

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Національний університет «Запорізька політехніка»

Інститут інформатики та радіoeлектроніки,
Факультет комп'ютерних наук і технологій
(повне найменування інституту, назва факультету)

Кафедра програмних засобів
(повне найменування кафедри)

Пояснювальна записка

до дипломного проєкту (роботи)

магістр

(ступінь вищої освіти)

на тему ДОСЛІДЖЕННЯ ТА ПРОГРАМНА РЕАЛІЗАЦІЯ МЕТОДІВ
ПРОГНОЗУВАННЯ ПАСАЖИРОПОТОКУ АЕРОПОРТУ
RESEARCH AND SOFTWARE IMPLEMENTATION OF METHODS FOR
FORECASTING AIRPORT PASSENGER TRAFFIC

Виконав: студент(ка) 2 курсу, групи КНТ-219м

Спеціальності 122 Комп'ютерні науки

(код і найменування спеціальності)

Освітня програма (спеціалізація)

Системи штучного інтелекту

Васильєв А.К.

(прізвище та ініціали)

Керівник Зеленьова І.Я.

(прізвище та ініціали)

Рецензент Скрупський С.Ю.

(прізвище та ініціали)

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Національний університет «Запорізька політехніка»
(повне найменування закладу вищої освіти)

Інститут, факультет ІРЕ, ФКНТ
Кафедра програмних засобів
Ступінь вищої освіти магістр
Спеціальність 122 Комп'ютерні науки
(код і найменування)

Освітня програма (спеціалізація) Системи штучного інтелекту
(назва освітньої програми (спеціалізації))

ЗАТВЕРДЖУЮ

Завідувач кафедри ПЗ, д.т.н, проф.
С.О. Субботін
“ ” 2020 року

З А В Д А Н Н Я

НА ДИПЛОМНИЙ ПРОЄКТ (РОБОТУ) СТУДЕНТА(КИ)

Васильєва Артема Костянтиновича

(прізвище, ім'я, по батькові)

1. Тема проєкту (роботи) Дослідження та програмна реалізація методів прогнозування пасажиропотоку аеропорту
Research and Software Implementation of Methods for Forecasting Airport Passenger Traffic

керівник проєкту (роботи) Зеленьова Ірина Яківна, к.т.н., доцент,
(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

затверджені наказом закладу вищої освіти від “3” листопада 2020 року № 301

2. Строк подання студентом проєкту (роботи) 15 грудня 2020 року

3. Вихідні дані до проєкту (роботи) рекомендована література, технічне завдання

4. Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити) 1. Аналіз проблеми та постановка завдань дослідження. 2. Матеріали і методи. 3. Опис програми. 4. Розробка програми. 5. Експлуатація та тестування програми. 6. Економіко-організаційна частина, 7. Охорона праці та безпека в надзвичайних ситуаціях.

5. Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень) Слайди презентації

6. Консультанти розділів проєкту (роботи)

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	прийняв виконане завдання
1-5 Основна частина	Зеленьова І.Я., доцент		
Економіко-організаційна частина	Пожуєва Т.О., професор		
Охорона праці та безпека в надзвичайних ситуаціях	Коробко О.В., ст. викладач		
Нормоконтролер	Андрєєв М.О., асистент		

7. Дата видачі завдання “ 7 ” вересня 2020 року.

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів дипломного проєкту (роботи)	Строк виконання етапів проєкту (роботи)	Примітка
1	Постановка завдання роботи	1 тиждень	Завдання, ТЗ
2	Аналіз предметної області	2 тиждень	Розділ 1
3	Розробка методу	3–4 тижні	Розділ 2
4	Розробка архітектури програми	5 тиждень	Розділ 3
5	Розробка програми	6–7 тижні	Розділ 4
6	Тестування та відлагодження програмного забезпечення	8 тиждень	Розділ 5
7	Розробка економіко-організаційної частини	9 тиждень	Розділ 6
8	Розробка заходів з охорони праці та безпеки в надзвичайних ситуаціях	10 тиждень	Розділ 7
9	Оформлення пояснювальної записки та документів до неї	11 тиждень	Додатки
10	Нормоконтроль та рецензування	12 тиждень	
11	Захист роботи	13 тиждень	

Студент(ка)

(підпис) Васильєв А.К.
(прізвище та ініціали)

Керівник проєкту (роботи)

(підпис) Зеленьова І.Я.
(прізвище та ініціали)

РЕФЕРАТ

Пояснювальна записка до дипломної кваліфікаційної роботи магістра:
121 с., 28 рис., 9 табл., 3 дод., 28 джерел.

АЕРОПОРТ, ПРОГНОЗУВАННЯ, ПАСАЖИРОПОТІК,
ВЕБЗАСТОСУНОК, PHP, MYSQL.

Об'єкт дослідження – процес прогнозування пасажиропотоку аеропорту. Предмет дослідження – програмне забезпечення підтримки роботи аеропорту.

Мета дослідження – розроблення програмного забезпечення прогнозування пасажиропотоку аеропорту для підвищення автоматизованості даного процесу та обґрунтованості рішень, що приймаються відносно планування діяльності аеропорту.

Матеріали, методи та технічні засоби: мова програмування PHP, система керування базами даних MySQL.

Результати. Виконано проєктування та реалізацію програмного забезпечення прогнозування пасажиропотоку аеропорту у вигляді вебзастосунку, що дозволяє виконувати облік діяльності аеропорту, включаючи роботу з авіалініями, виконанням рейсів, технічним обслуговуванням, а потім автоматизовано виконувати прогнозування пасажиропотоку на основі автоматично зібраних даних щодо реєстрації пасажирів за минулі періоди.

Висновки. Запропоновано метод прогнозування пасажиропотоку аеропорту, який ґрунтується на використанні методу опорних векторів з квадратичною функцією втрат, використовує в якості вхідних ознак пасажиропотік за минулі періоди, коливання курсу валют за цей період та номер базового періоду.

Галузь використання – програмна підтримка роботи аеропорту.

ABSTRACT

Explanatory note to the diploma qualifying work of the master: 121 pages, 28 figures, 9 tables, 3 appendixes, 28 sources.

AIRPORT, FORECASTING, PASSENGER FLOW, WEB APPLICATION, PHP, MYSQL.

The object of the research is the process of forecasting airport passenger traffic.

The subject of the research is software to support the airport functioning.

The purpose of the research is to develop software for forecasting airport passenger traffic to increase the automation of this process and the validity of decisions made regarding the planning of airport activities.

Materials, methods and technical means: PHP programming language, MySQL database management system.

Results. The design and implementation of airport passenger traffic forecasting software in the form of a web application was made. The developed application allows accounting of airport activities, including work with airlines, flights, maintenance. Such features allow to automatically perform passenger traffic forecasting based on automatically collected data on passenger registration for the previous periods.

Conclusions The method of airport passenger traffic forecasting is proposed, which is based on the usage of the method of support vector machines with quadratic loss function, usage of the passenger flow for previous periods, exchange rate fluctuations for this period and the number of the base period as inputs.

Field of use is the software support for the airport.

ЗМІСТ

	С.
Перелік скорочень та умовних познач	8
Вступ	9
1 Аналіз проблеми та постановка завдань дослідження	10
1.1 Проблема, тенденції та особливості галузі авіаперевезень	10
1.2 Огляд програмних засобів обліку роботи аеропорту	12
1.3 Висновки за розділом 1	19
2 Матеріали і методи	21
2.1 Методи прогнозування пасажиропотоку аеропорту	21
2.2 Метод опорних векторів з квадратичною функцією втрат	24
2.3 Запропонований метод прогнозування пасажиропотоку аеропорту	26
2.4 Висновки за розділом 2	28
3 Опис програми	29
3.1 Визначення вимог до вебзастосунку	29
3.2 Вибір засобів проєктування та реалізації програми	31
3.3 Проєктування бази даних	34
3.4 Висновки за розділом 3	40
4 Розробка програми	41
4.1 Опис реалізованої структури програми	41
4.2 Опис процесу прогнозування пасажиропотоку аеропорту	43
4.3 Опис розроблених скриптів	44
4.4 Висновки за розділом 4	48
5 Експлуатація та тестування програми	50
5.1 Призначення програми	50
5.2 Опис роботи з програмою	50
5.3 Висновки за розділом 5	59
6 Економіко-організаційна частина	60
6.1 Планування розробки програмного продукту	60
6.2 Визначення витрат на розробку програми	63

6.3 Розрахунок основної заробітної плати	63
6.4 Розрахунок додаткової заробітної плати.....	64
6.5 Розрахунок відрахувань до соціального внеску	64
6.6 Визначення витрат на матеріали	65
6.7 Витрати на спеціальне устаткування.....	66
6.8 Накладні витрати	69
6.9 Оцінка економічної ефективності	70
7 Охорона праці та безпека в надзвичайних ситуаціях.....	72
7.1 Аналіз потенційних небезпек	72
7.2 Заходи по забезпеченню безпеки.....	72
7.3 Заходи по забезпеченню виробничої санітарії та гігієни праці	74
7.4 Заходи по забезпеченню пожежної безпеки	79
7.5 Заходи по забезпеченню безпеки у надзвичайних ситуаціях.....	80
Висновки.....	82
Перелік джерел посилання	83
Додаток А Технічне завдання.....	86
Додаток Б Текст програми.....	91
Додаток В Слайди презентації	114

ПЕРЕЛІК СКОРОЧЕНЬ ТА УМОВНИХ ПОЗНАК

- ARIMA – AutoRegressive Integrated Moving Average;
LS-SVM – Least-Squares Support-Vector Machines;
ЕОМ – електронно-обчислювальна машина;
ПУЕ – правила улаштування електроустановок.

ВСТУП

Для будь-якого аеропорту одним з основних показників діяльності є пасажиропотік, адже від даного показника залежить завантаженість аеропорту, рейсів, а відповідно це впливає на кількість рейсів, на витрати на авіап перевезення, що згодом визначає прибуток, який буде отримано аеропортом. Тобто можна говорити про те, що пасажиропотік визначає зворотній зв'язок з замовниками, тобто пасажирями. Їх рівень задоволеності надаваними послугами та їх якістю визначає в результаті пасажиропотік. Тому будь-якому аеропорту важливо не тільки фіксувати такі показники, але і планувати їх, для чого потрібний інструмент прогнозування.

Дослідження і прогнозування обсягу ринку повітряних суден, як правило, проводиться в три етапи:

- виявлення чинників, що визначають попит на пасажирські авіап перевезення, визначення тенденцій розвитку транспортних потреб населення, формування прогнозу обсягу ринку пасажирських авіап перевезень;
- виявлення перспективних напрямків розвитку цивільної авіатехніки, визначення тенденцій розвитку авіатранспортної мережі, формування прогнозу розвитку парку пасажирських літаків;
- виявлення напрямків розвитку авіаційної промисловості, визначення тенденцій в процесах життєвого циклу пасажирських літаків; формування прогнозу обсягу ринку пасажирських літаків [1].

Дану роботу присвячено розв'язанню першого етапу, тобто прогнозуванню пасажиропотоку аеропорту. Обсяги та динаміка перевезень конкретного аеропорту можуть істотно відрізнятися навіть в межах країни, що загалом говорить про актуальність проблеми, що розглядається.

1 АНАЛІЗ ПРОБЛЕМИ ТА ПОСТАНОВКА ЗАВДАНЬ ДОСЛІДЖЕННЯ

1.1 Проблема, тенденції та особливості галузі авіаперевезень

Повітряний транспорт є стратегічно значущою ланкою економіки майже будь-якої країни, зокрема України, так як грає ключову роль у вирішенні завдань:

- забезпечення транспортної доступності перевезень для населення, підвищення якості життя населення та рівня розвитку економіки;
- використання транзитного потенціалу, інтеграції в світову транспортну систему і підвищення її конкурентоспроможності на глобальному ринку;
- забезпечення економічної безпеки держави шляхом збереження її транспортної цілісності;
- готовності до надзвичайних ситуацій в частині модернізації та експлуатації об'єктів інфраструктури повітряного транспорту [2].

Авіаперевезення напряму залежать від стану економіки загалом, але з іншого боку напряму впливають на саму економіку.

Щодо обсягів авіаперевезень за останні роки можна виділити наступне:

- у 2019 авіакомпанії світу збільшили пасажиропотік на 4,2%, як свідчать дані Міжнародної асоціації повітряного транспорту;
- темпи зростання істотно сповільнилися з 7,3%, зафіксованих за підсумками 2018 року, крім того, показник вперше з глобальної фінансової кризи 2009 року опустився нижче довгострокового тренду, який становить близько 5,5%;
- відзначається, що завантаженість пасажирських рейсів в 2019 році збільшилася на 0,7 процентного пункту – до рекордних 82,6%, попередній максимум в 81,9% був встановлений в 2018 році;
- зростання попиту на авіаперевезення перевищило розширення перевізних потужностей авіакомпаній, що склало 3,4%;

– усі регіони продемонстрували збільшення перевезень в пасажиро-кілометрах в 2019 році, у тому числі в Африці показник виріс на 4,9%, в Азіатсько-Тихоокеанському регіоні – на 4,8%, в Європі – на 4,2%, в Латинській Америці – також на 4,2%, на Близькому Сході – на 2,4% і в Північній Америці – на 4,1%;

– міжнародні пасажирські перевезення в минулому році зросли на 4,1% після підвищення на 7,1% в 2018 році, завантаженість таких рейсів збільшилася на 0,8 процентного пункту – до 82%, кількість посадкових місць піднялася на 3%;

– за підсумками 2019 року попит на місцеві рейси в цілому по світу збільшився 4,5% після підйому на 7,8% в 2018 році, при цьому всі регіони продемонстрували підвищення, лідером став Китай (збільшення у цій країні становило 7,8%), завантаженість внутрішніх рейсів зроста на 0,4 процентного пункту – до 83,7%, кількість посадкових місць підвищилося на 4,1% [3].

За той же час в Україні обсяг пасажирських перевезень на міжнародних маршрутах збільшився на 11%, кількість рейсів загалом збільшилась протягом 2019 року на 3,4%, пасажиропотік всіх українських аеропортів збільшився на 19,9, пасажиропотік окремих аеропортів України продемонстрував наступну динаміку:

- аеропорт Бориспіль – збільшення на 21,2%;
- аеропорт Жуляни – збільшення на 6%;
- аеропорт Львів – збільшення на 41,2%;
- аеропорт Харків – збільшення на 25%;
- аеропорт Запоріжжя – збільшення на 26,7%;
- аеропорт Дніпропетровськ – збільшення відбулось на 11,5% (менше ніж в 2 рази ніж у сусідньому Запоріжжі) [4].

Відповідно важливо не тільки знати фактичні обсяги авіаперевезень, але і розуміти, яка кількість переміщується в країну та з країни, відповідно на

отримання яких сум коштів можна розраховувати. Тобто важливими є якісні планування і прогнозування.

1.2 Огляд програмних засобів обліку роботи аеропорту

Інформаційна система «Аеропорт» – це комплексне рішення, покликане забезпечити підтримку виробничих, технологічних і фінансових процесів авіаційного підприємства, воно дозволяє вирішувати завдання, пов'язані з організацією, підготовкою та виконанням робіт із забезпечення авіаційних перевезень в аеропорту, розвиток системи здійснюється з урахуванням вимог міжнародних нормативних документів, рекомендацій і правил, що регламентують діяльність аеропортів в області наземного обслуговування повітряних суден, пасажирів і організації повітряного руху, щодо даної системи можна виділити наступне:

а) основними функціями системи є:

- 1) управління процесом обслуговування рейсів і пасажирів;
- 2) управління ресурсами (статичними і мобільними);
- 3) управління персоналом;
- 4) управління фінансами;
- 5) голосове і візуальне інформування [5];

б) робота системи направлена на вирішення таких завдань:

- 1) забезпечення безпеки польотів;
- 2) ефективне планування розкладу і добового плану обслуговування повітряних суден;
- 3) скорочення часу на обслуговування, підвищення якості;
- 4) ефективне планування і використання персоналу та ресурсів, задіяних в обслуговуванні;
- 5) скорочення витрат виробництва;
- 6) скорочення реакції на збої й затримки в технологічному процесі;

7) зменшення чисельності адміністративного та обслуговуючого персоналу за рахунок підвищення ефективності його роботи;

8) своєчасне надання максимального обсягу оперативної інформації кожному учаснику виробничого процесу із застосуванням різних технічних засобів [4];

в) до складу комплексу входять наступні програмні та функціональні модулі:

- 1) розклад рейсів;
- 2) добовий план обслуговування;
- 3) добовий план обслуговування – «сигнальное табло»;
- 4) планування персоналу і ресурсів;
- 5) облік доходів, формування актів форми С;
- 6) інтеграція з 1С;
- 7) мобільні пристрої для віддаленої роботи;
- 8) графічна панель «Карта польотів»;
- 9) графічна панель «План стоянок»;
- 10) графік руху повітряних суден;
- 11) графік завантаження аеропорту (з персоналу і ресурсів);
- 12) подання у вигляді стикувань рейсів;
- 13) електронний диктор;
- 14) інформаційне табло для інформування співробітників і пасажирів;
- 15) вебзастосунок;
- 16) система звітності та бізнес-аналітики [5].

Корпоративна інформаційна система «Аеропорт» призначена для автоматизації діяльності аеропорту, основною особливістю системи є всебічний охоплення основних видів діяльності аеропорту в єдиному програмному середовищі, як для виробничо-технологічних задач, так і для завдань управлінського обліку і бухгалтерського обліку, вона дозволяє

організувати ефективне управління діяльністю аеропорту в єдиному інформаційному і алгоритмічній просторі для прийняття рішень в ході здійснення виробничої діяльності аеропорту, дана система складається з великого набору взаємопов'язаних між собою програмних продуктів, зокрема:

а) розклад руху повітряних суден: даний комплекс призначений для ведення розкладу руху повітряних суден щодо аеропорту прив'язки, розклад описує стикувальні пари рейсів, для яких визначені період навігації, тижнева періодичність, аеропорти і час відправлення і прибуття, залежно від параметра налаштування комплексу розклад можна вводити в UTC або в місцевому часу аеропортів відправлення і прибуття, при цьому на екрані завжди відображаються обидва часу, один з яких вводиться, а інший розраховується, на підставі розкладу формується добовий план польотів, в який потрапляють тільки «узгоджені» рейси, у програмному комплексі є функція графічного представлення розкладу (добового плану польотів), графік оперативно формується на будь-який заданий користувачем період і служить для аналізу розкладу, наприклад, на предмет узгодження рейсів, реалізує наступні функції:

- 1) ведення поточного розкладу за місцевим часом і в UTC;
- 2) копіювання, зміщення і розподіл рейсів;
- 3) узгодження рейсів;
- 4) аналіз розкладу за довільно обраний період;
- 5) відображення розкладу на сайті аеропорту [6];

б) наземне і аеропортове обслуговування повітряних суден: призначене для автоматизації діяльності аеропортів в частині обліку аеропортового і наземного обслуговування повітряних суден, дозволяє вести розкладу руху повітряних суден в аеропорту, формувати добовий план польотів і доповнення до нього, а також формувати і друкувати зведено-завантажувальні відомості та акти аеропортового і наземного обслуговування повітряних суден, реалізує наступні функції:

1) розклад руху повітряних суден;

2) план польотів: добовий план польотів в аеропорту призначений для введення і коригування даних про прибуваючі і відбуваючі повітряні судна, добовий план польотів з маршрутом руху і плановим часом посадки та вильоту формується автоматично на підставі розкладу руху повітряних суден і може потім коригуватися і доповнюватися вручну диспетчерськими службами аеропорту, документ дозволяє формувати зведено-завантажувальну відомість, вводити заявки на додаткове обслуговування і вимоги на заправку, а також в автоматичному режимі відповідно до опису формувати акти на обслуговування повітряних суден в аеропорту, в акти, сформовані автоматично, можуть бути внесені зміни до моменту їх роздруківки, при формуванні актів враховується поточна (на дату вильоту) ціна послуги і для рейсів іноземних авіакомпаній курс тієї валюти, в якій оплачується послуга, можливе формування декількох актів на один рейс, наприклад, для випадку дозаправки повітряних суден або потреби в додаткових послугах, в цьому випадку при автоматичному формуванні в новий акт потрапляють тільки ті послуги (збори), які були відсутні у всіх попередніх актах, крім того, програма стежить за автоматичною нумерацією актів в межах року і не допускає подвійних номерів і пропусків в нумерації актів;

3) план завантаження;

4) опис актів з обслуговування;

5) реєстрація заявок на додаткове обслуговування повітряних суден;

6) введення вимог за заправку;

7) формування актів на обслуговування повітряних суден;

8) формування зведено-завантажувальної відомості;

9) опис операцій з обслуговування та їх трудомісткості;

10) формування статистичної звітності;

- 11) формування добового плану польотів;
- 12) формування рекламного розкладу;
- 13) формування графіка прильоту-вильоту по аеропорту;
- 14) зв'язок з системою «Вантажний склад»;
- 15) зв'язок з автоматизованим робочим місцем «Спецобслуговування пасажирів»;
- 16) зв'язок з програмним комплексом «Купол» [5];

в) відділ кадрів: призначена для автоматизації роботи кадрових служб середніх і великих підприємств, реалізує наступні функції:

1) структура підприємства: організаційна структура підприємства описується за допомогою ієрархічного довідника, в якому для кожного підрозділу вказується його батьківський підрозділ, крім того даний довідник дозволяє відстежувати історію зміни структури підприємства таким чином, що на будь-яку задану дату можна бачити існуючі на той момент підрозділи підприємства;

2) штатний розклад, штатно-посадові картки: штатний розклад призначений для обліку штатних одиниць на підприємстві, він дозволяє вести штатний розклад підприємства, а також розраховувати прогнозовані витрати на оплату праці, по ньому визначаються наявні на підприємстві вакансії при прийомі на роботу або переміщенні на іншу посаду, у штатно-посадовій картці зазначаються штатна одиниця (посада і підрозділ), кількість таких одиниць, категорія персоналу для звіту за формою 1Т (керівник, фахівець, робочий тощо) і умови оплати праці (оклад або коефіцієнт за тарифною сіткою, умови праці, інші доплати і надбавки, тривалість відпустки тощо), кожна картка має свій термін дії, а сукупність карток формують штатний розклад підприємства, у програмі передбачено ведення декількох видів штатного розкладу, при цьому тільки один штатний розклад в кожен момент часу може бути чинним, інші варіанти розглядаються як проекти штатного розкладу, які можуть використовуватися при

підготовці нового штатного розкладу або для аналізу планових витрат на заробітну плату;

- 3) змінні графіки роботи;
- 4) особова картка працівника;
- 5) посадова картка;
- 6) кадрові операції (прийом, переклад, відрядження, відпустка, звільнення, накази різні);
- 7) друк довідок і звітів по льотному складу [4];

г) технологічний графік та управління добовим планом польотів: добовий план польотів автоматично формується на підставі розкладу руху повітряних суден із заданою в параметрах програми періодичністю і на задану кількість днів вперед, технологічні графіки обслуговування повітряних суден, включених в добовий план польотів, формуються автоматично відповідно до опису технологічних операцій по прильоту і технологічних операцій по вильоту для різних типів повітряних суден і різних видів рейсу (міжнародний або внутрішній), в описі технологічної операції вказується статус рейсу, який встановлюється на момент початку операції і на момент закінчення операції, а також види рейсів, для яких застосовується дана операція, також вказуються зони аеропорту (використовується для інформування пасажирів) і дані по рейсу, які потрібно обов'язково вводити при виконанні даної операції, якщо технологічна операція не вимагає тривалості за часом (наприклад, реєстрація прильоту або вильоту), то вона відзначається, як подія, вказуючи підрозділ, відповідальний за виконання операції, можна управляти доступом на редагування даних по операції, реалізуючи наступні функції:

- 1) оперативне управління добовим планом польотів;
- 2) робота в реальному режимі часу;
- 3) опис технологічних операцій;
- 4) ведення технологічних графіків обслуговування повітряних суден;

5) інформаційне табло контролю виконання технологічних операцій: таке уявлення зручно для оперативного контролю виконання технологічних графіків, досить побіжного погляду на табло, щоб оцінити загальну картину обслуговування всіх повітряних суден, у той же час можна отримати детальну інформацію щодо рейсу, відкривши картку рейсу або картку технологічної операції, тут можна виконати реєстрацію прибуття літака і реєстрацію вильоту, робити скасування рейсу, поєднання рейсу або затримку рейсу, вводити і роздруковувати акт затримки і зведену завантажувальну відомість;

6) ведення історії виконання технологічних графіків;

7) автоматична обробка вхідних і вихідних телеграм;

8) акти на затримку рейсів: акт на затримку може формуватися з форми оперативного управління або з інформаційного табло, є також окремий електронний журнал, в якому реєструються всі акти на затримку рейсу;

9) зведені завантажувальні відомості;

10) фіксація послуг з обслуговування рейсів;

11) виведення інформації на сайт: інформація відображається у вигляді онлайн табло прильоту (прибуття) і онлайн табло вильоту (відправлення), інформація на онлайн табло оновлюється автоматично за поточним станом добового плану польотів із заданою в параметрах програми періодичністю;

12) зв'язок з розкладом;

13) зв'язок з системою «Наземне і аеропортове обслуговування»;

14) зв'язок з підсистемою інформування пасажирів [6].

Проведений огляд програмних засобів обліку роботи аеропорту дозволяє виділити наступне:

– сучасні системи в даній галузі мають велику функціональність, але при цьому кожна така система складається з великої кількості підсистем, часом з різних галузей, об'єднаних в єдину систему;

– незважаючи на широку функціональність щодо обліку операцій в системах відсутні засоби прогнозування пасажиропотоку, відповідно накопичені дані не використовуються для прогнозування, а результати прогнозування не інтегруються в систему;

– системи розроблені у вигляді настільних застосунків, що обмежує їх галузь використання і дещо виокремлює зі світу сучасного програмного забезпечення, що використовується в мережі, більш актуальним для таких напрямків є використання вебзастосунків, при цьому в існуючих таких системах вебзастосунки часто стоять осторонь, вимагаючи створення окремої підсистеми.

Тож можна вважати важливою розробку програмного забезпечення прогнозування пасажиропотоку аеропорту для вирішення виявлених і описаних проблем.

1.3 Висновки за розділом 1

Було розглянуто проблему авіап перевезень у світі, її стан та важливість з приведенням даних про тенденції галузі в світі загалом та в Україні зокрема.

Було виконано огляд програмних засобів, які можуть використовуватися для обліку роботи аеропорту, зокрема було проаналізовано засоби інформаційної системи «Аеропорт» та корпоративної інформаційної системи «Аеропорт», за результатами було виділено їх особливості, функціональні можливості, але при цьому було виявлено відсутність засобів прогнозування пасажиропотоку аеропортів у них.

За результатами виконаного огляду було сформульовано набір завдань:

- розроблення методу прогнозування пасажиропотоку аеропорту;
- проектування вебзастосунку;
- програмна реалізація та тестування вебзастосунку.

2 МАТЕРІАЛИ І МЕТОДИ

2.1 Методи прогнозування пасажиропотоку аеропорту

Задача прогнозування пасажиропотоку аеропорту належить до задач прогнозування часових рядів, коли за послідовністю попередніх значень змінної прогнозують її значення на майбутній період.

На сьогоднішній день для прогнозування часових рядів використовуються наступні методи:

а) методи прогнозування, засновані на згладжуванні, експоненційному згладжуванні і ковзному середньому:

1) “наївні” моделі прогнозування: при створенні “наївних” моделей передбачається, що деякий останній період прогнозованого часового ряду найкраще описує майбутнє цього прогнозованого ряду, тому в цих моделях прогноз, як правило, є дуже простою функцією від значень прогнозованої змінної в недалекому минулому;

2) середні і ковзаючі середні: найпростіша така модель на відміну від найпростішої “наївної” моделі, якій відповідав принцип “завтра буде як сьогодні”, організована за принципом “завтра буде як було в середньому за останній час”, така модель, звичайно стійкіша до флуктуацій, оскільки в ній згладжуються випадкові викиди щодо середнього, незважаючи на це, цей метод ідеологічно настільки ж примітивний як і “наївні” моделі і йому властиві майже ті ж самі недоліки, при цьому, як правило, значення часового ряду з недалекого минулого краще описують прогноз, ніж старіші значення цього ж ряду, тоді можна використовувати для прогнозування ковзне середнє (сене його полягає в тому, що модель бачить тільки найближче минуле (на T відліків за часом в глибину) і ґрунтуючись тільки на цих даних будує прогноз), при прогнозуванні досить часто використовується метод експоненційних середніх, який постійно адаптується до даних за рахунок нових значень;

3) регресивні методи прогнозування: поряд з описаними вище методами, заснованими на експоненціальному згладжуванні, вже досить довгий час для прогнозування використовуються регресійні алгоритми, коротко суть алгоритмів такого класу можна описати так: існує прогнозована змінна Y (залежна змінна) і відібраний заздалегідь комплект змінних, від яких вона залежить – $X_1, X_2, \dots, X_N$ (незалежні змінні), природа незалежних змінних може бути різною, наприклад, якщо припустити, що Y – рівень попиту на деякий продукт в наступному місяці, то незалежними змінними можуть бути рівень попиту на цей же продукт в минулий і позаминулий місяці, витрати на рекламу, рівень платоспроможності населення, економічна обстановка, діяльність конкурентів і багато іншого, головне – вміти формалізувати всі зовнішні чинники, від яких може залежати рівень попиту в числову форму [7];

б) методи Бокса-Дженкінса (AutoRegressive Integrated Moving Average (ARIMA)): в це сімейство входить декілька алгоритмів, найвідомішим і використовуваним з них є алгоритм ARIMA, він вбудований практично в будь-який спеціалізований пакет для прогнозування, у класичному варіанті ARIMA не використовуються незалежні змінні, моделі спираються тільки на інформацію, що міститься в передісторії прогнозованих рядів, що обмежує можливості алгоритму, в даний час в науковій літературі часто згадуються варіанти моделей ARIMA, що дозволяють враховувати незалежні змінні, на відміну від розглянутих раніше методик прогнозування часових рядів, методологія ARIMA не передбачає будь-якої чіткої моделі для прогнозування даної часової серії, дається лише загальний клас моделей, що описують часовий ряд і дозволяють якось виражати поточне значення змінної через її попередні значення, потім алгоритм, підлаштовуючи внутрішні параметри, сам вибирає найбільш підходящу модель прогнозування, існує ціла ієрархія моделей Боксу-Дженкінса [6];

в) нейромережеві моделі бізнес-прогнозування: в даний час найперспективнішим кількісним методом прогнозування є використання нейронних мереж, можна назвати багато переваг нейронних мереж над іншими алгоритмами:

1) при використанні нейронних мереж легко дослідити залежність прогнозованої величини від незалежних змінних: наприклад, продажі носять сезонний характер, мають тренд і якимось чином залежать від активності конкурентів, потрібно побудувати систему, яка б все це природним чином враховувала і будувала б короткострокові прогнози, у такій постановці завдання велика частина класичних методів прогнозування буде просто неспроможною, можна спробувати побудувати систему на основі нелінійної регресії, або варіації алгоритму ARIMA, що дозволяє враховувати зовнішні параметри, але це будуть моделі швидше за все малоефективні (за рахунок суб'єктивного вибору моделі) і вкрай негнучкі, використовуючи ж навіть найпростішу нейромережеву архітектуру (персептрон з одним прихованим шаром) і базу даних (з продажами і всіма параметрами) легко отримати працюючу систему прогнозування, при чому врахування або не врахування системою зовнішніх параметрів буде визначатися включенням або виключенням відповідного входу в нейронну мережу, більш досвідчений експерт може з самого початку скористатися будь-яким алгоритмом визначення важливості (наприклад, використовуючи нейронну мережу із загальною регресією і генетичним підстроюванням) і відразу визначити значимість вхідних змінних, щоб потім виключити з розгляду параметри, що мало впливають;

2) експерт не є заручником вибору математичної моделі поведінки часового ряду: побудова нейромережевої моделі відбувається адаптивно під час навчання, без участі експерта, при цьому нейронній мережі

пред'являються приклади з бази даних і вона сама підлаштовується під ці дані [7].

2.2 Метод опорних векторів з квадратичною функцією втрат

Метод опорних векторів є машинним алгоритмом, котрий навчається на прикладах та використовується для класифікації об'єктів: наприклад, метод опорних векторів може розрізнити аварійний режим роботи електромеханічної системи та класифікувати його за наявності попередніх досліджень, можливих за технологічними вимогами режимів роботи, такий підхід розкриває значні можливості для побудування адаптивних систем автоматичного керування, в основі методу лежить деяка математична сутність – алгоритм максимізації деякої математичної функції відносно наявного набору даних, в основі лежать чотири ключові поняття:

- відділяюча гіперплощина, що є математичною сутністю, що відділяє між собою класи об'єктів з однаковими ознаками;
- гіперплощина максимальної межі;
- м'яка межа;
- функція ядра [8].

Версія методу опорних векторів для регресії – це модель, яка побудована з використанням опорного методу класифікації і залежить тільки від підмножини даних для навчання, тому що штрафна функція при побудові моделі не звертає уваги на те, що точка лежить за межами краю:

- у машинному навчанні, метод опорних векторів – контрольоване навчання моделей з використання подібних алгоритмів для аналізу даних і розпізнавання шаблонів, даний метод використовується для задач класифікації та регресійного аналізу, основний метод опорних векторів приймає набір вхідних даних і прогнозує для кожного даного входу одну з двох можливих форм виходу;

– з огляду на набір навчальних спостережень (навчальну вибірку), кожне з яких позначене як таке, що належить до однієї з двох категорій, алгоритм навчання методу опорних векторів будує модель, яка визначає, до якої з двох категорій належать нові спостереження;

– модель методу опорних векторів – відображення даних у вигляді точок у просторі, так що між спостереженнями окремих категорій є розрив і він максимальний, потім нові спостереження відобразяться в тому ж просторі і будуть належати до однієї з категорій залежно від того, по який бік розриву вони відобразилися;

– на додаток до виконання лінійної класифікації, метод опорних векторів може ефективно виконувати нелінійну класифікацію, використовуючи довільну функцію ядра, неявно відображаючи вхідні дані в багатовимірні простори функцій [9].

Least-Squares Support-Vector Machines (LS-SVM) – це метод, який розв’язує множину лінійних рівнянь замість розв’язання проблеми квадратичного програмування.

Для побудови LS-SVM регресії використовується наступний алгоритм: розглянемо задачу відновлення залежності, припустимо, що дана навчальна вибірка $D_n\{(x_k, y_k): x_k \in X, y_k \in Y; k = 1, \dots, n\}$ обсягом n спостережень виду (2.1):

$$y_k = m(x_k) + e_k, k = 1, \dots, n, \quad (2.1)$$

де $e_k \in R$ є незалежною і однаково розподіленою помилкою;

$m(x)$ – невідома дійсна гладка функція [10].

Замість невідомої функції $m(x)$ використовується її апроксимація у вигляді $f(x) = \omega^T \varphi(x) + b$, при цьому відновлюючи залежність, отримується рішення (2.2):

$$\begin{bmatrix} 0 & 1_n^T \\ 1_n & \Omega + \frac{1}{\gamma} I_n \end{bmatrix} \begin{bmatrix} \hat{b} \\ \hat{\alpha} \end{bmatrix} = \begin{bmatrix} 0 \\ y \end{bmatrix}, \quad (2.2)$$

де $y = (y_1, \dots, y_n)^T$, $1_n = (1, \dots, 1)^T$, $\hat{\alpha} = (\hat{\alpha}_1, \dots, \hat{\alpha}_n)$, $\Omega_{kl} = \varphi(x_k)^T \varphi(x_l)$, для $k, l = 1, \dots, n$ – параметр регуляризації [10].

Результуюча LS-SVM модель має вигляд (2.3)-(2.5):

$$\widehat{y}_n(x) = \sum_{k=1}^n \hat{\alpha}_k K(x, x_k) + \hat{b}, \quad (2.3)$$

$$\hat{b} = \frac{1_n^T (\Omega + \frac{1}{\gamma} I_n)^{-1} y}{1_n^T (\Omega + \frac{1}{\gamma} I_n)^{-1} 1_n}, \quad (2.4)$$

$$\hat{\alpha} = \left(\Omega + \frac{1}{\gamma} I_n \right)^{-1} (y - 1_n \hat{b}), \quad (2.5)$$

де $K(x, x_k)$ – ядро скалярного добутку [10].

2.3 Запропонований метод прогнозування пасажиропотоку аеропорту

Запропонований метод прогнозування пасажиропотоку аеропорту базується на розглянутому методі LS-SVM, але при цьому враховується, що послідовність даних, які визначають кількість пасажирів, що було обслуговано в аеропорту, тільки частково впливають на кількість пасажирів, що буде обслуговано в аеропорту в майбутньому, а на це також впливають додаткові фактори. Для того щоб зменшити помилку у функції (2.1) потрібно визначити ці фактори.

Прогнозування буде відбуватися в короткостроковій та середньостроковій перспективі, тобто в якості базового періоду розглядається місяць та квартал відповідно.

У процесі дослідження до таких факторів було віднесено:

– зміна курсу валюти за період, що розглядається: Україна належить до країн, на економічну активність в яких значно впливає курс іноземної валюти, в якості якого в нашій країні розглядається курс американського долару, при цьому вплив високий, оскільки коливання достатньо високі;

– номер кварталу (місяця): на попит впливає також фактор сезонності, так найбільша кількість пасажирів обслуговується влітку і в основному припадає на 3 квартал, відповідно номер кварталу, на який відбувається прогнозування, буде визначати, на скільки тенденція до зміни пасажиропотоку за останні 3 квартали визначена загальною зміною пасажиропотоку, а на скільки – фактором сезонності.

Оскільки квартальний період вимагає накопичення даних за дуже довгий проміжок часу (десятки років), то застосовано також короткострокову перспективу, що може використовуватися для тактичного коригування показників, з якими відбувається робота.

Спочатку має бути накопичено набір даних, що включає кількість пасажирів, обслуговану за послідовність базових періодів, тобто за n місяців або m кварталів, та курс американського долара до гривні протягом аналогічних періодів, а також номер базового періоду (місяця, кварталу) для кожного значення.

Далі формуються вибірки даних (щомісячна та щоквартальна). Для прикладу розглянемо ідею для щоквартальної.

Перший екземпляр вибірки даних визначається шляхом взяття значень пасажиропотоку перших k кварталів, визначення номеру майбутнього, тобто $k+1$ кварталу, який розраховується як $(k+1) \bmod 4$, та обчислення коливання курсу валюти протягом k кварталів. Для визначення коливання обчислюється максимальне та мінімальне значення, а потім визначається частка, на яку максимальний курс перевищує мінімальний. Весь цей набір даних формує набір значень вхідних ознак екземпляра даних, на виході визначається кількість пасажирів у $k+1$ кварталі.

Далі шляхом зміщення на одну позицію формується другий екземпляр і т.д.

Модель, яка використовується в роботі методу, може бути представлена виразом (2.6):

$$y_{j+k+1} = f(y_{j+k}, y_{j+k-1}, \dots, y_j, (k+1) \bmod T, x_j^{j+k}), \quad (2.6)$$

де $y_{j+k+1}, y_{j+k}, y_{j+k-1}, \dots, y_j$ – кількість пасажирів у періоди часу $j+k+1, j+k, j+k-1, \dots, j$ відповідно;

x_j^{j+k} – коливання курсу валют протягом періоду часу з j до $j+k$;

T – кількість базових періодів у році (4 для кварталу, 12 для місяця).

Для отриманої вибірки даних застосовується метод LS-SVM, описаний у підрозділі 2.2.

У подальшому визначається екземпляр, який містить тільки значення вхідних ознак (значення вихідної ознаки для нього невідоме під час реального прогнозування або відоме під час тестування методу). За допомогою визначених на попередньому кроці параметрів обчислюється значення вихідної ознаки, що і повертається як результат роботи методу.

2.4 Висновки за розділом 2

Було проаналізовано існуючі методи, які можуть використовуватися для прогнозування пасажиропотоку аеропорту, розглянуто метод опорних векторів та метод опорних векторів з квадратичною функцією втрат. На основі проаналізованих засобів було запропоновано метод прогнозування пасажиропотоку аеропорту, який ґрунтується на використанні методу опорних векторів з квадратичною функцією втрат, використовує в якості вхідних ознак пасажиропотік за минулі періоди, коливання курсу валют за цей період та номер базового періоду.

3 ОПИС ПРОГРАМИ

За результатами проведеного аналізу проблеми було вирішено розробляти програмне забезпечення у вигляді вебзастосунку, що має універсифікувати роботу в межах всього аеропорту. Тому в результаті під час визначення вимог, вибору мови програмування використовувалась саме логіка роботи з вебзастосунком.

3.1 Визначення вимог до вебзастосунку

Вебзастосунок має забезпечувати виконання таких функціональних вимог:

- додавання даних про нові літаки, які обслуговуються в аеропорту;
- змінювання даних літаків, які вже обслуговуються в аеропорту;
- внесення даних про технічне обслуговування літака;
- фіксування результатів технічного обслуговування літака;
- додавання даних авіаліній, які працюють з аеропортом;
- змінювання даних авіаліній;
- відображення розкладу рейсів;
- відображення табло аеропорту;
- внесення даних про нові рейси, які виконуються з аеропорту;
- фіксування фактично виконаних польотів;
- реєстрація пасажирів;
- відображення зареєстрованих на політ;
- внесення даних багажу пасажирів;
- прогнозування кількості пасажирів;
- внесення даних членів екіпажу;
- внесення даних про польоти екіпажу;
- пошук пасажирів, літаків, рейсів;
- перегляд, редагування і вилучення записів про пасажирів;

- перегляд, редагування і вилучення записів про реєстрацію пасажирів;
- перегляд, редагування і вилучення записів про зареєстрований багаж;
- перегляд, редагування і вилучення записів про пасажирів;
- перегляд, редагування і вилучення записів про рейси;
- перегляд і вилучення записів про літаки;
- перегляд, редагування і вилучення записів про фактичні польоти літаків;
- перегляд, редагування і вилучення записів про технічне обслуговування;
- перегляд, редагування і вилучення записів про персонал;
- перегляд, редагування і вилучення записів про членів екіпажу;
- перегляд і вилучення записів про авіалінії;
- формування звіту про пасажиропотік аеропорту;
- формування звіту про роботу аеропорту;
- авторизація;
- внесення облікових записів персоналу.

Функції редагування і вилучення необхідні для запобігання внесенню помилкових даних.

Усі функції використовуються персоналом аеропорту. Для різних видів персоналу передбачено різні функції. Виділено такі види:

- менеджер аеропорту;
- менеджер авіалінії;
- менеджер з технічного обслуговування.

Діаграму прецедентів, що відповідає описаним функціям програми, сформульовано на рис. 3.1.

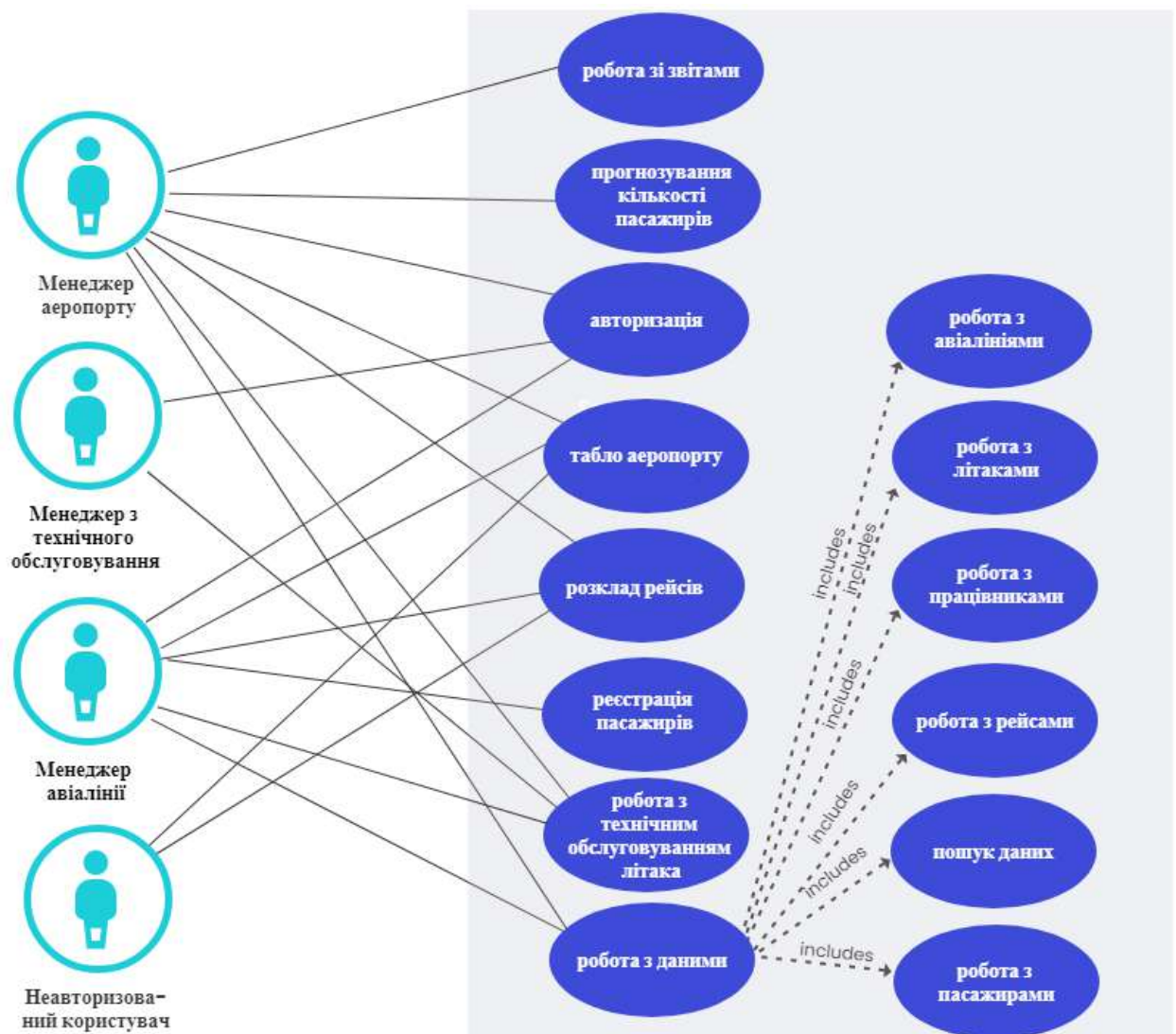


Рисунок 3.1 – Діаграма прецедентів роботи з системою

3.2 Вибір засобів проєктування та реалізації програми

PHP відома як мова програмування загального призначення, це серверна мова сценаріїв, яка в основному використовується для розробки вебсайтів, структура PHP також полегшує веброзробку, фреймворки PHP допомагають повторно використовувати той самий код, і немає необхідності писати довгий та складний код для вебзастосунків, фреймворки PHP в основному з відкритим кодом і можуть бути легко використані, PHP була написана в основному мовою C, але деякі компоненти написані також мовою C++, загалом відносно цієї мови можна виділити наступні її основні позитивні характеристики:

– відкритий код: PHP є відкритим і безкоштовним рішенням, що допомагає розробникам швидко та легко встановити його для використання, існує багато фреймворків PHP, і розробник може вибрати будь-який фреймворк для роботи, всі функції та інструменти будуть легко надані для обраного фреймворку, оскільки це рішення з відкритим кодом, це дозволяє швидко створювати систему за допомогою PHP та пришвидшувати веброзробку за допомогою легкого надання інструментів та інших функцій;

– незалежність від платформи: PHP в основному підтримується усіма операційними системами, такими як Windows, Unix, Linux тощо, розроблені вебзастосунки, засновані на PHP, можна легко запускати на будь-якій платформі, може бути легко інтегрована з іншою мовою програмування та базою даних, і немає потреби в повторній розробці, це допомагає заощадити багато зусиль та витрат;

– простота і легкість: PHP проста і легка у вивченні та кодуванні, в основному це організований і чистий код, що також допомагає новим розробникам, командні функції PHP можна легко вивчити та зрозуміти, загалом розробник, який знає будь-яку мову програмування, може легко працювати на PHP, навчитися просто, оскільки крива навчання невелика, при цьому мій власний практичний досвід тільки збільшує значимість даного фактору, синтаксис простий і гнучкий у використанні;

– база даних: PHP легко підключається до бази даних і надійно з'єднується з нею, має вбудований модуль, який використовується для легкого підключення до бази даних, існує багато вебзастосунків, які потребують потужної мови програмування з хорошою системою управління базами даних, PHP та підключення до бази даних вирішують мету розробки вебзастосунків, що також скорочує час підключення до системи управління базами даних, декілька баз даних можна інтегрувати з PHP;

– швидкість: мова PHP відома як найшвидша мова програмування порівняно з іншими, програми PHP можна легко завантажити, навіть за повільної швидкості Інтернету та передачі даних, інші програми вимагають

багато часу для підключення бази даних та отримання даних після виконання певних запитів до бази даних, PHP не стикається з цією проблемою, і вебсайт завантажується дуже легко та швидко, така швидкість PHP надає розробникові можливість розробляти вебзастосунки мовою програмування PHP;

- обслуговування коду: фреймворк PHP в основному використовується для спрощення розробки вебзастосунків та автоматичного обслуговування коду;

- підтримка – ця перевага PHP має велику підтримку в Інтернеті та спільноту, що допомагає новим розробникам допомогти у написанні коду та розробці вебзастосунків, документація, надана на офіційному сайті, допомагає використовувати різні функції PHP та її фреймворк, останні оновлення PHP випускає своєчасно, щоб розробнику було зручніше розробляти вебзастосунки;

- тестування: вебпрограми на основі PHP можна легко протестувати, блок PHP використовується для швидкого та легкого виконання модульного тестування, це також допомагає програмістам писати тестові кейси та виконувати тестування плавно, для вебзастосунків на основі PHP розробникам не потрібно писати додатковий код, фреймворки PHP допомагають в автоматизації різних завдань;

- безпека: вбудовані функції та інструменти фреймворків спрощують захист вебпрограм від зовнішніх атак та загроз безпеці, загрози безпеці можуть бути такими, як введення SQL, підробка даних, для захисту від цих загроз розробники використовують PHP-фреймворки для розробки вебзастосунків;

- стійкість: PHP є також стабільною мовою в порівнянні з іншими мовами програмування, існує давно, відповідно проблеми та помилки виправлялись протягом тривалого періоду часу, що робить останні версії мови дуже стабільними [11]-[13].

За обраними факторами прийняття рішень було сформульовано порівняння в табл. 3.1.

Таблиця 3.1 – Порівняння мов PHP та Python для веброзробки

Фактор	PHP	Python
Призначення	Загальне, але для веброзробки	Загальне
Веброзробка	+	+
Безкоштовність	+	+
Робота з базами даних	Широкий вибір (більше 20 систем)	Є підтримка

Для розробки програмного проєкту було обрано мову PHP, оскільки вона є основним вибором для веброзробки, має широкі засоби для цього, окрім того є простою у використанні, вивченні за необхідності, безкоштовною, а до того ж власний досвід веброзробки ґрунтується саме на цій мові.

Системою керування базами даних було обрано MySQL, оскільки вона є з одного боку безкоштовною, а з іншого достатньо потужною, надійною, зарекомендувала себе в практичних проєктах.

3.3 Проєктування бази даних

За результатами аналізу предметної області було визначено основні сутності предметної області, які потім було представлено у вигляді таблиць, а також додано до них таблиці, що реалізують зв'язки «багато-до-багатьох».

Таблиця Registration визначає реєстрацію пасажирів на рейс. За допомогою даної таблиці відбувається фактичне визначення в системі пасажирів, адже клієнт може забронювати місце, купити квиток, але фактично не летіти, а для прогнозування потрібно використовувати фактичні значення.

Таблиця Registration відповідає сутності, що містить такі характеристики:

- номер реєстрації id;
- пасажир passnum;
- політ flightnum;
- місце placenum;
- клас type;
- вартість квитка price.

Поле passnum є зовнішнім ключем з таблиці Passenger, визначає, який саме пасажир виконав реєстрацію.

Поле flightnum є зовнішнім ключем з таблиці Flight, визначає, на який саме політ виконану дану реєстрацію.

Таблиця Route визначає маршрут та особливості його виконання. Мається на увазі саме маршрут, яким він виконується конкретною авіакомпанією. Таблиця відповідає сутності, що містить наступні характеристики:

- номер id;
- тип type;
- маршрут places;
- тип літака planetype;
- дата і час вильоту departtime;
- час польоту flighttime;
- авіакомпанія line;
- розклад за днями тижня timetable;
- кількість квитків ticketsnum;
- кількість членів екіпажу crewnum;
- потреба пального fuel;
- дата завершення польотів за рейсом enddate;
- номер рейсу number.

Поле `line` є зовнішнім ключем з таблиці `Airline`, визначає, яка саме авіакомпанія виконує рейс за цим маршрутом.

Таблиця `Flight` визначає екземпляри конкретного польоту, який в межах маршруту, тобто рейсу, виконується. Тобто в цій таблиці зберігається інформація про те, як фактично було виконано рейс (в який час час). Таблиця відповідає сутності, що містить наступні характеристики:

- номер `id`;
- рейс `gate`;
- літак `planenum`;
- дата і час вильоту `departtime`;
- дата і час прильоту `arrivtime`.

Поле `gate` є зовнішнім ключем з таблиці `Route`, визначає, за яким саме маршрутом виконується політ.

Поле `planenum` є зовнішнім ключем з таблиці `Plane`, визначає, який саме літак виконує політ.

Таблиця `Passenger` дозволяє зберігати інформацію про пасажирів, що відзначаються наступними характеристиками:

- номер `id`;
- повне ім'я пасажирів (прізвище, ім'я та по батькові) `name`;
- паспорт (серія та номер) `passport`;
- номер телефону `phone`;
- національність `nationality`.

Таблиця `Luggage` призначена для збереження даних про багаж пасажирів. Багаж має такі характеристики:

- номер `id`;
- реєстрація `registr`, в межах якої перевозиться багаж (власник);
- тип `type`;
- вага `weight`;
- довжина `length`;
- ширина `width`;

- висота height.

Поле registr є зовнішнім ключем з таблиці Registration, визначає, якій саме реєстрації належить даний багаж.

Таблиця Airline визначає інформацію про авіалінії, зберігаючи про них такі характеристики:

- номер id;
- назва name;
- країна country;
- номер реєстрації regnum;
- опис description;
- дата створення creattime.

Таблиця Plane містить дані про кожен конкретний літак, який реалізує політ за заданим маршрутом в кожній конкретній ситуації, при цьому літак відзначається наступними характеристиками:

- номер id;
- тип type;
- виробник manuf;
- модель model;
- номер реєстрації regnum;
- серійний номер sernum;
- дата виробництва prodtdate;
- максимальна швидкість speed;
- кількість пасажирів passq;
- обсяг пального fuel;
- стан state.

Таблиця TO містить дані про технічне обслуговування літака, що визначається такими характеристиками:

- номер id;
- літак planenum;
- тип type;

- дата і час початку starttime;
- дата і час завершення endtime;
- відповідальний personnum;
- опис description;
- результат result.

Поле planenum є зовнішнім ключем з таблиці Plane, визначає, який саме літак проходив технічне обслуговування.

Таблиця Person містить дані про персонал, який обслуговує роботу аеропорту та літаків, включає наступні характеристики:

- номер id;
- повне ім'я (ім'я, прізвище, по батькові) name;
- паспорт passport;
- адреса проживання regaddr;
- ідентифікаційний код idcode;
- завершення контракту contractend;
- дата народження birthday;
- посада position;
- номер телефону phone;
- дата початку роботи contractbegin;
- опис description;
- логін login;
- пароль password.

Таблиця FlightCrew визначає екіпаж літака під час конкретного польоту, необхідна для реалізації зв'язку багато-до-багатьох між таблицями Flight та Crew, містить такі характеристики:

- номер id;
- номер члену екіпажу crewnum;
- номер польоту flightnum.

Поле crewnum є зовнішнім ключем з таблиці Crew, визначає, який саме член екіпажу виконує політ.

Поле `flightnum` є зовнішнім ключем з таблиці `Flight`, визначає, в якому саме польоті бере участь член екіпажу.

Схема бази даних сформульована у вигляді рис. 3.2.

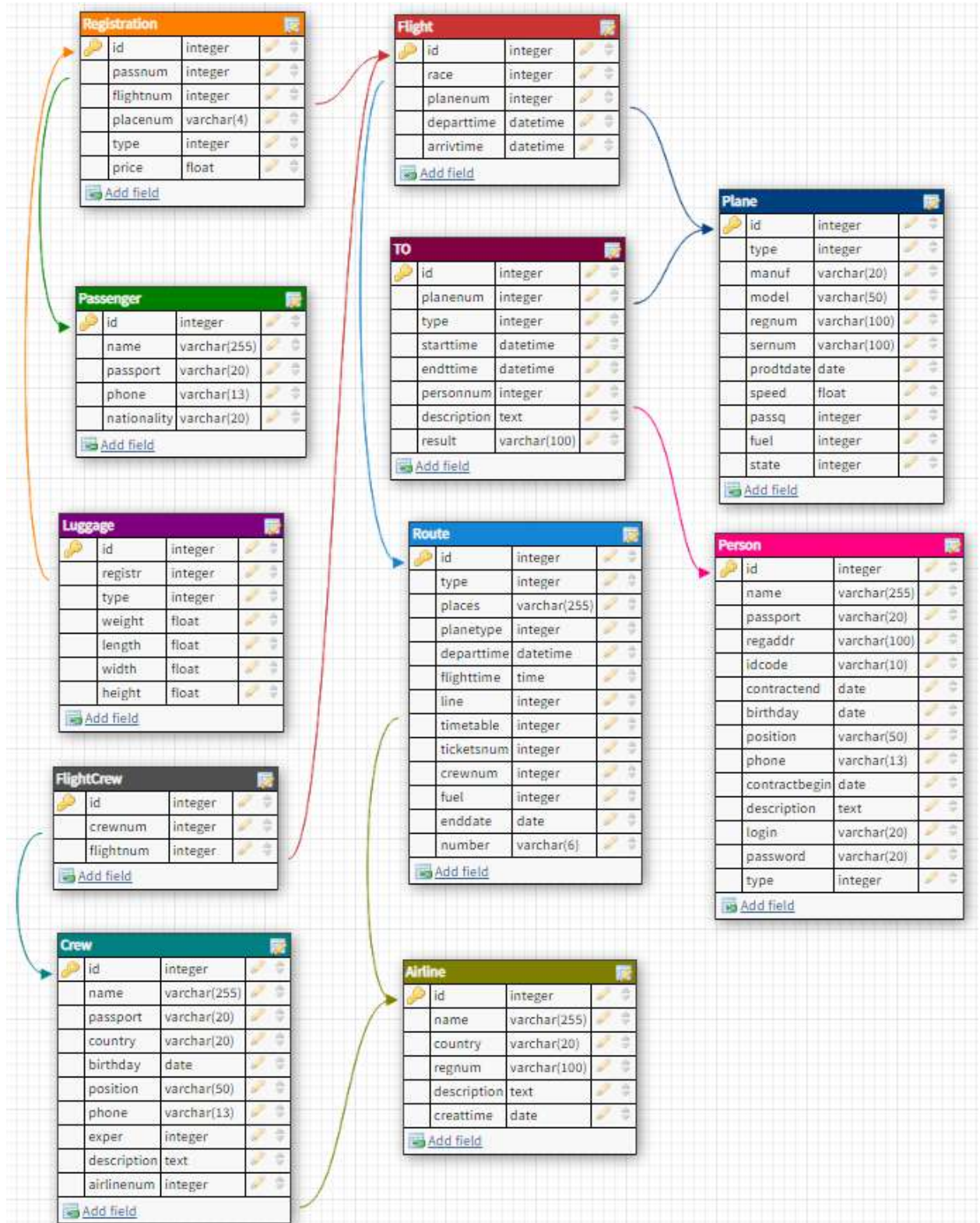


Рисунок 3.2 – Схема бази даних

Таблиця Crew містить дані про екіпаж літака, включає наступні характеристики:

- номер id;
- повне ім'я (ім'я, прізвище, по батькові) name;
- паспорт passport;
- країна country;
- дата народження birthday;
- посада position;
- номер телефону phone;
- години нальоту exper;
- опис description;
- авіалінія airlinenum.

Поле airlinenum є зовнішнім ключем з таблиці Airline, визначає, з якою авіалінією працює даний учасник екіпажу.

3.4 Висновки за розділом 3

Було визначено функціональні вимоги до програми, що розробляється, сформульовано діаграму прецедентів для роботи менеджера аеропорту, менеджера з технічного обслуговування, менеджера авіалінії та неавторизованого користувача з програмою. Було виконано порівняння засобів реалізації програми (мов PHP і Python), в результаті чого мову програмування PHP та систему керування базами даних MySQL було обрано для реалізації. Було запроєктовано базу даних, результат проєктування сформульовано у вигляді схеми бази даних.

4 РОЗРОБКА ПРОГРАМИ

4.1 Опис реалізованої структури програми

Для реалізації описаної в розділі 3 функціональності було забезпечено створення програмного проєкту з відповідним набором модулів.

До складу програмного проєкту входить папка `assets`, що включає до своєї структури такі папки:

- `css` – папка, що містить каскадні таблиці стилів застосунку;
- `js` – папка, що містить скрипти `javascript`;
- `sass` – папка, що містить файли скриптової метамови `Sass`;
- `webfonts` – папка, що містить шрифти, які використовуються в застосунку.

Для створення застосунку використано вільний шаблон [14].

Для реалізації вебзастосунку було створено наступний набір скриптів мовою `php`, основні з яких перелічено нижче.

Скрипт `menu.php` реалізує створення меню, яке є основою побудови всіх вебсторінок застосунку.

Скрипт `index.php` визначає базову сторінку вебзастосунку, що має відкриватися користувачу за замовчуванням. Скрипт виконує відображення табло аеропорту.

Скрипт `timetable.php` забезпечує виконання відображення розкладу рейсів.

Скрипт `registration.php` забезпечує виконання реєстрації пасажирів на рейс.

Скрипт `tech.php` забезпечує виконання внесення даних щодо технічного обслуговування літака.

Папка `predict` містить засоби виконання прогнозування пасажиропотоку. Скрипт `prediction.php` всередині неї забезпечує виконання відображення даних прогнозування пасажиропотоку аеропорту. Скрипт `method.php` безпосередньо є основою прогнозування і реалізує

запропонований метод. Бібліотечні засоби для реалізації методу знаходяться всередині даної папки в папці `library` і визначаються бібліотекою `libSVM` [15].

Скрипт `login.php` виконує авторизацію користувача.

Для реалізації пошуку виділено окрему папку `search`. Скрипт `search.php` всередині неї забезпечує базовий пошук, який звертається до інших скриптів пошуку. Скрипт `searchroute.php` виконує пошук рейсів.

Папка `report` містить засоби формування звітів. Скрипт `reportp.php` всередині папки виконує формування звіту про пасажиропотік аеропорту. Скрипт `reports.php` виконує формування статистичного звіту щодо роботи аеропорту.

Папка `data` містить набір скриптів, які дозволяють працювати безпосередньо з таблицями бази даних, надаючи засоби перегляду даних, а також їх редагування, додавання і вилучення. Деякі з цих скриптів описані нижче.

Скрипт `airlines.php` реалізує відображення переліку авіаліній та даних про них. Скрипт `addairline.php` виконує додавання нової авіакомпанії. Скрипт `editairline.php` виконує редагування даних про авіакомпанію. Скрипт `deleteairline.php` виконує вилучення даних авіакомпанії, використовується у випадку помилкового створення запису в таблиці.

Скрипт `planes.php` реалізує відображення переліку літаків, а скрипти `addplane.php`, `editplane.php` та `deleteplane.php` виконують відповідно додавання, редагування та вилучення літаків.

Скрипт `routes.php` надає засоби перегляду маршрутів, за якими виконується політ, скрипти `addroute.php`, `editroute.php` та `deleteroute.php` виконують відповідно додавання, редагування та вилучення маршрутів.

Скрипт `person.php` реалізує відображення переліку персоналу аеропорту, аналогічно передбачено скрипти для додавання, редагування та вилучення даних.

Скрипт `crew.php` відповідно реалізує відображення переліку членів екіпажу, скрипт `passengers.php` реалізує відображення переліку пасажирів за рейсом.

На рис. 4.1 сформульовано фізичне розгортання описаної програмної системи.

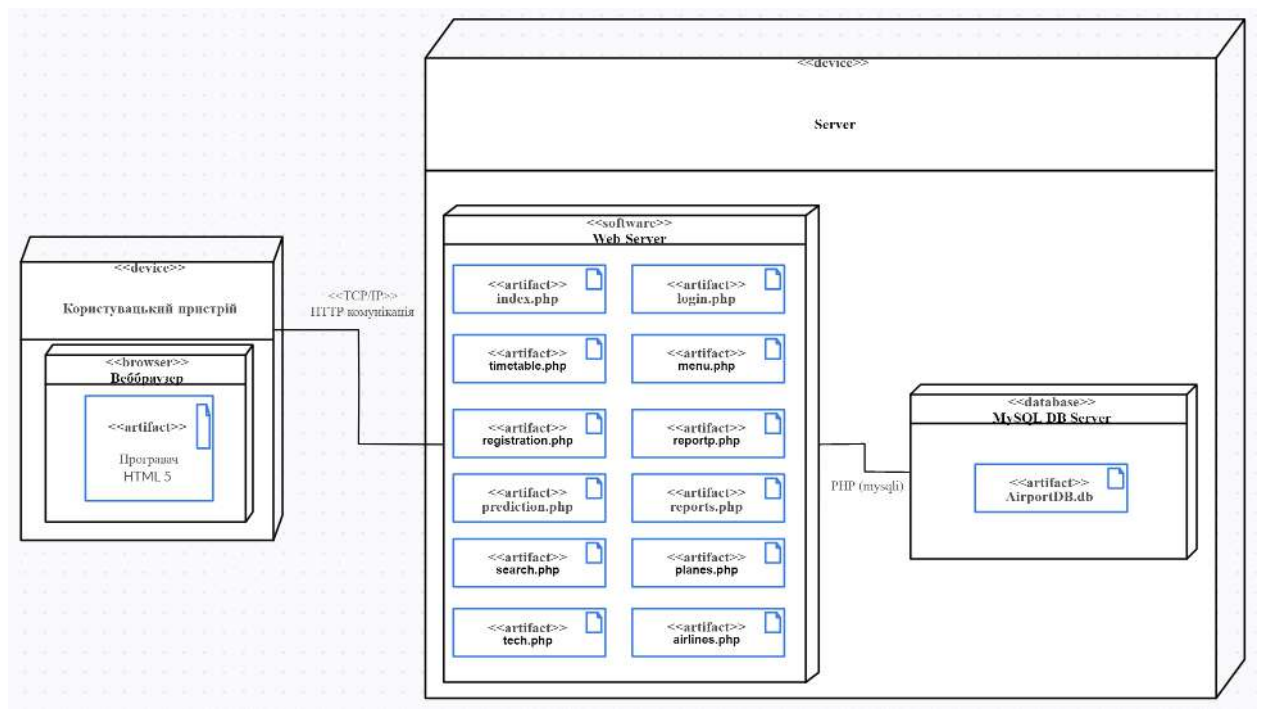


Рисунок 4.1 – Діаграма розгортання

4.2 Опис процесу прогнозування пасажиропотоку аеропорту

Прогнозування пасажиропотоку аеропорту в програмі виконується на основі базової процедури, яка включає в себе на найнижчому етапі реєстрацію пасажирів. Відповідні дані про реєстрацію накоплюються, а потім (коли завершується квартал) акумулюються і використовуються для прогнозування на наступний період (квартал), для чого застосовується запропонований метод. Після отримання результатів вони зберігаються на сервері у вигляді json-файлу, а потім кожного разу під час звернення менеджерів аеропорту до відповідного розділу програми – відтворюються. Візуально дану процедуру сформульовано на рис. 4.2.



Рисунок 4.2 – Схема функціонування процесу прогнозування пасажиропотоку аеропорту в програмі

4.3 Опис розроблених скриптів

Для розробки скриптів було використано процедурний підхід.

У скрипті `menu.php` визначено меню вебзастосунку, що використовується на кожній його сторінці. Дане меню має різний вигляд в залежності від ролі, яку виконує користувач.

Так пункт меню “Розклад” відображається всім окрім менеджера з технічного обслуговування:

```

<?php
    if(!isset($_SESSION["techmanager"])) {
?>
<li><a href="timetable.php">ПОЗКЛАД</a></li>
<?php endif ?>

```

Роль кожного авторизованого користувача визначається відповідною змінною сесії:

- techmanager – менеджер з технічного обслуговування;
- almanager – менеджер авіалінії;
- armanager – менеджер аеропорту.

Визначення можливості авторизації для неавторизованого користувача відбувається наступним чином:

```

<?php
    if(!isset($_SESSION["login"])) {
?>
<li><a href="login.php">АВТОРИЗАЦІЯ</a></li>
<?php }
    else {
        echo "<li><a href='logout.php'>ВИЙТИ</a></li>";
    }
?>

```

Робота скриптів побудована на витяганні даних з бази даних та використанні отриманих результатів під час формування сторінки. Наприклад, у скрипті index.php відбувається формування даних, що виводяться у вигляді табло аеропорту. Табло формується у двох варіантах: на приліт та на виліт. Для формування варіанту для прильоту виконується SQL-запит до бази даних з витяганням даних з таблиць авіаліній та маршрутів,

при цьому реалізується перевірка на те, що поточна дата раніше за останній день виконання відповідного рейсу, а день тижня для поточного дня співпадає з днем виконання рейсу:

```
$alldata = mysqli_query($db, "SELECT Airline.name, Route.places,
Route.departtime, Route.flighttime, Route.id FROM `Route` INNER JOIN
'Airline' ON Route.line = Airline.id WHERE date('m.d.y', Route.enddate) >= $cd
AND Route.places NOT LIKE 'Запоріжжя%' AND $dd =
var_dump((bool)($Route.timetable & (1 << ($dd - 1))))");
```

Відповідно після отримання даних відбувається їх розбір з виділенням кожного окремого запису та фіксування значень кожного окремого параметра в стовпці таблиці:

```
while($oned = mysqli_fetch_array($alldata)){ ?>
<tr>
<td><?php echo $oned[0]; ?></td>
<td><?php echo $oned[1]; ?></td>
<td><?php echo $oned[2]; ?></td>
<td><?php echo $oned[3]; ?></td>
```

Окремо відбувається заповнення позицій в таблиці, що відповідає стану польоту, для цього виконується окремий запит до бази даних, в якому витягаються дані польотів поточного дня, що відповідають вказаному рейсу, відповідно якщо такого запису немає, то політ вважається запланованим, якщо заповнений тільки час вильоту, то політ вважається таким, що знаходиться в очікуванні, а якщо зафіксовано і час вильоту, і прибуття, то політ вважається виконаним:

```

<td><?php $res = mysqli_query($db, "SELECT departtime, arrivtime
FROM `Flight` WHERE Flight.race = $oned[4] AND CAST(Flight.departtime AS
DATE) = CAST(GETDATE() AS DATE");
if (mysqli_num_rows ($res) == 0) {
    echo "<td>Запланований</td>";
}
else {
    $p = mysqli_fetch_array($res);
    if (!$p['arrivtime']) {
        echo "<td>Очікується, вилетів о ".$p['departtime']. "</td>";
    }
    else {
        echo "<td>Прилетів о ".$p['arrivtime']. "</td>";
    }
}
}

```

При відображенні даних таблиць, коли потенційно може бути отримано велику кількість даних, відбувається розділення результатів за сторінками. Наприклад, в скрипті flights.php для цього, по-перше, створено поле пошуку рейсу, для того щоб відображати тільки польоти за вказаним номером рейсу, це дозволяє звужити простір даних, що відображаються. По-друге, передбачено формування набору даних, що відповідає окремій сторінці. На кожній сторінці відображається тільки 10 результатів.

Визначення номеру сторінки відбувається через GET-запит, де вказується цей номер:

```

<?php
if (isset($_GET['pnum'])) {
    $pnum = $_GET['pnum'];
} else {

```

```

        $pnum = 1;
    }
}

```

Обчислення номеру першого запису, що відображається, реалізується наступним чином, при цьому `flightspp` визначає кількість результатів на сторінці (10):

```
$stepfl = ($pnum-1) * $flightspp;
```

Отримання пулу даних, що відповідає вказаному діапазону, реалізується наступним чином:

```

$alldata = mysqli_query($db, "SELECT Flight.id, Route.number,
Route.places, Plane.manuf, Plane.model, Plane.regnum, Flight.departtime,
Flight.arrivtime FROM `Flight` LEFT JOIN 'Route' ON Flight.race = Route.id
LEFT JOIN 'Plane' ON Flight. planenum = Plane.id WHERE Route.number LIKE
$_POST['route-num'].'%' LIMIT $stepfl, $flightspp");

```

Формування посилання для переходу на наступну сторінку зокрема реалізується наступним чином:

```

<a href="<?php if($pnum >= $flps){ echo '#'; } else { echo
"?pnum=" . ($pnum + 1); } ?>" class="page">Вперед</a>

```

4.4 Висновки за розділом 4

Описано структуру розробленого вебзастосунку з приведенням основних реалізованих скриптів. Створено діаграму розгортання.

Описано процес прогнозування пасажиропотоку аеропорту зі створенням схеми функціонування даного процесу.

Описано логіку реалізації розроблених скриптів мовою PHP.

5 ЕКСПЛУАТАЦІЯ ТА ТЕСТУВАННЯ ПРОГРАМИ

5.1 Призначення програми

Вебзастосунок призначений для виконання обліку діяльності аеропорту, включаючи роботу з авіалініями, виконанням рейсів, технічним обслуговуванням, що в результаті дозволяє автоматизовано виконувати прогнозування пасажиропотоку на основі автоматично зібраних даних щодо реєстрації пасажирів за минулі періоди.

5.2 Опис роботи з програмою

Для запуску вебзастосунку потрібно на сервері запустити вебсервер Apache та службу MySQL, попередньо перевірити встановлення інтерпретатора мови програмування PHP. Після цього можна буде підключатися до вебзастосунку з клієнтських комп'ютерів. Для цього потрібно ввести в адресному рядку веббраузера адресу вебсервера, тоді відкриється початкова сторінка вебсайту (рис. 5.1).

З вебзастосунком може працювати неавторизований користувач. Саме вигляд вебзастосунку для нього продемонстровано на рис. 5.1.

Користувач може скористатися меню, що складається з пунктів “Табло”, “Розклад” та “Авторизація”.

Для того щоб отримати більш широкий набір функцій, потрібно авторизуватися. Склад меню буде залежати від того, якого типу обліковий запис авторизовано.

На рис. 5.1 представлено табло аеропорту, тобто перелік рейсів за поточний день, у даному випадку на прибуття до аеропорту. Для відображення рейсів, що відправляються за аеропорту, потрібно натиснути на кнопку “Виліт”, після цього буде відображено рейси, що відправляються, у такій самій таблиці, а кнопка зміниться на “Приліт” для повернення до початкового вигляду.

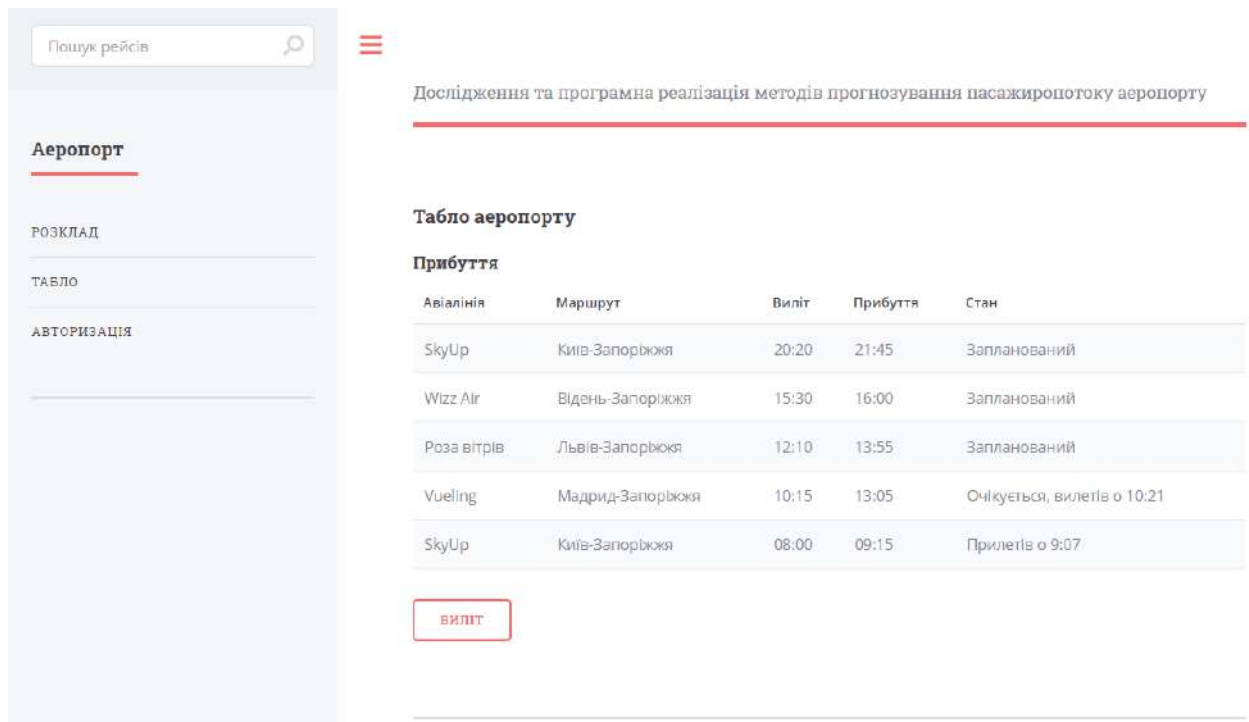


Рисунок 5.1 – Табло аеропорту

Табло містить інформацію про авіалінію, маршрут рейсу, час вильоту та прибуття рейсу за розкладом, а також дані стану. Усі рейси відсортовані за часом прибуття, від найпізнішого до самого раннього. Стан рейсу містить або запланований, якщо літак ще не вилетів, або очікуваний (з зазначенням часу вильоту), якщо літак тільки вилетів, але ще не прибув, або позначку про те, що літак вже прилетів з зазначенням фактичного часу приземлення.

На рис. 5.2 продемонстровано приклад перегляду розкладу рейсів на вказану дату.

На цій сторінці користувач має обрати дату, для чого використовується календар. Після натискання на кнопку “Відобразити” буде заповнено таблицю, розташовану нижче, де буде виведено рейси за вказану дату. Це рейси вильоту з аеропорту та прибуття в аеропорт. Для кожного рейсу також вказується, до якої дати він буде виконуватися. Відображаються як рейси, які вже було виконано, якщо обрано минулу дату, так і рейси, які мають виконуватися за розкладом, якщо було обрано майбутню дату.

Пошук рейсів

Аеропорт

РОЗКЛАД

ТАБЛО

АВТОРИЗАЦІЯ

Дослідження та програмна реалізація методів прогнозування пасажиропотоку аеропорту

Розклад рейсів

Оберіть дату

15.12.2020

ВІДОБРАЗИТИ

Номер	Авіалінія	Маршрут	Виліт	Прибуття	До
SK 328	SkyUp	Братислава-Запоріжжя	19:30	20:30	15.03.2021
CU 937	MAU	Запоріжжя-Прага	18:00	20:45	15.06.2021
WS 145	Lot	Варшава-Запоріжжя	16:45	17:50	20.11.2021
MI 491	Мотор Січ	Мінськ-Запоріжжя	14:40	16:10	17.02.2021
UK 028	MAU	Запоріжжя-Київ	07:00	08:00	10.04.2021

Рисунок 5.2 – Перегляд розкладу рейсів

Будь-яка робота з даними побудована на основі стандартної процедури взаємодії. Спочатку користувачу під час відкриття відповідного розділу пропонується таблиця зі всіма наявними даними та засобами пошуку. Наприклад, на рис. 5.3 продемонстровано приклад для роботи з рейсами в режимі для менеджера аеропорту.

У даному випадку користувач може задати потрібний йому номер рейсу, натиснути на кнопку “Обрати”, після чого в таблиці буде відображено тільки заданий номер рейсу (або якщо вказано частину номеру, то всі відповідні). Звідси менеджер може переглянути всі потрібні йому рейси, їх дані, при чому для переходу між наборами рейсу (в таблиці відображається тільки 10 записів) призначені кнопки, розташовані під таблицею, що дозволяють переходити між порціями даних.

Пошук рейсів

Аеропорт

Розклад

Таблиця

Організація

Прогнозування

Рейси

Звіт

Робота з даними

Вихід

Додати

ID	Номер	Тип	Маршрут	Літак	Виліт	Час	Авіакомпанія	Розклад	Місце	Екіпаж	Паливо	Дні	Редагувати	Вилучити
31	MI 491	П	Мінськ-Запоріжжя	Yk40	14:40	1:30	Motor CH	Вт 1т	50	4	2000	17.02.2021	РЕДАГУВАТИ	ВИЛУЧИТИ
32	MI 492	П	Запоріжжя-Мінськ	Yk40	10:00	1:30	Motor CH	Ср 1т	50	4	2000	17.02.2021	РЕДАГУВАТИ	ВИЛУЧИТИ
33	IE 761	П	Лондон-Запоріжжя	Boeing 737	18:00	3:10	Easyjet	Пн	162	5	10000	20.04.2021	РЕДАГУВАТИ	ВИЛУЧИТИ
34	IE 752	П	Запоріжжя-Лондон	Boeing 737	22:00	3:05	Easyjet	Пн	162	5	10000	20.04.2021	РЕДАГУВАТИ	ВИЛУЧИТИ
35	MI 491	П	Київ-Запоріжжя	Boeing 737	0:00	1:30	Air Moldova	Пт	100	5	2500	10.01.2021	РЕДАГУВАТИ	ВИЛУЧИТИ
36	MI 492	П	Запоріжжя-Київ	Boeing 737	18:00	1:30	Air Moldova	Пт	100	5	2500	10.01.2021	РЕДАГУВАТИ	ВИЛУЧИТИ
37	SK 326	П	Братислава-Запоріжжя	Boeing 737	19:30	1:00	SkyUp	Вт 4т	162	5	3000	15.03.2021	РЕДАГУВАТИ	ВИЛУЧИТИ
38	SK 329	П	Запоріжжя-Братислава	Boeing 737	22:30	1:00	SkyUp	Вт 4т	162	5	3000	15.03.2021	РЕДАГУВАТИ	ВИЛУЧИТИ
39	RS 082	П	Запоріжжя-Берлін	Embraer	06:00	2:15	MAU	Пн	92	4	4000	15.05.2021	РЕДАГУВАТИ	ВИЛУЧИТИ
40	RS 083	П	Берлін-Запоріжжя	Embraer	10:00	2:00	MAU	Пн	92	4	4000	15.05.2021	РЕДАГУВАТИ	ВИЛУЧИТИ

ПЕРША

НАЗАД

ВПЕРЕД

ОСТАННЯ

Рисунок 5.3 – Керування рейсами

Якщо на цій сторінці натиснути на кнопку “Додати”, то відкриється сторінка внесення даних нового рейсу (рис. 5.4).

Для нового рейсу потрібно вказати його номер, тип, маршрут, тип літака, час відправлення і прибуття, обрати з переліку авіаліній, вказати кількість членів екіпажу та необхідний обсяг палива, обрати дату завершення польотів за маршрутом, а також обрати розклад, для чого проставити прапорці біля тих днів тижня, за якими будуть виконуватися польоти.

Для кожного запису таблиці (рейсу) можна обрати кнопку “Редагувати” або “Вилучити” для редагування внесених даних або вилучення помилково створеного запису.

Приклад редагування даних літака продемонстровано на рис. 5.5. На цій сторінці відображається номер запису з таблиці бази даних, але він є неактивним (не можна змінити значення).

Пошук рейсів

Аеропорт

РОЗКЛАД

ТАБЛО

ОБСЛУГОВУВАННЯ

ПРОГНОЗУВАННЯ

ПОШУК

ЗВІТИ

РОБОТА З ДАНИМИ

ВИЙТИ

Додати рейс

Номер рейсу

Тип

Маршрут

Тип літака

Час відправлення

Час прибуття

Авіалінія

Кількість членів екіпажу

Необхідний обсяг палива

Розклад

Неділя

Понеділок

Вівторок

Середа

Четвер

П'ятниця

Субота

Дата завершення польотів

ЗБЕРЕГТИ

Рисунок 5.4 – Додавання нового рейсу

Пошук рейсів

Аеропорт

РОЗКЛАД

ТАБЛО

ОБСЛУГОВУВАННЯ

ПРОГНОЗУВАННЯ

ПОШУК

ЗВІТИ

РОБОТА З ДАНИМИ

ВИЙТИ

Редагування даних літака

Номер

Тип

Виробник

Модель

Номер реєстрації

Серійний номер

Дата виробництва

Максимальна швидкість

Кількість пасажирів

Обсяг пального

Стан

ЗБЕРЕГТИ

Рисунок 5.5 – Редагування даних літака

На рис. 5.6 продемонстровано приклад роботи менеджера з технічного обслуговування. Він може обрати потрібний літак за допомогою пошуку, а

нижче в таблиці для всіх відповідних літаків виводяться дані про виконане технічне обслуговування, які можна відредагувати або вилучити (якщо пройшло менше доби з внесення запису), окрім того можна створити новий запис за допомогою кнопки “Додати”.

Пошук рейсів

Аеропорт

ОБСЛУГОВУВАННЯ

Вийти

Дослідження та програмна реалізація методів прогнозування пасажиропотоку аеропорту

Технічне обслуговування

Вкажіть дані літака

Виробник

Серійний номер

або реєстраційний номер

Реєстраційний номер

ОБРАТИ

ДОДАТИ

ID	Тип	Літак	Сер. ном.	Реєстр. ном.	Початок	Завершення	Відповідальний	Результат	Редагувати	Вилучити
2	C-check (C2)	Boeing 737-8F2	37811	KP-JBA	15.09.2020 08:00	27.09.2020 17:00	Черненко А.П.	Готовий до експлуатації	РЕДАГУВАТИ	
51	Transit check	Boeing 737-8F2	37811	KP-JBA	18.11.2020 15:00	18.11.2020 15:15	Урусов П.І.	Дефектів не виявлено	РЕДАГУВАТИ	
62	Daily check	Boeing 737-8F2	37811	KP-JBA	25.11.2020 15:00	25.11.2020 15:30	Крайній І.І.	Дефектів не виявлено	РЕДАГУВАТИ	ВИЛУЧИТИ

Рисунок 5.6 – Робота менеджера з технічного обслуговування

Визначені наступні типи технічного обслуговування:

- transit check – виконується перед вильотом, який відбудеться не пізніше ніж за 3 години після посадки;
- daily check – перевірка, що виконується кожні 2 доби;
- weekly check – виконується раз на тиждень;
- a-check – виконується раз на місяць;
- b-check – виконується раз на 3 місяці;
- c-check – потребує виведення літака зі строю (виділяють C1, C2, C4, C6, C8);

- d-check – повна перевірка;
- sv – технічні роботи двигунів.

При визначенні для літака процедур автоматично змінюється його стан.

На рис. 5.7 продемонстровано приклад реєстрації пасажирів на рейс. Виконується менеджером авіалінії.

Менеджер має обрати конкретний рейс з переліку, що заповнюється даними рейсів, які виконуються поточної доби і ще не завершені, потім має внести дані пасажирів (вказати прізвище, ім'я, по батькові або тільки частину цих даних, номер телефону або номер паспорту, після чого натиснути на кнопку “Обрати”). Після того як відбувся пошук пасажирів випадаючий перелік заповнюється даними і можна обрати одного зі знайдених пасажирів. Якщо пасажирів не було знайдено, то можна натиснути на кнопку “Додати дані пасажирів”.

Рисунок 5.7 – Реєстрація пасажирів на рейс

На рис. 5.8 продемонстровано другу частину цієї сторінки. Тут можна задати вартість (можна пропустити), тип місця, а також безпосередньо саме

місце. Місця виводяться тільки ті, на які ще не було зареєстровано пасажира. Для підтвердження реєстрації потрібно натиснути на кнопку “Обрати”.

Пошук рейсів

Аеропорт

Розклад

Табло

Реєстрація на рейс

Обслуговування

Пошук

Робота з даними

Вийти

Вартість

Вартість

Тип

Економ

Місце

1A1B1C

2A2B2C2D2E2F

3A3B3C3D3E3F

4A4B4C4D4E4F

5A5B5C5D5E5F

6A6B6C6D6E6F

8A8B8C8D8E8F

9A9B9C9E9F

10A10B10C10D10E10F

11A11B11C11F

12A12B12C12D12E12F

13A13B13C13D13F

14B14C14D14E

15A15B15C15D15E15F

16A16B16C16D16E16F

17A17B17C17D17E17F

18A18B18C18D18E18F

19A19B19C19D19E19F

20A20B20C20D20E20F

21A21B21C21D21E21F

22A22B22C22D22E22F

ОБРАТИ

Рисунок 5.8 – Вибір місця пасажира під час реєстрації на рейс

На рис. 5.9 продемонстровано приклад виконання прогнозування пасажиропотоку аеропорту для менеджера аеропорту.

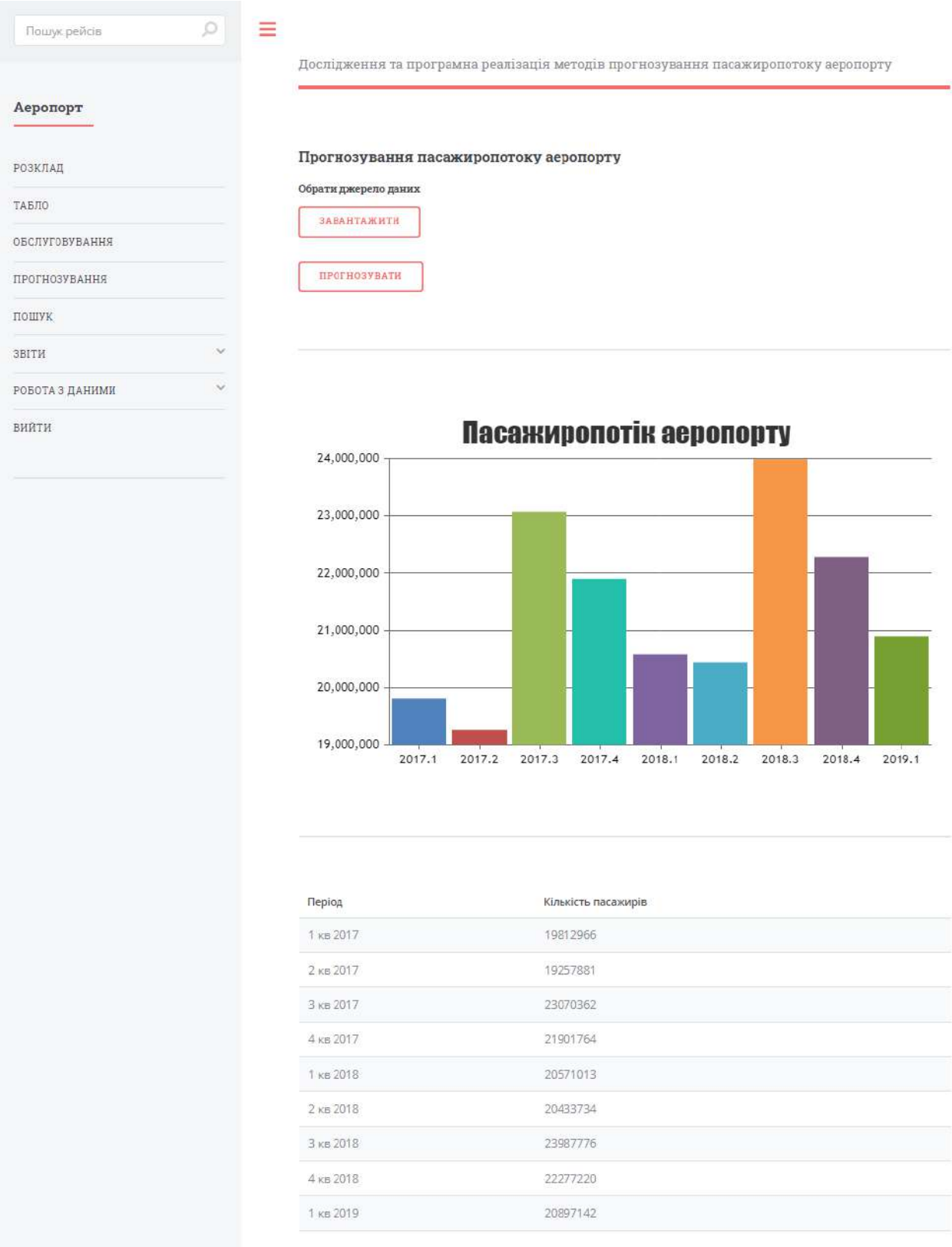
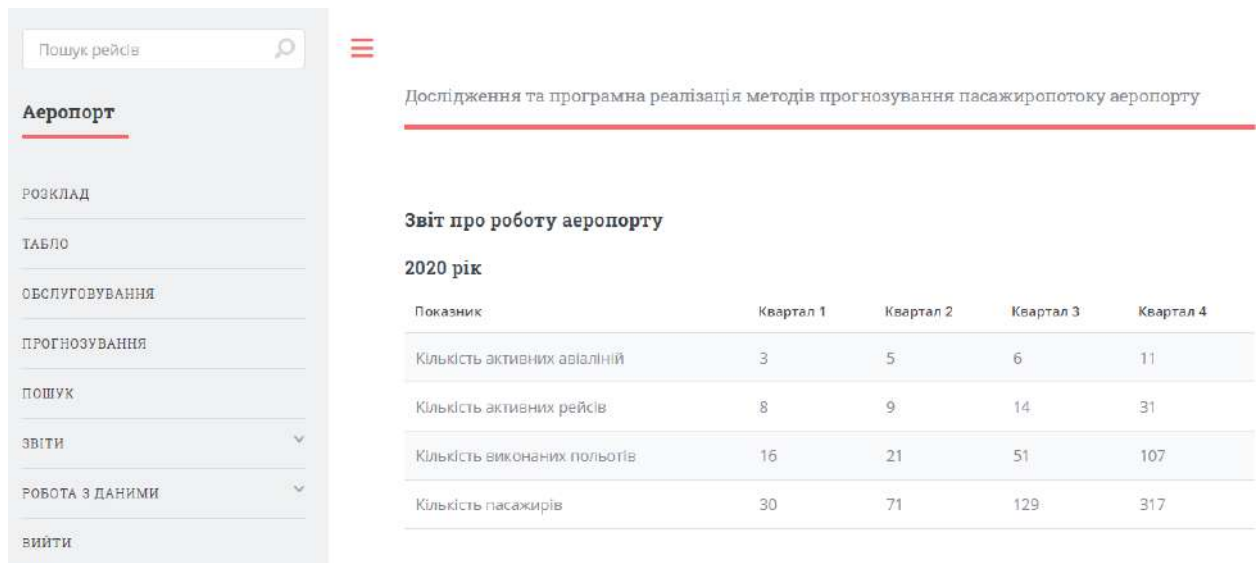


Рисунок 5.9 – Прогнозування пасажиропотоку аеропорту для менеджера аеропорту

На рис. 5.10 продемонстровано приклад формування звіту про роботу аеропорту за поточний рік для менеджера аеропорту.



Показник	Квартал 1	Квартал 2	Квартал 3	Квартал 4
Кількість активних авіаліній	3	5	6	11
Кількість активних рейсів	8	9	14	31
Кількість виконаних польотів	16	21	51	107
Кількість пасажирів	30	71	129	317

Рисунок 5.10 – Звіт про роботу аеропорту

5.3 Висновки за розділом 5

Описано процес роботи з вебзастосунком всіх типів користувачів, описано логіку роботи з приведенням основних користувацьких форм взаємодії.

6 ЕКОНОМІКО-ОРГАНІЗАЦІЙНА ЧАСТИНА

В магістерській роботі проведено дослідження та програмну реалізацію методів прогнозування пасажиропотоку аеропорту.

Розроблене в межах магістерської роботи програмне забезпечення призначене для обліку процесів у аеропорті, а в основі забезпечення керування процесами лежить прогнозування показника пасажиропотоку аеропорту як базового показника всієї діяльності. На підприємстві до цього дану функцію виконував аналітик. Впровадження програмного забезпечення дозволить скоротити 0,5 ставки аналітика, що заробляє 8000 грн./міс., шляхом автоматизації даного процесу.

6.1 Планування розробки програмного продукту

Процес розробки програмного продукту розбивається на декілька етапів та кожному етапу у відповідність зіставляється певна кількість виконавців, трудомісткість та тривалість робіт [16].

В роботі приймають участь:

- програміст, який працює 2,5 місяці з заробітною платнею 9800 грн./місяць;
- консультант, який працює 0,75 місяця з заробітною платнею 8000 грн./місяць.

Тривалість розробки програмного забезпечення та етапів які входять у процес розробки визначається за формулою (6.1):

$$T_{\text{ц}} = \frac{Q}{R}, \quad (6.1)$$

де $T_{\text{ц}}$ – довжина циклу, днів;

Q – трудомісткість, людино-днів;

R – кількість виконавців, людей [16].

У табл. 6.1 наведені роботи, які необхідно виконати у процесі розробки. Для кожного виду роботи визначається найменування, тривалість роботи, трудомісткість та виконавці.

Таблиця 6.1 – Характеристика робіт по розробці програми

Найменування роботи	Трудомісткість		Виконавці		Тривалість
	чол.- дн.	% до підсумку	спеціаль- ність	кіль. чол.	днів
1. Дослідження предметної області	10	14.28	Програміст	2	5
			Консультант		
2. Аналіз вимог до програмного продукту	10	14.28	Програміст	2	5
			Консультант		
3. Складання технічного завдання	10	14.28	Програміст	2	5
			Консультант		
4. Проектування архітектури програмного забезпечення	5	7.14	Програміст	1	5
5. Програмування	20	28.6	Програміст	1	20
6. Тестування та налагодження програмного забезпечення	5	7.14	Програміст	1	5
7. Розробка програмної документації	10	14.28	Програміст	1	10
Всього	70	100,0			55

Планування розробки відображається за допомогою лінійного графіка Ганта. Побудований графік Ганта зображено у табл. 6.2 та табл. 6.3.

Таблиця 6.2 – Лінійний графік Ганта, частина 1

Найменування робіт	Календарний період, дні							
	14–20.09	21–27.09	28.09–4.10	5–11.10	12–18.10	19–25.10	26–1.11	2–8.11
1. Дослідження предметної області								
2. Аналіз вимог до програмного продукту								
3. Складання технічного завдання								
4. Проектування архітектури програми								
5. Програмування								

Таблиця 6.3 – Лінійний графік Ганта, частина 2

Найменування робіт	Календарний період, дні		
	9–15.11	16–22.11	23–29.11
6. Тестування та налагодження програмного забезпечення			
7. Розробка програмної документації			

6.2 Визначення витрат на розробку програми

Для того щоб визначити витрати на розробку програмного забезпечення, виконується калькуляція кошторисної вартості робіт, яка включає наступні статті:

- основна заробітна плата;
- додаткова заробітна плата;
- відрахування на єдиний соціальний внесок;
- витрати на спеціальне обладнання;
- матеріали і комплектуючі вироби;
- накладні витрати;
- податки [16].

6.3 Розрахунок основної заробітної плати

Витрати за цією статтею складаються з планового фонду заробітної плати усіх працівників, зайнятих у розробці програмного продукту, розрахунок заробітної плати ведеться на основі відомостей про трудомісткість робіт (табл. 6.1) [16]. Результати розрахунків відображено в табл. 6.4.

Таблиця 6.4 – Розрахунок додаткової заробітної плати

Посада виконавця	Кіль- кість, осіб	Місяч- ний оклад	Кіль- кість, міс. роботи	Премії та доплати, грн.	Середня місячна зарплата, грн.	Сумарна зарплата, грн.
Програміст	1	9800	2,5	—	9800	24500
Консультант	1	8000	0,75	—	8000	6000
Всього	2					30500

6.4 Розрахунок додаткової заробітної плати

Додаткову заробітну плату, як правило, приймають рівною 10% від основної заробітної плати. Розрахунок додаткової заробітної плати та суми основної та додаткової заробітної плати наведено у табл. 6.5.

Таблиця 6.5 – Розрахунок додаткової заробітної плати та суми основної та додаткової заробітної плати

Посада виконавця	Додаткова заробітна плата, грн.	Сума основної та додаткової заробітної плати, грн.
Програміст	2450	26950
Консультант	600	6600
Всього	3050	33550

6.5 Розрахунок відрахувань до соціального внеску

Дані відрахування розраховуються у відсоткову відношенні від суми основної та додаткової заробітної плати: зазвичай при розрахунку відрахувань до соціального внеску також ураховують премії та доплати, у нашому випадку час розробки досить малий, тому премії та доплати приймаються рівними 0, також ураховують відрахування на соціальне страхування, яке становить 22% [16].

Розрахунок відрахувань до соціального внеску виконується на основі формули (6.2):

$$ВІД = (ЗП_{осн} + ЗП_{доп}) \cdot 0,22, \quad (6.2)$$

де $ЗП_{осн}$ – основна заробітна плата, грн.;

$ЗП_{доп}$ – додаткова заробітна плата, грн. [16].

Результат:

$$\text{ВІД} = 33550 \cdot 0,22 = 7381 \text{ грн.}$$

6.6 Визначення витрат на матеріали

У цю статтю включають вартість основних і допоміжних матеріалів, покупних напівфабрикатів і комплектуючих виробів, необхідних для розробки програми [16].

При розробці даного програмного продукту було використано:

- 2 пачки паперу А4, по 65 грн. кожна;
- 1 тонер фарби для принтера за 200 грн.;
- 4 ручки, по 10 грн. кожна;
- 2 маркери, по 20 грн. кожний;
- 8 CD-RW дисків по 18 грн. кожний.

Витрати на матеріали розраховуються за формулою (6.3):

$$M = \sum_{i=1}^n (C_i \cdot N_i \cdot (1 + K_{\text{т.з.}}) - C_{\text{іо}} \cdot N_{\text{іо}}), \quad (6.3)$$

де M – витрати на матеріали, покупні напівфабрикати і комплектуючі вироби, грн.;

$K_{\text{т.з.}}$ – коефіцієнт, що враховує транспортно-заготівельні витрати;

C_i – ціна i -го найменування матеріалу, напівфабрикату, комплектуючого, грн.;

N_i – кількість i -го матеріалу, напівфабрикату, комплектуючого;

$C_{\text{іо}}$ – ціна поворотних відходів i -го найменування матеріалу, грн.;

$N_{\text{іо}}$ – кількість зворотних відходів i -го найменування;

n – кількість найменувань матеріалів, напівфабрикатів, комплектуючих [16].

Результат:

$$M = 2 \cdot (1 + 0,03) \cdot 65 + (1 + 0,03) \cdot 200 + 4 \cdot (1 + 0,03) \cdot 10 + \\ + 2 \cdot (1 + 0,03) \cdot 20 + 8 \cdot (1 + 0,03) \cdot 18 = 570,62.$$

Витрати на матеріали дорівнюють 570 грн. 62 копійки.

6.7 Витрати на спеціальне устаткування

У даній статті ураховують витрати на оплату машинного часу електронно-обчислювальної машини (ЕОМ) для написання і налагодження програмних засобів, ці витрати розраховуються за формулою (6.4):

$$C_{mo} = P_{екс} \cdot t_{mo}, \quad (6.4)$$

де C_{mo} – витрати на оплату машинного часу, грн.;

$P_{екс}$ – експлуатаційні витрати на одну годину машинного часу цієї цифрової ЕОМ, грн./машино-год.;

t_{mo} – машинний час цифрової ЕОМ для написання і налагодження даного програмного продукту, машино-год. [16].

Для того, щоб розрахувати експлуатаційні витрати на одну годину машинного часу, необхідно скласти кошторис витрат на зміст і експлуатацію цього устаткування [16].

Амортизаційні відрахування визначаються за формулою (6.5):

$$A = \Phi_{\delta} \cdot \frac{H_a}{100}, \quad (6.5)$$

де Φ_{δ} – балансова вартість обчислювальної техніки, грн.;

H_a – норма амортизаційних відрахувань на повне відновлення обчислювальної техніки, % [16].

Балансова вартість обчислювальної техніки визначається як ціна обчислювальної техніки, помножена на коефіцієнт транспортно-заготівельних витрат і монтажу обладнання ($K=0,05$) [16].

У даному випадку вона становить:

$$28000 \cdot (1+0,05) = 29400 \text{ грн.}$$

Оскільки норма амортизації для персонального комп'ютера становить 25%, амортизаційні відрахування дорівнюють:

$$A = 29400 \cdot 0,25 = 7350 \text{ грн.}$$

Експлуатація обладнання розраховується за допомогою підсумовування витрат на силову електроенергію і допоміжні матеріали, при цьому витрати на електроенергію розраховують за формулою (6.6):

$$C_e = N_n \cdot \Phi_{ef} \cdot K_{зч} \cdot K_{зн} \cdot C_{ел.е}, \quad (6.6)$$

де N_n – номінальна потужність ЕОМ, кВт;

Φ_{ef} – річний ефективний фонд часу роботи ЕОМ, машино–год;

$K_{зч}$ – середній коефіцієнт завантаження по часу;

$K_{зн}$ – коефіцієнт завантаження по потужності;

$C_{ел.е}$ – ціна однієї кВт·години електроенергії, грн./(кВт·год) [16].

Відповідно:

$$C_e = 0,4 \cdot 1800 \cdot 0,9 \cdot 0,7 \cdot 2,11 = 867,76 \text{ грн.}$$

Витрати за статтею «Заробітна плата обслуговуючих робітників та відрахування на соціальне страхування в інші фонди» визначають за

формулою (6.7):

$$ЗП_{обсл} = \Phi ЗП_p \cdot (1 + K_{відр}) \cdot \frac{t_{обсл}}{\Phi_{еф.обсл}}, \quad (6.7)$$

де $\Phi ЗП_p$ – річний фонд заробітної плати (основної та додаткової) обслуговуючих робітників, грн;

$K_{відр}$ – коефіцієнт, що враховує відрахування на соціальне страхування та в інші фонди;

$t_{обсл}$ – час протягом року, необхідний на технічне обслуговування ЕОМ, год./рік;

$\Phi_{еф.обсл}$ – річний ефективний фонд часу обслуговуючого персоналу, год/рік [16].

Робочий комп'ютер обслуговується одним системним адміністратором із заробітною платою 4500 грн./міс., з чого виходить, що $\Phi ЗП_p = 54000$ грн. Системний адміністратор витрачає 2 години у місяць, отже $t_{обсл} = 24$ години у рік, а $\Phi_{еф.обсл} = 1750$ годин.

Відповідно можна обчислити:

$$ЗП_{обсл} = 54000 \cdot (1 + 0,22) \cdot \frac{24}{1750} = 903,50 \text{ грн.}$$

Поточний ремонт устаткування дорівнює 3% від балансової вартості устаткування, а саме:

$$7350 \cdot 0,03 = 220,5 \text{ грн.}$$

Інші витрати дорівнюють 3% від суми усіх попередніх статей витрат на утримання і експлуатацію обладнання:

$$(7350 + 867,76 + 903,50 + 220,5) \cdot 0,04 = 373,67 \text{ грн.}$$

У табл. 6.6 зведено усі дані стосовно кошторису витрат на утримання і експлуатацію устаткування.

Таблиця 6.6 – Кошторис витрат на утримання і експлуатацію устаткування

Найменування статей витрат	Сума, грн.
Амортизація устаткування	7350
Експлуатація устаткування	867,76
Зарплата обслуговуючого персоналу з відрахуваннями	903,50
Поточний ремонт устаткування	220,5
Інші витрати	373,67
Всього	9715.43

Експлуатаційні витрати на одну годину машинного часу складають:

$$P_{\text{екс}} = \frac{9715.43}{1800} = 5,4 \text{ грн./год.}$$

Розробка і відладка програмного продукту зайняла 55 днів по 8 робочих годин, отже $t_{\text{мо}} = 440$ год. Витрати на оплату машинного часу складають:

$$C_{\text{мо}} = 5,4 \cdot 440 = 2376 \text{ грн.}$$

6.8 Накладні витрати

До накладних витрат відносять витрати на загальне управління і загальногосподарські потреби (наприклад: заробітна плата апарату

управління, канцелярські витрати, утримання та експлуатацію будівель і т.д.), накладні витрати включають у вартість розробки програми непрямым шляхом – у відсотках (від 10 до 100 %) до основної заробітної плати розробників, у даному проєкті накладні витрати були прийняті за 40% від основної заробітної плати розробників, тобто 12200 грн. [16].

Результати підрахунків кошторисної вартості по розробці програмного забезпечення занесені у табл. 6.7.

Таблиця 6.7 – Підрахунки кошторисної вартості робіт по розробці програмного забезпечення

№	Найменування статей витрат	Сума, грн.	Питома вага до підсумку, %
1	Основна заробітна плата	30500	54.39
2	Додаткова заробітна плата	3050	5.44
3	Єдиний соціальний внесок	7381	13.16
4	Витрати на матеріали	570,62	1.02
5	Витрати на оплату машинного часу	2376	4.24
6	Накладні витрати	12200	21.75
7	Всього	56077,62	100

6.9 Оцінка економічної ефективності

Витрати на розробку програмного забезпечення склали 56077,62 гривень. Розроблене в межах магістерської роботи програмне забезпечення призначене для обліку процесів у аеропорті, а в основі забезпечення керування процесами лежить прогнозування показника пасажиропотоку аеропорту як базового показника всієї діяльності. На підприємстві до цього дану функцію виконував аналітик. Впровадження програмного забезпечення дозволить скоротити 0,5 ставки аналітика (8000 грн./міс.) шляхом автоматизації даного процесу.

Річна економія на зарплаті аналітика з урахуванням відрахувань на єдиний соціальний внесок становить 58560 гривень. Таким чином, витрати окупаються за один рік (нормативний термін окупності для соціальної сфери 6,67 року) і можна зробити висновок, що дана робота економічно доцільна.

7 ОХОРОНА ПРАЦІ ТА БЕЗПЕКА В НАДЗВИЧАЙНИХ СИТУАЦІЯХ

7.1 Аналіз потенційних небезпек

У ході виконання магістерської роботи досліджуються та програмно реалізуються методи прогнозування пасажиропотоку аеропорту. У процесі виконання даної роботи можуть реалізовуватися наступні потенційні небезпеки:

- незадовільні ергономічні характеристики робочого місця внаслідок нераціонального планування робочого місця, що може призвести до механічних травм, уражень електричним струмом та порушень кістково-м'язового апарату;

- неправильні дії персоналу під час надзвичайної ситуації внаслідок відсутності управління та паніки, що може призвести до травм та летальних наслідків [17].

7.2 Заходи по забезпеченню безпеки

Згідно до ДСТУ ISO 9241-5:2004 «Ергономічні вимоги до роботи з відеотерміналами в офісі. Вимоги до компонування робочого місця та до робочої пози» головними елементами робочого місця спеціаліста є стіл і крісло, основним робочим положенням є положення сидючи, раціональне планування робочого місця передбачає чіткий порядок і сталість розміщення предметів, засобів праці і документації, те, що потрібно для виконання робіт частіше, розташовано в зоні легкої досяжності робочого простору [17].

Згідно Правил улаштування електроустановок (ПУЕ) приміщення, в якому проводиться дослідження методів та розробка програмних засобів синтаксичного та інтелектуального аналізу веб-сторінок відноситься до «Приміщення без підвищеною небезпеки», у приміщенні повинні бути відсутні відкриті струмоведучі частини з напругою більше 12 В, при

виконанні правил ПУЕ враження електричним струмом можливо тільки у випадку несправності апаратури й живильних кабелів [18].

Електрична ізоляція забезпечується конструкціями пристроїв і способами їхнього підключення, у процесі експлуатації необхідно періодично робити перевірку якості ізоляції, шляхом виміру її опору, відповідно до ПУЕ в електроустановках до 1000 В мінімальне значення опору ізоляції силових і освітлювальних електропроводок на ділянці між суміжними запобіжниками або за останніми запобіжниками між будь-яким проведенням і землею становить не менш 0,5 МОм [19].

Подвійна ізоляція забезпечує застосування крім основної, робочої ізоляції струмоведучих частин ще одного шару додаткової ізоляції, що ізолює людину від металевих неструмоведучих частин, які можуть випадково виявитися під напругою, при застосуванні електроустаткування з подвійною ізоляцією не потрібно ні заземлення, ні занулення їхніх корпусів [20].

Засоби захисту від статичної електрики:

- вживання захисних екранів із заземленням;
- своєчасне провітрювання [20].

Згідно з Нормативним актом про охорону праці 0.00-1.28-10 «Правила охорони праці під час експлуатації електронно-обчислювальних машин» необхідно комплексно вирішувати проблеми ергономіки, гігієни і організації праці, регламентації режимів праці та відпочинку:

- по-перше, для збереження здоров'я працівників, необхідно дотримуватися всіх вимог до приміщень, де відбувається робота з комп'ютерами;

- по-друге, необхідно подбати про правильне розміщення робочих місць як щодо безпосередньо кімнати, так і щодо всієї будівлі підприємства;

- по-третє, слід забезпечити робітників сучасною комп'ютерною меблюю, яка була б зручна у використанні і не доставляла почуття дискомфорту, була пристосована до особливостей людського організму [21].

7.3 Заходи по забезпеченню виробничої санітарії та гігієни праці

Для забезпечення оптимальної освітленості в робочому приміщенні яке нормується згідно з Державними будівельними нормами В.2.5.-28-2006 «Природне та штучне освітлення» 200 Лк передбачено устрій системи штучного освітлення з використанням люмінесцентних ламп типу ЛБ або ЛД потужністю від 40 до 80 Вт, що встановлені у світильниках типу ЛСП [22].

Розрахуємо систему загального рівномірного штучного освітлення в приміщенні масажного кабінету.

Приймаємо розміри приміщення довжина $A = 6$ м, ширина $B = 4$ м, висота $H = 4$ м.

Висота робочої поверхні складає $h = 0,7$ м.

Згідно з Державними будівельними нормами В.2.5-28-2006 «Природне і штучне освітлення. Норми проектування» освітлення в кабінеті масажу нормуються як $E_n = 200$ Лк.

Коефіцієнти відбиття поверхонь приміщення (від: стелі – ρ_c ; стін – $\rho_{ст}$; підлоги – $\rho_{п}$) у світлому приміщенні $\rho_c=70\%$; $\rho_{ст}=50\%$; $\rho_{п}=30\%$;

Обираємо світильник типу ЛВП (IP=54 – ступінь захисту світильника, $[L/h] = 1,3$ – числове значення коефіцієнта світильника).

Коефіцієнт запасу k_z враховує зниження рівня освітленості з часом в результаті забруднення та старіння ламп, світильників і поверхонь приміщення, приймається в залежності від виробничих умов. Приймаємо $k_z = 1,4$.

Коефіцієнт нерівномірності (мінімального) освітлення z (відношення середньої освітленості до мінімальної освітленості), для люмінесцентних ламп низького тиску, як правило дорівнює $z = 1,1$.

Розрахуємо кількість рядів світильників у приміщенні за формулою (7.1):

$$N_p = \frac{B}{(H - h_p) \cdot (L / h)} = 0,93 \approx 1. \quad (7.1)$$

Визначимо максимально припустиму відстань між рядами світильників за формулою (7.2):

$$L_{\max} = \frac{B}{N_p}, \quad (7.2)$$

де B – ширина приміщення;

N_p – кількість рядів [22].

Розрахуємо:

$$L_{\max} = \frac{4}{1} = 4 \text{ м.}$$

Розраховуємо висоту підвісу світильника над робочою поверхнею за формулою (7.3)

$$h = \frac{L_{\max}}{[L / h]}, \quad (7.3)$$

$$h = \frac{4}{1,3} = 3,07 \text{ м.}$$

Знайдемо висоту звисання світильника від стелі за формулою (7.4):

$$h_3 = H - h_p - h, \quad (7.4)$$

де H – висота приміщення;

h_p – висота робочої поверхні;

h – висота підвісу світильника над робочою поверхнею [22].

Розрахуємо:

$$h_3 = 4 - 0,7 - 3,07 = 0,23 \text{ м.}$$

Чисельне значення індексу приміщення визначаємо за рівнянням (7.5):

$$i = \frac{AB}{h(A+B)}, \quad (7.5)$$

де A – довжина приміщення, м;

B – ширина приміщення, м;

h – висота розміщення світильників над робочою поверхнею, м [22].

Розрахуємо:

$$i = \frac{6 \cdot 4}{3,07 \cdot (6 + 4)} = \frac{24}{30,7} = 0,78.$$

Визначаємо сумарний світловий потік освітлювальної установки у даному виробничому приміщенні за формулою (7.6):

$$\Phi_{\Sigma} = \frac{E_n \cdot S \cdot k_3 \cdot z}{\eta}, \quad (7.6)$$

де Φ_{Σ} – розрахункове значення сумарного світлового потоку у приміщенні, лм;

E_n – нормоване значення освітленості, лк;

S – площа освітлюваної поверхні, м²;

k_3 – коефіцієнт запасу;

z – коефіцієнт нерівномірності (мінімальної) освітленості;

η – коефіцієнт використання світлового потоку [22].

Розрахуємо:

$$\Phi_{\Sigma} = \frac{200 \cdot 6 \cdot 4 \cdot 1,4 \cdot 1,1}{0,38} = \frac{7392}{0,38} = 19452 \text{ лм.}$$

Визначаємо умовну загальну кількість світильників у приміщенні за формулою (7.7):

$$N^* = AB / L_{\max}^2, \quad (7.7)$$

$$N^* = \frac{6 \cdot 4}{16} = 1,5 \approx 2 \text{ шт.}$$

Розрахуємо світловий потік умовного джерела світла за формулою (7.8):

$$\Phi_{\text{л}}^* = \frac{\Phi_{\Sigma}}{N_{\text{л}}}, \text{ лм,} \quad (7.8)$$

де Φ_{Σ} – розрахункове значення сумарного світлового потоку у приміщенні, лм;

$N_{\text{л}}$ – загальна кількість ламп у приміщенні, шт. [22].

Розраховується загальна кількість ламп у приміщенні за формулою (7.9):

$$N_{\text{л}} = N^* \cdot n, \quad (7.9)$$

де n – кількість ламп у світильнику (в ЛВП-світильнику знаходиться 2 лампи).

При цьому

$$N_{\text{л}} = 2 \cdot 2 = 4 \text{ шт.}$$

Тоді

$$\Phi_{\text{л}} = \frac{19452}{4} = 4863 \text{ лм.}$$

Обираємо лампу ЛБ 65, $\Phi_{\text{л}} = 4800$ лм, за формулою (7.10):

$$m = \frac{\Phi_{\text{л}}^*}{\Phi_{\text{л}}}, \quad (7.10)$$

$$m = \frac{4863}{4800} = 1,01.$$

Визначаємо оптимальну (фактичну) кількість світильників у приміщенні за формулою (7.11):

$$N = N \cdot m, \quad (7.11)$$

$$N = 2 \cdot 1,01 = 2 \text{ шт.}$$

Фактична кількість ламп у приміщенні за формулою (7.12):

$$N_{\phi\text{л}} = N_{\phi} \cdot n, \quad (7.12)$$

$$N_{\phi\text{л}} = 2 \cdot 2 \text{ шт.}$$

Визначаємо загальну розрахункову освітленість E_p у приміщенні, що створюється при застосуванні стандартних ламп, за формулою (7.13):

$$E_p = \frac{\Phi_{\text{л}} N_{\phi\text{л}} \eta}{S k_3 z} \quad (7.13)$$

$$E_p = \frac{4800 \cdot 4 \cdot 0,38}{6 \cdot 4 \cdot 1,4 \cdot 1,1} = \frac{7296}{36,96} = 197,4 \text{ лк.}$$

Тобто виконується умова $E_p = (-10\% \dots +20\%) \cdot E_{\text{н}}$.

Розрахуємо загальну потужність освітлювальної установки за формулою (7.14):

$$P_{\Sigma} = N_{\phi\lambda} \cdot P_{\lambda}, \text{ Вт}, \quad (7.14)$$

де P_{λ} – потужність вибраної стандартної лампи.

$$P_{\Sigma} = 4 \cdot 65 = 260 \text{ Вт}.$$

Вентиляцію зі штучним спонуканням слід передбачати для приміщень і зон без природного провітрювання. Допускається проектувати змішану вентиляцію з частковим використанням природного спонукання для припливу або видалення повітря [23].

7.4 Заходи по забезпеченню пожежної безпеки

В приміщенні обчислювального центру джерелом загоряння можуть бути електронні схеми ЕОМ, прилади для технічного обслуговування, електроустаткування, кондиціонери повітря, де за рахунок різних порушень відбувається перегрів тих чи інших елементів, електричні іскри, дуга, що можуть визвати загоряння горючих матеріалів, дерев'яні та пластикові стільці та столи, папір, перевантаження мережі живлення, людські чинники тощо [24].

Клас пожежі для приміщень обчислювального центру встановлюється згідно Державних будівельних норм В.1.1.7-2002 «Захист від пожежі. Пожежна безпека об'єктів будівництва» – пожежі твердих речовин, переважно органічного походження, горіння яких супроводжується тлінням (деревина, пластмаси, папір) – і визначається як клас А [24].

Категорія приміщення обчислювального центру за вибухопожежною небезпекою визначається згідно Нормативно-правового акту з пожежної безпеки Б.03.002-2007 «Норми визначення категорії приміщень, будинків та зовнішніх установок за вибухопожежною та пожежною небезпекою» – і визначається як пожежонебезпечність адмінприміщень: коридори та

проходи, що призначені для евакуації, мають як можна меншу довжину та мінімальну кількість поворотів, на усій протяжності проходу відсутні пороги або проміжні ступені, відповідно до Санітарних норм і правил 2.09.02-85 «Виробничі будівлі» [25].

Важливу роль у забезпеченні безпечного виходу людей грає протидимовий захист евакуаційних ходів, у будинках висотою до 9 поверхів незадимлюваність сходових кліток на час евакуації досягається їхньою ізоляцією від підвалів, горищ і поверхів, для цього влаштовані відособлені входи в підвали, вхід на сходову клітку з поверхів здійснюють через тамбур - шлюз із підпором повітря, відокремлюють горища від сходових кліток перекриттями з негорючих матеріалів [26].

У табл. 7.1 наведені відстані від найбільш віддаленого робочого місця до найближчого евакуаційного виходу.

Таблиця 7.1 – Відстані від найбільш віддаленого робочого місця до найближчого евакуаційного виходу

Об'єм приміщення, тис. м ³	Ступінь вогнестійкості будівлі	Відстань, м, при щільності людського потоку в загальному проході, люд/м ²		
		до 1	від 1 до 3	від 3 до 5
Незалежно від об'єму	I, II, III	Не обмежується		
	IV	160	95	65
	V	120	70	50

7.5 Заходи по забезпеченню безпеки у надзвичайних ситуаціях

Організувати і провести рятувальні та інші невідкладні роботи в осередку комбінованого ураження набагато складніше, ніж в осередках радіаційного (ядерного), хімічного або бактеріологічного (біологічного) ураження, це пояснюється складністю обстановки, яка може виникнути внаслідок застосування противником ядерної, хімічної і бактеріологічної

зброї або при одночасному виникненні аварії на атомній електростанції, хімічно небезпечному об'єкті, з метою досягнення максимальних результатів рятувальних та інших невідкладних робіт в осередку комбінованого ураження організовують та безперервно ведуть усі види розвідки: встановлюється жорсткий контроль за виконанням формуваннями робіт із знезараження маршрутів евакуації, території, майна, техніки; проведенням протиепідемічних, спеціальних профілактичних та санітарно-гігієнічних заходів; дотримуванням заходів безпеки, а також за своєчасною зміною формувань, зміна формувань проводиться при суворому дотриманні режимних заходів, замінені формування виводяться в райони, призначені старшим начальником, де проводять їх спеціальну обробку (в межах зони карантину або обсервації) [27]-[28].

ВИСНОВКИ

У дипломній кваліфікаційній роботі магістра розглянуто проблему підтримки авіаперевезень у світі, її стан, тенденції галузі в світі загалом та в Україні зокрема, виконано огляд програмних засобів, які можуть використовуватися для обліку роботи аеропорту.

Наукова новизна роботи полягає в запропонованому методі прогнозування пасажиропотоку аеропорту, який ґрунтується на використанні методу опорних векторів з квадратичною функцією втрат, використовує в якості входних ознак пасажиропотік за минулі періоди, коливання курсу валют за цей період та номер базового періоду.

У роботі визначено функціональні вимоги до програми, сформульовано діаграму прецедентів для роботи менеджера аеропорту, менеджера з технічного обслуговування, менеджера авіалінії та неавторизованого користувача з програмою. Було виконано порівняння засобів реалізації програми (мов PHP і Python), в результаті чого мову програмування PHP та систему керування базами даних MySQL було обрано для реалізації. Було запроєктовано базу даних, результат проєктування сформульовано у вигляді схеми бази даних.

Реалізовано вебзастосунок для прогнозування пасажиропотоку аеропорту. Вебзастосунок дозволяє виконувати облік діяльності аеропорту, включаючи роботу з авіалініями, виконанням рейсів, технічним обслуговуванням, а потім автоматизовано виконувати прогнозування пасажиропотоку на основі автоматично зібраних даних щодо реєстрації пасажирів за минулі періоди. Описано структуру розробленого вебзастосунку з приведенням основних реалізованих скриптів. Створено діаграму розгортання. Описано процес прогнозування пасажиропотоку аеропорту зі створенням схеми функціонування даного процесу.

Усі завдання дипломної кваліфікаційної роботи магістра виконано повністю.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Балашов, В.В. Методика прогнозирования внутрирегиональных и межрегиональных пассажиропотоков на магистральных авиалиниях России [Текст] / В.В. Балашов, А.В. Смирнов // Научный вестник Московского государственного технического университета гражданской авиации. – 2006. – № 100. – С. 165-171.
2. Губенко, А.В. Современное состояние и перспективы развития рынка пассажирских авиаперевозок в России [Текст] / А.В. Губенко, Ю.И. Растова, А.Р. Панкратова // Научный журнал НИУ ИТМО. Серия «Экономика и экологический менеджмент». – 2019. – № 2. – С. 82-90.
3. В 2019 году рост пассажирооборота авиакомпаний мира замедлился до 4,2% [Электрон. ресурс]. – Режим доступа : https://cfts.org.ua/news/2020/02/11/v_2019_godu_rost_passazhirooborota_aviakompaniy_mira_zamedlilsya_do_42_57287
4. Авіаперевезення в Україні: з яких аеропортів найбільше літають [Електрон. ресурс]. – Режим доступа : <https://www.unian.ua/tourism/news/10650834-aviaperevezennya-v-ukrajini-z-yakih-aeroportiv-naybilshe-litayut.html>
5. Информационная система «Аэропорт» [Электрон. ресурс]. – Режим доступа : <http://www.aviabit.ru/airport>
6. КИС Аэропорт [Электрон. ресурс]. – Режим доступа : <https://www.airsoft-bit.ru/airport/196-kis-aeroport>
7. Методы прогнозирования [Электрон. ресурс]. – Режим доступа : <http://masters.donntu.org/2006/kita/kiryan/library/art10.htm>
8. Шеремет, О.І. Метод опорних векторів (SVM) [Текст] / О.І. Шеремет, О.В. Садовой // Математичне моделювання. – 2013. – № 1 (28). – С. 13-17.
9. Метод опорных векторов – Supported Vector Machine (SVM) [Электрон. ресурс]. – Режим доступа : <http://statistica.ru/branches-maths/metod->

opornykh-vektorov-supported-vector-machine-svm/

10. Бобоев, Ш.А. Применение метода LS-SVM для анализа выборок LIDAR и MOTORCYCLE [Текст] / Ш.А. Бобоев // Сборник научных трудов НГТУ. – 2019. – № 1 (94). – С. 85-99.
11. Зандстра, М. PHP. Объекты, шаблоны и методики программирования [Текст] / М. Зандстра. – 5-е изд.. – СПб. : Диалектика, 2019. – 736 с.
12. Кузнецов, М. Самоучитель PHP 7 [Текст] / М. Кузнецов, И. Симдянов. – 2-е изд. – СПб. : Диалектика, 2018. – 448 с.
13. Advantages of PHP [Electronic resource]. – Access mode : <https://www.educba.com/advantages-of-php/>
14. Editorial by Pixelarity [Electronic resource]. – Access mode : <https://pixelarity.com/editorial>
15. PHP : SVM – Manual [Electronic resource]. – Access mode : <https://www.php.net/manual/ru/book.svm.php>
16. Остапенко, В.В. Методичні вказівки по економічному обґрунтуванню дипломних проєктів для студентів спеціальностей 7.080403 «Програмне забезпечення автоматизованих систем», 7.080402 «Інформаційні технології проєктування» і 7.091501 «Комп'ютерні та інтелектуальні системи і мережі» всіх форм навчання [Текст] / В.В. Остапенко, Е.М. Касьян, Є.М. Сердюк. – Запоріжжя : ЗНТУ, 2009. – 22 с.
17. ДСТУ ISO 9241-5:2004. Ергономічні вимоги до роботи з відеотерміналами в офісі. Частина 5. Вимоги до компонування робочого місця та до робочої пози (ISO 9241-5:1998, IDT) : Національний стандарт України [Текст]. – Видання офіційне. – [Уведено вперше ; чинний від 2006-01-01]. – Київ : Держстандарт України, 2006. – 27 с.
18. Правила улаштування електроустановок [Текст]. – К. : ОЕП "ГРІФЕ", 2006. – 77 с.
19. Жидецький, В. Ц. Основи охорони праці : підручник [Текст] / В. Ц. Жидецький. – 5-те вид., доп. – К. : Знання, 2014. – 373 с

20. Основи охорони праці : підручник [Текст] / В. В. Березуцький, Т. С. Бондаренко, Г. Г. Валенко та ін. ; за ред. проф. В. В. Березуцького. – Х. : Факт, 2005. – 327 с.

21. Вимоги щодо безпеки та захисту здоров'я працівників під час роботи з екранними пристроям : НПАОП 0.00-7.15-18 [Електрон. ресурс]. – На заміну НПАОП 0.00-1.28-10. – [Чинний від 2018-05-18]. – К. : Мінсоцполітики України, 2018. – Режим доступу: <http://zakon.rada.gov.ua/laws/show/z0508-18>. – (Нормативно-правовий акт охорони праці)

22. ДБН В.2.5-28:2018. Природне та штучне освітлення. Посібник : проблеми природного і штучного освітлення [Текст]. – К. : Етіл, 2019. – 180 с.

23. СНиП 2.04.05-91*. Отопление, вентиляция и кондиционирование [Текст] / Госстрой России. – М. : ГУП ЦПП, 1997. – 72 с.

24. ДБН В.1.1-7:2016 Пожежна безпека об'єктів будівництва. Загальні вимоги [Текст]. – Київ : Міністерство регіонального розвитку, будівництва та житлово-комунального господарства України, 2017. – 39 с.

25. Визначення категорій приміщень, будинків та зовнішніх установок за вибухопожежною та пожежною небезпекою : ДСТУ Б В.1.1-36:2016 [Текст]. – На заміну НАПБ Б.03.002-2007. – [Чинний від 2017-01-01]. – К. : Мінрегіонбуд України, 2016. – 66 с. – (Національний стандарт України)

26. Катренко, Л. А. Охорона праці. Курс лекцій. Практикум : навчальний посібник [Текст] / Л. А. Катренко, Ю. В. Кіт, І. П. Піскун. – 2-ге вид., стер. – Суми : ВТД «Університетська книга», 2007. – 303 с.

27. Стеблюк, М. І. Цивільна оборона та цивільний захист : навч. посіб. для вузів [Текст] / М. І. Стеблюк. – К. : Знання, 2013. – 487 с.

28. Шоботов, В. М. Цивільна оборона : навчальний посібник [Текст] / В. М. Шоботов. – 2-ге вид., перероб. – К. : Центр навчальної літератури, 2006. – 438 с.

ДОДАТОК А
Технічне завдання

Вступ

У роботі необхідно виконати розроблення програмного забезпечення прогнозування пасажиропотоку аеропорту для підвищення автоматизованості даного процесу та обґрунтованості рішень, що приймаються відносно планування діяльності аеропорту.

A.1 Підстави для розробки

Розробка виконується у відповідності з завданням, яке було визначено на дипломну кваліфікаційну роботу магістра за темою “Дослідження та програмна реалізація методів прогнозування пасажиропотоку аеропорту”, що затверджена наказом № 301 від 3 листопада 2020 р. по Національному університету «Запорізька політехніка».

A.2 Призначення розробки

Програма, що розробляється, призначена для виконання обліку діяльності аеропорту, включаючи роботу з авіалініями, виконанням рейсів, технічним обслуговуванням, що має в результаті дозволити автоматизовано виконувати прогнозування пасажиропотоку на основі автоматично зібраних даних щодо реєстрації пасажирів за минулі періоди.

A.3 Основні вимоги до програми

A.3.1 Вимоги до функціональних характеристик

Програма повинна реалізовувати наступні функціональні характеристики:

- додавання даних про нові літаки, які обслуговуються в аеропорту;
- змінювання даних літаків, які вже обслуговуються в аеропорту;

- внесення даних про технічне обслуговування літака;
- фіксування результатів технічного обслуговування літака;
- додавання даних авіаліній, які працюють з аеропортом;
- змінювання даних авіаліній;
- відображення розкладу рейсів;
- відображення табло аеропорту;
- внесення даних про нові рейси, які виконуються з аеропорту;
- фіксування фактично виконаних польотів;
- реєстрація пасажирів;
- відображення зареєстрованих на політ;
- внесення даних багажу пасажирів;
- прогнозування кількості пасажирів;
- внесення даних членів екіпажу;
- внесення даних про польоти екіпажу;
- пошук пасажирів, літаків, рейсів;
- перегляд, редагування і вилучення записів про пасажирів;
- перегляд, редагування і вилучення записів про реєстрацію пасажирів;
- перегляд, редагування і вилучення записів про зареєстрований багаж;
- перегляд, редагування і вилучення записів про пасажирів;
- перегляд, редагування і вилучення записів про рейси;
- перегляд і вилучення записів про літаки;
- перегляд, редагування і вилучення записів про фактичні польоти літаків;
- перегляд, редагування і вилучення записів про технічне обслуговування;
- перегляд, редагування і вилучення записів про персонал;
- перегляд, редагування і вилучення записів про членів екіпажу;
- перегляд і вилучення записів про авіалінії;

- формування звіту про пасажиропотік аеропорту;
- формування звіту про роботу аеропорту;
- авторизація;
- внесення облікових записів персоналу.

A.3.2 Вимоги до надійності

Програмне забезпечення має коректно обробляти введені користувачем дані, зокрема має передбачатися перевірка того, що поля таблиць бази даних, які повинні бути обов'язково вказані, не залишені порожніми. Для кожного типу користувачів має визначатися тільки та функціональність, яка відповідає його типу роботи.

A.3.3 Умови експлуатації

Для експлуатації програмного забезпечення на сервері має бути встановлено і запущено:

- вебсервер Apache;
- систему керування базами даних MySQL;
- інтерпретатор мови PHP.

На клієнтській стороні достатньо встановленого веббраузера та наявності мережного підключення до сервера.

A.3.4 Вимоги до складу та параметрів технічних засобів

Сервер має відповідати наступним мінімальним вимогам:

- тактова частота процесора – не менше 2,5 ГГц;
- оперативна пам'ять – не менше 8 Гб;
- вільний дисковий простір – не менше 500 Гб;
- пропускна здатність комп'ютерної мережі – не менше 100 Мбіт/с.

A.3.5 Вимоги до транспортування та збереження

При збереженні програми потрібно враховувати вимоги, що висуваються до носія, на якому вона зберігається, при цьому обов'язково потрібно забезпечити резервне копіювання. При доступі до програми з сервера потрібно забезпечити виконання всіх потрібних профілактичних робіт та вимог до роботи сервера.

A.3.6 Вимоги до маркування та упакування

У програмному застосунку має бути на кожній вебсторінці обов'язково вказано назву “Дослідження та програмна реалізація методів прогнозування пасажиропотоку аеропорту”.

ДОДАТОК Б
Текст програми

Б.1 Текст файлу index.php

```

<!DOCTYPE HTML>
<html>
    <head>
        <title>Дослідження та програмна реалізація методів
прогнозування пасажиропотоку аеропорту</title>
        <meta charset="utf-8" />
        <meta name="viewport" content="width=device-width,
initial-scale=1, user-scalable=no" />
        <link rel="stylesheet" href="assets/css/main.css" />
    </head>

    <body class="is-preload">
        <div id="wrapper">
            <div id="main">
                <div class="inner">
                    <header id="header">
                        <a
href="index.php" class="logo">Дослідження та програмна реалізація
методів прогнозування пасажиропотоку аеропорту</a>
                    </header>
                    <section id="banner">
                        <div
class="content">

                            <?php include "db.php" ?>

                            <h3>Табло аеропорту</h3>

                            <?php

                                if (isset($_POST['modebtn'])) {

                                    $mode = !$mode;

                                } else {

                                    $mode = False;

                                }

                            ?>

                            <?php

                                if (!$mode) {

                                    echo "<h4>Прибуття</h4>";

                                } else {

                                    echo "<h4>Відправлення</h4>";

```

```

}

?>

<div class="table-wrapper">

    <table>

        <thead>

            <tr>

                <th>Авіалінія</th>

                <th>Маршрут</th>

                <th>Виліт</th>

                <th>Прибуття</th>

                <th>Стан</th>

            </tr>

        </thead>

        <tbody>

            <?php

                if (!$mode) {

                    $cd = date("m.d.y");

                    $dd = date("w");

                    $alldata = mysqli_query($db, "SELECT Airline.name,
Route.places, Route.departtime, Route.flighttime, Route.id FROM
`Route` INNER JOIN `Airline` ON Route.line = Airline.id WHERE
date('m.d.y', Route.enddate) >= $cd AND Route.places NOT LIKE
'Запоріжжя%' AND $dd = var_dump((bool)($Route.timetable & (1 <<
($dd - 1))))");

                    while($oned = mysqli_fetch_array($alldata)){ ?>

                        <tr>

                            <td><?php echo $oned[0]; ?></td>

                            <td><?php echo $oned[1]; ?></td>

                            <td><?php echo $oned[2]; ?></td>

                            <td><?php echo $oned[3]; ?></td>

```

```

        <td><?php $res = mysqli_query($db, "SELECT
departtime, arrivtime FROM `Flight` WHERE Flight.race = $oned[4]
AND CAST(Flight.departtime AS DATE) = CAST(GETDATE() AS DATE)");

        if (mysqli_num_rows ($res) == 0) {

            echo "<td>Запланований</td>";

        }

        else {

            $p = mysqli_fetch_array($res);

            if (!$p[`arrivtime`]) {

                echo      "<td>Очікується,          вилетів          o
". $p[`departtime`]. "</td>";

            }

            else {

                echo      "<td>Прилетів          o
". $p[`arrivtime`]. "</td>";

            }

        }

    }

}

?>

<?php else {

    $cd = date("m.d.y");

    $dd = date("w");

    $alldata = mysqli_query($db, "SELECT Airline.name,
Route.places, Route.departtime, Route.flighttime, Route.id FROM
`Route` INNER JOIN `Airline` ON Route.line = Airline.id WHERE
date('m.d.y', Route.enddate) >= $cd AND Route.places LIKE
'Запоріжжя%' AND $dd = var_dump((bool)($Route.timetable & (1 <<
($dd - 1))))");

    while($oned = mysqli_fetch_array($alldata)){ ?>

        <tr>

            <td><?php echo $oned[0]; ?></td>

```

```

        <td><?php echo $oned[1]; ?></td>

        <td><?php echo $oned[2]; ?></td>

        <td><?php echo $oned[3]; ?></td>

        <td><?php $res = mysqli_query($db, "SELECT
departtime, arrivtime FROM `Flight` WHERE Flight.race = $oned[4]
AND CAST(Flight.departtime AS DATE) = CAST(GETDATE() AS DATE)");

        if (mysqli_num_rows ($res) == 0) {

            echo "<td>Запланований</td>";

        }

        else {

            $p = mysqli_fetch_array($res);

            if (!$p[`arrivtime`]) {

                echo      "<td>Очікується,          вилетів          o
".$p[`departtime`]."</td>";

            }

            else {

                echo      "<td>Прилетів          o
".$p[`arrivtime`]."</td>";

            }

        }

    }

    ?>

</td>

</tr>

<?php }?>

</tbody>

</table>

</div>

<ul
class="actions">

```

```

<?php
if
(!$mode) {

echo
"<li><a href="#" class="button big">ВИЛИТ</a></li>";

} else

echo
"<li><a href="#" class="button big">ПРИЛИТ</a></li>";

}

?>
</ul>
</div>
</section>

<div id="sidebar">
  <div class="inner">

      <section id="search"
class="alt">
          <form
method="post" action="search.php">
              <input
type="text" name="query" id="query" placeholder="Пошук рейсів" />
          </form>
          <?php include
"menu.php" ?>
      </section>
  </div>
</div>
<script src="assets/js/jquery.min.js"></script>
<script
src="assets/js/browser.min.js"></script>
<script
src="assets/js/breakpoints.min.js"></script>
<script src="assets/js/util.js"></script>
<script src="assets/js/main.js"></script>

</body>
</html>

```

Б.2 Текст файлу menu.php

```

<nav id="menu">

  <header class="major">
    <h2>Аеропорт</h2>
  </header>

```

```

<ul>
    <?php
        if(!isset($_SESSION["techmanager"]))
    {
        ?>
        <li><a href="timetable.php">ПОЗКЛАД</a></li>
        <?php } ?>

        <?php
            if(!isset($_SESSION["techmanager"]))
    {
        ?>
        <li><a href="index.php">ТАБЛО</a></li>
        <?php } ?>

        <?php
            if(isset($_SESSION["almanager"])) {
        ?>
        <li><a href="registration.php">РЕЄСТРАЦІЯ НА
РЕЙС</a></li>
        <?php } ?>

        <?php
            if(isset($_SESSION["login"])) {
        ?>
        <li><a href="tech.php">ОБСЛУГОВУВАННЯ</a></li>
        <?php } ?>

        <?php
            if(isset($_SESSION["apmanager"])) {
        ?>
        <li><a href="prediction.php">ПРОГНОЗУВАННЯ</a></li>
        <?php } ?>

        <?php
            if(isset($_SESSION["login"])) {
        ?>
        <li><a href="search/search.php">ПОШУК</a></li>
        <?php } ?>

        <?php
            if(isset($_SESSION["apmanager"])) {
        ?>
        <li>
            <span class="opener">ЗБИТИ</span>
            <ul>
                <li><a
href="report/reporttp.php">Пасажиропотік</a></li>
                <li><a href="
report/reports.php">Статистика</a></li>
            </ul>
        </li>
        <?php } ?>

        <?php

```

```

                                if(isset($_SESSION["login"])) {
?>
<li>
    <span class="opener">РОБОТА З ДАНИМИ</span>
    <ul>
        <li><a
href="data/airlines.php">Авіалінії</a></li>
        <li><a href="
data/planes.php">Літаки</a></li>
        <li><a href="
data/routes.php">Рейси</a></li>
        <li><a href="
data/person.php">Персонал</a></li>
        <li><a href="
data/crew.php">Екіпаж</a></li>
        <li><a href="
data/passengers.php">Пасажири</a></li>
    </ul>
</li>
<?php } ?>

<?php
                                if(!isset($_SESSION["login"])) {
?>
<li><a href="login.php">АВТОРИЗАЦІЯ</a></li>
<?php }
        else {
            echo "<li><a
href="logout.php">ВИЙТИ</a></li>";
        }
?>
</ul>

</nav>

```

Б.3 Текст файлу flights.php

```

<!DOCTYPE HTML>
<html>
    <head>
        <title>Дослідження та програмна реалізація методів
прогнозування пасажиропотоку аеропорту</title>
        <meta charset="utf-8" />
        <meta name="viewport" content="width=device-width,
initial-scale=1, user-scalable=no" />
        <link rel="stylesheet" href="assets/css/main.css" />
    </head>
    <body class="is-preload">
        <div id="wrapper">
            <div id="main">
                <div class="inner">
                    <header id="header">

```

```

                                <a
href="index.php" class="logo">Дослідження та програмна реалізація
методів прогнозування пасажиропотоку аеропорту</a>
                                </header>
                                <section id="banner">
                                    <div
class="content">

                                <?php include "db.php" ?>

                                <h3>Фактично виконані польоти</h3>

                                <h4>Вкажіть номер рейсу</h4>

                                <form method="post" action="#">

                                    <div class="row gtr-uniform">
                                        <div class="col-6 col-12-xsmall">

                                            <input type="text" name="route-num" id="route-
num" value="" placeholder="Номер" />

                                        </div>

                                    </div>

                                <br>

                                <ul
class="actions">

                                <li><input type="submit" name="setdateflightbtn"
value="ОБРАТИ"></li>

                                </ul>

                                </form>

                                </div>
                                </section>

                                <section>

                                <form method="post" action="addflight.php">

                                    <div class="row gtr-uniform">

                                        <div class="col-6 col-12-xsmall">

                                            <input type="submit" name="addflightbtn"
value="ДОДАТИ" />

                                        </div>

```

```

</div>

</form>

<div class="table-wrapper">

    <table>

        <thead>

            <tr>

                <th>ID</th>

                <th>Номер рейсу</th>

                <th>Маршрут</th>
                <th>Літак</th>

                <th>Номер літака</th>

                <th>Виліт</th>

                <th>Прибуття</th>

                <th>Редагувати</th>

                <th>Вилучити</th>

            </tr>

        </thead>

        <tbody>

            <?php $num = 0; ?>

            <?php

                if (isset($_GET['pnum'])) {

                    $pnum = $_GET['pnum'];

                } else {

                    $pnum = 1;

                }

                $flightspp = 10;

                $stepfl = ($pnum-1) * $flightspp;

                $flnum = "SELECT COUNT(*) FROM `Flight`

```

```

INNER JOIN `Route` ON Flight.race = Route.id WHERE Route.number
LIKE $_POST['route-num'].'%';

    $fl = mysqli_query($db,$flnum);

    $nm = mysqli_fetch_array($fl)[0];

    $flps = ceil($nm / $flightssp);

    $alldata = mysqli_query($db, "SELECT
Flight.id, Route.number, Route.places, Plane.manuf, Plane.model,
Plane.regnum, Flight.departtime, Flight.arrivtime FROM `Flight`
LEFT JOIN 'Route' ON Flight.race = Route.id LEFT JOIN 'Plane' ON
Flight. planenum = Plane.id WHERE Route.number LIKE $_POST['route-
num'].'%' LIMIT $stepfl, $flightssp");

    while($oned = mysqli_fetch_array($alldata)){ ?>
<?php $num = $num + 1; ?>

        <tr>

            <td><?php echo $oned[0]; ?></td>

            <td><?php echo $oned[1]; ?></td>

            <td><?php echo $oned[2]; ?></td>

            <td><?php echo $oned[3].$oned[4]; ?></td>

            <td><?php echo $oned[5]; ?></td>

            <td><?php echo $oned[6]; ?></td>

            <td><?php echo $oned[7]; ?></td>

            <?php $btnname = 'editflightbtn'.$num;

                echo          "<td><input          type='submit'
name="+$btnname+" value="РЕДАГУВАТИ" /></td>";

            ?>

            <?php $btnname = 'deleteflightbtn'.$num;

                echo          "<td><input          type='submit'
name="+$btnname+" value="ВИЛУЧИТИ" /></td>";

            ?>

        </tr>

    <?php }?>

    <tbody>

```

```

</table>

<ul class="allpgs">

    <li><a href="?pnum=1">Перша</a></li>

    <li class="<?php if($pnum <= 1){ echo 'unactive';
} ?>">

        <a href="<?php if($pnum <= 1){ echo '#'; }
else { echo "?pnum=" . ($pnum - 1); } ?>">Назад</a>

    </li>

    <li class="<?php if($pnum >= $flps){ echo
'unactive'; } ?>">

        <a href="<?php if($pnum >= $flps){ echo '#'; }
else { echo "?pnum=" . ($pnum + 1); } ?>">Вперед</a>

    </li>

    <li><a href="?pnum=<?php echo $flps;
?>">Остання</a></li>

</ul>

</section>

<div id="sidebar">
    <div class="inner">

        <section id="search"
class="alt">

            <form
method="post" action="search.php">

                <input
type="text" name="query" id="query" placeholder="Пошук рейсів" />
            </form>

            <?php include
"menu.php" ?>

        </section>

    </div>
</div>

<script src="assets/js/jquery.min.js"></script>
<script
src="assets/js/browser.min.js"></script>
<script
src="assets/js/breakpoints.min.js"></script>
<script src="assets/js/util.js"></script>
<script src="assets/js/main.js"></script>

</body>
</html>

```

Б.4 Текст файла AirportDB.sql

```

CREATE TABLE IF NOT EXISTS `Flight` (
  `id` INT NOT NULL AUTO_INCREMENT,
  `race` INT NOT NULL,
  `planenum` INT,
  `departtime` DATETIME,
  `arrivtime` DATETIME,
  PRIMARY KEY (`id`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8;

CREATE TABLE IF NOT EXISTS `Registration` (
  `id` INT NOT NULL AUTO_INCREMENT,
  `passnum` INT NOT NULL,
  `flightnum` INT NOT NULL,
  `placenum` varchar(4) NOT NULL,
  `type` INT NOT NULL,
  `price` FLOAT,
  PRIMARY KEY (`id`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8;

CREATE TABLE IF NOT EXISTS `Passenger` (
  `id` INT NOT NULL AUTO_INCREMENT,
  `name` varchar(255) NOT NULL,
  `passport` varchar(20) NOT NULL,
  `phone` varchar(13) NOT NULL,
  `nationality` varchar(20) NOT NULL,
  PRIMARY KEY (`id`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8;

CREATE TABLE IF NOT EXISTS `Route` (
  `id` INT NOT NULL AUTO_INCREMENT,
  `type` INT NOT NULL,
  `places` varchar(255) NOT NULL,
  `planetype` INT NOT NULL,
  `departtime` DATETIME NOT NULL,
  `flighttime` TIME NOT NULL,
  `line` INT NOT NULL,
  `timetable` INT NOT NULL,
  `ticketsnum` INT NOT NULL,
  `crewnum` INT NOT NULL,
  `fuel` INT NOT NULL,
  `enddate` DATE NOT NULL,
  `number` varchar(6) NOT NULL,
  PRIMARY KEY (`id`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8;

CREATE TABLE IF NOT EXISTS `Luggage` (
  `id` INT NOT NULL,
  `registr` INT NOT NULL,
  `type` INT NOT NULL,
  `weight` FLOAT NOT NULL,
  `length` FLOAT,
  `width` FLOAT,

```

```

        `height` FLOAT,
        PRIMARY KEY (`id`)
    ) ENGINE=InnoDB DEFAULT CHARSET=utf8;

```

```

CREATE TABLE IF NOT EXISTS `Airline` (
    `id` INT NOT NULL AUTO_INCREMENT,
    `name` varchar(255) NOT NULL,
    `country` varchar(20) NOT NULL,
    `regnum` varchar(100) NOT NULL,
    `description` TEXT,
    `createtime` DATE,
    PRIMARY KEY (`id`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8;

```

```

CREATE TABLE IF NOT EXISTS `Plane` (
    `id` INT NOT NULL AUTO_INCREMENT,
    `type` INT NOT NULL,
    `manuf` varchar(20) NOT NULL,
    `model` varchar(50) NOT NULL,
    `regnum` varchar(100) NOT NULL,
    `sernum` varchar(100) NOT NULL,
    `prodtdate` DATE NOT NULL,
    `speed` FLOAT,
    `passq` INT NOT NULL,
    `fuel` INT NOT NULL,
    `state` INT NOT NULL,
    PRIMARY KEY (`id`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8;

```

```

CREATE TABLE IF NOT EXISTS `TO` (
    `id` INT NOT NULL AUTO_INCREMENT,
    `planenum` INT NOT NULL,
    `type` INT NOT NULL,
    `starttime` DATETIME NOT NULL,
    `endtime` DATETIME,
    `personnum` INT NOT NULL,
    `description` TEXT,
    `result` varchar(100),
    PRIMARY KEY (`id`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8;

```

```

CREATE TABLE IF NOT EXISTS `Person` (
    `id` INT NOT NULL AUTO_INCREMENT,
    `name` varchar(255) NOT NULL,
    `passport` varchar(20) NOT NULL,
    `regaddr` varchar(100) NOT NULL,
    `idcode` varchar(10) NOT NULL,
    `contractend` DATE,
    `birthday` DATE NOT NULL,
    `position` varchar(50) NOT NULL,
    `phone` varchar(13) NOT NULL,
    `contractbegin` DATE NOT NULL,
    `description` TEXT,
    `login` varchar(20) NOT NULL,
    `password` varchar(20) NOT NULL,

```

```

        `type` INT NOT NULL,
        PRIMARY KEY (`id`)
    ) ENGINE=InnoDB DEFAULT CHARSET=utf8;

CREATE TABLE IF NOT EXISTS `Crew` (
    `id` INT NOT NULL AUTO_INCREMENT,
    `name` varchar(255) NOT NULL,
    `passport` varchar(20) NOT NULL,
    `country` varchar(20) NOT NULL,
    `birthday` DATE NOT NULL,
    `position` varchar(50) NOT NULL,
    `phone` varchar(13) NOT NULL,
    `exper` INT,
    `description` TEXT,
    `airlinenum` INT NOT NULL,
    PRIMARY KEY (`id`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8;
CREATE TABLE IF NOT EXISTS `FlightCrew` (
    `id` INT NOT NULL AUTO_INCREMENT,
    `crewnum` INT NOT NULL,
    `flightnum` INT NOT NULL,
    PRIMARY KEY (`id`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8;

ALTER TABLE `Flight`
ADD CONSTRAINT `Flight_fk0` FOREIGN KEY (`race`) REFERENCES
`Route`(`id`);

ALTER TABLE `Flight`
ADD CONSTRAINT `Flight_fk1` FOREIGN KEY (`planenum`)
REFERENCES `Plane`(`id`);

ALTER TABLE `Registration`
ADD CONSTRAINT `Registration_fk0` FOREIGN KEY (`passnum`)
REFERENCES `Passenger`(`id`);

ALTER TABLE `Registration`
ADD CONSTRAINT `Registration_fk1` FOREIGN KEY (`flightnum`)
REFERENCES `Flight`(`id`);

ALTER TABLE `Route`
ADD CONSTRAINT `Route_fk0` FOREIGN KEY (`line`) REFERENCES
`Airline`(`id`);

ALTER TABLE `Luggage`
ADD CONSTRAINT `Luggage_fk0` FOREIGN KEY (`registr`)
REFERENCES `Registration`(`id`);

ALTER TABLE `TO`
ADD CONSTRAINT `TO_fk0` FOREIGN KEY (`planenum`) REFERENCES
`Plane`(`id`);

ALTER TABLE `TO`
ADD CONSTRAINT `TO_fk1` FOREIGN KEY (`personnum`) REFERENCES
`Person`(`id`);

```

```
ALTER TABLE `Crew`
ADD CONSTRAINT `Crew_fk0` FOREIGN KEY (`airlinenum`)
REFERENCES `Airline`(`id`);
```

```
ALTER TABLE `FlightCrew`
ADD CONSTRAINT `FlightCrew_fk0` FOREIGN KEY (`crewnum`)
REFERENCES `Crew`(`id`);
```

```
ALTER TABLE `FlightCrew`
ADD CONSTRAINT `FlightCrew_fk1` FOREIGN KEY (`flightnum`)
REFERENCES `Flight`(`id`);
```

Б.5 Текст файлу addroute.php

```
<!DOCTYPE HTML>
<html>
  <head>
    <title>Дослідження та програмна реалізація методів
прогнозування пасажиропотоку аеропорту</title>
    <meta charset="utf-8" />
    <meta name="viewport" content="width=device-width,
initial-scale=1, user-scalable=no" />
    <link rel="stylesheet" href="assets/css/main.css" />
  </head>
  <body class="is-preload">
    <div id="wrapper">
      <div id="main">
        <div class="inner">
          <header id="header">
            <a
href="index.php" class="logo">Дослідження та програмна реалізація
методів прогнозування пасажиропотоку аеропорту</a>
          </header>
          <section id="banner">
            <div
class="content">

      <?php include "db.php" ?>

      <h3>Додати рейс</h3>

      <div class="table-wrapper">

        <table>

          <thead>

            <tr>

              <th>Номер рейсу</th>

              <th>Тип</th>
```

```

<th>Маршрут</th>

</tr>

</thead>

<tbody>

<tr>

<th>

<div class="col-6 col-12-xsmall">
  <input type="text" name="rnum"
value="" />

</div>

</th>

<th>

<div class="col-12">

<select name="rtype"
id="rtype">

<option
value="">Постійний</option>

<option
value="1">Постійний</option>

<option
value="2">Тимчасовий</option>

<option
value="3">Тестовий</option>

</select>

</div>

</th>

<th>

<div class="col-6 col-12-xsmall">

<input
name="rplaces" value="" />

</div>

</th>

```

</tr>

</tbody>

</table>

</div>

<div class="table-wrapper">

<table>

<thead>

<tr>

<th>Тип літака</th>

<th>Час відправлення</th>

<th>Час прибуття</th>

</tr>

</thead>

<tbody>

<tr>

<th>

<div class="col-6 col-12-xsmall">

<input type="text"

name="rpltype" value="" />

</div>

</th>

<th>

<div class="col-6 col-12-xsmall">

<input type="time" name="rarrt"

value="" />

</div>

</th>

<th>

<div class="col-6 col-12-xsmall">


```

                                $j = $j + 1;
                                echo                                "<option
value=" . $j . ">" . $oned[0] . "</option>";
                                } ?>
                                </select>
                                </div>
                                </th>
                                <th>
                                <div class="col-6 col-12-xsmall">
                                <input                                type="number"
name="rcrew" value="" />
                                </div>
                                </th>
                                <th>
                                <div class="col-6 col-12-xsmall">
                                <input                                type="number"
name="rfuel" value="" />
                                </div>
                                </th>
                                </tr>
                                </tbody>
                                </table>
                                </div>
                                <div class="table-wrapper">
                                <table>
                                <thead>
                                <tr>
                                <th>Розклад</th>
                                </tr>

```

</thead>

<tbody>

<tr>

<th>

<div class="col-6 col-12-

xsmall">

<input type="checkbox"

id="day0" name="day0">

<label

for="day0">Неділя</label>

<input type="checkbox"

id="day1" name="day1">

<label

for="day1">Понеділок</label>

<input type="checkbox"

id="day2" name="day2">

<label

for="day2">Вівторок</label>

<input type="checkbox"

id="day3" name="day3">

<label

for="day3">Середа</label>

<input type="checkbox"

id="day4" name="day4">

<label

for="day4">Четвер</label>

<input type="checkbox"

id="day5" name="day5">

<label

for="day5">П'ятниця</label>

<input type="checkbox"

id="day6" name="day6">

<label

for="day6">Субота</label>

</div>

</th>

```

</tr>

</tbody>

</table>

</div>

<div class="table-wrapper">

<table>

<thead>

<tr>

<th>Дата завершення польотів</th>

</tr>

</thead>

<tbody>

<tr>

<th>

<div class="col-6 col-12-xsmall">

<input type="date" name="rendd"

value="" />

</div>

</th>

</tr>

</tbody>

</table>

</div>

<form method="post" action="addflightdb.php">

<div class="row gtr-uniform">

<div class="col-6 col-12-xsmall">

<input type="submit" name="addflightbtn"

value="ЗБЕРЕГТИ" />

</div>

```

</div>

</form>

</div>

</section>

</div>

</div>

<div id="sidebar">

<div class="inner">

<section id="search"

class="alt">

<form

method="post" action="search.php">

<input

type="text" name="query" id="query" placeholder="Пошук рейсів" />

</form>

<?php include

"menu.php" ?>

</section>

</div>

</div>

<script src="assets/js/jquery.min.js"></script>

<script

src="assets/js/browser.min.js"></script>

<script

src="assets/js/breakpoints.min.js"></script>

<script src="assets/js/util.js"></script>

<script src="assets/js/main.js"></script>

</body>

</html>

ДОДАТОК В
Слайди презентації

Національний університет «Запорізька політехніка»
Кафедра програмних засобів

Дипломна кваліфікаційна робота магістра

**ДОСЛІДЖЕННЯ ТА ПРОГРАМНА РЕАЛІЗАЦІЯ
МЕТОДІВ ПРОГНОЗУВАННЯ
ПАСАЖИРОПОТОКУ АЕРОПОРТУ**

Виконав

студент групи КНТ-219м

А. К. Васильєв

Керівник

к.т.н., доцент

І. Я. Зеленцова

2020

Рисунок В.1 – Слайд 1

Об'єкт дослідження – процес прогнозування
пасажиропотоку аеропорту.

Предмет дослідження – програмне
забезпечення підтримки роботи аеропорту.

Мета роботи – розроблення програмного
забезпечення прогнозування пасажиропотку
аеропорту для підвищення автоматизованості
даного процесу та обґрунтованості рішень, що
приймаються відносно планування діяльності
аеропорту.

2

Рисунок В.2 – Слайд 2

Завдання

- ✓ розроблення методу прогнозування пасажиропотоку аеропорту;
- ✓ проєктування вебзастосунку;
- ✓ програмна реалізація та тестування вебзастосунку.

3

Рисунок В.3 – Слайд 3

Метод прогнозування пасажиропотоку аеропорту

Модель, яка використовується в роботі методу, може бути представлена виразом:

$$y_{j+k+1} = f(y_{j+k}, y_{j+k-1}, \dots, y_j, (k+1) \bmod T, x_j^{j+k}),$$

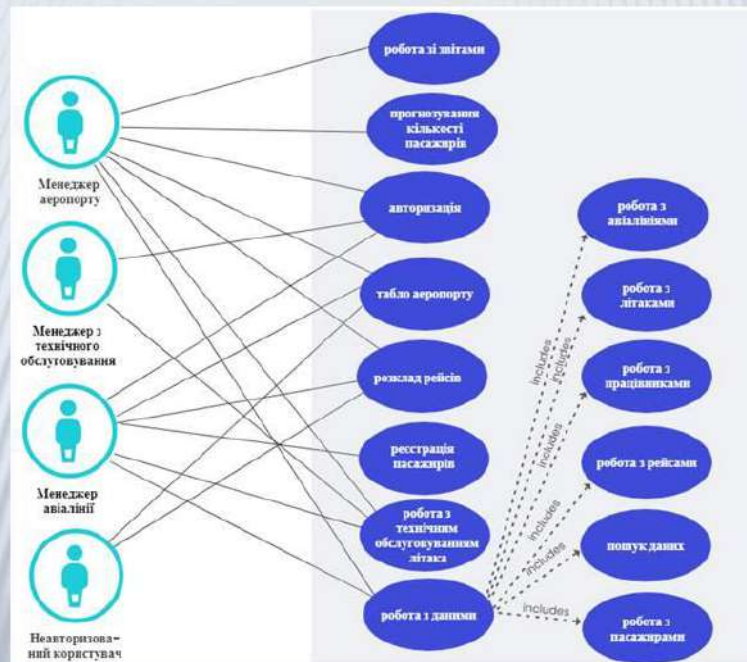
де $y_{j+k+1}, y_{j+k}, y_{j+k-1}, \dots, y_j$ – кількість пасажирів у періоди часу $j+k+1, j+k, j+k-1, \dots, j$ відповідно;

x_j^{j+k} – коливання курсу валют протягом періоду часу з j до $j+k$.

4

Рисунок В.4 – Слайд 4

Діаграма прецедентів



5

Рисунок В.5 – Слайд 5

Порівняння мов PHP та Python для веброзробки

Фактор	PHP	Python
Призначення	Загальне, але для веброзробки	Загальне
Веброзробка	+	+
Безкоштовність	+	+
Робота з базами даних	Широкий вибір (більше 20 систем)	Є підтримка

6

Рисунок В.6 – Слайд 6

The screenshot displays a database design tool interface with 11 tables, each represented by a colored card. The tables and their attributes are as follows:

- Agent** (Orange):
 - id (integer, PK)
 - password (integer)
 - loginnum (integer)
 - phonenum (varchar(15))
 - type (integer)
 - price (float)
- Flight** (Purple):
 - id (integer, PK)
 - route (integer)
 - time (integer)
 - plannum (integer)
 - departure (datetime)
 - arrivaltime (datetime)
- Passenger** (Green):
 - id (integer, PK)
 - name (varchar(255))
 - passport (varchar(20))
 - phone (varchar(13))
 - nationality (varchar(20))
- Language** (Dark Purple):
 - id (integer, PK)
 - register (integer)
 - type (integer)
 - weight (float)
 - length (float)
 - width (float)
 - height (float)
- Route** (Blue):
 - id (integer, PK)
 - type (integer)
 - place (varchar(255))
 - planarity (integer)
 - departure (datetime)
 - flighttime (time)
 - time (integer)
 - timetable (integer)
 - ticketnum (integer)
 - crewnum (integer)
 - fuel (integer)
 - endow (date)
 - number (varchar(8))
- FlightCrew** (Dark Green):
 - id (integer, PK)
 - crewnum (integer)
 - cinewum (integer)
- Crew** (Teal):
 - id (integer, PK)
 - name (varchar(255))
 - passport (varchar(20))
 - country (varchar(20))
 - position (varchar(40))
 - phonum (varchar(13))
 - exper (integer)
 - description (text)
 - airlineum (integer)
- Airline** (Olive):
 - id (integer, PK)
 - name (varchar(255))
 - country (varchar(20))
 - region (varchar(100))
 - description (text)
 - create (date)
- Plane** (Dark Blue):
 - id (integer, PK)
 - type (integer)
 - manuf (varchar(20))
 - model (varchar(10))
 - region (varchar(100))
 - vernum (varchar(100))
 - prothdate (date)
 - speed (float)
 - passig (integer)
 - fuel (integer)
 - state (integer)
- Person** (Pink):
 - id (integer, PK)
 - name (varchar(255))
 - password (varchar(20))
 - userid (varchar(100))
 - ofcode (varchar(20))
 - contractand (date)
 - workday (date)
 - position (varchar(50))
 - show (varchar(10))
 - contractbegin (date)
 - contractend (date)
 - description (text)
 - login (varchar(20))
 - password (varchar(255))
 - type (integer)

Arrows indicate foreign key relationships between attributes in different tables:

- Agent** to **Flight**: Agent.id to Flight.route
- Flight** to **Plane**: Flight.route to Plane.type
- Passenger** to **Flight**: Passenger.id to Flight.time
- Flight** to **Route**: Flight.id to Route.id
- Flight** to **FlightCrew**: Flight.id to FlightCrew.id
- Flight** to **Crew**: Flight.id to Crew.id
- Flight** to **Airline**: Flight.id to Airline.id
- Flight** to **Person**: Flight.id to Person.id
- Flight** to **Person**: Flight.time to Person.contractand
- Flight** to **Person**: Flight.planum to Person.userid
- Flight** to **Person**: Flight.departure to Person.contractbegin
- Flight** to **Person**: Flight.arrivaltime to Person.contractend
- Flight** to **Person**: Flight.fuel to Person.fuel
- Flight** to **Person**: Flight.number to Person.password
- Flight** to **Person**: Flight.endow to Person.login
- Flight** to **Person**: Flight.number to Person.password
- Flight** to **Person**: Flight.number to Person.type

Рисунок В.7 – Слайд 7

The diagram illustrates the experimental environment. On the left, a PC labeled 'Экспериментальный компьютер' (Experimental computer) contains a 'Базовый HTTP сервер' (Basic HTTP server). This PC is connected via a 'TCP/IP или IP-адресация' (TCP/IP or IP addressing) connection to a 'Web-Server' box. The 'Web-Server' box contains a grid of 14 files, each with a PHP icon: `<<artifacts>> index.php`, `<<artifacts>> login.php`, `<<artifacts>> favorite.php`, `<<artifacts>> menu.php`, `<<artifacts>> register.php`, `<<artifacts>> register.php`, `<<artifacts>> profile.php`, `<<artifacts>> reports.php`, `<<artifacts>> search.php`, `<<artifacts>> reports.php`, `<<artifacts>> test.php`, `<<artifacts>> games.php`, `<<artifacts>> action.php`, and `<<artifacts>> action.php`. The 'Web-Server' is also connected via a 'PHPmySQL' connection to a 'MySQL-DB-Server' box on the right, which contains a file `<<artifacts>> AirportDB.db`.

Рисунок В.8 – Слайд 8

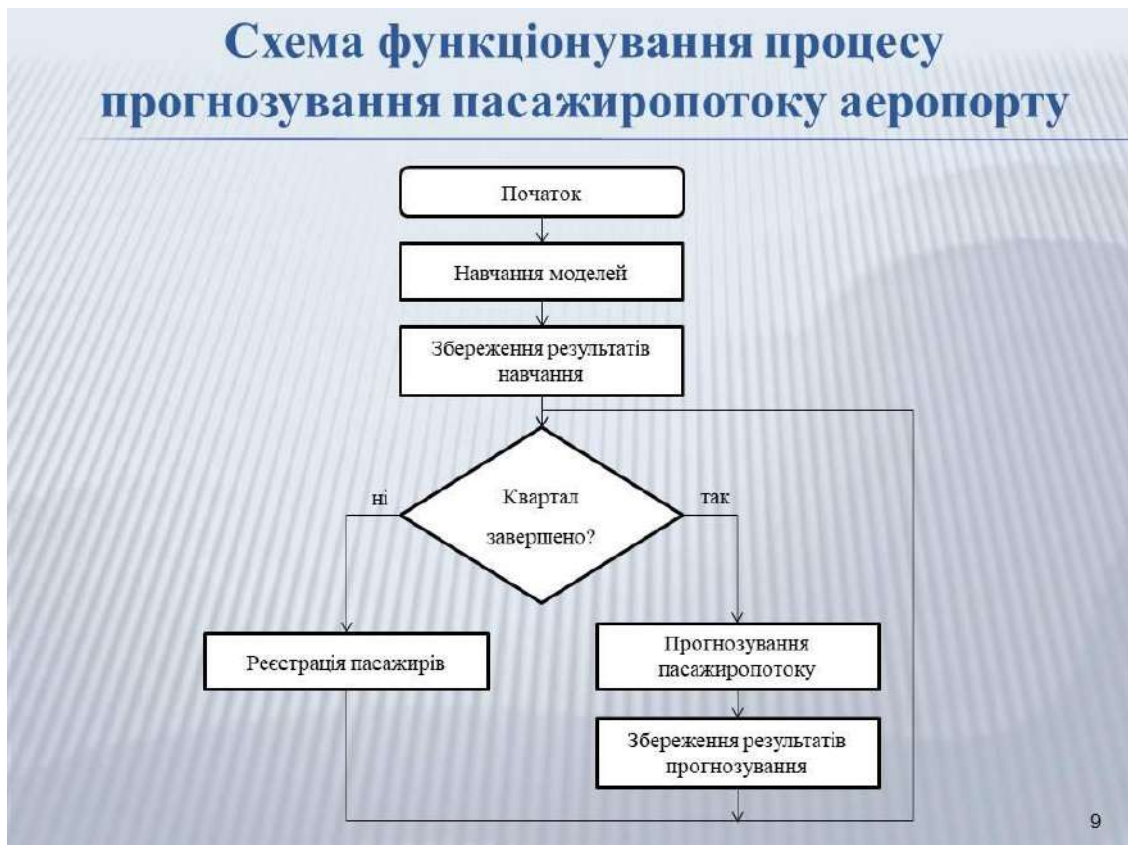


Рисунок В.9 – Слайд 9

Вебзастосунок

Пошук рейсів

Аеропорт

РОЗКЛАД
ТАБЛО
АВТОРИЗАЦІЯ

Дослідження та програмна реалізація методів прогнозування пасажиропотоку аеропорту

Табло аеропорту

Прибуття

Авіалінія	Маршрут	Виліт	Прибуття	Стан
SkyUp	Київ-Запоріжжя	20:20	21:45	Затриманий
Wizz Air	Варшава-Запоріжжя	19:30	16:00	Затриманий
Рейв еспресс	Львів-Запоріжжя	12:10	13:55	Затриманий
Vueling	Мадрид-Запоріжжя	10:15	13:05	Скінуватися, виліт 10:21
SkyUp	Київ-Запоріжжя	08:00	09:15	Прілетів о 9:07

Регістрація пасажирів

Обрати рейс

10:52 Запоріжжя-Київ

Ввести дані пасажирів

Прізвище, ім'я, по батькові

Адреса (заповнювати)

Телефон

10:52 Запоріжжя-Київ

10:52 Запоріжжя-Київ

Рисунок В.10 – Слайд 10

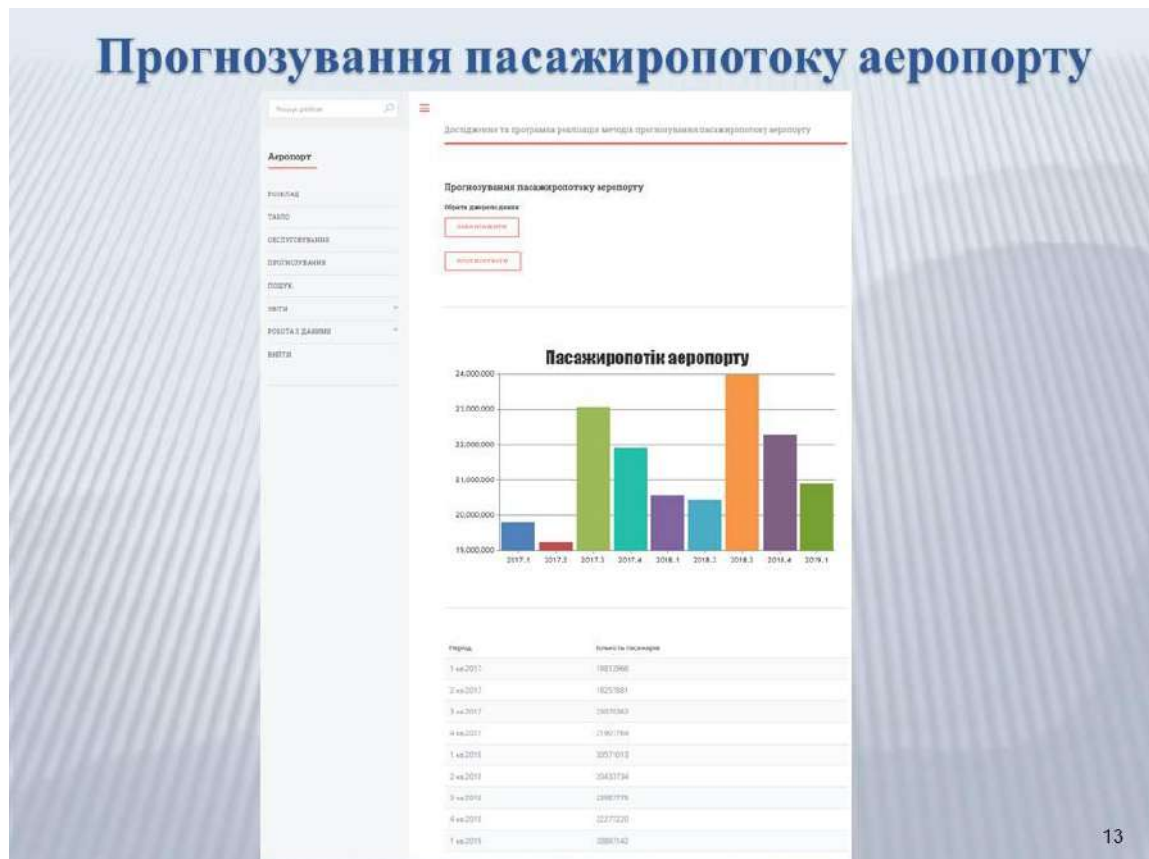
11

Функціональні особливості

12

Рисунок В.12 – Слайд 12

Прогнозування пасажиропотоку аеропорту



13

Рисунок В.13 – Слайд 13

Висновки

Наукова новизна роботи полягає в запропонованому методі прогнозування пасажиропотоку аеропорту, який ґрунтується на використанні методу опорних векторів з квадратичною функцією втраф, використовує в якості входних ознак пасажиропотік за минулі періоди, коливання курсу валют за цей період та номер базового періоду.

У роботі визначено функціональні вимоги до програми, сформульовано діаграму прецедентів для роботи менеджера аеропорту, менеджера з технічного обслуговування, менеджера авіалінії та неавторизованого користувача з програмою. Було виконано порівняння засобів реалізації програми (мов PHP і Python), в результаті чого мову програмування PHP та систему керування базами даних MySQL було обрано для реалізації. Було запроектовано базу даних, результат проектування сформульовано у вигляді схеми бази даних.

Реалізовано вебзастосунок для прогнозування пасажиропотоку аеропорту. Вебзастосунок дозволяє виконувати облік діяльності аеропорту, включаючи роботу з авіалініями, виконанням рейсів, технічним обслуговуванням, а потім автоматизовано виконувати прогнозування пасажиропотоку на основі автоматично зібраних даних щодо реєстрації пасажирів за минулі періоди. Описано структуру розробленого вебзастосунку з приведенням основних реалізованих скриптів. Створено діаграму розгортання. Описано процес прогнозування пасажиропотоку аеропорту зі створенням схеми функціонування даного процесу.

Усі завдання дипломної кваліфікаційної роботи магістра виконано повністю.

14

Рисунок В.14 – Слайд 14