

Универзитет „Св. Кирил и Методиј“ во Скопје

Факултет за информатички науки и компјутерско инженерство

Дипломска работа

**Апликација за резервација на место во канцеларија - употреба на   
.NET и React**

|  |  |
| --- | --- |
| **Ментор**  Проф. д-р Магдалена Костоска | **Кандидат**  Сања Срњакова  171025 |

Ноември, 2025

# Апстракт

Оваа дипломска работа опишува развој на веб-систем за менаџирање на просторот и паркингот во една канцеларија – со цел да се намалат трошоците за неискористени канцелариски бироа и простории, како и рамномерно распоредување на вработените преку неделата за Work from Office Days. Системот го користат 4 вида на корисници: Вработен, Менаџер на оддел, Регионален Менаџер и Администратор. Вработениот има можност преку системот да избере ден кога ќе сака да работи од канцеларија и да посочи кои ресурси му се потребни (канцелариско биро или паркин) и да го следи статусот на истото. Менаџерот има преглед од сите барања во неговиот оддел, може да направи резервација на конференциска сала и има за задача да го одобри или одбие барањето на вработениот во зависност од слободните места кои ги има на располагање. Регионалниот менаџер може да додаде или отстрани нов вработен, да резервира конференциска сала, да додаде оддел и да администрира со тоа кој има улога на Менаџер на оддел. Постои и улога на Администратор – која менаџира со целиот систем и ги има апсолутно сите пермисии за додавање, уредување и бришење во рамки на постоечките функционалности.

Решението е реализирано со .NET 8 (REST API, EF Core, CQRS, Hangfire) серверска логика, React 18 / Material UI за динамичен кориснички интерфејс и Microsoft SQL Server за перзистенција на податоци.

**Клучни зборови**: резервации, хибридна работа, ресурси во канцеларија, .NET 8, React, SQL Server.

**Содржина**

[Апстракт ii](#_Toc191661324)

[1. Вовед 1](#_Toc191661325)

[2. Сродни истражувања Error! Bookmark not defined.](#_Toc191661326)

[3. Методологија Error! Bookmark not defined.](#_Toc191661327)

[3. 1. Слики **Error! Bookmark not defined.**](#_Toc191661328)

[3. 2. Табели **Error! Bookmark not defined.**](#_Toc191661329)

[3. 3. Листи **Error! Bookmark not defined.**](#_Toc191661330)

[3. 3. 1. Подредени листи 10](#_Toc191661331)

[3. 3. 2. Неподредени листи **Error! Bookmark not defined.**](#_Toc191661332)

[3. 4. Математички формули **Error! Bookmark not defined.**](#_Toc191661333)

[3. 5. Кодови **Error! Bookmark not defined.**](#_Toc191661334)

[4. Резултати Error! Bookmark not defined.](#_Toc191661335)

[5. Заклучок 22](#_Toc191661336)

[6. Библиографија 23](#_Toc191661337)

# Вовед

Пандемијата на COVID-19 претставува пресвртна точка за ИТ-индустријата, особено во начинот на организација на работата. Моделот на работа на далечина стана широко прифатен, а многу компании усвоија хибриден пристап што комбинира работа од дома и работа од канцеларија. Овој трансфер отвори прашања поврзани со планирањето и користењето на канцелариските ресурси, координацијата на тимовите и обезбедувањето еднаков и транспарентен пристап до работните места и сали за состаноци.

Истовремено, трошоците за одржување на простор наменет за стотици вработени се значителни и често неоптимално искористени доколку поголемиот дел од неделата останува празен. Намалувањето на бројот на физички ресурси, доколку не е соодветно менаџирано, може да доведе до преоптоварување, нерамномерна распределба и незадоволство кај вработените. Потребен е систем кој ќе овозможи ефикасно резервирање, правично распределување и следење на искористеноста на канцелариските капацитети во хибридни услови.

Целта на овој труд е да предложи и имплементира софтверско решение за управување со канцелариските ресурси во хибриден модел на работа. Решението има за цел да обезбеди транспарентен процес на резервирање работни места и простории, да ја зголеми искористеноста на достапните ресурси преку правила и ограничувања дефинирани од компанијата и да им овозможи на вработените слободен избор на денови за работа од канцеларија без административен товар. На овој начин се постигнува рамнотежа меѓу флексибилноста на вработените и оптимизацијата на трошоците и капацитетите на организацијата.

# Користени технологии

Во рамки на развојот на веб-апликацијата за управување со канцелариски ресурси се користат современи технологии кои обезбедуваат стабилна, сигурна и скалабилна архитектура. Комбинацијата на избраните технологии овозможува високи перформанси и одржливост на backend инфраструктурата, конзистентен и пристапен кориснички интерфејс на frontend дел и сигурно управување со податоците во релациона база.

**2.1 Front-End**

**React 19**

**React** [1] претставува слободна и open-source библиотека со JavaScript основа, популарна и често користена за развој на мали апликации поради својата ефикасност (постигната со концептот на виртуален DOM), ре-употребливи компоненти и различни библиотеки и алатки кои овозможуваат побрзо тестирање и развој.

**TypeScript**

**TypeScript** [2], го надградува постоечкиот JavaScript со воведување на типизација, што овозможува побрза детекција на грешки, полесно дебагирање, подобра структура на кодот, полесна комуникација меѓу фронтенд и бекенд и поголема одржливост. Во контекстот на моделите и валидацијата, дефиницијата на типови значајно го намалуваат ризикот од неконзистентни модели.

**Material UI (MUI)**

**Material UI** [3] (MUI) претставува широко усвоена и препорачана библиотека, поддржана од реномирани платформи и организации, со долгорочен развоен циклус и јасна патека на одржување, што ја прави отпорна на застарување и високо доверлива од аспект на безбедност. Содржи сет на компоненти заснован на Material Design и овозможува брза изработка на доследен кориснички интерфејс (табели, формулари, пагинација, менија, известувања), како и лесно усогласување со визуелниот идентитет на апликацијата.

**i18next**

**i18next** [4] е библиотека која се користи за интернационализација на апликациите. Овозможува превод на апликацијата на повеќе јазици, притоа содржината на секој јазик може да се постави во еден посебен фајл, кој е лесно подложен на проширување, промени и одржување.

**2.2 Back-End**

**.NET 8 (ASP.NET Core)**

.**NET** [5] платформата на Microsoft е бесплатна алатка која нуди развој на различни хардверски и оперативни системи. Избрана поради еко-системот на вградени библиотеки, (кој овозможува побрза и поефикасна имплементација на инфраструктурата на кодот), како и безбедноста и сигурноста, (гарантирани со имплементирани подобренија за заштита на апликациите од малициозни закани).

**Entity Framework Core**

**Entity Framework** [6] e ORM мапиран framework креиран од Microsoft за развој на .NET апликации. Овозможува редуцирање на репетитивен код, мапирање на табелите од база во C# објекти и со интеграција на LINQ (Language Integrated Query) – испишување на query-ња во C# код – како и редуцирање на кодот за основните CRUD операции со објектите. Платформата овозможува автоматско следење на промени во ентитетите и истата е компатибилна на повеќе системи на бази на податоци.

**Hangfire**

**Hangfire** [7] е интегриран механизам за позадински задачи со перзистенција, распоредување и автоматски повторувања (retry). Се користи за периодични и одложени операции (во нашиот случај за ослободување истечени резервации), обезбедувајќи робустност и оперативна видливост преку dashboard.

**MediatR**

**MediatR** [8] е библиотека која ги спојува CQRS [9] (Command Query Responsibility Segregation – ги дели основните операции на два модели – еден за queries и еден за команди)(слика 1) и Mediator pattern [10] чија основа е комуникација помеѓу објектите со помош на објект – посредник. На овој начин се редуцира зависноста помеѓу објектите кои комуницираат.

A diagram of a software application

AI-generated content may be incorrect.

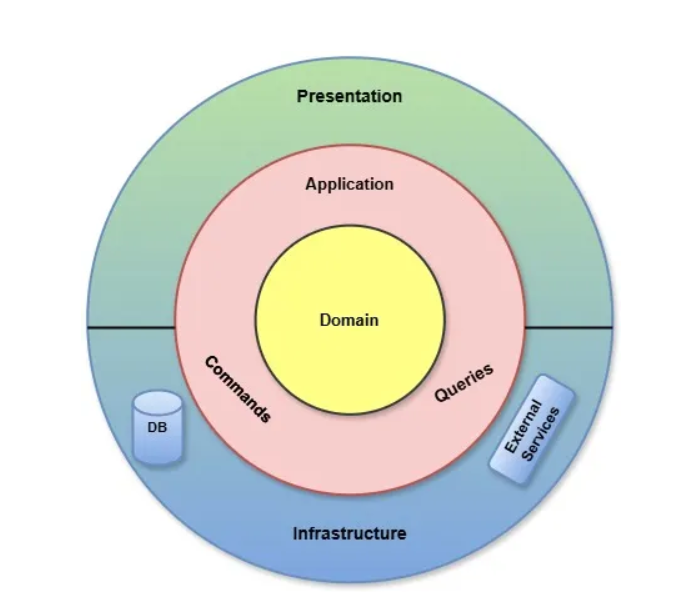
Слика 1 *Визуелизација на CQRS pattern*

**2.3 База на податоци**

**Microsoft SQL Server**

**Microsoft SQL Server** [11] е систем за менаџмент на релациони бази на податоци. Продукт на Microsoft еко-системот кој овозможува ефикасно и сигурно чување, промена и менаџирање со релациски податоци. Една од есенцијалните функции на Microsoft SQL Server е можноста да се интегрира со други системи и да овозможи пристап до своите податоци.

# Архитектура на решението

**** Апликацијата е изградена според принципите на **Clean Architecture** (слика 2) во комбинација со **CQRS**, што овозможува јасна поделба на одговорности, полесно тестирање и еволутивно проширување, додека **SOLID** принципите обезбедуваат код што е стабилен, лесен за одржување и безбедно проширлив. Слоевитоста обезбедува уреден проток на податоци и контрола, при што секој слој решава специфична задача без да ја наруши автономијата на другите.

Слика 2: *Визуелизација на CLEAN архитектура*

**Presentation**: влезот што ги прима барањата од UI и преку контролерите ги проследува кон останатите слоеви;

**Application**: ги спроведува бизнис правилата преку команди и барања (CQRS), координирајќи ја работата меѓу доменот и инфраструктурата;

**Domain**: јадрото на апликацијата кое е изолирано и независно од другите слоеви, ги содржи бизнис правилата и ентитетите;

**Infrastructure**: ја обезбедува перзистенцијата на податоците и интеграциите со други сервиси (EF Core/SQL Server, репозиториуми, Hangfire, надворешни сервиси);

**Contracts**: го дефинира типот на барања и одговори кои се содржани во endpoint-ите.

Овој распоред го одвојува читањето од запишувањето, го поедноставува кеширањето и трансакциите, и создава основа за скалабилен и одржлив систем.

## Серверска архитектура

A diagram of a command handling system

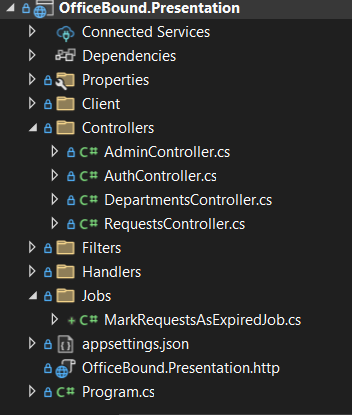
AI-generated content may be incorrect. CLEAN архитектурата се заснова на основниот принцип – Структурирање на кодот во форма на слоеви, каде што секој слој има своја специфична цел (Single Responsibility Principle or SRP).

Слика 3 *Визуелизација на проток на податоци со CLEAN + CQRS архитектура*

A screenshot of a computer

AI-generated content may be incorrect. За една акција од корисничката страна да се обработи на серверска страна и се добие повратна информација, истата изминува низ неколку слоеви.

Слика 4 *Визуелизација на архитектурата  
 на системот*

**3.1.1 Презентациски слој (Presentation Layer) (слика 5)**

Презентацискиот слој е точката од каде започнува интеракцијата со системот. Ги содржи контролерите (Controllers) како и Hangfire методите (Jobs) кои се одговорни циклусно извршување на операции. Контролерите ги користиме за примање на HTTP барања (request-и), нивна обработка (насочување кон апликацискиот слој) и враќање на соодветен одговор (response). Контролерите може да бидат од 4 типови: GET, POST, PUT, PATCH и DELETE.

Слика 5 *Приказ на контролерите и Презентацискиот слој*

**3.1.2 Апликациски слој (Application Layer) (слика 6)**

A screenshot of a computer

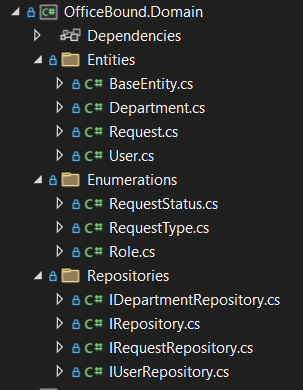
AI-generated content may be incorrect.

Слика 6 *Приказ на Апликациски слој со барања и команди*

Апликацискиот слој ја применува бизнис логиката на системот и е зависен од структурата на Доменскиот слој, чита команди и запишува операции од CQRS структурата, содржи мапирања на командните одговори, валидации, дефинира интерфејси и сервиси за посложени операции.

**3.1.3 Доменски слој (Domain Layer) (слика 7)**

Доменскиот слој е јадрото на една апликација - ја содржи бизнис логиката на системот и функционира изолирано од другите слоеви. Таа ги содржи ентитетите, енумерациите и сите објекти кои ја структурираат логиката на апликацијата.



Слика 7 *Приказ на Доменски слој*

**3.1.4 Contracts Layer**

Минимален слој кој само дефинира помошни објекти за трансфер на податоци (DTOs), објекти за испраќање и примање на податоци од endpoint-ите, исклучоци (Exceptions) и валидации.

A screenshot of a computer

AI-generated content may be incorrect.**3.1.5 Инфраструктурен слој (Infrastructure Layer) (слика 8)**

Ги користи интерфејсите од Апликацискиот слој, содржи интеграција со базата на податоци (Entity Framework) како и е место каде се имплементира интеграција со надворешни сервиси (доколку постои). Во овој слој се сместени миграциите и конфигурацијата со база на податоци, log-овите и зависи од Апликацискиот и доменскиот слој.

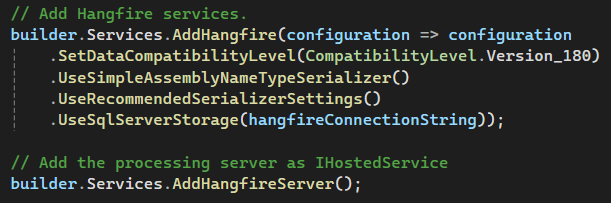
Слика 8 *Инфраструктурен слој со Repositories и Migrations*

## Интеграција со Hangfire

На серверска страна апликацијата користи интеграција со Hangfire рамката, open-source рамка за позадинско извршување на процеси, надвор од постоечкиот податочен тек (UnitOfWork). Нуди можност за повторувачки процеси регулирани со CRON изрази.

A yellow background with blue text

AI-generated content may be incorrect.За да може да се користи истата е потребна е инсталација на неколку библиотеки (слика 9).

За да го поставиме начинот на кој сакаме таа да функционира, ја конфигурираме во Program.cs фајлот (слика 10).

Слика 9 *Приказ на потребните Nugget пакети за инсталација на Hangfire framework-от*

Слика 10 *Приказ на конфигурација на Hangfire во Program.cs*

За апликации и акции кои се извршуваат во специфична временска рамка (пр. во нашата апликација поголем дел од операциите се извршуваат во работно време) добра пракса е овие позадински процеси да се извршуваат надвор од постоечката временска рамка (притоа да се земе во предвид и временската зона), со цел минимизирање на можноста од грешки (во нашиот случај надвор од работно време)(слика 11).

A computer screen shot of text

AI-generated content may be incorrect.

Слика 11 *Поставување на време за таргетирање на специфичниот процес*

Во нашиот систем функцијата која се извршува во позадина е промена на секое барање што е во статус “Се процесира” и датумот на важност му е постар од денешниот датум - да биде во статус “Истечено” (слика 12).

A screen shot of a computer program

AI-generated content may be incorrect.

Слика 12 *Илустрација на методот за маркирање на одминати барања како “истечени”*

## Опис на базата на податоци

Изгледот на базата на податоци е првиот чекор при планирање на одредена апликација. Од самата структура на податочниот модел во голема мера зависи брзината, одржливоста, скалабилноста и ефикасноста на апликацијата. За едно кориснично барање да се изврши брзо и со минимална шанса за грешки, се потребни добро дефинирани табели, релации, типови на податоци во базата.

Со помош на Entity Framework денес голем дел од овие дефиниции може да се изврши со code-first принцип, односно, прво се креираат класи (кои ќе станат ентитети во базата на податоци), додавање на соодветни анотации и валидатори на постоечките класи и потоа со креирање на миграции се моделира нашата база на податоци.

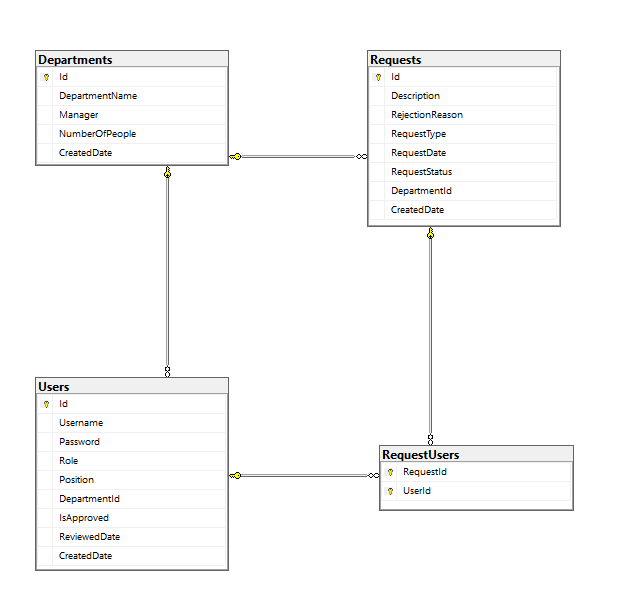
A screen shot of a computer program

AI-generated content may be incorrect.

Слика 13 *Пример за автоматски генерирана миграција за моделирање на база на податоци*

Миграциите се генерираат автоматски и изглед на една миграција е прикажан на сликата погоре (слика 13).

Предноста на SQL Server Management [12] базата е целиот Microsoft екосистем и интеграцијата со широка палета на алатки како PowerBI и SharePoint со што се олеснува работата со податоците од базата низ платформите, алатките за бизнис интелигенција, кои овозможуваат анализа на податоците за превземање бизнис решенија, како и можноста за интеграција со cloud (Microsoft Azure).

Како интегрирана околина за менаџирање со SQL инфраструктурата, Microsoft го има креирано SQL Server Management Studio. Со користење на SSMS може да се управуваат, конфигурираат, менаџираат, администрираат и креираат компоненти на SQL Server.

Слика 14 *Приказ на ER дијаграм од базата на податоци*

Дополнителна функција која ја нуди SSMS е и алатката за генерирање на релационен дијаграм (ER дијаграм) од постоечка база на податоци. Истата е искористена за креирање на дијаграмот кој ја рефлектира базата на податоци за нашата платформа и истиот е прикажан на слика 14.

За полесно тестирање и автоматизирање при проширување на функционалностите на системот, е додадена папка DevScripts – достапна за извршување само за Development и Testing околина. Со нивна помош може да се внесат иницијални корисници од секоја улога и може да се прескокне процесот на одобрување на корисник доколку се изврши соодветната скрипта.

Апликацијата OfficeBound е составена од 4 табели и тоа 3 главни табели и една меѓутабела, кои ги регулираат релациите помеѓу ентитетите Корисници, Барања и Оддели.

## Архитектура на Клиентска страна

На клиентска страна е користена type-based architecture (слика 15) (архитектура базирана на типови) каде кодот е групиран според типовите на податоци кои ги обработува. Препорачлива е за мали апликации бидејќи е лесна за имплементација доколку не се содржат многу компоненти, прифаќањето на бизнис логиката е интуитивно и лесно воочливо и со минималното вгнездување го се придонесува за да се намали когнитивното оптоварување.

* + 1. **api**

Во фолдерот **api** е сместен apiConnector-от кој е одговорен за поврзување со серверскиот дел, односно преку него ги испраќаме HTTP барања и добиваме соодветен одговор кој го прикажуваме назад кон корисникот.

* + 1. **assets**

Во фолдерот **assets** се содржат сите сите мултимедијални содржини кои ги користиме за подобрување на корисничкиот дизајн за нашиот систем

* + 1. **components**

A screenshot of a computer

AI-generated content may be incorrect.Во фолдерот **components** се сместени tsx компоненти (слика 14) – градбените единки на секоја реакт апликација. Оваа структура враќа HTML елементи кои се обработуваат со TypeScript код, а претставуваат мали единки кои се вметнуваат во една родителска компонента. Предноста на React технологијата е нејзиното користење на Virtual DOM (Virtual Document Object Model), или виртуелна репрезентација на корисничкиот интерфејс, што овозможува пре-лоадирање на одредена компонента – наместо целата страница.

Слика 15 *Приказ на структурата на компонентите*

* + 1. **contexts**

Во фолдерот **contexts** се содржани фајлови со кои ја пренесуваме состојбата на променливите низ компонентите, без да имаме потреба од мануелно подавање преку props.

* + 1. **models**

Во фолдерот **models** се содржани објектите кои ги користиме за мапирање, прикажување и комуникација со серверската страна и според функцијата што ја извршуваат, секоја од нив завршува со Dto – фајловите кои ги користиме за испраќање кон серверска страна и Response – фајлови во кои се мапирани повратните податоци од серверска страна.

* + 1. **router**

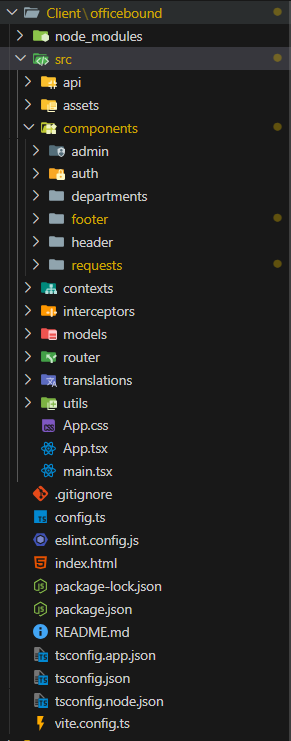
Во фолдерот **router** се содржи пренасочувањето на компонентите според акциите на корисникот.

* + 1. **translations**

Во фолдерот **translations** се содржи интеграцијата со библиотеката i18next и json фајлот во кои се содржи мапирање на текстовата содржина на статичките текстови од нашиот систем. Со имплементација на оваа библиотека е обезбедена основа за лесно имплементирање на достапност на апликацијата на повеќе јазици.

* + 1. **utils**

Во фолдерот **utils** се сместени константи, функции, променливи кои се користат во повеќе компоненти

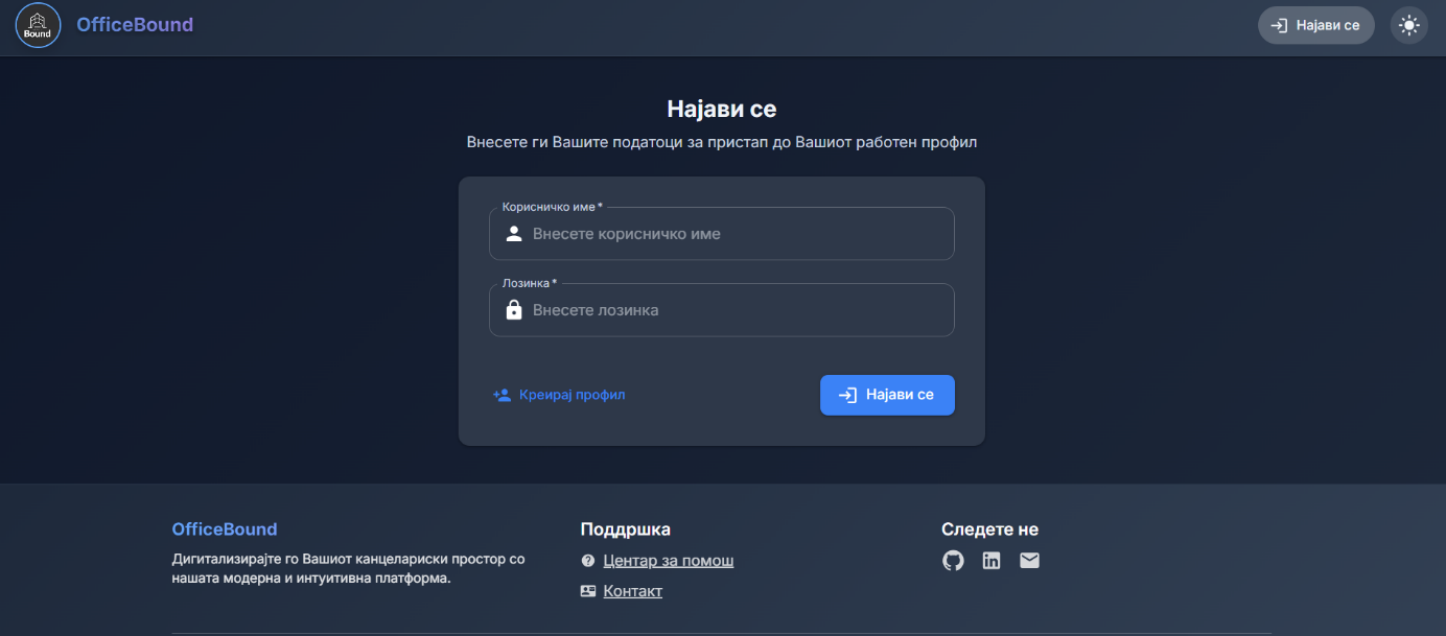


Слика 16 *Приказ на фолдерската структура на клиентска страна*

# Функционалности на системот и екрани

**4.1 Најава**

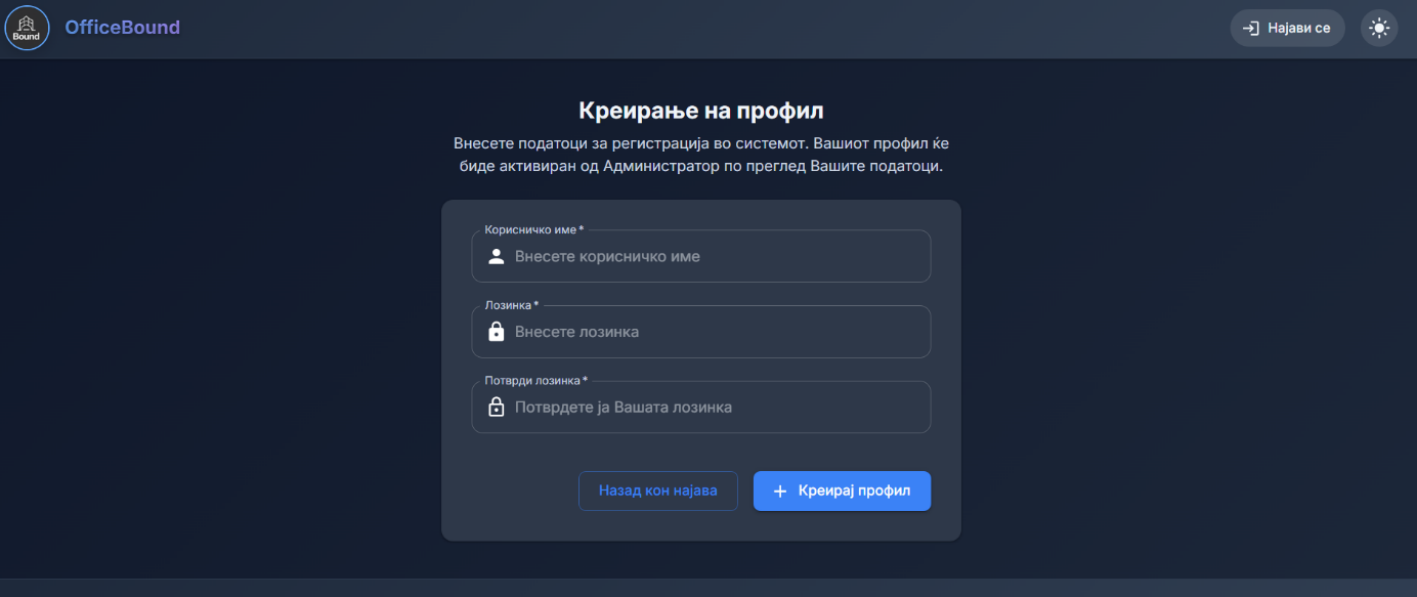
На екран 1 е прикажан првиот чекор кој е заеднички за сите 4 улоги.

Доколку корисникот нема креирано профил, на екранот е прикажано копче “Креирај профил” кое води до екранот за Креирање на нов профил. Бидејќи платформата претставува интерен систем за затворен круг на корисници (само вработените во една компанија) сите останати функционалности се заклучени за ненајавени корисници. Корисничкиот интерфејс се состои од два дизајни (светол и темен) и корисникот може да ги менува со клик на иконата во горниот десен агол.

Екран 1 *Најава на платформата*

**4.2 Креирање на профил**

Со пополнување на формата прикажана на Екран 2 корисникот е внесен во базата на податоци како профил за разгледување, и истиот може да биде одобрен за користење кога корисник со улога “Регионален Менаџер” ќе го потврди. Откога податоците ќе бидат потврдени, Регионалниот менаџер доделува улога на корисникот според неговата работна позиција, и според улогата, ќе му бидат достапни различните функционалности.

****

Екран 2 *Форма за креирање на профил*

**4.3 Креирање/Промена на барање за канцелариски ресурс**

A screenshot of a computer

AI-generated content may be incorrect. Со пополнување на формата на Екран 3 корисникот може да си резервира место во канцеларија. Поставена е валидација и ограничување, каде барањето може да се издаде за било кој датум, започнувајќи од утрешниот датум. Аналогно на тоа, може и да го промени своето барање доколку истото е во статус “Се процесира” или “Одобрено” а датумот не е помал од утрешниот датум.

Екран 3 Форма за креирање на ново барање од Вработен

Регуларниот вработен може да избере два типа на барања и тоа само Канцелариско биро, или канцелариско биро со паркинг. Вработениот со улога на Менаџер или Регионален менаџер, може да резервира и конференциска сала (екран 4). По пополнување на формата, барањето се испраќа до Менаџерот – за одобрување на канцелариски бироа и Регионалниот менаџер – за одобрување на конференциска сала.A screenshot of a computer

AI-generated content may be incorrect.

Екран 4 *Форма за креирање на барање од Корисник со улога Регионален Менаџер*

**4.4 Одобрување на барање за канцелариски ресурс**

A screenshot of a computer

AI-generated content may be incorrect.За да се одобри едно барање за ресурс, е потребно за датумот кога е избран тој ресурс да има слободни места. На екранот се прикажани и достапните места за ресурси од специфицираниот датум, а доколку не е наведен како основен се зема утрешниот датум.

Екран 5 *Листа на барања за одобрување од улога Регионален Менаџер*

Истиот се пресметува на начин, од вкупниот број на канцелариски бироа во конфигурација, се одзема вкупниот број на резервирани за тој ден. Доколку го одобриме барањето, а вкупниот број е надминат – се појавува валидација. На ист начин се пресметува и одобрување на тип ресурс со паркинг место или конференциска сала. За да може да се одобри едно барање, корисникот треба да има улога на Менаџер или Регионален Менаџер.

A screenshot of a computer

AI-generated content may be incorrect.

Екран 6 *Приказ на дијалог за откажување на постоечко барање*

Менаџерот има приказ на сите барања за неговиот оддел, додека регионалниот менаџер има приказ на сите креирани барања. Менаџерот може и да одбие барање, каде во Причина за откажување ја образложува својата одлука. Полето Причина за откажување на барање е задолжително, со поставена валидација од минимум 50 карактери. Регионалниот менаџер, има за задача и да ги гледа барањата за резервација на Конференциска сала од менаџерите, и според достапните места може да ги одобри или одбие барањата.

**4.5 Додавање на нов оддел**

A screenshot of a computer

AI-generated content may be incorrect.На екранот е прикажана формата за додавање нов оддел. Функционалноста за отворање на нов оддел може да ја пристапи само корисник со улога на Регионален Менаџер или Администратор.

Екран 7 *Форма за креирање на нов оддел*

A screenshot of a computer

AI-generated content may be incorrect.Одделот се креира на начин што се превземаат сите корисници со статус Вработен од базата на податоци и од постоечката листа на вработени селектираме еден со кој го пополнуваме полето Менаџер. На тој начин правиме промена во статусот на постоечкиот корисник. Во листата за избор на менаџери не се вклучени корисниците со улога Менаџер и Регионален Менаџер, бидејќи едно лице може да биде менаџер на само еден оддел.

Екран 8 *Форма за креирање на нов оддел со прикажан избор на Менаџер од листа на корисници*

**4.6 Промена на оддел**

A screenshot of a computer

AI-generated content may be incorrect. Промена на оддел може да изврши само Регионален Менаџер (или Администратор) и притоа истиот не може да го промени името на одделот – туку само бројот на вработени и неговиот менаџер. На екранот подолу е прикажана формата за промена на постоечки оддел.

Екран 9 *Екран за промена на оддел*

Доколку компанијата го има променето името на одделот, корисникот може да го отстрани постоечкиот и да креира нов оддел со ново име.

Кога претходен корисник кој бил со улога на Менаџер е променет од листата за селекција, корисникот се враќа на улога Вработен, доколку не биде отстранет од листата на корисници.

**4.7 Бришење/Отстранување на корисник**

Доколку корисникот престане да биде вработен во компанијата истиот може да се отстрани од Регионалниот Менаџер. Опцијата е достапна само за отстранување на регуларни корисници.

A screenshot of a computer

AI-generated content may be incorrect.

Екран 10 *Дијалог за потврда за отстранување на корисник*

A screenshot of a computer

AI-generated content may be incorrect.При избирање на опцијата за отстранување на вработен од листа на корисници се појавува дијалог каде Менаџерот наведува причина за отстранување на вработениот од системот.

Екран 11 *Приказ на дијалог за отстранување на корисник од системот со валидација*

За менаџер да биде отстранет од листа на корисници – прво треба да се промени одделот со кој раководел постоечкиот менаџер и да се назначи нов.

Доколку сакаме да ја промениме позицијата или да се отстрани од корисници Регионалниот Менаџер – функционалноста за промена или отстранување на Регионален Менаџер е достапна само за Администратор.

**4.6 Бришење/Отстранување на оддел**

Доколку постои промена во човечките ресурси на компанијата, постоечки оддел може да биде отстранет од Регионален Менаџер или Администратор. Притоа се појавува дијалог за конфирмација за отстранување и корисникот треба да наведе причина за истата која ќе содржи најмалку 50 карактери.

**4.7 Одобрување на нов корисник**

A screenshot of a computer

AI-generated content may be incorrect.Новите корисници можат да бидат одобрени или откажани од Регионален Менаџер или од Администратор. Откако Регионалниот Менаџер ќе ги прегледа и ќе утврди дека податоците се релевантни – тој го пополнува одделот во кој ќе припаѓа корисникот и неговата работна позиција и неговата улога во системот.

Екран 12 *Приказ на форма за одобрување на корисник*

A screenshot of a computer

AI-generated content may be incorrect.Улогите кои може да ги додели Регионалниот Менаџер се Вработен и Менаџер, а дополнително Администраторот може да ја додели и улогата на Регионален Менаџер, но притоа само еден корисник едновремено може да ја држи позицијата на Регионален Менаџер.

Екран 13 *Приказ на екран за одобрување на корисници*

**4.8 Филтрирање на барања**

A screenshot of a computer

AI-generated content may be incorrect.За полесен преглед на креираните барања за канцелариски ресурси се додадени филтри, кои го олеснуваат прегледот на истите. За да се олесни работата на менаџерите, барањата можат да се филтрираат по датум, оддел и статус.

Екран 14 *Приказ на екран за филтрирање на барања според датум*

A screenshot of a computer

AI-generated content may be incorrect. Дополнително, прикажана е листа со достапни канцелариски места, според која одобрувачот може да процени дали канцелариите имаат ресурси за да го опслужат постоечкото барање.

Екран 15 *Приказ на екран за филтрирање на барања според оддел*

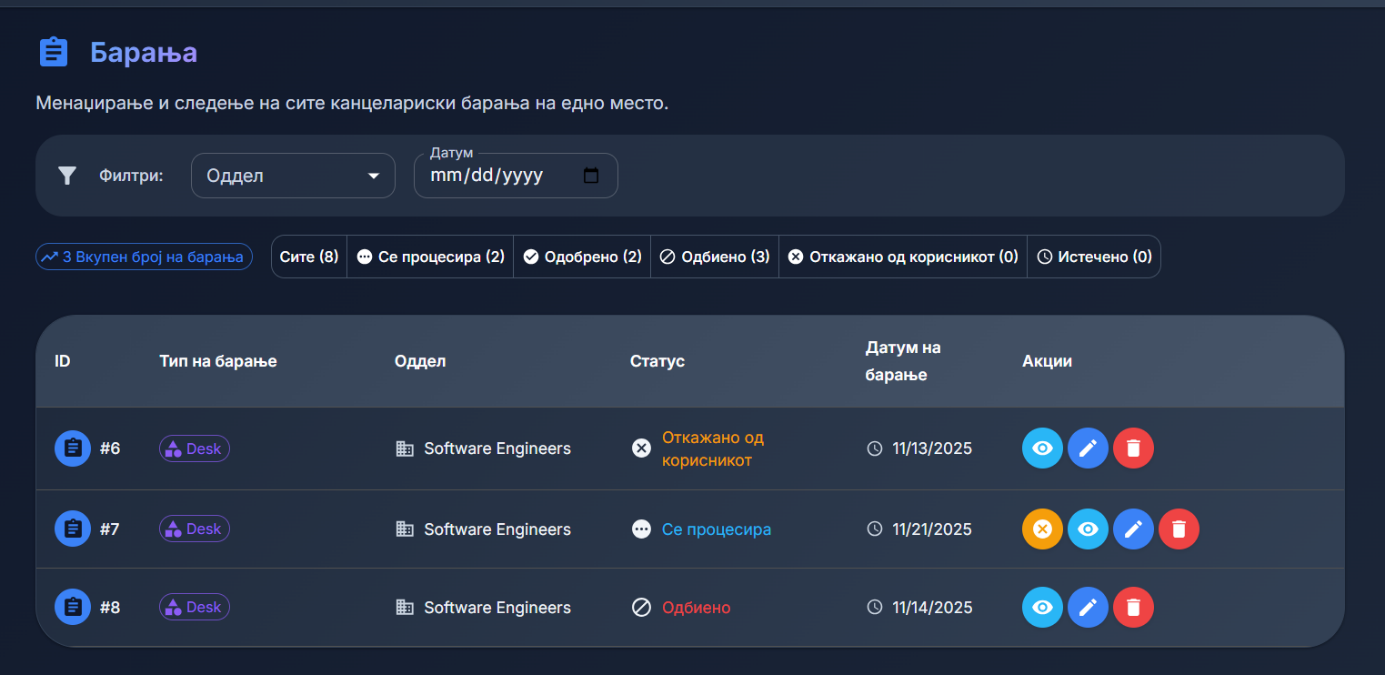
Вработениот може да ги користи постоечките филтри за преглед на своите барања според датум и нивниот статус, како што е прикажано на екранот.

**A screenshot of a computer

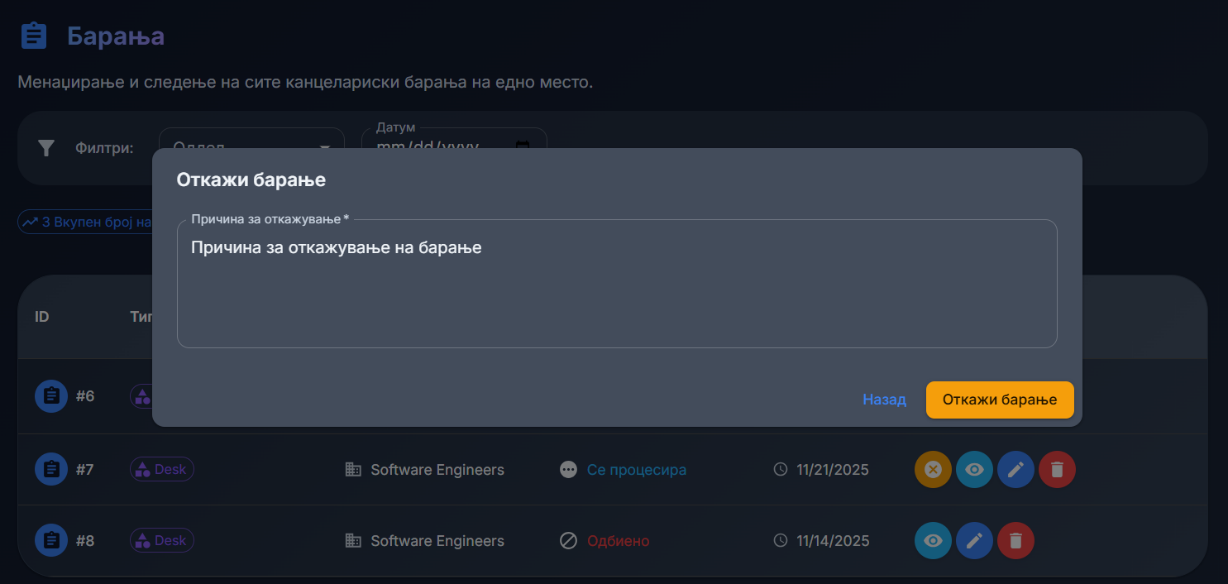
AI-generated content may be incorrect.**

Екран 16 *Приказ на резултати од филтер на корисник со улога Вработен*

**4.9 Откажување на барање од вработен**

Доколку корисникот се одлучи да го откаже барањето за канцелариски ресурс, барањето добива статус “Откажано од корисникот”. Притоа се отвора дијалог каде во причина за откажување вработениот треба да ја наведе својата со минимум 50 карактери.

Екран 17 *Приказ на екран со барања на корисник со улога Вработен*

**A screenshot of a computer

AI-generated content may be incorrect.**Оваа опција е достапна и видлива само за барања со валиден датум кои се во статус “Се процесира” или “Одобрено барање” и до истата има пермисии единствено корисникот кој го креирал барањето (екран 17).

Екран 18 *Приказ на форма за откажување на барање од корисник*

Истата, е достапна за преглед од Вработениот, неговиот Менаџер и Регионалниот Менаџер со опцијата “Преглед на барање”.

**4.10 Преглед на барање**

Екран 19 *Приказ на екран за преглед на барање*

A screenshot of a computer

AI-generated content may be incorrect. Доколку сака подетално да го прегледа своето барање, на сликата е прикажан екран каде корисник го прегледува своето барање. Корисник со улога на Вработен може да ги прегледува само своите барања, Менаџерот може да прави преглед на сите барања од неговиот оддел, додека Регионалниот Менаџер има пермисии да ги прегледува сите барања.

На екранот погоре е прикажана и причината за откажување, за барањата со статус “Откажано” и “Откажано од клиент”, како и корисничкото име на корисникот кој го креирал со што се овозможени анализи за најчестите причини за откажување на барањата и подобрување на условите за нивно редуцирање.

**A screenshot of a computer

AI-generated content may be incorrect.A screenshot of a computer

AI-generated content may be incorrect.4.11 Светол дизајн (Light Mode)**

Екран 21 *Приказ на екран со активирани валидации во светол дизајн*

Екран 20 *Приказ на екран со светол дизајн (light mode)*

# 

# Заклучок

Со платформата за резервација на место во канцеларија е овозможено планско регулирање на канцелариските ресурси. За вработените овозможува гарантиран пристап за работна атмосфера, додека за работодавците овозможува ефикасно користење на канцелариската инфраструктура и увид во користењето на истата. Со додадените филтри за пребарување по тим или корисник може да се утврди и потребата на секоја индивидуа од канцелариски простор, како и да се извршат бизнис анализи за споредба на продуктивноста помеѓу тимовите кои работат на далечина, хибридно или со лично присуство.

Со својот дизајн и технологии, апликацијата отвора простор за надоградување во повеќе правци. Првична идеја која би ја имплементирала во иднина, со цел апликацијата да добие поголема бизнис вредност би била додавање опција за генерирање на табели и графикони кои би ги илустрирале резултатите од постоечките филтри и би овозможиле поголема прегледност за бизнис анализи. Како второ, важно подобрување би издвоила функционалност за испраќање на меилови со нотификации за вработените, во моментот кога нивното барање е одобрено и ден пред резервацијата, за потсетување на дека претстои резервација. Дополнително, за зголемена безбедност при автентикација платформата би можела да се интегрира со Microsoft Outlook сервисите кои би се користеле како професионални меилови за најава и известувања од платформата.

Со едноставниот и модерен дизајн апликацијата е лесна, интуитивна и разбирлива за нови корисници и истата содржи јасни насоки и упатства кои придонесуваат за ефикасна интеракција помеѓу корисникот и системот.

# Библиографија

|  |  |
| --- | --- |
| [1] | <https://react.dev/learn> |
| [2] | <https://www.typescriptlang.org/docs/> |
| [3] | <https://mui.com/material-ui/> |
| [4] | <https://react.i18next.com/> |
| [5] | <https://learn.microsoft.com/en-us/dotnet/> |
| [6] | <https://learn.microsoft.com/en-us/ef/core/> |
| [7] | [https://docs.hangfire.io/en/latest/](https://mediatr.io/) |
| [8] | <https://mediatr.io/> |
| [9] | <https://learn.microsoft.com/en-us/azure/architecture/patterns/cqrs> |
| [10] | Book - *“Dive Into Design Patterns”* byAlexander Shvets |
| [11] | <https://learn.microsoft.com/en-us/sql/> |
| [12] | <https://learn.microsoft.com/en-us/ssms/sql-server-management-studio-ssms> |