courses through the study plan section. The interface is visible in Figure 2.5.

2.3.4 Agata

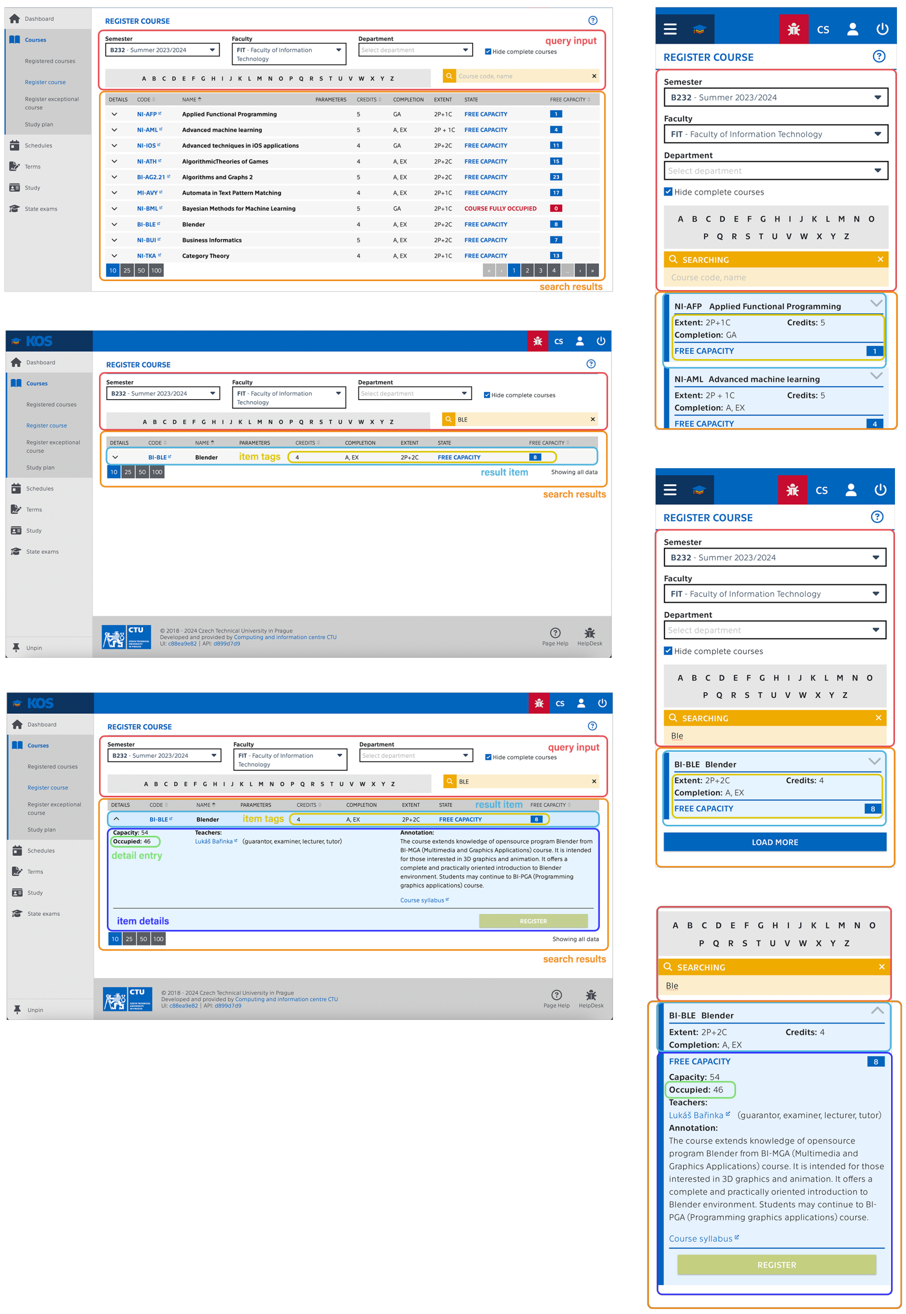
Description. Agata (CTU canteen menus) [57] is a system operated by the Service Facilities Administration (SFA) of the Czech Technical University in Prague. It provides information about the canteens and buffets managed by SFA, including menus and catering account balances.

Functionalities. On the main page, users can find a list of all food serving places, such as canteens and buffets. Each location may have an updated menu for breakfast, lunch, or dinner, depending on the time of day. Clicking on a specific canteen reveals the daily menu, which includes a list of meals categorized by counters, along with details such as price, weight, allergens, and photos if available.

Weekly menus provide basic information about the meals, including name, category, and weight, and may change frequently. They are typically available for 1 to 2 weeks in advance.

The system does not include search functionality, as its primary purpose is to provide up-to-date information about the meals available.

Three mobile apps have been developed to enhance the user experience, but none of them offer search functionality. However, one app significantly



Semester: B232 - Summer 2023/2024
Faculty: FIT - Faculty of Information Technology
Department: Select department
☑ Hide complete courses
A B C D E F G H I J K L M N O P Q R S T U V W X Y Z
Course code, name

DETAILS	CODE	NAME	PARAMETERS	CREDITS	COMPLETION	EXTENT	STATE	FREE CAPACITY
	NI-AFP	Applied Functional Programming		5	GA	2P+1C	FREE CAPACITY	1
	NI-AML	Advanced machine learning		5	A, EX	2P + 1C	FREE CAPACITY	4
	NI-IOS	Advanced techniques in iOS applications		4	GA	2P+2C	FREE CAPACITY	11
	NI-ATH	AlgorithmicTheories of Games		4	A, EX	2P+2C	FREE CAPACITY	15
	BI-AG2.21	Algorithms and Graphs 2		5	A, EX	2P+2C	FREE CAPACITY	23
	MI-AVY	Automata in Text Pattern Matching		4	A, EX	2P+1C	FREE CAPACITY	17
	NI-BML	Bayesian Methods for Machine Learning		5	GA	2P+1C	COURSE FULLY OCCUPIED	0
	BI-BLE	Blender		4	A, EX	2P+2C	FREE CAPACITY	8
	NI-BUI	Business Informatics		5	A, EX	2P+2C	FREE CAPACITY	7
	NI-TKA	Category Theory		4	A, EX	2P+1C	FREE CAPACITY	13

10 25 50 100 « ‹ 1 2 3 4 … › »

BLE
Showing all data

Capacity: 54
Occupied: 46
Teachers:
Lukáš Bařinka (guarantor, examiner, lecturer, tutor)
Annotation:
The course extends knowledge of opensource program Blender from BI-MGA (Multimedia and Graphics Applications) course. It is intended for those interested in 3D graphics and animation. It offers a complete and practically oriented introduction to Blender environment. Students may continue to BI-PGA (Programming graphics applications) course.
Course syllabus

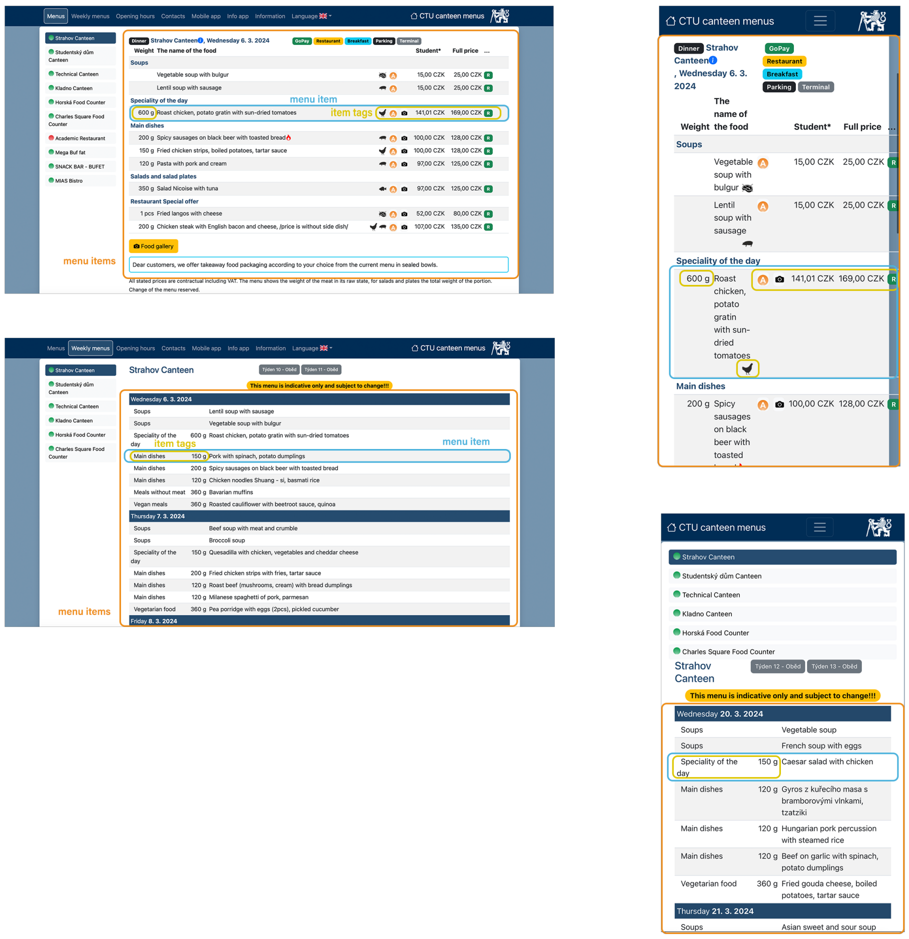
Extent: 2P+1C Credits: 5
Completion: GA

© 2018 - 2024 Czech Technical University in Prague
Developed and provided by Computing and information centre CTU
UI: c88ea9e82 | API: d899d7d9

Figure 2.5: KOS – UI structure

improves the user experience by providing well-structured daily menus with photos, optimized for both vertical and horizontal layouts.

Interface. The original website design, seen in Figure 2.6, appears outdated and lacks consistency, but the layout is straightforward and serves its primary purpose effectively. One of the mobile apps, developed by FIT CTU student Petr Laštovička for Android, visible in Figure 2.7, significantly improves the user experience compared to the original website. It features well-structured daily menus with photos, optimized for both vertical and horizontal layouts. Another honored mention should be an iOS app from Vaclav Halik, seen in Figure 2.8.



Strahov Canteen
Studentský dům Canteen
Technical Canteen
Kladno Canteen
Horská Food Counter
Charles Square Food Counter
Academic Restaurant
Mega Buf fat
SNACK BAR - BUFET
MIAS Bistro

Dinner Strahov Canteen, Wednesday 6. 3. 2024 GoPay Restaurant Breakfast Parking Terminal

Weight	The name of the food	Student*	Full price
Soups			
	Vegetable soup with bulgur	15,00 CZK	25,00 CZK
	Lentil soup with sausage	15,00 CZK	25,00 CZK
Speciality of the day			
600 g	Roast chicken, potato gratin with sun-dried tomatoes	141,01 CZK	169,00 CZK
Main dishes			
200 g	Spicy sausages on black beer with toasted bread	100,00 CZK	128,00 CZK
150 g	Fried chicken strips, boiled potatoes, tartar sauce	100,00 CZK	128,00 CZK
120 g	Pasta with pork and cream	97,00 CZK	125,00 CZK
Salads and salad plates			
350 g	Salad Nicoise with tuna	97,00 CZK	125,00 CZK
Restaurant Special offer			
1 pcs	Fried langos with cheese	52,00 CZK	80,00 CZK
200 g	Chicken steak with English bacon and cheese, /price is without side dish/	107,00 CZK	135,00 CZK

Dear customers, we offer takeaway food packaging according to your choice from the current menu in sealed bowls.

All stated prices are contractual including VAT. The menu shows the weight of the meat in its raw state, for salads and plates the total weight of the portion. Change of the menu reserved.

Strahov Canteen Týden 10 - Oběd Týden 11 - Oběd

This menu is indicative only and subject to change!!!

Wednesday 6. 3. 2024		
Soups		Lentil soup with sausage
Soups		Vegetable soup with bulgur
Speciality of the day	600 g	Roast chicken, potato gratin with sun-dried tomatoes
Main dishes	150 g	Pork with spinach, potato dumplings
Main dishes	200 g	Spicy sausages on black beer with toasted bread
Main dishes	120 g	Chicken noodles Shuang - si, basmati rice
Meals without meat	360 g	Bavarian muffins
Vegan meals	360 g	Roasted cauliflower with beetroot sauce, quinoa
Thursday 7. 3. 2024		
Soups		Beef soup with meat and crumble
Soups		Broccoli soup
Speciality of the day	150 g	Quesadilla with chicken, vegetables and cheddar cheese
Main dishes	200 g	Fried chicken strips with fries, tartar sauce
Main dishes	120 g	Roast beef (mushrooms, cream) with bread dumplings
Main dishes	120 g	Milanese spaghetti of pork, parmesan
Vegetarian food	360 g	Pea porridge with eggs (2pcs), pickled cucumber
Friday 8. 3. 2024		

Dinner Strahov Canteen, Wednesday 6. 3. 2024 GoPay Restaurant Breakfast Parking Terminal

Weight	The name of the food	Student*	Full price
Soups			
	Vegetable soup with bulgur	15,00 CZK	25,00 CZK
	Lentil soup with sausage	15,00 CZK	25,00 CZK
Speciality of the day			
600 g	Roast chicken, potato gratin with sun-dried tomatoes	141,01 CZK	169,00 CZK
Main dishes			
200 g	Spicy sausages on black beer with toasted	100,00 CZK	128,00 CZK

Strahov Canteen
Studentský dům Canteen
Technical Canteen
Kladno Canteen
Horská Food Counter
Charles Square Food Counter

Strahov Canteen Týden 12 - Oběd Týden 13 - Oběd

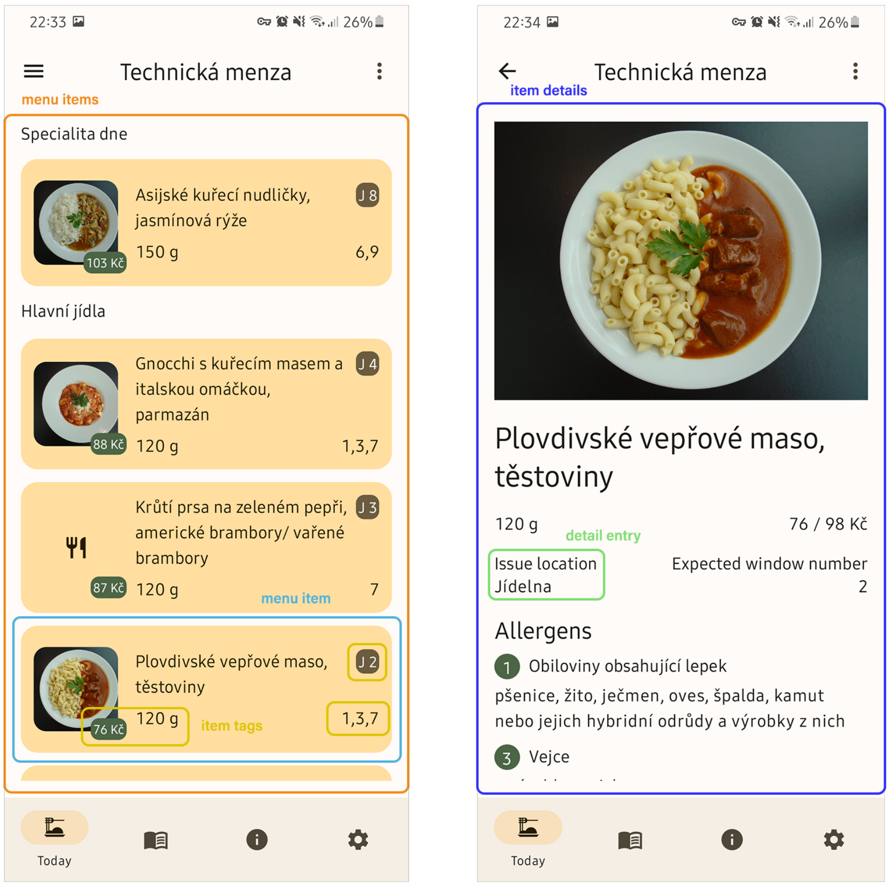
This menu is indicative only and subject to change!!!

Wednesday 20. 3. 2024		
Soups		Vegetable soup
Soups		French soup with eggs
Speciality of the day	150 g	Caesar salad with chicken
Main dishes	120 g	Gyros z kuřecího masa s bramborovými vlnkami, tzatziki
Main dishes	120 g	Hungarian pork percussion with steamed rice
Main dishes	120 g	Beef on garlic with spinach, potato dumplings
Vegetarian food	360 g	Fried gouda cheese, boiled potatoes, tartar sauce
Thursday 21. 3. 2024		
Soups		Asian sweet and sour soup

Figure 2.6: Agata – UI structure

2.3.5 Timetable

Description. Timetable (Fittable) [58] is a system for FIT, offering a more user-friendly alternative to the calendars available in the KOS system. It encompasses timetables for courses, teachers, rooms, and personal events.



22:33

Technická menza

Specialita dne

Asijské kuřecí nudličky, jasmínová rýže — J 8 — 150 g — 6,9 — 103 Kč

Hlavní jídla

Gnocchi s kuřecím masem a italskou omáčkou, parmazán — J 4 — 120 g — 1,3,7 — 88 Kč

Krůtí prsa na zeleném pepři, americké brambory/ vařené brambory — J 3 — 120 g — 7 — 87 Kč

Plovdivské vepřové maso, těstoviny — J 2 — 120 g — 1,3,7 — 76 Kč

22:34

Technická menza

Plovdivské vepřové maso, těstoviny

120 g — 76 / 98 Kč

Issue location
Jídelna

Expected window number
2

Allergens

1 Obiloviny obsahující lepek
pšenice, žito, ječmen, oves, špalda, kamut nebo jejich hybridní odrůdy a výrobky z nich

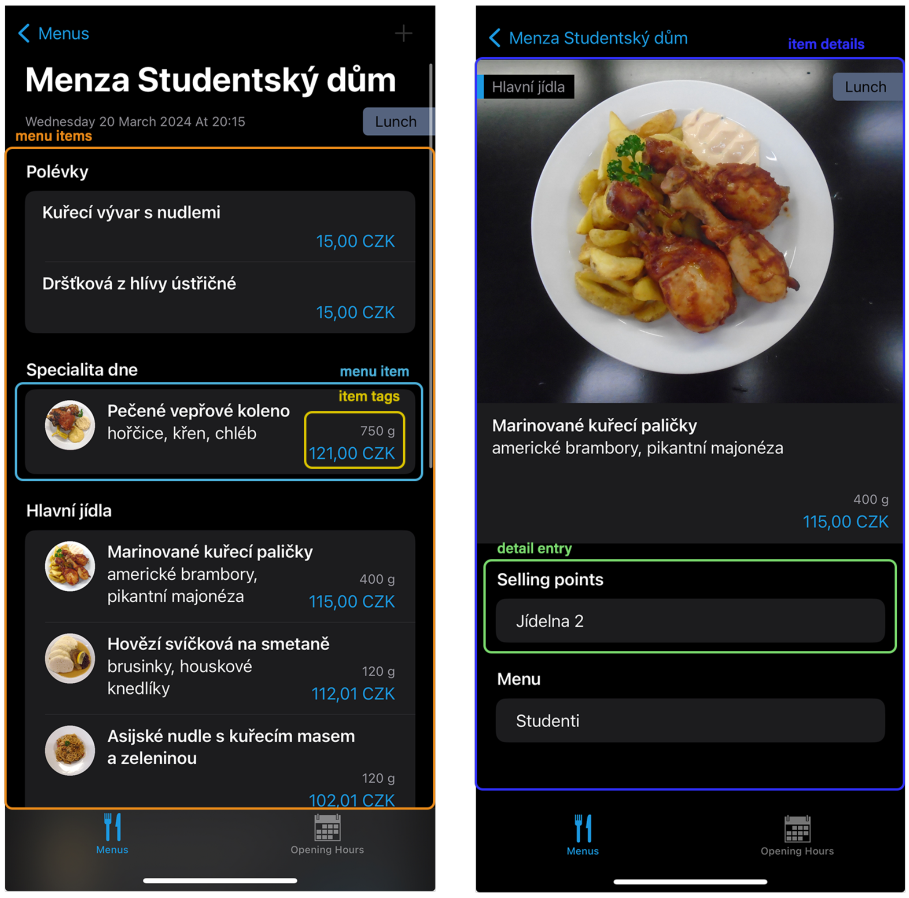
3 Vejce

Figure 2.7: Menu - Android App – UI structure

Functionalities. The system is divided into two systems, visible immediately on the main page. The second system, accessible only to users with a teacher or employee role, is an older original version. The first system, known as “Fittable”, is used by students.

On the main page of “Fittable”, users can view their personal schedule, which includes labs, tutorials, lectures, tests, exams, and other events. Each event preview in the calendar is clickable, providing, in case if course event, detailed information such as course code, time span, location, teachers, capacity, and parallel number. Additionally, users can navigate to calendars of other entities, such as rooms or course teachers, from course event details by clicking on respective links and vice versa.

Another essential functionality is the ability to search for teachers, courses, and rooms and view their timetables. The search capabilities surpass those of UserMap, enabling users to easily locate the timetables they need.



Menza Studentský dům

Wednesday 20 March 2024 At 20:15

Lunch

Polévky

Kuřecí vývar s nudlemi 15,00 CZK

Dršťková z hlívy ústřičné 15,00 CZK

Specialita dne

Pečené vepřové koleno hořčice, křen, chléb 750 g 121,00 CZK

Hlavní jídla

Marinované kuřecí paličky americké brambory, pikantní majonéza 400 g 115,00 CZK

Hovězí svíčková na smetaně brusinky, houskové knedlíky 120 g 112,01 CZK

Asijské nudle s kuřecím masem a zeleninou 120 g 102,01 CZK

Hlavní jídla

Lunch

Marinované kuřecí paličky
americké brambory, pikantní majonéza

400 g
115,00 CZK

Selling points

Jídelna 2

Menu

Studenti

Figure 2.8: Menu – iOS App – UI structure

Interface. The design adopts the CTU blue color scheme. In the mobile version, users see the schedule for the current day, while the desktop version displays the schedule for the entire week. The layout, seen in Figure 2.9, is minimalist, intuitive, and consistent across platforms. Users can customize their experience by enabling color differentiation for different event types and switching between horizontal and vertical layouts. Additionally, the option to filter events by type enhances usability, particularly when timetables are dense.

2.3.6 Projects FIT

Description. Projects FIT [59] is a system that offers a user-friendly approach to managing tasks related to thesis topic selection, submission, and overview for bachelors' and masters' theses FIT students. Essentially, it serves as a FIT-specific version of functionalities already available in the KOS system.

Functionalities. The primary function of the system is to assist students in managing their theses.

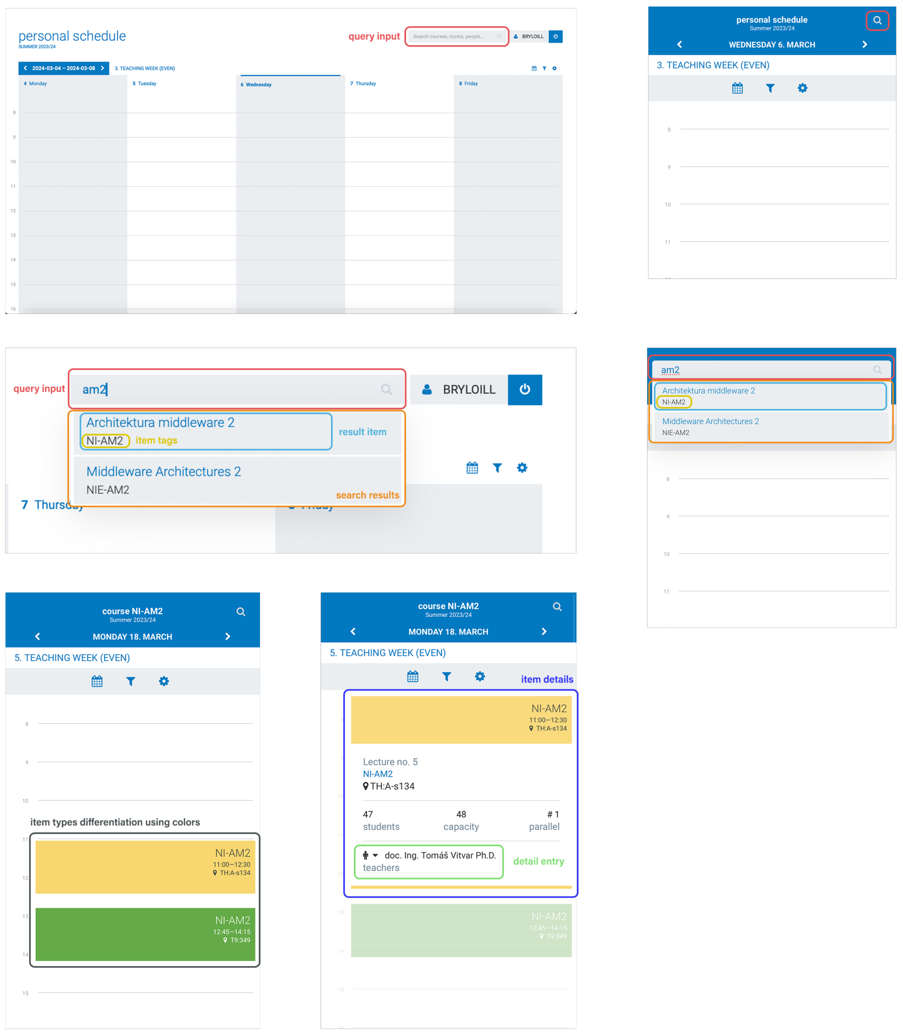


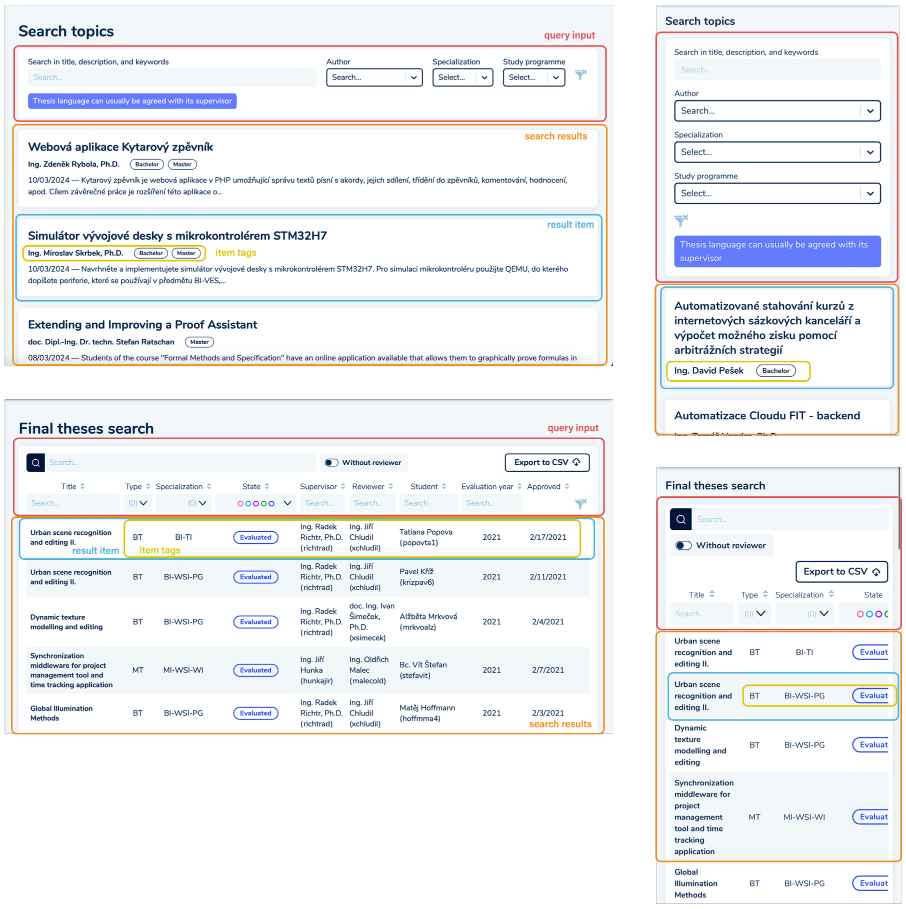
Figure 2.9: Timetable – UI structure

If a student has not yet secured a supervisor, they can search for available thesis topics in the “Search topics” section. By default, all currently assignable topics are visible, allowing users to browse through them easily. The search functionality allows users to search by topic name, description, and keywords, with additional filters available such as topic author, specialization, and study program. Once a user finds an interesting topic, they can view more details by clicking on the search results and navigating to the topic detail page.

Once a topic is assigned to a user following consultation with the supervisor and approval by the faculty department, the user can manage their newly added thesis, and most importantly, submit it.

To gather inspiration from other students’ theses, users can view the list of all theses in the “Search thesis” section. The system offers robust search capabilities, allowing users to search by thesis title, supervisor, reviewer, year, or by applying predefined filters for thesis type, specialization, or state. Clicking on a search result redirects the user to a page with thesis details, similar to what is available when viewing their own thesis.

Interface. The interface is minimalist, straightforward, and utilizes the CTU blue color scheme, visible in Figure 2.10. The search process is user-friendly, offering various methods to find desired results. Search result previews for both topics and theses provide essential information (title, type, specialization, supervisor, reviewer, etc.) for better user understanding. The use of different colors for thesis state tags enhances the distinction between search results, with states such as proposed, assigned, evaluated, etc. The layout is well-structured, placing related information close together.



Thesis language can usually be agreed with its supervisor

Webová aplikace Kytarový zpěvník
Ing. Zdeněk Rybola, Ph.D. Bachelor Master
10/03/2024 — Kytarový zpěvník je webová aplikace v PHP umožňující správu textů písní s akordy, jejich sdílení, třídění do zpěvníků, komentování, hodnocení, apod. Cílem závěrečné práce je rozšíření této aplikace o...

Simulátor vývojové desky s mikrokontrolérem STM32H7
Ing. Miroslav Skrbek, Ph.D. Bachelor Master
10/03/2024 — Navrhněte a implementujete simulátor vývojové desky s mikrokontrolérem STM32H7. Pro simulaci mikrokontroléru použijte QEMU, do kterého dopíšete periferie, které se používají v předmětu BI-VES,...

Extending and Improving a Proof Assistant
doc. Dipl.-Ing. Dr. techn. Stefan Ratschan Master
08/03/2024 — Students of the course "Formal Methods and Specification" have an online application available that allows them to graphically prove formulas in

Automatizované stahování kurzů z internetových sázkových kanceláří a výpočet možného zisku pomocí arbitrážních strategií
Ing. David Pešek Bachelor

Automatizace Cloudu FIT - backend

Title	Type	Specialization	State	Supervisor	Reviewer	Student	Evaluation year	Approved
Urban scene recognition and editing II.	BT	BI-TI	Evaluated	Ing. Radek Richtr, Ph.D. (richtrad)	Ing. Jiří Chludil (xchludil)	Tatiana Popova (popovta1)	2021	2/17/2021
Urban scene recognition and editing II.	BT	BI-WSI-PG	Evaluated	Ing. Radek Richtr, Ph.D. (richtrad)	Ing. Jiří Chludil (xchludil)	Pavel Kříž (krizpav6)	2021	2/11/2021
Dynamic texture modelling and editing	BT	BI-WSI-PG	Evaluated	Ing. Radek Richtr, Ph.D. (richtrad)	doc. Ing. Ivan Šimeček, Ph.D. (xsimecek)	Alžběta Mrkvová (mrkvoalz)	2021	2/4/2021
Synchronization middleware for project management tool and time tracking application	MT	MI-WSI-WI	Evaluated	Ing. Jiří Hunka (hunkajir)	Ing. Oldřich Malec (malecold)	Bc. Vít Štefan (stefavit)	2021	2/7/2021
Global Illumination Methods	BT	BI-WSI-PG	Evaluated	Ing. Radek Richtr, Ph.D. (richtrad)	Ing. Jiří Chludil (xchludil)	Matěj Hoffmann (hoffmma4)	2021	2/3/2021

Figure 2.10: Projects – UI structure

2.3.7 Other Systems

Aside from the systems previously discussed, there are several other platforms utilized by FIT students. One notable example is Grades, a system focused on providing students with their course classifications. Due to its focus on classification-related information and lack of search functionality, it was not further elaborated above.

The remaining systems primarily revolve around specific groups of courses:

- **Progtest** – a platform for submitting programming tasks in various programming languages, as well as quizzes,
- **Marast** – a system managed by the Department of Applied Mathematics, featuring quizzes related to math courses,
- **LearnShell** – a platform offering quizzes on scripting using Bash/Shell.
- **DBS Portal** – a system dedicated to supporting activities related to the BI-DBS course (Database Systems),
- **SOS Portal** – the Student Submission System, utilized by various courses for homework submissions or to support semester projects.
- **and others ...**

These systems are tailored to specific course agendas and are not universally used by all students. Their usage is typically limited to the duration of the enrolled course, and they do not prioritize search functionality.

2.4 Existing Prototype Analysis

The design process of the user interface for an application like WhereIS would be incomplete without analyzing the prototype version created by Bc. T. Heger [1] as part of his bachelor thesis. This section delves deeper into how the app was conceptualized during its initial development stages. We aim to gather information about the user flow in the application, what information is accessible and how it is presented to the user.

2.4.1 Functionality

The prototype allowed users to search for rooms, teachers, and courses. It also featured the ability to find available computer and seminar rooms on campus.

For teachers, users could see their timetable link, office room code, department and contact info.

Course details included a link to the course page, the course timetable, and details about the two nearest upcoming events related to the course.

Room details provided information like the room timetable link, the building floor number, and details about the two closest upcoming events associated with the room.

2.4.2 User Interface

The user was required to log in to the application using their CTU credentials, without which the application was inaccessible. Upon logging in, the user was directed to the home page of the application, seen in Figure 2.11, which featured the main feature of the application: a search query input field. After entering the query, the user had to press the Search button, which redirected them to a new page displaying the search results.

Each search result – whether it be a room, teacher, or course – presented comprehensive information about the subject, seen in Figure 2.12. The search engine results page did not use result surrogates. Typically, one search result occupied the entire screen space, making it difficult to navigate the page when there were multiple search results. Additionally, no visual indicators, such as icons, were used for search detail entries.

If the user's query was not effective, they had to return to the home page and repeat the operation from the beginning. The functionality to conduct a repeated search while remaining on the search page was absent. Additionally, the option to filter the search results was absent.

The user interface was primarily designed for desktop screen sizes and proportions, making it almost impossible to use on mobile platforms due to the sizes of UI elements.

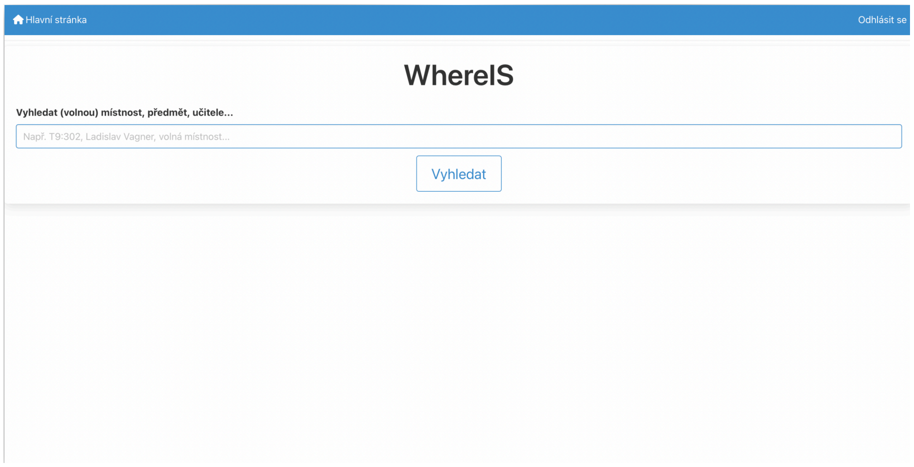

Hlavní stránka
Odhlásit se
WhereIS
Vyhledat (volnou) místnost, předmět, učitele...
Např. T9:302, Ladislav Vagner, volná místnost...
Vyhledat

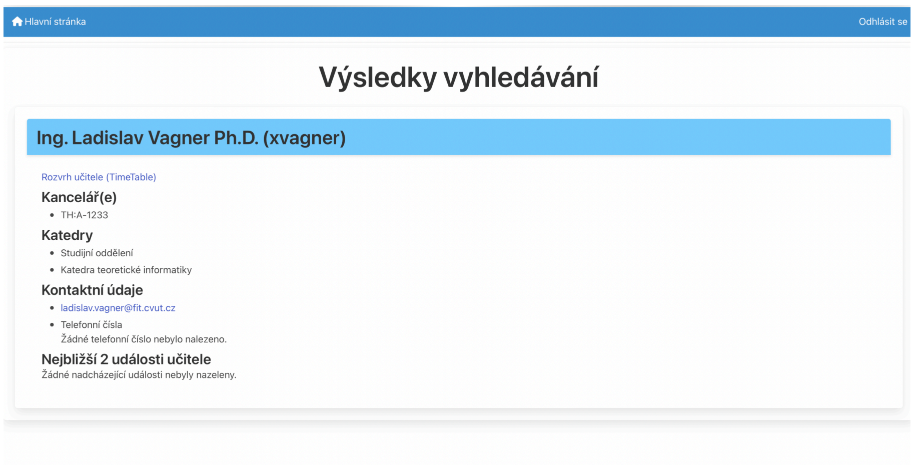
Figure 2.11: Prototype – UI – Home page

2.4.3 Usability Testing and Heuristic Analysis

Bc. T. Heger's bachelor thesis included a prototype evaluation, which encompassed heuristic analysis and usability testing.

The heuristic analysis uncovered several issues, including:

- Lack of navigation,
- absence of status indication during search result loading,



Výsledky vyhledávání

Ing. Ladislav Vagner Ph.D. (xvagner)

Rozvrh učitele (TimeTable)

Kancelář(e)
- TH:A-1233

Katedry
- Studijní oddělení
- Katedra teoretické informatiky

Kontaktní údaje
- ladislav.vagner@fit.cvut.cz
- Telefonní čísla
  Žádné telefonní číslo nebylo nalezeno.

Nejbližší 2 události učitele
Žádné nadcházející události nebyly nazeleny.

Figure 2.12: Prototype – UI – Teacher search result

- limited use of icons,
- lack of advanced search options for experienced users,
- absence of a help page for users.

Usability testing revealed additional insights and proposals. Ones focusing solely on user interface-related problems are:

- The need for search functionality to be accessible on the results page to avoid users having to navigate back to the main page,
- challenges posed by the large number of search results displayed on a single page

2.4.4 Conclusion

The analysis of the prototype version by Bc. T. Heger underscores the importance of refining the user interface for WhereIS. This section provided insights into initial conceptualization and identified key areas for improvement, including navigation, visual indicators, and search functionality accessibility. The thesis will address these shortcomings, aiming to enhance usability and functionality for users.

2.5 User Requirements Questionnaire

To gain insights into the target user group, as described in Section 1.5.1.3, a User Requirements Questionnaire was conducted in cooperation with Bc. Tomáš Heger.

The questionnaire was distributed through the primary information channel commonly used by FIT students – the faculty Discord[14] server.

The questionnaire consisted of three parts:

- **Demographic Questions**
- **Current Experiences and Ideas for Improvement**
- **Proposed Functionalities**

The complete questionnaire can be found in the thesis Appendix B.

We have selected the most notable results, which will be discussed in this section.

2.5.1 Demographic Questions

The majority of respondents (66 %) are bachelor students, while 20 % are master students. Additionally, 33 % have been part of FIT for less than a year, while nearly half of the respondents have been at FIT for 3 years or more. This diverse representation ensures relevance from both perspectives: those who are new to FIT and those who are more experienced and may have established opinions on the current situation.

2.5.2 Current Experiences and Ideas for Improvement

Among all the systems provided in the questionnaire, the Courses system is the most commonly used. Following closely behind are KOS and Agata, with Timetable and UserMap coming next.

The Courses system is also the most beloved, with no responses rating satisfaction with the system lower than 4. Timetable comes in second place, with the majority of ratings at 4. KOS follows, while Agata and UserMap received more mixed responses, with Agata averaging a rating of 3 and UserMap showing an almost even distribution of ratings.

System usage is almost evenly divided between desktop and mobile versions.

From the open-ended questions, there are some interesting findings as well:

- some systems not mentioned in the questionnaire but used by students include Grades, FitWiki, Moodle, mobile apps for canteen menus, and systems used to support certain courses (Progtest, Marast, etc.),
- the biggest frustrations mentioned include the fact that systems are separated and sometimes it's difficult to even discover their existence. Additionally, users find the systems not straightforward enough and sometimes behave unexpectedly,
- respondents find it challenging to navigate the systems, find available rooms, access better FAQs, access information about all course teachers, and view current canteen opening hours,

[14]Discord is a voice, video, and text communication service utilized by over a hundred million people to socialize and communicate with friends and communities

- respondents expressed a desire for the ability to compare and have an overview of search entities, access floor plans, navigate to rooms, view weekly menus, to better interconnect systems (KOS, UserMap, Timetable), or have all information in one centralized location.

2.5.3 Proposed Functionalities

According to the open-ended questions results, there is significant interest among respondents for several proposed functionalities:

- **Search for Free seminar or computer Rooms**: 86 % of respondents expressed a desire for this feature, with 66 % rating it as highly desirable (5) and 20 % rating it as desirable (4).
- **Fast access to upcoming Events**: 80 % of respondents indicated that they would like quick access to an overview of their upcoming events or schedule.
- **Comparison of canteen Menus**: 73 % of respondents find it useful to have the ability to compare canteen menus.
- **Additional room information**: 66 % of respondents stated that they would find it helpful to have additional information about rooms, such as the floor number, floor plan, and address.

Regarding the ability to save entities as “favourites” for quick access:

- **Canteens and Courses** were the most desired entities to be saved as favourites, followed by Free Rooms and Teachers.

For accessibility settings:

- The most requested functionality is **dark mode**, with 86 % of respondents rating it as highly desirable (4 or 5). Other accessibility settings received mixed responses, averaging around a rating of 3.

Some additional user ideas mentioned in the last open-ended question include:

- persistence of user settings such as language and dark mode,
- information about dormitories,
- access to faculty social events,
- ability to search for free rooms in the future by setting the date and time span.

2.5.4 Conclusion

The survey results highlight the primary issue with the current systems: the lack of interconnection among various systems, coupled with unexpected behavior in faculty systems and difficulty in finding necessary information.

Users expressed a desire for a single platform where they can search, compare, and have an overview of different entities within the faculty.

Interestingly, the most beloved systems among users are Courses and Timetable, indicating that incorporating design choices from these systems into the WhereIS interface design phase could be beneficial.

The proposed ideas in the third section received positive feedback from respondents, suggesting that implementing these functionalities could enhance user experience.

2.6 Conclusion

Based on the analysis of existing faculty and university systems, the application prototype, and the questionnaire results, it's evident that the prototype's functionality scope is not sufficient. Therefore, it should be expanded to include all the new search items and functionalities we've discovered in this phase.

The next Chapter 3 will leverage the insights gathered to advance the design process of the new user interface for WhereIS application.

Chapter **3**

Design

In this chapter, we will outline the detailed specification of the WhereIS user interface solution. This specification will be based on the analysis conducted in previous chapters, incorporating inputs from potential users, and utilizing selected methodologies discussed earlier. The chapter will conclude by presenting a lo-fi prototype of the application.

3.1 Description

3.1.1 Project Description

Product Statement: The **web application** is designed to **ease the lives, by aggregating information from various school information systems**, of **students at our faculty, particularly newcomers**. a key feature of this application is a search tool that delivers relevant results based on user queries.

Business Requirements:

- **Save users' time**
- **Simplify the retrieval of information within the faculty**
- **Consolidate information from multiple information systems into a single location**
- **Serve as the initial step in searching for information at the faculty, similar to Google**

Constraints:

- Access to the application, as per CTU policy, have only users with a CTU account,
- the set of available data and its quality are limited by the data quality in university, faculty and external systems,
- the application is targeted exclusively at FIT students.

3.1.2 Personas

As described in Chapter 1, three persona types have been defined. The primary user persona is a first-year student, who may face challenges with the Czech language as they are a foreigner. This user group is often the most disoriented when navigating faculty and university systems, as well as the campus. The second group consists of users who are already familiar with all systems due to their extensive experience as faculty members. The third group comprises non-academic employees who have a CTU account, as access to the application is restricted to users with these credentials.

Persona A: First-year student at FIT

- **Name:** Irvan
- **Age:** 20
- **Gender:** Male
- **Interests:** Computer games, programming, aesthetic photos on Instagram, 3D graphics
- **Typical Day:** Wakes up almost every morning for school, hesitant to ask older peers about the exact location of the room his course is taught. Spends the whole day in lectures or self-study. If he has a free afternoon, he goes to the canteen for lunch. In the evening, he returns to his dorm, plays computer games for an hour, prepares for the next day, and goes to sleep. On weekends, he likes to visit the Old Town and take photos of architecture.
- **Background:** Grew up in Ukraine, speaks Czech poorly. Came to the Czech Republic for better education and life. a diligent 1 year student of bachelor program.

Persona B: Czech student and teacher at FIT

- **Name:** Josef
- **Age:** 24
- **Gender:** Male
- **Interests:** C++, web application development, teaching younger students, mathematics
- **Typical Day:** Goes to work daily, studies and teaches at FIT on selected days. Prefers to save money and cannot cook, hence eats at the university canteen. As a vegan, he sometimes struggles to find the right canteen that caters to his diet. Enjoys having a beer in the evening with his students and classmates. In his remaining free time, he prepares teaching materials, studies, does programming, and responds to students' emails.
- **Background:** Graduated from a prestigious high school in Prague, showed great talent in mathematics from a young age. Interested in computer science, he chose to study at FIT, where he now also teaches.

Persona C: Administrative staff of FIT

- **Name:** Hilda
- **Age:** 53
- **Gender:** Female
- **Interests:** Knitting, cooking shows, the TV series “Ulice”, childcare
- **Typical Day:** Prepares breakfast for her children, then heads to work. Makes coffee with colleagues and chats about the latest episode of “Ulice”. Once or twice a week, she handles the study department’s agenda, including enrolling in or canceling courses, issuing documents, approving requests, or forwarding them to superiors. She is proficient in all school systems and knows the study regulations. Follows a diet, so she brings her lunch in containers. Leaves work a bit early to cook dinner for her children.
- **Background:** Dropped out of university due to her first marriage. Initially worked in a restaurant, then took a re-qualification course in Microsoft Office, and succeeded in getting a job at the FIT CTU.

3.1.3 User Goals

After analyzing the responses from the user questionnaire, examining existing faculty systems, evaluating external systems, and reviewing Bc. T. Heger’s prototype, we have identified the primary User Goals of the application. These goals, outlined below, encapsulate the key objectives that users aim to achieve when utilizing the application.

User Goals:

1. Log in to / log out of the application.
2. Find contact information, office location, and timetable of employees.
3. Search for upcoming events related to employees, courses and rooms.
4. Locate a course page, study materials, and course timetable.
5. Find an available study room.
6. Look up the location of rooms within the faculty.
7. Search for canteens, their menus, and locations.
8. To have a fast access to user-defined subset of search items.
9. Show a quick overview of the user’s timetable.
10. Find contact information of CTU dormitories.
11. Search for the information about FIT departments, such as it’s employees, courses taught and location.

12. Locate the buildings' addresses by their code.
13. Have an overview of upcoming faculty social events and it's detailed information.
14. Find out where and when the given meal will be served.
15. Have an access to the application usage guide.
16. Switch language.

3.1.4 Use Cases

The list of Use Cases presented below is grouped by the pages on which they appear. The application will consist of a total of five pages: the Login page, Home page, Search Results page (SERP), Result Detail page and Help page.

Use Cases:

1. *Use Cases* on the **Login page**:
   a) Login to the application **(UG 1)**
   b) Switch language **(UG 16)**
   c) Get to help page **(UG 15)**
2. *Use Cases* on the **Home page**:
   a) Logout from the application **(UG 1)**
   b) Switch language **(UG 16)**
   c) Search for a given query **(UG 2-7, 10-14)**
   d) Filter results by type **(UG 2-7, 10-14)**
   e) View personal timetable **(UG 9)**
   f) View menus of favourite canteens **(UG 8)**
   g) View list of favourite courses **(UG 8)**
   h) Search for available rooms for given date, time and duration **(UG 5)**.
   i) Get to help page **(UG 15)**
3. *Use Cases* on the **Search Results page**:
   a) Logout from the application **(UG 1)**
   b) Switch language **(UG 16)**
   c) Search for a given query **(UG 2-7, 10-14)**
   d) Filter results by type **(UG 2-7, 10-14)**
   e) View the Result's Details page **(UG 2-7, 10-14)**
   f) Get to help page **(UG 15)**
4. *Use Cases* on the **Result's Details page**:

a) **Room (UG 3, 6)**
   i. Click-through to the room's timetable.
   ii. Show the room's floor plan and location.
   iii. Provide the code of the building, in which the room is located.
   iv. Display upcoming events in the room.

b) **Course (UG 3, 4, 8)**
   i. Click-through to the course timetable.
   ii. Click-through to the course study materials.
   iii. Click-through to the course syllabus.
   iv. Display course's department.
   v. Mark course as favourite.
   vi. Display upcoming course events.
   vii. Display employees teaching the course in current semester.

c) **Employee (UG 2, 3)**
   i. Display contact information and the employee's office.
   ii. Click-through to the employee's timetable.
   iii. Click-through to the employee's UserMap page.
   iv. Display of upcoming events for the employee.
   v. Display employee's departments.
   vi. Display employee photo.
   vii. Display courses employee teaches in current semester.

d) **Canteen (UG 7-8)**
   i. Display of the canteen's location.
   ii. Click-through to the Agáta system.
   iii. Display of canteen's location.
   iv. Mark canteen as favourite.
   v. Display of the canteen's current menu.
   vi. Display of the canteen's week menu.

e) **Dormitory (UG 10)**
   i. Display the dormitory location.
   ii. Display the dormitory reception contact information.
   iii. Click-through to the ISKAM [15] page.
   iv. Click-through to the dormitory student club page.
   v. Click-through to the dormitory SFA page.

f) **Meal (UG 14)**
   i. Display the meal's name, price, weight, allergens and photo.
   ii. Display meal's serving canteen and date.

g) **Department (UG 11)**
   i. Display location of the department.

[15]ISKAM is a SFA system for dormitory reservation as well as management of current accommodation.

ii. Display list of employees working under the department.
iii. Display courses offered by the department.

h) **Building (UG 12)**

i. Display building's address.
ii. Display list of rooms in the building.
iii. Display list of departments in the building.

i) **Event (UG 13)**

i. Display event name, date and location of the event.
ii. Click-through to event page.
iii. Provide registration information.

j) **Free Rooms (UG 5)**

i. Display of currently available computer rooms.
ii. Display of currently available seminar rooms.
iii. Display of room details for an available room (Room Result's Detail page).

5. *Use Cases* on the **Help page**:

a) Find out how to use the application **(UG 15)**

3.1.5 Scenarios

Login to the Application (UG 1)
Use Case: The user expects a form with "Username" and "Password" fields on the page to log into the application. If unsure how to log in, a guide or a link to a guide (in Czech or English) on how to use a CTU account to log in is expected on the page. Correct CTU login details should lead to redirection to the application's home page.

- *Scenario No. 1 (The user knows how to log in):*

  1. The user opens the application in a browser.
  2. The system displays a form with "Username" and "Password" fields and a link to login instructions.
  3. The user enters their details and clicks "Log in".
     – If the details are correct, the user is logged in to the application.
     – Otherwise, the user is prompted to re-enter their information.

- *Scenario No. 2 (The user does not know how to log in):*

  1. The user opens the application in a browser.
  2. The system shows "Username" and "Password" fields and a link to login instructions.
  3. The user clicks the link to the instructions, opening in a new tab.
  4. After reading, the user returns to the page, enters their details, and clicks "Log in".
  5. The system verifies the details.

a) If correct, the user is logged in to the application.
b) Otherwise, they are asked to re-enter their information.

Logout from the Application (UG 1)
Use Case: The user expects an option to log out from the application in the header of each page. After clicking this option, the user expects to be logged out, so they will need to log in again.

1. The user clicks on the logout option in the top right corner of the application's header.
2. The system logs out the user, redirects to the initial screen, and displays a message about successful logout.

Language Switching (UG 16)
Use Case: The user expects an option to choose the language in which the application is displayed in the header of each page. Upon selecting this option, the user expects a list of all languages the application supports. After choosing a language, the user expects the application to switch to the selected language and display the same page where the language change was made.

1. The user clicks on the language change option in the application header.
2. The system displays a list of supported languages.
3. The user selects the language they want the application to switch to.
4. The system switches the application to the chosen language and displays the same page where the user made the language change.

Search for a Given Query (UG 2-7, 10-14)
Use Case: The user expects a search query text input field on the home page and the search results page, into which they can enter the query they wish to search. The user anticipates that as soon as they start typing the query, they will be redirected to the results page, where results will gradually appear and update in real-time as the query is being typed, sorting by relevance to the entered query. Once the user completes their query, they expect to see the item they were searching for on the search results page.

1. The user begins to fill in the search query text input field with their query on the home page or results page.
2. As the query is being entered, the system displays the page with current results for the entered query, progressively updating and sorting the results by relevance.
3. The user stops entering their query.
4. The system completes the search, sorting, and displays results corresponding to the user's query (and selected item types).

    a) If no results are found, the user is informed of this situation (e.g., with a sentence "No results were found for the entered query.").

Filtering Results by Type (UG 2-7, 10-14)

Use Case: The user expects the ability to limit the types of items returned by a search on both the home page and the search results page. After selecting specific item types, only results belonging to those types are expected to be displayed.

1. The user conducts a search using a search query text input field.
2. The system shows that all item types are present in the search results.
3. The user clicks on an “All” item type.
4. The system shows the list of all item types and indicates that all of these types will not be present in a search results.
5. User clicks on a item types, which they want to be present in a search results.
6. The system displays only wanted item types for current query.

Displaying Personal Timetable (UG 9)

Use Case: On the home page, the user expects a quick overview of events in their timetable. The user anticipates a message if there are no events in their timetable.

1. The user views the application’s home page.
2. The system displays an overview of the user’s events.
   a) If the user has no events, the system informs them (e.g., “You have no upcoming events.”).

Displaying Menus of Favourite Canteens (UG 8)

Use Case: The user expects to see today’s menu for their favourite canteens on the home page. The user also expects to see a message if no canteen is added to favourites or if a favourite canteen does not have a menu set for the day.

1. The user accesses the application’s home page.
2. The system displays the user’s favourite canteens and their menus for the day.
   a) If the user has not marked any canteen as a favourite, the system notifies the user (e.g., “You have not marked any items as favourite yet.”).
   b) If a canteen has no menu set for the day, the system informs the user (e.g., “The canteen does not have a menu set for today.”).

Displaying Details of Search Results (UG 2-7, 10-14)

Use Case: The user expects to open a specific search result to view its details.

1. The user conducts a search.
2. The system displays the search results.
3. The user clicks on the search result of their choice.

4. The system reveals the details page of the selected result.
    a) Clicking the result title again will hide the search result details page.

Click-through to Room Timetable (UG 3)
Use Case: The user expects the ability to view the timetable of a specific room in the room's search result details.

1. The user searches for the desired room.
2. The system displays the found room.
3. The user opens the search result details page of the found room.
4. The system shows an option to view the room's timetable in the room details.
5. The user clicks on "Room Timetable".
6. The system redirects the user to an external system with the timetable of the selected room.

Displaying Floor Plan and Location of a Room (UG 6)
Use Case: The user expects the room's search result details to include the building's address with a link to maps showing its location. The user also expects detailed information about the room's location, such as the floor within the building where the room is located, or a floor plan highlighting the room, aiding in navigation.

1. The user searches for the desired room.
2. The system displays the found room.
3. The user opens the search result details page of the found room.
4. The system shows room details with the building's address linked to maps, floor number, and a graphical floor plan highlighting the room's location.
    a) If the floor is not provided, system shows and empty image.

Displaying Upcoming Events in a Room (UG 3)
Use Case: The user expects to view upcoming events in a specific room within the room's search result details. The system should notify the user if there are no upcoming events in the room.

1. The user searches for a specific room.
2. The system displays the found room.
3. The user opens the search result details page of the found room.
4. The system presents the room details along with a list of upcoming events in that room.
    a) If there are no upcoming events, the system informs the user with the message (e.g., "Room doesn't have any upcoming events.").

Click-through to Course Timetable (UG 4)
Use Case: The user expects to view the timetable for a specific course within the course's search result details.

- *Scenario No. 1:*
  1. The user searches for the desired course.
  2. The system displays the found course.
  3. The user opens the search result details page of the found course.
  4. The system shows detail of the course.
  5. The user clicks on "Course Timetable" within the course details.
  6. The system redirects the user to an external page with the course timetable.
- *Scenario No. 2:*
  1. The user clicks on the course's code on the home page in the list of favourite entities.
  2. The system displays the course's details page.
  3. The user clicks on "Course Timetable" within the course details.
  4. The system redirects the user to an external page with the course timetable.

Click-through to Course Study Materials (UG 4)
Use Case: The user expects to access study materials, for a specific course within the course's search result details.

- *Scenario No. 1:*
  1. The user searches for the desired course.
  2. The system displays the found course.
  3. The user opens the search result details page of the found course.
  4. The system shows detail of the course.
  5. The user clicks on "Study Materials" within the course details.
  6. The system redirects the user to an external page with the course's study materials.
- *Scenario No. 2:*
  1. The user clicks on the course's code on the home page in the list of favourite entities.
  2. The system displays the course's details page.
  3. The user clicks on "Study Materials" within the course details.
  4. The system redirects the user to an external page with the course's study materials.

Click-through to Course Syllabus (UG 4)
Use Case: The user expects to view the course syllabus in the course's search result details, learning about the course content, assessment methods, tutors, prerequisites, etc.

- *Scenario No. 1:*
  1. The user searches for the desired course.
  2. The system displays the found course.
  3. The user opens the search result details page of the found course.
  4. The system shows detail of the course.
  5. The user clicks on "Syllabus" within the course details.
  6. The system redirects the user to an external page with the course syllabus.
- *Scenario No. 2:*
  1. The user clicks on the course's code on the home page in the list of favourite entities.
  2. The system displays the course's details page.
  3. The user clicks on "Syllabus" within the course details.
  4. The system redirects the user to an external page with the course syllabus.

Displaying Upcoming Events for a Course (UG 3)
Use Case: The user expects the ability to view upcoming events for a course within the course's search result details. If there are no upcoming events for the course, the system should notify the user.

- *Scenario No. 1:*
  1. The user searches for the desired course.
  2. The system displays the found course.
  3. The user opens the search result details page of the found course.
  4. The system shows the course details with upcoming events.
     a) If there are no upcoming events, the system informs the user (e.g., "Course doesn't have any upcoming events.").
- *Scenario No. 2:*
  1. The user clicks on the course's code on the home page in the list of favourite entities.
  2. The system displays the course's details page with upcoming events.
     a) If there are no upcoming events, the system informs the user (e.g., "Course doesn't have any upcoming events.").

Adding a Course to Favourites (UG 8)
Use Case: When viewing the details of a course not already in favourites, the user expects an indication that the course is not in favourites and the option to add it to favourites. The user anticipates a notification of successful addition, and that the course will then be displayed on the home page.

1. The user searches for a course they wish to add to favourites.
2. The system displays the found course.
3. The user opens the search result details page of the found course.
4. The system shows the course details.
5. The user clicks on the unfilled star icon in the course details.
6. The system adds the course to favourites, notifies the user of the successful addition with a notification, fills in the star icon, and displays the course on the home page.

Removing a Course from Favourites (UG 8)
Use Case: When viewing the details of a favourite course, the user expects an indication that the course is a favourite and the option to remove it from favourites. Before removal, the user expects to be asked for confirmation. Upon removal, the system should notify the user of the successful action, and the course will no longer appear on the home page.

- *Scenario No. 1:*
  1. The user clicks on the code of a course on the home page in the list of favourite entities.
  2. The system displays the course's details page.
  3. The user clicks the filled star icon next to the name of the course.
  4. The system displays a pop-up window asking for action confirmation.
  5. The user confirms by clicking "Yes".
     a) If the user clicks "No", the pop-up closes, and the course remains in favourites.
  6. The system removes the course from favourites, notifies the user of successful removal, removes the course from the list of favourite entities on the home page.
- *Scenario No. 2:*
  1. The user searches for a course they wish to remove from favourites.
  2. The system displays the found course.
  3. The user opens the search result details page of the found course.
  4. The system shows the course details.
  5. The user clicks the filled star icon in the course details.

6. The system displays a pop-up window asking for action confirmation.
7. The user confirms by clicking “Yes”.
   a) If the user clicks “No”, the pop-up closes, and the course remains in favourites.
8. The system removes the course from favourites, notifies the user of successful removal, and unfills the star icon.

Displaying Contact Information and Office of an Employee (UG 2)
Use Case: The user expects to view the selected employee’s contact details in the employee’s search result details page. This includes the office assigned to the employee with a link to find this room in the WhereIS system, the departments the employee is associated with a link to find this department in the WhereIS system, and the employee’s email and possibly phone number for contact. If any of these details are missing, the user expects the system to notify them.

1. The user searches for the desired employee.
2. The system displays the found employee.
3. The user opens the search result details page of the found employee.
4. The system shows the employee’s details page with available contact information and the office code with a link to find the room in the WhereIS system and with a link to find this department as well.
   a) If any contact details of the employee are not found/filled out, the system notifies the user (e.g., “This contact information is not provided.”).

Click-through to Employee’s Timetable (UG 2)
Use Case: The user expects to view the timetable of a specific employee within the employee’s search result details.

1. The user searches for the desired employee.
2. The system displays the found employee.
3. The user opens the search result details page of the found employee.
4. The system shows detail of the employee.
5. The user clicks on “Employee’s Timetable” within the employee’s details.
   a) If the employee doesn’t teach any of the listed courses, the click-through option is not provided.
6. The system redirects the user to an external system with the selected employee’s timetable.

Displaying Upcoming Events for an Employee (UG 3)

Use Case: The user expects to see the nearest events in the timetable of a specific employee within the employee's search result details. If the employee has no events in their timetable, the user expects the system to notify them.

1. The user searches for the desired employee.
2. The system displays the found employee.
3. The user opens the search result details page of the found employee.
4. The system shows detail of the employee with a list of upcoming events in the employee's timetable.
   a) If there are no events in the timetable, the system displays an informational message (e.g., "Employee doesn't have any upcoming events.").

Displaying Canteen Location (UG 7)

Use Case: The user expects to find information about the canteen's address and a clickable link to a map showing the canteen's location in the canteen's detail page.

- *Scenario No. 1:*
  1. The user searches for the desired canteen.
  2. The system displays the found canteen.
  3. The user opens the search result details page of the found canteen.
  4. The system shows the canteen's detail with its location (address) as a link to an external map application.
  5. The user clicks on the address link within the canteen's details.
  6. The system redirects the user to an external map application.
- *Scenario No. 2:*
  1. The user clicks on the canteen's name on the home page in the list of favourite canteens.
  2. The system displays the canteen's details page with the location (address) as a link to an external map application.
  3. The user clicks on the address link within the canteen's details.
  4. The system redirects the user to an external map application.

Click-through to the Agáta System (UG 7)

Use Case: The user expects an option for a quick transition to the Agáta dining system (e.g., to check the account balance) on canteen's detail page.

- *Scenario No. 1:*
  1. The user searches for any canteen.
  2. The system displays the found canteen.
  3. The user opens the search result details page of the found canteen.

4. The system shows the canteen's details page.
5. The user clicks on "Canteen Account" within the canteen details page.
6. The system redirects the user to the external Agáta system page.

- *Scenario No. 2:*

   1. The user clicks on the name of a canteen on the home page in the list of favourite canteens.
   2. The system displays the canteen's details page.
   3. The user clicks on "Canteen Account" within the canteen details page.
   4. The system redirects the user to the external Agáta system page.

Displaying the Canteen Today's Menu (UG 7)
Use Case: The user expects to see the menu of a selected canteen or their favourite canteens, including a list of currently served meals with photos. If the canteen does not have today's menu listed at the moment, the user expects the system to notify them.

- *Scenario No. 1:*

   1. The user searches for a specific canteen.
   2. The system displays the found canteen.
   3. The user opens the search result details page of the found canteen.
   4. The system shows the canteen's detail page with the current menu.
      a) If there is no menu currently listed, the system displays an informational message (e.g., "Canteen's today's menu is currently empty.").

- *Scenario No. 2:*

   1. The user views the home page.
   2. The system displays the current menus of all the user's favourite canteens.
      a) If a canteen does not have a current menu listed, the system displays an informational message (e.g., "Canteen's today's menu is currently empty."), for that canteen.

Adding a Canteen to Favourites (UG 8)
Use Case: When viewing the details of a canteen not already in favourites, the user expects an indication that the canteen is not in favourites and the option to add it to favourites. The user anticipates a notification of successful addition, and that the canteen will then be displayed on the home page.

1. The user searches for a canteen they wish to add to favourites.
2. The system displays the found canteen.
3. The user opens the search result details page of the found canteen.

4. The system shows the canteen details.
5. The user clicks on the unfilled star icon in the canteen details.
6. The system adds the canteen to favourites, notifies the user of the successful addition with a notification, fills in the star icon, and displays the canteen on the home page.

Removing a Canteen from Favourites (UG 8)
Use Case: When viewing the details of a favourite canteen, the user expects an indication that the canteen is a favourite and the option to remove it from favourites. Before removal, the user expects to be asked for confirmation. Upon removal, the system should notify the user of the successful action, and the canteen will no longer appear on the home page.

- *Scenario No. 1:*
  1. The user clicks on the name of a canteen on the home page in the list of favourite canteens.
  2. The system displays the canteen's details page.
  3. The user clicks the filled star icon next to the name of the canteen.
  4. The system displays a pop-up window asking for action confirmation.
  5. The user confirms by clicking "Yes".
     a) If the user clicks "No", the pop-up closes, and the canteen remains in favourites.
  6. The system removes the canteen from favourites, notifies the user of successful removal, removes the canteen from the list of favourite canteens on the home page.
- *Scenario No. 2:*
  1. The user searches for a canteen they wish to remove from favourites.
  2. The system displays the found canteen.
  3. The user opens the search result details page of the found canteen.
  4. The system shows the canteen details.
  5. The user clicks the filled star icon in the canteen details.
  6. The system displays a pop-up window asking for action confirmation.
  7. The user confirms by clicking "Yes".
     a) If the user clicks "No", the pop-up closes, and the canteen remains in favourites.
  8. The system removes the canteen from favourites, notifies the user of successful removal, and unfills the star icon.

Click-through to Room Timetable (UG 3)
Use Case: The user expects the ability to view the timetable of a specific room in the room's search result details.

1. The user searches for the desired room.
2. The system displays the found room.
3. The user opens the search result details page of the found room.
4. The system shows an option to view the room's timetable in the room details.
5. The user clicks on "Room Timetable".
6. The system redirects the user to an external system with the timetable of the selected room.

Display the Dormitory Location. (UG 10)
Use Case: The user expects the dormitory's search result details to include the dormitory address with a link to maps showing its location.

1. The user searches for the dormitory.
2. The system displays the found dormitory.
3. The user opens the search result details page of the found dormitory.
4. The system shows dormitory details page with the address link to maps.
5. The user clicks on the address link within the canteen's details.
6. The system redirects the user to an external map application.

Display the Dormitory Reception Contact Information. (UG 10)
Use Case: The user expects the dormitory's search result details to include the dormitory's contact information such as phone number, email etc.

1. The user searches for the dormitory.
2. The system displays the found dormitory.
3. The user opens the search result details page of the found dormitory.
4. The system shows dormitory details page with the contact information.

Click-through to the ISKAM Page (UG 10)
Use Case: The user expects an option for a quick transition to ISKAM system page from dormitory details page.

1. The user searches for the dormitory.
2. The system displays the found dormitory.
3. The user opens the search result details page of the found dormitory.
4. The system shows an option to open the ISKAM system page.
5. The user clicks on "ISKAM".
6. The system redirects the user to an external system.

Click-through to the Dormitory Student Club Page (UG 10)
Use Case: The user expects an option to find out which student club operates in a given dormitory.

1. The user searches for the dormitory.
2. The system displays the found dormitory.
3. The user opens the search result details page of the found dormitory.
4. The system shows an option to open the Student Club page.
5. The user clicks on "Student Club".
6. The system redirects the user to an external web page.

Click-through to the SFA Page (UG 10)
Use Case: The user expects an option for a quick transition to dormitory's SFA page from dormitory details page.

1. The user searches for the dormitory.
2. The system displays the found dormitory.
3. The user opens the search result details page of the found dormitory.
4. The system shows an option to open the SFA system page.
5. The user clicks on "SFA".
6. The system redirects the user to an external system.

Displaying Currently Free Rooms (UG 5)
Use Case: The user expects to find a list of available room types, including a sub-lists for rooms showing names of rooms currently not in use. The user expects to be informed if no rooms of given type are available.

- *Scenario No. 1:*
    1. The user searches for "available room" or a similar term.
    2. The system displays the types of available rooms found.
    3. The user expands the wanted type.
    4. The system shows a list of currently available rooms.
        a) If no rooms of selected type are currently available, the system displays an informational message (e.g., "No free rooms at the moment.").
- *Scenario No. 2:*
    1. The user keeps the default values in free rooms search panel on a home page and presses search button.
    2. The system displays the types of available rooms found.
    3. The user expands the wanted type.
    4. The system shows a list of currently available rooms.

  a) If no rooms of selected type are currently available, the system displays an informational message (e.g., “No free rooms at the moment.”).

Displaying Free Rooms for Given Date, Time and Duration (UG 5)
Use Case: The user expects to find a list of available room types, including a sub-lists for rooms showing names of rooms currently not in use on given date, time and duration. The user expects to be informed if no rooms of given type are available for given search criteria.

1. The user sets the wanted search criteria in free rooms search panel on a home page and presses search button.
2. The system displays the types of available rooms found.
3. The user expands the wanted type.
4. The system shows a list of available rooms on given date, time and for given duration.
   a) If no rooms of selected type are available, the system displays an informational message (e.g., “No free rooms for this search criteria.”).

Displaying Details of an Free Room (UG 5)
Use Case: The user expects to view details of a selected available room.

1. The user does a search for a free room using one of the two mentioned ways.
2. The system displays the types of available rooms.
3. The user expands the desired room type.
4. The system shows a list of available rooms.
5. The user clicks on a specific room if available.
6. The system displays the detail page of the selected room.

Displaying Employees of a Department (UG 11)
Use Case: The user expects to view list of employees working under specific department.

1. The user searches for a department, for which they wants to know the employees.
2. The system displays the found department.
3. The user opens the search result details page of the found department.
4. The system shows the detail of the department.
5. The user expands a list of employees in the detail.
6. The system displays all employees working under the department.

Displaying Courses of a Department (UG 11)
Use Case: The user expects to view list of courses offered by specific department.

1. The user searches for a department, for which they wants to know the offered courses.
2. The system displays the found department.
3. The user opens the search result details page of the found department.
4. The system shows the detail of the department.
5. The user expands a list of courses in the detail.
6. The system displays all courses offered by the department.

Displaying Department Location (UG 11)
Use Case: The user expects to find information about the department's location and a clickable link to a map showing the department's location in the department's detail page.

1. The user searches for desired department.
2. The system displays the found department.
3. The user opens the search result details page of the found department.
4. The system shows the detail of the department with its location (address) as a link to an external map application and building code.
5. The user clicks on the address link within the department's details.
6. The system redirects the user to an external map application.

Displaying Building Location (UG 12)
Use Case: The user expects to find information about the building's location and a clickable link to a map showing the building's location in the building's detail page.

1. The user searches for desired building.
2. The system displays the found building.
3. The user opens the search result details page of the found building.
4. The system shows the detail of the building with its location (address) as a link to an external map application.
5. The user clicks on the address link within the building's details.
6. The system redirects the user to an external map application.

Displaying Rooms in Building (UG 12)
Use Case: The user expects to find list of rooms located in desired building.

1. The user searches for desired building.

2. The system displays the found building.
3. The user opens the search result details page of the found building.
4. The system shows the detail of the building.
5. The user expands a list of rooms in the detail.
6. The system displays all rooms located in the building.

Displaying Departments in Building (UG 12)
Use Case: The user expects to find list of departments located in desired building.

1. The user searches for desired building.
2. The system displays the found building.
3. The user opens the search result details page of the found building.
4. The system shows the detail of the building.
5. The user expands a list of departments in the detail.
6. The system displays all departments located in the building.

Displaying Event Date and Location (UG 14)
Use Case: The user expects to find an essential information about the event.

1. The user searches for event.
2. The system displays the found event.
3. The user opens the search result details page of the found event.
4. The system shows the event detail page of the event with date and location.

Click-through to the Event Page (UG 15)
Use Case: The user expects an option to visit an event page for desired event.

1. The user searches for the event.
2. The system displays the found event.
3. The user opens the search result details page of the found event.
4. The system shows an option to open the Event page.
5. The user clicks on “Event Page”.
6. The system redirects the user to an external web page.

Provide Registration Information for Event (UG 14)
Use Case: The user expects to find the registration information for a desired event.

1. The user searches for event.

2. The system displays the found event.
3. The user opens the search result details page of the found event.
4. The system provides a link to an external registration system.
    a) If an event doesn't need a reservation, the link is not present in the registration field and the message (e.g., "No registration needed.") is showed.

Displaying Meal Serving Information (UG 14)
Use Case: The user expects to find serving information about desired meal. In those information user expects to see meal's student and normal price, serving date, canteen which is serving the meal, photo and allergens.

1. The user searches for desired meal.
2. The system displays the found meal.
3. The user opens the search result details page of the found meal.
4. The system shows the detail of the meal with all information.

3.1.6 Tasks and Task Groups

In this section, all Tasks that can be performed by the user will be listed and then grouped into related categories.

3.1.6.1 Tasks

- Searching for an employee's office
- Searching for departments of an employee
- Searching for an employee's email address
- Searching for an employee's phone number
- Searching for an employee's timetable
- Searching for upcoming events of an employee
- Searching for employee's courses
- Searching for a room
- Searching for a room location
- Searching for floor plans of a room
- Searching for a room timetable
- Searching for upcoming events in a room
- Searching for a course
- Searching for a course timetable
- Searching for a course department
- Searching for a course page (syllabus)
- Searching for course study materials
- Searching for upcoming events for a course
- Searching for course's teachers
- Searching for the location of a canteen
- Searching for a canteen's current menu
- Searching for a canteen's week menu
- Searching for a meal

- Searching for a meal serving information
- Searching for a building
- Searching for building's location
- Searching for rooms in building
- Searching for departments in building
- Searching for a department
- Searching for department's employees
- Searching for department's location
- Searching for a CTU dormitory
- Searching for location of CTU dormitory
- Searching for contact information of CTU dormitory
- Searching for FIT CTU event
- Searching for FIT CTU event's event page
- Searching for FIT CTU event registration information
- Searching for an available room in given time for given duration
- Displaying a personal overview of favourite items
- Marking a canteen as "favourite"
- Marking a course as "favourite"
- Filtering results by item type
- Displaying the detail of a result
- Using help page
- Switching languages
- User login
- User logout
- Reporting a problem in the application

3.1.6.2 Task Groups

As previously mentioned, the system will consist of five pages:

Pages

- Login page – L
- Home page – H
- Search Results page – R
- Result's Details page – D
- Help page – F

Using this notation, we can illustrate how various Tasks will be connected to these five pages:

By visual appearance

- Displaying a short overview of the user's timetable (H)
- Displaying user's favourite items (H)
- Displaying the search results (R)

- Displaying the detail of an search result (R)
  - **Employee**
    - Office (D)
    - Departments (D)
    - Email address (D)
    - Phone number (D)
    - Timetable (D)
    - Upcoming events (D)
    - Photo (D)
    - Courses employee teaches (D)
  - **Course**
    - Study materials (D)
    - Syllabus (D)
    - Timetable (D)
    - Upcoming events (D)
    - Employees teaching course in current semester (D)
    - Adding course to favourites (D)
  - **Room**
    - Timetable (D)
    - Location (D)
    - Building (D)
    - Floor plan (D)
    - Floor number (D)
    - Upcoming events (D)
  - **Canteen**
    - Location (D)
    - Agáta system (D)
    - Today’s Menu (D)
    - Week Menu (D)
    - Adding canteen to favourites (D)
  - **Building**
    - Location (D)
    - Departments (D)
    - Rooms (D)
  - **Department**
    - Location (D)
    - Building (D)
    - Employees (D)
  - **Event**
    - Date (D)
    - Location (D)

    * Event page (D)
    * Registration information (D)
  - **Meal**
    * Canteen (D)
    * Serving date (D)
    * Prices (D)
    * Allergens (D)
    * Weight (D)
    * Photo (D)
  - **Dormitory**
    * Location (D)
    * Contact information (D)
    * Student club URL (D)
    * SFA page (D)
    * ISKAM system (D)
  - **Free rooms**
    * Seminar rooms (H, D)
    * Computer room (H, D)
    * Room detail (H, D)

The list of User Actions offers a comprehensive view of the Tasks users can execute on particular pages. This is further visualized in the Task Graph depicted in Figure 3.1, demonstrating the transitions between pages triggered by specified Actions.

User Actions

- Login
- Logout (H, R, D, F)
- Switch language (L, H, R, F)
- Filter Results by type (H, R)
- Mark Result as “Favourite” (D)
- Unmark Result as “Favourite” (H, D)
- Search for Employee (H, R)
- Search for Room (H, R)
- Search for Course (H, R)
- Search for Canteen (H, R)
- Search for Meal (H, R)
- Search for Building (H, R)

- Search for Department (H, R)
- Search for Dormitory (H, R)
- Search for FIT CTU event (H, R)
- Search for Free Computer Room (H, R)
- Search for Free Seminar Room (H, R)
- Search for Free Computer and Seminar Rooms from given date and time and with given duration (H)
- Read a Help page (L, H, R, F)
- Report a problem in the application (L, H, R, F)

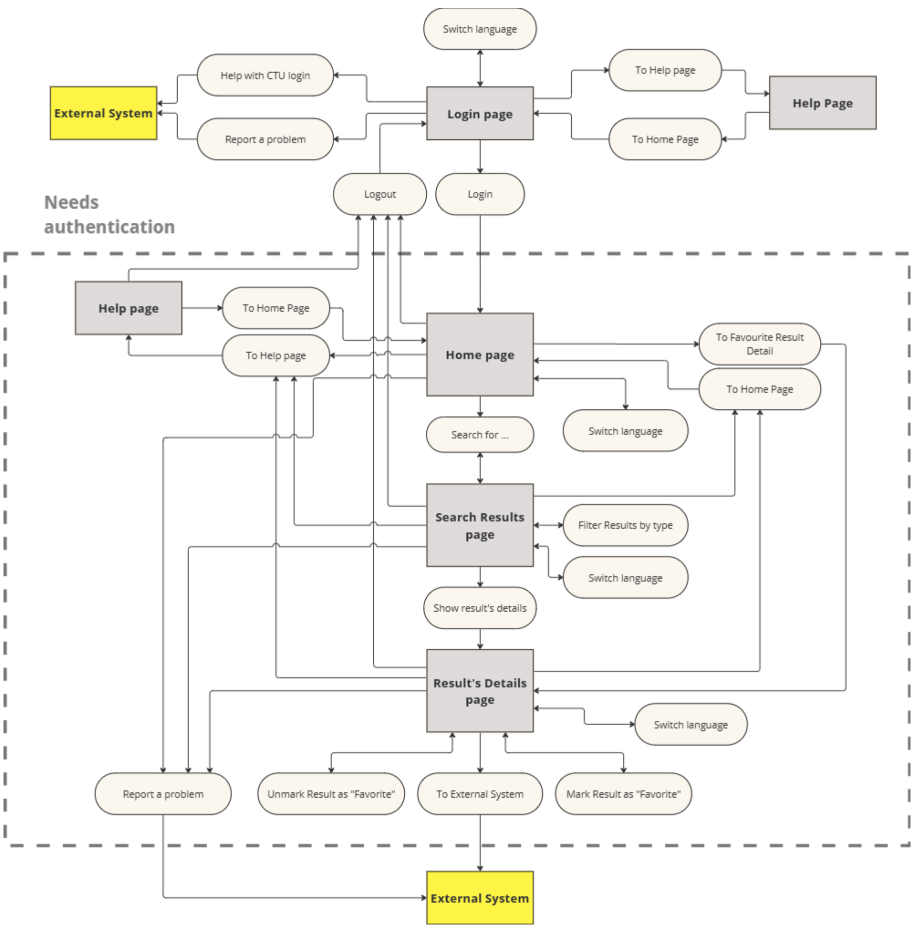


Figure 3.1: Task Graph of WhereIS user interface

3.2 Lo-Fi Prototype

We made a simple Lo-Fi prototype using wireframes to quickly show how the app's interface might look. We focused on four main features: Employees (teachers), Courses, Rooms, and Canteens, to speed up the process and save time. The prototype doesn't include some features like checking weekly canteen menus or searching for available rooms within specific time slots. This was done to test the prototype quickly with a small group of users and get their feedback on the design. Both desktop and mobile versions were created since these are the most common layouts used by users.

3.2.1 Wireframes Desription

The user begins on a Login page featuring a prominent button to access the application. The language selection option is available right away in the application header, along with a link to the login Help page for users who may need assistance with their credentials. You can see the layout is visible in Figure 3.2.

Upon logging in, the user is directed to the Home page. The layout of the Home page draws inspiration from the Courses system home page, as analysed in Section 2.3.1, as depicted in Figure 3.3.

The Home page features a search query text input field with item filters below it. Each filter indicates its state, whether it's toggled on or off. Below the filters, users can view a scrollable timetable overview, displaying several events from their personal timetable.

The bottom section of the homepage is dedicated to favourite items, with the most prominent ones being canteens and their daily menus. These are presented in a scrollable box, similar to the events. If the user doesn't have any upcoming events or favourite items, this is clearly communicated, as shown in Figure 3.4.

The layout of event cards for the timetable combines design present in both the Courses and Timetable systems, as analysed in Section 2.3.5. Several design iterations were tested, as shown in Figure 3.5.

When the user begins typing in the query text input field, the Home page automatically transitions to the Search Results page. The filters and input field remain unchanged, but search results appear dynamically as the user modifies the query (search-as-you-type functionality). This behavior is inspired by the Courses system. The layout of the Search Results page is depicted in Figure 3.6. If the query returns no results, a message is displayed to the user, as shown in Figure 3.7. If the user clears the text input field, they are automatically returned to the Home page.

To avoid overwhelming the user with information, as it was in Bc. T. Heger's prototype, discussed in Section 2.4, each search result within the Search Result page is presented as a surrogate with a title and item type specified as a tag. This design choice draws from the use of tags in Courses and Projects, as described in Section 2.3.6.

When the user clicks on a search result title, it functions as a dropdown. The dropdown content itself represents the Result's Detail page. If the user clicks on the search title again, the Result's Detail page is hidden. The Result's Detail pages for Course, Room, Canteen, and Employee(Teacher) are depicted in Figures 3.8, 3.9, 3.10, and 3.11 respectively.

Each Result's Detail pages and event cards in the timetables integrate links to other item's Detail pages, facilitating navigation through interconnected item types within the application. For instance, a user can click on an employee's office Room to automatically trigger a search for this room's Detail page.

The Canteen menu items were inspired by existing apps mentioned in Section 2.3.4.

Another interesting feature is the synthetic "List" item, which is used to group the search results. This is employed, for example, as a search result for currently free rooms. Two lists are presented, as a room can be either a computer or seminar room. The list layout is depicted in Figure 3.12.

All pages, except for the Login page, feature a header with a home button and a log out button. The home button redirects the user to the Home page and disables the search function.

The WhereIS wireframe prototype innovates by behaving more like a Single Page Application (SPA), eliminating the need to redirect user to other pages while preserving the user's action history. This means that users can easily navigate back to previous search results or opened Result Detail pages without any redirection.

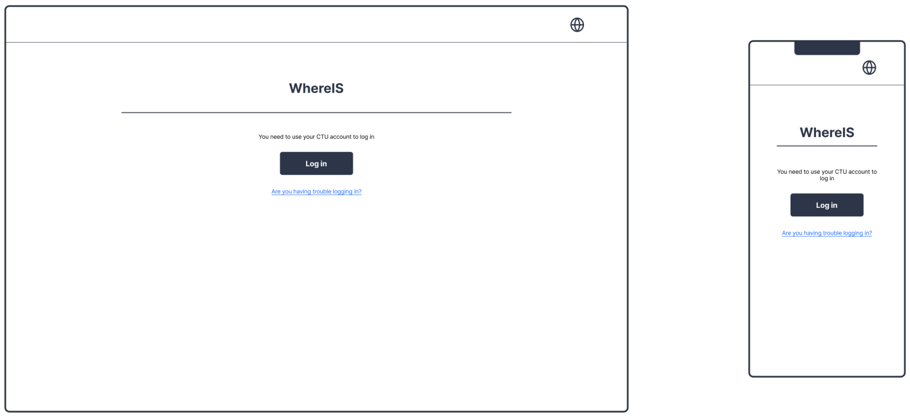

WhereIS
You need to use your CTU account to log in
Log in
Are you having trouble logging in?
WhereIS
You need to use your CTU account to log in
Log in
Are you having trouble logging in?

Figure 3.2: Wireframe – Login page (L)

3.2.2 Design Evaluation

To initiate the first changes in the proposed WhereIS user interface design, the wireframes were presented to a small sample of potential users. Overall, the layout received positive feedback from the testers, who appreciated its simplicity and the use of a single search query text input field for different item types. Features like searching for free rooms and marking favourite canteens were particularly well-received.

However, one significant flaw was identified in the design – the representation of the item filters. Placing them directly on the Home page and styling them like buttons led users to interpret them as shortcuts for quickly searching

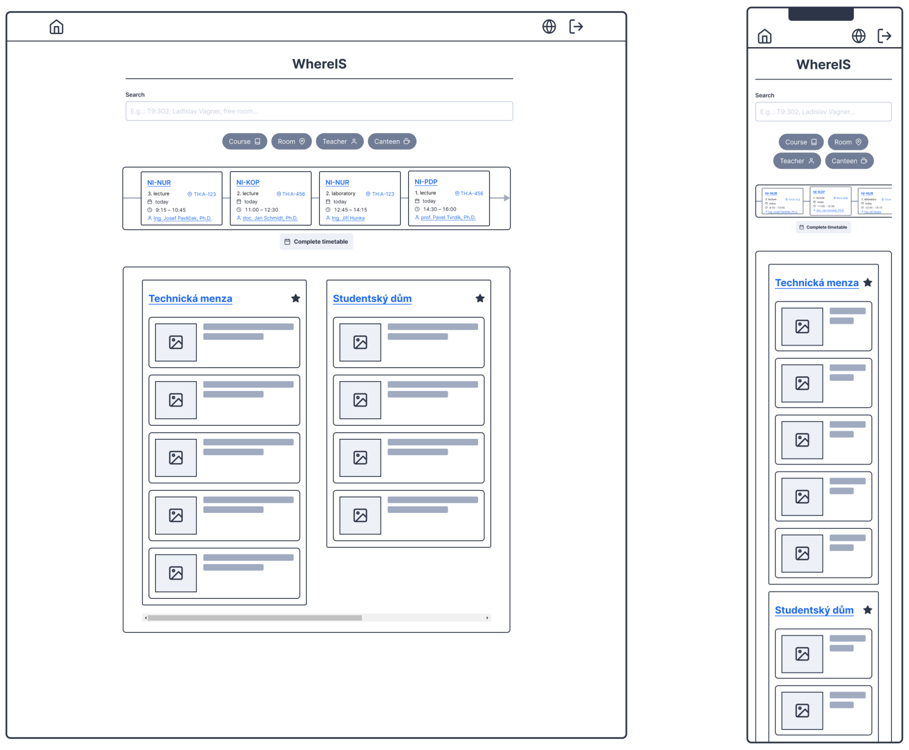


Figure 3.3: Wireframe – Home page (H)

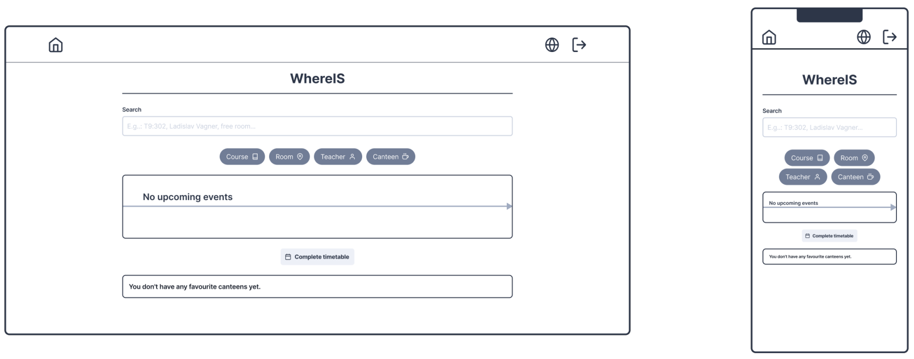

WhereIS
Search
E.g..: T9:302, Ladislav Vagner, free room...
Course
Room
Teacher
Canteen
No upcoming events
Complete timetable
You don't have any favourite canteens yet.

Figure 3.4: Wireframe – Home page (H) – Empty home page

for all items of a certain type, such as "Get All from...". Consequently, the filters were removed from the Home page, and their appearance was modified.

Now, filters are only visible on the Search Results page, concealed under the "All" filter. To visually distinguish them as filters, a checkbox was added next to each filter name. Clicking on the "All" filter reveals all available filters, which are initially deactivated (resulting in empty search results). Users can

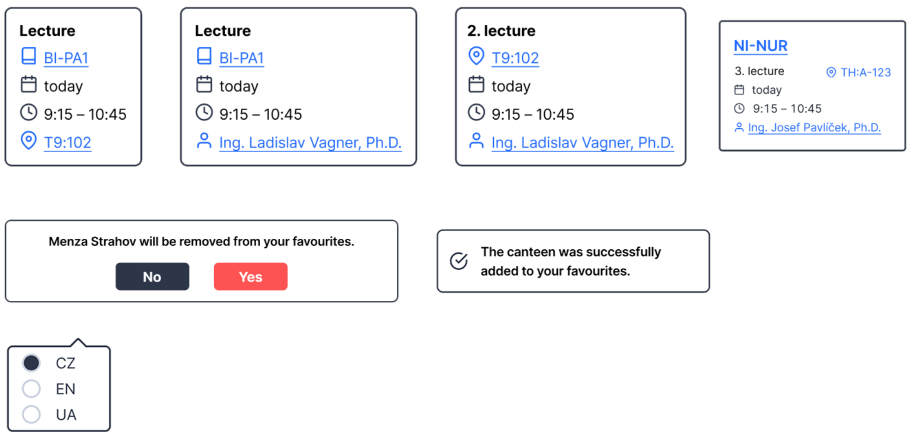

Lecture
BI-PA1
today
9:15 – 10:45
T9:102
Lecture
BI-PA1
today
9:15 – 10:45
Ing. Ladislav Vagner, Ph.D.
2. lecture
T9:102
today
9:15 – 10:45
Ing. Ladislav Vagner, Ph.D.
NI-NUR
3. lecture
TH:A-123
today
9:15 – 10:45
Ing. Josef Pavlíček, Ph.D.
Menza Strahov will be removed from your favourites.
No
Yes
The canteen was successfully added to your favourites.
CZ
EN
UA

Figure 3.5: Wireframe – other UI elements and their possible layouts

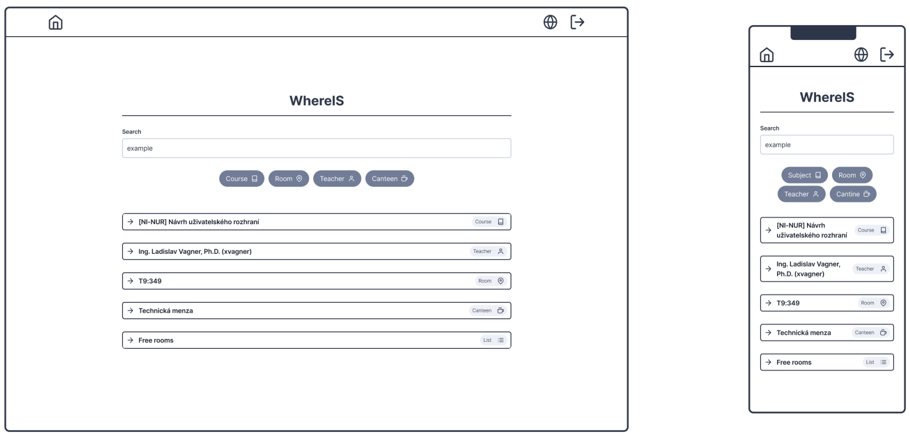


Figure 3.6: Wireframe – Search Results page (R)

then select the item filters they want to apply. To reset and view results for all item types again, users can click on “All” to hide the filters.

Given the positive reception of the overall layout, the remaining elements of the user interface, not depicted in the wireframes, will follow the same design principles. The final version of the user interface will be presented in the next Chapter 4.

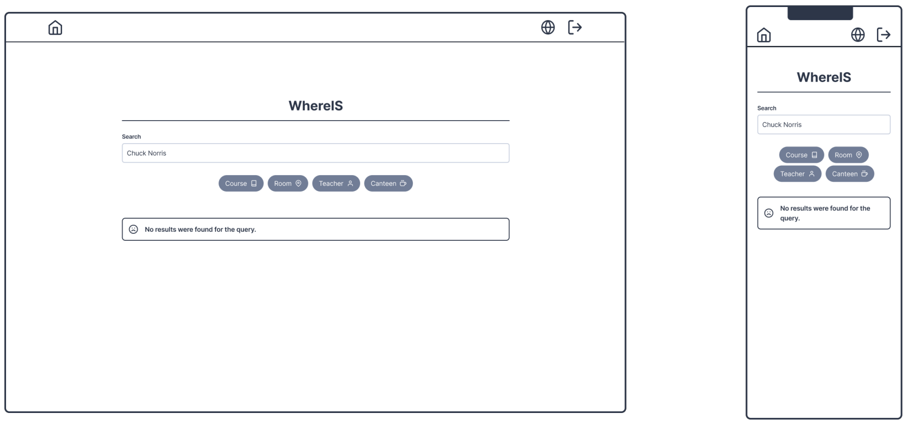


Figure 3.7: Wireframe - Search Results page (R) – No results

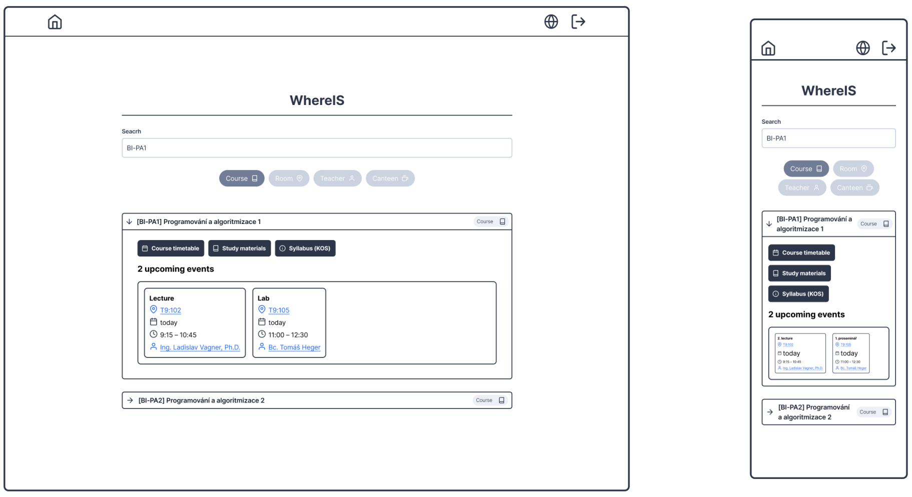


Figure 3.8: Wireframe – Result's Details page (D) – Course

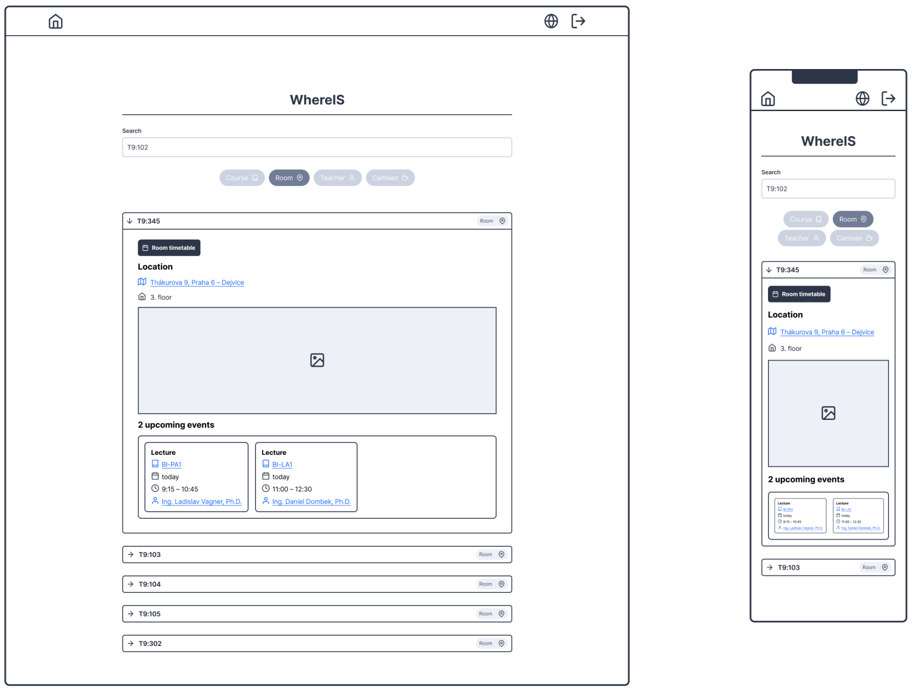


Figure 3.9: Wireframe – Result's Details page (D) – Room

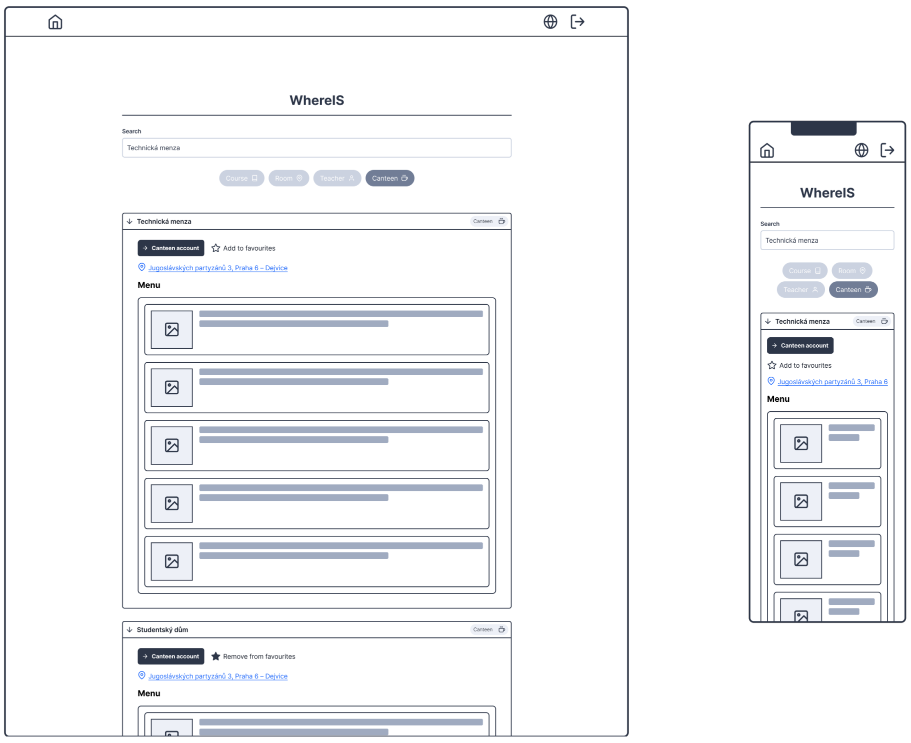


Figure 3.10: Wireframe – Result's Details page (D) – Canteen

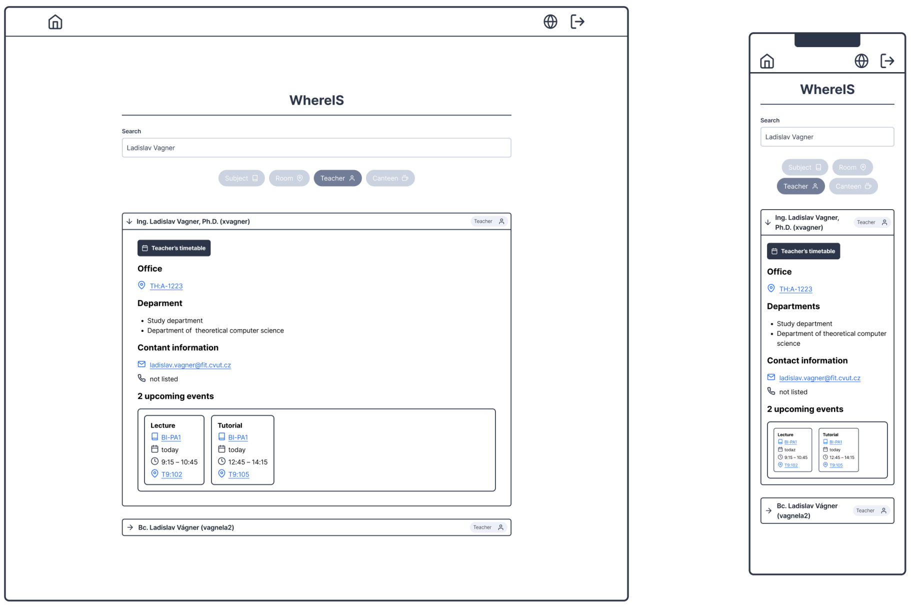


Figure 3.11: Wireframe – Result's Details page (D) – Teacher

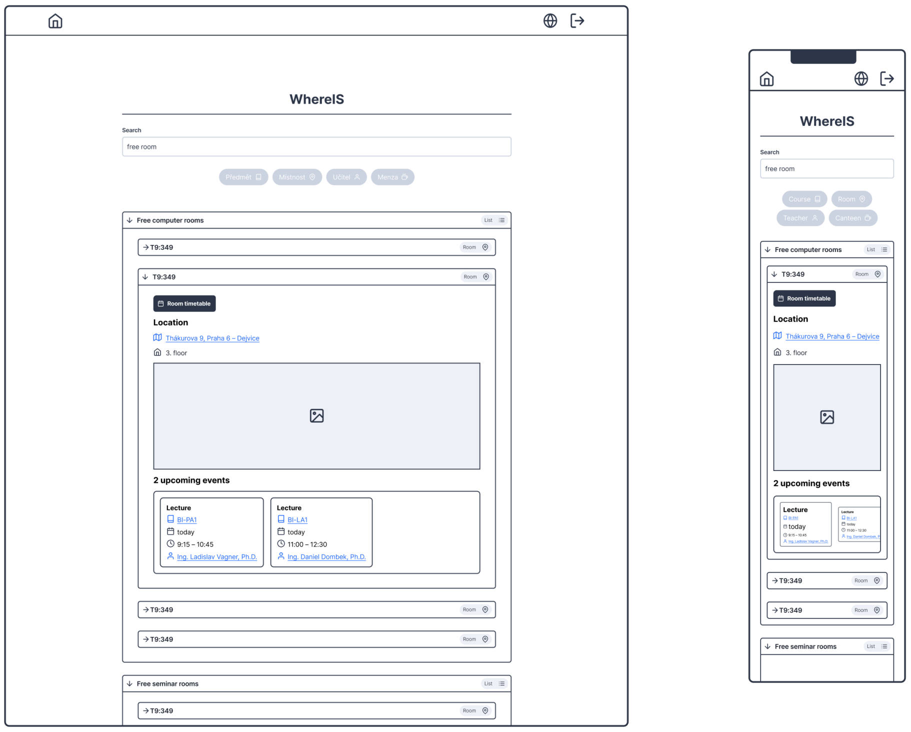


Figure 3.12: Wireframe – Result's Details page (D) – List

Chapter 4

Implementation

This chapter details the implementation of wireframe designs into the frontend client implementation, exploring the structure of application pages, management of user data, integration with the backend REST API, and features such as authentication and localization. It also discusses the React-based frontend architecture, covering routing, page management, and UI consistency.

4.1 Routing and Layout

The application uses client-side routing, where the client determines which page to serve based on the URL's pathname. The pathname is accessible through the built-in DOM `document.location.pathname` variable. Since the original React package doesn't support client-side routing out of the box and we're not using a more complex framework like Next.js [16], we've integrated the `react-router-dom` package [17], which is widely recognized as a solution to this problem. To implement routing, we utilize components such as `<BrowserRouter>`, `<Routes>`, and `<Route>`.

All routes are specified in the main React component, `<App>`, which is defined in the file `App.js`. The structure of the main component is illustrated in Listing 4.1. The root route `"/"` dictates that all nested child routes will utilize the `<Layout>` component, seen in Listing 4.2, as the page *layout* by setting the `element` prop value. This layout includes a `<Header>`, `<Outlet>` (or Content), and `<Footer>` components, ensuring consistency across pages while accommodating unique content. Additionally, this component determines whether the "Scroll to top" arrow in the bottom right corner of each page should appear.

The application offers five distinct routes:

1. `"/"`: serves as the index page or the Home page of the application.
2. `"/search"`: displays search results to users (Search Results page). Users can also use `?q=` parameter to specify the search query.
3. `"/login"`: directs users to the Login page, where the authentication process happens.

[16]Available from: `https://nextjs.org/`
[17]Available from: `https://github.com/remix-run/react-router`

```
<UserProvider>
  <NotificationProvider>
    <div className="App">
      <BrowserRouter>
        <Routes>
          <Route path="/" element={<Layout />}>
            <Route index element={
                <PrivateRoute element={<Container/>}/>
            }/>
            <Route path="search" element={
                <PrivateRoute element={<Container/>}/>
            }/>
            <Route path="login" element={<LoginPage />} />
            <Route path="help" element={<HelpPage />} />
            <Route path="*" element={<EmptyPage />} />
          </Route>
        </Routes>
      </BrowserRouter>
    </div>
  </NotificationProvider>
</UserProvider>
```

Listing 4.1: `<App>` component structure

```
function Layout() {

  const [showScrollToTop, setShowScrollToTop] = useState(false);

  .
  .
  .

  return (
    <div>
      <Header />
      <Outlet />
      <Footer />
      {showScrollToTop && <ScrollToTopArrow />}
    </div>
  )
}
```

Listing 4.2: `<Layout>` component structure

4. `"/help"`: guides the user on the Help page
5. `"/*"`: serves as the fallback or “404‘’ page

To restrict access to certain parts of the application, such as the Home page and Search Results page, we utilize the `<PrivateRoute>` component. Acting as middleware, this component checks whether the session storage contains the valid `userInfo` variable, which indicates that the user has successfully logged in to the application. If the user is not authenticated, the `<PrivateRoute>` component automatically redirects them to the Login page located at `"/login"`.

4.2 Pages

The implementation adheres to the page structure outlined in the application wireframes, provided in Section 3.2. While the Home page, Search Results page, and Result Detail page are interconnected, allowing for seamless transitions between them, they are encapsulated within the `<Container>` component for code organization purposes. The remaining pages maintain their original separation.

- `<Container>`: implements the Home page, Search Results page, and Result Detail page on the `"/"` and `"/search"` routes,
- `<HelpPage>`: implements the Help page on the `"/help"` route,
- `<LoginPage>`: implements the Login page on the `"/login"` route,
- `<EmptyPage>`: implements the ”404” page on the `"/*"` route.

The `<Container>` component facilitates the seamless transition between the Home page and the Search Results page. When a user begins typing in the search query field at the top of the webpage, the page dynamically transforms into the Search Results page. Conversely, when the user clears the search query input field, the page automatically reverts to the Home page.

4.3 Context

In React, we use *Context* to share important data among components. Context allows us to pass data through the component tree without manually passing props at every level. This means we can easily share values like preferences, theme settings, or user information across all components without having to pass them down manually. It simplifies data management and communication between components that are not directly connected in the hierarchy.

The implementation has 2 contexts: `UserContext` and `NotificationContext`.

`UserContext`. The `UserContext` handles the storage and sharing of user-related data, including the username, preferred language, and favourite items. These details, such as the preferred language and user’s favourite items, are retrieved from the application’s REST API, which will be elaborated upon later in Section 4.5.

```
const NotificationContext = React.createContext();

const NotificationProvider = ({ children }) => {

  const [notification, setNotification] = useState(null);

  const triggerNotification = (message, type) => {
    setNotification({message: message, type: type});
    setTimeout(() => {
      setNotification(null);
    }, 5000); // Dismiss after 5 seconds
  };

  return (
    <NotificationContext.Provider value={triggerNotification}>
      {children}
        {notification &&
          <Notification message={notification.message}
                        type={notification.type}
          />
        }
    </NotificationContext.Provider>
  );
};

export {NotificationProvider, NotificationContext};
```

Listing 4.3: `<Layout>` component structure

`NotificationContext`. The `NotificationContext` is responsible for generating pop-up text notifications across different sections of the application. Whenever a component requires notifying the user about an action or event, this context comes into play. For example, it informs users about successful or failed operations, like adding or removing items from their favourites. This functionality is accessed through the `triggerNotification` function, which takes two parameters: the notification text and its state. Currently, there are two notification states available: *success* and *failure*. These states are accessible via the config's `notifications` object, which stores the notification icon and CSS class name.

Each context is encapsulated within its own file: `UserContext.js` and `NotificationContext.js`. Within each module, the Context object and its respective `Provider` component are exported. These Provider components, integrated at the highest level of the `<App>` component, allow other components to subscribe to context changes. This organizational pattern is illustrated in Listing 4.1, where the `<UserProvider>` and `<NotificationProvider>` components are utilized. The implementation of one of the contexts can be seen in Listing 4.3.

```
<DetailBase>
  <DetailHeader/>
  <ItemDetail>
    <DetailLinks>
    . . . links to external systems
    </DetailLinks>
    <DetailInfo>
    . . . item detail entries
    </DetailInfo>
  </ItemDetail>
</DetailBase>
```

Listing 4.4: `<DetailBase>` structure

4.4 Search Result Details

Each search result on a Search Results page is encapsulated within a `<DetailBase>` component, which comprises two parts: the `<DetailHeader>` and the `<ItemDetail>` components and can be seen in Listing 4.4. The `<DetailHeader>` acts as a surrogate for a search result, displaying the title and item type. Following a click on the surrogate, as outlined in the wireframes, a dropdown revealing the `<ItemDetail>` component appears. This component is specific to each item type, such as Course, Room, or Canteen, with each possessing its unique *detail* component. Since the data for the Result Detail page (i.e., the `<ItemDetail>` component) is not initially present, it is fetched from the REST API, which will be described in the subsequent Section 4.5.

4.4.1 Favourite Item logic

To enhance the favourite items functionality in the `<ItemDetail>` componets, the `useFavouriteStatus` hook was created. This hook encapsulates the management of the item's favourite state on the detail page. It provides access to the `favouriteStatus` variable, which indicates whether the user has added the item to their favourites. Additionally, it offers a `handleFavourite` function that should be invoked upon clicking "add/remove from favourites" along with a `renderConfirmationModal` function. The latter returns the `<ConfirmationModal>` component, which appears when a user attempts to remove an item from their favourites.

To implement this favourite functionality, REST API needs to support it for the specified item type. If so, it's enough to simply integrate the code snippet found in Listing 4.5 into the `<ItemDetail>` component. The parameters for `useFavouriteStatus` include the item payload (`data` prop) and its label, as it should appear in the confirmation modal.

4.5 REST API Integration

The application utilizes an REST API provided by its backend server, running on the same domain as the client and prefixed with `/api/v1/`. This REST

```
const { favourite,
        handleFavourite,
        renderConfirmationModal } =
            useFavouriteStatus( props.data, props.data.code );

<IconTextButton
    onClick={handleFavourite}
    text={...}
    icon={favourite ?  "favourite_icon" : "classic_icon" }
/>

{renderConfirmationModal()}
```

Listing 4.5: `useFavouriteStatus` hook usage in Course item

API, version 1, was collaboratively designed with Bc. T. Heger to meet the Requirements and User Goals outlined in the Chapter 3.

The REST API supports various functionalities, including:

- retrieving search results based on a query,
- fetching item details,
- accessing user events, favourite items, and preferred language settings,
- getting lists of available rooms based on specific search criteria,
- retrieving the current and weekly menu for the canteen,
- setting user favourite items and preferred language.

4.5.1 Search

The core functionality of the app, the search feature, is supported by the `/api/v1/search?query={prompt}` endpoint, where the `prompt` is a text from a search query text input field.

Whenever the user modifies the search query input field, a request is sent to this endpoint. To prevent unnecessary load on the server, a debounce method is applied. This means that the request is triggered only after a certain delay following the last change to the query value. In this case, the debounce time is set to 400 ms, as specified in the constant defined in `config` file. Consequently, the request is made at intervals of at least 400 ms.

The request response comprises a list of item "views" objects. Each object contains properties such as the item type, id, title, and optionally, a URL attribute. This URL directs the frontend to the REST API endpoint where the detail item properties can be retrieved. The information from "views" is used to create a search result surrogates. The structure of search results list is seen in Listing 4.6.

This URL is utilized by all `<DetailBase>` components, except for one: the *List* item. Unlike other items, the List view does not possess detailed

```
[
  {
    "rank": 0,
    "type": "string",
    "id": 0,
    "title": "string",
    "?url": "string"
  },
  .
  .
  .
]
```

Listing 4.6: List object with search results

information within the response. Instead, its contents are immediately provided within the view object itself. Essentially, the List view serves as a collection of other view objects, presenting a grouped list of search results. For instance, it might display all available free seminar or computer Rooms.

4.5.2 Item Details

Item details are obtained from the endpoint `/api/v1/{item_type}/{id}`, which is accessible through the URL provided in the item "view" object. All item detail objects share a similar structure, with key attributes including:

- `id`: unique identifier of the item,
- `type`: type of the item,
- `urls`: a list of URL objects leading to external pages, which are then typically represented as black buttons at the top of the item detail,
- `events`: list of timetable `Event` objects associated with the item, such as room events or course events.

Each `<ItemDetail>` component receives the item detail object through the `data` prop. The component then accesses this prop to utilize the provided data for rendering purposes.

4.5.3 User Preferences

Information regarding user preferences, such as preferred language, favourite items, and user events, is accessible through various endpoints:

- `/api/v1/user/language`: provides data about the user's preferred language,
- `/api/v1/user/favourites`: returns item view objects representing the user's favourite items. These items are then displayed in the user dashboard,
- `/api/v1/user/events`: retrieves data regarding the user's events.

```
{% extends 'base.html.twig' %}
{% block body %}
  {% if app.user != null %}
    <script type="text/javascript">
      window.userInfo = {
        'username': '{{ app.user.userIdentifier|escape("js") }}'
      }
    </script>
  {% endif %}
  <div id="root"></div>
{% endblock %}
```

Listing 4.7: Sending user credentials in Twig template `index.html.twig`

4.6 User Authentication

To access data from the endpoints, users must be authenticated. Without authentication, successful responses from the application REST API are not possible.

The user authentication process occurs on the Login page when the user clicks the Login button. This process and its security is handled by the application backend. Upon clicking, the user is redirected to `https://auth.fit.cvut.cz/login.html` where CTU credentials are required. Upon successful authentication through this third-party system, the user is redirected back to the application, and the Home page is displayed. During this redirection, the username is stored in the application backend using Symfony UserProvider. The `userInfo` variable is then set within the HTML Twig boilerplate [18], seen in the Listing 4.7, on the backend server. Through this variable the frontend knows that the user successfully went through the authentication process.

4.7 Item Configuration

Each item possesses a shareable set of properties used in the search process logic or for visual representation. These properties are encapsulated within the `ItemConfiguration` class to facilitate scalability, particularly in the ease of addition of new items.

The properties include:

- `type`: item type.
- `component`: `<ItemDetail>` component corresponding to the item, displayed in the search result dropdown.
- `color`: CSS class name defining the item's color, utilized in various UI elements.
- `icon`: CSS icon class name, identical to the color property.
- `useFilter`: indicates whether the item should be included in search filters.

[18]Twig is a template engine for the PHP programming language.

```
class ItemConfigurationBuilder {
  constructor(type) {
    this.itemConfiguration = new ItemConfiguration()
    ...
  }

  setComponent(component){
    ...
  }

  .
  .
  .

  setEndpointPattern(endpointPattern) {
    ...
  }

  build() {
    ...
    return this.itemConfiguration
  }
}
```

Listing 4.8: Item configuration builder

- `endpointPattern`: URL pattern specifying where to fetch detailed information about the item.
- `fetch`: implementation of the fetch function, retrieving the item detail data and returning the prepared `<ItemDetail>` component with the response data passed as the `data` prop. In the event of an error during the fetch, the `<ErrorMessage>` component is returned.

To ease the creation of new configurations, the `ItemConfigurationBuilder` is introduced. This builder encapsulates the repetitive logic of creating the `fetch` function for each item. Most properties retain default values, with developers using the `set` method to modify specific properties. Upon completion of the configuration, the `build` method is called, returning the prepared `ItemConfiguration` object. These objects are then stored in the `config` file within the global `ItemConfigurations` object.

4.8 CSS

The application's responsiveness across various platforms and screen sizes is achieved through the utilization of CSS flexbox and media queries. Media queries are employed to assess factors such as screen size, orientation, or device type. Depending on these factors, CSS styles are dynamically adjusted to ensure optimal presentation and usability.

```
export const ItemConfigurations = {
  "course":  new ItemConfigurationBuilder("course")
            .setIcon("fa-book")
            .setComponent(CourseDetail)
            .build(),
  .
  .
  .
}
```

Listing 4.9: Item configuration example

For instance, distinct CSS rules may be applied for large landscape-oriented screens compared to small smartphone screens. These differences could involve adjustments to content spacing or image sizes to enhance the user experience and accommodate varying display environments.

Finding the right balance for gap values, especially on narrow smartphone screens with nested blocks, posed a significant challenge. This was particularly evident in scenarios where multiple dropdowns were nested within each other, such as displaying the day menu within the week menu for the canteen. In such cases, the available space for the innermost elements, like the menu itself, was limited.

The challenge was to reduce the gap sufficiently to ensure an optimal user experience without compromising usability or visual clarity. Balancing the spacing to accommodate nested elements on smaller screens while maintaining readability and usability remains an ongoing consideration for enhancing the application's design and responsiveness.

4.9 Localization

To enable multi-language support within the application, the `i18n` [19] library is employed. Currently, the application supports localization in two languages: Czech and English. The locale files contain translations for various string literals used throughout the application. Additionally, the i18n library facilitates the handling of pluralization, allowing for different translation variants based on the context.

4.10 Build and Deployment

Build. The React application is built on the backend server using Webpack Encore. Unlike traditional setups where a separate server hosts the React build, here the main application server serves the compiled React client. The React app is integrated within the HTML Twig template file, specifically `index.html.twig`. Within this file, the `<div id="root"></div>` element is defined, which acts as the entry point for the React application, as illustrated in Listing 4.10 in the `index.js` file.

[19]Available from: `https://www.i18next.com/`

```
const root = ReactDOM.createRoot(document.getElementById('root'));
root.render(
  <React.StrictMode>
    <App />
  </React.StrictMode>
);
```

Listing 4.10: React application entry point

```
<!DOCTYPE html>
<html>
<head>
  <meta charset="UTF-8">
  <title>{% block title %}WhereIS{% endblock %}</title>
  <meta name="viewport"
        content="width=device-width,initial-scale=1"
  >
  {% block stylesheets %}
    ...
    {{ encore_entry_link_tags('index') }}
  {% endblock %}
</head>
<body>
{% block body %}{% endblock %}
{% block javascripts %}
  {{ encore_entry_script_tags('index') }}
{% endblock %}
</body>
</html>
```

Listing 4.11: React app inclusion to Twig in `base.html.twig`

Once the build process is complete and the bundle is prepared, the React JavaScript code generated by Encore is included in the `base.html.twig` template. This inclusion is showcased in Listing 4.11.

Deployment. The entire application is deployed using Docker containers, with container orchestration managed by `docker compose`. a dedicated container is allocated for Webpack Encore, responsible for building the production version of the React JavaScript code. This container operates once, executes the build process, and then shuts down.

For development purposes, the `encore dev --watch` command can be utilized to monitor changes made to the code, instantly reflecting them in the deployed application. In this scenario, the container remains active, continuously monitoring for alterations in the React application code files.

Presently, the frontend implementation of the application does not incorporate any Continuous Integration (CI) practices for running tests or performing static code analysis.

4.11 Implemented User Interface

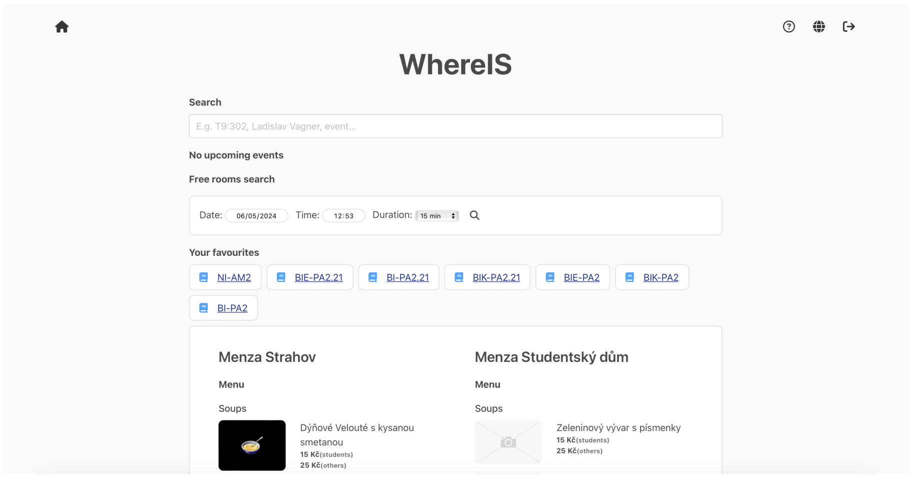
The final visual design sticks closely to the layouts provided in the wireframes in Section 3.2. All Detail pages also maintain the structure as proposed in the wireframes. Images of these pages can be found in the Appendix D.

The major difference in the final design compared to the initial proposals is the appearance of the filters. Issues with the original filter design were highlighted in Section 3.2 during the analysis of feedback from user testing of the prototype wireframes. The proposed modifications based on this feedback were subsequently implemented in the final design.

Filters were removed from the Home page, as can be seen in Figure 4.1 which presents the Home page design. During the search process, as depicted in Figure 4.2, the "All" filter is displayed just below the search query text input field. a small but significant addition to the search panel interface is an indication of how many search results were found, along with the number of results hidden due to the use of filters.

After clicking on the "All" filter, the filters are applied, as shown in Figure 4.4. This results in all search results being hidden from the user, and the counter for hidden results is displayed. Subsequently, when the user selects "Dormitory" and "Canteen" as item types, these categories become visible in the search results, reflecting the updated filter settings. This visibility change is depicted in Figure 4.3.

After another click on the "All" filter, the filter panel is hidden again, and all item types are allowed to be present in the search results.



WhereIS

Search

E.g. T9:302, Ladislav Vagner, event...

No upcoming events

Free rooms search

Date: 06/05/2024 Time: 12:53 Duration: 15 min

Your favourites

NI-AM2 BIE-PA2.21 BI-PA2.21 BIK-PA2.21 BIE-PA2 BIK-PA2 BI-PA2

Menza Strahov

Menu

Soups

Dýňové Velouté s kysanou smetanou
15 Kč(students)
25 Kč(others)

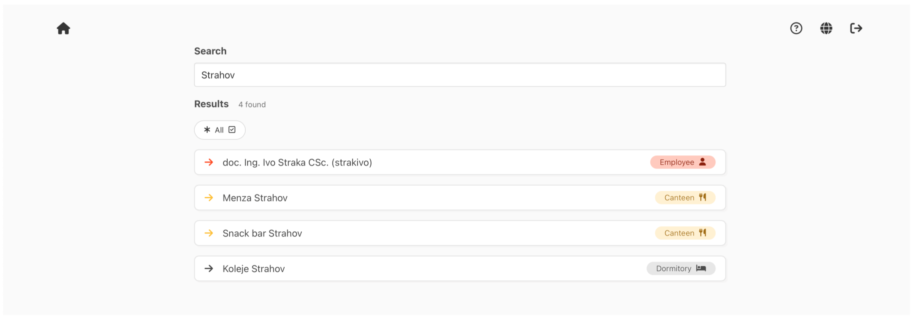
Menza Studentský dům

Menu

Soups

Zeleninový vývar s písmenky
15 Kč(students)
25 Kč(others)

Figure 4.1: Implemented UI – Home page



Search

Strahov

Results 4 found

All

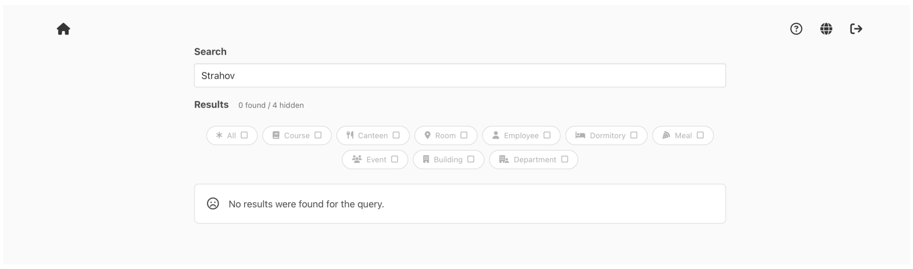
doc. Ing. Ivo Straka CSc. (strakivo) Employee

Menza Strahov Canteen

Snack bar Strahov Canteen

Koleje Strahov Dormitory

Figure 4.2: Implemented UI – Filters not applied



Search

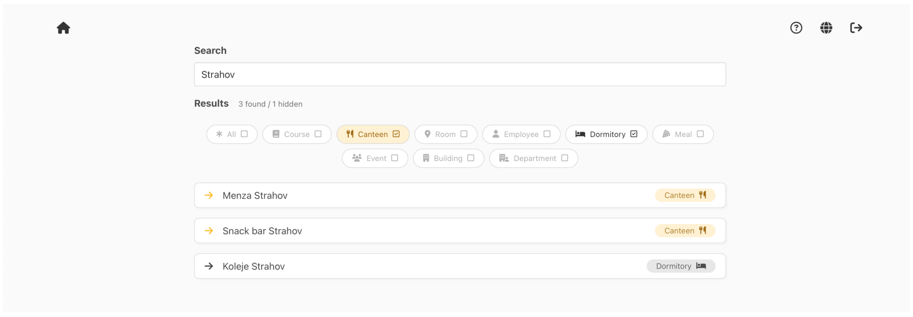
Strahov

Results 0 found / 4 hidden

All Course Canteen Room Employee Dormitory Meal Event Building Department

No results were found for the query.

Figure 4.3: Implemented UI – All filters disabled



Search

Strahov

Results 3 found / 1 hidden

All Course Canteen Room Employee Dormitory Meal Event Building Department

Menza Strahov Canteen

Snack bar Strahov Canteen

Koleje Strahov Dormitory

Figure 4.4: Implemented UI – Some filters applied

Chapter 5

Evaluation

This chapter focuses on evaluating the implementation. The evaluation process follows the methods outlined in Chapter 1 of this thesis. It begins with a Heuristic Analysis conducted by the thesis author, followed by Usability Testing. The findings and relevance of both testing approaches are then assessed in the conclusion section of this chapter.

5.1 Heuristic Analysis

This section presents the Heuristic Analysis conducted to evaluate the implementation of user interface design of the application. The evaluation process involved assessing the interface against established usability principles outlined by the Nielsen Norman Group. Each criterion was examined to identify any usability issues or violations. Subsequently, recommended solutions were proposed to address the identified issues. The findings from the heuristic analysis are outlined in the subsequent subsections.

5.1.1 Process Description

The evaluation session utilized 15 scenarios from the usability testing phase. Each scenario walkthrough was assessed using criteria outlined in the Nielsen Norman Group Heuristic Evaluation Workbook [46].

These criteria include:

H1 Visibility of System Status

- Does the design clearly communicate its state?
- Is feedback presented quickly after user actions?

H2 Match Between System and the Real World

- Will user be familiar with the terminology used in the design?
- Do the design's controls follow real-world conventions?

H3 User Control and Freedom

- Does the design allow users to go back a step in the process?

- Are exit links easily discoverable?
- Can users easily cancel an action?
- Is Undo and Redo supported?

H4 Consistency and Standards

- Does the design follow industry conventions?
- Are visual treatments used consistently throughout the design?

H5 Error Prevention

- Does the design prevent slips by using helpful constraints?
- Does the design warn users before they perform risky actions?

H6 Recognition Rather Than Recall

- Does the design keep important information visible, so that users do not have to memorize it?
- Does the design offer help in-context?

H7 Flexibility and Efficiency of Use

- Does the design provide accelerators like keyboard shortcuts and touch gestures?
- Is content and functionality personalized or customized for individual users?

H8 Aesthetic and Minimalist Design

- Is the visual design and content focused on the essentials?
- Have all distracting, unnecessary elements been removed?

H9 Help Users Recognize, Diagnose, and Recover from Errors

- Does the design use traditional error message visuals, like bold, red text?
- Does the design offer a solution that solves the error immediately?

H10 Help and Documentation

- Is help documentation easy to search?
- Is help provided in context right at the moment when the user requires it?

The Section 5.1.2 follows the proposed structure: each finding – an issue – includes a recommendation for how it could be solved.

Sce.	**Heu.**	**Issue**	**Recommended Solution**
All	H9	Error messages indicate that something went wrong but lack a solution.	Enhance error messages with more details depending on the error type. Provide a solution or at least a ”Repeat request“ option.
2	H4	The link to Google Maps doesn't match the style of other external links, potentially causing inconsistency.	Though intentional, consider aligning the Google Maps link with other external links for better consistency. This link is placed near contact information, which may also be a link (phone or email) with the ability to open mail or phone applications.
5	H9	When an employee doesn't teach any courses in the current semester, it's not explicitly stated.	Add an explicit text message indicating that the employee doesn't teach any courses in the current semester when the dropdown list doesn't appear.
5	H9	Similarly, when a course has no teachers in the current semester, it's not explicitly stated.	Add an explicit text message stating that the course is not taught in the current semester when the dropdown list doesn't appear.
6, 7, 14	H5	Removing the prompt in the search query field resets and hides applied filters, potentially frustrating users.	Keep filters visible at all times and avoid resetting them unless intentionally toggled by the user.
6	H7	Searching for all events doesn't always return relevant results first.	Improve backend ranking to prioritize relevant search results.
6, 7, 14	H10	The Help page lacks guidance on searching for all items of a given type using keywords.	Add information to the Help page and provide quick search buttons on the homepage to clarify how to retrieve all results of a given item.
9, 11	H4	Clicking on the canteen title from the Home page doesn't follow standard in-app link conventions, potentially confusing users.	Consider making the canteen title blue colored and underlined to match typical link styles. (Note: The current design choice aimed to keep the interface simple, since bold title links could be visually overwhelming.)
10	H10	Users may be unsure why certain courses appear in results when searching in the syllabus.	Add an explicit indicator to result surrogates to clarify that they're present in search results due to a match in the course syllabus.
12	H10	When searching for free rooms, users aren't informed about the duration and start time of room availability.	Add information to the free rooms list indicating that the rooms are available from the current time for 45 minutes.

Table 5.1: Violated Nielsen Heuristics and recommended solutions

5.1.2 Findings

In this section, for each test scenario, we will mention only the heuristics which were violated by the user interface during the scenarios walkthrough. For each violation the recommended solution is provided. These can be seen in Table 5.1.

5.2 Usability Testing

This section details the Usability Testing conducted to evaluate the implementation of the application's user interface design with potential users. The findings from both the surveys and the application testing sessions are then presented, highlighting key observations and recommendations for improvement.

5.2.1 Process Description

The initial stage of the testing included a Pre-test Survey that collected demographic information from participants. The questions from this survey are detailed in the Appendix C.

Once the Pre-test Survey was completed, the application testing part began. We introduced the user to the application testing process, including the rules and principles. a total of 15 scenarios were prepared, with most scenarios testing several user goals at once. Each scenario was conceptualized as a real-life case that could occur during application usage. The description of the scenarios and their expected fulfillment can be found in the Appendix C. During the fulfillment of the scenarios, notes were taken about the users' behaviors and any interesting remarks about how the testers were using the app. These notes will be part of the overall evaluation of the usability testing in the Section 5.2.2.

The application was deployed on a remote server and was accessible via `https://whereis.sophgus.eu` in order to make the application accessible on all devices. Each application testing session was conducted in person, utilizing either our laptop, or the participant's smartphone, PC or tablet. The testing process was recorded via video, with two types of recordings utilized: desktop screen capture paired with webcam footage for tests conducted on our laptop, and mobile, PC and tablet screen capture alongside webcam footage from our laptop for user devices tests [20].

After the application testing, each tester filled out the Post-test Survey, which was about their overall experience with the application during testing, as well as any possible ideas and critiques. The contents of the Post-test Survey can also be found in the Appendix C of this thesis.

5.2.2 Findings

5.2.2.1 Pre-test Survey

Twelve testers participated in the usability testing. Key demographic data from the Pre-test Survey revealed: ages ranged from 19 to 27 years, with half being 21 or younger; all testers were from the Czech Republic or Slovakia; 10 were students, 1 serving as teacher, and another 1 fulfilling dual roles as both

[20]All recordings are obtainable by contacting the thesis author on `bryloill@fit.cvut.cz`

student and teacher; over half (58.3 %) were in their first year of bachelor studies, while the rest had been at the faculty for four years or more.

5.2.2.2 Application Testing

During application testing some interesting observations were noted. After grouping them together we've extracted these:

- one of the issues users encountered was the unexpected disappearance of filters when transitioning back to the Home page after erasing the search query. This resulted in the reset of selected filters, causing confusion among users. a potential solution to address this problem is to maintain the visibility of filters at all times, but only activate them explicitly when the user initiates filter usage by clicking on the "All" filter. Once activated, the filter context remains consistent and visible until the user clicks on the "All" filter again, signaling the end of the filtering process,
- during testing, it was encouraging to see that some users turned to the Help page to find solutions, demonstrating its usefulness. However, the Help page was incomplete at the time of testing. This highlights the importance of ensuring that the Help page is fully prepared before the initial public deployment,
- notable that the majority of testers didn't revert back to the Home page to initiate a new search process, opting instead to remain on the last search results page and simply modify the existing search query,
- certain testers encountered difficulty locating the Free room search panel on the Home page, despite its prominent visibility. Some attempted to locate this search panel within the Free rooms List item, aiming to search for free rooms within a specific time span. To address this issue, it may prove beneficial to incorporate an informational message or shortcut button inside the List,
- users anticipated the ability to remove an item from their favourites directly on the Home page,
- on certain mobile devices, the size of UI elements appeared smaller than intended,
- on larger screen sizes, the elements could be grouped more closely together.

5.2.2.3 Post-test Survey

Some users were surprised by the application's reliance on a single search bar, noting its functional potential. While the search interface was generally praised for its convenience and robustness, some users experienced issues with disappearing filters after full erase of search query.

Challenges during testing included adapting to the MacBook device, which was our testing laptop, unfamiliar search methods, and non-intuitive filtering processes. Despite these issues, user satisfaction was high, with 58.3 % rating the usability as highest (5 out of 5) and the remainder giving a 4.

Concerns were minor, ranging from the absence of category-based searches without text filters to desires for a dark mode and larger buttons. The color scheme, which is mostly white, was sometimes a point of critique.

Overall, 41.7 % of users gave the application the highest score, indicating that it outperforms current school systems. Another 16.7 % provided neutral feedback, while the rest rated the application positively.

Users suggested some improvements like adding a dark mode, enhancing font customization, and implementing quick-access toolbars for better navigation. They also noted the need for more comprehensive search capabilities, including cross-links and the ability to search within course scripts.

5.3 Conclusion

The Heuristic Analysis identified several areas for improvement and minor inconsistencies in the design. However, these issues were not severe, and the recommended fixes will be valuable for future WhereIS design iterations.

Analysis of the Usability Testing Pre-test Survey revealed that the majority of participants are our target users – newcomers to FIT CTU. Additionally, individuals with extensive experience in the faculty's student life and university systems also participated, including Person a and Person B.

The application testing phase provided valuable insights, with testing conducted on all types of devices – PCs, laptops, smartphones and tablets – allowing for comprehensive assessment of UI responsiveness. While usability testing highlighted areas for improvement in usability and ease of use, overall, the current state of the user interface exhibits production-ready qualities and is ready for publication.

The Post-test Survey showed overwhelmingly positive reactions to the resulting implementation, along with valuable recommendations and ideas for the next iteration of the WhereIS user interface.

The combined results of both testing approaches offer a comprehensive set of findings, which will inform future work on the application.

Conclusion

The thesis commenced with an exploration of the User-Centered Design methodology. This approach, as its name implies, delved into understanding the requirements of FIT members to offer solutions that assist them in attaining their User Goals in solving academy and non-academy agenda.

The analysis began by examining various university and faculty systems, evaluating their strengths, weaknesses, and overall functionality. The User Requirements Questionnaire aided in obtaining direct feedback and proposals from potential users. Through a combination of this analysis and gathered requirements, along with the opportunity to leverage an existing application prototype by Bc. T. Heger, the design was made to develop a production-ready application user interface, implemented as a web client using React. The design and development process followed an iterative approach to ensure that critical changes and additions were made at the appropriate stages.

The application user interface underwent testing using two methods: Heuristic Analysis and Usability Testing with FIT participants. This evaluation phase demonstrated that the application design process was headed in the right direction and consequently received overwhelmingly positive reactions. This indicates that such a solution is indeed a valuable addition to FIT's current systems infrastructure. The goals outlined in the thesis Introduction were indeed achieved.

Additionally, the testing provided critical feedback and ideas on how to further improve the application. This signifies that it's not the end but rather the beginning of the journey for this newly created member of FIT systems family. The system which is **created by students for students**.

Improvements and Future Work

While the dark mode was highly requested, it wasn't implemented in this phase since the thesis focuses on designing and describing core interface functionality. Since the dark mode doesn't directly affect the user interface structure, it will be designed in the next iterations as a great feature to enhance usability even more.

This aligns with a functionality that allows users to customize their Home page layout. Different users have varying preferences for what they consider important information. This user dashboard feature could consider not only the size and position of elements like favourite items and the user timetable but also the data displayed within them. It would be beneficial to provide users with the option to choose which parts of item detail pages to display on their Home page.

Shortcuts for searching specific item types can be also beneficial. For instance, users can click on a button to display all faculty social events, enabling them to search within this subset of items.

When searching for a Room, the system could integrate with the faculty **Navigate** system. This feature would enable users to find a path to a given Room from defined starting locations. Unfortunately, during the development of this thesis, the Navigate system did not offer an API or URL query parameters to prepare the search in advance. Once Navigate provides this capability, it could function as another external button on the Room detail page. Users would only need to input the starting position, and the system would automatically fill in the destination parameter.

Employee (teacher) events are not currently displayed on the employee detail page due to limitations in the application's scope rights to access this information through the API. But this can be changed immediately as rights will be provided.

The application should be published and made accessible to FIT members. This will enable us to gather more insights and critique, which will inform the next iteration of the application.

Bibliography

1. HEGER, Tomáš. *Application WhereIS*. Prague, 2022. Available also from: `https://dspace.cvut.cz/handle/10467/102082`.
2. NORMAN, Donald A. Design principles for human-computer interfaces. In: *Proceedings of the SIGCHI conference on Human Factors in Computing Systems - CHI '83*. New York, New York, USA: ACM Press, 1983, pp. 1–10. ISBN 0897911210. Available from DOI: `10.1145/800045.801571`.
3. NORMAN, Donald A. *The design of everyday things*. Revised and expanded edition. New York: Basic Books, 2013. ISBN 978-0-465-05065-9.
4. FRIESS, Erin. The Sword of Data: Does Human-Centered Design Fulfill Its Rhetorical Responsibility? *Design Issues*. 2010, vol. 26, no. No. 3 Summer 2010, pp. 40–50.
5. *Enterprise Design Thinking*. 2023. Available also from: `https://www.ibm.com/design/approach/design-thinking/`.
6. VREDENBURG, Karel. Industry briefs: IBM. *Interactions*. 2001, vol. 8, no. 2, pp. 41–44. ISSN 1072-5520. Available from DOI: `10.1145/361897.361921`.
7. LUCENA, Percival; BRAZ, Alan; CHICORIA, Adilson; TIZZEI, Leonardo. IBM Design Thinking Software Development Framework. In: *Agile Methods*. Cham: Springer International Publishing, 2017, pp. 98–109. ISBN 978-3-319-55906-3. Available from DOI: `10.1007/978-3-319-55907-0_9`.
8. MAO, Ji-Ye; VREDENBURG, Karel; SMITH, Paul W.; CAREY, Tom. The state of user-centered design practice. *Communications of the ACM*. 2005, vol. 48, no. 3, pp. 105–109. ISSN 0001-0782. Available from DOI: `10.1145/1047671.1047677`.
9. MAO, Ji-Ye; VREDENBURG, Karel; SMITH, Paul W.; CAREY, Tom. User-centered design methods in practice: a survey of the state of the art. *CASCON '01: Proceedings of the 2001 conference of the Centre for Advanced Studies on Collaborative research*. 2001, vol. 11, no. 11, pp. 12–.
10. HUSSAIN, Zahid; SLANY, Wolfgang; HOLZINGER, Andreas. Current State of Agile User-Centered Design: A Survey. In: *HCI and Usability for e-Inclusion*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2009, pp. 416–427. ISBN 978-3-642-10307-0. Available from DOI: `10.1007/978-3-642-10308-7_30`.

11. ALAO, Olujimi Daniel; PRISCILLA, Ezihe Amarachi; AMANZE, Ruth Chinkata; KUYORO, Shade Oluwakemi; ADEBAYO, Adewale Olanrewaju. User-Centered/User Experience Uc/Ux Design Thinking Approach for Designing a University Information Management System. *Ingénierie des systèmes d information.* 2022, vol. 27, no. 4, pp. 577–590. ISSN 16331311. Available from DOI: `10.18280/isi.270407`.

12. JOSHI, Ankur; KALE, Saket; CHANDEL, Satish; PAL, D. Likert Scale: Explored and Explained. *British Journal of Applied Science & Technology.* 2015, vol. 7, no. 4, pp. 396–403. ISSN 22310843. Available from DOI: `10.9734/BJAST/2015/14975`.

13. *Brainstorming.* 2024. Available also from: `https://dictionary.cambridge.org/dictionary/english/brainstorming`.

14. *Product Statement Template.* 2024. Available also from: `https://www.edrawmind.com/templates/product-statement-template.html`.

15. MOORE, Geoffrey A. *Crossing the chasm: marketing and selling disruptive products to mainstream customers.* Third edtion. New York: HarperBusiness, 2014. ISBN 978-0-06-229298-8.

16. GOTTESDIENER, Ellen. *Requirements by Collaboration: Workshops for Defining Needs.* 1st ed. Addison-Wesley Professional, 2002. ISBN 9780201786064.

17. GOLDSMITH, Robin F. *Discovering Real Business Requirements for Software Project Success.* 1st ed. Artech House, 2004. ISBN 1580537715.

18. ELLIS, Keith; BERRY, Daniel M. Quantifying the impact of requirements definition and management process maturity on project outcome in large business application development. *Requirements Engineering.* 2013, vol. 18, no. 3, pp. 223–249. ISSN 0947-3602. Available from DOI: `10.1007/s00766-012-0146-3`.

19. WONG, M.L.; KHONG, C.W.; THWAITES, H. Applied UX and UCD Design Process in Interface Design. *Procedia - Social and Behavioral Sciences.* 2012, vol. 51, pp. 703–708. ISSN 18770428. Available from DOI: `10.1016/j.sbspro.2012.08.228`.

20. EJIKE, Jude; AKCAY, Ediz Edip. Applying User-Centered Design Methods to Improve The Experience of the NHS APP. 2024. Available from DOI: `10.32388/BYAENM`.

21. PUTRA, Singgih Tanu; EDY, Edy. Analysis and Design Ecommerce with User Centered Design (UCD) Method at PT. Multi Prima Mandiri Sukses. *Tech-E.* 2020, vol. 4, no. 1, pp. 21–29. ISSN 2581-1916. Available from DOI: `10.31253/te.v4i1.405`.

22. ISACKER, Karel Van; SLEGERS, Karin; GEMOU, Maria; BEKIARIS, Evangelos. A UCD Approach towards the Design, Development and Assessment of Accessible Applications in a Large Scale European Integrated Project. In: *Universal Access in Human-Computer Interaction. Addressing Diversity.* Berlin, Heidelberg: Springer Berlin Heidelberg, 2009, pp. 184–192. ISBN 978-3-642-02706-2. Available from DOI: `10.1007/978-3-642-02707-9_20`.

23. MIASKIEWICZ, Tomasz; KOZAR, Kenneth A. Personas and user-centered design: How can personas benefit product design processes? *Design Studies.* 2011, vol. 32, no. 5, pp. 417–430. ISSN 0142694X. Available from DOI: `10.1016/j.destud.2011.03.003`.

24. MIASKIEWICZ, Tomasz; GRANT, Susan Jung; KOZAR, Kenneth A. A Preliminary Examination of Using Personas to Enhance User-Centered Design. *AMCIS 2009 Proceedings.* 2009, vol. 2009. Available also from: `http://aisel.aisnet.org/amcis2009/697`.

25. HJALMARSSON, Anders; GUSTAFSSON, Eva; CRONHOLM, Stefan. Exploring the Use of Personas in User-Centered Design of Web-based e-services. *IConference 2015 Proceedings.* [N.d.]. Available also from: `http://hdl.handle.net/2142/73633`.

26. LOSADA, Begoña. Flexible Requirement Development through User Objectives in an Agile-UCD Hybrid Approach. In: *Proceedings of the XIX International Conference on Human Computer Interaction.* New York, NY, USA: ACM, 2018, pp. 1–8. ISBN 9781450364911. Available from DOI: `10.1145/3233824.3233865`.

27. HEARST, Marti A. User Interfaces for Search. In: *Modern Information Retrieval: The Concepts and Technology Behind Search.* 2nd ed. Addison-Wesley Professional, 2011. ISBN 978-0321416919.

28. COCKBURN, Alistair. Structuring Use cases with goals. *Journal of Object-Oriented Programming.* 1997, no. Nov-Dec 1997.

29. COCKBURN, Alistair; FOWLER, Martin. Question time! about use cases. In: *Proceedings of the 13th ACM SIGPLAN conference on Object-oriented programming, systems, languages, and applications.* New York, NY, USA: ACM, 1998, pp. 226–229. ISBN 1581130058. Available from DOI: `10.1145/286936.286960`.

30. PAVLÍČEK, Josef. *NI-NUR: User Interface Design.* Prague, 2023. Available also from: `https://bk.fit.cvut.cz/en/predmety/00/00/00/00/00/00/06/11/37/p6113706.html`.

31. FOWLER, Martin. *UML Distilled: A Brief Guide to the Standard Object Modeling Language.* 3rd ed. Addison-Wesley Professional, 2004. ISBN 9780321193681.

32. JOKELA, Timo. Making user-centred design common sense. In: *Proceedings of the second Nordic conference on Human-computer interaction.* New York, NY, USA: ACM, 2002, pp. 19–26. ISBN 1581136161. Available from DOI: `10.1145/572020.572024`.

33. COWEN, Michael; LEMON, Alan; GILL-HESSELGRAVE, Deborah. User-Centered Design (UCD) Process Description. *TECHNICAL REPORT 2061.* 2014. Available also from: `https://apps.dtic.mil/sti/pdfs/ADA615926.pdf`.

34. EDFORS, Emmy; SVERRESON, Albin. *Utilizing user centered design to mitigate security threats.* Lund, Sweden, 2022. Available also from: `https://lup.lub.lu.se/luur/download?func=downloadFile%5C&recordOId=9101319%5C&fileOId=9101320`.

35. RICK, Jochen; FRANCOIS, Phyllis; FIELDS, Bob; FLECK, Rowanne; YUILL, Nicola; CARR, Amanda. Lo-fi prototyping to design interactive-tabletop applications for children. In: *Proceedings of the 9th International Conference on Interaction Design and Children*. New York, NY, USA: ACM, 2010, pp. 138–146. ISBN 9781605589510. Available from DOI: `10.1145/1810543.1810559`.

36. WIBAWANI, Sri; DAMALIANA, Aviolla Terza; SETIAWAN, Ariyono; KUSUMA, Irma Dwi; DIYASA, I Gede Susrama Mas. Wireframe Creation on SIOBEL Application User Interface Design using User Centered Design. *Information Technology International Journal*. 2023, vol. 1, no. 2, pp. 56–65. ISSN Online-ISSN 3025-3125. Available also from: `https://itijournal.org/index.php/ITIJ/article/view/12/7`.

37. CHEN, Jieshan; CHEN, Chunyang; XING, Zhenchang; XIA, Xin; ZHU, Liming; GRUNDY, John; WANG, Jinshui. Wireframe-based UI Design Search through Image Autoencoder. *ACM Transactions on Software Engineering and Methodology*. 2020, vol. 29, no. 3, pp. 1–31. ISSN 1049-331X. Available from DOI: `10.1145/3391613`.

38. GUNNULFSEN, Michael K. Gunnulfsen. *Scalable and Efficient Web Application Architectures: Thin-clients and SQL vs. Thick-clients and NoSQL Michael K.* Oslo, 2013.

39. *React*. 2024. Available also from: `https://opensource.fb.com/projects/react`.

40. DELICE, Elif Kılıç; GÜNGÖR, Zülal. The usability analysis with heuristic evaluation and analytic hierarchy process. *International Journal of Industrial Ergonomics*. 2009, vol. 39, no. 6, pp. 934–939. ISSN 01698141. Available from DOI: `10.1016/j.ergon.2009.08.005`.

41. NIELSEN, Jakob. Finding usability problems through heuristic evaluation. In: *Proceedings of the SIGCHI conference on Human factors in computing systems - CHI '92*. New York, New York, USA: ACM Press, 1992, pp. 373–380. ISBN 0897915135. Available from DOI: `10.1145/142750.142834`.

42. NIELSEN, Jakob; MOLICH, Rolf. Heuristic evaluation of user interfaces. In: *Proceedings of the SIGCHI conference on Human factors in computing systems Empowering people - CHI '90*. New York, New York, USA: ACM Press, 1990, pp. 249–256. ISBN 0201509326. Available from DOI: `10.1145/97243.97281`.

43. NIELSEN, Jakob. *10 Usability Heuristics for User Interface Design*. 1994. Available also from: `https://www.nngroup.com/articles/ten-usability-heuristics/`.

44. MULLER, Michael J.; MATHESON, Lisa; PAGE, Colleen; GALLUP, Robert. Methods & tools: participatory heuristic evaluation. *Interactions*. 1998, vol. 5, no. 5, pp. 13–18. ISSN 1072-5520. Available from DOI: `10.1145/285213.285219`.

45. TONDELLO, Gustavo F.; KAPPEN, Dennis L.; MEKLER, Elisa D.; GANABA, Marim; NACKE, Lennart E. Heuristic Evaluation for Gameful Design. In: *Proceedings of the 2016 Annual Symposium on Computer-Human Interaction in Play Companion Extended Abstracts*. New York, NY, USA: ACM, 2016, pp. 315–323. ISBN 9781450344586. Available from DOI: 10.1145/2968120.2987729.

46. *Heuristic Evaluation Workbook*. 2024. Available also from: https://media.nngroup.com/media/articles/attachments/Heuristic_Evaluation_Workbook_1_Fillable.pdf.

47. CHAUHAN, Sudhanshu; PANDA, Nutan Kumar. Foundation. In: *Hacking Web Intelligence*. Elsevier, 2015, pp. 1–13. ISBN 9780128018675. Available from DOI: 10.1016/B978-0-12-801867-5.00001-X.

48. OLIVEIRA, Bruno; LOPES, Carla Teixeira. The Evolution of Web Search User Interfaces - An Archaeological Analysis of Google Search Engine Result Pages. In: *Proceedings of the 2023 Conference on Human Information Interaction and Retrieval*. New York, NY, USA: ACM, 2023, pp. 55–68. ISBN 9798400700354. Available from DOI: 10.1145/3576840.3578320.

49. KOCHUMMEN, Preethy Ann. *What is Search UI? Best Practices and Examples*. 2023. Available also from: https://www.argoid.ai/blog/what-is-search-ui.

50. ORMAZABAL, Xabier. *7 examples of stellar search UIs*. 2024. Available also from: https://www.algolia.com/blog/ux/7-examples-of-great-site-search-ui/.

51. USHA, M. A Hybrid Page Ranking Algorithm for Organic Search Results. *International Journal for Research in Applied Science and Engineering Technology*. 2017, vol. V, no. VIII, pp. 2348–2359. ISSN 23219653. Available from DOI: 10.22214/ijraset.2017.8335.

52. SINGHAL, Amit. *Introducing the Knowledge Graph: things, not strings*. 2012. Available also from: https://blog.google/products/search/introducing-knowledge-graph-things-not/.

53. *FIT CTU COURSES*. 2019. Available also from: https://courses.fit.cvut.cz/.

54. *UserMap*. 2024. Available also from: https://usermap.cvut.cz/.

55. CHALUPSKÁ, Ilona. *CTU Brand Style Guide*. [N.d.]. Available also from: https://www.cvut.cz/en/ctu-logo.

56. *KOS*. 2018. Available also from: https://kos.cvut.cz/.

57. *CTU canteen menus*. 2023. Available also from: https://agata-new.suz.cvut.cz/.

58. *Fittable*. 2022. Available also from: https://timetable.fit.cvut.cz/new/.

59. *Projects FIT*. 2021. Available also from: https://projects.fit.cvut.cz/.

Appendix A

Glossary

API	Application Programming Interface
UCD	User-Centered Design
UI	User Interface
UCSD	User-Centered System Design
HCD	Human-Centered Design
IBM	International Business Machines Corporation
SUS	System Usability Scale
NHS	National Health Service
GUI	Graphical User Interface
HTML	HyperText Markup Language
DOM	Document Object Model
AJAX	Asynchronous Javascript And XML
SERP	Search Engine Results Page
ORCID	Open Researcher and Contributor ID
SFA	Service Facilities Administration
FAQ	Frequently Asked Questions
UG	User Goal
UC	Use Case
SPA	Single Page Application
JS	JavaScript
CSS	Cascading Style Sheets
CI	Continuous Integration

Appendix B

User Requirements Questionnaire Description

Demographic Questions

1. **Question:** What is your study program?
   - Bachelor's
   - Master's
   - Doctoral
   - Lifelong learning
   - I am not studying anymore
2. **Question:** Do you teach any subject at school?
   - Yes
   - No
3. **Question:** How long have you been working at the faculty?
   - Less than 1 year
   - 1 to 3 years
   - More than 3 years

Current Experiences and Ideas for Improvement

In this section, we aim to gather your experiences with the current information systems at our school and collect ideas for enhancements.

1. **Question:** How often do you use the existing information systems of the faculty/school?
   - KOS
   - Courses
   - Timetable

- Agata (Canteen system)
- UserMap

Frequency Options:

- Never
- Occasionally
- Sometimes
- Often
- Always

2. **Question:** On a scale from 1 to 5, how would you rate your overall satisfaction with each of these systems?

   - KOS
   - Courses
   - Timetable
   - Agata (Canteen system)
   - UserMap

   Satisfaction Scale:

   - 1 – Great dissatisfaction
   - 2 – Dissatisfaction
   - 3 –Neutrality
   - 4 – Satisfaction
   - 5 – Great satisfaction

3. **Question:** Do you use any other information system not listed above? If so, which one and for what purpose?
4. **Question:** What are the biggest challenges or frustrations you experience when searching for information in these systems?
5. **Question:** Which information do you find difficult to locate in the current systems?
6. **Question:** What features or functionalities would you like to have available in the current systems?

Device Usage

1. **Question:** On which device do you most frequently use the current information systems?

   - Mobile
   - Tablet
   - Desktop / Laptop

Proposed Functionalities

We have some ideas for functionalities and want to find out if you think they are useful.

1. **Question:** Would you appreciate a functionality that helps to find currently available computer and seminar rooms?

   Rating Scale:

   - 1 – Definitely not
   - 2
   - 3
   - 4
   - 5 – Definitely yes

2. **Question:** Would you value having quick access to your upcoming events in the application?

   Rating Scale:

   - 1 – Definitely not
   - 2
   - 3
   - 4
   - 5 – Definitely yes

3. **Question:** Would you like the ability to compare the menus of your favourite cafeterias side by side?

   Rating Scale:

   - 1 – Definitely not
   - 2
   - 3
   - 4
   - 5 – Definitely yes

4. **Question:** Would you find it useful to have more detailed information about room locations (building address, floor, floor plan, etc.)?

   Rating Scale:

   - 1 – Definitely not
   - 2
   - 3
   - 4
   - 5 – Definitely yes

5. **Question:** Would you consider the ability to save various entities to favourites for quick access useful?

   Entities:

   - Cafeterias
   - Courses
   - Teachers
   - Available Rooms

   Rating Scale:

   - 1 – Definitely not
   - 2
   - 3
   - 4
   - 5 – Definitely yes

6. **Question:** To which other entities (information) would you like to have quick access?

7. **Question:** Would you be interested in features for improved accessibility mentioned below?

   Features:

   - Dark mode
   - Adjustable text size
   - Language selection
   - Mode for color blindness (food photos)

   Interest Level:

   - 1 – Certainly not
   - 2
   - 3
   - 4
   - 5 – Certainly yes

8. **Question:** What else, not previously mentioned, would you like to see in the application?

Appendix C

Usability Test Description

Pre-test Survey

1. **Question:** Your name and surname
2. **Question:** Your username (CTU)
3. **Question:** Your age
4. **Question:** Are you a student from a country other than the Czech Republic and Slovakia?
    - Yes
    - No
5. **Question:** At faculty, are you:
    - Student
    - Teacher
    - Student and teacher
    - Non-academic employee
6. **Question:** How many years have you been at FIT?

Application Testing

1. **Scenario:** You are studying the subject BI-PA2.21 and need to find out where Mr. Ing. Ladislav Vagner, Ph.D. (xvagner) could be located.

   Expected scenario fulfillment:

    - *Scenario No. 1 (User searches for Mr. Vagner):*
        a) The user navigates to the application's Home page in a browser.
        b) They input a partial query, including "Ing. Ladislav Vagner, Ph.D. (xvagner)" into the text search field.
        c) After entering part of the query, the user pauses.
        d) The system processes the search and presents them to the user.

    e) The user selects Mr. Vagner's employee Detail page from the search results.
    f) Within the employee Detail page, the user finds the location of Mr. Vagner's office.

- *Scenario No. 2 (User starts by searching for BI-PA2.21):*
    a) The user opens the application's Home page in a browser.
    b) They input "BI-PA2.21" into the text search field.
    c) Upon entering the query, the user pauses.
    d) The system processes the search and presents them to the user.
    e) The user accesses the Detail page for the "BI-PA2.21" course.
        i. If the user finds the "BI-PA2.21" event with Mr. Vagner as a participant, they note the event's room location. Alternatively,
        ii. The user identifies Mr. Vagner in the list of teachers for the "BI-PA2.21" course and proceeds as in Scenario No. 1.

2. **Scenario:** You are in the first semester of bachelor's studies at FIT and have lecture in room TK:BS. You don't know where this room is located. Find out the address of this room.

   Expected scenario fulfillment:

   a) The user begins to fill in the text search field with "TK:BS" on the Home page or Search Results page.
   b) After entering the query, the user pauses.
   c) The system processes the search and presents them to the user.
   d) User navigates to the Detail page for the "TK:BS" room.
   e) Within the Detail page, the user locates and notes the address.

3. **Scenario:** Find out the number of rooms in the building where room TK:BS is located.

   Expected scenario fulfillment:

   a) The user remains on the Detail page for the "TK:BS" room.
   b) They identify the link to the "TK" building within the room Detail page.
   c) Clicking on the link, the user opens the Detail page for the "TK" building.
   d) Within the "TK" building Detail page, the user locates the list of rooms situated in this building and proceeds to count them.

4. **Scenario:** Find out the closest event in room T9:345.

   Expected scenario fulfillment:

   a) The user begins to fill in the text search field with "T9:345" on the Home page or Search Results page.
   b) After entering the query, the user pauses.

c) The system processes the search and presents them to the user.
d) User navigates to the Detail page for the "T9:345" room.
e) Within the Detail page, the user locates the first event.

5. **Scenario:** You liked the teaching style of the teachers of the course NI-DDW, so you would like to find out what other courses they teach (finding information about one of the teachers is enough).

   Expected scenario fulfillment:

   a) The user begins filling the text search field with "NI-DDW" on either the Home page or the Search Results page.
   b) After entering the query, the user pauses.
   c) The system processes the search and presents the results to the user.
   d) The user navigates to the Detail page for the "NI-DDW" course.
   e) Within the Detail page, the user finds the list of teachers.
   f) The user clicks on the first teacher in the list.
   g) By clicking on the link, the user opens the Detail page for the employee.
   h) Within the employee's Detail page, the user locates the list of courses taught by that employee.

6. **Scenario:** You have heard that various events are often organized at FIT, and you want to find out about all of them. Filter out any irrelevant results from the search. Find out the time and place of one selected event.

   Expected scenario fulfillment:

   a) The user begins entering "event" or "událost/akce" into the text search field on either the Home page or the Search Results page.
   b) After entering the query, the user pauses.
   c) The system processes the search and presents the results to the user.
   d) The user identifies the "All" filter button and clicks on it.
   e) In the filter list, the user selects only the "event" item type.
   f) The user is presented with filtered results.
   g) The user navigates to the Detail page for the selected event.
   h) Within the event Detail page, the user finds the location and time span of the event.

7. **Scenario:** You are hungry. Find out how many canteens you have to choose from for having a lunch.

   Expected scenario fulfillment:

   a) The user begins entering "canteen" or "menza" into the text search field on either the Home page or the Search Results page.
   b) After entering the query, the user pauses.

   c) User counts the search results with “Canteen” item label.

8. **Scenario:** You like the Technická menza and the Menza Studentský dům, and it’s often hard for you to decide where to go. Find out how to have a quick access to the menus of the above-mentioned canteens.

   Expected scenario fulfillment:

   a) The user remains on the Search Results page with canteens.
   b) User opens the Detail page for Technická menza and clicks on the“Add to favourites” button at the top of the page.
   c) User opens the Detail page for Menza Studentský dům and clicks on the “Add to favourites” button at the top of the page.
   d) User clicks on the “Home page” button in the webpage header to return to the Home page.
   e) User notices that the canteens are now listed in the Favourites section.

9. **Scenario:** At one of your favourite canteens, you want to find out in advance what food you can look forward to the next day.

   Expected scenario fulfillment:

   a) The user remains on the Home page.
      i. User selects one of the two favourite canteens: Technická menza or Menza Studentský dům and clicks on the canteen title. Alternatively,
      ii. User searches for one of the two canteens.
   b) User opens the canteen’s Detail page.
   c) User navigates to the bottom part of the Detail page under the current canteen menu and locates the week menu dropdown.
   d) User clicks on the week menu dropdown and opens the detail for tomorrow’s menu (or the next opening day after today).

10. **Scenario:** You want to find the course where you will learn about the Pumping Lemma. Find out the rest of this course syllabus.

    Expected scenario fulfillment:

    a) The user begins filling the text search field with “Pumping Lemma” on either the Home page or the Search Results page.
    b) After entering the query, the user pauses.
    c) The system processes the search and presents the results to the user.
    d) User navigates to the Detail page for one of the listed “AAG” courses.
    e) Within the top part of the Detail page, the user locates the “Syllabus” external button.

11. **Scenario:** You no longer like the Technická menza and want to remove it from your favourites.

    Expected scenario fulfillment:

a)
   i. User locates the Technická menza detail page using the Search Results page. Alternatively,
   ii. User locates the Technická menza in the Home page favourites section and clicks on the title.

b) User opens the canteen's Detail page.

c) User clicks on the "Remove from favourites" button at the top of the Detail page.

d) User confirms the removal in the confirmation modal.

12. **Scenario:** Tomorrow you have a 30-minute break after 08:30 AM between classes. You don't have your laptop with you, so you want to find a room with a computer to prepare your homework.

    Expected scenario fulfillment:

    a) User navigates to the Home page.
    b) User finds the "Free rooms" search panel.
    c) User sets the date for tomorrow and the time to 08:30 AM.
    d) User sets the duration to 30 minutes.
    e) User clicks on the magnifying button.
    f) User is presented with results: a dropdown list of seminar and computer rooms if the criteria were fullfilled.
    g) User opens the list of computer rooms if present.

13. **Scenario:** You are an international student at FIT, and you find it difficult to navigate in the Czech language. Change the language of the application to English.

    Expected scenario fulfillment:

    a) User finds the globe icon in the application header.
    b) User clicks on the globe icon.
    c) User selects the "en" option.

14. **Scenario:** You are not from Prague, so you are looking for options for accommodation in dormitories. Find the email of the dormitory you would write to.

    Expected scenario fulfillment:

    a) The user starts entering "dormitory", "kolej" or "ubytování" into the text search field on either the Home page or the Search Results page.
    b) After entering the query, the user pauses.
    c) The system processes the search and presents the results to the user.
    d) User clicks on one of the dormitory options to access its Detail page.
    e) Within the Detail page, the user locates the email contact information.

15. **Scenario:** You have a craving for guláš. Find out when and where you can satisfy your craving.

    Expected scenario fulfillment:

    a) The user begins entering "guláš" into the text search field on either the Home page or the Search Results page.
    b) After entering the query, the user pauses.
    c) The system processes the search and presents the results to the user.
    d) User clicks on one of the meals from the search results.
    e) User discovers the time and location for serving this meal on the meal Detail page.

Post-test Survey

1. **Question:** Your name and surname
2. **Question:** Your username (CTU)
3. **Question:** Your age
4. **Question:** Did anything surprise or catch you off guard while using the application during testing?
5. **Question:** What was difficult for you during testing?
6. **Question:** What specifically did you dislike about the application?
7. **Question:** On a scale from 1 to 5, rate whether using WhereIS is easier than searching in current school systems?

   Rating Scale:

   - 1 – Much more complicated
   - 2
   - 3
   - 4
   - 5 – Much more easier
8. **Question:** On a scale from 1 to 5, would you use the WhereIS application if it were available?

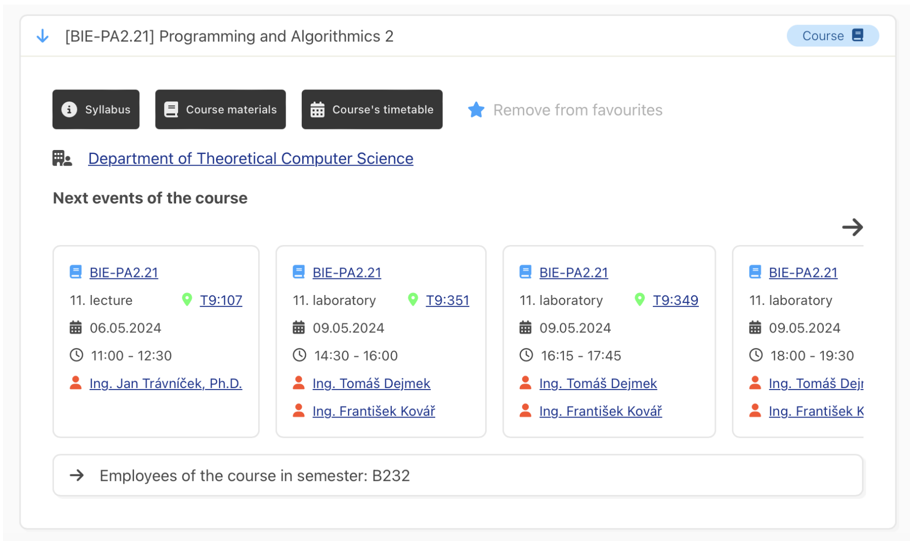
   Rating Scale:

   - 1 – Definitely not
   - 2
   - 3
   - 4
   - 5 – Definitely yes
9. **Question:** If yes, for what purpose?

10. **Question:** On a scale from 1 to 5, rate the usability of the application.
    Rating Scale:
    - 1 – Very poor
    - 2
    - 3
    - 4
    - 5 – Very good
11. **Question:** On a scale from 1 to 5, rate the clarity of the application.
    Rating Scale:
    - 1 – Very unclear
    - 2
    - 3
    - 4
    - 5 – Very clear
12. **Question:** What do you think is missing from the application?
13. **Question:** What would you do differently regarding the design?

Appendix D

Implemented User Interface



[BIE-PA2.21] Programming and Algorithmics 2

Course

Syllabus | Course materials | Course's timetable | Remove from favourites

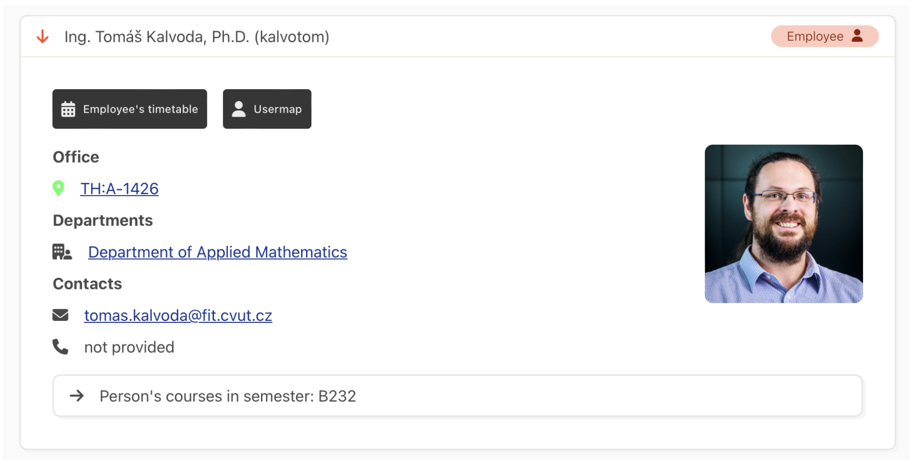
Department of Theoretical Computer Science

Next events of the course

BIE-PA2.21	BIE-PA2.21	BIE-PA2.21	BIE-PA2.21
11. lecture T9:107	11. laboratory T9:351	11. laboratory T9:349	11. laboratory
06.05.2024	09.05.2024	09.05.2024	09.05.2024
11:00 - 12:30	14:30 - 16:00	16:15 - 17:45	18:00 - 19:30
Ing. Jan Trávníček, Ph.D.	Ing. Tomáš Dejmek	Ing. Tomáš Dejmek	Ing. Tomáš Dej
	Ing. František Kovář	Ing. František Kovář	Ing. František K

Employees of the course in semester: B232

Figure D.1: Implemented UI – Course Detail page



Ing. Tomáš Kalvoda, Ph.D. (kalvotom)

Employee

Office

TH:A-1426

Departments

Department of Applied Mathematics

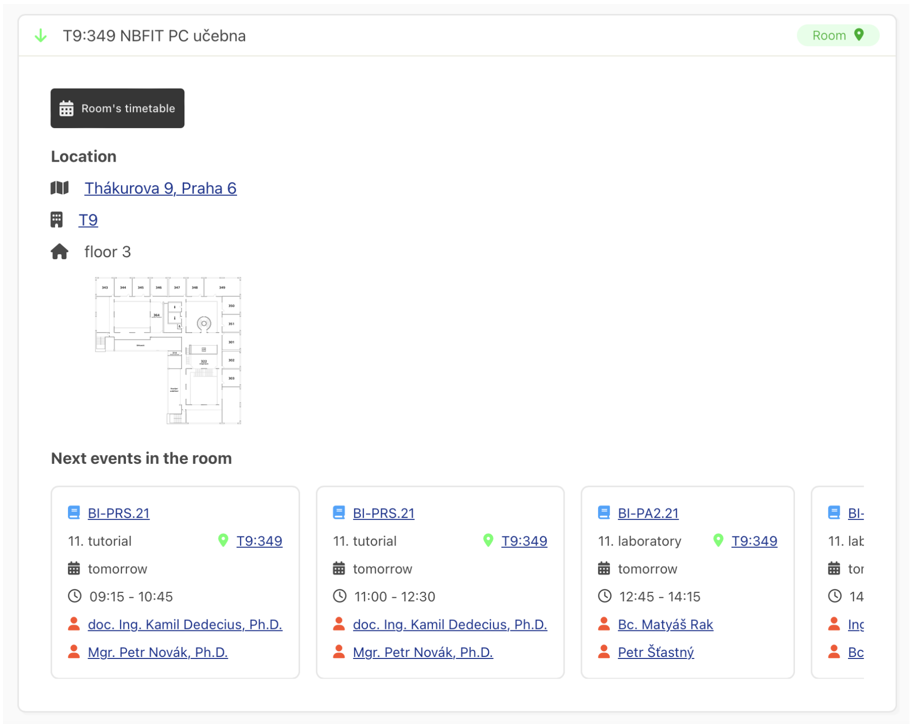
Contacts

tomas.kalvoda@fit.cvut.cz

not provided

Person's courses in semester: B232

Figure D.2: Implemented UI – Employee Detail page



T9:349 NBFIT PC učebna

Room

Location

Thákurova 9, Praha 6

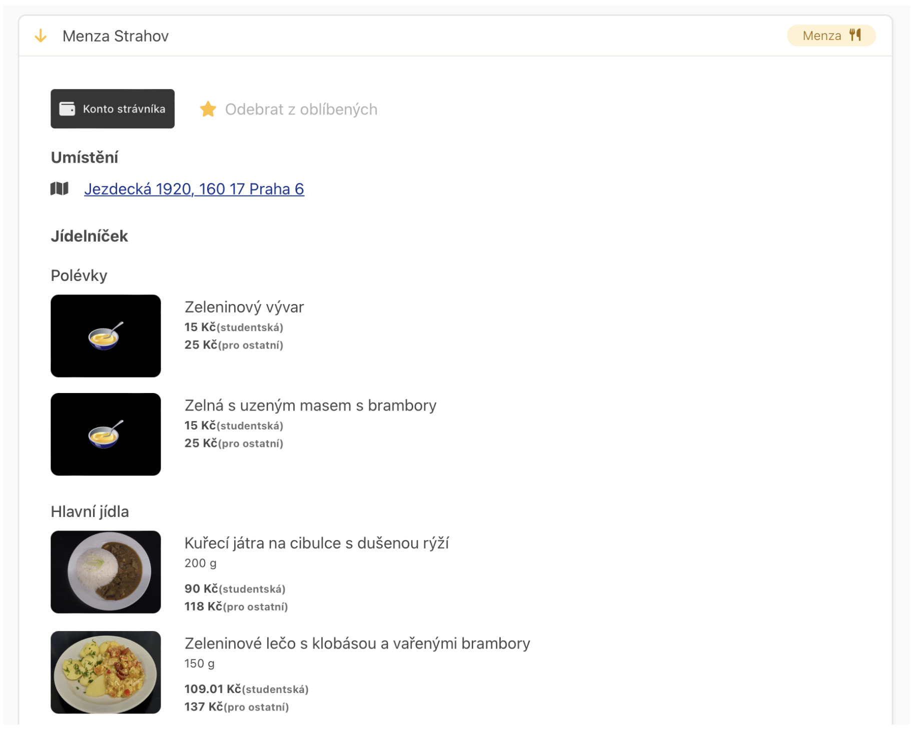
T9

floor 3

Next events in the room

BI-PRS.21	BI-PRS.21	BI-PA2.21	BI-
11. tutorial T9:349	11. tutorial T9:349	11. laboratory T9:349	11. lab
tomorrow	tomorrow	tomorrow	tor
09:15 - 10:45	11:00 - 12:30	12:45 - 14:15	14
doc. Ing. Kamil Dedecius, Ph.D.	doc. Ing. Kamil Dedecius, Ph.D.	Bc. Matyáš Rak	Inc
Mgr. Petr Novák, Ph.D.	Mgr. Petr Novák, Ph.D.	Petr Šťastný	Bc

Figure D.3: Implemented UI – Room Detail page



Menza Strahov

Menza

Konto strávníka

Odebrat z oblíbených

Umístění

Jezdecká 1920, 160 17 Praha 6

Jídelníček

Polévky

Zeleninový vývar
15 Kč(studentská)
25 Kč(pro ostatní)

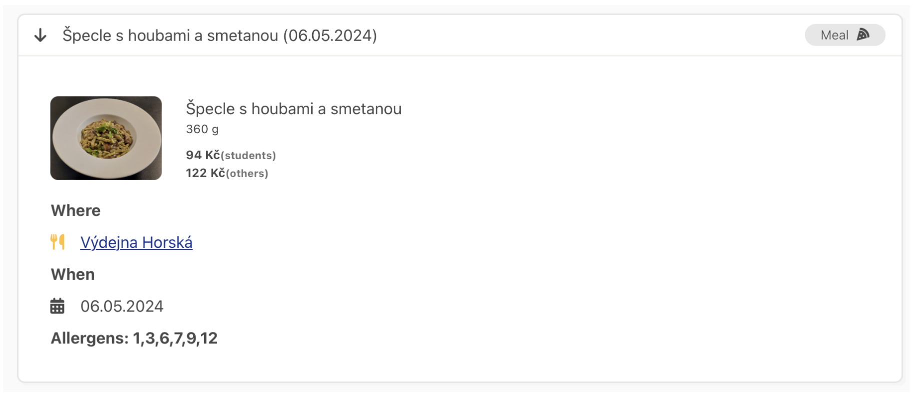
Zelná s uzeným masem s brambory
15 Kč(studentská)
25 Kč(pro ostatní)

Hlavní jídla

Kuřecí játra na cibulce s dušenou rýží
200 g
90 Kč(studentská)
118 Kč(pro ostatní)

Zeleninové lečo s klobásou a vařenými brambory
150 g
109.01 Kč(studentská)
137 Kč(pro ostatní)

Figure D.4: Implemented UI – Canteen Detail page



Špecle s houbami a smetanou (06.05.2024)

Meal

Špecle s houbami a smetanou
360 g
94 Kč(students)
122 Kč(others)

Where

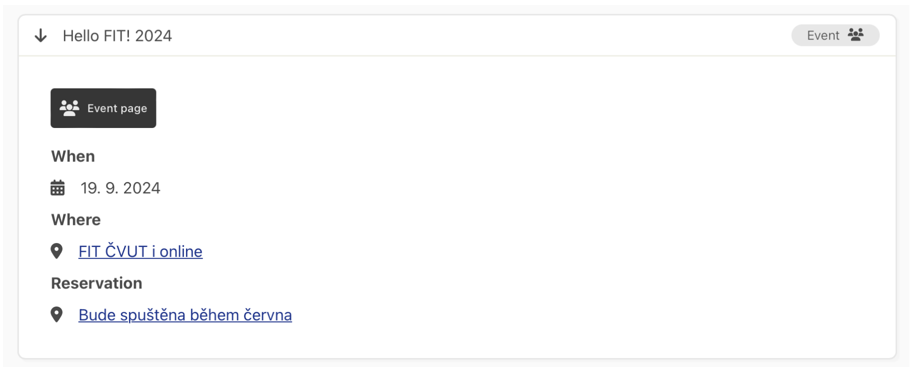
Výdejna Horská

When

06.05.2024

Allergens: 1,3,6,7,9,12

Figure D.5: Implemented UI – Meal Detail page


Hello FIT! 2024

Event

Event page

When

19. 9. 2024

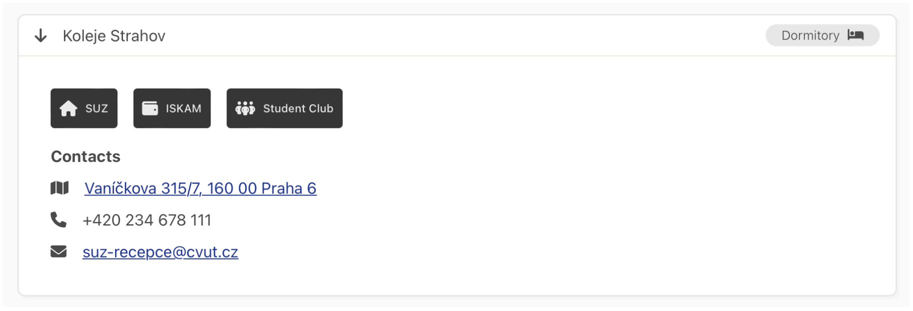
Where

FIT ČVUT i online

Reservation

Bude spuštěna během června

Figure D.6: Implemented UI – Even Detail page


Koleje Strahov

Dormitory

SUZ

ISKAM

Student Club

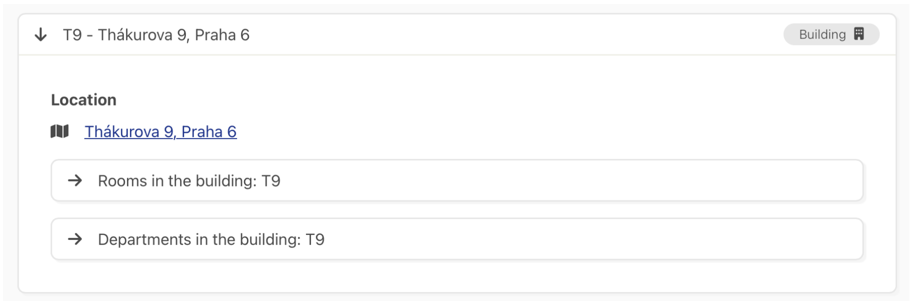
Contacts

Vaníčkova 315/7, 160 00 Praha 6

+420 234 678 111

suz-recepce@cvut.cz

Figure D.7: Implemented UI – Dormitory Detail page


T9 - Thákurova 9, Praha 6

Building

Location

Thákurova 9, Praha 6

Rooms in the building: T9

Departments in the building: T9

Figure D.8: Implemented UI – Building Detail page

Appendix **E**

Attached ZIP Structure

```
/
|-  readme.txt             - short archive description
|-  text
    |- thesis.pdf          - thesis text in PDF format
|-  aplikace-whereis/      - application source files
    |-  templates/         - Twig HTML files
        |- base.html.twig  - base Twig template with bundled React app
        |- index.html.twig - template to include app root and userInfo
    |-  front/             - frontend JavaScript files
        |- src/
            |- components/ - UI reusable components
            |- config/     - application configuration definitions
            |- contexts/   - context providers
            |- fetchers/   - fetch functions
            |- helpers/    - helper functions
            |- hooks/      - reusable hooks
            |- locales/    - translations
            |- pages/      - page component definitions
            |- styles/     - CSS files
            |- App.js      - application's main component definition
            |- i18n.js     - translations configuration
            |- index.js    - entrypoint of React application
    |- package.json        - frontend dependencies
    |- webpack.config.json - Webpack build configuration
    |- yarn.lock           - Yarn lock file with dependencies versions
    |- README.md           - Readme file with usage and deployment guide
```

Listing E.1: Attached ZIP structure - frontend related files