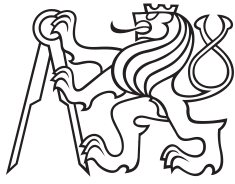


Bachelor Project



**Czech
Technical
University
in Prague**

F3

**Faculty of Electrical Engineering
Department of Computer Science**

Progressive web application for online psychotherapy

Selina Kadyrova

**Supervisor: Ing. Kyrlyo Bulat
May 2023**

I. Personal and study details

Student's name: **Kadyrova Selina**

Personal ID number: **499357**

Faculty / Institute: **Faculty of Electrical Engineering**

Department / Institute: **Department of Computer Science**

Study program: **Software Engineering and Technology**

II. Bachelor's thesis details

Bachelor's thesis title in English:

Progressive web application for online psychotherapy

Bachelor's thesis title in Czech:

Progresivní webová aplikace pro online psychoterapii

Guidelines:

Develop a progressive web application to help people with their search for mental health professionals.

Project goals:

- * Research and comparison of existing online psychotherapy platforms,
- * Design of the application architecture and its user interface,
- * Selection of technologies suitable for implementation,
- * Application implementation,
- * Design test scenarios,
- * Manual and automated testing.

The application will support the following functionality:

- * Clients can search and connect with psychotherapists online
- * Psychotherapists can register and showcase their profiles in the list of available professionals

Bibliography / sources:

1. Richards, Mark. Fundamentals of Software Architecture: An Engineering Approach. O'Reilly Media, 2020.
2. Progressive Web Apps (PWAs) [online] URL: https://developer.mozilla.org/en-US/docs/Web/Progressive_web_apps
3. Ater, Tal. Building Progressive Web Apps. O'Reilly Media, 2017
4. Blodyk, Gerardus. Progressive Web Apps Standard Requirements. 5STARCOOKS, 2022

Name and workplace of bachelor's thesis supervisor:

Ing. Kyrylo Bulat System Testing IntelIgent Lab FEE

Name and workplace of second bachelor's thesis supervisor or consultant:

Date of bachelor's thesis assignment: **28.02.2023** Deadline for bachelor thesis submission: **05.02.2024**

Assignment valid until: **16.02.2025**

Ing. Kyrylo Bulat
Supervisor's signature

Head of department's signature

prof. Mgr. Petr Páta, Ph.D.
Dean's signature

III. Assignment receipt

The student acknowledges that the bachelor's thesis is an individual work. The student must produce her thesis without the assistance of others, with the exception of provided consultations. Within the bachelor's thesis, the author must state the names of consultants and include a list of references.

Date of assignment receipt

Student's signature

Acknowledgements

I would like to express my deepest gratitude to my supervisor, Ing. Kyrylo Bulat, for his invaluable guidance, support, and expertise throughout the entire process of this bachelor thesis. His mentorship and insightful feedback have been instrumental in shaping the direction and quality of this work. I am also indebted to my parents for their unwavering love, encouragement, and belief in my abilities. Their constant support and understanding have been my pillar of strength throughout my academic journey. Furthermore, I extend my appreciation to all my teachers and professors who have imparted their knowledge and skills, cherishing my passion for learning. Their dedication and commitment to education have greatly influenced my growth as a student and researcher. Lastly, I would like to express my heartfelt thanks to all those who have contributed to this thesis, directly or indirectly. Your contributions, advice, and encouragement have played a huge role in the completion of this project. I am truly grateful for the support and encouragement I have received, and I am honored to have had the opportunity to undertake this bachelor's thesis.

Declaration

I hereby declare that this thesis represents my own work which has been done after registration for the Bachelor's degree at Czech Technical University, and has not been previously included in a thesis or dissertation submitted to this or any other institution for a degree, diploma or other qualifications.

Prague, May 23, 2023

Abstract

This bachelor thesis focuses on the design and implementation of a prototype for a Progressive Web Application platform for online psychotherapy services. The thesis encompasses an exploration of PWA technology, an analysis of existing solutions, the design of system architecture, the selection of appropriate technologies, the implementation process of the prototype, and considerations for future improvements.

Keywords: Progressive web application, online psychotherapy, mental health, web application, development, REST, Java, Spring Framework, JavaScript, React

Supervisor: Ing. Kyrlo Bulat

Abstrakt

Tato bakalářská práce se zaměřuje na návrh a implementaci prototypu platformy progresivní webové aplikace pro online psychoterapeutické služby. Práce zahrnuje průzkum technologie PWA, analýzu existujících řešení, návrh architektury systému, výběr vhodných technologií, proces implementace prototypu a úvahy o budoucích vylepšeních.

Klíčová slova: Progresivní webová aplikace, online psychoterapie, duševní zdraví, webová aplikace, vývoj, REST, Java, Spring Framework, JavaScript, React

Překlad názvu: Progresivní webová aplikace pro online psychoterapii

Contents

1 Introduction	1	7 Implementation	37
1.1 Motivation	1	7.1 Backend	37
1.2 Goals	1	7.1.1 Project Initialization	37
2 Progressive Web Application	3	7.1.2 Version control	38
2.1 What is PWA?	3	7.1.3 Persistence logic	38
2.2 PWA: Synthesis of web-applications and platform-dependent applications	3	7.1.4 Business logic	39
2.3 PWA principles	4	7.1.5 Security	42
2.4 Technical Considerations	4	7.2 Frontend	42
3 Existing solutions and competitors analysis	7	7.2.1 React Routes	42
3.1 Solutions in the Europe	7	7.2.2 Service Workers	42
3.1.1 Hedepy	7	7.2.3 Communication with server	43
3.2 Solutions in Commonwealth of Independent States	8	7.3 Application deploy	44
3.2.1 Yasno	8	7.3.1 Backend	44
3.3 Solutions in the world	9	7.3.2 Frontend	44
3.3.1 BetterHelp	9	8 Testing	45
3.3.2 TalkSpace	10	8.1 Unit testing	45
3.4 Summary	11	8.2 Integration testing	45
4 Analysis	13	8.3 User testing	46
4.1 Functional requirements	13	8.3.1 Testing scenarios	46
4.2 Non-functional requirements	14	8.4 Results	46
4.3 Use cases	15	9 Further steps	47
4.3.1 Profile-Related Use Cases	15	10 Conclusion	49
4.3.2 Session-Related Use Cases	17	Abbreviations	51
4.3.3 Non-categorised Use Cases	18	Bibliography	53
5 Application design	21		
5.1 Application architecture	21		
5.1.1 Tiers	22		
5.1.2 Service Workers	23		
5.2 Class diagram	24		
5.3 Deployment diagram	24		
5.4 User interface design	25		
5.4.1 Questionnaire design	25		
5.4.2 Schedule design	28		
6 Technologies	31		
6.1 Server-side technologies	31		
6.1.1 Framework	31		
6.1.2 HTTPS	32		
6.2 Client-side technologies	32		
6.2.1 Language	32		
6.2.2 Framework	33		
6.3 Database	35		

Figures

3.1 Hedepy	7
3.2 Yasno	8
3.3 BetterHelp	9
3.4 TalkSpace	10
4.1 Use Case Actors Diagram	15
4.2 Profile-related use cases	16
4.3 Session-related use cases	17
4.4 Non-categorised use Cases	18
5.1 Layered architecture diagram...	22
5.2 Sequence diagram.....	23
5.3 Class diagram	24
5.4 Deployment diagram	25
5.5 Design of the screen therapist's questionnaire step 1	26
5.6 Design of the screen therapist's questionnaire step 2	27
5.7 Design of the screen therapist's questionnaire step 3	28
5.8 Design of the screen therapist's schedule	29



Chapter 1

Introduction

This bachelor's thesis focuses on the development of a progressive web application (PWA) platform designed specifically for online psychotherapy. With the increasing demand for accessible and convenient mental health services, the need for an innovative and user-friendly platform has become evident.[1] The prototype aims to provide a comprehensive solution that connects clients seeking therapy with professional therapists through a secure and efficient online platform.

The field of psychotherapy has witnessed a significant shift towards online platforms in recent years. Online psychotherapy offers numerous advantages, such as eliminating geographical barriers, providing flexibility in scheduling, and ensuring privacy and confidentiality for clients. Moreover, the advancements in web technologies, particularly the emergence of PWAs, have opened up new possibilities for creating responsive and interactive applications that can function seamlessly across various devices.



1.1 Motivation

The main motivation to take the thesis with the theme of an online platform for psychotherapy is the increasing prevalence of mental health issues and the need for accessible and convenient treatment options. An online platform for psychotherapy can provide a solution by allowing individuals to receive therapy from the comfort and privacy of their own homes, potentially reducing barriers such as transportation and time constraints that can prevent people from seeking help. Additionally, an online platform can potentially expand the reach of therapy services to underserved or remote areas, and provide more flexible appointment options for busy individuals. Overall, an online platform for psychotherapy can help improve access to mental health care and support individuals in managing their mental health and well-being.



1.2 Goals

The goal of this thesis is to design and develop a prototype that leverages the capabilities of PWAs to deliver an exceptional user experience for both

clients and therapists. The prototype will incorporate features such as secure user authentication, therapist profiles, appointment scheduling, messaging capabilities, and the ability to access therapy resources. By utilizing PWA technologies, the platform will offer the convenience of offline access, push notifications, and an app-like experience without the need for installation.

Chapter 2

Progressive Web Application

The following chapter explores the concept of PWAs, describing the core concepts and the necessary steps for their creation.

2.1 What is PWA?

Progressive web application is a web application developed using the latest web development technologies that provides a user experience similar to that of a native application on any device or platform, including desktops, browsers, and smartphones. They can work offline, send push notifications, have an icon on the user's home screen, and load quickly even on slow networks.[2]

2.2 PWA: Synthesis of web-applications and platform-dependent applications

PWAs are a type of application that combine the best aspects of both web and platform-specific applications. They inherit the accessibility and versatility of web apps, which means they are available on any device with a browser installed and can be launched via a regular web link. This provides user flexibility and ease of access to the application.

At the same time, PWAs also inherit the functionality of platform applications, which provides the user with a wider range of capabilities. They can interact with the device's internal files, work offline, provide the user with quick access, and integrate with the user's device. For example, PWAs can store data for offline access and utilize the device to create a richer user experience.

To visualize which features PWAs inherit from both types of applications, below is a table:[3]

- **Service Workers** - Service Workers function as proxy servers located at the interface between the browser, the user's application, and the network. The main purpose of their use is to provide an optimized offline user experience.[5] If the requested resource is within their scope, the Service Worker decides how to provide it: whether to retrieve the resource from the local cache or download it from the network. In this way, the request is processed as if it were taking place without the direct intervention of the Service Worker.
- **Web App Manifest** - the web app manifest is a JSON file that provides metadata about the app, such as the app's name, icons, and display mode. It enables the app to be installed on the user's device, providing a native-like experience.[6]
- **Responsive Design** - PWAs should be designed to be responsive and adaptable to different screen sizes and orientations. This ensures a consistent and user-friendly experience across various devices.
- **HTTPS** - Service Worker functions exclusively in the context of HTTPS connections. This choice is due to the increased level of security that HTTPS provides over HTTP.

Chapter 3

Existing solutions and competitors analysis

In this chapter, I analyzed the existing platform for online psychotherapy. It describes popular services for delivering online therapy with their advantages and disadvantages, and further, there would be some reflection on what my product should have to be successful.

3.1 Solutions in the Europe

3.1.1 Hedepy

Hedepy¹ is a platform that aims to increase the availability of psychotherapy in Europe for everyone. The platform provides various types of psychotherapy, including individual, couples, adolescent and mentoring psychotherapy. Hedepy brings together more than 400 psychotherapists from 12 countries, speaking 12 different languages, and specializes in a wide range of psychological problems.



Figure 3.1: Hedepy

Advantages:

- appealing and user-friendly design,
- availability of reviews and certificates on the therapist's profile page,
- discounts for students,
- on the platform you can get psychological help in 12 different languages.

Disadvantages:

- there is no way to leave and read reviews in the profiles of specialists,
- sometimes bugs occur during the use of the system,
- after the specialist search test, a list of recommended psychologists appears and a video presentation with sound starts playing immediately,
- there is no possibility to choose the language in which you want to conduct sessions.

■ 3.3 Solutions in the world

■ 3.3.1 BetterHelp

BetterHelp³ is the largest online psychotherapy platform with over 300,000 psychotherapists from around the world. The service provides support through video calls, phone calls, and chat. Their services are available to clients around the clock, from any internet-connected device. This provides the convenience of accessing psychotherapy anytime and from anywhere.



Figure 3.3: BetterHelp

Advantages:

- the widest range of offered languages,
- the widest range of topics therapists work with,
- gift certificates for sessions,
- 24-hour support,
- appealing and user-friendly design.

Disadvantages:

- after registration the user is immediately provided with a therapist without the possibility of independent selection,
- too large questionnaire at registration,
- there is no information about the duration of the session on the website

²<https://yasno.live>

³<https://www.betterhelp.com>

3.4 Summary

During my analysis, I also encountered platforms with a specific focus, such as providing therapy for couples⁵, teenagers⁶, or the LGBTQ+ community⁷. Even though my work aimed to develop a platform that would assist with a wide range of issues, it was still crucial for me to examine these platforms. However, these platforms shared the same benefits and drawbacks that were mentioned earlier.

In conclusion, the existing market for online psychotherapy platforms is diverse, with various solutions targeting different geolocation markets. Four popular solutions were analyzed in this study: Hedepy, Yasno, BetterHelp, and Talkspace. Each platform has its own advantages and disadvantages. Based on this analysis, it is important to consider the functionality that the new platform will rely on. This includes features such as a comprehensive registration process, an adaptive cost system, a wide range of offered languages, the ability to choose the language in which you want to conduct sessions, independent selection of therapists, and the possibility to see the list of psychologists on the platform without registration. Additionally, it is important to consider the inclusion of reviews, certificates, and a mental health library to enhance the user experience. This information will inform the design and development of my platform and ensure that it offers a comprehensive solution for online psychotherapy services.

⁵<https://www.regain.us>

⁶<https://www.teencounseling.com>

⁷<https://www.pridecounseling.com>

Chapter 4

Analysis

This chapter describes functional and non-functional requirements, and use cases that were defined for the prototype of the PWA platform for online psychotherapy.

4.1 Functional requirements

Functional requirements are core features or functions of a product that developers must include in order for users to be able to do what they need to do.[7] For my application were identified the following functional requirements:

1. **Registration** - the system will allow new users to register for an account by logging in with their Google account credentials.
2. **Log in** - the system will provide a secure login process for registered users to access the app.
3. **Take the questionnaire** - The app will provide a questionnaire for clients to fill out, which will help determine their needs and preferences for psychotherapy.
4. **Therapists selection** - the app will match clients with suitable therapists based on the answers provided in the questionnaire.
5. **Session management** - the system will allow clients to schedule and manage appointments with their therapist.
6. **Appointment reminders** - the system will send notifications to clients to remind them of upcoming appointments and provide any relevant session information.
7. **Change of therapist** - the system will allow clients to switch to a different therapist if they are not satisfied with their current one.
8. **Upload documents during therapist's registration** - the system will allow therapists to upload and share documents related to their education, certification, and licensing during the registration process.

9. **Search for a therapist** - the system will provide search functionality for clients to find therapists based on their location, specialty, and availability.
10. **Google Meet video calls** - the system will integrate with Google Meet, a third-party video conferencing app, to provide a reliable means of conducting virtual therapy sessions between clients and their chosen therapists.
11. **Leave feedback** - the system will allow clients to provide feedback and ratings for their therapists after each session.
12. **Customization of therapist's profile** - the system will allow therapists to customize their profiles to showcase their specialties and experience.
13. **Displaying schedule** - the app will provide a feature that allows therapists to view their schedules and appointments with their clients.
14. **User profile management** - the system will provide a feature for clients and therapists to change their account information, such as email address, password, and profile picture.
15. **Confirmation or denial of therapists' requests** - the system will allow admins to confirm or deny therapists' requests according their education, work-experience and specializations they work with.

4.2 Non-functional requirements

Non-functional requirements, which are not directly related to the system's functionality, instead describe how the system should operate.[7] For my application were identified the following non-functional requirements:

1. **Download an application on a user's mobile device** - the system will provide users with the ability to install the application on their smartphones or tablets.
2. **User restraint** - the application will provide limited access to some functionality depending on the user currently logged in or their role. For example, therapists should have access to all client session notes, while clients should only have access to their own session notes.
3. **Multiplatform** - the system will be compatible with 3 major platforms (Windows, iOS, Android) and their latest versions.
4. **Responsive display** - the system should be legibly displayed on all devices with different screen sizes and resolutions, including desktops, laptops, smartphones, and tablets.
5. **Offline accessibility** - the system will provide access to essential functionality in offline mode, including viewing session notes, scheduling appointments, and updating account information.

4.3 Use cases

A use case refers to a written description of how users are expected to carry out tasks on a website.[8] To define the use cases, it's important to identify the actors who will be performing them. Therefore, based on the domain concept model, the following actors have been created.

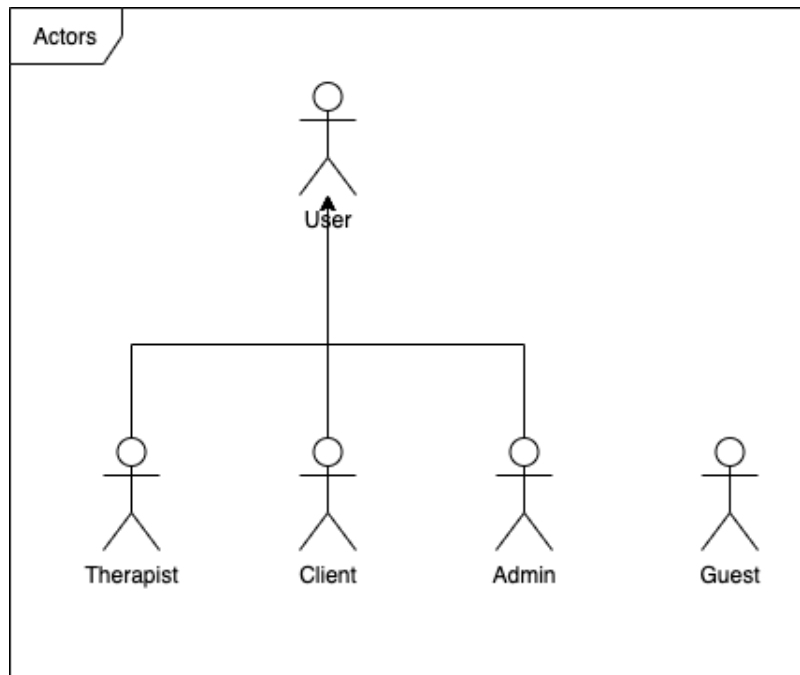


Figure 4.1: Use Case Actors Diagram

To provide a clear illustration of how users interact with the application, use case diagrams were created based on the defined functional requirements. The use cases were categorized into separate diagrams based on their specific functionalities.

4.3.1 Profile-Related Use Cases

The following list shows the functionality in the profile

1. **Create an account** - new users create an account so that their data are saved and they can get access to the full functionality of the application.
2. **Log in** - users log in using their credentials so that they can access their accounts.
3. **Log out** - users log out so that unauthorized access to their data is prevented.
4. **Edit profile** - clients and therapists edit their profiles so that their data is up-to-date.

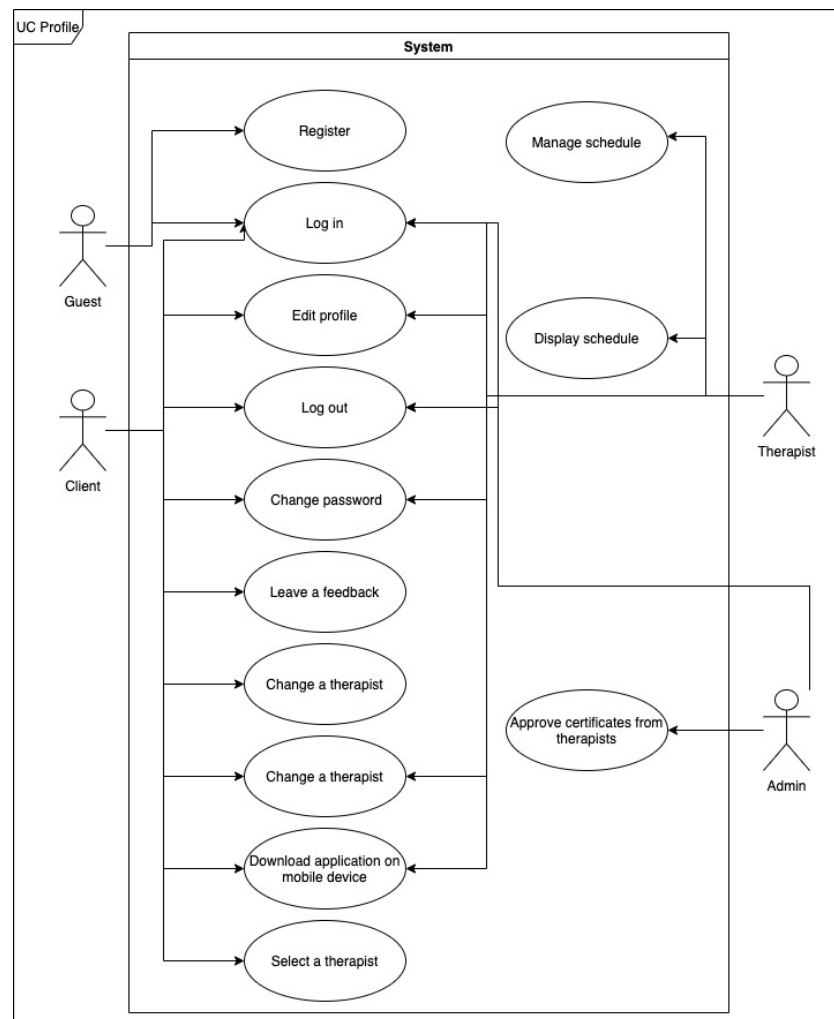


Figure 4.2: Profile-related use cases

5. **Leave feedback** - clients leave feedback to the therapist so that they can leave their opinion about the therapist's work.
6. **Change a therapist** - clients change a therapist so that they can find a new one in case they are not satisfied with the current therapist.
7. **Download application on mobile device** - users download the application on their mobile devices so that they have quick access to the application and receive push notifications.
8. **Manage schedule** - therapists manage their schedules so that they can easily schedule their appointments with patients and ensure maximum efficiency in their work.
9. **Add new qualifications** - therapists add new qualifications so that they could update their information about their education.

10. **Select the therapist** - clients select the therapist so that they can find the specialist for their needs and preferences.
11. **Display schedule** - therapists display their schedule so that they can view all reservations during a picked period of time.
12. **Approve certificates from therapists** - admins approve certificates from therapists so that they can ensure that only qualified professionals are on the platform.

4.3.2 Session-Related Use Cases

The following list shows the functionality related to the session between the therapist and the client

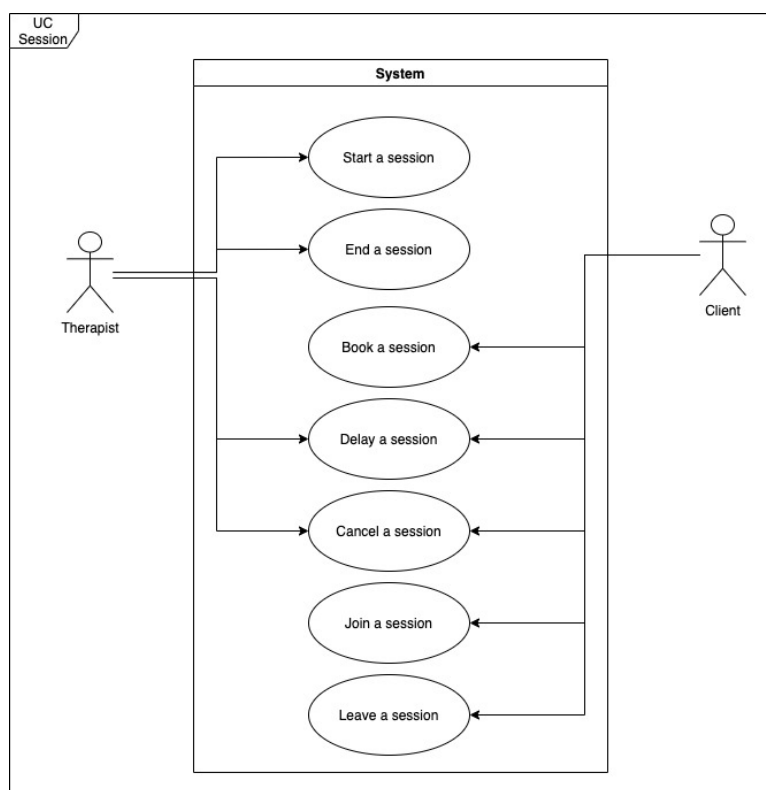


Figure 4.3: Session-related use cases

1. **Start a session** - therapists start a session so that communicate and work on solving psychological problems in real-time via video link with clients.
2. **End a session** - therapists end a session so that they can fill out a report so that clients can get an update on the work done.
3. **Join a session** - clients join a session so that they communicate and work on solving psychological problems in real-time via video link with therapists.

4. **Leave a session** - clients leave a session so that they can view reports done by therapists.
5. **Book a session** - clients book a session so that they can get help from the chosen therapist at the chosen time.
6. **Delay session** - clients and therapists delay sessions so that they can choose a suitable time if the session was picked at an inconvenient time.
7. **Cancel session** - clients and therapists cancel sessions so that they can free the selected time.

4.3.3 Non-categorised Use Cases

The following list shows functionality with no category.

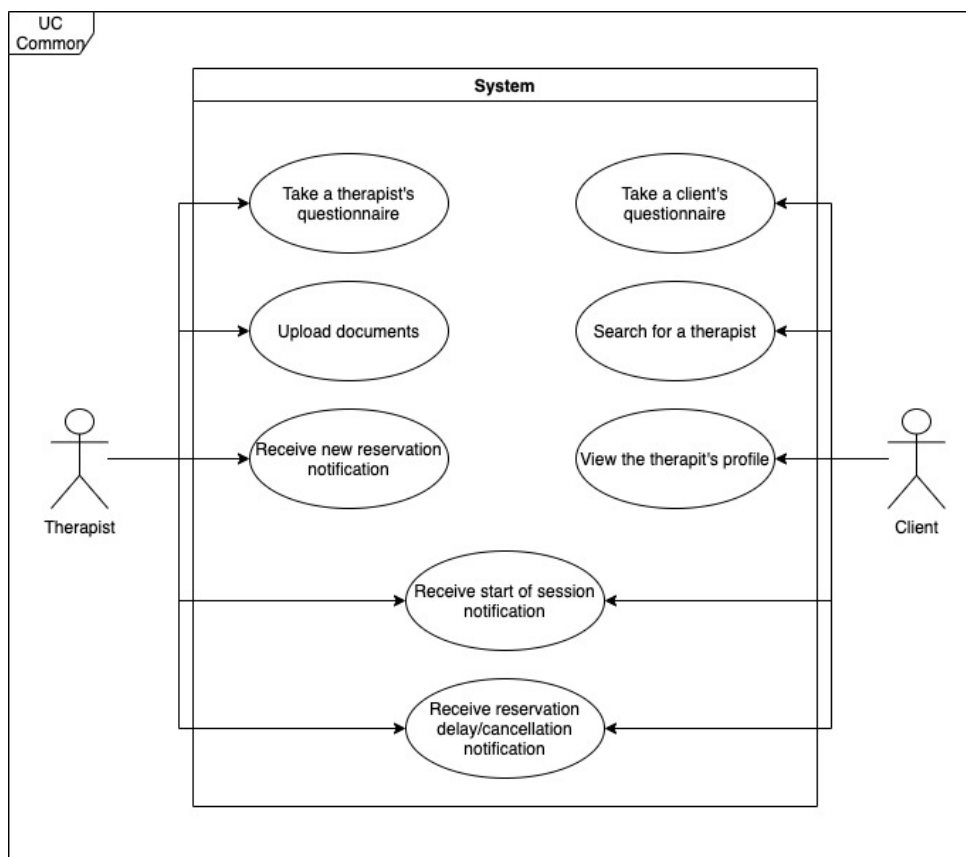


Figure 4.4: Non-categorised use Cases

1. **Take a client's questionnaire** - clients take a questionnaire so that the system can choose for them suitable therapists.
2. **Take a therapist's questionnaire** - therapists take a questionnaire so that they can finish the registration process.

3. **Upload documents** - therapists upload documents so that they can prove their experience and skills with educational diplomas and certificates.
4. **Search for a therapist** - clients search for a therapist so that they can choose the most preferable one.
5. **View the therapist's profile** - clients view the therapist's profile so that they can get more information about the viewed therapist.
6. **Receive start of session notifications** - clients and therapists receive notifications so that they are aware of the session that has begun.
7. **Receive new reservation notifications** - therapists receive notifications so that they are aware of a new reservation.
8. **Receive reservation delay/cancellation notifications** - clients and therapists receive notifications so that they are aware of delayed or canceled reservations.



Chapter 5

Application design



5.1 Application architecture

The project does not involve the development of a large and complex application. Initially, it was decided to consider only monolithic architectures, specifically, the layered architecture style. A monolithic system means an application in which everything is deployed as a single functional unit.[9]

Layered architecture style is a method of organizing an application in which its structure is broken down into horizontal layers, each of which is responsible for performing well-defined functions and tasks.[10] The number of layers can vary, but in the context of this project it was decided to limit ourselves to three key layers: the Presentation Layer, the Application Layer and the Database Layer. There are two fundamental reasons for this choice: ease of development and the concept of 'Layers of Isolation'.

The concept of 'Layers of Isolation' is to strictly isolate the layers of the system from each other.[10] This implies that when a request is passed from the upper layers of the application to the lower layers, there is an explicitly defined order that determines which layer should process the request and in what sequence. This isolation of layers provides a reliable separation of functional areas and allows changes made in one layer not to affect other layers of the system, which also simplifies the development process.

On the following image

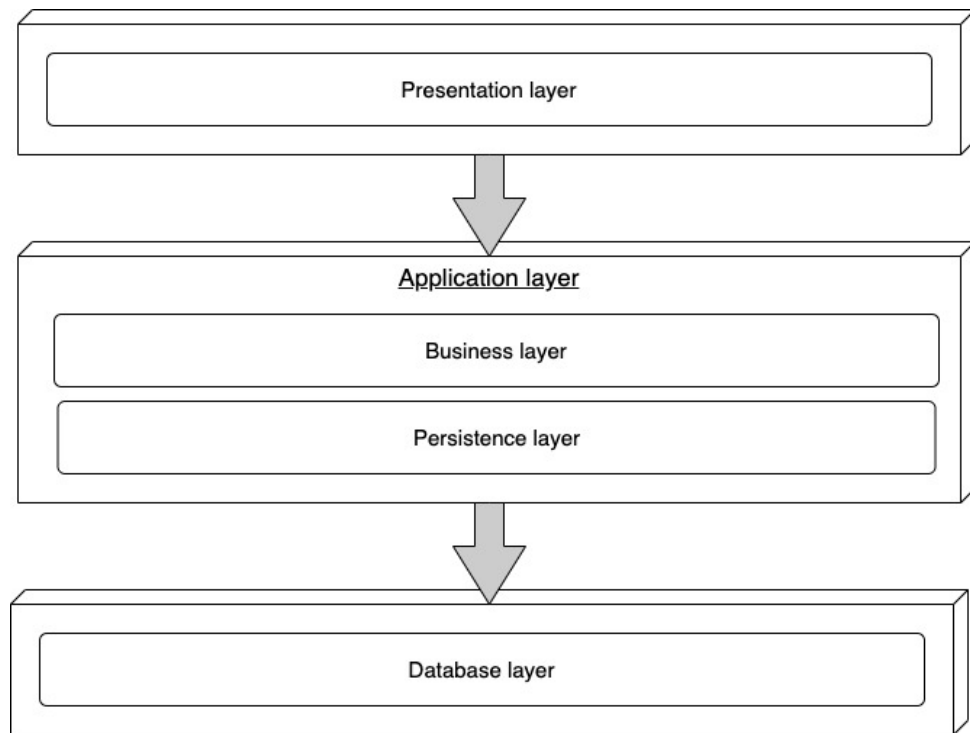


Figure 5.1: Layered architecture diagram

■ 5.1.1 Tiers

■ Presentation layer

The Presentation Layer acts as a representation of the UI and serves as a key component for data visualization. This layer is responsible for displaying information received from the server in a format that is understandable and accessible to users.

■ Application layer

Application layer is a module that includes business logic and data persistence logic. Business logic is a component of the application responsible for processing of incoming requests and generating appropriate responses for the client. Persistence logic deals with the operations of recording and retrieving information from the database.

In this project it was decided to combine these two levels into one module, which allows centralized management of both logic systems, simplifying the development and testing process.

■ Database layer

The data storage tier is responsible for storing all the data used by the app. This layer is where the app's data is stored in a database, which can be

accessed by the application layer as required. The database used in this app is a relational database that stores data in tables.

5.1.2 Service Workers

Service workers as a part of Presentation layer are a key component of the PWA applications architecture. They are background scripts that run in the browser and can run independently of open tabs. Their role is to cache static content, such as HTML, CSS, JavaScript, and images, using the Cache API.[11] In addition to caching, service workers enable PWA applications to persist and run in the background even after the browser tab is closed. They also have the capability to handle push notifications and background sync, further enhancing the functionality and user experience of the PWA app.

For a better understanding how Service workers work in background here is the sequence diagram describing how Service worker is registered, how it caches the prescribed files and how it extracts these files from cache:

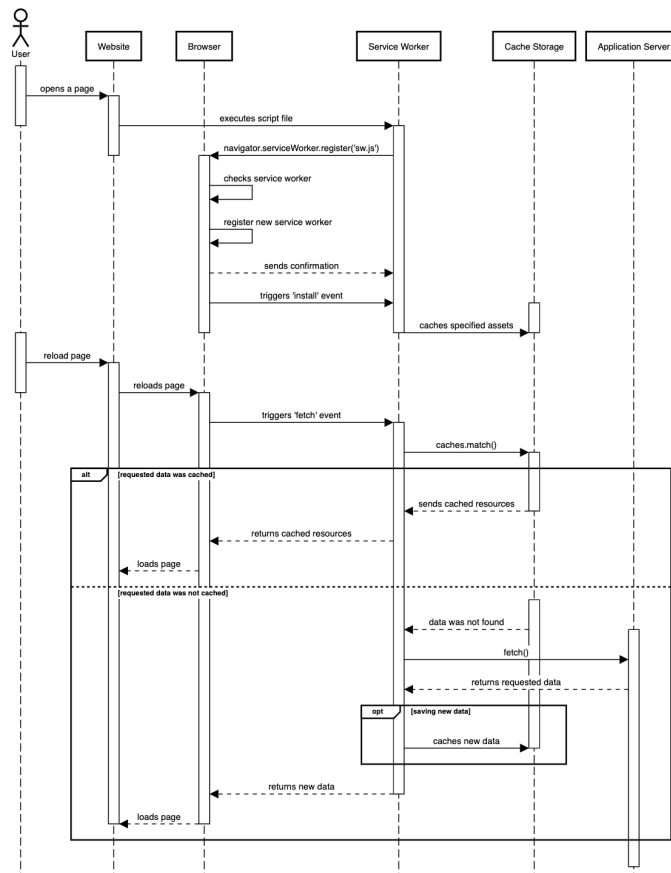


Figure 5.2: Sequence diagram

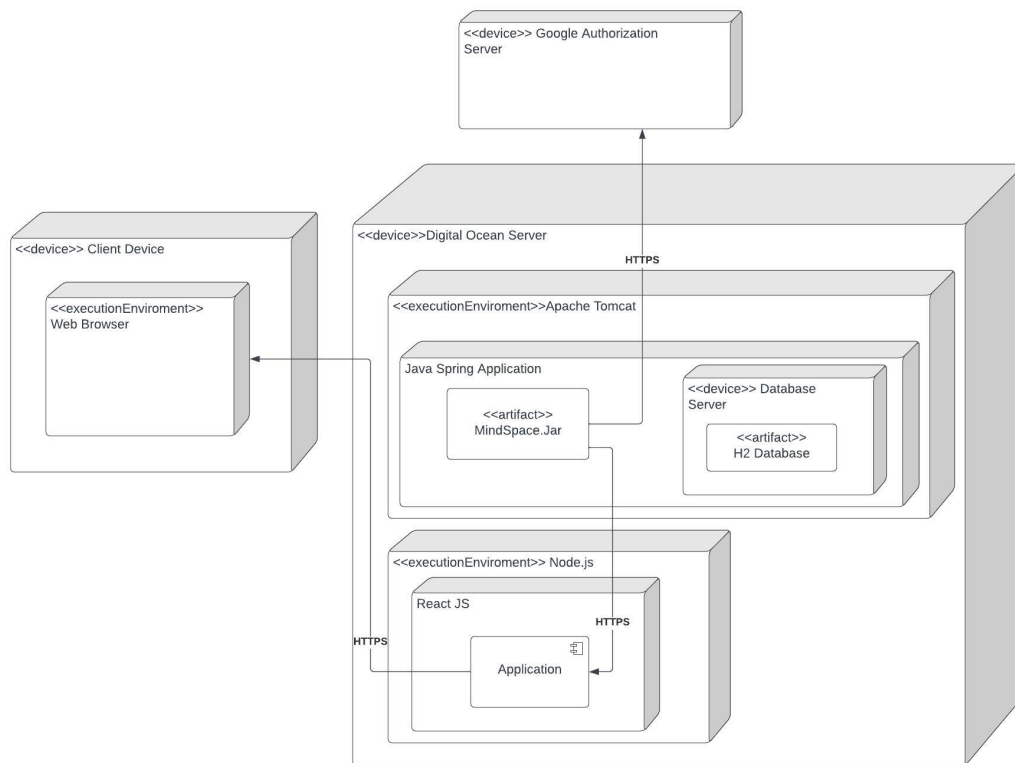


Figure 5.4: Deployment diagram


5.4 User interface design

Based on the analysis of functional and non-functional requirements, a high fidelity prototype was developed using the Figma tool. This prototype was developed according to the key principles of UI design, including intuitiveness, attractiveness, consistency and usability.

5.4.1 Questionnaire design

The application is structured into three roles, but the main two are the client and the therapist. One of the key functions of the app is the process of completing a questionnaire when registering. The answers provided in the questionnaire allow clients to find suitable therapists and psychotherapists to apply to be added to the list of qualified professionals. The questionnaires for both roles are separated into several logical segments.

Below are images 5.6, 5.5 and 5.6 showing all stages of the questionnaire for therapists:

Home Our therapists **For therapists** Log in Find a therapist

1 Personal information **2** Education **3** Contact information

Working with MindSpace means:

1. Quick client acquisition.
2. The opportunity to work from any convenient location. Sessions are conducted online.
3. An established payment system from clients that doesn't require your involvement. You simply receive payments for sessions directly to your card.
4. Convenient and straightforward service: a personal dashboard with a list of clients, scheduling, video communication – all on the MindSpace platform.
5. Involvement in a dynamic and growing community with selected experienced psychotherapists. Exclusive webinars from top specialists.
6. Free group supervision sessions.

How to start collaborating with MindSpace: Please fill out the application. If your experience and education match, we will invite you for an interview.

Firstname

Surname

Date of birth

Tell us about yourself in free form. What do you think we need to know about you to understand what kind of specialist you are?

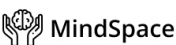
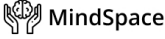
Home About Services Contact

Figure 5.5: Design of the screen therapist's questionnaire step 1



[Home](#)
[Our therapists](#)
[For therapists](#)
[Log in](#)
[Find a therapist](#)

✓ Personal information

2 Professional information

3 Contact information


What is your higher education? Write about basic psychological (related) training or retraining: 1. Year of graduation
 2. Name of institution of higher education 3. Name of department and specialization 4. Indicate academic degree (bachelor's, master's) or academic degree (if any).*

Autosize height based on content lines

Choose the topics you work with:


- Depression and anxiety
- Interpersonal relationships
- Stress and emotion management
- Self-esteem and self-acceptance
- Trauma and loss
- Addictions and habits
- Personal growth and development
- Professional issues
- Emotional and psychosomatic problems
- Sexual issues

Please attach photos of detailed diplomas and certificates confirming the training. Required documents:
 (1) diploma of basic psychological (related) training / retraining (2) documents of training in the method.
 If the training is not completed, please attach a certificate from the training institution.



Click or drag file to this area to upload
Support for a single or bulk upload. Maximum file size 2MB.

If the documents you are attaching indicate a different last and first name, attach documentation to support the change in personal information



Click or drag file to this area to upload
Support for a single or bulk upload. Maximum file size 2MB.

Do you belong to any psychotherapeutic community. If yes, which one?

Autosize height based on content lines

When did you start counseling? For money, not as part of a training program. Be sure to write the month, not just the year.

example

Do you have any online experience? If yes, how many years?

Autosize height based on content lines

How many clients do you have in your practice right now?

example

How long did the longest therapy session among your clients take? (Specify years, months or number of sessions)

example

Are you in personal psychotherapy?

Please select

Do you undergo regular supervisions?

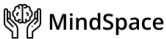
Please select

In which languages you can lead sessions?

example

Back

Next step



[Home](#)
[About](#)
[Services](#)
[Contact](#)

Figure 5.6: Design of the screen therapist's questionnaire step 2

MindSpace Home Our therapists For therapists Log in Find a therapist

Personal information Professional information 3 Contact information

Attach your photo. Photo requirements: 1/ Color 2/ Face in the center and well-lit 3/ Size not less than 1MB 4/ In the format (name).jpg

Click or drag file to this area to upload
Support for a single or bulk upload. Maximum file size 2MB.

Phone number to contact
example

E-mail address for contacting
example

Password
example

Repeat password
example

Back Finish

MindSpace Home About Services Contact

Figure 5.7: Design of the screen therapist's questionnaire step 3

5.4.2 Schedule design

The second key aspect that was focused on during the prototyping process was the display of the timetable. It was important to ensure that the schedule was clear and familiar to most users. An interface similar to Google Calendar, which is widely known and popular, was chosen as a reference. Below is an image illustrating the schedules for therapists:

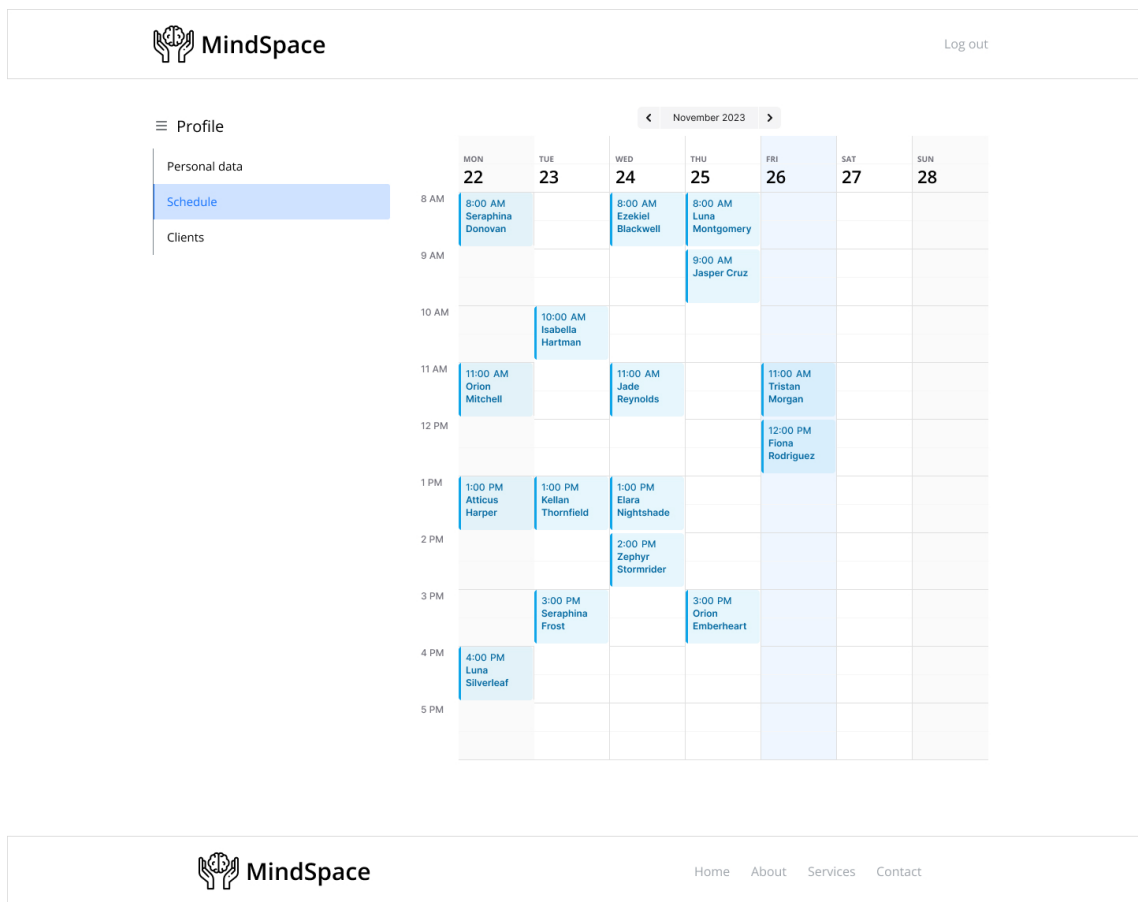


Figure 5.8: Design of the screen therapist's schedule

Chapter 6

Technologies

This chapter will provide an introduction to the selected technologies utilized in the development of the application. These technologies were chosen to ensure the effective implementation of the project concept and the creation of a functional software solution.

6.1 Server-side technologies

This section explored the server-side technologies employed in the development of the application.

6.1.1 Framework

Spring

A fundamental idea behind the Spring Framework is to employ Dependency Injection (DI) and Inversion of Control (IoC). These ideas assist in streamlining the procedure of creating and managing objects, decreasing the association between application components, and enhancing the adaptability and scalability of the application. By utilizing DI and IoC, the Spring Framework permits the development of easily maintainable and testable applications.

In the Spring Framework, DI and IoC operate by not creating objects directly within the code. Instead, the objects are described in configuration files and are created and managed by the Spring container. The Spring container is responsible for instantiating objects and providing dependencies between them according to specific rules.

This technique reduces the complexity and association of code. Instead of objects creating each other directly, the container provides the dependencies. This leads to more adaptable and scalable code, allowing for easy replacement of one implementation with another without requiring changes to the code that uses the dependency.[12]

My decision to use the Spring framework for my application's backend was based on a combination of my past experience with it and the convenience it offers in developing the northern part of the application. This choice allowed me to allocate more time to client-side development, which is crucial for PWA

applications. Overall, the Spring framework proved to be the best fit for my needs.

■ Spring Boot

Spring Boot is an open-source Java-based framework used to create and deploy stand-alone, production-grade Spring-based applications quickly and easily. It provides an opinionated approach to application configuration and development by reducing the amount of boilerplate code required. With Spring Boot, developers can create web applications, RESTful services, and other types of applications with minimal effort. It also includes several pre-configured features and libraries, such as embedded servers, which makes it easy to develop and deploy Spring-based applications. Spring Boot also supports a wide range of tools and plugins, such as Maven and Gradle, which simplify the build and deployment process.[13]

■ 6.1.2 HTTPS

HTTPS is a crucial requirement for PWA. It ensures that the communication between the user's browser and the web server is encrypted and secure, preventing any third-party from eavesdropping or tampering with the transmitted data. This is especially important for PWA applications that handle sensitive user data, such as personal information, login credentials, and financial transactions. HTTPS also enables the PWA application to be installed on the user's device, as most modern web browsers require a secure connection for this feature to work. As a result, the use of HTTPS in PWA enhances user trust, privacy, and security, making it an essential aspect of modern web development.

■ 6.2 Client-side technologies

The following section will explore the client-side technologies utilized in the application's development.

■ 6.2.1 Language

■ JavaScript

JavaScript (JS) is a high-level, interpreted programming language that is commonly used to add interactivity and dynamic behavior to web pages. JavaScript is a flexible language that supports a range of programming paradigms, including object-oriented, functional, and imperative programming styles. It has a variety of built-in data types and objects, including arrays, strings, numbers, and functions, and it supports dynamic typing, which means that variables do not need to be declared with a specific data type.

¹Web Services

JavaScript is used to create dynamic web content, such as animations, pop-ups, and form validation. It is also commonly used to create interactive web applications, such as single-page applications (SPAs) and progressive web applications.[16]

■ TypeScript

TypeScript is a language that is a typed superset of JavaScript. It is designed to add rules about how different kinds of values can be used, and to check a program for errors before execution based on the kinds of values it uses, thus being a static type checker. TypeScript preserves the runtime behavior of JavaScript, which means that if code was moved from JavaScript to TypeScript, it will run the same way, even if TypeScript thinks that the code has type errors. TypeScript erases the types once the compiler is done checking the code, so the resulting compiled code has no type information. TypeScript's static type checking is useful when writing larger applications, as it helps catch errors before they occur and makes it easier to manage larger codebases.[17]

■ Choice

When selecting the technology stack for my PWA, I opted for JavaScript. This was primarily due to my prior experience with the language, its alignment with the project requirements, and the wealth of available resources on PWA development that were primarily based on JavaScript.

■ 6.2.2 Framework

■ React

React is an open-source JavaScript library used for building user interfaces (UIs) and developing single-page applications. It was developed by Facebook and is now maintained by Facebook and a community of developers. React is a popular choice for building UIs due to its simplicity, performance, and flexibility.

Here are some of the key advantages of using React:[19]

1. **Component-based architecture:** React is based on a component-based architecture, which means that UIs are broken down into small, reusable components. This makes it easier to manage complex UIs and enables developers to reuse code across different parts of the application.
2. **Declarative programming:** React uses a declarative programming approach, which means that developers describe the desired state of the UI, rather than imperatively defining how the UI should be updated. This makes it easier to reason about the application's behavior and reduces the likelihood of bugs and errors.

■ 6.3 Database

The choice of the H2 database for this project is due to several reasons, described below. The first is that H2 is compact and embedded[21], ideal for development and testing as it does not require a separate server to be installed. The second is that H2 offers fast data access, helping to speed up the development process.[21] In addition, H2 is developed in Java, which guarantees its compatibility with Spring Boot and facilitates integration.[21] Moreover, H2 supports basic SQL functions, which makes it functional enough to create a prototype application.

Chapter 7

Implementation

In this chapter, I explained how the project was implemented by following the analysis and utilizing the chosen technologies. The goal was to provide a comprehensive understanding of how the project concept was turned into a fully functioning software solution.

7.1 Backend

7.1.1 Project Initialization

After conceptualizing the project, the initial step was to initialize the Spring Boot project using the Spring Initializr tool. This powerful tool simplifies the project setup process by providing a user-friendly interface to configure essential project details.

In the Spring Initializr, the project dependencies were carefully chosen to align with the project requirements. The selected dependencies were as follows:

1. **Spring Web:** This dependency facilitated the development of web applications, providing essential components and features for handling HTTP requests and responses.
2. **Lombok:** By including the Lombok dependency, the project gained the advantage of reducing boilerplate code. Lombok's annotations allowed for the automatic generation of common code snippets, streamlining the development process.
3. **Spring Data JPA:** The Spring Data JPA dependency integrated seamlessly with the project, enabling easy and efficient data access through the Java Persistence API (JPA). This simplified database interactions and provided convenient repository abstractions.
4. **H2 Database:** The H2 Database dependency added support for an in-memory database, allowing for quick and efficient testing and development. It provided a lightweight and easily manageable database solution.

5. **Spring Security:** The Spring Security dependency was included to enhance the project's security capabilities. It provided robust authentication and authorization mechanisms, safeguarding the application against unauthorized access.
6. **OAuth2 Client:** The OAuth2 Client dependency enabled seamless integration with OAuth2 providers, facilitating secure and standardized authentication and authorization workflows.

7.1.2 Version control

After initializing the project, the next step was setting up the local development environment, which included implementing a version control system. Version control systems are essential tools in software development, offering various benefits and simplifying the development process.

Git, the chosen version control system for this project, provided a reliable framework for monitoring code alterations. It allowed me to track the project's progress, revisit previous versions, and address any issues that arose.

7.1.3 Persistence logic

At the initial stage of development of the server part of the system, the main focus was on the creation and implementation of persistence logic.

Model

The classes located in the 'model' folder are annotated using `@Entity`. They represent tables in the database that correspond to the application's data structures.

Here is an example of one entity class below:

```
@Entity
@Table(name = "users")
@NoArgsConstructor
@AllArgsConstructor
@Getter @Setter
public abstract class User extends AbstractEntity {
    private String name;

    private String surname;

    @Column(unique = true)
    private String phoneNumber;

    @Column(unique = true)
    private String email;

    private String password;
```

```

    @Enumerated(EnumType.STRING)
    private UserType userType;

    public enum UserType {
        CLIENT, THERAPIST, ADMIN;
    }

    private boolean registrationFinished;

    @Enumerated(EnumType.STRING)
    private Gender gender;

    public enum Gender {
        MALE, FEMALE, NOT_STATED
    }
}

```

■ Repository

Repository classes are created to interact with the database. They are responsible for creating, reading, updating and deleting (CRUD) data. The `@Repository` annotation is used to denote such classes. In our project, these classes implement the `JpaRepository` interface, which includes basic CRUD operations as well as functionality for pagination and sorting data.

Here is an example of one repository class below:

```

@Repository
public interface TherapistRepository extends JpaRepository<Therapist, Integer> {
}

```

■ 7.1.4 Business logic

The business logic describes all the basic functionality of the application

■ Service

In the services layer, classes are labeled with the `@Service` annotation. They describe data processing and coordination between different system components.

Here is an example of time cell service method that generates time cells for 30 days in advance in a schedule which belongs to one of the therapists in a system:

```

@Override
public void generateTimeCells(Schedule schedule) {
    LocalDateTime startDateTime = LocalDateTime.now()
        .plusDays(1)
}

```

```

        .withHour(0)
        .withMinute(0)
        .withSecond(0)
        .withNano(0);
    LocalDateTime endDateTime = startDateTime.plusDays(30);

    for (LocalDateTime date = startDateTime;
        date.isBefore(endDateTime);
        date = date.plusDays(1)) {
        for (int hour = 9; hour <= 17; hour++) {
            TimeCell timeCell = new TimeCell();
            LocalDateTime startTime = LocalDateTime
                .of(date.toLocalDate(), LocalTime.of(hour, 0));
            LocalDateTime endTime = startTime.plusHours(1);

            timeCell.setStartTime(startTime);
            timeCell.setEndTime(endTime);
            timeCell.setSchedule(schedule);

            timeCellRepository.save(timeCell);
        }
    }
}

```

■ DTO

DTO classes placed in the API folder do not participate in the business logic. However, they serve as wrappers for data, making it easier to transfer information between different parts of the system.

Here is an example of one DTO class below:

```

public record UserRequest(
    String name,
    String surname,
    String number,
    String email
) {
}

```

■ Controller

Controllers are responsible for receiving requests coming from the frontend, delegating those requests by calling the appropriate services, and generating responses that are then passed back to the frontend. Controller classes are marked with the `@RestController` annotation, which is used to create RESTful web services.^[22]

Here is an example of one controller class below:


```
@RestController
@RequestMapping("/reservations")
public class ReservationController {

    private final ReservationServiceImpl reservationService;

    public ReservationController(ReservationServiceImpl reservationService) {
        this.reservationService = reservationService;
    }

    @PostMapping
    public ResponseEntity<CreateReservationResponse> createReservation(
        @RequestBody ReservationRequest request
    ) {
        return new ResponseEntity<>(reservationService
            .createReservation(request), HttpStatus.CREATED
        );
    }

    @GetMapping("/{id}")
    public ResponseEntity<ReservationResponse> getReservation(
        @PathVariable Integer id
    ) {
        return new ResponseEntity<>(
            reservationService.getReservation(id), HttpStatus.OK
        );
    }

    @DeleteMapping("/{id}")
    public ResponseEntity<Void> cancelReservation(@PathVariable Integer id) {
        reservationService.cancelReservation(id);
        return ResponseEntity.noContent().build();
    }

    @PutMapping("/{reservationId}/delay/{timeCellId}")
    public ResponseEntity<Void> delayReservation(
        @PathVariable Integer reservationId,
        @PathVariable Integer timeCellId
    ) {
        reservationService.delayReservation(reservationId, timeCellId);
        return ResponseEntity.noContent().build();
    }
}
```



```

        registration
      );
    },
    (error) => {
      console.error('Service worker registration failed: ${error}');
    },
  );
} else {
  console.error("Service workers are not supported.");
}

```

Also below is an example of how the service worker sends a notification that the user has problems with the internet connection.

```

self.addEventListener('fetch', function(event) {
  event.respondWith(
    fetch(event.request).catch(function() {
      self.registration.showNotification("No Internet Connection", {
        body: "You are currently offline. Some features may not be available.",
      });
    })
  );
});

```

7.2.3 Communication with server

TanStack Query library was used in order to optimize the code and reduce repetitions when implementing queries to the server. This library is a powerful tool for asynchronous state management in TypeScript/JavaScript.[24] In the default.js file, functions were developed for basic HTTP methods including GET, POST, PUT, and DELETE.

An example of one such function is shown below:

```

export async function makeGetRequest(url, accessToken) {
  const response = await fetch(url, {
    method: "GET",
    headers: {
      "Authorization": `Bearer ${accessToken}`
    }
  });

  if (!response.ok) {
    throw new Error('Error while making GET request to ${url}')
  }

  return response.json();
}

```

Queries that did not require asynchronous control were defined in separate components using TanStack Query.

Here is an example how a query looks like:

```
const {
  isPending,
  isError,
  data,
  isFetched,
  error
} = useQuery({
  queryKey: ['clientProfile'],
  queryFn: () => fetchCurrentUser(accessToken)
})
```

7.3 Application deploy

This section describes the backend and frontend deployment process.

7.3.1 Backend

In the root folder of the project there is a directory `.github/workflows`, where the `deploy.yml` file is located. This file defines a pipeline that tracks commits in the main branch. When a new commit occurs, GitHub Actions will initiate the project build process. The JAR file targets the target directory. A Dockerfile is then created, which generates a Docker image that is then fluffed in Docker Hub. At the same time, DigitalOcean has a project set up that monitors changes to Docker Hub. From there, a Docker image with the tag 'latest' is automatically extracted. In DigitalOcean, a container for the backend is automatically launched from this image. After each change in the main branch, this process is repeated, allowing the application to function on a secure HTTPS connection.

7.3.2 Frontend

Digital Ocean accesses the `mind-space` folder where the source code for the frontend is located. Digital Ocean recognizes the project as a React application and copies its code base to its server.

Chapter 8

Testing

This chapter focuses on the testing approach used in application development. It discusses various testing methodologies, including unit testing, integration testing, and user testing, to ensure quality application functionality.

8.1 Unit testing

In this application, unit testing is performed using the JUnit and Mockito frameworks. JUnit, widely used in Java, structures the writing and execution of unit tests through annotations, assertions, and test runs.[25] Mockito allows to create mock objects to simulate dependencies, providing isolation of the parts of the code under test.[26] The application implements unit tests for services, in particular CRUD operations, checking the correct behavior and functionality of services and controllers.

8.2 Integration testing

Integration testing is used to verify the functionality and compatibility of controllers. Testing frameworks such as JUnit and Mockito facilitate seamless integration of application components. JUnit provides a structured approach to defining and executing integration tests. Mockito is used to create object mockups that mimic dependencies and external systems. Controller testing ensured that API endpoints functioned correctly, ensuring successful interaction with underlying services and increasing the reliability of the application.

Tested controllers in the application:

1. AdminController,
2. ClientsController,
3. ReservationController,
4. TherapistController,
5. UsersController.

■ 8.3 User testing

This section describes user testing of the application by independent users. Test scenarios were created for testing purposes, and the testing process is described below.

■ 8.3.1 Testing scenarios

Since the application has different roles, the test scenarios that define specific user actions to verify functionality are divided into therapist and client scenarios.

■ Client

Test scenarios for clients include a description of basic functions using use cases such as taking a questionnaire, selecting a therapist, making a reservation, and joining a session.

■ Therapist

To test the application from the therapist's side, scenarios were developed that involve sending a application. Once the application is approved, the therapist opens their profile, views their schedule and joins a session with a specific client.

■ 8.4 Results

Tests of the developed application were performed through unit tests to analyze separate parts of the code. The backend operation was analyzed through integration testing. The frontend was tested through accessibility tests to make sure it works properly in various modern browsers. Further, the application was offered for end-user testing. Based on the results of these tests, methods for improving the application were identified, as described in the Further Steps chapter.



Chapter 9

Further steps

The prototype application already provides basic functions to demonstrate the key features of the system. Based on the results of testing the application, directions for its further development were identified:

1. Implementing a built-in system for video calls, improving communication capabilities.
2. Extending the user interface, including adding a dark theme and support for multiple languages, to improve usability.
3. Adding a chat function between client and therapist, ensuring effective communication.
4. Introducing a payment system, simplifying financial transactions between users and the service.



Chapter 10

Conclusion

In this bachelor's thesis, the goals were achieved: a prototype platform of a progressive web application for online psychotherapy services was designed and implemented. Based on the study of PWA technology and analysis of existing solutions, a reliable system architecture was developed. The technology selection and implementation process resulted in a functional prototype.

Using the capabilities of PWA, the platform offers clients a convenient way to connect with professional psychotherapists. The thesis also identified potential areas for future enhancements. These improvements promise to further enhance the functionality of the platform.

As a result, this work becomes the basis for further development of the platform and offers a promising solution to meet the changing needs of clients and therapists.



Abbreviations

ACID	Atomicity, Consistency, Isolation, Durability
API	Application Programming Interface
CORS	Cross-Origin Resource Sharing
CRUD	Create, Read, Update, Delete
CSS	Cascading Style Sheets
CSV	Comma-Separated Values
DI	Dependency Injection
DOM	Document Object Model
HTML	Hypertext Markup Language
HTTP	Hypertext Transfer Protocol
HTTPS	Hypertext Transfer Protocol Secure
IoC	Inversion of Control
JPA	Java Persistence API
JS	JavaScript
JSON	JavaScript Object Notation
ORDBMS	Object-Relational Database Management System
PWA	Progressive Web Application
REST	Representational State Transfer
RSS	Remote Service Specification
SOAP	Simple Object Access Protocol
SQL	Structured Query Language

SSL Secure Sockets Layer

TLS Transport Layer Security

UI User Interface

URL Uniform Resource Locator

XML Extensible Markup Language



Bibliography

- [1] *Increasing demand for mental health services*, [online]. URL: <https://www.apa.org/pubs/reports/practitioner/2022-covid-psychologist-workload>
- [2] Tal Alter. *Building Progressive Web Apps* Publisher O'Reilly, 2017.
- [3] *Progressive Web Apps*, [online]. URL: <https://web.dev/learn/pwa/progressive-web-apps>
- [4] *Progressive Web Apps: Escaping Tabs Without Losing Our Soul*, [online]. URL: <https://infrequently.org/2015/06/progressive-apps-escaping-tabs-without-losing-our-soul/>
- [5] *Service Workers*, [online]. URL: https://developer.mozilla.org/en-US/docs/Web/API/Service_Worker_API
- [6] *Web App Manifest*, [online]. URL: <https://learn.microsoft.com/en-us/microsoft-edge/progressive-web-apps-chromium/how-to/>
- [7] *Functional and Nonfunctional Requirements: Specification and Types*, [online]. URL: <https://www.altexsoft.com/blog/business/functional-and-non-functional-requirements-specification-and-types/#:~:text=Functional%20requirements%20are%20product%20features,system%20behavior%20under%20specific%20conditions.>
- [8] *Use Cases*, [online]. URL: <https://www.usability.gov/how-to-and-tools/methods/use-cases.html#:~:text=A%20use%20case%20is%20a,when%20that%20goal%20is%20fulfilled.>
- [9] *Analysis of Monolithic and Distributed Systems – Learn System Design*, [online]. URL: <https://www.geeksforgeeks.org/analysis-of-monolithic-and-distributed-systems-learn-system-design/>
- [10] Mark Richards, Neal Ford. *Fundamentals of Software Architecture* Publisher O'Reilly, 2020.

- [11] *Making PWAs work offline with Service workers*, [online]. URL: https://developer.mozilla.org/en-US/docs/Web/Progressive_web_apps/Tutorials/js13kGames/Offline_Service_workers
- [12] *Spring*, [online]. URL: <https://docs.spring.io/spring-framework/docs/current/reference/html/core.html#spring-core>
- [13] *What is Java Spring Boot*, [online]. URL: <https://www.ibm.com/topics/java-spring-boot>
- [14] *How to make PWAs installable*, [online]. URL: https://developer.mozilla.org/en-US/docs/Web/Progressive_web_apps/Tutorials/js13kGames/Installable_PWAs
- [15] *What is HTTPS?*, [online]. URL: <https://www.ssl.com/faqs/what-is-https/>
- [16] *JavaScript*, [online]. URL: <https://developer.mozilla.org/en-US/docs/Web/JavaScript>
- [17] *TypeScript*, [online]. URL: <https://devdocs.io/typescript/typescript-from-scratch>
- [18] *React*, [online]. URL: <https://legacy.reactjs.org/docs/getting-started.html>
- [19] *React advantages*, [online]. URL: <https://www.fastcomet.com/blog/advantages-and-disadvantages-of-react-js>
- [20] *Angular*, [online]. URL: <https://angular.io/guide/what-is-angular>
- [21] *H2 Database engine*, [online]. URL: <https://www.h2database.com/html/main.html>
- [22] *Spring controller*, [online]. URL: <https://www.baeldung.com/spring-controller-vs-restcontroller>
- [23] *React Router overview*, [online]. URL: <https://reactrouter.com/en/main/start/overview>
- [24] *TanStack Query*, [online]. URL: <https://tanstack.com/query/latest>
- [25] *JUnit 5*, [online]. URL: <https://junit.org/junit5/docs/current/user-guide/>
- [26] *Mockito*, [online]. URL: <https://site.mockito.org>