

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

Національний аерокосмічний університет ім. М. Є. Жуковського
«Харківський авіаційний інститут»

Факультет ракетно – космічної техніки

Кафедра вищої математики та системного аналізу

Пояснювальна записка до дипломної роботи

магістра

(освітньо-кваліфікаційний рівень)

на тему **«Автоматизована система бронювання площ жилого та нежилого фондів»**

ХАІ.405.463-м.124. 1504012.200

Виконав: студент 6 курсу групи № 463-м
спеціальності 124 «Системний аналіз»
освітньої програми «Системний аналіз і
управління»

(шифр і назва напрямку підготовки (спеціальності))

Дейко О.В.

(прізвище й ініціали студента)

Керівник: Макарічев В.О.

(прізвище й ініціали)

Рецензент: Колодяжний В.М.

(прізвище й ініціали)

Міністерство освіти і науки України
Національний аерокосмічний університет ім. М. Є. Жуковського
«Харківський авіаційний інститут»

Факультет Факультет ракетно – космічної техніки
(повне найменування)

Кафедра вищої математики та системного аналізу
(повне найменування)

Рівень вищої освіти другий (магістрський)

Спеціальність 124 «Системний аналіз»
(код та найменування)

Освітня програма «Системний аналіз і управління»

ЗАТВЕРДЖУЮ
Завідувач кафедри

_____ Ніколаєв О.Г.
(підпис) (ініціали та прізвище)
 «__» _____ 20__ р.

З А В Д А Н Н Я
НА ДИПЛОМНУ РОБОТУ СТУДЕНТА

Дейку Олександр Володимировичу
(Прізвище, ім'я, по батькові)

1. Тема роботи Автоматизована система бронювання площ жилого та нежилого фондів

керівник роботи к. ф.-м. наук, доцент Макарічев Віктор Олександрович
(Прізвище, ім'я, по батькові, науковий степінь, Вчене звання)

затверджені наказом вищого навчального закладу № 1819-уч від 03.11.2020 р.

2. Термін подання студентом роботи «__» _____ року

3. Вихідні дані до роботи мобільний додаток

4. Зміст пояснювальної записки (перелік завдань, які потрібно розв'язати):

1. Опис автоматизованої системи з точки зору системного аналізу.

2. Модельний вигляд об'єкта дослідження.

3. Вибір платформи та середовища розробки програмного продукту.

4. Розробка програмного продукту.

5. Тестування та виправлення помилок у функціоналі програмного продукту.

6. Визначення розрахунку собівартості та ціни програмного продукту.

5. Перелік графічного матеріалу (з точним зазначеним обов'язкових креслень)

6. Консультанти розділів роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
Загальна частина			
Економічна частина			

Нормоконтроль _____ « _____ » _____ 20__ р.
(підпис) (ініціали та прізвище)

7. Дата видачі завдання

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів дипломного проекту (роботи)	Строк виконання етапів проекту (роботи)	Примітка
1	Огляд апріорної інформації по заданій тематиці		
2	Визначення актуальності проблеми, цілей, завдань и методів дослідження		
3	Опис автоматизованої системи для розподілу завдань з точки зору системного аналізу		
4	Модельний вигляд об'єкта дослідження		
5	Вибір платформи та середовища розробки програмного продукту		
6	Розробка програмного продукту		
7	Тестування та виправлення помилок у функціоналі програмного продукту		
8	Визначення розрахунку собівартості та ціни програмного продукту		
9	Оформлення пояснювальної записки		

Студент

_____ (Підпис)

_____ (прізвище та ініціали)

Керівник проекту (роботи)

_____ (Підпис)

_____ (прізвище та ініціали)

РЕФЕРАТ

Пояснювальна записка до дипломної роботи містить в собі: 69 с., 18 рис., 19 табл., 19 джерел.

АВТОМАТИЗОВАНА СИСТЕМА, МОБІЛЬНИЙ ДОДАТОК БРОНЮВАННЯ ПЛОЩ, МОВА C#, ПРОГРАМНИЙ ПРОДУКТ, СИСТЕМНИЙ АНАЛІЗ, ШАБЛОН MVC.

Об'єктом дослідження являється автоматизована система бронювання площ жилого та нежилого фондів.

Мета роботи: можлива розробка специфікації автоматизованої системи бронювання площ жилого та нежилого фондів.

Методи досліджень: методи системного аналізу, теорії алгоритмів та структури даних.

Задачі дослідження:

- Розробка методики до структурування вхідних даних, специфікація дій над ними і швидка відповідь системі для виконання певних дій та операцій;
- Специфікація структури вихідних даних для загального розуміння клієнту в простій формі;

Доказується системність об'єкта дослідження, наводиться структурний, функціональний, інформаційний, класифікаційний опис системи. Визначаються основні вхідні й вихідні величини. Розглядаються основні методи автоматизованої системи бронювання площ жилого та нежилого фондів.

Показано та проаналізовано хід роботи програмного продукту, зроблено висновки щодо автоматичного вибору приміщення клієнту для бронювання.

Розраховано повну вартість на створення мобільного додатку для дипломної роботи з економічної точки зору.

В ході роботи зроблено висновок, що сучасний простий і зрозумілий новому користувачеві додаток був би незамінним на ринку праці, розроблено відповідний програмний продукт.

РЕФЕРАТ

Пояснительная записка к дипломной работе включает в себя: 69 с., 18 рис., 19 табл., 19 источников.

АВТОМАТИЗИРОВАННАЯ СИСТЕМА, МОБИЛЬНОЕ ПРИЛОЖЕНИЕ БРОНИРОВАНИЯ ПЛОЩАДЕЙ, СИСТЕМНЫЙ АНАЛИЗ, ЯЗЫК C#.

Объектом исследования является автоматизированная система бронирования площадей жилого и нежилого фондов.

Цель работы: возможна разработка спецификации автоматизированной системы бронирования площадей жилого и нежилого фондов.

Методы исследований: методы системного анализа, теории алгоритмов и структуры данных.

Основные задачи:

- Разработка методики к структурированию входных данных, спецификация действий над ними и быстрый ответ системе для выполнения определенных действий и операций;
- Спецификация структуры исходных данных для общего понимания клиенту в простой форме.

Доказывается системность объекта исследования, приводится структурный, функциональный, информационный, классификационный описание системы. Определяются основные входные и выходные величины. Рассматриваются основные методы автоматизированной системы бронирования площадей жилого и нежилого фондов.

Показан и проанализирован ход работы программного продукта, сделаны выводы относительно автоматического выбора помещения клиенту.

Рассчитано полную стоимость на создание мобильного приложения для дипломной работы с экономической точки зрения.

В ходе работе сделан вывод, что современное простое и понятное новому пользователю приложение было бы незаменимым на рынке труда, разработан соответствующий программный продукт.

ABSTRACT

Explanatory note to the source contains: 69 p., 18 draw., 19 tabl., 21 sources.

AUTOMATED SYSTEM, LANGUAGE C#, MOBILE APPLICATION FOR RESERVATION PREMISES, PATTERN MVC, SYSTEM ANALYSIS.

The object of the research is an automated reservation system for residential and non-residential premises.

Purpose of work: possible to develop a specification for an automated reservation system for residential and non-residential premises.

Research methods: methods of system analysis, theory of algorithms and data structure.

Main tasks:

- Development of a methodology for structuring input data, specifying actions on them and a quick response to the system to perform certain actions and operations;
- Specification of the structure of the initial data for a common understanding by the client in a simple form.

The systemicity of the research object is proved, structural, functional, information, classification description of the system is provided. The main input and output values are determined. The main methods of an automated reservation system for residential and non-residential premises are considered.

The progress of the software product is shown and analyzed, conclusions are drawn regarding the automatic selection of premises for the client for booking.

The total cost of creating a mobile application for thesis from the economic point of view is calculated.

In the course of the work, it was concluded that a modern, simple and user-friendly application would be indispensable in the labor market, developed a software product.

ЗМІСТ

Вступ.....	10
1. Автоматизована система бронювання площ житлових та нежитлових фондів	13
1.1. Визначення та основні поняття автоматизованої системи бронювання площ жилого та нежилого фондів	13
1.2. Огляд досліджень, публікацій та електронних джерел.....	16
1.3. Цілі та завдання дослідження	19
2. Опис автоматизованої системи бронювання площ жилого та нежилого фондів з точки зору системного аналізу	21
2.1. Структурний опис системи.....	21
2.2. Функціональний опис системи	24
2.3. Інформаційний опис системи	28
2.4. Класифікаційний опис системи	31
3. Модельне представлення об'єкта дослідження і його характеристики.....	33
3.1. Вхідні й вихідні величини	33
3.2. Дослідження області допустимих значень вхідних величин	33
3.3. Прогнозування області значень вихідних величин	34
3.4. Можливі методи рішення модельного представлення	35
4. Технічна частина.....	37
4.1. Вибір середі розробки програмного продукту.....	37
4.2. Шаблон MVC.....	42
4.3. Процес роботи мобільного додатку.....	43
4.4. База даних для мобільного додатку.....	50

	8
Висновок по четвертому розділу.....	52
5. Економічна частина.....	53
5.1. Опис програмного продукту.....	53
5.2. Визначення трудовитрат робіт	53
5.3. Перелік робіт для створення програмного продукту	54
5.4. Склад виконавців роботи та розрахунок заробітної плати	54
5.5. Перелік необхідного обладнання для створення ПП	57
5.6. Розрахунок собівартості та повної вартості програмного продукту ...	62
Економічний висновок.....	64
Висновки.....	65
Перелік джерел посилання:	68

СКОРОЧЕННЯ ТА УМОВНІ ПОЗНАЧЕННЯ

ПЗ – Програмне забезпечення

ПП – Програмний продукт

API – Інтерфейс додатку

MVC – архітектура сервера для додатку

UI – Інтерфейс користувача

ВСТУП

У сучасному діловому середовищі приміщення для проведення зустрічей стають вкрай затребуваними. Але пошук і бронювання вільного приміщення може займати багато часу. Даний процес можна спростити завдяки UI-панелями для бронювання переговорних кімнат та інтерфейсу візуальної навігації. Безпосереднє підключення до популярних календарних сервісів. А майбутній інтерфейс буде забезпечувати централізований перегляд статусу приміщень і їх місцезнаходження в реальному часі.

Користувачі можуть забронювати приміщення безпосередньо за допомогою всього декількох натискань на самій панелі для бронювання зі свого мобільного пристрою. Доступний широкий вибір монтажних опцій для фільтрації площ, включаючи широку можливість отримати заброньоване приміщення, не привертає великої уваги для витрати часу для отримання бажаного результату завдяки додатку.

Проглянувши багато схожих додатків для бронювання приміщень, зробив висновок, що вони не відповідають повністю даній тематиці, в собі мають різноманітну рекламу в великій кількості, що ускладнює користування цими додатками, а також відсутність автоматизованої системи, яка повністю по своїм критеріям рекомендує найближчу вільну кімнату, яка знаходиться на одному з поверхів, що задовольняє клієнта і в найбільш зручний час. В зв'язку з цим дана розробка подібної системи є актуальною.

Рішення поставлених завдань і шляхи досягнення мети представлені в наступних розділах поточної роботи:

У першому розділі даної дипломної роботи приведено огляд інформаційних джерел. Також в цьому розділі представлена основна проблематика автоматизованої системи бронювання площ жилого та нежилого фондів у вигляді дерева проблем. Сюди входять цілі у вигляді дерева цілей, які складені на основі проблематики даної тематики. На основі розглянутих інформаційних джерел

сформовано об'єкт, предмет, ціль, задача та методи дослідження:

Об'єкт дослідження – Автоматизована система бронювання площ жилого та нежилого фондів.

Предмет дослідження – структура автоматизованої системи для бронювання площ жилого та нежилого фондів.

Мета дослідження – можлива розробка специфікації автоматизованої системи для бронювання площ жилого та нежилого фондів.

Задачі дослідження:

- Розробка методики до структурування вхідних даних, специфікація дій над ними і швидка відповідь системі для виконання певних дій та операцій;
- Специфікація структури вихідних даних для загального розуміння клієнту в простій формі;

Методи дослідження:

- Методи системного аналізу;
- Методи теорії алгоритмів та структури даних;

У розділі під номером два наведені такі описи системи з точки зору системного аналізу:

- Структурний;
- Функціональний;
- Інформаційний;
- Класифікаційний.

В третьому розділі описано модельне уявлення об'єкта та його характеристики, куди входить опис проектування для створення додатка, опис структури та рівнів додатка. Також в цьому розділі були розглянуті вхідні і вихідні величини досліджуваної системи, було описано процес роботи автоматизованої системи для розподілу задач, дослідження з вибору необхідних алгоритмів та програмних рішень, за допомогою яких був створений клієнт-серверний додаток. Вказані деякі можливості оптимізації системи.

У четвертому розділі обґрунтований вибір середовища розробки про-

грамного забезпечення, були представлені результати роботи додатку та продемонстровані основні можливості програми.

У п'ятому розділі проводиться розрахунок економічної складової програмного продукту.

У висновку наведено опис рекомендацій, що були дотримані, та шляхи рішення поставленої задачі. А також описана практична значущість програми для поточної роботи.

1. АВТОМАТИЗОВАНА СИСТЕМА БРОНЮВАННЯ ПЛОЩ ЖИЛОГО ТА НЕЖИЛОГО ФОНДІВ

1.1. Визначення та основні поняття автоматизованої системи бронювання площ жилого та нежилого фондів

В період розробки програмного продукту та його визначення можемо виділити декілька основних понять.

Автоматизована система – сукупність керованого об'єкта й автоматичних керуючих пристроїв здійснює збирання інформації з об'єкта керування, передає, перетворює й обробляє її, формує керуючі команди та виконує їх на керованому об'єкті, тобто ті функції, які піддаються автоматизації [1].

Житловий та нежитловий фонд – сукупність всіх приміщень незалежно від форм власності, включаючи будинки, спеціалізовані будинки (гуртожитки, готелі-притулки, будинки маневреного фонду, житлові приміщення з фондів житла для тимчасового поселення змушених переселенців та осіб, визнаних біженцями, житлові приміщення з фонду житла для тимчасового поселення громадян, які втратили житло в результаті звернення стягнення на житлове приміщення, яке придбане за рахунок кредиту банку або іншої кредитної організації яких засобів цільової позики, наданого юридичною особою на придбання житла, і закладено у забезпечення повернення кредиту або цільової позики, спеціальні будинки для самотніх людей похилого віку, будинки-інтернати для інвалідів, ветеранів та інші), квартири, службові житлові приміщення, інші житлові приміщення в інших будівлях.

Десктопний додаток Outlook – поштовий клієнт, розроблений Microsoft для роботи з електронною поштою, він також є повноцінним органайзером, що надає функції календаря, планувальника завдань, записника і менеджера контактів для зв'язку робітників одного підприємства. Але зараз є функціонал для бронювання приміщення, що не є зручним процесом, більшість користувачів не

використовують функціонал для бронювання. Тому є потреба в розробці даного мобільного додатку.

Бронювання приміщень в сучасному діловому світі являються важливою частиною в сферах, які мають серйозні зустрічі, конференції чи аренди для робітників, замовників чи клієнтами.

Велика потреба такої автоматизованої системи являється в зменшенні використаного часу, який завжди є головною частиною в нашому житті. При бронюванні не потрібно використовувати десктопний пристрій, що обмежує виконати даний процес в будь-якому місці з доступом до інтернету та прив'язує користувача до певного місця, що не є дуже зручно.

Цілком очевидно, що для того, щоб заздалегідь забронювати бажане приміщення, достатньо буде використати мобільний додаток, в якому всього за декілька натискань клієнт отримає відповідь від програми з успішним завершенням отримання приміщення під свої потреби.

З кожним днем статистика гаджетів проти десктопів показує, що люди в сучасному світі з кожним днем все більше використовують смартфони і все менше – робочі станції, тому що суспільство завжди знаходиться в рухові, в якому б куточку світу не знаходився, а це є набагато практичніше вирішення всіх своїх особистих проблем. Такий підхід повинен зменшити затрати часу на виконання своїх дій [2,с.308].

В даний час існують кілька технологій для створення програмного продукту, що надають можливість розробити клієнт-серверний додаток (Visual Studio 2019 на мові C# (CSharp) з використанням MVC, Microsoft SQL Express 2014).

C# з використанням MVC – середа для розробки програмних продуктів, надає можливість розробляти клієнт-серверні додатки, написання сайтів, якими користуються весь світ з кожним днем все більше, переходячи від десктопних додатків, на які, з часом буде все менша потрібність.

Також, варто звернути увагу на ціну ліцензування даного програмного забезпечення. На даний час доступні три версії:

1. навчальна (з обмеженнями в основному функціоналі і кількістю використовуваних блоків)
2. університетська (для використання в навчальних закладах для проведення дослідницьких експериментів)
3. професійна (для комерційного використання в компаніях і фірмах)

Вартість ліцензії для одного користувача професійної версії становить немалих коштів, що можуть дозволити собі великі компанії з багатотисячним оборотом, але не власники малого та середнього бізнесу. Таким чином, в даний момент вкрай актуальним є питання розробки програмного забезпечення, яке зможуть собі дозволити власники малого та середнього бізнесу.

Microsoft SQL Express 2014 – система управління реляційними базами даних (РСУБД), розроблена корпорацією Microsoft. Являється безплатною версією з деякими обмеженнями. Використовується для роботи з базами даних розміром від персональних до великих баз даних масштабу підприємства, конкурує з іншими СУБД в цьому сегменті ринку[3].

Mongo Db – документо-орієнтована система керування базами даних (СКБД) з відкритим вихідним кодом, яка не потребує опису схеми таблиць. MongoDB займає нішу між швидкими і масштабованими системами, що оперують даними у форматі ключ/значення, і реляційними СКБД, функціональними і зручними у формуванні запитів. MongoDB підтримує зберігання документів в JSON-подібному форматі, має досить гнучку мову для формування запитів, може створювати індекси для різних збережених атрибутів, ефективно забезпечує зберігання великих бінарних об'єктів, підтримує журналювання операцій зі зміни і додавання даних в БД, може працювати відповідно до парадигми Map/Reduce, підтримує реплікацію і побудову відмовостійких конфігурацій. У MongoDB є вбудовані засоби із забезпечення шардінгу (розподіл набору даних по серверах на основі певного ключа), комбінуючи який з реплікацією даних можна побудувати горизонтально масштабований кластер зберігання, в якому відсутня єдина точка відмови (збій будь-якого вузла не позначається на роботі БД), підтримується автоматичне відновлення після збою і перенесення наван-

таження з вузла, який вийшов з ладу. Розширення кластера або перетворення одного сервера на кластер проводиться без зупинки роботи БД простим додаванням нових машин[4].

По-друге, існує велика проблема з навчальними матеріалами по роботі з даними програмними пакетами. Це пов'язано, в основному, з вузькою спрямованістю додатків. Більшість існуючих джерел інформації та прикладів не доступні для розуміння користувачів без відповідної кваліфікації. Незважаючи на те, що компанії активно проводять семінари і постачають масою інструкцій та настанов, ймовірність того, що користувач, який зіткнувся з проблемою, відшукає її рішення в мережі вкрай мала. Це відбувається через величезну кількість можливостей додатків і, в той же час, недостатню кількість навчальних матеріалів. Як наслідок, користувач не може швидко зорієнтуватися і вирішити проблему, знайти її рішення на сайті виробника або призначених для користувача форумах.

1.2. Огляд досліджень, публікацій та електронних джерел

В століття автоматизації деякі технічні операції як і раніше виконуються вручну, відбираючи настільки дорогоцінний час. Однією з таких операцій є бронювання приміщень.

Наприклад, робітник великої компанії має проводити безліч ділових зустрічей, співбесід, конференцій. В широкому доступі для бронювання кімнати в різних схожих сферах праці використовують програму Outlook, про яку описано в попередньому розділі. Йому потрібно забронювати приміщення доки є доступ до воркстанції, а якщо це незапланована зустріч і з хвилини на хвилину потрібно до неї приєднатися, то нема можливості забронювати кімнату доки знаходишся в процесі поїздки до офісу, що негативно позначається на ефективності підприємства.

Наприклад, маючи розроблений мобільний додаток, що простий в користуванні та не має зайвих функцій, які не стосуються процесу броні, клієнт має

можливість заздалегідь забронювати приміщення для особистих потреб в процесі поїздки чи перельоту до іншої країни, знаючи, що житлове місце призначене йому. Такий процес має мінімізувати час на пошук та бронювання приміщень в різних сферах, які орудуються великою кількістю фондів з різнорідними характеристиками[5].

Тому постають такі актуальні проблеми:

- заявки клієнтів довго обробляються в системі;
- тривале очікування відповіді на бронь;
- тривалий пошук приміщення;

В інтернет-джерелах було знайдено декілька програм, які частково відповідають поставленій задачі у роботі. Один з таких найближчих додатків до цілі являється «Extron» – пошук і бронювання переговорних кімнат. Вона містить такі ж функції, що задовольняють наші цілі, але додаток не має автоматичної системи для бронювання носерних фондів, що являється одним із важливих процесів.

В джерелах [7,8] проблемами розробленої системи розподілу завдань являються:

- відсутність програми в вільному доступі;
- робітник пристойно витрачає часу на пошук;
- можлива дисконунікація в команді;
- перегляд статусу приміщення потребує окремий тип календаря;

На основі розглянутих інформаційних джерел і описаних в них проблем на рисунку 1.1, представлено дерево проблем.

В результаті аналізу проблем в системі розподілу завдань, зробимо висновок і представимо дерево цілей (рисунок 1.2).



Рисунок 1.1– Дерево проблем



Рисунок 1.2 – Дерево цілей

З вище представлених дерев проблем та цілей логічно буде перефразувати ці поняття на більш зручні задачі в сфері системного аналізу та програмування для реалізації нашої програми.

Такі проблеми й цілі як:

- кругообіг посилянь запитів на сервер та швидкість відповідей клієнтам;
- економія відведеного часу на пошук;
- завантаженість серверів не призначених для запланованих дій;
- обробка запитів;
- відповіді клієнтам;

потрібно віднести до однієї задачі досліджування – розробка методики до структурування вхідних даних і специфікація дій над ними. Тому що всі ці проблеми залежать від того:

- як швидко автоматизована система впорається з визначенням вхідних даних;
- оптимально підбере вільну площу клієнту по відповідним критеріям зі зручним для нього використанням.

Саме словосполучення «автоматизована система» дає зрозуміти, що трата часу на пошук є мінімальною. Чим краще оптимізована система, тим швидше здійснюється кругообіг бронювань кімнат, що приносить більший прибуток підприємству, і звісно є сенс розширення.

В даній роботі для дослідження можливості розробки автоматизованої системі були вибрані такі проблеми й цілі як:

- реакція клієнтів на надання послуг
- використання робітниками відведеного часу.

1.3. Цілі та завдання дослідження

Проведений огляд і аналіз наявної інформації дозволив сформулювати ціль, об'єкт дослідження, предмет дослідження, задачі дослідження та методи дослідження.

Об'єкт дослідження – Автоматизована система бронювання площ жилого та нежилото фондів.

Предмет дослідження – структура автоматизованої системи для бронювання площ жилого та нежилото фондів.

Мета дослідження – можлива розробка специфікації автоматизованої системи для бронювання площ жилого та нежилото фондів.

Задачі дослідження:

- Розробка методики до структурування вхідних даних, специфікація дій над ними і швидка відповідь системі для виконання певних дій та операцій;
- Специфікація структури вихідних даних для загального розуміння клієнту в простій формі;

Методи дослідження:

- Методи системного аналізу;
- Методи теорії алгоритмів та структури даних;

2. ОПИС АВТОМАТИЗОВАНОЇ СИСТЕМИ БРОНЮВАННЯ ПЛОЩ ЖИЛОГО ТА НЕЖИЛОГО ФОНДІВ З ТОЧКИ ЗОРУ СИСТЕМНОГО АНАЛІЗУ

2.1. Структурний опис системи

По результатам аналізу інформаційних джерел було наведено структурний опис автоматизованої системи.

Принцип роботи автоматизованої системи:

Для входу в програму користувачеві можливо потребується авторизуватися в системі даної компанії, якщо дані дійсні – власник матиме змогу здійснити пошук найближчої зручнішої кімнати та забронювати її за допомогою автоматизації. Спочатку клієнт вибирає необхідні йому вхідні параметри пошуку. Далі ці дані надходять в серверну частину для обробки певними методами такі як:

- Отримання об'єктів по геокоординат гаджету та вхідних параметрів;
- Фільтрація об'єктів;
- Сортування об'єктів по різним характеристикам;
- Повернення результатів на мобільний додаток в виді списку;

Після бронювання приміщення, даний об'єкт має такі статуси:

- Доступний до певного часу;
- Зайнятий до певного часу;
- Вільний;

Також при переході в календар певної кімнати є можливість перегляду всіх вільних та заброньованих часових рамок для номерного фонду в будь-який день.

Є можливість відмінити бронь і відразу ж зайняті часові рамки стануть доступні іншим охочим співробітникам.

Розроблено схему системи:

1. Автоматизована система бронювання площ жилого та нежилого фондів

1.1. Вхідні параметри для пошуку приміщення

1.1.1. Вибір локації

1.1.2. Місткість приміщення

1.1.3. Вибір додаткового обладнання

1.1.4. Вибір часу для бронювання

1.2. Алгоритм обробки параметрів

1.2.1. Алгоритм місцезнаходження клієнта

1.2.2. Алгоритм отримання об'єктів

1.2.3. Алгоритм фільтрації об'єктів

1.2.4. Алгоритм сортування об'єктів

1.3. Вихідні параметри

1.3.1. Список бажаних об'єктів

1.3.1.1. Бронювання обраного об'єкту

На рисунку 2.1 представлена структурна схема автоматизованої системи бронювання площ.



Рисунок 2.1 – Структурна схема автоматизованої системи

Для того щоб визначити, чи є наш об'єкт дослідження системою, необхідно показати, якими властивостями, типами елементного складу, елементів, зв'язків, структури він володіє.

Властивості системи:

1) Емерджентність

Вхідні дані потрібні для аналізу вибору потрібних об'єктів.

Структуризація – опрацьовує дані в відповідному форматі.

Оброблені дані потрібні для отримання бажаного результату що задовольняє користувача.

2) Цілісність

Якщо не подати вхідних даних, то не буде й оброблених вихідних даних.

3) Адитивність

Присутня можливість бронювати приміщення необмежену кількість разів

4) Синергізм

Система не володіє такою властивістю.

5) Прогресуюча систематизація

Характеризується тим, що всі етапи залежать один від одного, так як отримати оптимальний результат можливо, тільки якщо об'єднати всі етапи системи.

6) Ізоморфізм

Дані які подаються в якості вихідних об'єктів схожі один на одного, кожна сутність має ідентичну структуру, але різні значення(назва об'єкту, країни, міста, вулиці тощо).

методи фільтрації та сортування застосовуються до всіх об'єктів.

Тип елементного складу – змішаний.

Тип елементів – інформаційний.

Тип зв'язків між елементами – інформаційні.

Тип структури – ієрархічна.

2.2. Функціональний опис системи:

Функціональний опис системи проводиться для того, щоб визначити важливість системи, її ролі при рішенні поставленої задачі, а також відношення з другими системами.

Дана система є багатофункціональною.

Функціональний опис системи відноситься до структури, представлений на сторінці 23.

Функції системи першого рівня представлені в таблиці 2.1.

Таблиця 2.1 – Функції системи першого рівня

Код	Елемент	Функція
1	Автоматизована система бронювання площ	Автоматично підбирає список найближчих вільних приміщень клієнту .

Функції системи другого рівня представлені в таблиці 2.2.

Таблиця 2.2 – Функції системи другого рівня

Код	Елемент	Функція
1.1	Вхідні параметри для пошуку приміщення	Забезпечує пошук об'єктів по необхідним параметрам
1.2	Алгоритм обробки параметрів	Забезпечує фільтрування й сортування об'єктів
1.3	Вихідні параметри	Забезпечує отримання бажаного результату в виді списку об'єктів.

Функції системи третього рівня представлені в таблиці 2.3.

Таблиця 2.3 – Функції системи третього рівня

Код	Елемент	Функція
1.1.1	Вибір локації	Забезпечує пошук об'єктів на певній локації.
1.1.2	Місткість приміщення	Забезпечує знаходження об'єктів з певною кількістю.
1.1.3	Вибір додаткового обладнання	Забезпечує знаходження об'єктів з певним обладнанням в кімнаті.
1.1.4	Вибір часу для бронювання	Забезпечує знаходження об'єктів в певний час бронювання.
1.2.1	Алгоритм місцезнаходження клієнта	Забезпечує підібрати найближчі вільні номерні фонди
1.2.2	Алгоритм отримання об'єктів	Забезпечує отримати об'єкти в зручному вигляді для подальших дій
1.2.3	Алгоритм фільтрації об'єктів	Забезпечує отримати певні об'єкти, які є зручними у використанні клієнтом
1.2.4	Алгоритм сортування об'єктів	Забезпечує отримати список об'єктів в деякій визначеності зручної клієнту
1.3.1	Список бажаних об'єктів	Забезпечує клієнту використання цих об'єктів для своїх власних цілей

Функції системи четвертого рівня представлені в таблиці 2.4.

Таблиця 2.4 – Функції системи четвертого рівня

Код	Елемент	Функція
1.3.1.1	Бронювання обраного об'єкту	Забезпечує клієнту отримати бажане найближче приміщення для проведення зустрічі

Параметри системи першого рівня представлені в таблиці 2.5.

Таблиця 2.5 – Параметри системи першого рівня

Код	Елемент	Параметри
1	Автоматизована система бронювання площ	Тип вхідних даних Кількість вхідних даних Процес обробки даних Тип вихідних даних

Параметри системи другого рівня представлені в таблиці 2.6.

Таблиця 2.6 – Параметри системи другого рівня

Код	Елемент	Параметри
1.1	Вхідні параметри для пошуку розміщення	Тип даних Розмір даних Кількість даних
1.2	Алгоритм обробки параметрів	Середній час роботи алгоритму
1.3	Вихідні параметри	Тип даних Розмір даних Кількість даних

Параметри системи третього рівня представлені в таблиці 2.7.

Таблиця 2.7 – Параметри системи третього рівня

Код	Елемент	Параметри
1.1.1	Вибір локації	Тип даних Числові поля

Продовження таблиці 2.7

1.1.2	Місткість приміщення	Тип даних Числове поле
1.1.3	Вибір додаткового обладнання	Типи даних Множинний вибір
1.1.4	Вибір часу для бронювання	Часовий тип даних Єдиний вибір
1.2.1	Алгоритм місцезнаходження клієнта	Тип даних Кількість даних
1.2.2	Алгоритм отримання об'єктів	Кількість даних Тип даних Розмір даних
1.2.3	Алгоритм фільтрації об'єктів	Кількість даних на фільтрування Тип даних
1.2.4	Алгоритм сортування об'єктів	Кількість даних на сортування Тип даних
1.3.1	Список бажаних об'єктів	Остаточна кількість даних Тип даних Розмір даних

Параметри системи четвертого рівня представлені в таблиці 2.8.

Таблиця 2.8 – Параметри системи четвертого рівня

Код	Елемент	Параметри
1.3.1.1	Бронювання обраного об'єкту	Розмір даних Тип даних Кількість властивостей

Підсумкова і сумарна кількість функцій:

- Функції першого рівня – 1;

- Функції другого рівня – 3;
- Функції третього рівня – 9;
- Функції четвертого рівня – 1.

В таблиці 2.9 розглянуті фактори системи.

Таблиця 2.9 – Фактори системи

Функціональність	Ранг	Фактори	
		Системоруйнівні	Системоутворюючі
Багатофункціональна	Обслуговування	Відсутність ін-тернету	Доступність до ін-тернету

2.3. Інформаційний опис системи

Інформаційний опис повинен давати уявлення про організацію та управління системою. Це найбільш повне описання, так як описує систему вже в процесі її функціонування. Інформаційний опис визначає залежність морфологічних та функціональних властивостей системи від якості та кількості внутрішньої (про середовище) і зовнішньої (що надходить із зовнішнього середовища) інформації.

Перелік елементів системи:

Вхідні параметри – 1;

Алгоритм отримання місцезнаходження клієнта – 2;

Алгоритм отримання об'єктів – 3;

Алгоритм фільтрування об'єктів – 4;

Алгоритм сортування об'єктів – 5;

Список бажаних об'єктів – 6;

Заброньований номерний фонд – 7;

Властивості деталей елементів системи представлені в таблиці 2.10.

Таблиця 2.10 – Властивості деталей елементів системи

№ п/п	Найменування	Позначення	Кількість властивостей	Примітка
1	Вхідні параметри	a_1	1	1(1) – передача інформації для бронювання фонду
2	Алгоритм отримання місцезнаходження клієнта	a_2	2	1(2) – отримує геокоординати користувача 2(2) – автоматизація системи
3	Алгоритм отримання об'єктів	a_3	2	1(3) – отримує всі об'єкти 2(3) – автоматизація системи
4	Алгоритм фільтрування об'єктів	a_4	2	1(4) – фільтрує отримані дані 2(4) – автоматизація системи
5	Алгоритм сортування об'єктів	a_5	2	1(5) – сортування даних 2(5) – автоматизація системи
6	Список бажаних об'єктів	a_6	1	1(6) – отримання бажаних об'єктів в виді списку
7	Заброньований номерний фонд	a_7	1	1(7) – Бронювання найзручнішого приміщення з точки зору клієнта

Середньгеометричне число властивостей на один елемент:

$$\bar{a} = \sqrt[7]{a_1 * \dots * a_8} = \sqrt[7]{1 * 2 * 2 * 2 * 2 * 1 * 1} = \sqrt[7]{48} \approx 1.49$$

Структура об'єкта у вигляді графа представлена на рисунку 2.2.

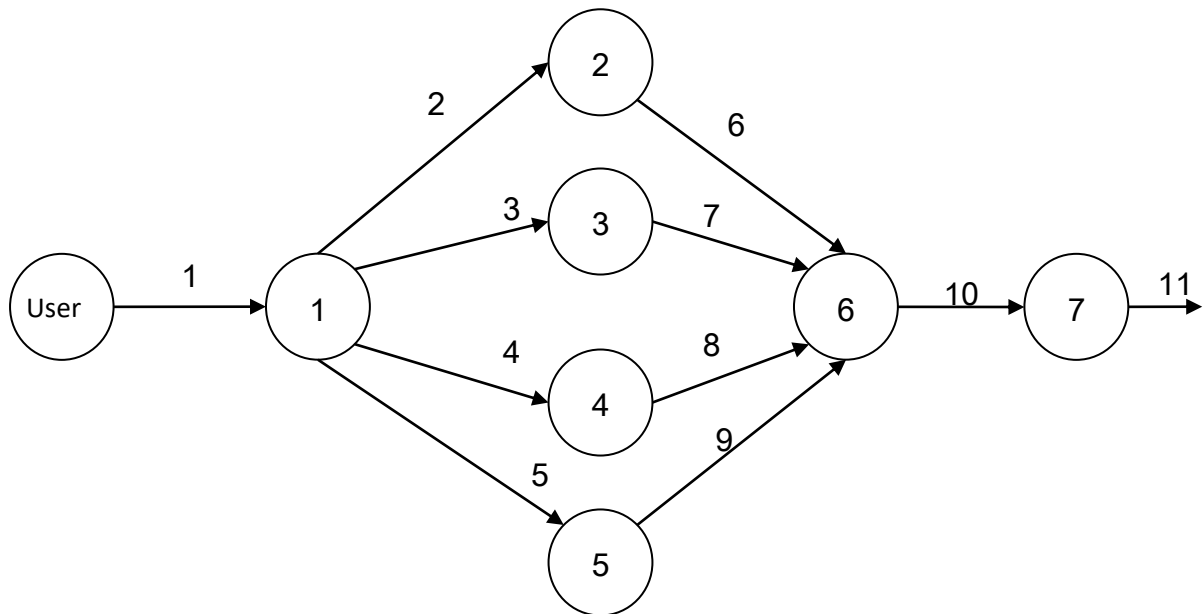


Рисунок 2.2 – Структура об'єкта у вигляді графа

Зв'язки системи між елементами:

- 1 – Користувач – вхідні параметри
- 2 – Вхідні параметри – алгоритм отримання місцезнаходження клієнта
- 3 – Вхідні параметри – алгоритм отримання об'єктів
- 4 – Вхідні параметри – алгоритм фільтрування об'єктів
- 5 – Вхідні параметри – алгоритм сортування об'єктів
- 6 – Алгоритм отримання місцезнаходження клієнта – список бажаних об'єктів
- 7 – Алгоритм отримання об'єктів – список бажаних об'єктів
- 8 – Алгоритм фільтрування об'єктів – список бажаних об'єктів
- 9 – Алгоритм сортування об'єктів – список бажаних об'єктів
- 10 – Список бажаних об'єктів – заброньований номерний фонд
- 11 – Заброньований номерний фонд – головне меню

Кількість зв'язків на один елемент системи представлено в таблиці 2.11.

Таблиця 2.11 – Кількість зв'язків на один елемент системи

№ п/п	Найменування елемента	Кількість зв'язків
1	Вхідні параметри	5
2	Алгоритм отримання місцезнаходження клієнта	2
3	Алгоритм отримання об'єктів	2
4	Алгоритм фільтрування об'єктів	2
5	Алгоритм сортування об'єктів	2
6	Список бажаних об'єктів	5
7	Оброблений документ	2

Середньгеометричне число зв'язків на один елемент:

$$Y = \sqrt[7]{5 * 5 * 2 * 2 * 2 * 2 * 2} = \sqrt[7]{800} \approx 2.6.$$

2.4. Класифікаційний опис системи

Класифікація – це поділ сукупності об'єктів на класи за найбільш суттєвими ознаками, які називаються класифікаційними.

Результати класифікації системи представлені в таблиці 2.12.

Таблиця 2.12 – Результати класифікації системи

№ п/п	Ознака класифікації	Тип системи за ознакою	Визначення
1	За зв'язком системи з навколишнім середовищем	Відкрита	Взаємодіє з навколишнім середовищем

Продовження таблиці 2.12

2	За походженням	Штучна	Створена людиною
3	За об'єктивності існування	Реальна	Складається з природних і штучних об'єктів
4	За типом опису законів функціонування	Параметризована система	Відомий закон функціонування з точністю до параметра
5	За способом управління системою	з комбінованим керуванням	блок управління системи, що забезпечує управління, підпорядковується також зовнішньому управлінню
6	По дії	Технічна	Сукупність взаємопов'язаних фізичних елементів
7	За централізацією	Децентралізована система	Всі складові їх компоненти приблизно однаково значущі.
8	За однорідністю структури	Гетерогенна структура	Складаються з різнорідних елементів, що не володіють властивістю взаємозамінності
9	За типом складності	Алгоритмічна	Система має сукупність алгоритмів
10	За мірністю	Одномірна	Один вхідний параметр та один вихідний параметр.
11	По організованості	Добре організована	Визначені всі елементи, зв'язки і залежності між елементами і цілями
12	За безперервністю	Неперервна	Всі елементи і зв'язки мають неперервний характер
13	За обумовленістю дії	Детермінована	Поведінку можна однозначно передбачити

3. МОДЕЛЬНЕ ПРЕДСТАВЛЕННЯ ОБ'ЄКТА ДОСЛІДЖЕННЯ І ЙОГО ХАРАКТЕРИСТИКИ

3.1. Вхідні й вихідні величини

Вхідними величинами являються параметри пошуку приміщень в панелі інтерфейсу програми. Їх відмічає користувач для знаходження очікуваних даних в своїх інтересах. Потім ці параметри відправляються запитом на сервер для обробки об'єктів, а саме: фільтрування сортування даних. Потім відбувається відправка із серверу відповідних результатів на панель інтерфейсу в якості вихідних величин.

Вихідними величинами являється список об'єктів, які описують приміщення. Вони мають велику кількість різних характеристик. Після вибору певного номерного фонду з'являється можливість його бронювання користувачем .

3.2. Дослідження області допустимих значень вхідних величин

Проект реалізую на об'єктно-орієнтованій мові програмування C# (C Sharp) для реалізації цілей.

Серед актуальних мов програмування я вибрав саме цей продукт, так як він являється таким:

- простий;
- сучасний;
- високорівнева мова програмування;
- об'єктно-орієнтована мова з безпечною системою типів;
- інтегрована підтримка багатьох різних структур даних;

Зовсім різні ІТ-компанії серед багатьох інших сучасних мов використовують саме цей продукт для виконання своїх цілей. Тому на сьогодні він є актуальним на світовому ринку праці.

Серед інших програм має такі переваги:

- зручність в користуванні;
- проста наочність;
- сучасна;

Для вхідних даних потрібен клієнт з встановленим відповідним додатком, який відмітить критерії пошуку під свої зручності, а саме:

- Вибір локації;
- Місткість приміщення;
- Множинний вибір додаткового обладнання;
- Вибір часу для бронювання.

Областю допустимих значень для локації являються одиночний вибір країни, міста, офісу із випадуючого списку радіо-кнопок, для вибору місткості та часу також використовуються перемикачі, для указання додаткового обладнання використано множинний вибір чек-боксів.

3.3. Прогнозування області значень вихідних величин

Вхідними величинами в мобільному додатку є об'єкти у вигляді списку з різними вказаними характеристиками . Областю допустимих значень являються строкові, числові, часові, та вкладені складні типи даних. Вони несуть в собі смислову інформацію про повний опис об'єкта.

В строковий тип даних входять:

- Ідентифікатор об'єкту;
- Назва об'єкту;
- Назва вулиці;
- Назва міста;
- Назва країни;
- Список назв обладнань;

В числовий тип даних входять:

- Місткість об'єкту;

- Час в мілісекундах зайнятості приміщення;
- Статус доступності кімнати;
- Поверх;

В часовий тип даних входять:

- Дата старту й кінця бронювання;
- Теперішня дата;

До складних вкладених типів даних відносяться об'єкти, які описують складну предметну область, наприклад для співробітників, які запрошені на зустріч до цього фонду.

3.4. Можливі методи рішення модельного представлення

Основною ціллю автоматизованої системи бронювання площ житлових та нежитлових фондів являється мінімізувати дії клієнта для бронювання приміщень з мінімальною витратою часу.

Система, в якій для отримання і обробки інформації, а також для управління, використовуються різні автоматичні пристрої. Термін «автоматизована» має на увазі обов'язкову та основну участь алгоритмів в системі в бекенд частині. Таку систему часто називають ергатичною. Ці системи значно підвищують ефективність управління, від якого безпосередньо залежить соціальна ефективність, оскільки управління містить в собі технічну та соціальну частини[8].

Програма має список приміщень, які містять різні характеристики. Цей список повинен бути для того, щоб підібрати зі значної вибірки максимально близьку до критерій кімнату для клієнта.

Розглянемо ситуацію з бронюванням номерного фонду: клієнт в першу чергу встановлює свої критерії пошуку. Потім ці входні параметри відправляються до серверної частини.

В самому процесі роботи для того, щоб підібрати потрібні об'єкти, програма спочатку структурує дані, тобто перезаписує собі в певному форматі, щоб надалі легше можна було з ними працювати, автоматизувати. Під словом авто-

матизація входить:

- Алгоритм отримання місцезнаходження клієнта
- Алгоритм отримання об'єктів
- Алгоритм фільтрування об'єктів
- Алгоритм сортування об'єктів

Після того як всі об'єкти були оброблені, переглянуті програма подає їх на вихід клієнту на панель інтерфейсу мобільного додатку у вигляді списку вільних приміщень на час, який вказав користувач.

Нижче на рисунку 3.1 показано схему процесу автоматизованої системи для бронювання площ житлових та нежитлових фондів.

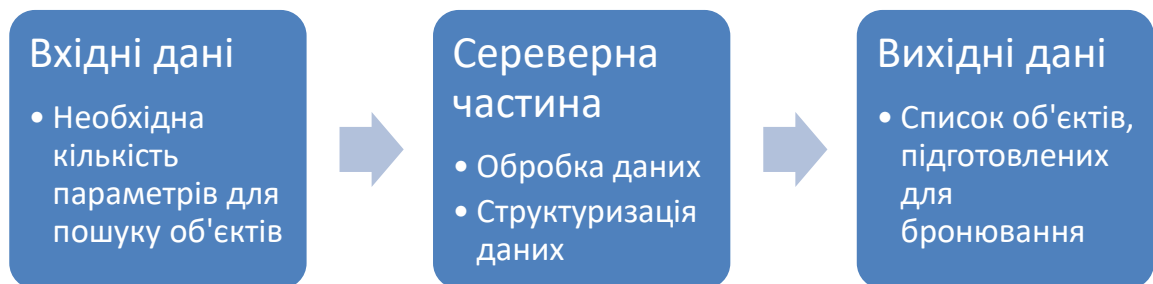


Рисунок 3.1 – Схема процесу автоматизованої системи для бронювання площ житлових та нежитлових фондів

4. ТЕХНІЧНА ЧАСТИНА

4.1. Вибір середі розробки програмного продукту

Важним елементом являється вибір середі розробки для мобільного додатку, що складається з трьох частин:

- Data Base частина;
- Backend частина;
- UI частина.

Проаналізувавши різні мови програмування було визначено, що найближче підходить розробка backend частини мовою C# на платформі .NET Core з використанням шаблону MVC, UI частина розроблена на фреймворці Angular мовою TypeScript, а Data Base частина знаходиться на Mongo DB, але це умовно, тому що вибір дуже залежить від конкретного розробника, його стилю і досвіду.

C# (вимовляється як "сі Шарп") - проста, сучасна об'єктно-орієнтована і типобезпечна мова програмування. C # відноситься до широко відомому сімейству мов C, і здається добре знайомим кожному, хто працював з C, C ++, Java.

Важлива особливість таких компонентів - це модель програмування на основі властивостей, методів і подій. Кожен компонент має атрибути, які надають декларативні відомості про компоненті, а також вбудовані елементи документації. C # надає мовні конструкції, безпосередньо підтримують таку концепцію роботи. Завдяки цьому C# відмінно підходить для створення і застосування програмних компонентів.

Ось лише кілька функцій мови C#, що забезпечують надійність і стійкість додатків: "прибирання сміття" автоматично звільняє пам'ять, зайняту знищеними і невикористовуваними об'єктами; "обробка виключень" надає структурований і розширюваний спосіб виявляти і обробляти помилки; "сувора типізація мови" не дозволяє звертатися до неініціалізованих змінних, виходити за межі індексованих масивів або виконувати неконтрольоване приведення типів[11].

Має такі переваги:

- підтримка корпорацією Майкрософт. На відміну від Java, якій не пішов на користь перехід у власність Oracle, C # добре розвивається завдяки зусиллям Microsoft;
- останнім часом багато вдосконалюється;
- багато синтаксичного цукру. Синтаксичний цукор - це такі конструкції, які створені для полегшення написання і розуміння коду (особливо якщо це код іншого програміста) і не грають ролі при компіляції;
- середній поріг входження. Синтаксис схожий на C, C ++ або Java, що полегшує перехід для інших програмістів. Для новачків це також один з найперспективніших мов для вивчення;
- велика спільнота програмістів;
- багато вакансій на посаду C # програміста в будь-якому регіоні.

Є й недоліки:

- орієнтованість, в основному, тільки на .NET (на Windows платформу);
- безкоштовність тільки для невеликих компанії, учнів і програмістів-одинаків.

Платформа ASP.NET Core представляє технологію від компанії Microsoft, призначену для створення різного роду веб-додатків: від невеликих веб-сайтів до великих веб-порталів і веб-сервісів.

З одного боку, ASP.NET Core є продовженням розвитку платформи ASP.NET. Але з іншого боку, це не просто черговий реліз. Вихід ASP.NET Core фактично означає революцію всієї платформи, її якісна зміна.

Для обробки запитів тепер використовується новий конвеєр HTTP, який заснований на компонентах Katana і специфікації OWIN. А його модульність дозволяє легко додати свої власні компоненти.

Якщо підсумувати, то можна виділити наступні ключові відмінності ASP.NET Core від попередніх версій ASP.NET:

- Новий легкий і модульний конвеєр HTTP-запитів
- Можливість розгортати додаток як на IIS, так і в рамках свого вла-

сного процесу

- Використання платформи .NET Core і її функціональності
- Поширення пакетів платформи через NuGet
- Інтегрована підтримка для створення та використання пакетів NuGet
- Єдиний стек веб-розробки, що поєднує Web UI і Web API
- Конфігурація для спрощеного використання в хмарі
- Вбудована підтримка для впровадження залежностей
- Можливість розширення
- Кросплатформеність: можливість розробки і розгортання додатків

ASP.NET на Windows, Mac і Linux

- Розвиток як open source, відкритість до змін

Ці та інші особливості і можливості стали основою для нової моделі програмування.

Angular – це фреймворк від компанії Google для створення клієнтських додатків. Перш за все він націлений на розробку SPA-рішень (Single Page Application), тобто односторінкових додатків.

Angular надає таку функціональність, як двостороннє зв'язування, що дозволяє динамічно змінювати дані в одному місці інтерфейсу при зміні даних моделі в іншому: шаблони, маршрутизація і так далі.

Однією з ключових особливостей Angular є те, що він використовує в якості мови програмування TypeScript.

TypeScript представляє мову програмування на основі JavaScript, це строго типізований і компільований, буде ближче до програмістам Java, C # та інших строго типізованих мов. Хоча на виході компілятор створює все той же JavaScript, який потім виконується браузером. Однак сувора типізація зменшує кількість потенційних помилок, які могли б виникнути при розробці на JavaScript.

TypeScript реалізує багато концепцій, які властиві об'єктно-орієнтованим мовам, як, наприклад, спадкування, поліморфізм, інкапсуляція і модифікатори доступу і так далі.

потенціал TypeScripta дозволяє швидше і простіше писати великі складні комплексні програми, відповідно їх легше підтримувати, розвивати, масштабувати і тестувати.

Крім того, він є кросплатформним, а це значить, що для розробки ми можемо використовувати як Windows, так і MacOS або Linux.

MongoDB реалізує новий підхід до побудови схем баз даних, де немає таблиць, запитів SQL, зовнішніх ключів і багатьох інших речей, які притаманні об'єктно-реляційних баз даних.

Вся система MongoDB може представляти не тільки одну базу даних, що знаходиться на одному фізичному сервері. Функціональність MongoDB дозволяє розташувати кілька баз даних на декількох фізичних серверах, і ці бази даних зможуть легко обмінюватися даними і зберігати цілісність.

Одним з популярних стандартів обміну даними та їх зберігання є JSON (JavaScript Object Notation). JSON ефективно описує складні за структурою дані. Спосіб зберігання даних в MongoDB в цьому плані схожий на JSON, хоча формально JSON не використовується. Для зберігання в MongoDB застосовується формат, який називається BSON (Бісон) або скорочення від binary JSON.

BSON дозволяє працювати з даними швидше: швидше виконується пошук і обробка. Хоча треба зазначити, що BSON на відміну від зберігання даних в форматі JSON має невеликий недолік: в цілому дані в JSON-форматі займають менше місця, ніж в форматі BSON

Створення функціоналу програми (API):

За вище описаними платформи, середі розробки та інших сторонніх хелперів в побудуванні програмного продукту створюємо додаток.

Один з найважливіших пунктів додатку є CRUD методи які повністю управляють об'єктами.

CRUD (скор. Від англ. Create, read, update, delete - «створити, прочитати, оновити, видалити») - що позначає чотири базові функції, які використовуються при роботі з персистентного сховищами даних [1]:

- створення;

- читання;
- редагування;
- видалення.

Коли сервер отримав доступ щоб отримати список всіх приміщень, за допомогою бекенд частини можемо переглянути в вигляді json об'єктів(рисунок 4.1). Щоб отримати дані потрібно бути авторизованим в програмі, тобто бути робітником цієї компанії. Потім компанія надає архів зареєстрованих кімнат з усіма дповоненнями від країни до кількості сидячих місць в середині кімнати!

Але багато даних є спотвореними тому необхідно було розробити UI адмін панель. Для якого доступ є, хто занесений в БД під роллю адмін. В цьому додатку є можливість під коректувати деталі приміщення.

```
{
  "dataHubId": "4060741400306553064",
  "name": "Italy",
  "isAvailable": true,
  "cities": [
    {
      "dataHubId": "4060741400306553094",
      "dataHubCountryId": "4060741400306553064",
      "name": "Milan",
      "isAvailable": true,
      "offices": [
        {
          "dataHubId": "4060741400334745984",
          "dataHubCityId": "4060741400306553094",
          "address": "Via Morimondo, 26",
          "geoCoordinates": {
            "latitude": 45.445031,
            "longitude": 9.1504393
          },
          "isCoordinatesSetManually": false,
          "isAvailable": true
        }
      ]
    }
  ]
},
```

Рисунок 4.1 – Структура у вигляді json об'єкта

Також я використовую одну із зручних утіліт для огляду бекенд частини через Swagger: це машиннозчитуване уявлення RESTful API, яке забезпечує підтримку інтерактивної документації, створення клієнтських пакетів SDK. На рисунку 4.2 зображено вигляд цієї утіліти.

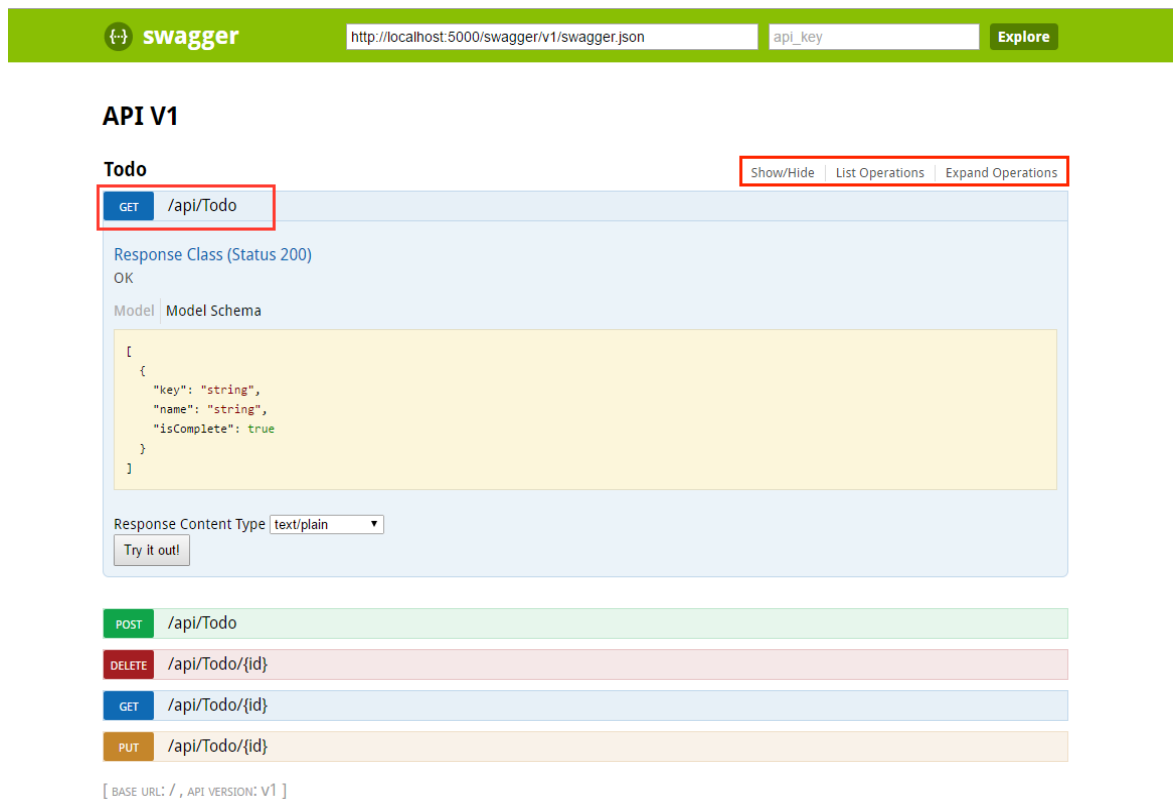


Рисунок 4.2 – Зображення swagger

4.2. Шаблон MVC

Платформа ASP.NET MVC представляє собою фреймворк для створення сайтів і веб-додатків за допомогою реалізації паттерна MVC.

Концепція паттерна (шаблону) MVC (model - view - controller) передбачає поділ додатка на три компоненти:

Контролер (controller) представляє клас, що забезпечує зв'язок між користувачем і системою, поданням і сховищем даних. Він отримує вводяться користувачем дані і обробляє їх. І в залежності від результатів обробки відправляє користувачеві певний висновок, наприклад, у вигляді подання.

Представлення (view) - це власне візуальна частина або призначений для користувача інтерфейс програми. Як правило, html-сторінка, яку користувач бачить, зайшовши на сайт.

Модель (model) - представляє сховище даних нашого проекту і реагує на команди контролера, змінюючи свій стан.

На рисунку 4.3 зображена взаємодія компонентів шаблону MVC.

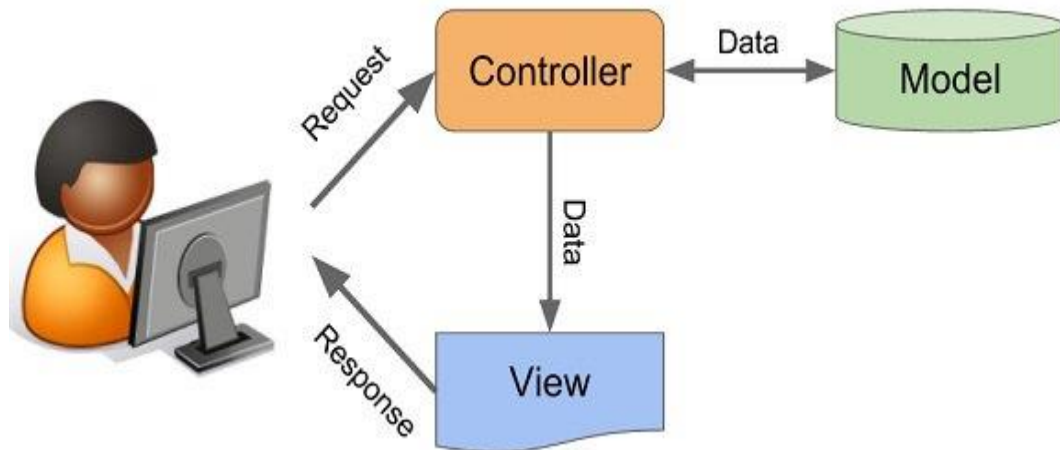


Рисунок 4.3 – Взаємодія компонентів шаблону MVC

У цій схемі модель є незалежним компонентом - будь-які зміни контролера або представлення не зачіпають модель. Контролер і представлення є відносно незалежними компонентами, і нерідко їх можна змінювати незалежно один від одного. Завдяки цьому реалізується концепція поділу відповідальності, в зв'язку з чим легше побудувати роботу над окремими компонентами. Крім того, внаслідок цього додаток має кращу тестованість. І якщо нам, припустимо, важлива візуальна частина або фронтенд, то ми можемо тестувати уявлення незалежно від контролера. Або ми можемо зосередитися на бекенді і тестувати контролер.

Конкретні реалізації та визначення даного патерну можуть відрізнитися, але в силу своєї гнучкості і простоти він став дуже популярним останнім часом, особливо в сфері веб-розробки.

4.3. Процес роботи мобільного додатку

При дипломній роботі був розроблений відповідний додаток, що відповідає критеріям, описаним у попередніх главах.

Взаємодія клієнт-серверного додатку:

- 1) посилається запит на сервер;
- 2) контролер приймає запит виконує обробку, в разі потреби звертається до

моделі;

- 3) модель в свою чергу змінює свій стан;
- 4) відправляє дані в контролер;
- 5) контролер зробивши необхідні дії з даними віддає їх в представлення, що йде відповідно юзеру в виді UI.

Для прикладу було взято сферу підприємства, які мають безліч номерних фондів в різних філіях, які знаходяться по всьому світу. Для подібної сфери зроблено мобільний додаток, який дає можливість забронювати приміщення для проведення зустрічей чи своїх цілей тощо.

При відкритті мобільного додатку в даний момент знаходимося на головній сторінці, що зображено на рисунку 4.4.

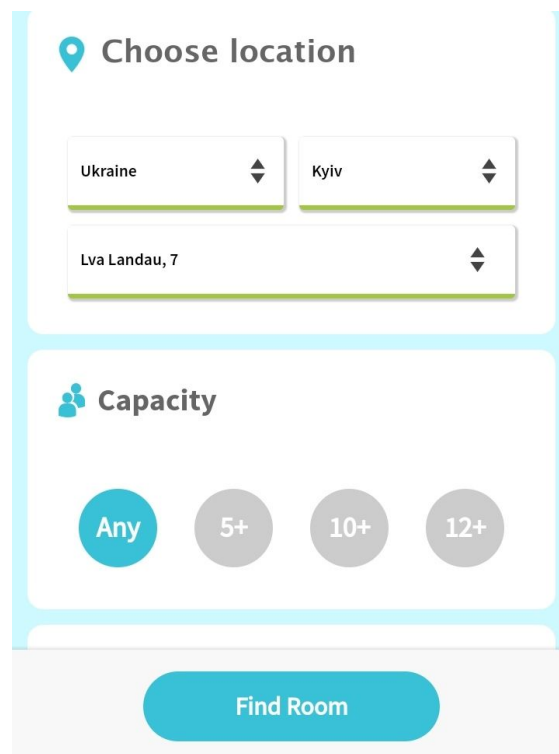


Рисунок 4.4 – Інтерфейс головної сторінки

Користувач бачить навігаційне меню, через яке формує критерії під свої зручності для пошуку доступних кімнат, потім натискає кнопку «Find room». На рисунку 4.5 відображено сторінку, в якій, відображено результат пошуку всіх доступних кімнат.

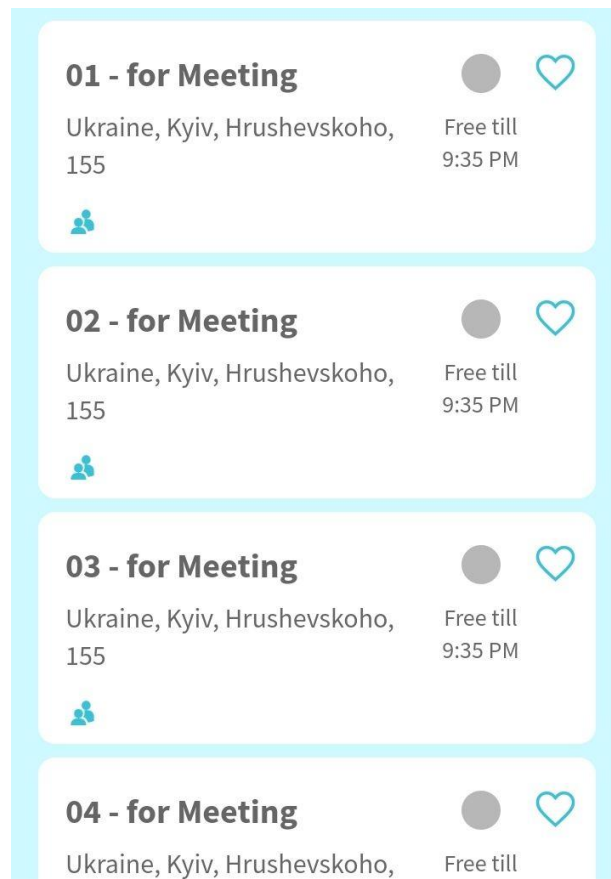


Рисунок 4.5 – Результати пошуку кімнат

Юзер натискає на одну із них і потрапляє на сторінку, де зображено детальний опис кімнати і є можливість вибору часових рамок та вводу назви зустрічі, що зображено на рисунку 4.6.

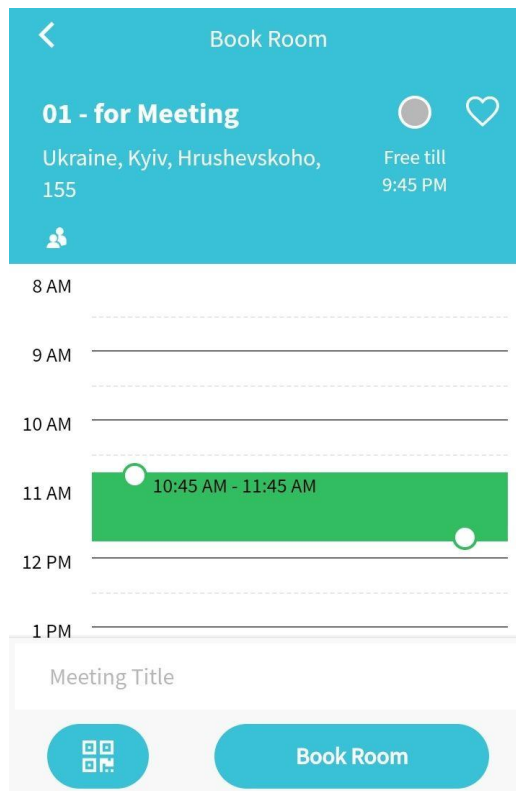


Рисунок 4.6 – Сторінка детального опису кімнати

При натиску на кнопку «Book Room» відправляється запит на серверну частину для бронювання кімнати, заміни статусу на зайнято, запису в календар зустрічей користувача. При завершенні перелічених процесів поверх сторінки з'являється повідомлення про успішне бронювання(рисунок 4.7) і вертаємося на цю ж саму сторінку для бронювання.

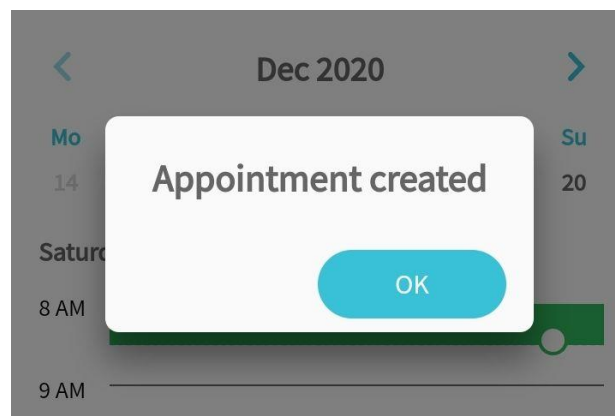


Рисунок 4.7 – Повідомлення успішного бронювання

На рисунку 4.8 бачимо, що ті часові рамки підсвічуються іншим кольором, що означає недоступність кімнати в цей час і не можна забронювати її.

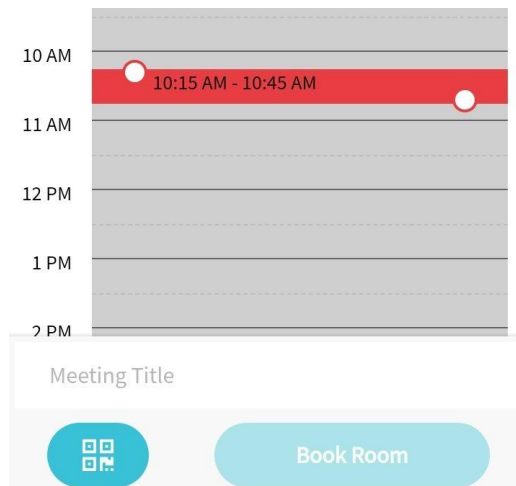


Рисунок 4.8 – Рамки часу недоступності кімнати

Якщо перейти в календар, натиснувши на іконку внизу панелі, після переходу, можна побачити всі свої зустрічі по відповідним кімнатам(рисунок 4.9).

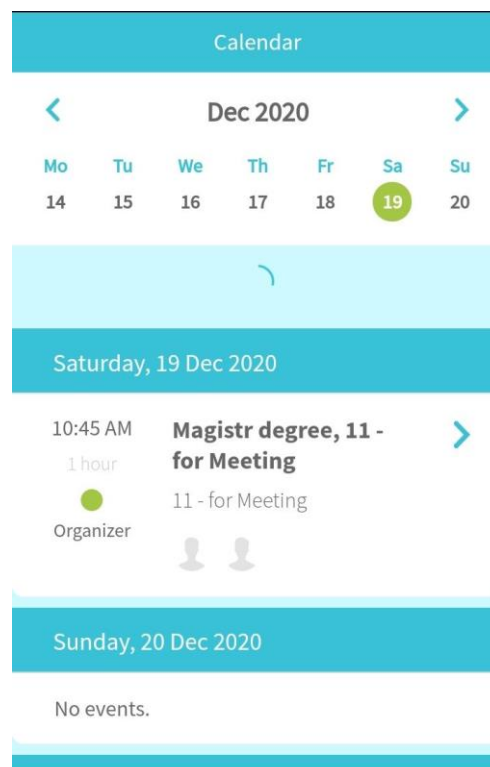


Рисунок 4.9 – Календар

Натиснувши на саму зустріч клієнт перейде на її деталі, що зображено на рисунку 4.10, якщо роль клієнта – організатор, то є можливість відмінити її, натиснувши на кнопку «Cancel Meeting».

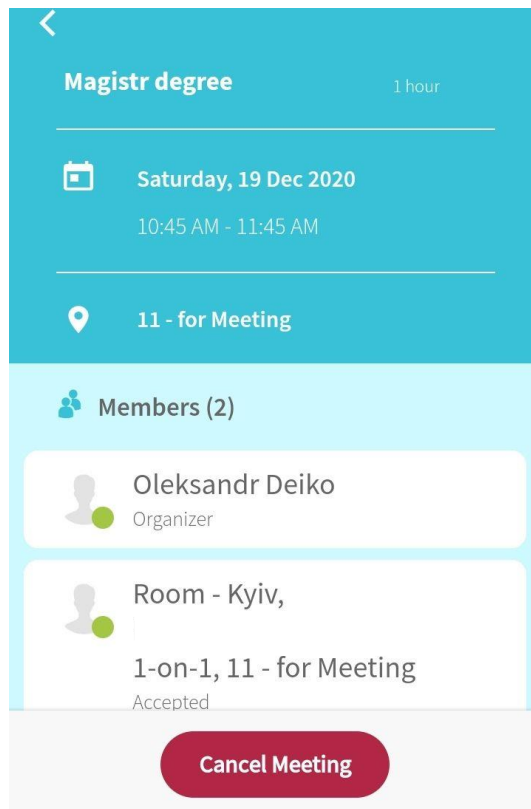


Рисунок 4.10 – Деталі зустрічі

При натиску «Cancel meeting» проходить процес відміни зустрічі, що відбувається на серверній частині, потім сторінка оновлюється клієнт знаходиться в кладці календар, і цієї зустрічі уже не буде, а користувач отримає спливаюче вікно повідомлення про відміну.

Також дана розробка має можливість швидкого пошуку найближчих клієнту приміщень, достатньо натиснути вверху екрану «Quick search rooms» запуститься даний процес на деякий час, як зображено на рисунку 4.11.

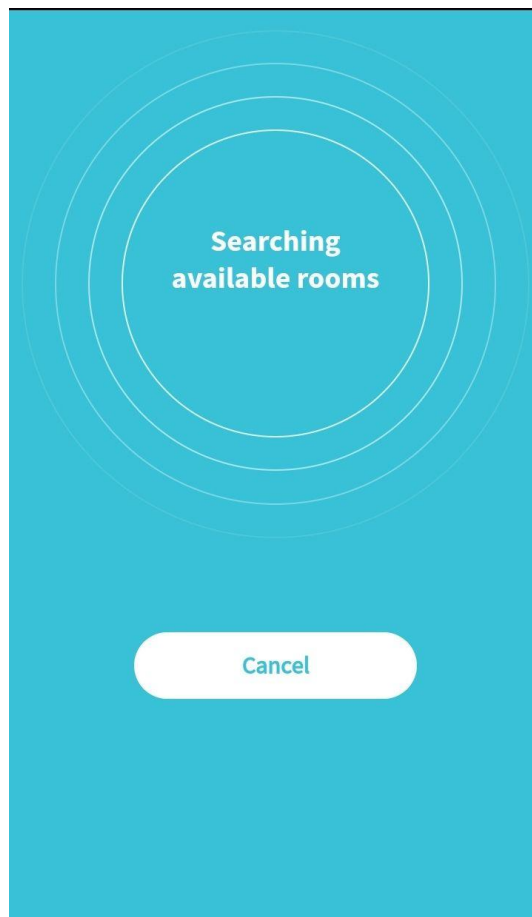


Рисунок 4.11 – процес пошуку найближчих кімнат

Після завершення пошуку користувач отримає наблизчі до нього вільні приміщення у вигляді списку, які дозволено відразу забронювати на найближчі 30 хвилин, список даних зображений на рисунку 4.12.

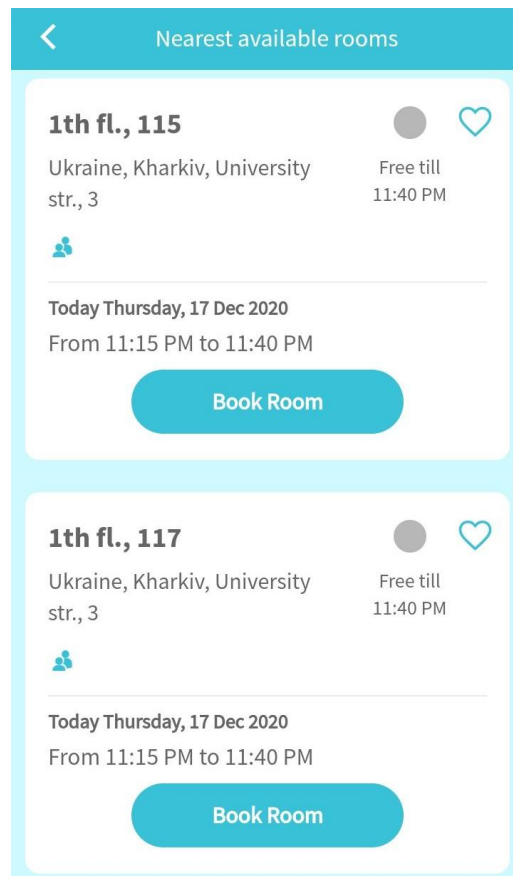


Рисунок 4.12 – список доступних найближчих кімнат клієнту

Дана операція має сенс використовуватися клієнтом, коли негайно позапланово потрібно зустрітися з іншим учасником, щоб обговорити певні цілі на найближчий час. Або ж коли приходиш до офісу зі своїми партнерами по проекту і потрібно негайно обговорити дещо, ця функція є незамінною і ефективною в даному додатку.

4.4. База даних для мобільного додатку

Розробка бази даних починається з проектування. На етапі проектування визначається структура бази даних: перелік необхідних таблиць, їх поля і логічні зв'язки. Структура таблиці визначається переліком необхідних стовпців, їх типом даних, послідовністю, обмеженнями, значеннями за замовчуванням, а також ключем (або ключами) таблиці. Багато деталей залежить від архітектури програмного продукту [12].

Процес проектування бази даних на основі принципів нормалізації представляє собою послідовність переходів від неформального словесного опису інформаційної структури предметної області до формалізованого опису об'єктів предметної області в термінах деякої моделі [13].

Всі сутності в додатку прийнято виділяти в окремі моделі. Залежно від поставленого завдання і складності програми можна виділити різну кількість моделей. В даній програмі було створено такі моделі:

- клас для приміщення;
- клас для користувачів;
- клас для країн, які включають в собі ієрархію;

Один з основних сутностей показано в таблиці 4.1.

Таблиця 4.1 – Формалізований опис об'єкту

Назва таблиці	Ідентифікатор	Тип даних
Room	fullName	STRING
	shortName	STRING
	capacity	INTEGER
	floor	INTEGER
	dateRange	DATE
	officeID	STRING
	ID	STRING
	equipment	ARRAY

На рисунку 4.13 показана логічна схема бази даних.

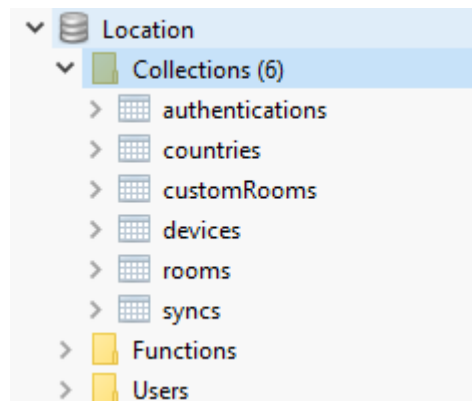


Рисунок 4.13 – логічна схема

Висновок по четвертому розділу

В даному розділі було показано процес роботи програмного продукту, а саме клієнт-серверного мобільного додатку, який обробляє вхідні параметри, вибрані клієнтом під свої зручності для пошуку відповідних кімнат та їх бронювання, а також перегляду статусу кімнат учасником в реальному часі.

Після процесу користування програмним продуктом, переконалися, що, дійсно користувач мінімізував свої дії для бронювання приміщення всього в три кліки. Також цей процес дозволено в будь-якому місці знаходження з доступом до інтернету. Без мобільного додатку, користувачеві потрібно було мати ввімкнену воркстанцію, в якій є доступ до мережі, потім витратити певний час для знаходження потрібної кімнати чи приблизної для своїх потреб та забронювати через електронну пошту Outlook.

5. ЕКОНОМІЧНА ЧАСТИНА

5.1. Опис програмного продукту

Дана глава присвячена кількісній оцінці матеріальних витрат на виробництво представленого в дипломній роботі програмного продукту, призначеного для дослідження автоматизованої системи для розподілу різнорідних завдань.

Під програмним продуктом (ПП) розуміється програмне забезпечення (ПЗ) як результат людської діяльності, виставлений на ринку масового покупця в якості товару і має ненульову споживчу вартість.

Продукт розроблений в програмному середовищі Visual Studio 2019 C# та використовує клієнт-серверну технологію Microsoft SQL Server Express 2014.

5.2. Визначення трудовитрат робіт

Метою економічного розділу є розрахунок собівартості і вартості програмного продукту.

Для того щоб оцінити вартість розробленого програмного продукту необхідно:

1. скласти перелік робіт, які слід виконати, потім розрахувати трудовитрати на їх виконання;
2. розрахувати заробітну плату розробників;
3. розрахувати витрати на матеріали, комплектуючі та машинний час;
4. відрахування на соціальні заходи.

У витрати на розробку програмного продукту також входять: вартість малоцінних і швидкозношуваних предметів, вартість оренди комп'ютера, відрахування з заробітної плати і т.д.

До переліку етапів робіт, які необхідно виконати входить:

1. формулювання постановки задачі;
2. проектування програмного продукту;
3. розробка програмного продукту;

4. впровадження продукту.

Тривалість кожного етапу визначається за формулою:

$$T = \frac{t}{n}, \quad (5.1)$$

де T – тривалість етапу в робочих днях;

t – трудовитрати етапу;

n – кількість виконавців, одночасно зайнятих на певному етапі роботи.

5.3. Перелік робіт для створення програмного продукту

Для керівництва ходом робіт і ведення всього проекту в цілому необхідна посада керівника. Для проектування підсистеми, її подальшої налагодження і введення в експлуатацію необхідно участь програміста та дизайнера інтерфейсу.

Розрахуємо тривалість розробки продукту за видами робіт. Результати розрахунків містяться в таблиці 5.1.

В кінцевому підсумку, ми отримали, що термін створення програмного продукту – 46 днів.

5.4. Склад виконавців роботи та розрахунок заробітної плати

Дані про посадові оклади і склад виконавців роботи занесені в таблицю 5.2. Тривалість робочого місяця буде в середньому вважати 22 дня. Робочий день – восьмигодинний.

Таблиця 5.1 – Перелік робіт

№ етапу	Найменування етапу	Тривалість, дн.			Кількість, дн.			Тривалість
		Керівник	Програміст	Дизайнер	Керівник	Програміст	Дизайнер	
Розробка технічного завдання								
1	Організаційна підготовка до створення ПП	1			1			1
2	Розробка технічного завдання на постановку задачі	2			1			2
Разом:		3			1			3
Постановка задачі								
1	Розробка математичної моделі	3			1			3
2	Розробка опису завдання і технічного завдання	5			1			5
Разом:		8			1			8
Розробка програмного продукту								
1	Розробка алгоритмів	10	5		1	1		5
2	Розробка документації		5			1		5
3	Розробка програми		20	20		1	1	20
	Тестування		5			1		5
Разом:		10	35	20	1	1	1	35
Всього:		21	35	20	1	1	1	46

Таблиця 5.2 – Склад виконавців роботи

Посада	Посадовий оклад, грн.	
	Місячний	Добовий
Керівник	11000	500
Програміст	13200	600
Дизайнер	11000	500

Заробітна плата – винагорода за працю залежно від кваліфікації працівника, складності, кількості, якості та умов виконуваної роботи, а також компенсаційні виплати і стимулюючі виплати.

До витрат на заробітну плату праці відносяться основна і додаткова заробітна плата персоналу, зайнятого безпосередньо при виконанні конкретної теми: науковці, науково-технічний, науково-допоміжний персонал і виробничі робітники.

Основна заробітна плата ($ЗП_{осн}$) складається з суми середньої добової заробітної плати керівника, дизайнера та програміста, помноженої на тривалість їх праці відповідно і розраховується за формулою:

$$ЗП_{осн} = ЗП_{добкер} * T_{кер} + ЗП_{добпрог} * T_{прог} + ЗП_{добдиз} * T_{диз} \quad (5.2)$$

де $ЗП_{добкер}$, $ЗП_{добпрог}$, $ЗП_{добдиз}$ – добова заробітна плата керівника, програміста і дизайнера відповідно (таблиця 5.3);

$T_{кер}$, $T_{прог}$, $T_{диз}$ – тривалість праці керівника, програміста і дизайнера відповідно (таблиця 5.2).

Таким чином, основна заробітна плата, розрахована за формулою (5.2) дорівнює:

$$\begin{aligned} ЗП_{осн} &= 500 * 21 + 600 * 35 + 500 * 20 = \\ &10500 + 21000 + 10000 = 41500 \text{ (грн)}. \end{aligned}$$

Далі проводиться розрахунок додаткової заробітної плати ($ЗП_{\text{дод}}$), яка становить 20% від основної заробітної плати і розраховується:

$$ЗП_{\text{дод}} = ЗП_{\text{осн}} * Н_{\text{дод}}, \quad (5.3)$$

де $Н_{\text{дод}}$ – коефіцієнт додаткової зарплати, рівний 20%.

Отже, за формулою (6.3) додаткова заробітна плата дорівнює:

$$ЗП_{\text{дод}} = 41500 * 0,2 = 8300 \text{ (грн)}.$$

Разом, загальний фонд заробітної плати становить:

$$ЗП = ЗП_{\text{осн}} + ЗП_{\text{дод}} = 41500 + 8300 = 49800 \text{ (грн)}.$$

Нарахування на заробітну плату (єдиний соціальний внесок – $ЗП_{\text{соц}}$) складають 22% і розраховується за формулою:

$$ЗП_{\text{соц}} = ЗП * Н_{\text{соц}}, \quad (5.4)$$

де $Н_{\text{соц}}$ – коефіцієнт єдиного соціального внеску, рівний 22%.

Отже, за формулою (5.4) нарахування на заробітну плату становлять:

$$ЗП_{\text{соц}} = 49800 * 0,22 = 10956 \text{ (грн)}.$$

5.5. Перелік необхідного обладнання для створення програмного продукту

Вартість необхідного обладнання, а також їх призначення і необхідну кількість представлені в таблиці 5.3.

Таблиця 5.3 – Вартість основного обладнання

№ п/п	Найменування матеріалу	Призначення	Кількість, шт.	Ціна, грн
1	Ноутбук HP ENVY dv6 з Windows 10 Pro	Апаратне забезпе- чення для роботи	2	44000

Вартість додаткових послуг, а також їх призначення і необхідну кількість представлені в таблиці 5.4.

Таблиця 5.4 – Вартість додаткових послуг

№ п/п	Найменування матеріалу	Призначення	Кількість, шт.	Ціна, грн
1	Проведення інтернету TRIOLAN	Допомога для розро- бки програмного продукту	46	2000

Таблиці 5.5 – Вартість матеріалів

№ п/п	Найменування матеріалу	Призначення	Кількість, дн.	Ціна, грн.
1	Папір формату А4 (па- кунок 500 листів)	Оформлення доку- ментації та звітів	1	200
2	Картридж для принтера	Печать документації	1	500
3	Флеш – карта (8 Гб)	Збереження важли- вих даних	1	200
Разом:				900

Проведемо розрахунок амортизації (АМ) обладнання, в нашому випадку для ноутбуку HP ENVY dv6. Амортизацію обладнання як елемент собівартості даної продукції пропонується розраховуватимуть за такою методикою:

1. З урахуванням первісної вартості обладнання і річної норми амортизації розраховується річна сума амортизаційних відрахувань по даному виду обладнання за формулою:

$$AM_{j\text{рік}} = C_j * H_{\text{ам}}, \quad (5.5)$$

де $H_{\text{ам}}$ – коефіцієнт річної норми амортизації, який дорівнює 25%;

C_j – первісна вартість обладнання,;

$AM_{j\text{рік}}$ – сума річної амортизації обладнання j виду.

2. Визначається величина амортизаційних відрахувань у розрахунку на одну годину роботи обладнання даного виду:

$$AM_{j\text{год}} = \frac{AM_{j\text{рік}}}{T_j}, \quad (5.6)$$

де $AM_{j\text{год}}$ – величина амортизації обладнання даного виду протягом однієї години його використання;

T_j – річний фонд роботи даного обладнання, який дорівнює 1976 годинам.

3. В залежності від часу використання обладнання в процесі виготовлення продукту розраховується розмір амортизаційних відчислень, пов'язаних з виробництвом одиниці даного виду:

$$AM_j = AM_{j\text{год}} * t_j, \quad (5.7)$$

де AM_j – амортизація обладнання даного виду при виготовленні продукту;

t_j – час використання обладнання даного види при виготовленні продукту, який дорівнює 368 годинам.

Якщо використовувати різне обладнання, то величина амортизації як елемент собівартості продукту повинна бути визначена шляхом підсумовування амортизаційних відрахувань по окремих видах обладнання:

$$AM = \sum_{i=1}^m AM_i, \quad (5.8)$$

де AM – сума амортизації по продукту;

m – кількість видів обладнання.

Отже, за формулою (5.5) річна сума амортизаційних відрахувань для ноутбуку HP ENVY dv6:

$$AM_{\text{ноутрік}} = 44000 * 0,25 = 11000 \text{ (грн)}.$$

Тоді, за формулою (5.6) величина амортизаційних відрахувань на одну годину для ноутбуку HP ENVY dv6:

$$AM_{\text{ноутгод}} = 11000/1976 = 5,56 \text{ (грн)}.$$

За формулою (5.7) величина амортизаційних відрахувань залежно від часу використання ноутбуку HP ENVY dv6:

$$AM_{\text{ноут}} = 5,56 * 368 = 2046,08 \text{ (грн)}.$$

За формулою (5.8) сума амортизації по продукту:
в даному випадку сума амортизації складається з одного обладнання

$$AM = 2046,08 \text{ (грн)}.$$

Вартість технологічної електроенергії обчислюється за формулою:

$$S_{\text{Э}} = T_{\text{ар}} * T * W, \quad (5.9)$$

де $T_{\text{ар}}$ – тариф електроенергії за один кВт, рівний 2,01 грн.;

T – кількість годин роботи;

W – споживана технологічна потужність ($W = 0,032$ кВт).

Отже, за формулою (5.9) вартість технологічної електроенергії:

$$S_{\text{Э}} = 2,01 * 368 * 0,032 = 23,67 \text{ (грн)}.$$

Вартість освітлювальної електроенергії розраховується за формулою:

$$S = T_{\text{ар}} * T * W, \quad (5.10)$$

де W – споживана потужність освітлювальним прибором ($W = 0,1$ кВт).

Отже, за формулою (5.10) вартість освітлювальної електроенергії:

$$S = 2,01 * 368 * 0,1 = 74 \text{ (грн)}.$$

Вартість додаткових послуг розраховується за формулою:

$$S_{\text{ДодПосл}} = \sum_{i=1}^m S_{\text{посл}_i}, \quad (5.11)$$

де $S_{\text{посл}}$ – вартість однієї додаткової послуги;

m – кількість додаткових послуг.

Отже, за формулою (5.11) вартість додаткових послуг:

$$S_{\text{ДодПосл}} = 2900 \text{ (грн)}.$$

5.6. Розрахунок собівартості та повної вартості програмного продукту

Собівартість – це вартісна оцінка використовуваних в процесі виробництва продукції (робіт, послуг) природних ресурсів, сировини, матеріалів, палива, енергії, основних фондів, трудових ресурсів та інших витрат на її виробництво і реалізацію. Собівартість дорівнює сумі всіх витрат на розробку проекту і розраховується за формулою:

$$C = 3П + 3П_{\text{соц}} + AM + S_{\text{Э}} + S + S_{\text{ДодПосл}}, \quad (5.12)$$

Отже, з формули (5.12) собівартість становить:

$$C = 49800 + 10956 + 2046,08 + 23,67 + 74 + 2900 = 65800 \text{ (грн)}.$$

Промоніторивши ситуацію на ринку праці відповідних програмних продуктів, прибуток від розробленого програмного продукту (П) назначено:

$$П = C * 0,2 \quad (5.13)$$

Отже, з формули (5.13) прибуток становить:

$$П = 65800 * 0,2 = 13160 \text{ (грн)}.$$

Повна вартість розробленого програмного продукту без ПДВ розраховується за наступною формулою:

$$ПВ_{\text{безПДВ}} = (C + П), \quad (5.14)$$

де $ПВ_{\text{безПДВ}}$ – повна вартість розробленого програмного продукту без ПДВ.

Отже, за формулою (5.14) повна вартість розробленого програмного

продукту без ПДВ:

$$ПВ_{\text{безПДВ}} = 65800 + 13160 = 78960 \text{ (грн.)}$$

Повна вартість розробленого програмного продукту з ПДВ розраховується за наступною формулою:

$$ПВ_{\text{зПДВ}} = ПВ_{\text{безПДВ}} + Н_{\text{ПДВ}} * ПВ_{\text{безПДВ}}, \quad (5.15)$$

де $ПВ_{\text{зПДВ}}$ – повна вартість розробленого програмного продукту з ПДВ;

$Н_{\text{ПДВ}}$ – податок на додану вартість, який дорівнює 20%.

Отже, за формулою (5.15) повна вартість розробленого програмного продукту з ПДВ:

$$ПВ_{\text{зПДВ}} = 78960 + 78960 * 0,2 = 94752 \text{ (грн.)}$$

Калькуляційні розрахунки на розробку програмного забезпечення представлені в таблиці 5.6.

Таблиця 5.6 – Статті калькуляції на розробку програмного продукту

№	Стаття калькуляції	Витрати, грн.
1	Основна заробітна плата	41500
2	Додаткова заробітна плата	8300
3	Загальна заробітна плата	49800
4	Єдиний соціальний внесок	10956
5	Амортизація	2046,08
6	Вартість технологічної електроенергії	23,67

Продовження таблиці 5.6

7	Вартість освітлювальної електроенергії	74
8	Вартість інших матеріалів	900
9	Вартість послуг інтернету	2000
10	Загальна собівартість розробки	65800
11	Прибуток	13160
12	Повна вартість розробленого програмного продукту без ПДВ	78960
13	Повна вартість розробленого програмного продукту з ПДВ	94752

Економічний висновок

В даному розділі було:

1. розглянуто кілька організацій, які є ринком збуту для даного продукту;
2. визначені трудовитрати робіт створення програмного продукту;
3. призначені виконавці роботи, а також проведені розрахунки заробітної плати та соціального внеску, які склали 41500 грн. та 10956 грн. відповідно;
4. складено перелік робіт для кожного з виконавців;
5. складено перелік необхідного обладнання, для яких розраховані витрати на амортизацію у розмірі 2046,08 грн.;
6. проведений розрахунок вартості електроенергії: технологічної та освітлювальної. Витрати склали 23,67 грн. та 74 грн. відповідно.
7. проведені розрахунки собівартості (65800 грн.), прибутку (13160 грн.), повної вартості розробленого програмного продукту (94752 грн.).

ВИСНОВКИ

В даній роботі аналізується процес створення можливої автоматизованої системи для бронювання площ житлового та нежитлового фонду.

При аналізі запропонованої системи автоматизації результати показали, що:

- Мінімізувалися дії користувача для виконання даного процесу, а саме бронювання приміщення;
- Значно скоротився час для виконання даних дій;
- Автоматизація дозволяє швидко підібрати максимально близькі по критеріям приміщення для клієнта.
- Потрібно більше пам'яті та сучасні ЕОМ на сервері для обробки даних, ніж це було до автоматизації системи.

Аналіз в роботі дозволив зробити основні висновки, до яких можна віднести такі:

- актуальність роботи, яка в даний час є дуже важливою частиною на ринку праці;
- дерево проблем, містить такі проблеми:
 - Неефективність маніпулювання площ;
 - Перебір зайвої інформації;
 - Трата незапланованого часу;
 - Неактуальність в діях бронювання площі;
 - Можливість клієнтів уникання функціональності бронювання;
 - Системи не призначені для великої кількості запитів;
 - Можливості сбоїв системи від незапланованих завантажень;
 - Клієнти незадоволені наданими послугами;
- дерево цілей, посилаючись на дерево проблем, може бути представлено наступними під цілями:
 - Зручний інтерфейс без зайвих елементів;

- Виконання заданої цілі в один клік;
- Актуальність швидкого процесу;
- Окрема система для повного функціонування бронювання площ;
- своєчасно обробляються запити;
- Швидкі відповіді клієнтам;
- Зацікавленість робітників в користуванні даним сервісом;
- Швидка популярність даної системи;
- Система складається з чотирьох рівнів, для кожного з яких визначені функції та параметри;
- Знайдені середньгеометричне число властивостей і зв'язків на один елемент, які приблизно дорівнюють 1,49 та 2,6 відповідно;
- Проведена класифікація системи, а також наступні описи:
 - морфологічний, який описує структуру системи, а також властивості кожного елемента системи;
 - функціональний, в основі якого лежать складові частини системи, їх функції, вхідні та вихідні дані;
 - інформаційний, який надає аналіз властивостей і зв'язків кожного елемента системи.

Програмний продукт був створений, використовуючи всі необхідні етапи розробки: постановку задачі, проектування системи, кодування, тестування і налагодження. Особлива увага приділялася проектування системи, розробці специфікацій вимог і математичної моделі, як до найголовніших етапів, від яких залежало якість майбутньої системи.

Під час виконання завдання були закріплені і застосовані на практиці навички, отримані під час навчання. Були використані навички проектування інформаційних систем, закріплено знання з веб-технологій, баз даних, а також принципів ООП. Доповнення розроблено за допомогою технологій C#, Angular та архітектури MVC. Робота з базою даних Mongo здійснювалася за допомогою класів налаштувань у вигляді json файлів, в яких реалізована основна логіка роботи з базою даних. Проект дав можливість відчути особливості етапів роз-

робки, ознайомитися з системами контролю версій і отримати досвід рівномірного і раціонального розподілу часу.

Проведені економічні розрахунки дають повне уявлення по затратах на впровадження даного проекту. Прораховані розрахунки собівартості (65800 грн.), прибутку (13160 грн.), повної вартості розробленого програмного продукту (94752 грн.), що є цілком доцільним для виробництва такої продукції.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ:

1. Фрідман Л. М. Основи проблемології / Серія: Проблемологія. М: Синтег, 2001. 228 с..
2. Ожегов, Н.Ю. ЛІТЕРАТУРА А., Основи менеджменту. / А. Мескон, М. Альберт, Ф. М. Хедоурі.- Справа, 1994.- 708 с. С.І. Тлумачний словник російської мови: 80000 слів і фразеологічних висловів / С.І Шведова.- М.: АЗ', 1995.
3. Ускач А.Ф., Гогунський В.Д., Яковенко О.Є. Моделі завдання розподілу в теорії розкладу / Автоматика. Автоматизація. Електротехнічні комплекси та системи.-2006-№ 2 (18) .- С.98-104
4. Лях, Ю. В. Автоматизована система розподілу завдань між співробітниками організації / Ю. В. Лях // Інформаційні технології і управління: матеріали 50-ї наукової конференції аспірантів, магістрантів і студентів. (Мінськ, 24 - 28 березень 2014 року). - Мінськ: БДУІР, 2014. - С. 25 - 26.
5. ТОВ "Організатор ЛідерТаск" - незалежний розробник програмного забезпечення. [Електронний ресурс]: портал – Режим доступу: – <https://www.leadertask.ru/content/view/1026> , – вільний. – 09.11.2018.
6. Мамон, М. Мопона практиці /М. Мамон. – Аpress, 2010. – 424с.
7. [Електронний ресурс]: портал – Режим доступу: – <https://habr.com/company/angels-it/blog/148742/> , – вільний. – 21.05.2019.
8. Єрмолаєва В. В., Калашников Д. А. Автоматизовані системи управління // Молодий вчений. - 2016. - №11. - С. 166-168.
9. Вентцель О.С. Исследование операций, М:Наука, 1980.
10. Беляєв, Г.Т. Технічні засоби ФСУ ТП / Г.Т. Беляєв, В.Ф. Кузьміцький, Н.І. Смирнов. - М.: Изд-во МЕІ, 1983. - 420 с. Гнеденко Б.В., Коваленко И.Н. Введение в теорию массового обслуживания. М.: Наука, 1987.
11. Організація баз даних та знань / О. В. Тарасов, Л. А. Павленко, М. Ю. Лосев, В. В. Федько – Х.: Вид. ХНЕУ, 2013. – 83 с.

12. Федько В.В. Лабораторний практикум з модуля "Основи баз даних та знань" навчальної дисципліни "Організація баз даних та знань" / В. В. Федько, О. В. Тарасов, М. Ю. Лосев – Х.: Вид. ХНЕУ, 2011. – 192 с.
13. Программное обеспечение. [Электронный ресурс]: портал – Режим доступа: http://www.tadviser.ru/index.php/Статья:Программное_обеспечение, – вільний. – 18.05.2019.
14. Единый социальный взнос. [Электронный ресурс]: портал – Режим доступа: <http://index.minfin.com.ua/index/social/>, – вільний. – 29.05.2019.
15. Современный экономический словарь [Текст] / Б.А. Райзберг, Л.Ш. Лозовский, Е.Б. Стародубцева – 2-е изд., испр. М.: Инфра-М,1999. – 479 с.
16. Новый англо-русский словарь-справочник. Экономика.[Текст] /О.В. Сиполс – М.:Флинта,Сиполс,2010. – 126 с.
17. Макроэкономика. [Текст] / Р. Пиндайк, Д. Рубинфельд, «Дело», 2000.376-377 с.
18. С# 6.0. Справочник. Полное описание языка, 123-134с.
19. Албахари Бен, Албахари Джозеф. С# 7.0 in a Nutshell, 2018. –346 – 476 с.