

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ  
НАЦІОНАЛЬНИЙ АВІАЦІЙНИЙ УНІВЕРСИТЕТ

Кафедра комп'ютеризованих систем управління

ДОПУСТИТИ ДО ЗАХИСТУ

Завідувач кафедри

Литвиненко О.Є.

“ \_\_\_\_\_ ” \_\_\_\_\_ 2022 р.

# ДИПЛОМНИЙ ПРОЄКТ

(ПОЯСНЮВАЛЬНА ЗАПИСКА)

ВИПУСКНИКА ОСВІТНЬОГО СТУПЕНЯ

«БАКАЛАВР»

Тема: Інформаційний додаток реєстрації авіапасажирів в умовах карантину

Виконавець: \_\_\_\_\_ Маліцький О. О.

Керівник: \_\_\_\_\_ Апенько Н.В.

Нормоконтролер: \_\_\_\_\_ Тупота Є.В.

Київ 2022

# НАЦІОНАЛЬНИЙ АВІАЦІЙНИЙ УНІВЕРСИТЕТ

Факультет кібербезпеки, комп'ютерної та програмної інженерії

Кафедра комп'ютеризованих систем управління

Спеціальність 126 «Інформаційні системи та технології»

ЗАТВЕРДЖУЮ

Завідувач кафедри

Литвиненко О.Є.

«\_\_\_\_\_» \_\_\_\_\_ 2022 р.

## ЗАВДАННЯ

### на виконання дипломного проєкту

Маліцького Олександра Олексійовича

(прізвище, ім'я, по батькові випускника в родовому відмінку)

1. Тема дипломної роботи (проєкту): Інформаційний додаток реєстрації авіапасажирів в умовах карантину

затверджена наказом ректора від 15.04.2022 р. № 340/ст

2. Термін виконання роботи (проєкту): з 09.05.2022 р. по 06.06.2022 р.

3. Вихідні дані до роботи (проєкту): \_\_\_\_\_

4. Зміст пояснювальної записки:

1) Аналіз предметної області реєстрації авіапасажирів

2) Проектування та структура інформаційного додатку реєстрації пасажирів.

3) Розробка програмного забезпечення інформаційної системи

5. Перелік обов'язкового графічного (ілюстративного) матеріалу:

## 6. Календарний план-графік

№ пор	Завдання	Термін виконання	Відмітка про виконання
1	Провести аналіз літератури за темою дипломного проекту та аналіз існуючих систем	16.04.2022– 23.05.2022	
2	Визначити функціонал інформаційної системи. Підготувати текст другого розділу пояснювальної записки.	21.05.2022 – 25.05.2022	
3	Розробити структуру програмних засобів системи.	28.05.2022 – 01.06.2022	
4	Розробити програмні засоби та зробити підготовку до тестування	04.06.2022 – 08.06.2022	
5	Протестувати інформаційну систему.	11.06.2022 – 13.06.2022	
6	Написати пояснювальну записку. Підготувати презентацію та графічні матеріали.	14.06.2022– 19.06.2022	
7	Захист дипломного проекту	20.06.2022	

7. Дата видачі завдання: « 15 » квітня 2022 р.

Керівник дипломного проекту \_\_\_\_\_ Апенько Н.В.  
(підпис керівника) (П.І.Б.)

Завдання прийняв до виконання \_\_\_\_\_ Маліцький О.О.  
(підпис випускника) (П.І.Б.)

## РЕФЕРАТ

Пояснювальна записка до дипломного проекту «Інформаційний додаток реєстрації авіапасажирів в умовах карантину»: 41 с., 28 рис., 1 табл., 10 інформаційних джерел.

### ІНФОРМАЦІЙНІ ТЕХНОЛОГІЇ, ПРОГРАМУВАННЯ, ОНЛАЙН ДОДАТОК

**Об'єкт дослідження** – інформаційний додаток реєстрації авіапасажирів в умовах карантину.

**Предмет дослідження** – система реєстрації авіапасажирів.

**Мета дипломного проекту** – розробити програмно-технологічне рішення, яке допоможе інформувати авіапасажирів про реєстрацію в аеропорті під час карантину

**Прогнози та припущення щодо розвитку об'єкта дослідження** – запропоновані підходи до розробки інформаційних систем можуть бути застосовані при розробці систем, схожих за вимогами функціональності та надійності.

## ЗМІСТ

ВСТУП.....	6
Розділ 1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ РЕЄСТРАЦІЇ АВІАПАСАЖИРІВ...	8
1.1. Існуючі засоби для реєстрації пасажирів .....	8
1.2. Вибір мови програмування для розробки застусунку....	<b>Error! Bookmark not defined.</b>
1.3. Висновки до розділу .....	14
Розділ 2 ПРОЕКТУВАННЯ ТА СТРУКТУРА ІНФОРМАЦІЙНОГО ДОДАТКУ РЕЄСТРАЦІЇ ПАСАЖИРІВ.....	16
2.1. Інженерія вимог.....	16
2.2. Моделювання бізнес-процесів в системі.....	18
2.3. Висновки до розділу .....	24
Розділ 3 РОЗРОБКА ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ ІНФОРМАЦІЙНОЇ СИСТЕМИ .....	25
3.1. Опис інтерфейсу додатку .....	25
3.2. Опис методів програмування.....	26
3.3. Керівництво з використання для користувача .....	29
3.4. Тестування програмного продукту .....	36
3.5. Висновки до розділу .....	38
Висновки .....	39
Список використаних джерел .....	41

## ВСТУП

Авіатранспорт протягом багатьох років залишається популярним видом транспорту як серед українців, так і серед інших країн світу. За темпами розвитку повітряний транспорт, як наймолодший, випереджає всі інші види транспорту, водночас саме авіаційний транспорт має величезну роль в міжконтинентальних перевезеннях в порівнянні з іншими. Свою популярність літаки отримали саме завдяки ряду переваг, серед яких найважливішими та найсуттєвішими є швидкість, безпека, та комфорт. Завдяки цим перевагам та своїй популярності, авіакомпаніям необхідно впроваджувати нові технології для зручності та полегшення обслуговування великої кількості пасажирів.

З стрімким розвитком інформаційних технологій у XXI столітті величезна кількість операцій, які раніше проводилися лише вручну, інформації, яка існувала лише у фізичній формі, з прогресом вся інформація в світі переходить у цифровий простір. З розвитком технологій достатньо мати електронний пристрій з доступом до мережі Інтернет, і вся інформація з літературою, новинами може бути доступною в будь який час. Години очікування своєї черги, інформація яку потрібно було дізнатись фізично, велика кількість паперових книг не є проблемою з звичайним смартфоном з доступом до Інтернету, вся інформація з усього світу стає доступною.

По всьому світі авіакомпанії ведуть між собою боротьбу за покупців, адже чим більше переваг у компанії, чим зручніше та краще буде пасажирові, тим краще будуть продаватись квитки на рейси, ніж у конкурентів. У свою чергу кожна авіакомпанія з розвитком ІТ сфери шукає різні способи скористатися новими технологіями для покращення або полегшення процесів, як наприклад онлайн бронювання місць в транспорті, електронна черга. Конкуренція між компаніями та бажання зробити авіарейси більш доступними має значний вплив на всю сферу в цілому.

Сучасний стан розвитку інформаційних технологій та авіаційної сфери, значний вплив на весь світ та прогрес в цілому призупинила пандемія *COVID-19*. Ця пандемія має значний вплив на велику кількість сфер, які за допомогою *IT* спеціалістів можуть бути автоматизованими, тобто без контакту людини з людиною. Нові інформаційні засоби можуть стати дуже в пригоді для полегшення та покращення сфери в цілому.

Недостатня розробленість проблем усунення пандемії в аеропорті, проблема інформування пасажирів щодо існуючих карантинних заходів, низький рівень якості їх виконання не дозволяє аеропортам функціонувати згідно з світовими вимогами. Тому було вирішено розробити власну систему інформаційного додатку реєстрації пасажирів, та для зручності використання користувачам розробити її як *web*-додаток. Дане рішення визначає мету, завдання і зміст дипломного проектування.

Об'єкт дослідження – інформаційний додаток реєстрації пасажирів в умовах карантину.

Предмет дослідження – система реєстрації авіапасажирів.

Мета дипломного проекту – вдосконалення інформаційної бази аеропортів на основі інформування користувачів щодо рейсів та карантинних заходів.

Виходячи з поставленої мети, виділимо кілька завдань, які необхідно вирішити для досягнення цієї мети:

- аналіз предметної області реєстрації авіапасажирів;
- визначення вимог до технологічного рішення;
- проектування та структуризація додатку;
- розроблення додатку за обраною структурою, згідно з вимогами.

# РОЗДІЛ 1

## АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ РЕЄСТРАЦІЇ АВІАПАСАЖИРІВ

### 1.1. Існуючі засоби для реєстрації пасажирів

Авіапереліт завжди був статусною послугою. Спочатку авіакомпанії возили дорого і комфортно, згодом авіатранспорт оптимізувався і став включати набір стандартних послуг щоб стати більш доступними для кожної людини, та все одно залишався одним з найдорожчих видів перевезень.

Кожен рік авіаперельоти стають все більш доступними. Колись більшість людей переважно літали тільки в якісь сусідні країни, та на даний момент такий транспорт є популярним навіть при необхідності відправитись до сусіднього міста. З розвитком технологій сам авіатранспорт став відрізнятись від більшості інших способів транспорту не тільки тим, що забезпечує найбільш комфортні умови, але й завдяки доступним цінам.

Літак – це найбільш безпечний вид транспорту на сьогоднішній день. Це демонструє, в першу чергу, статистика, адже катастрофи трапляються дуже рідко, в той час такі способи як автомобільний транспорт або морський відрізняються набагато більшим процентом нещасних випадків.

На цей час щоб замовити собі квиток, достатньо лише знайти необхідний маршрут, дізнатись необхідну інформацію про сам рейс, забронювати місце на політ за допомогою мобільного зв'язку, онлайн реєстрації, або ж в самому аеропорті, після чого відправитись до аеропорту в назначений час

Кафедра КСУ				НАУ 22 05 01 000 ПЗ			
Виконав	Маліцький О.О.			АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ РЕЄСТРАЦІЇ АВІАПАСАЖИРІВ	Літера	Лист	Листів
Керівник	Апенько Н.В.					9	41
Консульт.					ІТ-461 126		
Н - контр.	Тупота Є.В.						
Зав. каф.	Литвиненко О.Є.						



## Мобільна реєстрація

Однією з існуючих засобів реєстрації авіапасажирів сьогодні є мобільна реєстрація. Мобільна реєстрація здійснюється за допомогою мобільного додатку від самої авіакомпанії. Це простий і практичний спосіб для бронювання квитків, адже в додатку існують функції сканування документів та бронювання квитків, з подальшим друком посадкового талону або підтвердження рейсу. Прикладом може бути додаток компанії *WizzAir* на безкоштовній основі

Приклад мобільної реєстрації зображений на Рис1.1.

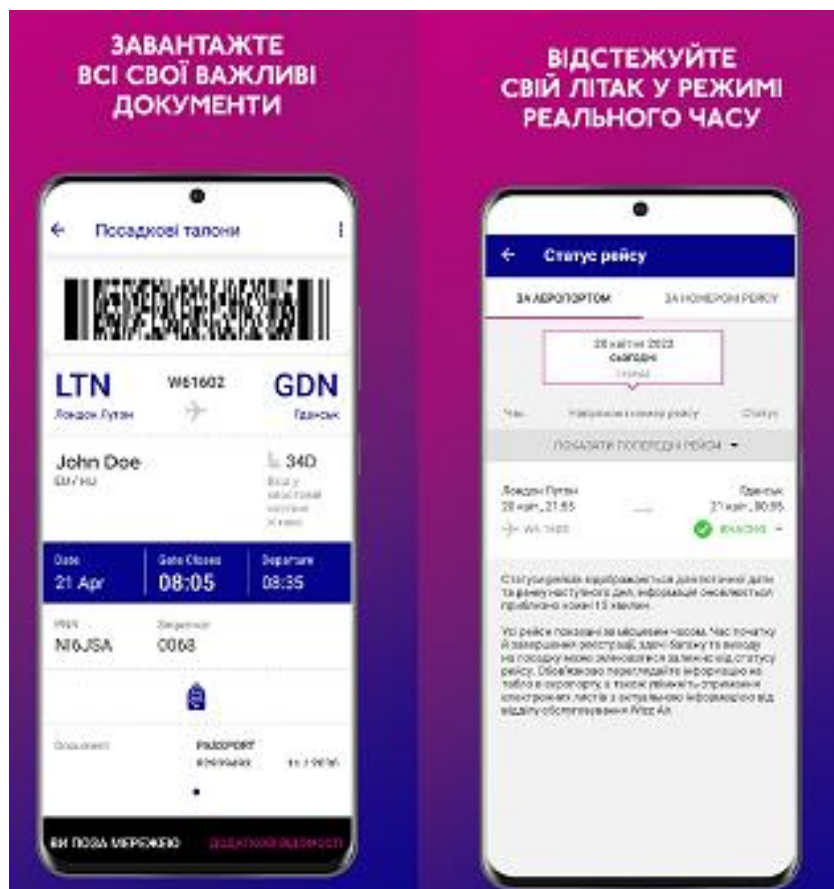


Рис1.1 Додаток авіакомпанії *WizzAir*

Головною проблемою мобільної реєстрації та її невеликої популярності є необхідність користувача завантажити саму програму для подальшого використання, адже з існуючими аналогами набагато зручніше зареєструватись через онлайн сайт аеропорту, або ж самостійно за стійкою реєстрації в аеропорті. Сама система мобільного додатку час від часу потребує вдосконалення інтерфейсу,

технічного обслуговування, користувачеві не завжди буде зручно обновлювати програму щоб зареєструватись на рейс.

### Реєстрація в аеропорті

Реєстрація в самому аеропорті на даний момент є самою популярною процедурою реєстрації пасажирів. Цей спосіб є самим звичним для реєстрації, адже він існує набагато довше ніж мобільні додатки, або ж онлайн реєстрація.

Але цей спосіб не є найкращим, через популярність існує велика ймовірність черг в самому аеропорті, а зважаючи на пандемію, цей спосіб реєстрації не є раціональним. Авіакомпанії зазвичай користуються службовими стойками в залі аеропортів, сама ж реєстрація пасажира здійснюється авіакомпанією або ж хендлінговим агентом від імені перевізника.

Приклад службових стійок для реєстрації представлений на рис 1.2



Рис 1.2 Службові стійки реєстрації в аеропорті

Основними недоліками реєстрації в аеропорті є час очікування на реєстрацію в аеропорті, яка починається за 2-3 години до вильоту, велика ймовірність черги

при реєстрації, сам процес реєстрації від 15 хвилин до 4 годин залежно від аеропорту призначення та авіакомпанії.

### Онлайн реєстрація

Одна з найпопулярніших і найпростіших видів реєстрації авіапасажирів на сьогодні. Цей спосіб суттєво зберігає час відведений на посадку в аеропорті. Онлайн-реєстрація передбачає процедуру підтвердження пасажиром своєї присутності на рейсі через інтернет та самостійного роздрукування своїх посадкових талонів. Залежно від перевізника та типу рейсу на цьому етапі можна вибрати обсяг багажу та конкретне місце. Авіакомпанії активно рекомендують таку послугу, оскільки вона легша та швидша, заощаджує час як пасажирів, так і сервісних працівників. Плюс за допомогою реєстрації в форматі онлайн пасажирів передбачені окремі стійки, де черг немає, або кількість людей значно менше. Онлайн реєстрація доступна раніше, ніж в самому аеропорті, і всю необхідну інформацію також можна дізнатись за допомогою онлайн-ресурсів.

Для онлайн реєстрації пасажирів знадобиться номер бронювання (PNR-номер) або номер електронного квитка. В онлайн форматі отримати доступ до бронювання, використовуючи номер бронювання або електронного квитка, та персональні дані. Після онлайн реєстрації пасажирів необхідно роздрукувати посадковий талон або номер бронювання, і пред'явити посадковий талон і паспорт в самому аеропорті.

Прикладом для онлайн реєстрації може бути вебсайт компанії *RyanAir* представлений на Рис 1.3

Discover The Best Deals from Wizz Air Within The Next Month To Lisbon

The screenshot shows a flight search interface for Ryan Air. At the top, there's a header: "Discover The Best Deals from Wizz Air Within The Next Month To Lisbon". Below this are two input fields: "FROM" with a placeholder "Input origin" and a dropdown arrow, and "TO" with "Lisbon (LIS)" and a close button. Below these are five flight cards, each with a route, dates, price, and a "Book Now" button.

Route	Dates	Price	Seen	Round-trip
Budapest (BUD) to Lisbon (LIS)	08 Jun 2022 - 15 Jun 2022	HUF68,780*	4 hrs ago	Yes
Warsaw (WAW) to Lisbon (LIS)	27 Jun 2022 - 12 Jul 2022	PLN1,128*	1 day ago	Yes
Bucharest (OTP) to Lisbon (LIS)	25 Jun 2022 - 05 Jul 2022	RON1,028*	2 hrs ago	Yes
Sofia (SOF) to Lisbon (LIS)	13 Jun 2022 - 20 Jun 2022	BGN333*	3 hrs ago	Yes
London (LTN) to Lisbon (LIS)	22 Jun 2022 - 25 Jul 2022	GBP93*	1 day ago	Yes

Рис1.3 Сайт авіакомпанії *Ryan Air*

Зважаючи на те що онлайн реєстрація є найпростішою, зберігає час як пасажирові так і сервісних працівників, зважаючи на існуючу проблему пандемії *COVID-19* у світі, та для її інформування пасажирів було вирішено створити інформаційний додаток реєстрації пасажирів в умовах карантину.

Цей додаток має забезпечити пасажира всією необхідною інформацією для успішного використання послуг аеропорту, для усунення можливих проблем реєстрації користувача, та дотримання карантинних заходів в аеропорті.

## 1.2 Вибір мови програмування для розробки застосування

Для розробки інформаційного додатку було обрано мову *C#* на базі платформи *.NET Framework*, а саме *ASP.NET*. Ця мова використовує об'єктно-орієнтований підхід до програмування. Об'єктно-орієнтований підхід користується великою популярністю, тому що дозволяє не запам'ятовувати всю інформацію, а працювати з вхідними даними. Можливість використання готовою конструкції, а компілятор виконає всю роботу. Зручність читання коду та висока швидкість

розробки перебиває мінуси меншої продуктивності, порівняно з іншими мовами програмування. Велика кількість бібліотек та шаблонів є ще одним плюсом для вибору мови програмування, що дозволяє менше витратити час на прописання шаблону. До плюсів можна віднести велику кількість навчального та довідкового матеріалу, практично всі відповіді на можливі запитання можна знайти в документації, або в довідкових збірках.

Платформа *.Net Framework* користується великою популярністю серед програмістів, оскільки рівних цій мові практично немає. Використовуючи *.NET Framework*, можна розробляти будь-які програми, оскільки вона забезпечує сумісність різних служб, написаних на різних мовах і здатна усунути можливі конфлікти між різними версіями. Основними перевагами фреймворку можна виділити :

- проста та дієва система хешування;
- кросплатформенність;
- велика кількість бібліотек та шаблонів;
- велика кількість користувачів;
- нескладне обслуговування.

В розробці інформаційного додатку буде використано кросплатформенний фреймворк *ASP.NET CORE* для створення веб-додатків на платформі *.NET* з відкритим вихідним кодом (*open source software*), тобто доступний код для перегляду, вивчення та редагування, щоб переконатися в відсутності вразливостей для користувача, або ж може бути своєю документацією.

*ASP.NET Core* дозволяє:

- створювати веб програми та служби;
- використовувати засоби розробки у *Windows, macOS, Linux*;
- переглядати продукт в хмарі або в локальному середовищі.

Сам *ASP.NET Core* це модифікація старішого фреймворку *ASP.NET 4x*, з архітектурними змінами, що формують більш раціональну та модульну платформу. Цей фреймворк може легко інтегруватися з поширеними клієнтськими бібліотеками та платформами, у тому числі *Angular, React, Bootstrap*.

Серед основних переваг, через які цей фреймворк обирають програмісти по всьому світу можна зазначити:

- можливість запуску на *Windows, macOS, Linux*;
- розроблено для тестування;
- відкритий вихідний код;
- можливість спільного використання серверної та клієнтської логіки додатків, написаних на *.NET*;
- інструментарій, що спрощує процес веб-розробки;
- високі показники продуктивності;

Також була обрана мова *SQL* – це діалогова мова програмування для взаємодії користувача з базами даних, що застосовується для формування запитів, створення схем бази даних, модифікації, оновлення і керування базами даних. Сама по собі *SQL* не є мовою програмування або окремим продуктом. Будучи вбудованою в прикладні програми вона може виступати як інструкція для керуванням даних. Інтегрованою програмою вона має функції зміни, перевірки, та захисту даних.

Для розгляду бази даних та процесу корегування процесом створення самої БД, було обрано СУБД *Microsoft SQL Server*, розроблена корпорацією *Microsoft*, як сервер даних для збереження даних та надання даних у відповідь на запити інших застосунків, які можуть виконуватись на сервері, або в мережі.

*Microsoft SQL Server* як мову запитів використовує версію *SQL*, що отримала назву *Transact-SQL (T-SQL)*, яка є реалізацією *SQL-92*. Також *T-SQL* дозволяє використовувати додатковий синтаксис процедур.

Користуючись нею я створив базу даних до якої будуть занесені дані про пасажирів: логін, пароль, персональні дані; та інформацію про рейси.

### 1.3 Висновки до розділу

У даному розділі був проведений аналіз предметної області реєстрації авіапасажирів. Були досліджені компоненти реєстрації, сучасні технології, та

описані їх параметри. Було обґрунтовано використання онлайн реєстрації пасажирів у порівнянні з іншими існуючими системами. Аналіз показав, що інформаційний додаток реєстрації пасажирів є ефективною та якісною послугою для пасажирів.

Була проаналізована та обрана мова програмування *C#*, платформою *.NET Framework*. Було з'ясовано що *ASP.NET Core* є кращим вибором для розробки інформаційного додатку через велику кількість переваг. Було обрано базу даних, а саме СУБД *Microsoft SQL Server*, для зручності управління середніх за розміром баз даних, та через зручний функціонал.

РОЗДІЛ 2  
ПРОЕКТУВАННЯ ТА СТРУКТУРА ІНФОРМАЦІЙНОГО ДОДАТКУ  
РЕЄСТРАЦІЇ ПАСАЖИРІВ

2.1. Інженерія вимог

Інженерія вимог – це розділ інженерії програмного забезпечення, який включає в себе збір та опрацювання вимог до розроблювальної програмної системи, визначення основних цілей функцій, розробка алгоритмів та характеристик, що стосуються програмного додатку, займається документуванням та валідацією поставлених вимог. Самі ж вимоги – це твердження стосовно характеристик та властивостей, які повинен мати програмний продукт, які виникають в процесі аналізу та розробки.

Вимоги проходять чотири фази розробки:

- збір необхідної інформації щодо виявлення вимог;
- аналіз знайдених вимог, формування та встановлення часу, їх пріоритетів;
- документувація поставлених вимог;
- специфікації знайдених вимог.

Основними проаналізованими вимогами до програмного продукту мають бути: вимоги користувачів, функціональні вимоги, нефункціональні вимоги, та операційні вимоги.

Вимоги користувачів

Процес вивчення потреб та цілей користувача, класифікація та перетворення їх на вимоги до системи апаратури та ПЗ, встановлення та вирішення конфліктів між вимогами визначення пріоритетів, визначення принципів взаємодії зі середою функціонування.

Кафедра КСУ				НАУ 22 05 01 000 ПЗ			
<i>Виконав</i>	Малицький О.О.			ПРОЕКТУВАННЯ ТА СТРУКТУРА ІНФОРМАЦІЙНОГО ДОДАТКУ РЕЄСТРАЦІЇ ПАСАЖИРІВ	<i>Літера</i>	<i>Аркуш</i>	<i>Аркушів</i>
<i>Керівник</i>	Апенько Н.В.					16	41
<i>Консульт.</i>					ІТ-461 126		
<i>Норм. контр.</i>	Тупота С. В.						
<i>Зав. Каф.</i>	Литвиненко О.Є.						



Такі вимоги мають стосуватися лише зовнішньої поведінки програмного продукту.

Однією з форм відображення вимог користувачів є діаграма прецедентів. Діаграма прецедентів зображує всі можливі взаємодії, які можуть виникнути між користувачем та програмним продуктом. Сама діаграма відображає, які можливості має користувач при роботі з інформаційним додатком.

Діаграма прецедентів зображена на рис 2.1



Рис 2.1 Діаграма прецедентів

### Функціональні вимоги

Функціональні вимоги – це вимоги, що визначають задачі та функції, які має реалізувати розробник програмної системи, відповідають за функціонал продукту. Опис функціональних вимог полягає в структурованому списку функцій та поведінку системи в процесі реакції на них.

Сам інформаційний додаток має відповідати наступним функціональним вимогам:

- можливість реєстрації та аутентифікації користувачів;
- збереження даних авторизації;
- редагування особистої інформації;

- коректне виведення даних про рейс;
- представлення інформації про карантинні заходи;
- пошук рейсів.

#### Нефункціональні вимоги

Вимоги, які є спрямованими на опис інтерфейсу, атрибутів, характеристик та властивостей програмної системи. Нефункціональні вимоги визначають якою системою вона має бути, що повинна виконувати. Нефункціональні вимоги визначаються ще на першій стадії розробки, на етапі аналізу вимог. Серед нефункціональних вимог можна виділити основні - вимоги до інтерфейсу, апаратні вимоги та операційні вимоги.

При розробці інформаційного додатку важливо дотримуватись:

- простоти та зручності інтерфейсу;
- доступності елементів продукту;
- збереження цілісності даних;
- вчасного виправлення можливих помилок.

## 2.2 Моделювання бізнес-процесів в системі

Для того, щоб краще розібратися в процесі поведінки системи активно використовувалася схеми алгоритмів, за допомогою яких були складені діаграми видів діяльності, діаграми взаємодії, діаграма станів, на основі яких відбувається подальша програмна реалізація системи. Щоб описати послідовності діяльності, які являють собою певний стан, в якому щось виконується, найкраще використовувати діаграми видів діяльності. Дана діаграма встановлює основні правила послідовності дій, які необхідно дотримуватися в тій чи іншій ситуації. Діаграми видів діяльності широко використовують у проектуванні. Графи видів діяльності можна використовувати для демонстрації різних рівнів деталізації проектування

Діяльність може містити вхідні та/або вихідні дуги діяльності, що показують потоки управління та потоки даних.

Якщо потік з'єднає дві діяльності, він є потоком управління. Якщо потік закінчується об'єктом, він є потоком даних.

Модель входу користувача в систему представлена на рис. 2.2.

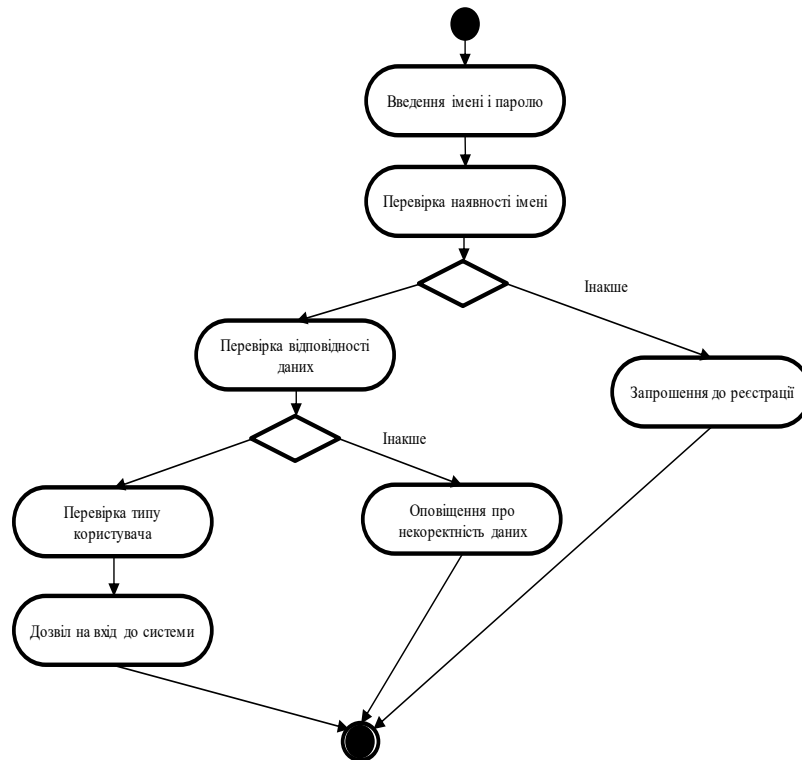


Рис 2.2 Модель входу користувача в систему

Наступною схемою буде модель реєстрації користувача в системі, представлена на рис 2.3.

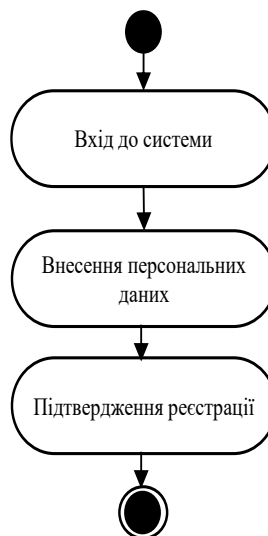


Рис 2.3 Реєстрація користувача в системі

На рис 2.4 зображена схема алгоритму реєстрації користувача на рейс

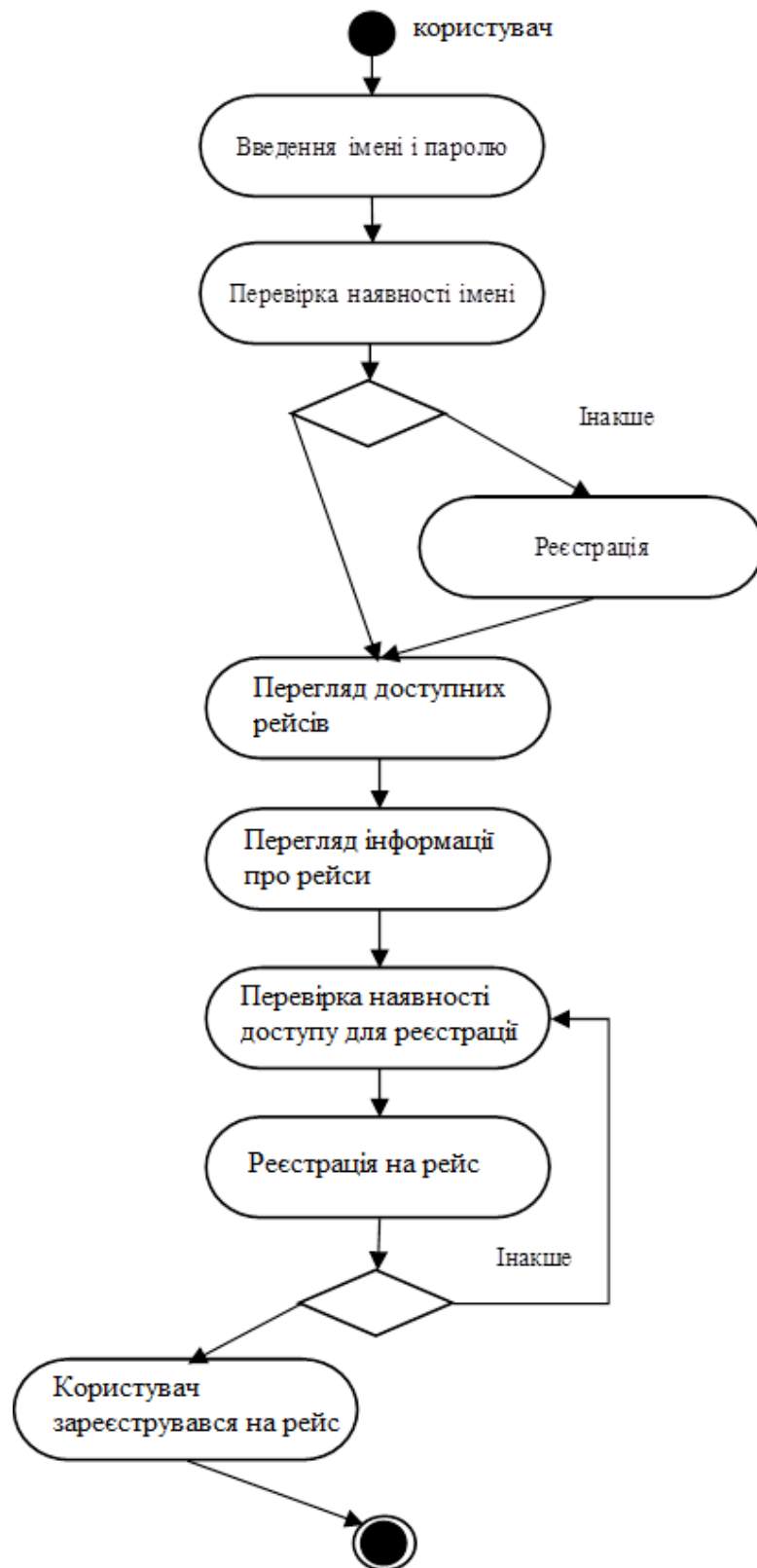


Рис 2.4 Діаграма реєстрації користувача на рейс

Моделі взаємодії використовуються на більш розвинених стадіях аналізу.

Діаграма послідовності, що дозволить нам побачити повідомлення, властивих поведінки деякої системи, якими обмінюються об'єкти системи відповідно до встановлених між ними зв'язками.

Діаграма послідовності входу користувача в систему представлена на рис. 2.5.

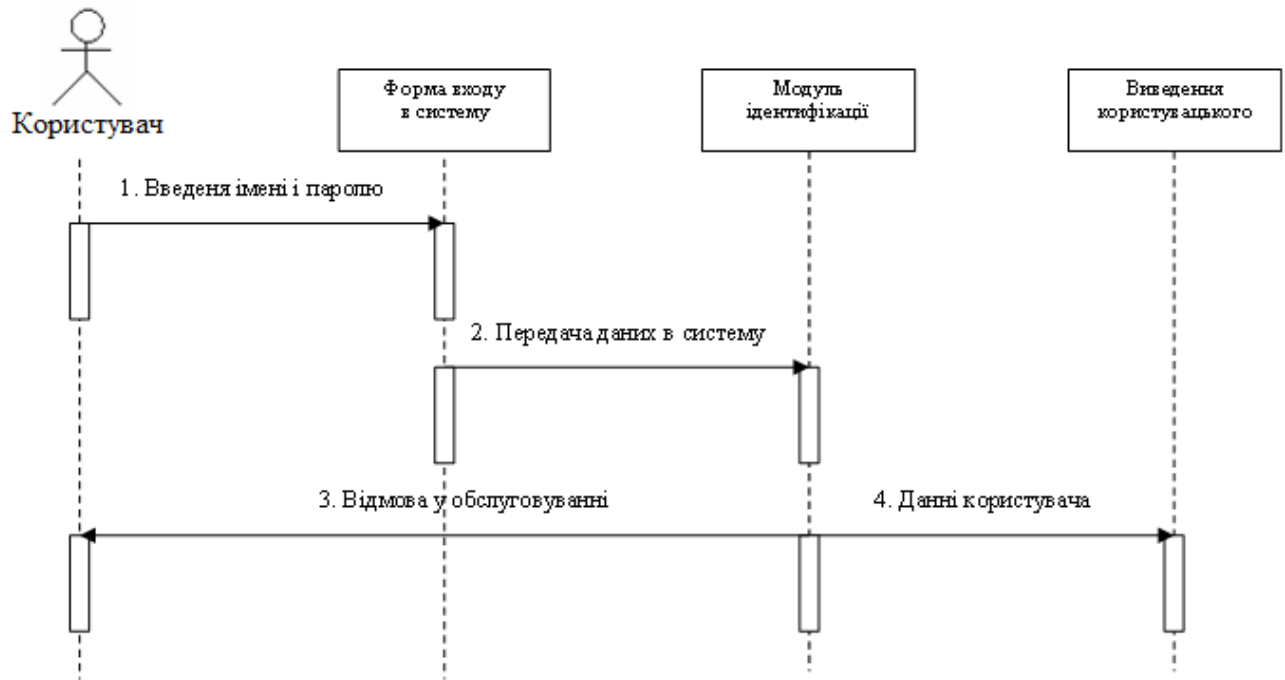


Рис 2.5 Послідовність входу користувача в систему

Для описання змін станів в системі буде використана система станів. Оскільки діаграма станів визначає для кожного стану об'єкта дію, яке виконує об'єктом під час одержання ним нагадування про подію. Один і той же об'єкт може виконувати різні дії у відповідь на одну і ту ж подію в залежності від стану об'єкта.

На рис. 2.6 представлена діаграма станів процесів реєстрації пасажира на рейс.

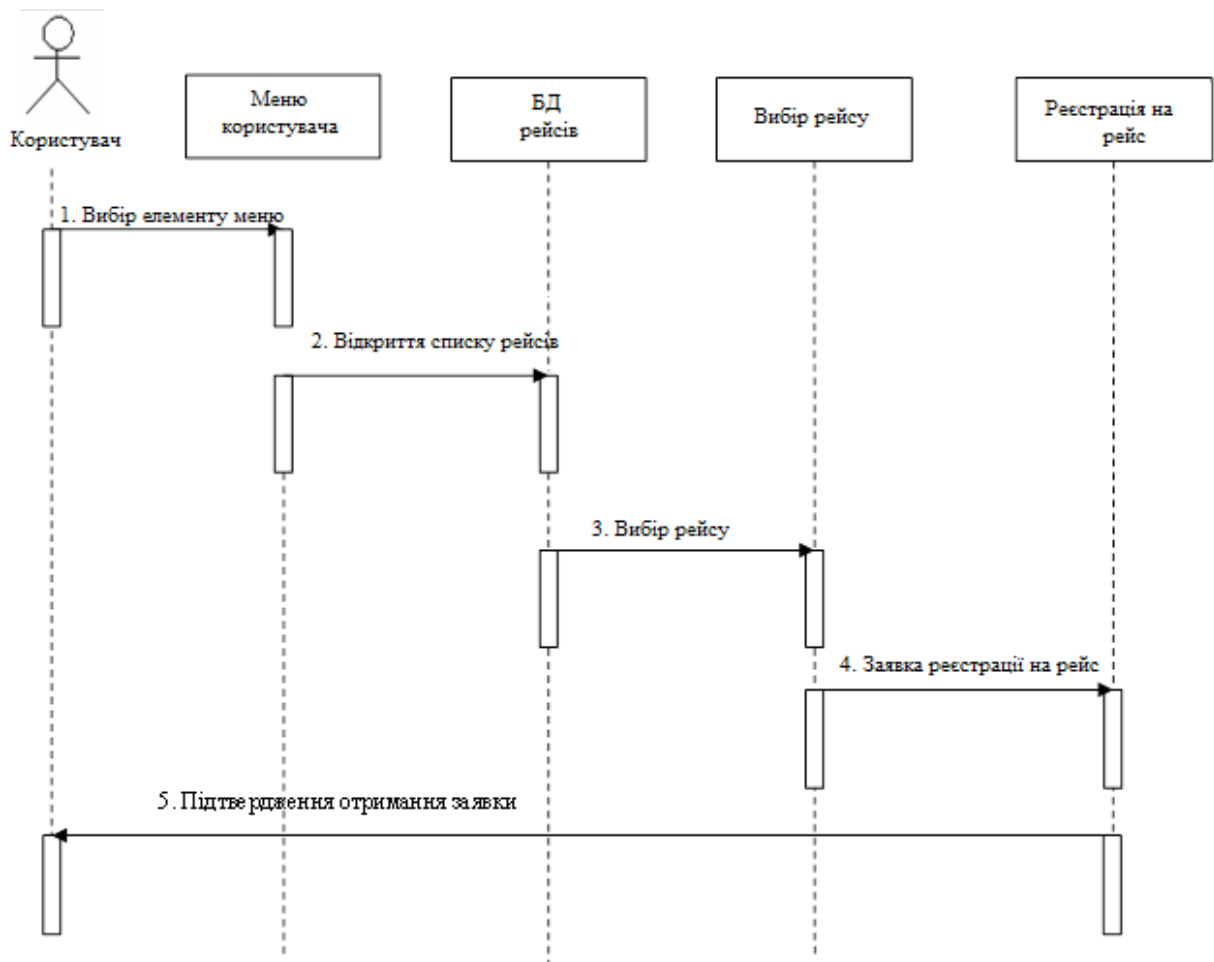


Рис. 2.6. Діаграма станів реєстрації користувача

### Структурна схема процесів системи

Структурна схема призначена для відображення загальної структури пристрою, тобто його основних блоків, вузлів, частин та головних зв'язків між ними. Завдяки структурній схемі буде зрозуміло навіщо потрібний даний веб-застосунок і як він працює.

Прикладом структурної схеми була розроблена карта сайту, для зручнішого користування користувачами, рис 2.7.

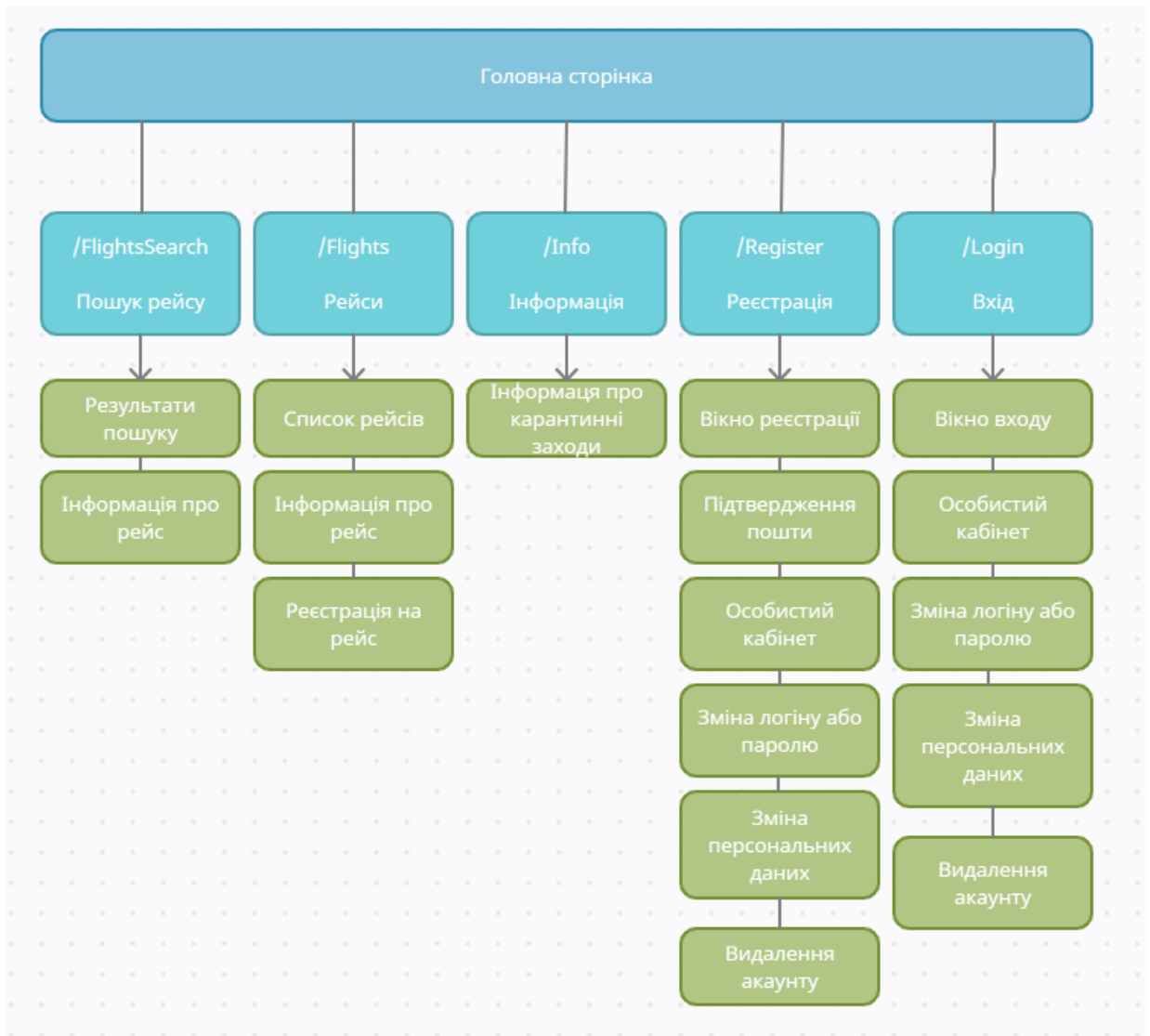


Рис 2.7 Карта сайту

На карті сайту зображено функціонал та зв'язки між основними блоками, та їх взаємодія.

Головна сторінка – (домашня сторінка) початкова сторінка веб-сайту, з відомостями про тематику сайту, відображає п'ять основних блоків, з якими зв'язується:

1. Пошук рейсу – сторінка з пошуком необхідного рейсу, можливий пошук за допомогою номера рейсу, часом відправлення, та місцем прибуття.

Результати пошуку – сторінка з знайденими в базі даних рейсами, відображення їх для користувача.

Інформація про рейс – сторінка з інформацією про обраний рейс.

2. Список рейсів – сторінка з повним списком доступних користувачеві рейсів, з відображенням інформації про рейси та можливістю реєстрації.

Реєстрація на рейс – сторінка з відображенням наявності рейсу, та можливості реєстрації в черзі на рейс.

3. Інформація – сторінка з інформацією для пасажирів.

Інформація про карантинні заходи – сторінка з відображенням інформації щодо карантинних заходів, правил щодо реєстрації.

4. Реєстрація – сторінка для реєстрації нових користувачів.

Вікно реєстрації – процес реєстрації нових користувачів у систему.

Підтвердження пошти – сторінка з підтвердженням користувача електронної пошти.

Особистий кабінет – сторінка після реєстрації користувача до системи.

Зміна логіну або паролю – сторінка з зміною користувачем свого логіну або паролю.

Зміна персональних даних – сторінка з редагуванням даних користувача, функцією додавання номера телефону.

Видалення акаунту – сторінка з можливістю видалення існуючого акаунту з бази даних.

5. Вхід – сторінка для входу користувачів в систему.

Особистий кабінет – сторінка після входу користувача до системи.

### 2.3 Висновки до розділу

В даному розділі було побудовано модель функціонування системи та на основі *case*-діаграм, проведено проектування та моделювання системи. Зображена діаграма прецедентів, моделі входу та реєстрації пасажирів в систему. Розібрано алгоритм реєстрації користувача на рейс, послідовність входу користувача в систему та розписана карта сайту.



РОЗДІЛ 3  
РОЗРОБКА ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ  
ІНФОРМАЦІЙНОЇ СИСТЕМИ

3.1. Опис інтерфейсу додатку

Важливим етапом є вибір мови реалізації. Для *Internet* -програмування в наш час існує широкий спектр мовних засобів і питання вибору правильної мовної основи є одним з найактуальніших. Мова повинна забезпечувати розробнику максимум необхідного для написання функціоналу системи, і побудови інтерфейсу системи, він повинен забезпечити закінченому проекту достатню швидкодію, та усунути можливість існування проблем користувача при експлуатації написаного додатк. Після зважування всіх «за» і «проти» вирішено було зупинитися на мові *CSS* в силу наступних причин:

- дана мова призначена для написання *Internet*-додатків;
- *CSS* має вбудований набір функцій, що дозволяють реалізувати захист даних на сервері і аутентифікацію користувачів;
- покращені стилі для *HTML*;
- простота обслуговування.

Далі необхідно було обрати систему управління базами даних, зваживши всі «переваги» і «недоліки», мною було обрана *СУБД Microsoft SQL Server*.

Кафедра КСУ				НАУ 22 05 01 000 ПЗ			
<i>Виконав</i>	Малицький О.О.			РОЗРОБКА ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ ІНФОРМАЦІЙНОЇ СИСТЕМИ	<i>Літера</i>	<i>Аркуш</i>	<i>Аркушів</i>
<i>Керівник</i>	Апенько Н.В.					25	41
<i>Консульт.</i>					ІТ-461 126		
<i>Норм. контр.</i>	Тупота С. В.						
<i>Зав. Каф.</i>	Литвиненко О.Є.						

Дана СУБД має низку істотних переваг:

- наявність реалізацій під різні платформи операційних систем;
- наявність сучасних засобів підтримки цілісності даних таких як тригери, транзакції, констрейнтів;
- високі показники продуктивності;
- наявність можливостей роботи з *web* -додатками.

### 3.2 Опис методів програмування

Для запису даних про рейси, класи будуть описуватись за допомогою мови програмування *C#*. В Прикладі на рис 3.1 зображено публічний клас «Рейси», в якому є публічні змінні які зберігають зміст та текст про рейси.

```
public class flights
{
    Ссылка: 9
    public int ID { get; set; }
    Ссылка: 10
    public string Flight { get; set; }
    Ссылка: 9
    public string Time { get; set; }
    Ссылка: 9
    public string Appointment { get; set; }
    Ссылка: 10
    public string Carrier { get; set; }
    Ссылка: 9
    public string Terminal { get; set; }
    Ссылка: 9
    public string Status { get; set; }
    Ссылка: 0
    public flights()
    {
```

Рис 3.1 Клас сутності «Рейси»

В даному класі міститься всі інформація про рейс, а саме: час, прибуття, перевізник, термінал, та статус самого рейсу.

Опис класів рівня доступу до даних:

- *ID* – номер користувача;

- *User* – клас з даними користувача, такими як ім'я та прізвище, пошта, логін,
- *Password* – клас з хешованим паролем користувача та посиланням на відповідного користувача;
- *Flights* – клас з даними про рейси, з інформацією про рейс, час, прибуття, перевізник, термінал, та статус самого рейсу.

Для відображення рейсів з бази даних використовується функція `View _context.flights`, зображена на рисунку 3.2.

```
public async Task<IActionResult> Index()
{
    return _context.flights != null ?
        View(await _context.flights.ToListAsync()) :
        Problem("ApplicationDbContext.flights is null.");
}
```

Рис 3.2 Перегляд рейсів з БД

Для пошуку рейсів з бази даних використовується функція пошуку запити по всім рейсам `_context.flights`, зображена на рис 3.3.

```
public async Task<IActionResult> ShowFlightsResults(string SearchFlight)
{
    return View("Index", await _context.flights.Where( f=>f.Flight.Contains(SearchFlight)).ToListAsync());
}
```

Рис 3.3 Пошук рейсів в БД

Для перегляду інформації про рейси з бази даних, використовується *Details* про необхідний *рес*, зображений на рис 3.4.

```

public async Task<IActionResult> Details(int? id)
{
    if (id == null || _context.flights == null)
    {
        return NotFound();
    }

    var flights = await _context.flights
        .FirstOrDefaultAsync(m => m.ID == id);
    if (flights == null)
    {
        return NotFound();
    }

    return View(flights);
}

```

Рис 3.4 Відображення інформації про рейс

При створенні нової програми модель даних часто змінюється, і в кожному випадку використовується синхронізація з базою даних. При кожному використанні моделі даних – додаванні, видаленні або використанні класів сутностей або використанні класу *DbContext* –можемо видалити базу даних, після чого *EF* створює нову, яка відповідає даним моделям, та виконує її тести з даними.

Для збереження даних користувачів в базі даних було використано *migration name*

Цей спосіб для синхронізації бази даних з моделлю даних добре працює до розгортання програми в робочому середовищі. Коли програма виконується в робочому середовищі, вона зазвичай зберігає дані, які необхідно зберегти, і небажано втрачати все при кожній зміні, наприклад, додавання нового стовпця. Функція міграцій *EF Core* вирішує цю проблему, дозволяючи *EF* оновити схему бази даних замість створення бази даних.

“Users”, відображенням імені, пошти, номеру телефону, зображено на рис 3.5.

```

name: "AspNetUsers",
columns: table => new
{
    Id = table.Column<string>(nullable: false),
    UserName = table.Column<string>(maxLength: 256, nullable: true),
    NormalizedUserName = table.Column<string>(maxLength: 256, nullable: true),
    Email = table.Column<string>(maxLength: 256, nullable: true),
    EmailConfirmed = table.Column<bool>(nullable: false),
    PasswordHash = table.Column<string>(nullable: true),
    PhoneNumber = table.Column<string>(nullable: true),
    PhoneNumberConfirmed = table.Column<bool>(nullable: false),
    LockoutEnd = table.Column<DateTimeOffset>(nullable: true),
},

```

Рис 3.5 Збереження інформації про користувачів

Для збереження даних про рейсів в базі даних було використано migration name "Flights", з відображенням інформації про рейси, зображено на рис 3.6.

```

name: "flights",
columns: table => new
{
    ID = table.Column<int>(type: "int", nullable: false)
        .Annotation("SqlServer:Identity", "1, 1"),
    Flight = table.Column<string>(type: "nvarchar(max)", nullable: false),
    Time = table.Column<string>(type: "nvarchar(max)", nullable: false),
    Appointment = table.Column<string>(type: "nvarchar(max)", nullable: false),
    Carrier = table.Column<string>(type: "nvarchar(max)", nullable: false),
    Terminal = table.Column<string>(type: "nvarchar(max)", nullable: false),
    Status = table.Column<string>(type: "nvarchar(max)", nullable: false)
},
constraints: table =>
{
    table.PrimaryKey("PK_flights", x => x.ID);
});

```

Рис 3.6 Збереження інформації про рейс

Інтерфейс програми був розроблений як *web*-застосунок за допомогою *ASP.NET CORE* у *Microsoft Visual Studio 2022*.

### 3.3 Керівництво з використання для користувача

Для зручного користування додатком було вирішено розробити керівництво для використання, в якому розписані всі можливі дії з інтерфейсом додатку.

Після входу до *web*-застосунку за необхідним посиланням, інтерфейс додатку зустріне користувача головною сторінкою, з інформацією про необхідність реєстрації для подальшого перегляду списків рейсів, інформації про рейси та про карантинні заходи.

На рис 3.7 представлено вікно з головною сторінкою

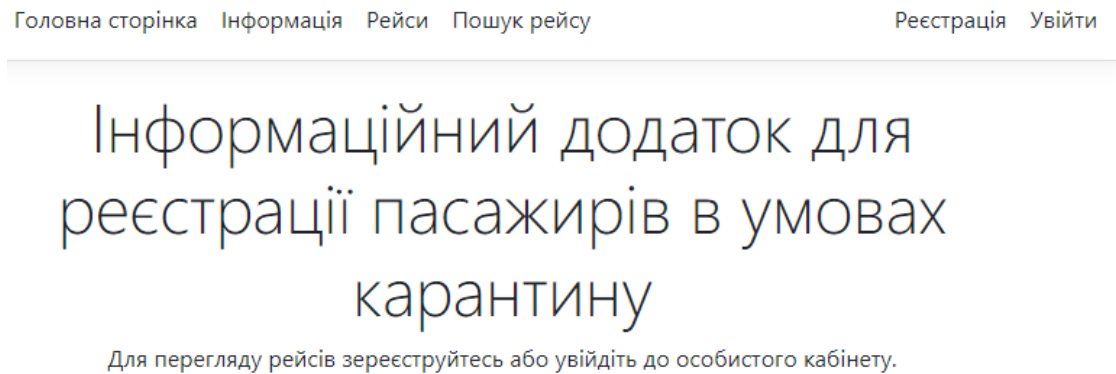


Рис 3.7 Головна сторінка додатку

На рис 3.8-3.9 представлено вікно з реєстрацією або входом до особистого кабінету.

## Реєстрація

### Створити новий акаунт.

Рис 3.8 Реєстрація до особистого кабінету

# Увійти

Використовуйте  
локальний обліковий  
запис для входу.

Електронна пошта  
hanman.man@ukr.net

Пароль  
••••••••

Пам'ятай мене?

**Увійти**

[Забули свій пароль?](#)

Рис 3.9 Вхід до особистого кабінету

Після реєстрації або входу до системи користувачеві будуть доступні вкладки з рейсами, інформацією про карантинні заходи, можливість редагування особистої інформації в кабінеті та пошук необхідного рейсу.

На рис 3.10 представлено вікно користувача після реєстрації або входу до системи.

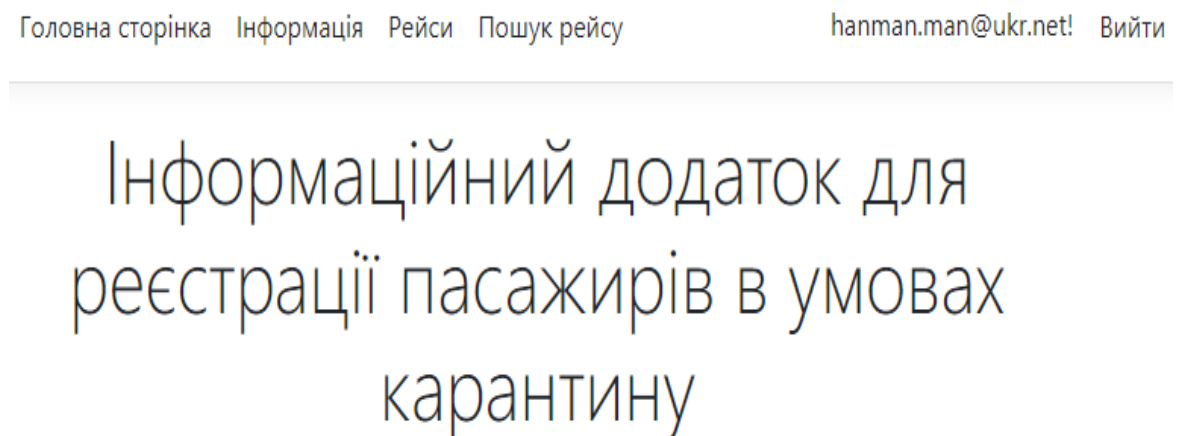


Рис 3.10 Користувач увійшов до особистого кабінету

Після реєстрації користувачеві буде доступна функція керування своїм обліковим записом. Можливість додання номера телефону, зміни електронної пошти, зміни паролю, або ж можливість видалення свого облікового запису.

На рис 3.11 зображено вікно з обліковим записом користувача.

## Керуйте своїм обліковим записом Змініть налаштування облікового запису

The screenshot shows a user profile management interface. On the left, there is a vertical menu with four items: 'Профіль' (Profile), 'Електронна пошта' (Email), 'Пароль' (Password), and 'Особисті дані' (Personal data). The 'Профіль' item is highlighted with a blue background. To the right of the menu, the 'Профіль' section is active. It contains three input fields: the first is labeled 'Ім'я користувача' (Username) and contains the text 'hanman.man@ukr.net'; the second is labeled 'Телефонний номер' (Phone number) and is empty; the third is a blue button labeled 'Зберегти' (Save).

Рис 3.11 Керування обліковим записом користувача

На рис 3.12 зображено вікно з можливістю зміни паролю для користувача

## Керуйте своїм обліковим записом Змініть налаштування облікового запису

The screenshot shows a password change interface. On the left, there is a vertical menu with four items: 'Профіль' (Profile), 'Електронна пошта' (Email), 'Пароль' (Password), and 'Особисті дані' (Personal data). The 'Пароль' item is highlighted with a blue background. To the right of the menu, the 'Змінити пароль' (Change password) section is active. It contains three input fields: the first is labeled 'Поточний пароль' (Current password) and is empty; the second is labeled 'Новий пароль' (New password) and is empty; the third is labeled 'Підтвердити новий пароль' (Confirm new password) and is empty. Below the input fields is a blue button labeled 'Оновити пароль' (Update password).

Рис 3.12 Зміна паролю



Після процесу реєстрації та редагування особистої інформації користувач може оглянути інформацію про карантинні заходи, зображену на рис 3.13

## Інформація про карантинні заходи

**Для реєстрації на рейс необхідно мати документ про вакцинацію або пцр тест**

У зв'язку з карантинними заходами для реєстрації в аеропорті необхідно мати при собі документ, який підтверджує наявність 1-2 дози вакцини, або наявність пцр-тесту

**Зверніть увагу, що пцр-тест дійсний 48-72 години з моменту його здачі. Якщо у вас відсутні дози вакцинації або пцр-тест, його можна отримати в самому аеропорті через 4-6 години після здачі.**

Рис 3.13 Інформація про карантинні заходи

Або оглянути список доступних рейсів для користувача, переглянути їх детальну інформацію, та за бажанням зареєструватися в черзі в самому аеропорті

На рис 3.14 представлено вікно з переліком доступних рейсів для користувача

## Рейси

Рейс	Час	Призначення	Перевізник	Термінал	Статус
W6 1613	5.55	Hurghada	Wizz Air	A	cancelled <a href="#">Інформація про рейс</a>   <a href="#">Зареєструватись в черзі</a>
ENT 4801	5.55	Bazylea	Wizz Air	A	cancelled <a href="#">Інформація про рейс</a>   <a href="#">Зареєструватись в черзі</a>
W6 1999	5.55	Enfidha	Enter Air	B	cancelled <a href="#">Інформація про рейс</a>   <a href="#">Зареєструватись в черзі</a>
W6 1575	6.00	Kutaisi	Wizz Air	B	cancelled <a href="#">Інформація про рейс</a>   <a href="#">Зареєструватись в черзі</a>
AF 1247	6.10	Edinburgh	Air France	C	cancelled <a href="#">Інформація про рейс</a>   <a href="#">Зареєструватись в черзі</a>
W6 1613	6:20	Stambul	Wizz Air	C	cancelled <a href="#">Інформація про рейс</a>   <a href="#">Зареєструватись в черзі</a>
ENT 4921	6:30	Heraklion	Enter Air	C	cancelled <a href="#">Інформація про рейс</a>   <a href="#">Зареєструватись в черзі</a>
LO 135	6:30	Miami	Wizz Air	A	cancelled <a href="#">Інформація про рейс</a>   <a href="#">Зареєструватись в черзі</a>

Рис 3.14 Список рейсів

В програмному застосунку існує функція пошуку необхідного рейсу за номером рейсу, часом, або місцем призначення.

На рис 3.15 зображено систему пошуку необхідного рейсу.

Головна сторінка Інформація Рейси Пошук рейсу hanman.man@ukr.net!

## Пошук рейсу

[Пошук](#)

[Повернутись назад](#)

Рис 3.15 Система пошуку рейсів

Після введення необхідної назви рейсу, наприклад “W6”, користувач отримає результат пошуку з рейсами, в яких є назва рейсу з ініціалами W6

На рис 3.16 зображено результат пошуку рейсів

Головна сторінка Інформація Рейси Пошук рейсу hanman.man@ukr.net!

## Результат пошуку рейсів

Рейс	Час	Призначення	Перевізіник	Термінал	Статус
W6 1613	5.55	Hurghada	Wizz Air	A	cancelled <a href="#">Інформація про рейс</a>   <a href="#">Зареєструватись в черзі</a>
W6 1999	5.55	Enfidha	Enter Air	B	cancelled <a href="#">Інформація про рейс</a>   <a href="#">Зареєструватись в черзі</a>
W6 1575	6.00	Kutaisi	Wizz Air	B	cancelled <a href="#">Інформація про рейс</a>   <a href="#">Зареєструватись в черзі</a>
W6 1613	6:20	Stambul	Wizz Air	C	cancelled <a href="#">Інформація про рейс</a>   <a href="#">Зареєструватись в черзі</a>

Рис 3.16 Результат пошуку

При перегляді рейсів, після процесу пошуку необхідного рейсу користувач може переглянути повну інформацію про рейс, та зареєструватися в черзі до аеропорту.

На рис 3.17 зображено вікно відображення інформації про рейс

## Інформація про рейс

<b>Рейс</b>	W6 1613
<b>Час</b>	5.55
<b>Прибуття</b>	Hurghada
<b>Перевізник</b>	Wizz Air
<b>Термінал</b>	A
<b>Статус</b>	cancelled

[Повернутись назад](#)

Рис 3.17 Перегляд інформації про рейс

На рис 3.18 зображено вікно реєстрації в черзі на рейс для користувача

## Реєстрація

### Інформація про рейс

<b>Рейс</b>	W6 1613
<b>Час</b>	5.55
<b>Призначення</b>	Hurghada
<b>Carrier</b>	Wizz Air
<b>Термінал</b>	A
<b>Статус</b>	cancelled

[Зареєструватись в черзі](#) | [Повернутись до списку](#)

Рис 3.18 Процес реєстрації в черзі на рейс

### 3.4 Тестування програмного продукту

Створення сайту – процедура складна, де кожен учасник команди виконує конкретні дії. Завершальний етап усього процесу – *web* тестування сайтів. Дана процедура грає найважливішу роль в створенні ресурсу, так як саме від якості тестування залежить подальше життя проекту. Як показує практика, багато розробників не приділяють належну увагу цьому етапу, покладаючись виключно на свої знання і досвід. Ресурс, який працює некоректно, викликає у користувачів негативні враження, в результаті довіра до компанії падає, що негативно впливає на її репутацію.

Основна мета тестування – перевірка функціонування ресурсу на відповідність до висунутих вимог. Тестування продукту спеціально створюють штучні ситуації, які можуть в майбутньому виникнути в роботі з ресурсом. Проводиться аналіз поведінки ресурсу на висунутих умовах. Коли помилки усуваються, сайт знову тестують.

Тестування на помилки здійснюється різними методами, основними методами для тестування веб-проекту будуть:

- тестування функціональності;
- перевірка *usability*(зручності) сайту;
- тест продуктивності.

Функціональне тестування сайту – вид перевірки, що виявляє неправильну роботу функціонала програми.

Серед основних правил функціонального тестування є:

- перевірка правильності роботи головних функцій ресурсу;
- коректність внутрішніх посилань;
- перевірка призначених для користувача форм;
- перевірка сторінок реєстрації та авторизації користувача;
- коректність роботи видалення та редагування даних користувача.

*Usability* тестування дозволяє перевірити, на скільки зручний сайт для користувача, на скільки легко йому знайти ту чи іншу інформацію. Одним словом, комфортність виконання бажаних дій.

Основною метою тестування є:

- визначення зрозумілості сайту для користувача;
- перевірка навігації по сайту;
- оцінка можливого враження користувача від продукту;
- виявлення зайвого для ресурсу.

Для перевірки веб-сайту було використано *Test-case*. Тестовий випадок(*test-case*) – це спосіб тестування, який описує сукупність кроків, умов і параметрів, для перевірки реалізації функцій.

Для видів *test-case* було обрано правильність набору введення даних користувача, зображену на таблиці 3.1

Таблиця 3.1

*Test-cases*

<i>Action</i>	<i>Expected result</i>	<i>Test result</i>
Введення некоректної пошти	Перевірка пошти	<i>Failed</i>
Введення некоректного паролі	Перевірка паролі	<i>Failed</i>
Введення коректних даних	Перевірка даних	<i>Passed</i>

*Test-case* показав відображення при введенні коректних та некоректних даних користувача, вимоги відображені правильно.

За рахунок проведення випробувань були усунуті ряд помилок, пов'язаних з часом виконання запитів до БД, некоректним відображенням даних, виправлена помилка з некоректним редагуванням даних користувача.

За допомогою проведення тестів були перевірені якість та продуктивність роботи користувача з системою, та збільшена швидкість ресурсу, швидкість завантаження сторінок.

### 3.5 Висновки до розділу

В результаті проведеної роботи була розроблена та протестована система інформаційного додатку для реєстрації авіапасажирів. Було зображено керівництво користувача. Перевірена якість зберігання інформації про користувачів, можливість редагування їх особистої інформації, відображення інформації про карантинні заходи, коректна інформація про рейс, можливість реєстрації в черзі до аеропорту.

Реалізовану систему було проаналізовано та протестовано на локальному сервері, у процесі якого було вирішено ряд помилок, та підвищена якість і продуктивність самої системи для користувачів.

## ВИСНОВКИ

Метою даного дипломного проєкту була розробка інформаційного додатку реєстрації авіапасажирів в умовах карантину, створена для простішого інформування доступності рейсів та інформування карантинних заходів для пасажирів.

Для досягнення цієї мети було поставлені наступні питання: визначити особливості реєстрації авіапасажирів; виявлення проблем в ефективності та якості обслуговування; розробка практичного прототипу інформаційного додатку для доцільності використання.

Був проведений аналіз підходів до реалізації інформаційного додатку. Було обрано найбільш зручну систему для управління додатку. Інтерфейс засобу був розроблений з метою зручності використання та повнотою наданої інформації. Пасажири можуть легко й швидко дізнатись необхідну інформацію.

Авіатранспорт – це важливий і популярний вид транспорту серед серед усіх країн світу. Це наймолодший вид транспорту, водночас саме авіаційний транспорт має величезну роль в міжконтинентальних перевезеннях в порівнянні з іншими. Швидкість, безпека, та комфорт це важлива частина в авіаперевезеннях. Завдяки цим перевагам та своїй популярності, авіакомпаніям необхідно впроваджувати нові технології для зручності та полегшення обслуговування великої кількості пасажирів.

В результаті проведеної роботи було реалізовано інформаційну систему, що включає в себе:

- опис предметної області, засноване на інформації з існуючих видів реєстрації;
- перевірка на вимоги до веб-додатків;
- модель інформаційної системи;

– зручний інтерфейс що дозволяє виробляти основні операції з інформацією в системі, видалення, додавання та інші дії з рядками бази даних;

– проведені випробування системи.

Використання цього додатку буде простим та інформативним для кожного пасажера. Сам додаток є web-додатком і може бути відкритий на будь якому пристрої з доступом до мережі інтернет.

Дана система виконує такі завдання: зберігання інформації про користувачів, можливість редагування їх особистої інформації; відображення інформації про карантинні заходи; надає користувачеві повну інформацію про рейс; дає можливість реєстрації в черзі до аеропорту.

Розроблений інформаційний додаток реєстрації авіапасажирів відповідає усім описаним вимогам, нормам в проектуванні.



## СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Фрейен Бен *HTML5* и *CSS3*. Разработка сайтов для любых браузеров и устройств; Питер – Москва, 2014. – 304 с.
2. Буч, Гради Введение в *UML* от создателей языка / Гради Буч, Джеймс Рамбо, Ивар Якобсон. – М.: ДМК Пресс, 2015. – 496 с.
3. C# documentation [Электронный ресурс].  
<https://docs.microsoft.com/uk-ua/dotnet/csharp/>
4. Транспортна система [Электронный ресурс].  
<https://geografiamozil2.jimdofree.com/>
5. Бази даних *sql* [Электронный ресурс]  
<https://site-do.ru/db/sql1.php>
6. Вимоги до програмного забезпечення [Электронный ресурс]  
[https://www.wiki.uk-ua.nina.az/вимоги\\_до\\_програмного\\_забезпечення](https://www.wiki.uk-ua.nina.az/вимоги_до_програмного_забезпечення)
7. *Entity Framework Core* [Электронный ресурс].  
<https://metanit.com/sharp/entityframeworkcore/>
8. Об'єктно-орієнтоване програмування. Лабораторний практикум; Національний авіаційний університет; Гамаюн В. П., Зудов О. М., НАУ-друк, 2010. – 48 с.–
9. ДСТУ 3008-95 Документація. Звіти у сфері науки і техніки. Структура і правила оформлення.
- 10 .Бойченко С.В., Іванченко О.В. Положення про дипломні роботи (проекти) випускників Національного авіаційного університету. – К.: НАУ, 2017. – 63 с.