

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

Національний університет кораблебудування імені адмірала Макарова
Навчально-науковий інститут комп'ютерних наук та управління проектами
Кафедра інформаційних управляючих систем та технологій

"Допущений до захисту"

Завідувач кафедри ІУСТ

Гарант освітньої програми

ДОЦЕНТ, К.Т.Н.

(посада, науковий ступінь вчене звання)

_____ Михелєв І.Л.
(підпис)

" ____ " _____ 2022 р.

КВАЛІФІКАЦІЙНА РОБОТА

на здобуття ступеня вищої освіти "бакалавр"

на тему: "Розробка інформаційної системи авіакомпанії онлайн бронювання та
реалізації авіаквитків"

Виконав: студент групи 4142

_____ Міхняєв К.Л.
(підпис)

Керівник роботи:

ДОЦЕНТ, К.Т.Н.

(посада, науковий ступінь вчене звання)

_____ Ажищев В.Ф.
(підпис)

м. Миколаїв – 2022 р.

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

Національний університет кораблебудування імені адмірала Макарова
 Навчально-науковий інститут комп'ютерних наук та управління проектами
 Кафедра інформаційних управляючих систем та технологій
 Спеціальність 122 "Комп'ютерні науки"
 Освітня програма "Комп'ютерні науки"

"ЗАТВЕРДЖУЮ"

Гарант освітньої програми

ДОЦЕНТ, К.Т.Н.

(посада, науковий ступінь вчене звання)

Гайда А.Ю.

(підпис)

" ____ " ____ 2022 р.

ЗАВДАННЯ НА КВАЛІФІКАЦІЙНУ РОБОТУ на здобуття ступеня вищої освіти "бакалавр"

Студенту Міхняєву Костянтину Леонідовичу

(прізвище, ім'я, по батькові)

1. Тема роботи: "Розробка інформаційної системи авіакомпанії онлайн бронювання та реалізації авіаквитків"

Керівник роботи Ажищев Віктор Федорович

Затверджені наказом ректора № 269-уч від "18" травня 2022 року.

2. Термін подання роботи: 20 червня 2022 р.

3. Вихідні дані до проекту (роботи): ДСТУ щодо обробки інформації, літературні джерела, технічна документація на існуючі аналоги систем

4. Перелік питань, що належать до розробки (найменування розділів): Вступ. 1.Розробка концепції інформаційної системи. 2.Розробка технічного проекту інформаційної системи. 3.Реалізація проекту інформаційної системи. 4.Охорона праці. Висновки.

5. Перелік презентаційних матеріалів: Слайди: Розробка інформаційної системи авіакомпанії онлайн бронювання та реалізації авіаквитків; Актуальність, ціль, завдання; Об'єкт дослідження; Концепція системи; Технічний проект; Моделі бази даних; Програмне забезпечення; Технічне забезпечення; Використання системи.

6. Консультанти розділів роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
1	Ажищев В.Ф., доцент		
2	Ажищев В.Ф., доцент		
3	Ажищев В.Ф., доцент		
4	Ажищев В.Ф., доцент		

7. Дата видачі завдання 14 лютого 2022 р.

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів дипломного проекту (роботи)	Строк виконання етапів роботи	Примітка
1	Розробка змісту інформаційної системи		
2	Розробка концепції інформаційної системи		
3	Розробка технічного проекту інформаційної системи		
4	Реалізація проекту інформаційної системи		
5	Охорона праці		
6	Вступ, висновки, список використаної літератури, додатки		
7	Презентація		
8			
9			
10			

Студент

(підпис)

Міхняєв К.Л.

(ПІБ)

Керівник роботи

(підпис)

Ажищев В.Ф.

(ПІБ)

Анотація

Дипломний проект на тему: “Розробка системи авіакомпанії онлайн бронювання та реалізації авіаквитків” має об’єм 86 сторінок, містить розділи – 4, рисунки – 25, використані джерела – 32.

Ціль проекту полягає у підвищенні ефективності роботи авіакомпанії за рахунок впровадження інформаційної системи з віддаленого бронювання та реалізації авіаквитків.

Завдання проекту полягає у розробці інформаційної системи.

У першому розділі зроблений опис об’єкту дослідження і аналіз предметної області, сформовані вимоги, на основі яких розроблена концепція інформаційної системи.

У другому розділі розроблений технічний проект системи. Представлені загальносистемні рішення для вибору типових програмних засобів. Спроектовано базу даних. Описані алгоритми роботи проекту. Визначені програмні і технічні засоби реалізації і експлуатації проекту.

У третьому розділі розроблена робоча документація та проведена апробація проекту, яка показала коректну роботу усіх програмних модулів інформаційної системи.

У четвертому розділі представлені правила охорони праці під час роботи за комп’ютером.

Актуальність та новизна проекту полягає у підвищенні ефективності роботи та конкурентоспроможності авіакомпанії за рахунок web-додатка бронювання та реалізації авіаквитків та впровадження на цій основі інформаційної системи.

Ключові слова: авіакомпанія, авіаквиток, квиток, бронювання, реалізація, купівля, модель, схема, рисунок.

Annotation

Thesis project on the topic: "Development of the airline's online booking and sale of airline tickets" has a volume of 86 pages, contains sections – 4, figures – 25, used sources – 32.

The aim of the project is to increase the efficiency of the airline through the introduction of an information system for remote booking and sale of tickets.

The task of the project is to develop an information system.

The first section describes the object of study and analysis of the subject area, formed the requirements on the basis of which the concept of information system.

In the second section the technical design of the system is developed. System-wide solutions for the selection of standard software are presented. A database has been designed. Algorithms of project work are described. Software and hardware for project implementation and operation are identified.

In the third section, working documentation was developed and the project was tested, which showed the correct operation of all software modules of the information system.

The fourth section presents the rules of labor protection when working on a computer.

The relevance and novelty of the project is to increase the efficiency and competitiveness of the airline through a web-application for booking and selling tickets and the introduction of information systems on this basis.

Keywords: airline, air ticket, ticket, booking, sale, purchase, model, scheme, drawing.

Зміст

Вступ.....	7
Розділ 1. Розробка концепції інформаційної системи авіакомпанії онлайн бронювання та реалізації авіаквитків.....	8
1.1 Опис об'єкту дослідження і аналіз предметної області.....	8
1.2 Вимоги до інформаційної системи.....	19
1.3 Розробка концепції інформаційної системи.....	23
Розділ 2. Розробка технічного проекту інформаційної системи авіакомпанії онлайн бронювання та реалізації авіаквитків.....	27
2.1 Загальносистемні рішення.....	27
2.2 Рішення по інформаційному забезпеченню.....	31
2.2.1 Розробка концептуальної моделі даних.....	31
2.2.2 Розробка логічної моделі даних.....	33
2.2.3 Розробка фізичної моделі даних.....	33
2.3 Рішення з математичного забезпечення.....	35
2.4 Рішення з програмного забезпечення.....	39
2.5 Рішення з технічного забезпечення.....	50
Розділ 3. Реалізація проекту інформаційної системи авіакомпанії онлайн бронювання та реалізації авіаквитків.....	52
Розділ 4. Охорона праці.....	57
Висновки.....	63
Список використаної літератури.....	64
Додаток А - Інструкція користувачу.....	67
Додаток Б - Текст програми.....	73

Вступ

Темою дипломної роботи є розробка інформаційної системи авіакомпанії онлайн бронювання та реалізації авіаквитків. Ціллю інформаційної системи є надання можливості бронювати та купувати авіаквитки у будь-який час, у будь-якому місці, незалежно від розкладу авіакомпанії чи аеропорту, через web-додаток.

У першому розділі проведена розробка концепції інформаційної системи. Зроблений опис об'єкту дослідження і аналіз предметної області. Сформовані вимоги до інформаційної системи. На основі сформованих вимог розроблена концепція, яка визначає функціонал інформаційної системи.

У другому розділі розроблений технічний проект інформаційної системи. Наведені загальносистемні рішення для обробки електронних платежів, введення сайту, формування та розсилання електронних листів. У інформаційному забезпеченні представлені концептуальна, логічна та фізична моделі бази даних. У математичному забезпеченні представлена блок-схема алгоритму роботи проекту. У програмному забезпеченні описані засоби та технології розробки проекту. У технічному забезпеченні описані характеристики пристроїв користувача та адміністратора для задовільної експлуатації інформаційної системи.

У третьому розділі представлена робоча документація, де описані механізми підказок для форм, полів, кнопок та перевірки даних введених користувачем. Представлені знімки екрану з працюючою інформаційною системою.

У четвертому розділі представлені правила охорони праці під час роботи за комп'ютером.

Актуальність та новизна проекту полягає у підвищенні зручності купівлі та бронювання авіаквитків за рахунок можливості зробити це віддалено з персонального комп'ютеру, смартфона, планшета або будь-якого іншого пристрою, який може запускати браузер. Це дозволило підвищити конкурентоспроможність у сфері цивільної авіації та привабливість для потенційних клієнтів.

Розробка інформаційної системи авіакомпанії онлайн бронювання та реалізації авіаквитків.

1. Розробка концепції інформаційної системи авіакомпанії онлайн бронювання та реалізації авіаквитків.

1.1 Опис об'єкту дослідження і аналіз предметної області.

З винаходом повітряного транспорту стало можливим перевозити різні вантажі і не тільки, повітрям, заощаджуючи цим час і гроші. Найпоширенішим повітряним транспортом досі є літак. Перевагами такого виду транспорту стали мала ціна та час подорожі, значні зручності порівняно з іншими видами транспорту. Значний відсоток усіх авіаційних перевезень становлять перевезення цивільного населення.

Нині даним видом перевезень займаються авіакомпанії, фірми, які отримали спеціальну ліцензію від держави. Авіакомпанії розрізняються за масштабом як невеликі, що здійснюють регіональні перевезення, так і великі, що займаються міжнародними та міжконтинентальними перевезеннями.

Схема представництва авіакомпаній у різних аеропортах представлена на рисунку 1.1.

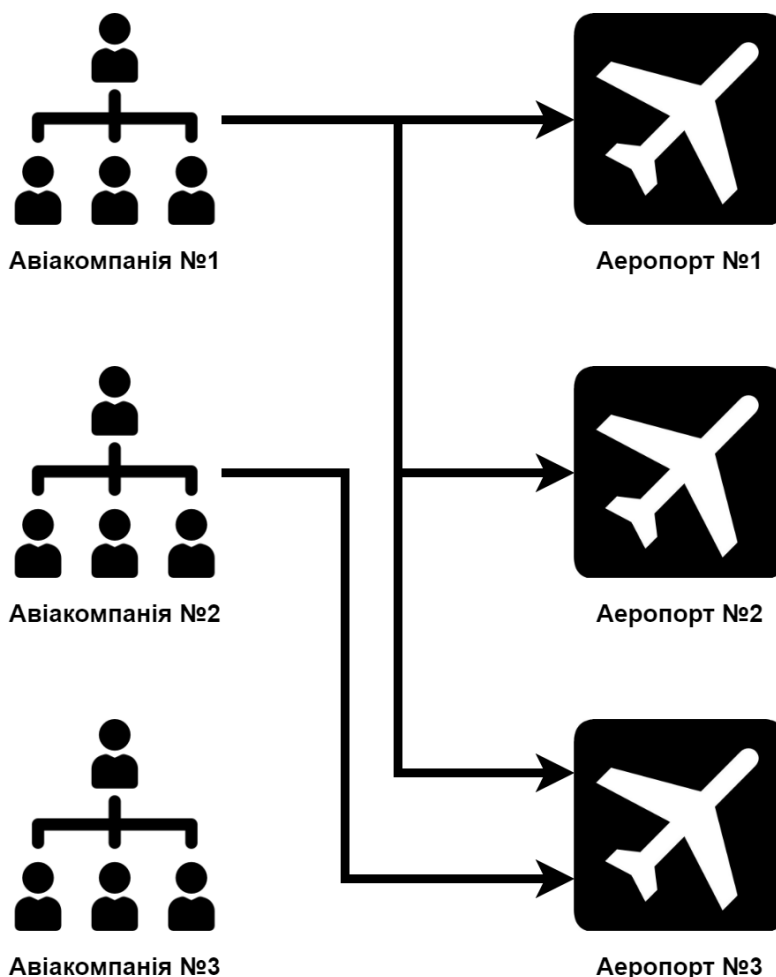


Рисунок 1.1 – Схема представництв авіакомпаній у різних аеропортах

Аеропорти та авіакомпанії мають симбіотичні відносини – вони потребують один одного, щоб процвітати. Зрештою, і аеропорти, і авіакомпанії обслуговують одного клієнта: пасажирів. Однак вони залишаються окремими структурами, підпорядкованими законам і договорам, які регулюють повітряний простір і забезпечують справедливе ділове середовище, а в деяких випадках переслідують інші цілі.

Авіакомпанії пов’язані з аеропортами через партнерство, і, як знає кожен, хто намагався встигнути на рейс, важливо, щоб ваш перевізник був доступний в аеропорту. Партнерство авіакомпаній – це те, що спрямовує трафік до аеропорту, наповнюючи його бізнес і магазини клієнтами, поки вони очікують вильоту.

Аеропорти є невід'ємною частиною авіакомпаній, оскільки вони забезпечують такі засоби, як злітно-посадкова смуга, засоби обробки та вантажні засоби, необхідні авіакомпаніям для ведення бізнесу. В одному аеропорті можуть мати представництва у безлічі різних авіакомпаній. Також є варіанти коли авіакомпанія не має представництва ні в одному з аеропортів (рисунок 1.1).

Оскільки подорожі розширюються на міжнародному рівні, аеропорти все більше відіграють життєво важливу роль як сполучний зв'язок. Хоча авіакомпанії можуть мати потужні локальні мережі для продажу та просування, коли вони переміщуються на міжнародний рівень, ці зв'язки часто набагато слабкіші, якщо вони взагалі існують. Аеропорти призначення можуть допомогти стимулювати попит на міжнародні авіакомпанії, приносячи користь авіакомпанії, оскільки місцеві відносини надають довіри та забезпечуючи продовження планування маршруту авіакомпанії до пункту призначення цього аеропорту.

Зміна в політиці, яку називають дерегуляцією або лібералізацією авіакомпаній, змінила відносини між авіакомпанією та аеропортом з регульованих урядом відносин до таких, які існують на розсуд авіакомпанії. Це відкрило двері для переговорів, оскільки, озброївшись свободою вибору, авіакомпанії тепер можуть вести переговори за найкращою ціною, коли запускають нову послугу. Деретуляція також перемістила джерело доходу аеропортів від авіакомпаній, які раніше сплачували гарантований збір, на прибуток безпосередньо від пасажирів.

Загалом між аеропортами та авіакомпаніями формуються три різні типи відносин:

- Угоди – вони обговорюються і часто включають компроміси.
- Ставки за рішенням – вони встановлюються після консультацій, часто включаючи постанови, положення та дозволи.
- Гібридний – у деяких випадках аеропорти можуть встановлювати тарифи у партнерстві з розподілом доходу.

Відносини також будуть відрізнятися залежно від країни. Наприклад, у Сполучених Штатах аеропорт діє більше як координатор послуг і орендодавець, тоді як у Франції та Великобританії аеропорт володіє сумішню державних і приватних компаній, що позиціонує авіакомпанію як більшого клієнта аеропорту. Не тільки державна та приватна власність може диктувати відносини між аеропортами та авіакомпаніями, але й державне регулювання, включаючи положення, що стосуються плати за обслуговування пасажирів, плати за посадку та розвитку, може вплинути на відносини. Кожен суб'єкт у відносинах має конкретні різні цілі.

Вимоги авіакомпаній:

- Оптимізований дохід завдяки ефективності;
- Швидкі терміни виконання;
- Мінімальні аеропортові збори;
- Додаткова місткість аеропорту;
- Зручні розклади;
- Якісний наземний транспорт;
- Виняткове обслуговування клієнтів;
- Популярні заклади харчування та магазини.

Вимоги аеропорти:

- Стабільна діяльність;
- Зростання;
- Використання, яке відповідає потребам усіх операторів, а не лише одного;
- Можливість залучати пасажирів як глобальний центр через мережу авіакомпаній;
- Різноманітність у відносинах авіакомпаній.

Аеропорти працюють, щоб збалансувати свої інтереси з (часто різними) інтересами кількох авіакомпаній. Як і будь-які симбіотичні стосунки, близькі

стосунки можуть створити певну напругу, але в кінцевому підсумку обом сторонам краще разом, а не працювати ізольовано.

Інформація про взаємодію аеропортів та авіакомпаній підготовлені на основі матеріалу [19].

Основні завдання авіакомпанії у сфері обслуговування пасажирських перевезень:

- Експлуатація закріплених за ними внутрішніх та міжнародних ліній на основі свідоцтва експлуатанта;
- Підтримка льотної придатності повітряних суден.
- Забезпечення якісного обслуговування авіаційної техніки та наземного обладнання;
- Забезпечення безпеки та регулярності польотів, високої культури обслуговування пасажирів на внутрішніх та міжнародних авіалініях;
- Підготовка кваліфікованих кадрів для польотів як на внутрішніх, так і на міжнародних авіалініях;
- Взаємодія з різними міністерствами, відомствами і т. д., спрямоване на поліпшення якості обслуговування споживачів повітряного транспорту.

Функції авіакомпаній у сфері обслуговування пасажирських перевезень:

- Забезпечення безпеки польотів з питань організації та проведення обслуговування повітряних перевезень;
- Забезпечення високої регулярності відправлень повітряного транспорту авіакомпаній, систематичний аналіз регулярності їх відправлень, вживаючи оперативні заходи щодо усунення причин порушення регулярності руху, що залежать від служб авіакомпаній;
- Участь у розслідуванні причин затримок рейсів, порушень технологічного графіка обслуговування, допущених з вини служб авіакомпаній;
- Забезпечення якості проведення сервісу повітряних перевезень та надання додаткових послуг пасажирам;

- Здійснення заходи, спрямовані на вдосконалення організації повітряних перевезень, підвищення якості та культури обслуговування пасажирів;
- Прийом заявок на перевезення пасажирів, у тому числі на організацію чартерних рейсів;
- Вжиття заходів щодо підвищення комерційного завантаження повітряного транспорту та рентабельності повітряних перевезень;
- Надавати послуги з продажу повітряних перевезень;
- Здійснювати розрахунок за повітряне перевезення;
- Здійснювати спеціальне обслуговування окремих категорій пасажирів: бізнес класу, учасників спеціальних програм;
- Організація обслуговування транзитних та трансферних пасажирів та обробку їх багажу на рейсах авіакомпаній, надання їм готелю та харчування, відповідно до стандартів авіакомпаній;
- Забезпечення якісне оформлення перевізних документів;
- Координація роботи служби сервісу з організації перевезень та наземного обслуговування з державними органами на повітряному транспорті;
- Організація експертизи для встановлення причин недостачі, пошкодження багажу;
- Проведення претензійної роботи за скаргами пасажирів;
- Здійснення контролю за дотриманням правил внутрішніх та міжнародних перевезень, застосуванням тарифів, індикаторів, інструкцій та інших документів, що регламентують організацію перевезення пасажирів та багажу;
- Маркетинг ринку авіаперевезень, робіт і послуг;
- Розробка та подання пропозицій щодо вдосконалення обслуговування авіаперевезень та надання необхідних послуг споживачам авіакомпаній;
- Організація та вдосконалення системи обслуговування пасажирів, надання послуг, у тому числі організація та вдосконалення взаємодії з аеропортами,

агентствами та іншими організаціями щодо забезпечення обслуговування повітряних перевезень;

- Розробка нормативного та методичного забезпечення повітряних перевезень;

- Участь у розробці договорів на наземне обслуговування, аналіз виконання договірних зобов'язань з питань організації наземного обслуговування повітряних перевезень;

- Розробка пропозицій щодо отримання ліцензій та дозволів, у тому числі від інших країн під час міжнародних повітряних перевезень;

- Розробка пропозицій щодо отримання та подальшого здійснення комерційних прав на повітряні лінії;

- Участь у розробці тарифів та зборів за повітряні перевезення, авіаційні роботи та послуги, у складанні розкладу руху повітряного транспорту авіакомпанії;

- Організація професійної підготовки та підвищення рівня кваліфікації персоналу служби сервісу.

- Ведення обліку та звітності щодо обсягів перевезень пасажирів, багажу.

Завдання та функції визначені на основі матеріалу [20].

Організаційна структура авіакомпанії представлена рисунку 1.2 [21].

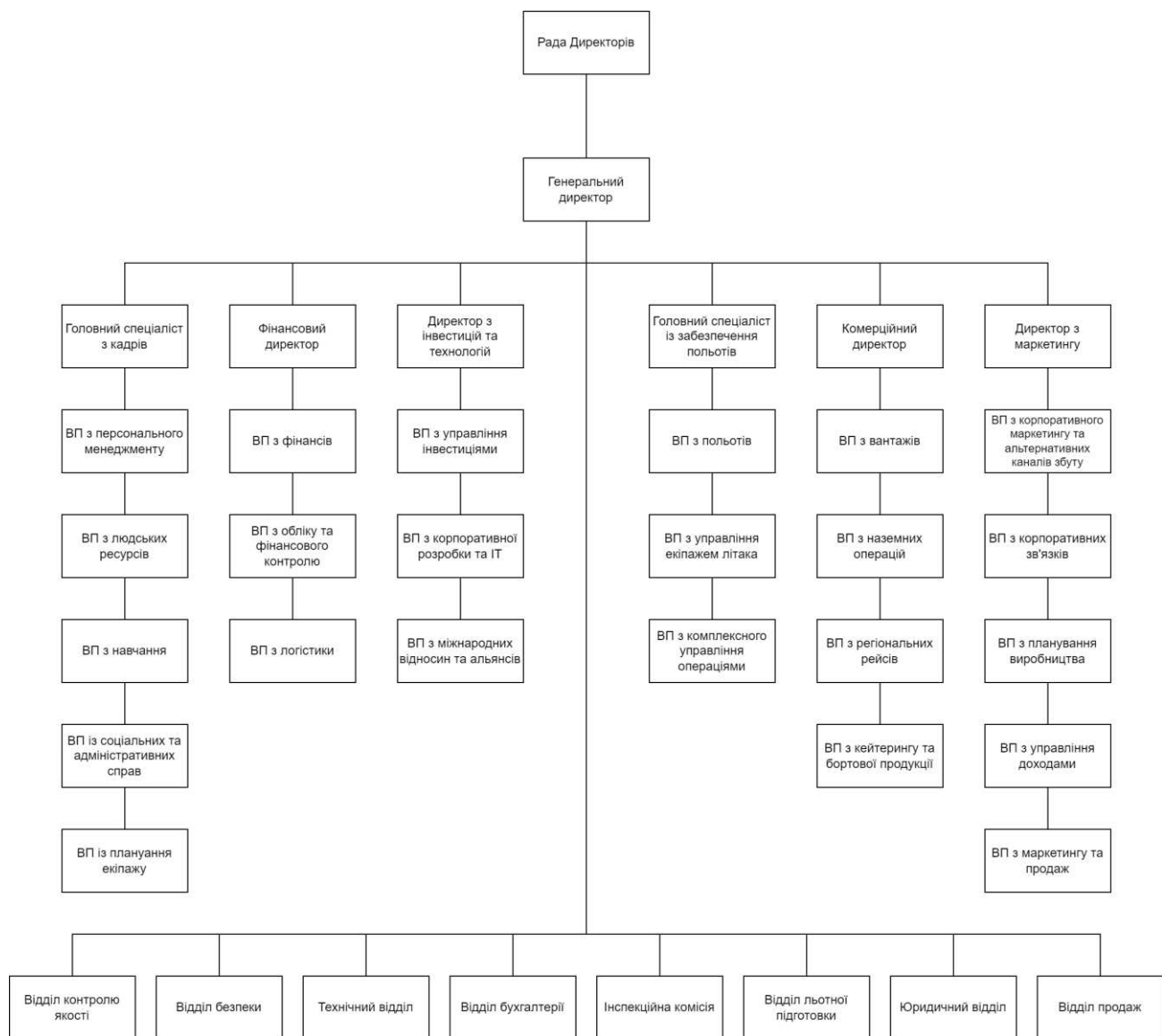


Рисунок 1.2 - Організаційна структура авіакомпанії

На чолі всієї структури стоїть рада директорів. Рада директорів є органом управління, який відповідає за розробку стратегії авіакомпанії, загальне керівництво її діяльністю та контроль діяльності виконавчого органу. Рада директорів здійснює свої функції відповідно до Статуту, Кодексу корпоративного управління та інших внутрішніх документів авіакомпанії. Вони приймають всі основні рішення щодо авіакомпанії, ґрунтуючись на інтересах учасників ради.

На чолі самої авіакомпанії стоїть генеральний директор, який є президентом авіакомпанії. Він є одноосібним виконавчим органом. Стежить за роботою всієї фірми та сприяє точному та своєчасному виконанню всіх рішень ради директорів.

Головний спеціаліст з кадрів спостерігає за всіма аспектами політики, практики та операцій у галузі управління людськими ресурсами та трудовими відносинами авіакомпанії. Йому підпорядковуються віце-президент з персонального менеджменту, віце-президент з людських ресурсів, віце-президент із навчання, віце-президент з соціальних та адміністративних справ, віце-президент із планування екіпажу.

Фінансовий директор відповідає за керування фінансовими діями авіакомпанії. Йому підпорядковуються віце-президент з фінансів, віце-президент з обліку та фінансового контролю, віце-президент з логістики.

Директор з інвестицій та технологій досліджує короткострокові та довгострокові потреби авіакомпанії та використовує капітал для інвестицій, покликаних допомогти авіакомпанії досягти своїх цілей. Йому підпорядковуються віце-президент з управління інвестиціями, віце-президент з корпоративної розробки та ІТ, віце-президент з міжнародних відносин та альянсів.

Головний спеціаліст з забезпечення польотів відповідає за розробку, коригування маршрутів польоту, відповідність стандартам безпеки, безперервне спостереження за польотами. Йому підпорядковуються віце-президент з польотів (головний пілот), віце-президент з управління екіпажем літака, віце-президент з комплексного управління операціями.

Комерційний директор відповідає за комерційну стратегію та розвиток авіакомпанії. Йому підпорядковуються віце-президент з вантажів, віце-президент з наземних операцій, віце-президент з регіональних рейсів, віце-президент з кейтерингу та бортової продукції.

Директор з маркетингу здійснює нагляд за плануванням, розробкою та реалізацією маркетингових та рекламних ініціатив авіакомпанії. Йому

підпорядковуюються віце-президент з корпоративного маркетингу та альтернативних каналів збуту, віце-президент з корпоративних зв'язків, віце-президент з планування виробництва, віце-президент з управління доходами, віце-президент з маркетингу та продаж.

Відділ контролю якості забезпечує моніторинг авіаційного обладнання, програм та процедур для забезпечення дотримання стандартів якості, встановлених міжнародною організацією цивільної авіації та державними органами цивільної авіації, шляхом створення незалежної системи забезпечення якості. Керує відділом віце-президент із контролю якості.

Відділ безпеки відповідає за безпеку польоту та всього, що з цим пов'язано, контролю доступу до літака, перевірки документів, огляду багажу та пасажирів. Керує відділом віце-президент з безпеки.

Технічний відділ займається впровадженням нових розробок у галузі проектування та технологій літаків та двигунів. Керує відділом віце-президент з технологій.

Відділ бухгалтерії веде бухгалтерський та податкового облік фінансово-господарської діяльності авіакомпанії. Керує відділом віце-президент з бухгалтерського обліку та звітності.

Інспекційна комісія повітряних суден забезпечує впевненість, що літаки функціонують належним чином, провівши серію спеціальних перевірок літаків перед кожним польотом. Керує комісією віце-президент інспекційної комісії повітряних суден.

Відділ льотної підготовки забезпечує професійну підготовку всім пілотів авіакомпанії. Керує відділом віце-президент з льотної підготовки.

Юридичний відділ відповідає за спостереження за всіма юридичними справами в авіакомпанії. Вони займаються як внутрішніми, так і зовнішніми юридичними проблемами, і їм доручено робити все, що може, щоб уберегти

авіакомпанію від юридичних проблем. Керує відділом віце-президент з юриспруденції.

Відділ продажів займається пошуком нових клієнтів та роботою з вже існуючою базою з метою продати пропозицію авіакомпанії, в тому числі авіаквитки. Керує відділом віце-президент з продажу.

На основі інформації наведеної вище, авіакомпанія - складний та потужний механізм зі своїми інформаційними системами, але нас цікавить лише частина авіакомпанії, що відповідає за бронювання та реалізацію авіаквитків приватним особам. Для реалізації авіаквитків у авіакомпаній існують власні пункти торгівлі, як у аеропорту у відділі відповідної авіакомпанії, так і за межами аеропортів у філіях. Там же представлено розклад всіх рейсів з точною інформацією про них, такими як дата, час, ціна, модель літака та його код, код самого рейсу, пункти відбуття та призначення, клас тощо. Будь-який бажаючий може прийти і прямо на місці вибрати собі рейс, спираючись на свої бажання, підійти до каси і купити квиток на цей рейс. Клієнт зможе вибрати місце в літаку, клас, так як на кожен рейс продають два види квитків, економ та бізнес класу, після чого потрібно надати інформацію про себе, таку як паспортні дані, номер телефону тощо, після касир занесе потрібну інформацію в базу даних і пов'яже з обраним рейсом і місцем в літаку, клієнт здійснить оплату готівкою або через карту, після чого він зможе забрати квиток.

Однак той факт, що дізнатися інформацію про доступні рейси та купити квиток можна лише "на місці" викликає досить значні незручності для людини, оскільки потребує витрат часу та ресурсів, які хотілося б заощадити. На касира лягає досить значне навантаження на обслуговування великої кількості клієнтів. Так само він виступає в ролі консультанта і надає більш детальну інформацію про рейс, відповідає на питання, що сприяє ускладненню його роботи. З такою організацією реалізації квитків знижується ефективність та швидкість роботи каси, відкидається можливість планування, знижується рівень зручності для клієнта, що

може призвести до зменшення кількості клієнтів та зниження продажів авіакомпанії.

Для вирішення цих проблем можна реалізувати можливість вибору, купівлі та бронювання квитків через веб-додаток, з наступним викупом у касі авіакомпанії, або сформувати та використовувати електронний квиток. Таким чином клієнт зможе в будь-якому зручному для нього, у будь-який зручний для нього час, переглянути потрібну інформацію про рейси, купити або забронювати квиток на відповідний рейс, незалежно від графіка роботи аеропорту та авіакомпанії. Таким чином знизиться навантаження на касу, оскільки частина клієнтів зможе скористатися послугами каси без необхідності бути особисто.

На підставі проведеного аналізу діяльності авіакомпанії необхідно визначити вимоги для інформаційної системи, що розробляється.

1.2 Вимоги до інформаційної системи.

Інформаційна система авіакомпанії онлайн бронювання та реалізації авіаквитків призначена для підвищення ефективності бронювання та реалізації авіаквитків та комфорту клієнтів авіакомпанії, а саме:

- Здатність бронювати та купувати авіаквитки у будь-який час, у будь-якому місці, незалежно від розкладу авіакомпанії чи аеропорту, через web-додаток;
- Зменшення навантаження на касирів, шляхом делегування обов'язку вводу всієї потрібної інформації для бронювання та купівлі авіаквитків клієнту.

Розглянемо вимоги, пред'явлені к розроблювальній системі.

Функціональні вимоги.

Треба розробити web-додаток, який має забезпечувати виконання таких функцій користувачу:

- Бронювання та реалізації авіаквитків;
- Формування електронних авіаквитків;
- Формування електронних чеків;

- Передача інформації про бронювання та покупку авіаквитків авіакомпанії;
- Здача авіаквитків назад строком не пізніше, ніж за 7 днів до дати та часу рейсу;

Треба розробити АРМ, який має забезпечувати виконання таких функцій адміністратору:

- Формування звіту про продані авіаквитки;
- Формування звіту про заброньовані авіаквитки;
- Формування звіту про здані назад авіаквитки;
- Ведення сайту.

Оскільки інформаційна система представлена у вигляді web-додатку, програма повинна бути орієнтована на користувача, що не має спеціальних навичок у роботі з комп'ютерами, смартфонами, планшетними комп'ютерами та смарт-технікою. Користувач повинен мати стабільне підключення до Інтернету та електронну пошту, до якої він має доступ.

Робоче місце адміністратора представлене у вигляді АРМ, де адміністратор зможе отримати звіти, сформовані за потреби.

Вимоги до складу і параметрів технічних та програмних засобів.

Користувач:

Оскільки інформаційна система реалізована у вигляді web-додатку, вона працюватиме по принципу сесій, відправляючи запити по серверу авіакомпанії, які він оброблює, і отримуючи відповіді, які web-додаток показує користувачу у зручному та зрозумілому йому виді.

Для взаємодії з інформаційною системою користувачу потрібен браузер. Тобто до пристроїв, через які можливо взаємодіяти з інформаційною системою, відносяться всі пристрої технічних характеристик яких достатньо для запуску браузера.

На основі інформації наведеної вище сформулюємо технічні характеристики пристрою користувача (сформульовані на основі системних вимог браузеру Google Chrome):

Для Windows:

- Процесор Intel Pentium 4 або пізнішої версії із підтримкою SSE3;
- Об'єм оперативної пам'яті не менше 512 мб;
- Відео пам'ять не менше 64 мб;
- Вільне місце на жорсткому диску не менше 350 мб;
- Розширення екрану не менше 800 × 600;
- Клавіатура;
- Миша.

Для Linux:

- Технічні характеристики аналогічні Windows.

Далі технічні характеристики формуватимуться на основі версії ОС, на базі якої працює пристрій, так як мінімальні вимоги до операційної системи, що підтримується, завжди перевершують технічні характеристики, які необхідні для коректної роботи браузера.

Для MacOS:

- Пристрій працює на базі OS X El Capitan (10.11) і версіях вище.

Для Android:

- Пристрій працює на базі Android 5.0 (Lollipop) і версіях вище.

Для IOS:

- Пристрій працює на базі iOS 12.2 (iPadOS 12.2) і версіях вище.

Адміністратор:

Оскільки частина системи з якою взаємодіє адміністратор представлена у вигляді АРМ, він зможе взаємодіяти з нею тільки через комп'ютер підключений до мережі авіакомпанії.

На основі інформації наведеної вище сформулюємо технічні характеристики пристрою адміністратора:

- Операційна система Windows 7 або пізніша версія;
- Процесор Intel Pentium 4 або пізнішої версії із підтримкою SSE3;
- Об'єм оперативної пам'яті не менше 512 мб;
- Відео пам'ять не менше 64 мб;
- Вільне місце на жорсткому диску не менше 350 мб;
- Розширення екрану не менше 800 × 600;
- Клавіатура;
- Миша.

Авіакомпанія вже має працюючий сервер та базу даних з якими працює інформаційна система.

Інформаційна система повинна бути реалізована у вигляді web-додатку для користувача, та АРМ для адміністратора.

Web-додаток повинен забезпечувати коректність даних, які передаються на сервер та повинен бути реалізований з урахуванням усіх вимог інтернет-безпеки. База даних повинна оновлюватися автоматично після внесення до неї нових даних. Інформаційна система повинна відповідати потребам авіакомпанії.

АРМ повинен забезпечувати відстеження коректності роботи (моніторинг) сайту та формування звітів про рух авіаквитків у системі.

На основі представлених вимог до інформаційної системи розроблюється концепція.

1.3 Розробка концепції інформаційної системи.

Концептуальна модель інформаційної системи представлена на рисунку 1.3.

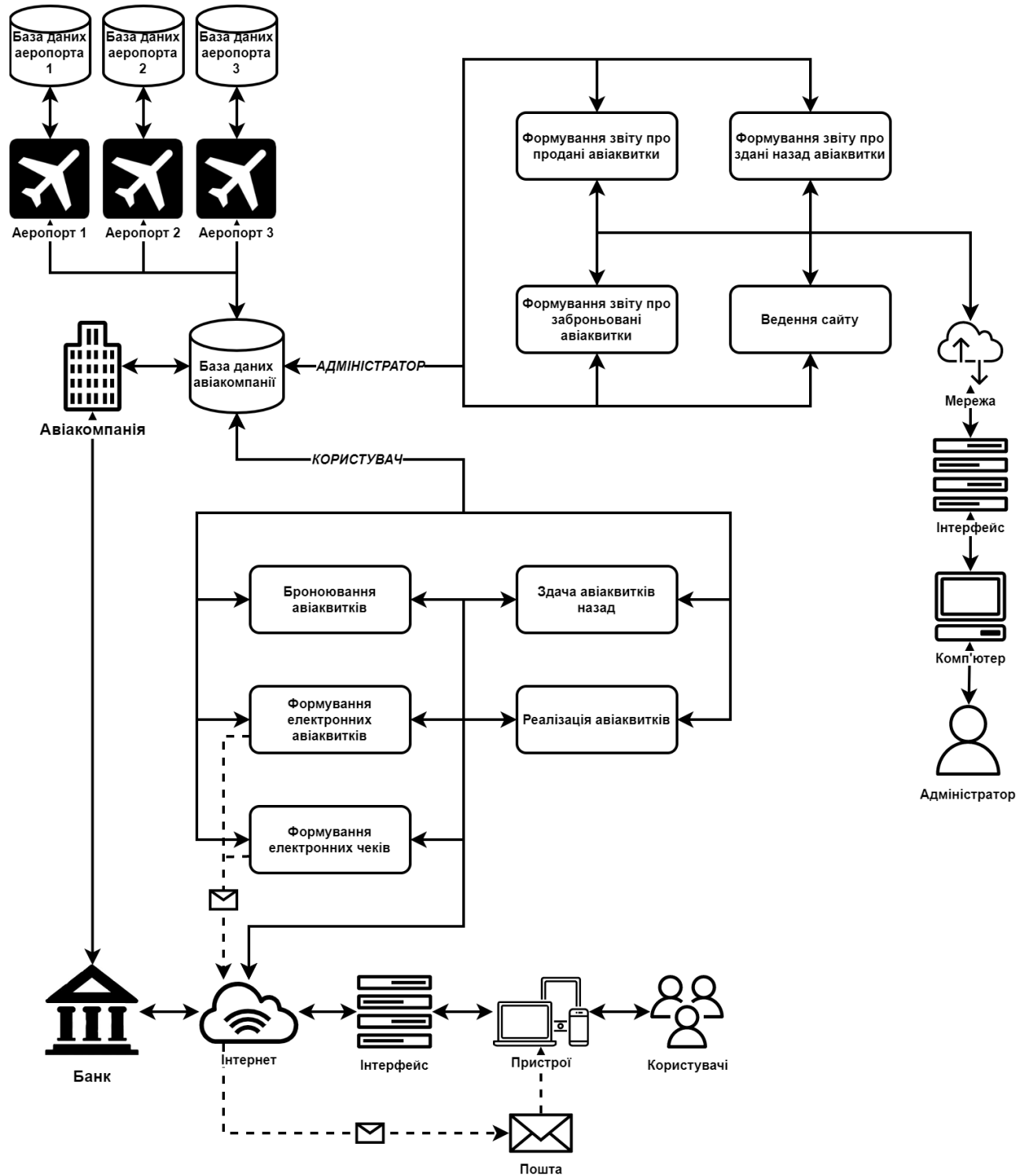


Рисунок 1.3 - Концептуальна модель інформаційної системи

Інформаційна система авіакомпанії онлайн бронювання та реалізації авіаквитків призначена для часткової автоматизації операцій бронювання та реалізації авіаквитків.

Бронювання та купівля авіаквитків здійснюється у web-додатку (сайт). Користувач заходить на сайт та бачить перед собою головне меню, на якому розташований блок с формою для пошуку бажаного рейсу за критеріями, такими як: місто відбуття, місто прибуття, ціна, дата та час, клас та інші. Користувач вводить бажані критерії рейсу, підтверджує критерії і додаток відправляє запит до серверу, а саме до бази даних. Після чого отримує відповідь у вигляді переліку рейсів, що підходять. Вибирає серед них потрібний, натискає на нього та переходить на сторінку з детальним описом рейсу. Після того як користувач переконався в тому, що це точно потрібний рейс, він переходить на сторінку з формою для вводу особистих даних, де він вводить паспортні дані, такі як ПІБ, дата народження, громадянство, реєстрація, також дані для зв'язку, такі як пошта. Також користувач вводить дані про багаж, якщо він є, такі як вага та кількість місць, вибирає місце в літаку. Користувач може додати інших пасажирів та ввести їх дані, дані про їх багаж та вибрати місце в літаку, якщо потрібно купити або забронювати не один авіаквиток. Потом переходить на сторінку остаточного підтвердження вибраного рейсу та введених даних, перевіряє їх та вибирає купівлю чи бронювання авіаквитків.

Якщо вибрана функція купівлі, користувача перекидає на сторінку з формою для вводу платіжної інформації, для оплати квитка. Усі дані про купівлю заносяться у базу даних. Після на вказану пошту приходить чек та електронний авіаквиток (авіаквитки, якщо купівля відбувалася не для однієї особи) у форматі PDF, який користувач зможе роздрукувати чи показати у аеропорту з смартфона чи планшета. На електронному авіаквитку вказані дані рейсу та пасажирів, які перевіряють у аеропорту при посадці на рейс.

Якщо вибрана функція бронювання, на вказану пошту приходить електронний авіаквиток (авіаквитки, якщо купівля відбувалася не для однієї особи) у форматі PDF, який користувач зможе роздрукувати чи показати у аеропорту з смартфона чи планшета. Користувач повинен викупити авіаквиток (авіаквитки) у касі аеропорту чи пункту торгівлі авіакомпанії, не пізніше ніж за день до дати та часу вильоту. Користувач повинен назвати номер заброньованого авіаквитка, після чого будуть перевірені усі дані, які користувач вносив про себе та інших клієнтів, якщо такі є.

Користувач має можливість скасувати своє бронювання або повернути кошти за квиток, терміном не пізніше ніж за 7 днів до дати та часу вильоту літака. Для цього на головній сторінці сайту розташована кнопка, яка перекидає користувача на сторінку з формою. У ній користувач повинен вказати свої ПІБ, дату народження, пошту, яка використовувалася при бронюванні або купівлі авіаквитка, номер електронного квитка та натиснути кнопку, підтверджуючи дані та дію. Якщо авіаквиток був заброньований, то бронювання буде анульоване і на вказану пошту прийде лист з підтвердженням анулювання. Якщо авіаквиток був куплений то після натискання користувач перейде на сторінку з формою для введення реквізитів для повернення коштів за авіаквиток. Після вводу реквізитів користувач повинен натиснути на кнопку та підтвердити дані та дію. Після цього куплений авіаквиток буде анульований, на пошту прийде лист з підтвердженням анулювання та на вказані реквізити будуть зараховані кошти за авіаквиток.

АРМ адміністратора призначене для ведення сайту та формування звітів, які допоможуть оптимізувати роботу інформаційної системи. Адміністратор має можливість запросити звіт про продані авіаквитки, звіт про заброньовані авіаквитки, звіт про здані назад авіаквитки, які будуть автоматично сформовані та представлені йому у вигляді файлу у PDF-форматі. Для ведення сайту АРМ використовується типове рішення у вигляді сервісу для моніторингу роботи сайту “Uptrends”. Сервіс налаштований відповідно до вимог авіакомпанії.

У частині системи пов'язаній з транзакцією коштів використовується типове рішення у вигляді сервісу для обробки електронних платежів “2Checkout”. Сервіс налаштований відповідно до вимог авіакомпанії.

У частині системи пов'язаній з формуванням та відправкою електронних листів використовується типове рішення у вигляді сервісу для розсилання електронних листів “SendPulse”. Сервіс налаштований відповідно до вимог авіакомпанії.

Для реалізації інформаційної системи використані різноманітні інструменти:

- Мова програмування – JavaScript;
- Мова програмування – PHP;
- Мова гіпертекстової розмітки – HTML;
- Мова каскадних таблиць стилів – CSS;
- Система управління базами даних – MySQL;
- Декларативна мова програмування – SQL.

Для створення диплому використані такі інструменти:

- Програмне забезпечення для створення діаграм – draw.io;
- Програмне забезпечення для роботи з текстом – Word;
- Програмне забезпечення для роботи з растровою графікою – Paint.

Технічні засоби використані відповідно до вимог до інформаційної системи.

На основі розробленої концепції інформаційної системи розроблюється технічний проект.

2. Розробка технічного проекту інформаційної системи авіакомпанії онлайн бронювання та реалізації авіаквитків.

2.1 Загальносистемні рішення.

Існує велика кількість сервісів для моніторингу сайту, обробки електронних платежів та розсилання електронних листів, з яких вибрано Uptrends, 2Checkout та SendPulse відповідно. Далі роздивимося більш детально, що собою представляють ці сервіси.

Uptrends - сервіс для моніторингу сайтів, серверів, додатків. Компанія була заснована в 2007 році і з того часу піднялася на перші рядки зарубіжних рейтингів у своїй ніші. Незважаючи на широку спеціалізацію, сервіс успішно працює у кількох напрямках, завдяки команді висококваліфікованих спеціалістів.

Компанія пропонує клієнтам моніторинг, тобто глибокий аналіз та контроль сайтів, веб-додатків, API, серверів та перевірку ефективності мережі. Набір простих інструментів та статистичних даних допоможе встановити якість їх роботи, а значить своєчасно знайти та усунути неполадки. Наявність 184 серверів на п'яти континентах дозволить провести глибокий аналіз аудиторії власного сайту та з'ясувати, де його популярність є найвищою серед користувачів. Оскільки діяльність веб-ресурсу та робота всієї команди безпосередньо залежить від API, за допомогою моніторингу можна виявити приховані проблеми у ньому та підвищити ефективність.

Сервіс пропонує опцію RUM – моніторинг реальних користувачів, тобто перевірку в режимі реального часу. Вона йде бонусом до кожного продукту, клієнту потрібно лише погодитись або відмовитися від її використання. На сайті доступна безкоштовна перевірка швидкості веб-ресурсу, пробний аналіз його роботи, який проводитиметься з кількох точок відразу для підвищення достовірності даних, моніторингу DNS та виявлення своєї ір-адреси.

Будь-який продукт перед покупкою можна випробувати у справі на 30 безкоштовних днів. Користувачі можуть прочитати історії інших клієнтів сервісу

про співпрацю з компанією, відвідавши на сайті спеціальне "Customer Cafe" - майданчик для спілкування.

Сервіс допоможе оптимізувати роботу сайту клієнта та звести до мінімуму тимчасові витрати команди співробітників, виявивши сильні сторони веб-ресурсу, на які потрібно робити наголос. Також моніторинг допоможе синхронізувати роботу сервісу для успішної інтеграції до іноземної аудиторії. За допомогою інструментів контролю роботи сайту можна не тільки підвищити її ефективність, а й вивести ресурс на новий світовий рівень.

Інформація про Uptrends підготовлена на основі матеріалу [22].

2Checkout – це сервіс обробки електронних платежів, призначений для компаній, які продають фізичні товари або цифрові продукти, наприклад програмне забезпечення, онлайн.

2Checkout допомагає бізнесу приймати онлайн-платежі з банківських карток у 200 країнах. Крім обробки платежів, компанія пропонує клієнтам доступ до більш ніж 120 інтеграцій кошиків для зручного оформлення замовлення. Найповніший тарифний план 2Monetize включає й інші функції, такі як управління рахунками, допомога у дотриманні податкового та нормативного законодавства, інструменти для підвищення конверсії та більш ніж 45 способів оплати.

Платформа пропонує дві гнучкі моделі оплати – модель зареєстрованого продавця та модель постачальника платіжних послуг. У першому варіанті 2Checkout бере на себе фінансову відповідальність за обробку транзакцій вашого бізнесу. І тут вона займається управлінням податку з продажу, виставленням рахунків, забезпеченням відповідності тощо.

У свою чергу, вибравши модель постачальника платіжних послуг, ви повинні самостійно відповідати за управління платежами. Якщо ви вирішите використовувати 2Checkout, ви можете проконсультуватися зі службою підтримки компанії, щоб вирішити, який із цих варіантів (або їх суміщення) найкраще підходить для вашого бізнесу.

Крім того, що 2Checkout підтримує платежі в 200 країнах, він доступний більш ніж на 30 мовах і пропонує 100 валют для відображення та виставлення рахунків. Це означає, що покупці можуть робити покупки своєю мовою, оплачувати їх зручними для них способами та бачити ціни у місцевій валюті. Так, американські продавці можуть приймати популярні кредитні картки, Apple Pay, АСН-переклади (АСН & Direct Deposit - вид грошових переказів у США через спеціальну електронну мережу між американськими банками), PayPal та багато іншого.

Сервіс надає два варіанти кошика: розміщений на сервері та вбудований. Розміщена на сервері, як впливає з назви, знаходиться на хостингу 2Checkout і надсилає клієнтів для оплати через власний механізм оформлення замовлення. Можна вибрати один із декількох шаблонів кошика і надалі налаштувати їх під свій бізнес.

Вбудований кошик є вікном оформлення замовлення у форматі iFrame, яке відкривається на вашому сайті і дозволяє клієнтам оформляти замовлення прямо звідти. Обидва кошики оптимізовані для мобільних пристроїв.

Також 2Checkout легко інтегрується з більш ніж 100 провідними сервісами з власними кошиками, такими як BigCommerce, Magento, Shopify та WooCommerce.

Платформа включає інструментальну панель, яка дозволяє переглядати звіти, що охоплюють різні галузі вашого бізнесу, включаючи фінанси, маркетинг, аналітику поведінки клієнтів та підписки. Крім автоматизації роботи зі звітами ви можете проводити рекламні акції та кампанії, пропонувати купони та знижки. З тарифним планом 2Monetize у вас є можливість оптимізувати SEO та відстежувати поведінку покупців на сторінці та у кошику.

Інформація про 2Checkout підготовлена на основі матеріалу [23].

SendPulse - це онлайн-сервіс, що дозволяє створювати найбільш персоналізовані розсилки по e-mail, SMS та Viber. Є опції моніторингу та детального аналізу масових розсилок, багатопотоковий режим, плюс налаштування SMTP, проксі та ротацій текстів.

- Список функцій та особливостей системи SendPulse:
 - Серія листів - Складання автоматичної послідовності із запланованих листів;
 - Спліт-тест - маркетингове дослідження для зростання ефективності розсилок з порівнянням листів;
 - Статистика — звіти про розсилки для визначення ефективних деталей;
 - White Label – згадка у розсилках виключно бренду вашої компанії;
 - Персоналізація – вставка даних щодо конкретного клієнта;
 - Шаблони - готові листи та розсилки для простоти роботи;
 - Респонсивність - адаптація під пристрої з різними роздільними здатністю екранів;
 - HTML редактор для редагування листів;
 - Планувальник — Зміна дати та часу розсилки для більш точних дій;
 - Безкоштовні повідомлення в пробний період;
 - Додавання необмеженої кількості адрес до розсилок;
 - Графіки та схеми з виконаних розсилок;
 - Карта кліків — з'ясувати найбільш корисні посилання та кількість переходів;
 - Експорт даних;
 - Форми підписки - генерація своїх форм, зміни стилю, форм, редагування та внесення полів;
 - Посилання для відписки;
 - Сторінки відписки для заповнення клієнтом причин відмови від розсилки;
 - Блокування адрес для конкретних розсилок;
 - Списки розсилки для ефективної кампанії.
- Інформація про SendPulse підготовлена на основі матеріалу [24].

2.2 Рішення по інформаційному забезпеченню.

Для забезпечення роботи інформаційної системи потрібна база даних. Вона вже спроектована та використовується у інших системах авіакомпанії. Для управління базою даних використовується СУБД MySQL.

MySQL — вільна система керування реляційними базами даних, яка була розроблена компанією «ТсХ» для підвищення швидкодії обробки великих баз даних. Ця система керування базами даних (СКБД) з відкритим кодом була створена як альтернатива комерційним системам. MySQL з самого початку була дуже схожою на mSQL, проте з часом вона все розширювалася і зараз MySQL — одна з найпоширеніших систем керування базами даних. Вона використовується, в першу чергу, для створення динамічних вебсторінок, оскільки має чудову підтримку з боку різноманітних мов програмування.

Інформація про MySQL підготовлена на основі матеріалу [25].

Нижче представлені концептуальна, логічна та фізична модель бази даних. Створені моделі за допомогою програмного забезпечення draw.io

2.2.1 Розробка концептуальної моделі даних.

Концептуальна модель бази даних це наочна діаграма, намальована у прийнятих позначеннях, яка показує зв'язок між об'єктами та його характеристиками. Створюється концептуальна модель для подальшого проектування бази даних та переведення її, наприклад, в реляційну базу даних.

Концептуальна модель бази даних представлена на рисунку 2.1.

У авіакомпанії вже є готова база даних у якій наявний функціонал лише для реєстрування проданих авіаквитків. Для реалізації додаткового функціоналу, сформованого відповідно до вимог інформаційної системи, а саме реєстрування заброньованих та анульованих авіаквитків, база даних авіакомпанії модифікована для відповідності вимогам та представлена на концептуальній, логічній та фізичній моделях. Модифікація проведена технічними спеціалістами авіакомпанії.

2.2.2 Розробка логічної моделі даних.

Логічна (дatalogічна) модель є модель бази даних, яка не прив'язана до конкретної СУБД. У ній виділяють основні об'єкти БД та визначають зв'язки між цими об'єктами. Іноді визначаються типи даних окремих об'єктів. Ця модель побудована шляхом Сутність-зв'язок (Entity Relationship).

Логічна модель бази даних представлена на рисунку 2.2.

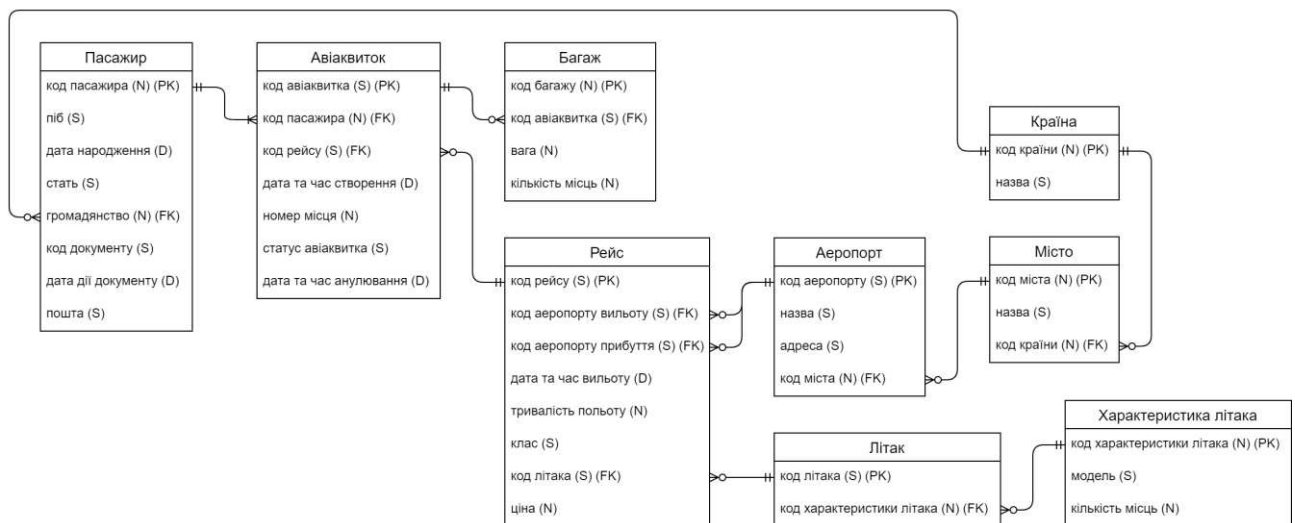


Рисунок 2.2 - Логічна модель бази даних

2.2.3 Розробка фізичної моделі даних.

Фізична модель бази даних - це модель даних, яка визначає, яким чином надаються дані, і містить всі деталі, необхідні СУБД для створення бази даних.

Фізична модель бази даних представлена на рисунку 2.3.

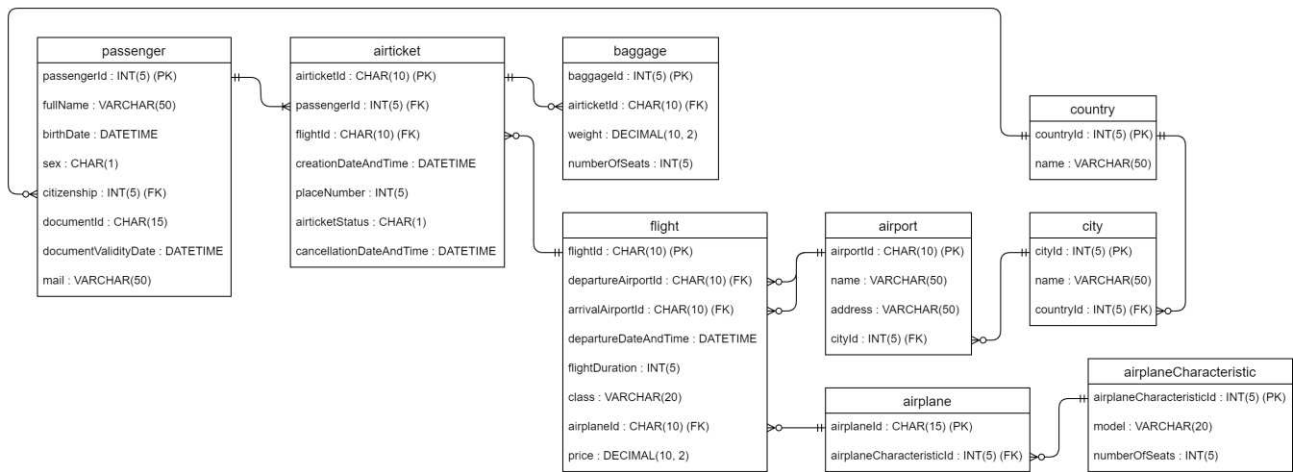


Рисунок 2.3 - Фізична модель бази даних

Для заповнення частини бази даних, що пов’язана з пасажиром (таблиці “passenger” та “baggage”) використовується інформація надана користувачем та дані з документу, що засвідчує особу.

Для заповнення частини бази даних, що пов’язана з рейсами (таблиця “flight”) використовується розклад польотів, який вже сформований авіакомпанією і знаходиться у базі даних авіакомпанії.

Для заповнення частини бази даних, що пов’язана з аеропортами (таблиця “airport”) використовується інформація у відкритому доступі про аеропорти.

Для заповнення частини бази даних, що пов’язана з містами та країнами (таблиці “city” та “country”) використовується інформація у відкритому доступі про міста та країни.

Для заповнення частини бази даних, що пов’язана з літаками (таблиці “airplane” та “airplaneCharacteristic”) використовується технічний паспорт літака.

Для заповнення частини бази даних, що пов’язана з купленими, заброньованими або анульованими авіаквитками (таблиця “airticket”) використовується вся, вище представлена, інформація.

На основі представлених вище рішень з інформаційного забезпечення розроблені рішення з математичного забезпечення інформаційної системи.

2.3 Рішення з математичного забезпечення.

Блок-схема загального алгоритму роботи частини користувача системи представлена на рисунку 2.4.

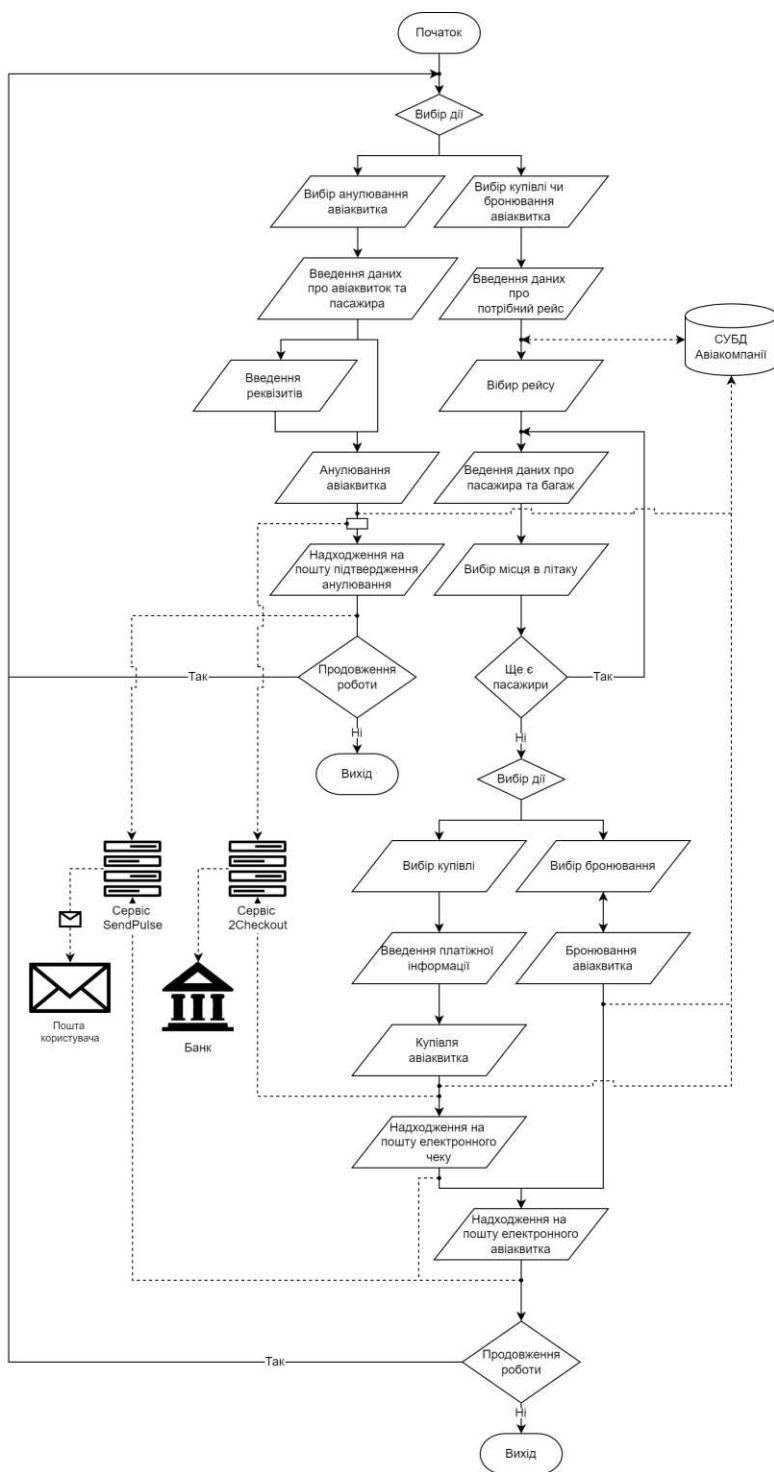


Рисунок 2.4 – Блок-схема загального алгоритму роботи частини користувача системи

Бронювання та купівля авіаквитків здійснюється у web-додатку (сайт). Користувач заходить на сайт та бачить перед собою головне меню, на якому розташований блок с формою для пошуку бажаного рейсу за критеріями, такими як: місто відбуття, місто прибуття, ціна, дата та час, клас та інші. Користувач вводить бажані критерії рейсу, підтверджує критерії і додаток відправляє запит до серверу, а саме до бази даних. Після чого отримує відповідь у вигляді переліку рейсів, що підходять. Вибирає серед них потрібний, натискає на нього та переходить на сторінку з детальним описом рейсу. Після того як користувач переконався в тому, що це точно потрібний рейс, він переходить на сторінку з формою для вводу особистих даних, де він вводить паспортні дані, такі як ПІБ, дата народження, громадянство, реєстрація, також дані для зв'язку, такі як пошта. Також користувач вводить дані про багаж, якщо він є, такі як вага та кількість місць, вибирає місце в літаку. Користувач може додати інших пасажирів та ввести їх дані, дані про їх багаж та вибрати місце в літаку, якщо потрібно купити або забронювати не один авіаквиток. Потом переходить на сторінку остаточного підтвердження вибраного рейсу та введених даних, перевіряє їх та вибирає купівлю чи бронювання авіаквитків.

Якщо вибрана функція купівлі, користувача перекидає на сторінку з формою для вводу платіжної інформації, для оплати квитка. Усі дані про купівлю заносяться у базу даних. Після на вказану пошту приходить чек та електронний авіаквиток (авіаквитки, якщо купівля відбувалася не для однієї особи) у форматі PDF, який користувач зможе роздрукувати чи показати у аеропорту з смартфона чи планшета. На електронному авіаквитку вказані дані рейсу та пасажирів, які перевіряють у аеропорту при посадці на рейс.

Якщо вибрана функція бронювання, на вказану пошту приходить електронний авіаквиток (авіаквитки, якщо купівля відбувалася не для однієї особи) у форматі PDF, який користувач зможе роздрукувати чи показати у аеропорту з смартфона чи планшета. Користувач повинен викупити авіаквиток (авіаквитки) у

касі аеропорту чи пункту торгівлі авіакомпанії, не пізніше ніж за день до дати та часу вильоту. Користувач повинен назвати номер заброньованого авіаквитка, після чого будуть перевірені усі дані, які користувач вносив про себе та інших клієнтів, якщо такі є.

Користувач має можливість скасувати своє бронювання або повернути кошти за квиток, терміном не пізніше ніж за 7 днів до дати та часу вильоту літака. Для цього на головній сторінці сайту розташована кнопка, яка перекидає користувача на сторінку з формою. У ній користувач повинен вказати свої ПІБ, дату народження, пошту, яка використовувалася при бронюванні або купівлі авіаквитка, номер електронного квитка та натиснути кнопку, підтверджуючи дані та дію. Якщо авіаквиток був заброньований, то бронювання буде анульоване і на вказану пошту прийде лист з підтвердженням анулювання. Якщо авіаквиток був куплений то після натискання користувач перейде на сторінку з формою для введення реквізитів для повернення коштів за авіаквиток. Після вводу реквізитів користувач повинен натиснути на кнопку та підтвердити дані та дію. Після цього куплений авіаквиток буде анульований, на пошту прийде лист з підтвердженням анулювання та на вказані реквізити будуть зараховані кошти за авіаквиток.

У частині системи пов'язаній з транзакцією коштів використовується типове рішення у вигляді сервісу для обробки електронних платежів “2Checkout”. Сервіс налаштований відповідно до вимог авіакомпанії.

У частині системи пов'язаній з формуванням та відправкою електронних листів використовується типове рішення у вигляді сервісу для розсилання електронних листів “SendPulse”. Сервіс налаштований відповідно до вимог авіакомпанії.

Блок-схема загального алгоритму роботи частини адміністратора системи представлена на рисунку 2.5.

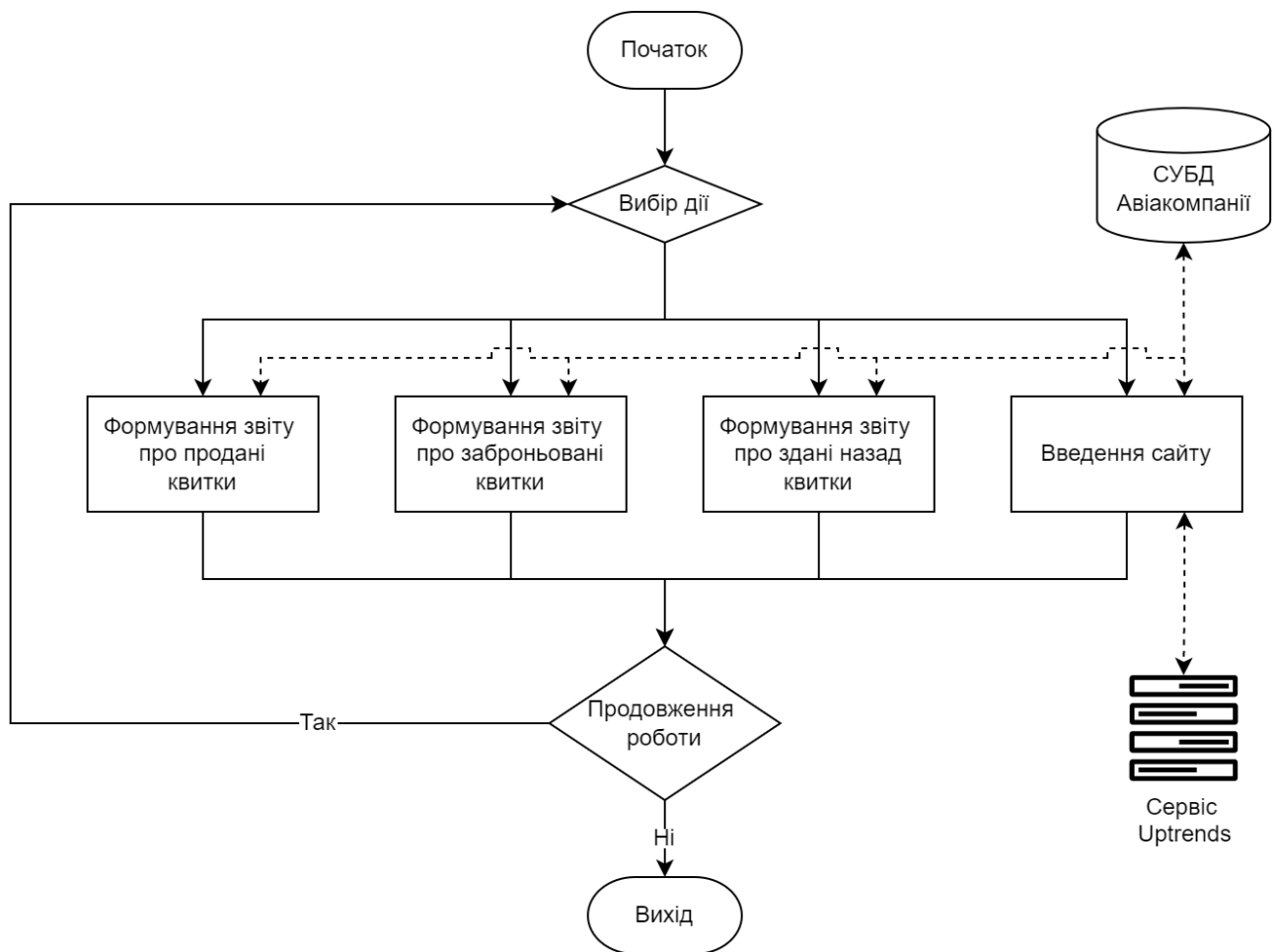


Рисунок 2.5 – Блок-схема загального алгоритму роботи частини адміністратора системи

АРМ адміністратора призначене для ведення сайту та формування звітів, які допоможуть оптимізувати роботу інформаційної системи. Адміністратор має можливість запросити звіт про продані авіаквитки, звіт про заброньовані авіаквитки, звіт про здані назад авіаквитки, які будуть автоматично сформовані та представлені йому у вигляді файлу у PDF-форматі. Для ведення сайту АРМ використовується типове рішення у вигляді сервісу для моніторингу роботи сайту “Uptrends”. Сервіс налаштований відповідно до вимог авіакомпанії.

На основі представлених вище рішень з математичного забезпечення розроблені рішення з програмного забезпечення інформаційної системи, які

дозволить реалізувати функції, що відповідають вимогам до інформаційної системи.

2.4 Рішення з програмного забезпечення.

Програмне забезпечення розроблювалося на основі рішень з математичного забезпечення.

Частина системи, що пов'язана з інформаційним забезпеченням, тобто база даних, розроблена на основі СУБД MySQL та з використанням декларативної мови програмування SQL.

MySQL — вільна система керування реляційними базами даних, яка була розроблена компанією «ТсХ» для підвищення швидкодії обробки великих баз даних. Ця система керування базами даних (СКБД) з відкритим кодом була створена як альтернатива комерційним системам. MySQL з самого початку була дуже схожою на mSQL, проте з часом вона все розширювалася і зараз MySQL — одна з найпоширеніших систем керування базами даних. Вона використовується, в першу чергу, для створення динамічних вебсторінок, оскільки має чудову підтримку з боку різноманітних мов програмування.

MySQL має подвійне ліцензування. MySQL може розповсюджуватися відповідно до умов ліцензії GPL. Але за умовами GPL, якщо якась програма використовує бібліотеки MySQL, то вона теж повинна розповсюджуватися за ліцензією GPL. Проте це може розходитися з планами розробників, які не бажають відкривати сирцеві тексти своїх програм. Для таких випадків передбачена комерційна ліцензія компанії Oracle, яка також забезпечує якісну сервісну підтримку. В разі використання та розповсюдження програмного забезпечення з іншими вільними ліцензіями, такими як BSD, Apache, MIT та інші, MySQL дозволяє використання бібліотек MySQL за ліцензією GPL.

MySQL виникла як спроба застосувати mSQL до власних розробок компанії: таблиць, для яких використовувалися ISAM — підпрограми низького рівня для

індексного доступу до даних. У результаті був вироблений новий SQL-інтерфейс, але API-інтерфейс залишився в спадок від mSQL.

MySQL — компактний багатопотоковий сервер баз даних. Характеризується високою швидкістю, стійкістю і простотою використання.

MySQL вважається гарним рішенням для малих і середніх застосувань. Сирцеві коди сервера компілюються на багатьох платформах. Найповніше можливості сервера виявляються в UNIX-системах, де є підтримка багатопоточності, що підвищує продуктивність системи в цілому.

Можливості сервера MySQL:

- Простота у встановленні та використанні;
- Підтримується необмежена кількість користувачів, що одночасно працюють із БД;
- Кількість рядків у таблицях може досягати 50 млн;
- Висока швидкість виконання команд;
- Наявність простої і ефективної системи безпеки.

Інформація про MySQL підготовлена на основі матеріалу [25].

SQL (/ˌɛsˌkjuːˈɛl/ або /ˈsiːkwəl/, англ. Structured query language — мова структурованих запитів) — декларативна мова програмування для взаємодії користувача з базами даних, що застосовується для формування запитів, оновлення і керування реляційними БД, створення схеми бази даних та її модифікації, системи контролю за доступом до бази даних. Сама по собі SQL не є ані системою керування базами даних, ані окремим програмним продуктом. На відміну від дійсних мов програмування (C або Pascal), SQL може формувати інтерактивні запити або, будучи вбудованою в прикладні програми, виступати як інструкції для керування даними. Окрім цього, стандарт SQL містить функції для визначення зміни, перевірки та захисту даних.

SQL — це діалогова мова програмування для здійснення запиту і внесення змін до бази даних, а також керування базами даних. Багато баз даних підтримує

SQL з розширеннями до стандартної мови. Ядро SQL формує командна мова, яка дозволяє здійснювати пошук, вставку, оновлення і вилучення даних за допомогою використання системи керування і адміністративних функцій. SQL також включає CLI (Call Level Interface) для доступу і керування базами даних дистанційно.

Основу бази даних SQL Server утворює сервер або ядро бази даних. Ядро бази даних відповідає за обробку запитів, які надходять від клієнтів, і передачу відповідних результатів клієнтським компонентам.

Перша версія SQL була розроблена на початку 1970-х років у IBM. Ця версія мала назву SEQUEL і була призначена для обробки та пошуку даних, що містилися в реляційній базі даних IBM, System R. Мова SQL надалі була стандартизована Американськими Держстандартами (ANSI) в 1986. На початку SQL була запланована як мова запитів і управління даними, а пізніші модифікації SQL створені продавцями системи управління базами даних, які додали процедурні конструкції, control-of-flow команд і темпоральні розширення мов. З випуском стандарту SQL:1999 такі розширення були формально запозичені як частина мови SQL через Persistent Stored Modules (SQL/PSM).

Інформація SQL підготовлена на основі матеріалу [26].

Розроблено програмне забезпечення для бази даних, з використанням, приведених вище, інструментів, проведено налагодження з формування та ведення бази даних.

Web-додаток розроблений з використанням мови програмування JavaScript, мови гіпертекстової розмітки HTML та мови каскадних таблиць стилів CSS.

JavaScript (/ˈdʒɑːvɑːˌskript/; аббр. JS /ˈdʒeɪ.es/) — мультипарадигменна мова програмування. Підтримує об'єктно-орієнтований, імперативний та функціональний стилі. Є реалізацією специфікації ECMAScript (стандарт ECMA-262).

JavaScript зазвичай використовується як вбудована мова для програмного доступу до об'єктів програм. Найширше застосування знаходить у браузерях як мову сценаріїв надання інтерактивності веб-сторінкам.

Основні архітектурні риси: динамічна типізація, слабка типізація, автоматичне керування пам'яттю, прототипне програмування, функції як об'єкти першого класу.

На JavaScript вплинули багато мов, при розробці була мета зробити мову схожою на Java. Мовою JavaScript не володіє будь-яка компанія або організація, що відрізняє його від низки мов програмування, що використовуються у веб-розробці.

Назва «JavaScript» є зареєстрованим товарним знаком корпорації Oracle США.

JavaScript є об'єктно-орієнтованою мовою, але використовуване в мові прототипування обумовлює відмінності у роботі з об'єктами порівняно з традиційними клас-орієнтованими мовами. Крім того, JavaScript має ряд властивостей, властивих функціональним мовам – функції як об'єкти першого класу, об'єкти як списки, каринг, анонімні функції, замикання – що надає мові додаткової гнучкості.

Незважаючи на схожий із Сі синтаксис, JavaScript у порівнянні з мовою Сі має корінні відмінності:

- Об'єкти з можливістю інтроспекції;
- Функції як об'єкти першого класу;
- Автоматичне приведення типів;
- Автоматичне складання сміття;
- Анонімні функції.

У мові відсутні такі корисні речі, як:

- Стандартна бібліотека: зокрема, відсутній інтерфейс програмування програм по роботі з файловою системою, управління потоками введення-виводу, базових типів для бінарних даних;
- Стандартні інтерфейси до веб-серверів та баз даних;

- Система управління пакетами, яка відстежувала б залежності і автоматично встановлювала їх.

Інформація про JavaScript підготовлена на основі матеріалу [27].

HTML (від англ. HyperText Markup Language - "мова гіпертекстової розмітки") - стандартизована мова розмітки документів для перегляду веб-сторінок у браузері. Веб-браузери отримують HTML документ від сервера за протоколами HTTP/HTTPS або відкривають з локального диска, далі інтерпретують код в інтерфейс, який відображатиметься на екрані монітора.

Елементи HTML є будівельними блоками сторінок HTML. За допомогою HTML різні конструкції, зображення та інші об'єкти, такі як інтерактивна веб-форма, можуть бути вбудовані в сторінку, що відображається. HTML надає засоби для створення заголовків, абзаців, списків, посилань, цитат та інших елементів. Елементи HTML виділяються тегами, записаними з використанням кутових дужок. Такі теги, як `<img/>` та `<input/>`, безпосередньо вводять контент на сторінку. Інші теги, такі як `<p>`, оточують і оформляють текст у собі і можуть включати інші теги як піделементів. Браузери не відображають HTML-теги, але використовують їх для інтерпретації вмісту сторінки.

Мова XHTML є суворішим варіантом HTML, він слідує синтаксису XML і є додатком мови XML в області розмітки гіпертексту.

У HTML можна вбудувати програмний код мовою програмування JavaScript, керувати поведінкою та змістом веб-сторінок. Також включення CSS в HTML описує зовнішній вигляд та макет сторінки.

Мова гіпертекстової розмітки HTML був розроблений британським ученим Тімом Бернерсом-Лі приблизно в 1986-1991 роках у стінах ЦЕРНу в Женеві у Швейцарії. HTML створювався як мова для обміну науковою та технічною документацією, придатний для використання людьми, які не є фахівцями в галузі верстки. HTML успішно справлявся із проблемою складності SGML шляхом визначення невеликого набору структурних та семантичних елементів —

дескрипторів. Дескриптори також часто називають "тегами". За допомогою HTML можна легко створити відносно простий, але красиво оформлений документ. Крім спрощення структури документа, HTML внесена підтримка гіпертексту. Мультимедійні можливості було додано пізніше.

Першим загальнодоступним описом HTML був документ "Теги HTML", вперше згаданий в Інтернеті Тімом Бернерсом-Лі наприкінці 1991. У ньому описуються 18 елементів, що становлять початковий, відносно простий дизайн HTML. За винятком тега гіперпосилання, на них сильно вплинув SGMLguid, внутрішній формат документації, заснований на стандартній узагальненій мові розмітки (SGML), у CERN. Одинадцять із цих елементів усе ще існують у HTML 4.

Спочатку мова HTML була задумана і створена як засіб структурування та форматування документів без їх прив'язки до засобів відтворення (відображення). В ідеалі, текст з розміткою HTML повинен був без стилістичних та структурних спотворень відтворюватися на обладнанні з різним технічним оснащенням (кольоровий екран сучасного комп'ютера, монохромний екран органайзера, обмежений за розмірами екран мобільного телефону або пристрою та програми голосового відтворення текстів). Однак сучасне застосування HTML дуже далеке від його початкового завдання. Наприклад, тег <table> призначений для створення в документах таблиць, але іноді використовується для оформлення розміщення елементів на сторінці. З часом основна ідея платформонезалежності мови HTML була принесена в жертву сучасним потребам у мультимедійному та графічному оформленні.

Інформація про HTML підготовлена на основі матеріалу [28].

CSS (/si:eses/ англ. Cascading Style Sheets "каскадні таблиці стилів") - формальна мова опису зовнішнього вигляду документа (веб-сторінки), написаного з використанням мови розмітки (найчастіше HTML або XHTML). Також може застосовуватися до будь-яких XML-документів, наприклад SVG або XUL.

CSS використовується творцями веб-сторінок для завдання кольорів, шрифтів, стилів, розташування окремих блоків та інших аспектів представлення цих веб-сторінок. Основною метою розробки CSS є огороження та відокремлення опису логічної структури веб-сторінки (яке виробляється за допомогою HTML або інших мов розмітки) від опису зовнішнього вигляду цієї веб-сторінки (яке тепер виробляється за допомогою формальної мови CSS). Такий поділ може збільшити доступність документа, надати більшу гнучкість та можливість управління його поданням, а також зменшити складність та повторюваність у структурному вмісті.

Крім того, CSS дозволяє представляти один і той же документ у різних стилях або методах виведення, таких як екранне представлення, друковане представлення, читання голосом (спеціальним голосовим браузером або програмою читання з екрана) або при виведенні пристроями, що використовують шрифт Брайля.

Інформація про CSS підготовлена на основі матеріалу [29].

Розроблений web-додаток, з використанням, приведених вище, інструментів, проведено налагодження щодо взаємодії з сервером авіакомпанії.

Частина системи, що пов'язана з сервером, розроблена з використанням мови програмування PHP.

PHP (/pi:.eitʃ.pi:/ англ. PHP: Hypertext Preprocessor — «PHP: препроцесор гіпертексту»; спочатку PHP/FI (Personal Home Page / Form Interpreter), а пізніше названий Personal Home Page Tools — «Інструменти для створення персональних веб-сторінок») — С-подібна скриптова мова загального призначення, що інтенсивно використовується для розробки веб-додатків. В даний час підтримується переважна більшість хостинг-провайдерів і є одним з лідерів серед мов, що застосовуються для створення динамічних веб-сайтів.

Мова та її інтерпретатор (Zend Engine) розробляються групою ентузіастів у межах проекту з відкритим кодом. Проект розповсюджується під власною ліцензією, несумісною з GNU GPL.

В області веб-програмування, зокрема серверної частини, PHP — одна з найпопулярніших сценарних мов (поряд із JSP, Perl та мовами, що використовуються в ASP.NET).

Популярність у сфері побудови веб-сайтів визначається наявністю великого набору вбудованих засобів та додаткових модулів для розробки веб-програм. Основні з них:

- Автоматичне вилучення POST- та GET-параметрів, а також змінних оточення веб-сервера в певні масиви;
- Взаємодія з великою кількістю різних систем управління базами даних через додаткові модулі (MySQL, MySQLi, SQLite, PostgreSQL, Oracle Database (OCI8), Microsoft SQL Server, Sybase, ODBC, mSQL, IBM DB2, Cloudscape та Apache Derby, Informix, Ovrinos SQL, Lotus Notes, DB++, DBM, dBase, DBX, FrontBase, FilePro, Ingres II, SESAM, Firebird та InterBase, Paradox File Access, MaxDB, інтерфейс PDO, Redis);
- Автоматизоване відправлення HTTP-заголовків;
- Робота з HTTP-авторизацією;
- Робота з cookies та сесіями;
- Робота з локальними та віддаленими файлами, сокетом;
- Обробка файлів, що завантажуються на сервер;
- Робота з XForms;
- Створення та робота з API;
- Створення програму з графічним інтерфейсом користувача при використанні Фреймворку Qt Designer;
- Створення консольних програм.

Станом на 2010 роки використовується сотнями тисяч розробників; згідно з рейтингом корпорації ТЮВЕ, що базується на даних пошукових систем, у травні 2016 року PHP знаходився на 6 місці серед мов програмування.

Входить до LAMP - поширений набір ПЗ для створення та хостингу веб-сайтів (Linux, Apache, MySQL, PHP), а також LEMP - не менш поширений набір ПЗ (Linux, Nginx, MySql, PHP).

Серед сайтів, які використовують PHP - Facebook, Wikipedia, Yahoo!, Baidu.

Інформація про PHP підготовлена на основі матеріалу [30].

Розроблена серверна частина, з використанням, приведених вище, інструментів, проведено налагодження щодо взаємодії з web-додатком.

Розроблене програмне забезпечення для моніторингу сайту на основі сервісу Uptrends.

Uptrends - сервіс для моніторингу сайтів, серверів, додатків. Компанія була заснована в 2007 році і з того часу піднялася на перші рядки зарубіжних рейтингів у своїй ніші. Незважаючи на широку спеціалізацію, сервіс успішно працює у кількох напрямках, завдяки команді висококваліфікованих спеціалістів.

Компанія пропонує клієнтам моніторинг, тобто глибокий аналіз та контроль сайтів, веб-додатків, API, серверів та перевірку ефективності мережі. Набір простих інструментів та статистичних даних допоможе встановити якість їх роботи, а значить своєчасно знайти та усунути неполадки. Наявність 184 серверів на п'яти континентах дозволить провести глибокий аналіз аудиторії власного сайту та з'ясувати, де його популярність є найвищою серед користувачів. Оскільки діяльність веб-ресурсу та робота всієї команди безпосередньо залежить від API, за допомогою моніторингу можна виявити приховані проблеми у ньому та підвищити ефективність.

Інформація про Uptrends підготовлена на основі матеріалу [22].

Розроблене програмне забезпечення для обробки електронних платежів на основі сервісу 2Checkout.

2Checkout – це сервіс обробки електронних платежів, призначений для компаній, які продають фізичні товари або цифрові продукти, наприклад програмне забезпечення, онлайн.

2Checkout допомагає бізнесу приймати онлайн-платежі з банківських карток у 200 країнах. Крім обробки платежів, компанія пропонує клієнтам доступ до більш ніж 120 інтеграцій кошиків для зручного оформлення замовлення. Найповніший тарифний план 2Monetize включає й інші функції, такі як управління рахунками, допомога у дотриманні податкового та нормативного законодавства, інструменти для підвищення конверсії та більш ніж 45 способів оплати.

Інформація про 2Checkout підготовлена на основі матеріалу [23].

Розроблене програмне забезпечення для розсилання електронних листів на основі сервісу SendPulse.

SendPulse - це онлайн-сервіс, що дозволяє створювати найбільш персоналізовані розсилки по e-mail, SMS та Viber. Є опції моніторингу та детального аналізу масових розсилок, багатопотоковий режим, плюс налаштування SMTP, проксі та ротацій текстів.

Інформація про SendPulse підготовлена на основі матеріалу [24].

Для захисту передачі даних використано протокол HTTPS.

HTTPS (аббр. від англ. HyperText Transfer Protocol Secure) — розширення протоколу HTTP для підтримки шифрування з метою підвищення безпеки. Дані у протоколі HTTPS передаються поверх криптографічних протоколів TLS або застарілого у 2015 році SSL. На відміну від HTTP із TCP-портом 80, для HTTPS за замовчуванням використовується TCP-порт 443.

Протокол був розроблений компанією Netscape Communications для браузера Netscape Navigator у 1994 році.

HTTPS не є окремим протоколом. Це стандартний HTTP, що працює через шифровані транспортні механізми SSL і TLS. Він забезпечує захист від атак, заснованих на прослуховуванні мережного з'єднання — від сніфферських атак та атак типу man-in-the-middle, за умови, що будуть використовуватись шифруючі засоби та сертифікат сервера перевірений і йому довіряють.

За умовчанням HTTPS URL використовує 443 TCP-порт (для незахищеного HTTP - 80). Щоб підготувати веб-сервер для обробки https-з'єднань, адміністратор повинен отримати та встановити сертифікат відкритого та закритого ключа для цього веб-сервера. У TLS використовується як асиметрична схема шифрування (для вироблення загального секретного ключа), так і симетрична (для обміну даними, зашифрованими загальним ключем). Сертифікат відкритого ключа підтверджує належність цього відкритого ключа власнику сайту. Сертифікат відкритого ключа та сам відкритий ключ надсилаються клієнту під час встановлення з'єднання; закритий ключ використовується для розшифровки повідомлень від клієнта.

Існує можливість створити такий сертифікат, не звертаючись до центру сертифікації. Підписуються такі сертифікати тим самим сертифікатом і називаються самопідписаними (self-signed). Без перевірки сертифіката якимось іншим способом (наприклад, дзвінок власнику та перевірка контрольної суми сертифіката) таке використання HTTPS піддається атаці посередника.

Ця система також може бути використана для автентифікації клієнта, щоб забезпечити доступ до сервера лише авторизованим користувачам. Для цього адміністратор зазвичай створює сертифікати для кожного користувача та завантажує їх у браузер кожного користувача. Також прийматимуться всі сертифікати, підписані організаціями, яким довіряє сервер. Такий сертифікат зазвичай містить ім'я та адресу електронної пошти авторизованого користувача, які перевіряються при кожному з'єднанні, щоб перевірити особу користувача без введення пароля.

HTTPS для шифрування використовується довжина ключа 40, 56, 128 або 256 біт. Деякі старі версії браузерів використовують довжину ключа 40 біт (наприклад, IE версій до 4.0), що пов'язано з експортними обмеженнями в США. Довжина ключа 40 біт не є надійною. Багато сучасних сайтів вимагають використання нових версій браузерів, що підтримують шифрування з довжиною ключа 128 біт, щоб

забезпечити достатній рівень безпеки. Шифрування з довжиною ключа 128 біт значно ускладнює підбір паролів та доступ до особистої інформації.

Традиційно на одній IP-адресі може працювати лише один HTTPS-сайт. Для роботи кількох HTTPS-сайтів із різними сертифікатами застосовується розширення TLS під назвою Server Name Indication (SNI).

Інформація про HTTPS підготовлена на основі матеріалу [31].

На основі представлених вище рішень з програмного забезпечення розробляються рішення з технічного забезпечення інформаційної системи, яке дозволить реалізувати функції, що відповідають вимогам до інформаційної системи.

2.5 Рішення з технічного забезпечення.

Для взаємодії з інформаційною системою користувачу потрібен браузер. Тобто до пристроїв, через які можливо взаємодіяти з інформаційною системою, відносяться всі пристрої технічних характеристик яких достатньо для запуску браузера.

На основі інформації наведеної вище сформулюємо технічні характеристики пристрою користувача (сформульовані на основі системних вимог браузеру Google Chrome):

Для Windows:

- Процесор Intel Pentium 4 або пізнішої версії із підтримкою SSE3;
- Об'єм оперативної пам'яті не менше 512 мб;
- Відео пам'ять не менше 64 мб;
- Вільне місце на жорсткому диску не менше 350 мб;
- Розширення екрану не менше 800 × 600;
- Клавіатура;
- Миша.

Для Linux:

- Технічні характеристики аналогічні Windows.

Далі технічні характеристики залежать формуватимуться на основі версії ОС, на базі якої працює пристрій, так як мінімальні вимоги до операційної системи, що підтримується, завжди перевершують технічні характеристики, які необхідні для коректної роботи браузера.

Для MacOS:

- Пристрій працює на базі OS X El Capitan (10.11) і версіях вище.

Для Android:

- Пристрій працює на базі Android 5.0 (Lollipop) і версіях вище.

Для IOS:

- Пристрій працює на базі iOS 12.2 (iPadOS 12.2) і версіях вище.

Оскільки частина системи з якою взаємодіє адміністратор представлена у вигляді АРМ, він зможе взаємодіяти з нею тільки через комп'ютер підключений до мережі авіакомпанії.

На основі інформації наведеної вище сформулюємо технічні характеристики пристрою адміністратора:

- Операційна система Windows 7 або пізніша версія;
- Процесор Intel Pentium 4 або пізнішої версії із підтримкою SSE3;
- Об'єм оперативної пам'яті не менше 512 мб;
- Відео пам'ять не менше 64 мб;
- Вільне місце на жорсткому диску не менше 350 мб;
- Розширення екрану не менше 800 × 600;
- Клавіатура;
- Миша.

Сервер, на якому працюватиме вся розроблювана інформаційна система, належить авіакомпанії та оброблює усі інформаційні потоки авіакомпанії. Технічні характеристики та налаштування відповідають вимогам авіакомпанії.

Адмініструють та проводять технічне обслуговування сервера працівники авіакомпанії.

На основі розробленого технічного проекту інформаційної системи реалізується проект інформаційної системи.

3. Реалізація проекту інформаційної системи авіакомпанії онлайн бронювання та реалізації авіаквитків.

Для експлуатації системи розроблена робоча документація за допомогою якої користувач та адміністратор можуть використовувати функціонал інформаційної системи.

Користувацька частина інформаційної системи містить велику кількість форм для заповнення, кожна зі своїми полями. Для полегшення використання системи кожне поле містить підказку для коректного заповнення.

Кожна кнопка користувацької частини інформаційної системи має невелику довідку щодо функціоналу, яка розташована біля неї. Розроблено модуль відстеження заповнених форм та коректності заповнень, який не дозволить перейти до наступного етапу купівлі, бронювання або анулювання авіаквитків якщо форма незаповнена, заповнена неповністю або заповнена неправильно.

Адміністративна частина інформаційної системи здебільшого складається із сервісу Uptrends. Робоча документація до сервісу Uptrends представлена за допомогою літератури, яку надає провайдер послуг сервісу.

Формування звітів має власні налаштування. Можна налаштувати формування звіту використовуючи матеріали обмежені заданим часом або критеріями. Модуль формування звітів містить форми для заповнення, кожна зі своїми полями. Для полегшення використання системи кожне поле містить підказку для коректного заповнення.

Кожна кнопка адміністративної частини інформаційної системи має невелику довідку щодо функціоналу, яка розташована біля неї. Розроблено модуль

відстеження заповнених форм та коректності заповнень, який не дозволить перейти до наступного етапу формування звіту якщо форма незаповнена, заповнена неповністю або заповнена неправильно.

Розроблено модуль відображення повідомлень про помилки як для користувачької так і для адміністративних частин інформаційної системи.

Проведено апробацію інформаційної системи. Перевірено на коректність роботи всіх модулів як користувальницької, так і адміністративної частини інформаційної системи. Перевірено коректність обміну даних між сервером та користувачем, між сервером та адміністратором. Перевірено коректність інтеграції та роботи сторонніх сервісів, використаних при розробці інформаційної системи. Перевірено коректність даних для заповнення бази даних.

Головна сторінка web-додатку інформаційної системи представлена на рисунку 3.1 [32].

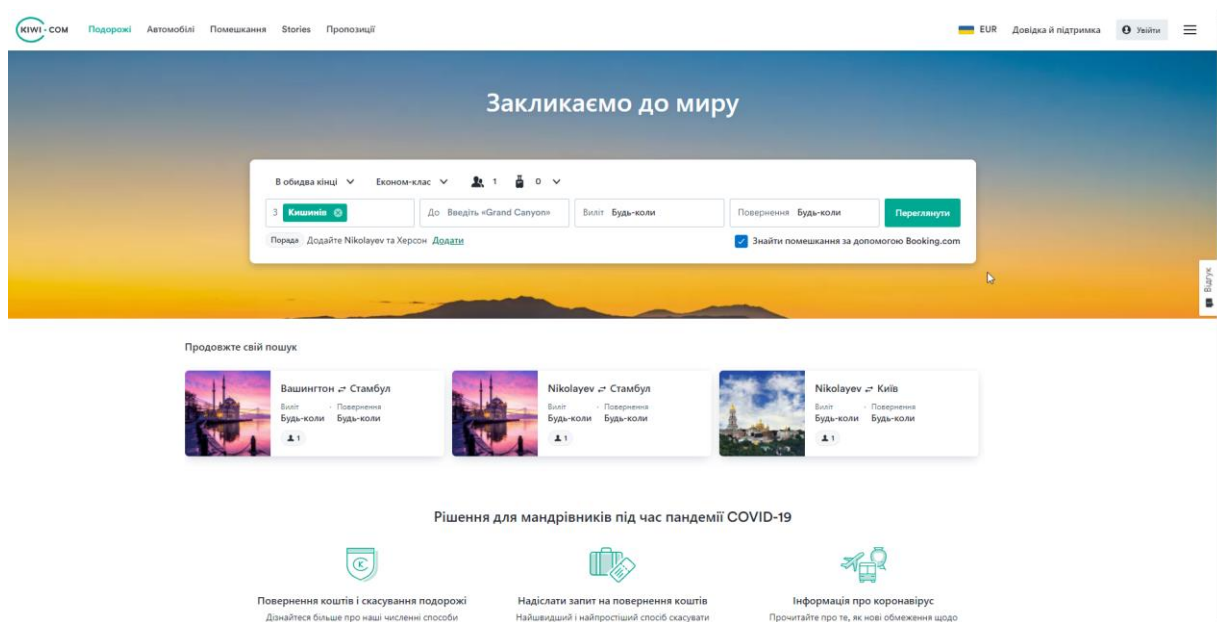


Рисунок 3.1 – Головна сторінка web-додатку

Сторінка зі списком знайдених рейсів по введеним критеріям представлена на рисунку 3.2 [32].

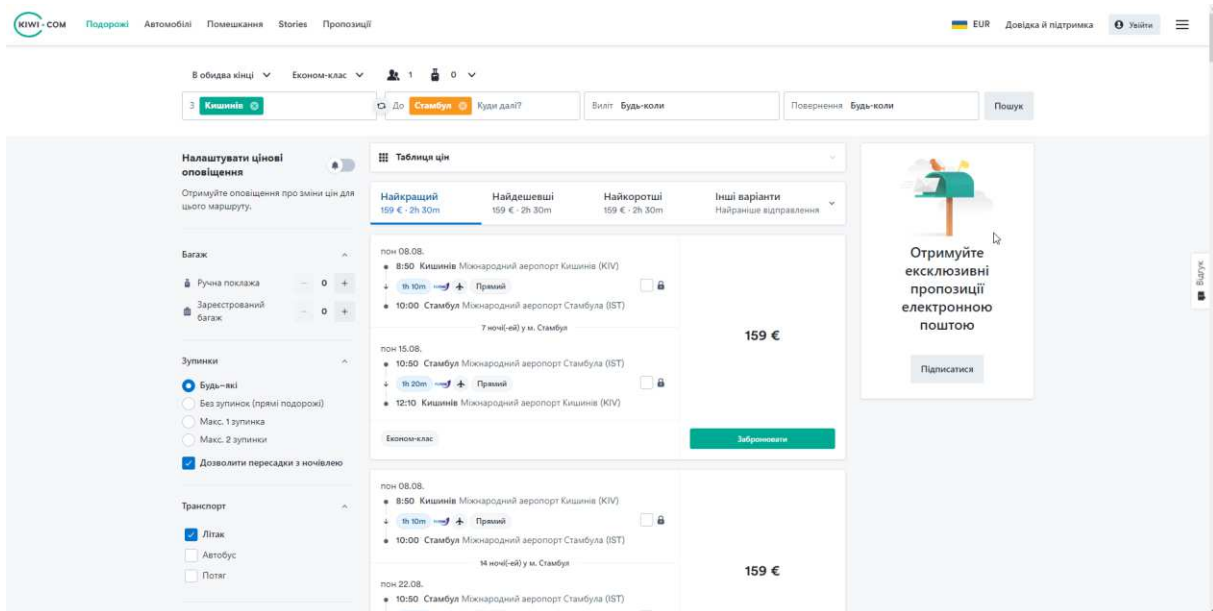


Рисунок 3.2 – Сторінка зі списком знайдених рейсів по введеним критеріям

Сторінка з формою для введення інформації про пасажирів та багаж представлена на рисунку 3.3 [32].

The screenshot shows the KIWI.COM website interface for passenger information. The form is divided into several sections: 'Контактна інформація' (Contact information) with fields for email and phone; 'Основний пасажир' (Main passenger) with fields for name, surname, date of birth, and passport number; and 'Ручна поклажа' (Hand luggage) with a section for selecting baggage options. There are also checkboxes for 'Hand luggage' and 'Baggage'. The form is designed to collect necessary information for flight booking.

Рисунок 3.3 – Сторінка з формою для введення інформації про пасажирів та багаж

Сторінка остаточного підтвердження вибраного рейсу та введених даних представлена на рисунку 3.4 [32].

Кишинів → Стамбул і назад

Огляд

Маршрут

FLY ONE 08:50 - 10:00 1h 10m
08.08. Кишинів KIV → Стамбул IST

7 ночей(ей) у м. Стамбул

FLY ONE 10:50 - 12:10 1h 20m
15.08. Стамбул IST → Кишинів KIV

Пасажири

Mr. Jake Dolanap 32.01.2000 12.01.2025 Без страховки

Бажаж

1x Особиста річ: 40 × 20 × 30 см, 10 kg J. Dolanap

Місця

1x Кишинів → Стамбул вилетів: призначене місце J. Dolanap

1x Стамбул → Кишинів вилетів: призначене місце J. Dolanap

До оплати (EUR) **211,47 €**

AirHelp+

Рисунок 3.4 - Сторінка остаточного підтвердження вибраного рейсу та введених даних

Сторінка з формою для введення реквізитів для оплати представлена на рисунку 3.5 [32].

Платіжні відомості

Використати промокод

Оплата

Платіжна картка ☒ PayPal ☐

Номер платіжної картки
1111 2222 3333 4444

Застінена строка дії (MM/YY) Код безпеки (CVV/CVC)
MM YY 123

Ім'я власника картки
напр., Anastasiya Shevchenko

Я приймаю(ю) та приймаю(ю) умови гарантії Kiwi.com (Пов'язані з цим гарантією повернення коштів і скасуванням, а також Політикою конфідційності компанії Kiwi.com (вибраного(ї) мною перевізника(ів)). Я також погоджуюся на негайне обслуговування компанії Kiwi.com підтвердити, що не можу відкликати від даного(их) про надання послуг.

Я доручаю компанії Kiwi.com забронювати ці подорожі у вибрані перевізники, використовуючи адресу електронної пошти з платіжної картки, яку компанія Kiwi.com створила для мене. Цим заявляю, що це мої персональні дані, та я надаю їх компанії Kiwi.com для цієї цілі цього бронювання. Я розумію, що можу здійснювати управління своїм бронюванням лише через компанію Kiwi.com, і дозволяю їй від мого імені організувати від перевізника електронні листи й повідомлення мій контактну інформацію.

Додаткова інформація

☐ На надсилайте мені спеціальні пропозиції та інші рекламні матеріали. [Відключити інформацію](#)

До оплати (EUR) **211,47 €**

Назад **Заплатити 211,47 € (0.00)**

Вебсайт www.kiwi.com належить компанії Kiwi.com s.r.o. (Словачка Республіка) та обслуговується нею.

Рисунок 3.5 - Сторінка з формою для введення реквізитів для оплати

Сторінка для ведення сайту представлена на рисунку 3.6 [33].

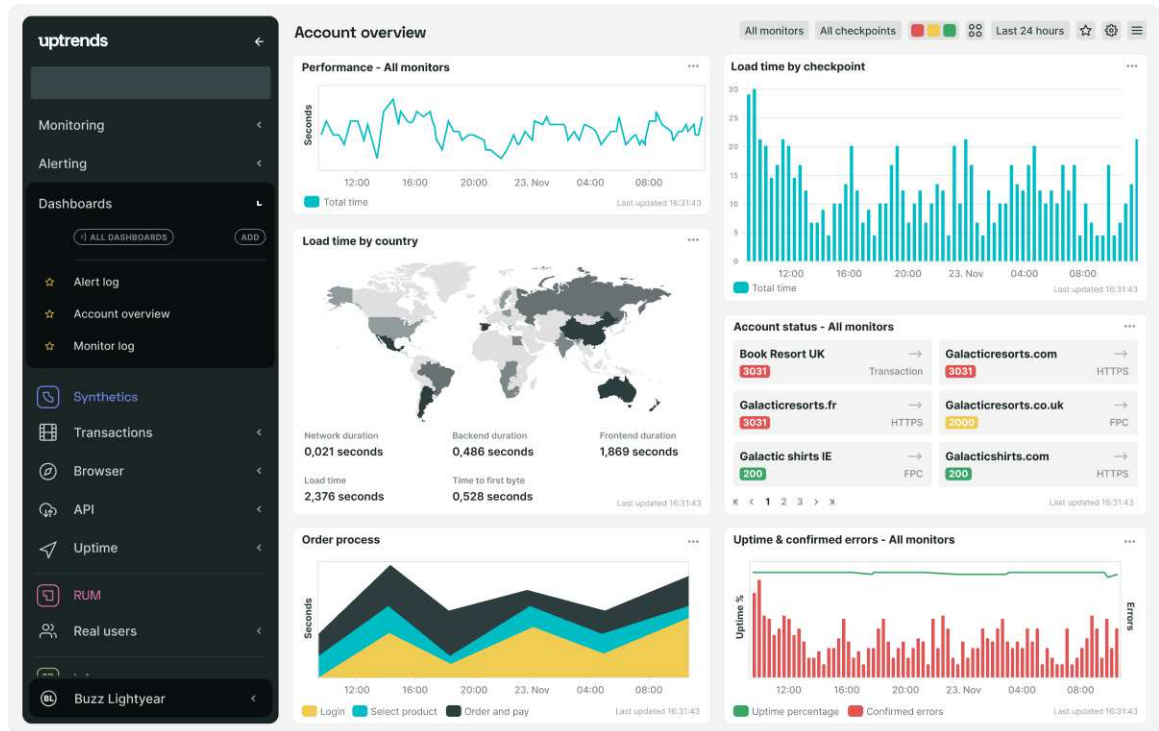


Рисунок 3.6 - Сторінка для ведення сайту

Сторінка відображення звітів у вигляді діаграм представлена на рисунку 3.7 [33].

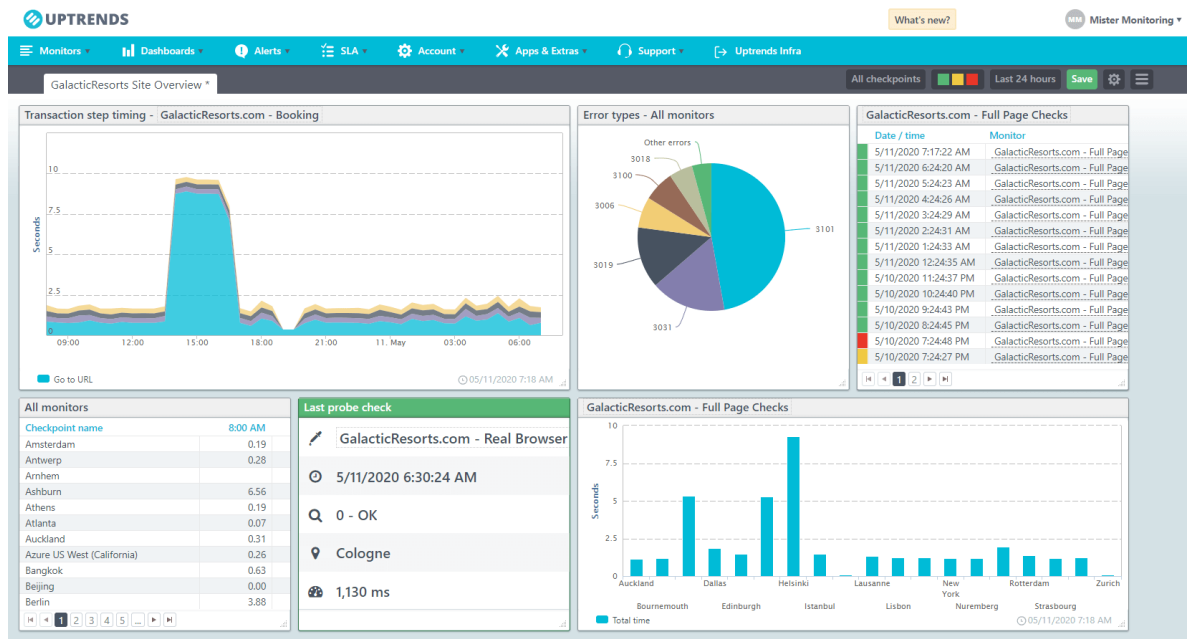


Рисунок 3.7 - Сторінка відображення звітів у вигляді діаграм

Апробація та експлуатація інформаційної системи дала позитивний результат. Ефективність від розробки та запровадження інформаційної системи полягає в економії часу і трудовитрат, оскільки частина клієнтів будуть використовувати розроблене рішення та навантаження на касирів зменшиться, збільшенні клієнтської бази, оскільки розроблене рішення дозволить підвищити конкурентоспроможність у сфері цивільної авіації та привабливість для потенційних клієнтів завдяки зручностям використання розробленого рішення.

4. Охорона праці.

Вимоги безпеки та захисту здоров'я працівників при роботі з екранними пристроями затверджено наказом Міністерства соціальної політики від 14.02.2018 № 207. Ці вимоги поширюються на всіх суб'єктів господарювання незалежно від форм власності, організаційно-правової форми та видів діяльності та встановлюють мінімальні вимоги безпеки та захисту здоров'я при здійсненні роботи, пов'язаної з використанням екранних пристроїв незалежно від їх типу та моделі.

Вимоги не поширюються на:

- Робочі місця здобувачів освіти у комп'ютерних класах (кабінетах, аудиторіях тощо) навчальних закладів;
- Робочі місця пілотів, водіїв або операторів транспортних засобів, обладнані екранними пристроями у системах обробки даних на борту засобів сполучення, екранні пристрої у складі машин та обладнання, що переміщуються у процесі роботи;
- Робочі місця працівників, які займаються обслуговуванням, ремонтом та налагодженням екранних пристроїв;
- Портативні системи обробки даних, якщо вони не завжди використовуються на робочому місці;
- Обчислювальні машини (калькулятори), касові апарати та пристрої з невеликими пристроями індикації даних або результатів вимірювання;

- Цифрові друкарські машини, які обладнані візуальними дисплейними терміналами (дисплейні друкарські машини);
- Планшети, смартфони, мобільні телефони.

Роботодавець повинен проінформувати працівників під розписку про умови праці та наявність на їх робочих місцях небезпечних та шкідливих виробничих факторів (фізичних, хімічних, біологічних, психофізіологічних), які виникають при роботі з екранними пристроями та ще не усунуті, а також про можливі наслідки їх впливу на здоров'я працівників.

Роботодавець повинен забезпечити навчання та перевірку знань працівників з питань охорони праці та безпечного використання екранних пристроїв до початку роботи з ними, а також у випадках модифікації та організації роботи обладнання.

Роботодавець повинен вжити відповідних заходів, щоб забезпечити відповідність робочого місця працівника цим вимогам.

Під час облаштування робочого місця працівника з екранними пристроями необхідно вибирати таке обладнання, яке не створює надмірного шуму і не виділяє надмірного тепла. Рівні шуму на робочих місцях осіб, які працюють з екранними пристроями, повинні відповідати вимогам Санітарних норм виробничого шуму, ультразвуку та інфразвуку ДСН 3.3.6.037-99, затверджених ухвалою Головного державного санітарного лікаря України від 01.12.1999 № 37.

Роботодавець повинен за рахунок тривалості робочої зміни організувати внутрішні регламентовані перерви для відпочинку відповідно до Державних санітарних правил та норм роботи з візуальними дисплейними терміналами електронно-обчислювальних машин ДСанПІН 3.3.2.007-98, затверджених постановою Головного державного санітарного лікаря від 10.12.1998 № 7.

Роботодавець повинен забезпечити власним коштом проведення медичних оглядів працівників відповідно до вимог Порядку проведення медичних оглядів працівників певних категорій, затвердженим наказом Міністерства охорони здоров'я України від 21.05.2007 № 246.

За результатами цих оглядів роботодавець у разі потреби повинен забезпечити виконання відповідних оздоровчих заходів.

Роботодавець зобов'язаний за необхідності проводити лабораторні дослідження умов праці працівників з метою виявлення шкідливих та небезпечних факторів виробничого середовища, тяжкості та напруженості трудового процесу (зокрема щодо виявлення ризиків, пов'язаних із погіршенням зору, порушенням фізичного стану, стресом) та здійснювати заходи щодо усунення виявлених ризиків.

Робочі місця працівників з екранними пристроями мають бути спроектовані і мати такі розміри, щоб працівники мали простір зміни робочого становища і рухів.

Для забезпечення безпеки та захисту здоров'я працівників все випромінювання від екранних пристроїв має бути зведене до гранично допустимого рівня (вплив на людину факторів середовища - шуму, вібрації, забруднювачів, температури тощо, що не викликає соматичних або психічних розладів, а також змін стану здоров'я, працездатності, поведінки, що виходять за межі пристосувальних реакцій з точки зору безпеки та охорони здоров'я працівників).

Організація робочого місця працівника з екранними пристроями повинна забезпечувати відповідність всіх елементів робочого місця та їх розташування ергономічним, антропологічним, психофізіологічним вимогам, а також характером виконуваних робіт.

Висвітлення робочого місця працівника з екранними пристроями має створювати відповідний контраст між екраном та навколишнім середовищем (з урахуванням виду роботи) та відповідати вимогам ДСанПН 3.3.2.007-98.

Мікроклімат виробничих приміщень з робочими місцями працівників з екранними пристроями повинен підтримуватися на постійному рівні та відповідати вимогам Санітарних норм мікроклімату виробничих приміщень ДСН 3.3.6.042-99, затверджених постановою Головного державного санітарного лікаря України від 0.12.1999 №42 (далі - ДСП 3.3.6.042-99).

Робочий стіл або робоча поверхня повинні бути достатнього розміру та мати поверхню з низькою відбивною здатністю, допускати гнучкість при розміщенні екрана, клавіатури, документів та відповідного обладнання.

Робоче крісло має бути стійким і дозволяти працівникові з екранними пристроями легко рухатися та займати зручне положення.

Сидіння має регулюватися по висоті, спинка сидіння як по висоті, так і по нахилу. Слід передбачати підніжку для тих, кому це потрібно для зручності.

Мінімальні вимоги безпеки під час роботи з екранними пристроями:

- Щодня перед початком роботи необхідно очищати екранні пристрої від пилу та інших забруднень;
- Після завершення роботи екранні пристрої слід відключати від електричної мережі;
- У разі виникнення аварійної ситуації необхідно негайно відключити пристрій від електричної мережі.

Не допускається:

- Виконувати технічне обслуговування, ремонт та налагодження екранних пристроїв безпосередньо на робочому місці працівника під час роботи з екранними пристроями;
- Відключати захисні пристрої, самовільно проводити зміни у конструкції та складі екранних пристроїв або їх технічне налагодження;
- Працювати з екранними пристроями, де під час роботи виникають нехарактерні сигнали, нестабільне зображення на екрані та інші несправності.

При виконанні робіт операторського типу, пов'язаних з нервово-емоційною напругою, у приміщеннях при роботі з екранними пристроями, на пультах та постах управління технологічними процесами та в інших приміщеннях повинні дотримуватися оптимальних умов мікроклімату відповідно до вимог ГСН 3.3.6.042-99.

Мінімальні вимоги безпеки до екранних пристроїв:

- Екранні пристрої не повинні бути джерелом ризику для працівників;
- Все випромінювання, крім видимої частини електромагнітного спектра, має бути зведено до незначного рівня з погляду безпеки та охорони здоров'я працівників;
- Символи на екранних пристроях повинні бути чіткими, відповідного розміру. Між символами та рядками символів має бути належна відстань;
- Зображення на екрані має бути стабільним, без миготінь чи інших видів нестабільності;
- Яскравість та/або контрастність символів повинні легко регулюватися працівником під час роботи з екранними пристроями, а також швидко адаптуватися до навколишніх умов;
- Вибираючи екрани, слід віддавати перевагу таким екранам, які легко та вільно обертаються та нахиляються відповідно до потреби працівника;
- При необхідності може використовуватися окрема підставка або регульований стіл для розміщення екрана;
- Екран не повинен відблискувати або відображати світло, щоб не викликати дискомфорту у працівника під час роботи з екранними пристроями;
- Вибираючи клавіатуру, слід віддавати перевагу такій клавіатурі, яка відкидається і є автономною (відокремленою від екрану), щоб працівник міг вибрати зручну робочу позу та уникнути втоми рук (кисті та верхньої частини руки);
- Поверхня клавіатури має бути матовою, щоб уникнути відображення. Розташування клавіш і клавіші повинні полегшувати роботу з клавіатурою. Позначення клавіш має бути досить контрастним та розбірливим;
- Обладнання, що входить до робочої станції, не повинно виділяти надлишкового тепла, що може заподіяти незручності працівникам при роботі з екранними пристроями;

- При розробці, виборі, замовленні та модифікації програмного забезпечення, а також при розробці задач, що передбачають використання обладнання з екранними пристроями, роботодавець повинен керуватися таким програмним забезпеченням, яке відповідає розв'язуванню завданням і просто у використанні, а де необхідно адаптованим до рівня знань та досвіду працівника.

Відповідно до статті 43 Закону України «Про охорону праці» за порушення законодавства про охорону праці та невиконання приписів (розпоряджень) посадових осіб органів виконавчої влади з нагляду за охороною праці юридичні та фізичні особи, які згідно із законодавством використовують найману працю, залучаються органами виконавчої влади з нагляду за охороною праці до сплати штрафу у порядку, встановленому законом.

Сплата штрафу не звільняє юридичну чи фізичну особу, яка відповідно до законодавства використовує найману працю, від усунення виявлених порушень у визначені терміни.

Максимальний розмір штрафу не може перевищувати п'яти відсотків середньомісячного фонду заробітної плати за попередній рік юридичної або фізичної особи, яка відповідно до законодавства використовує найману працю.

Несплата або неповна сплата юридичними або фізичними особами, які відповідно до законодавства використовують найману працю, штрафу тягне за собою нарахування пені на несплачену суму штрафу (його частини) з розрахунку 120% річних облікової ставки Національного банку України, що діяла в період такої несплати, за кожен день прострочення.

Інформація про охорону праці підготовлена на основі матеріалу [34].

Висновки

У дипломному проекті розроблена інформаційна система авіакомпанії онлайн бронювання та реалізації авіаквитків. Поставленою ціллю інформаційної системи є надання можливості бронювати та купувати авіаквитки у будь-який час, у будь-якому місці, незалежно від розкладу авіакомпанії чи аеропорту, через web-додаток.

Розроблена концепції інформаційної системи. Зроблений опис об'єкту дослідження і аналіз предметної області. Сформовані вимоги до інформаційної системи. На основі сформованих вимог розроблена концепція, яка визначає функціонал інформаційної системи.

Розроблений технічний проект інформаційної системи. Наведені загальносистемні рішення для обробки електронних платежів, введення сайту, формування та розсилання електронних листів. У інформаційному забезпеченні представлені концептуальна, логічна та фізична моделі бази даних. У математичному забезпеченні представлена блок-схема алгоритму роботи проекту. У програмному забезпеченні описані засоби та технології розробки проекту. У технічному забезпеченні описані характеристики пристроїв користувача та адміністратора для задовільної експлуатації інформаційної системи.

Розроблена робоча документація, де описані механізми підказок для форм, полів, кнопок та перевірки даних введених користувачем. Проведена апробація проекту, яка показала коректну роботу усіх програмних модулів інформаційної системи.

На завершення роботи представлені правила охорони праці під час роботи за комп'ютером.

Актуальність та новизна проекту полягає у підвищенні зручності купівлі та бронювання авіаквитків за рахунок можливості зробити це віддалено з персонального комп'ютеру, смартфона, планшету або будь-якого іншого пристрою, який може запускати браузер. Це дозволило підвищити конкурентоспроможність у сфері цивільної авіації та привабливість для потенційних клієнтів.

Список використаної літератури

1. Белов В.В., В.И. Чистякова Проектирование информационных систем: учебник для студ. учреждений высш. проф. Образования; под ред. В.В. Белова. М: Издательский центр «Академия», 2013. 352с.
2. Гагарина Л.Г., Кокорева Е.В., Виснадул Б.Д. Технология разработки программного обеспечения: учебное пособие/ под ред. Л.Г. Гагариной. – М.: ИД «ФОРУМ»: ИНФА- М, 2008.— 400 с.: ил.— (Высшее образование).
3. ГОСТ 34.601-90 «Информационная технология. Комплекс стандартов на автоматизированные системы. Автоматизированные системы. Стадии создания».
4. ГОСТ 34.602-89 «Информационная технология. Комплекс стандартов на автоматизированные системы. Техническое задание на создание автоматизированной системы».
5. ГОСТ 19.701-90 Единая система программной документации.
Схемы алгоритмов, программ данных и систем.
6. М.Фаулер UML. Основы, 3-издание, пер. с английского А.Петухова. Изд-во Символ, СПб. 2005.
7. Грекул В.И., Г.Н. Денищенко. Проектирование информационных систем: курс лекций. М. Интернет – Ун-т Информ. технологий, 2005. 304с.
8. Клевцов С.И. Анализ и формирование требований к программному обеспечению информационных систем сбора и обработки данных: учебное пособие. Таганрог: Изд-во ТТИЮФУ, 2007. 100с.
9. Марченко А.В. Проектування інформаційних систем 2016 90с. Режим доступу:https://elearning.sumdu.edu.ua/free_content/lectured:de1c9452f2a161439391120eef364dd8ce4d8e5e/20160217112601/content-20160217112601.pdf; Назва з екрана.
10. Пальмов С.В. Проектирование информационных систем. М: ИСТ- 72, 2011. 51 с.
11. Ременяк Л.В. Проектування інформаційних систем: конспект лекцій. Одеса, Одеський державний екологічний університет, 2016. 152.

12. Хомоненко А.Д., Цыганков В.М., Мальцев М.Г. Базы данных: Учебник для высших учебных заведений. Под ред. Проф. А.Д. Хомоненко. – 6-е изд. доп. – СПб.: КОРОНА – Век. 2009. – 736 с.
13. Грекул В. И. Проектирование информационных систем. Практикум. М.: Национальный открытый университет «ИНТУИТ», 2012. –159 с.
14. Вендров А. М. Проектирование программного обеспечения экономических информационных систем. М.: Финансы та статистика, 2006. – 544 с.
15. UML [Электронный ресурс]: Вікіпедія. Вільна енциклопедія. – Режим звернення: <https://ru.wikipedia.org/wiki/UML> – дата звернення: 06.04.2021.
16. Джошуа Блох. Ефективна Java;[В оригіналі] 2016. –440 с
17. Джексон Г. Проектирование реляционных баз данных для использования с ЭВМ: Пер. с англ. – М.: Мир, 1991. –252 с.
18. Марченко А.В. Проектування інформаційних систем 2016 90с. Режим доступу:https://elearning.sumdu.edu.ua/free_content/lectured:delc9452f2a161439391120eef364dd8ce4d8e5e/20160217112601/content-20160217112601.pdf;
19. <https://www.floridatechonline.com/blog/aviation-management/how-airlines-and-airports-work-together/#:~:text=Airline%20partnership%20is%20what%20drives,airlines%20to%20conduct%20their%20business.>
20. <https://www.arhivinfo.ru/1-1171.html>
21. <https://www.pinterest.com/pin/464011567854426736/>
22. <https://actualtraffic.ru/site/uptrends/>
23. <https://spark.ru/user/142104/blog/87996/obzor-sistemi-dlya-priema-platezhej-2sheckout>
24. <https://startpack.ru/application/sendpulse>
25. <https://uk.wikipedia.org/wiki/MySQL>
26. <https://uk.wikipedia.org/wiki/SQL>
27. <https://ru.wikipedia.org/wiki/JavaScript>

28. <https://ru.wikipedia.org/wiki/HTML>
29. <https://ru.wikipedia.org/wiki/CSS>
30. <https://ru.wikipedia.org/wiki/PHP>
31. <https://ru.wikipedia.org/wiki/HTTPS>
32. <https://kiwi.com>
33. <https://uptrends.com>
34. <https://hrliga.com/index.php?module=news&op=view&id=20212>

Додаток А – Інструкція користувачу

На головній сторінці представлена форма для пошуку бажаного рейсу, в яку користувач може ввести критерії для пошуку і отримати список відповідних рейсів. Форма для пошуку потрібного рейсу представлена на рисунку А.1 [32].

The screenshot shows a flight search interface. At the top, there are dropdowns for 'В один кінець' (One way), 'Економ-клас' (Economy class), and a passenger count of 1. Below these are input fields for the origin 'Париж' (Paris) and destination 'Барселона' (Barcelona). There is a 'Куди далі?' (Where to?) button, a 'Виліт' (Departure) field with 'Будь-коли' (Anytime) selected, and a 'Без повернення' (No return) option. A 'Пошук' (Search) button is on the right.

Рисунок А.1 - Форма для пошуку потрібного рейсу

Список відповідних рейсів представлений на рисунку А.2 [32].

The screenshot displays a list of flight results for the Paris to Barcelona route. Each result includes the date, flight details, price, and a 'Забронювати' (Book) button.

Дата	Виліт	Час в польоті	Клас	Ціна	Додаткова інформація
птн 26.08.	8:55 Париж Аеропорт Бове (BVA)	1h 45m	Економ-клас	14 €	3 місць(-я) залишилися за цією ціною
чтв 25.08.	8:55 Париж Аеропорт Бове (BVA)	1h 45m	Економ-клас	14 €	1 місце залишилося за цією ціною
пон 10.10.	8:55 Париж Аеропорт Бове (BVA)	1h 45m	Економ-клас	14 €	

Рисунок А.2 – Список відповідних рейсів

Пошук буде неможливий якщо обов'язкові поля не заповнені. У цьому випадку обов'язкові незаповнені поля матимуть індикатор у вигляді червоного обведення. Неможливість пошуку представлено на рисунку А.3 [32].

В обидва кінці ▼ Економ-клас ▼ 1 0 ▼

3 До Введіть «Machu Picchu» Виліт Будь-коли Повернення Будь-коли Пошук

Рисунок А.3 – Нemoжливiсть пошуку

Теж саме стосується усіх інших форм, наприклад форма для інформації про пасажирів та його багаж, яку треба буде заповнити після вибору рейсу. У цьому випадку крім червоного обведення біля кожного поля буде повідомлення про необхідність заповнення або некоректне заповнення. Форма для інформації про пасажирів та його багаж представлена на рисунку А.4 [32].

Контактна інформація

Електронна пошта:

Телефон:

Основний пасажир: Дорослий (від 16 років) ▼

Вказуйте всі імена й прізвища точно так, як вони записані в паспорті/ідентифікаційній картці, щоб уникнути проблем під час посадки.

Ім'я:

Прізвище(-а):

Громадянство:

Стать:

Дата народження:

Ручна поклажа

Виберіть один варіант:

1 × особиста річ

Має вміщатися під переднім сидінням



40 × 20 × 25 cm

0 €

Популярно

Пакет послуг для перевезення ручної поклажі

1 × особиста річ + 1 × ручна поклажа (10 кг) + Пріоритетна посадка (Ryanair)

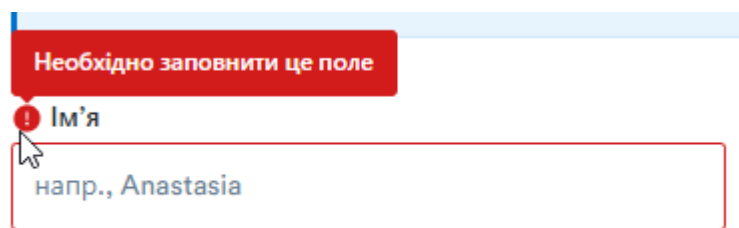
  

40 × 20 × 25 cm 55 × 20 × 40 cm Не стійте в черзі

30.8 €

Рисунок А.4 - Форма для інформації про пасажирів та його багаж

Індикація та повідомлення про необхідність заповнення представлені на рисунку А.5 [32].



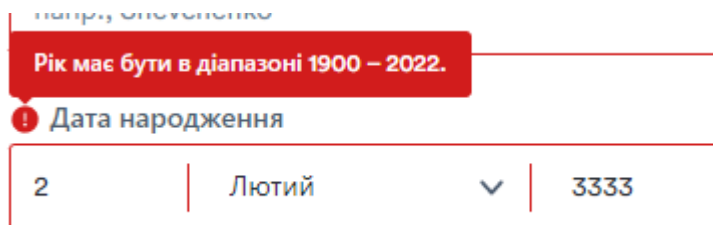
Необхідно заповнити це поле

Ім'я

напр., Anastasia

Рисунок А.5 - Індикація та повідомлення про необхідність заповнення

Індикація та повідомлення про некоректне заповнення представлені на рисунку А.6 [32].



Рік має бути в діапазоні 1900 – 2022.

Дата народження

2 | Лютий | 3333

Рисунок А.6 - Індикація та повідомлення про некоректне заповнення

За потреби можливо додати інших пасажирів якщо купити або забронювати потрібно не один авіаквиток. Кнопка для цього розташована під формою для інформації про пасажирів та його багаж. У цьому разі потрібно буде ввести інформацію про додаткових пасажирів та їх багаж. Кнопка для додавання пасажирів представлена на рисунку А.7 [32].

Ручна поклажа ⓘ
Виберіть один варіант:

1 × особиста річ
Має вміститися під переднім сидінням


 40 x 20 x 25 cm

0 €
ⓘ

Популярна

Пакет послуг для перевезення ручної поклажі
1 × особиста річ + 1 × ручна поклажа (10 кг) + Пріоритетна посадка (Ryanair)


 40 x 20 x 25 cm


 55 x 20 x 40 cm


 Не стійте в черзі

30.8 €
ⓘ

✈️ **Порада:** Подорожуйте безпечно, додавши замість цього зареєстрований багаж. Ви можете взяти менше речей у салон, уникаючи в такий спосіб додаткового переміщення.

Зареєстрований багаж ⓘ
Виберіть один варіант:

⚡ **Як заощадити кошти:** оплатіть зареєстрований багаж зараз, а не після бронювання, щоб зекономити до 25% на зборах авіакомпаній.

☒ Без зареєстрованого багажу

Страхування подорожуючих Здійснюється компанією AXA Assistance ⓘ [Порівняння та Умови](#)

ⓘ **Покриває захворювання COVID-19 і пов'язане з ним лікування**

✓ Travel Plus успішно додано. [Змінити](#)

+ Додати іншого пасажирів

Рисунок А.7 - Кнопка для додавання пасажирів

Після закінчення вводу інформації потрібно буде вибрати місце у літаку. Сторінка з вибором місця в літаку представлена на рисунку А.8 [32].

Виберіть місце на схемі

☐ 3 додатковим простором для ніг (27 € – 38 €)
 ☐ Стандартне (17 € – 31 €)
 ☐ Недоступне місце * Місце, щодо якого діють певні правила

	A	B	C	D	E	F
Вихід						Вихід
Економ-клас (ряди 1 – 33)						
1	☐	☐	☐			
2	☐	☐	☐	☐	☐	☐
3	☐	☐	☐	☒ JS	☐	☐
4	☐	☐	☐	☐	☐	☐

Рисунок А.8 - Сторінка з вибором місця в літаку

Остаточним кроком для купівлі авіаквитка буде оплата. Для цього потрібно ввести платіжну інформацію у відповідну форму та підтвердити купівлю натиснувши на відповідну кнопку. Форма для введення платіжної інформації представлена на рисунку А.9 [32].

Оплата VISA

☒ Платіжна картка
 ☐ PayPal

Номер платіжної картки

1111 2222 3333 4444

Закінчення строку дії (MM/PP)

MM YY

Код безпеки (CVC/CVV)

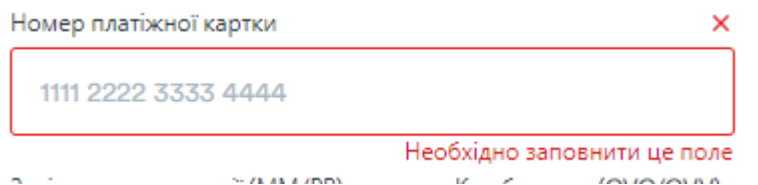
123

Ім'я власника картки

напр., Anastasia Shevchenko

Рисунок А.9 - Форма для введення платіжної інформації

Як і інші форми має індикацію та повідомлення про незаповненість обов'язкових полів. Індикація та повідомлення про незаповненість обов'язкових полів представлені на рисунку А.10 [32].



Номер платіжної картки

1111 2222 3333 4444

Необхідно заповнити це поле

Рисунок А.10 - Індикація та повідомлення про незаповненість обов'язкових полів

Додаток Б – Текст програми

connect.php

```
<?php
$host = 'localhost';
$database = 'airticketbooking';
$user = 'aven';
$password = "divide";

$con = mysqli_connect($host, $user, $password, $database);

if (!$con) echo "Не удалось подключиться к СУБД: " . mysqli_connect_error();
else mysqli_select_db($con, "info");
```

fillAirplaneSelector.php

```
<?php

include("connect.php");

global $con;

$sql = "SELECT id FROM airplane";
$result = mysqli_query($con, $sql) or die('Запрос невозможен!');
$row = mysqli_fetch_array($result);

$airplaneIdArray = array();

do {
    array_push($airplaneIdArray, $row['id']);
} while ($row = mysqli_fetch_array($result));

for ($i = 0; $i < count($airplaneIdArray); $i++) {
    echo "<option>$airplaneIdArray[$i]</option>";
}
```

fillCitySelector.php

```
<?php

include("connect.php");

global $con;

$sql = "SELECT * FROM city";
$result = mysqli_query($con, $sql) or die('Запрос невозможен!');
$row = mysqli_fetch_array($result);

$cityArray = array();
```

```
do {
    array_push($cityArray, array("id" => $row['id'], "name" => $row['name']));
} while ($row = mysqli_fetch_array($result));

echo json_encode($cityArray, JSON_UNESCAPED_UNICODE);
```

search.php

```
<?php

include("connect.php");

global $con;

$option = $_POST["option"];

$sql = "SELECT t.id AS id, t.name AS departure, city.name AS arrival,
flight.departureDateAndTime AS 'date', flight.flightDuration AS duration,
flight.class AS class, flight.airplaneId AS airplane, flight.price AS price
FROM(SELECT flight.id, city.name, flight.departureCityId,
flight.arrivalCityId
FROM flight, city
WHERE flight.departureCityId = city.id) AS t, city, flight
WHERE t.id = flight.id AND city.id = t.arrivalCityId" . $option . "
ORDER BY id";

$array = array();

$result = mysqli_query($con, $sql) or die('Запрос невозможен!');
$row = mysqli_fetch_array($result);

if($row == null) {
    echo "<p style='margin-top: 10px; margin-left: 10px;'>Ничего не найдено... \_(
ツ)_/ </p>";
    exit;
}

$flightArray = array();

do {
    array_push($flightArray, array(
        "id" => $row['id'],
        "departure" => $row['departure'],
        "arrival" => $row['arrival'],
        "date" => $row['date'],
        "duration" => $row['duration'],
        "class" => $row['class'],
        "airplane" => $row['airplane'],
        "price" => $row['price']));
} while ($row = mysqli_fetch_array($result));

echo "<tr>
<th>Код</th>
```



```

                <th>Отбытие</th>
                <th>Прибытие</th>
                <th>Дата</th>
                <th>Длительность</th>
                <th>Класс</th>
                <th>Самолёт</th>
                <th>Цена</th>
                <th class='create'+</th>
            </tr>";

for ($i = 0; $i < count($flightArray); $i++) {
    echo
    "<tr>
        <td>" . $flightArray[$i]['id'] . "</td>
        <td class='changeable'>" .
$flightArray[$i]['departure'] . "</td>
        <td class='changeable'>" . $flightArray[$i]['arrival']
. "</td>
        <td class='changeable'>" . $flightArray[$i]['date'] .
"</td>
        <td class='changeable'>" . $flightArray[$i]['duration']
. "</td>
        <td class='changeable'>" . $flightArray[$i]['class'] .
"</td>
        <td class='changeable'>" . $flightArray[$i]['airplane']
. "</td>
        <td class='changeable'>" . $flightArray[$i]['price'] .
"</td>
        <td class='delete'>-</td>
    </tr>";
}

```

create.php

```

<?php

include("connect.php");

$id = $_POST['id'];
$departureCityId = $_POST['departureCityId'];
$arrivalCityId = $_POST['arrivalCityId'];
$departureDateAndTime = $_POST['departureDateAndTime'];
$flightDuration = $_POST['flightDuration'];
$class = $_POST['class'];
$airplaneId = $_POST['airplaneId'];
$price = $_POST['price'];

$sql = "INSERT INTO flight (id, departureCityId, arrivalCityId,
departureDateAndTime, flightDuration, class, airplaneId, price)
VALUES (N'" . $id . "',
" . $departureCityId . ",
" . $arrivalCityId . ",

```

```
'" . $departureDateAndTime . "',
" . $flightDuration . ",
N'" . $class . "',
N'" . $airplaneId . "',
" . $price . ")";

global $con;
$result = mysqli_query($con, $sql) or die("Запрос Невозможен");
echo $result;
```

delete.php

```
<?php

include("connect.php");

$id = $_POST["id"];

$sql = "DELETE FROM flight WHERE id = '" . $id . "'";

global $con;
$result = mysqli_query($con, $sql) or die("Запрос Невозможен");
echo $result;
```

update.php

```
<?php

include("connect.php");

$attributeArray = [
    1 => "departureCityId",
    2 => "arrivalCityId",
    3 => "departureDateAndTime",
    4 => "flightDuration",
    5 => "class",
    6 => "airplaneId",
    7 => "price",
];

$id = $_POST['id'];
$attributeIndex = $_POST['attributeIndex'];
$attributeValue = $_POST['attributeValue'];

$sql = "UPDATE flight SET " . $attributeArray[$attributeIndex] . " = '" .
$attributeValue . "'
WHERE id = '" . $id . "'";

global $con;
```

```
$result = mysqli_query($con, $sql) or die("Запрос Невозможен");
echo $result;
```

processor.js

```
$(document).ready(()=>{

    class City {

        constructor(id, name) {
            this.id = id;
            this.name = name;
        }

        getId() {
            return this.id;
        }

        getName() {
            return this.name;
        }

        toString() {
            return this.name;
        }
    }

    let messageLabel = $("#messageLabel");

    let isInput;

    let cityId = [];

    fillCitySelector("#departureCitySelector", "Город Отбытия");
    fillCitySelector("#arrivalCitySelector", "Город Прибытия");

    $("#searchForm").on("submit", function(event) {

        messageLabel.empty();

        let option = "";

        let departureCitySelector =
document.getElementById("departureCitySelector");
        let departureCityId = cityId[departureCitySelector.selectedIndex - 1];
        let departureCityName = departureCitySelector.value;
        let departureIndex = departureCitySelector.selectedIndex;

        let arrivalCitySelector = document.getElementById("arrivalCitySelector");
        let arrivalCityId = cityId[arrivalCitySelector.selectedIndex - 1];
        let arrivalCityName = departureCitySelector.value;
        let arrivalIndex = arrivalCitySelector.selectedIndex;
```

```

let _class = document.getElementById("classInput").value;
let priceFrom = document.getElementById("priceFromInput").value;
let priceTo = document.getElementById("priceToInput").value;
let date = document.getElementById("dateInput").value;

if(departureIndex > 0 && arrivalIndex > 0 && departureCityId ===
arrivalCityId) {
    messageLabel.empty().append("Поиск: город отбытия(" + departureCityName
+ ") и город прибытия(" + arrivalCityName + ") не могут быть одинаковыми.");
    event.preventDefault();
    return;
}

if(parseInt(priceFrom) > parseInt(priceTo)) {
    messageLabel.empty().append("Поиск: цена от(" + priceFrom + ") должна
быть меньше или равна цене до(" + priceTo + ").");
    event.preventDefault();
    return;
}

if(departureIndex > 0) {
    option += " AND t.departureCityId = '" + departureCityId + "'";
}

if(arrivalIndex > 0) {
    option += " AND t.arrivalCityId = '" + arrivalCityId + "'";
}

if(!(_class === "")) {
    option += " AND class = '" + _class + "'";
}

if!(priceFrom === "") {
    option += " AND price >= '" + priceFrom + "'";
}

if!(priceTo === "") {
    option += " AND price <= '" + priceTo + "'";
}

if!(date === "") {
    option += " AND departureDateAndTime = '" + date + ":00'";
}

$.ajax({
    async: false,
    url: "../php/search.php",
    type: "POST",
    dataType: "html",
    data: {
        option: option
    },
    beforeSend: () => console.log("searching..."),
    success: (data) => {

```

```

        $("table").empty().append(data);
    }
});
addChangeOnTable();
addDeleteOnTable();
addCreateOnTable();

console.log("-----");
event.preventDefault();

});

function addChangeOnTable() {

    let index;
    let oldValue;
    let currentCell;
    let row;

    $('.changeable').click(function() {

        messageLabel.empty();

        if(isInput) return;

        isInput = true;

        if ($(this).text() !== "") {
            oldValue = $(this).text();
        }

        currentCell = this;
        row = $(this).parent();

        index = $(this).closest('td').index();

        switch(index) {
            case 1:
            case 2:
                $(this).empty().append("<form id='updateForm'><label><select
id='temporary'></select></label><button type='submit'
id='acceptUpdate'></button><span id='declineUpdate'></span></form>");
                fillCitySelector("#temporary", "");
                $('#temporary option:contains(' + oldValue +
                ')').attr("selected", "selected");
                break;
            case 3:
                $(this).empty().append("<form id='updateForm'><label><input
id='temporary' type='datetime-local' required='required' value='" +
                oldValue.replace(" ", "T") + "'><button type='submit'
id='acceptUpdate'></button><span id='declineUpdate'></span></form>");
                break;
            case 4:

```

```

        $(this).empty().append("<form id='updateForm'><label><input
id='temporary' type='number' required='required' min='1' max='9999' step='1'
value='" + oldValue + "'></label><button type='submit'
id='acceptUpdate'></button><span id='declineUpdate'></span></form>");
        break;
    case 5:
        $(this).empty().append("<form id='updateForm'><label><input
id='temporary' type='text' required='required' maxlength='10' value='" + oldValue +
"'></label><button type='submit' id='acceptUpdate'></button><span
id='declineUpdate'></span></form>");
        break;
    case 6:
        $(this).empty().append("<form id='updateForm'><label><select
id='temporary'></select></label><button type='submit'
id='acceptUpdate'></button><span id='declineUpdate'></span></form>");
        fillAirplaneSelector("#temporary", "");
        $('#temporary option:contains(' + oldValue +
')').attr("selected", "selected");
        break;
    case 7:
        $(this).empty().append("<form id='updateForm'><label><input
id='temporary' type='number' required='required' min='1' max='9999' step='any'
value='" + oldValue + "'></label><button type='submit'
id='acceptUpdate'></button><span id='declineUpdate'></span></form>");
        break;
    default:
        return;
}

$("#updateForm").on("submit", function(event) {

    messageLabel.empty();

    let newValue;
    let newTableValue;
    let id = $(row).find("td").eq(0).html();

    switch (index) {
        case 1:
            let departureCitySelector =
document.getElementById("temporary");
            let arrivalCityName = $(row).find("td").eq(2).html();
            newTableValue =
departureCitySelector.options[departureCitySelector.selectedIndex].value;
            newValue = cityId[departureCitySelector.selectedIndex];
            if(arrivalCityName === newTableValue) {
                messageLabel.empty().append("Редактирование: город
отбытия(" + newTableValue + ") и город прибытия(" + arrivalCityName + ") не могут
быть одинаковыми.");
                event.preventDefault();
                return;
            }
            break;
        case 2:

```

```

        let arrivalCitySelector =
document.getElementById("temporary");
        let departureCityName = $(row).find("td").eq(1).html();
        newTableValue =
arrivalCitySelector.options[arrivalCitySelector.selectedIndex].value;
        newValue = cityId[arrivalCitySelector.selectedIndex];
        if(departureCityName === newTableValue) {
            messageLabel.empty().append("Редактирование: город
отбытия(" + departureCityName + ") и город прибытия(" + newTableValue + ") не могут
быть одинаковыми.");
            event.preventDefault();
            return;
        }
        break;
    case 6:
        let airplaneSelector =
document.getElementById("temporary");
        newTableValue = newValue =
airplaneSelector.options[airplaneSelector.selectedIndex].value;
        break;
    case 3:
        newTableValue = newValue =
document.getElementById("temporary").value.replace("T", " ");
        break;
    case 7:
        newTableValue = newValue =
document.getElementById("temporary").value;
        break;
    default:
        newTableValue = newValue =
document.getElementById("temporary").value;
        break;
    }

    if (newTableValue === "" || newTableValue === oldValue)
$(currentCell).replaceWith("<td class='changeable'>" + oldValue + "</td>");
    else {
        $.ajax({
            async: false,
            url: "../php/update.php",
            type: "POST",
            dataType: "html",
            data: {
                id: id,
                attributeIndex: index,
                attributeValue: newValue
            },
            beforeSend: () => console.log("updating..."),
            success: (data) => {
                if (data === "1") {
                    if(index === 3) newTableValue += ":00";
                    $(currentCell).replaceWith("<td
class='changeable'>" + newTableValue + "</td>");
                }
                else $(currentCell).replaceWith("<td

```

```

class='changeable'>" + oldValue + "</td>");
        }
    });

    }
    isInput = false;
    addChangeOnTable();

    event.preventDefault();

    console.log("-----");

});

$("#declineUpdate").click(function() {

    messageLabel.empty();

    $(currentCell).replaceWith("<td class='changeable'>" + oldValue +
"</td>");

    isInput = false;
    addChangeOnTable();
});

});

}

function addDeleteOnTable() {
    $(".delete").click(function() {

        messageLabel.empty();

        if(isInput) return;

        let row = $(this).parent();
        let id = $(row).find("td").eq(0).html();

        $.ajax({
            async: false,
            url: "../php/delete.php",
            type: "POST",
            dataType: "html",
            data: {
                id: id,
            },
            beforeSend: () => console.log("deleting..."),
            success: (data) => {
                if(data === "1") row.remove();
                else messageLabel.empty().append("Удаление: невозможно удалить
запись " + id + ".");
            }
        });

        console.log("-----");
    });
}

```



```

    });
}

function addCreateOnTable() {
    isInput = false;

    $(".create").click(function () {

        messageLabel.empty();

        if (isInput) return;

        let table = document.getElementById("table");

        let createRow = table.insertRow(1);

        $(createRow).append(
            "<td><label><input form='createForm' id='createId' type='text' " +
            "required='required' minlength='6' maxlength='6' pattern='[A-Z]{2}[0-9]{4}' " +
            "placeholder='AA0000' value='AA0000'></label></td>" +
            "<td><label><select " +
            "id='createDepartureCity'></select></label></td>" +
            "<td><label><select " +
            "id='createArrivalCity'></select></label></td>" +
            "<td><label><input form='createForm' " +
            "id='createDepartureDateAndTime' type='datetime-local' required='required' " +
            "value='2021-08-13T10:00'></label></td>" +
            "<td><label><input form='createForm' id='createFlightDuration' " +
            "type='number' required='required' min='1' max='9999' step='1' " +
            "value='1'></label></td>" +
            "<td><label><input form='createForm' id='createClass' " +
            "type='text' required='required' value='Первый'></label></td>" +
            "<td><label><select " +
            "id='createAirplaneId'></select></label></td>" +
            "<td><label><input form='createForm' id='createPrice' " +
            "type='number' required='required' min='1' max='9999' step='any' " +
            "value='1.00'></label></td>" +
            "<td><form id='createForm' style='display: inline- " +
            "block'><button type='submit' id='acceptCreation'>Да</button></form><span " +
            "id='declineCreation'>Нет</span></td>"
        );

        fillCitySelector("#createDepartureCity", "");
        fillCitySelector("#createArrivalCity", "");
        fillAirplaneSelector("#createAirplaneId", "");

        isInput = true;

        $("#createForm").on("submit", function(event) {

            messageLabel.empty();

            let idInput = document.getElementById("createId");
            let id = idInput.value;

```

```

        let departureCitySelector =
document.getElementById("createDepartureCity");
        let departureCityId =
cityId[departureCitySelector.selectedIndex];
        let departureCityName =
departureCitySelector.options[departureCitySelector.selectedIndex].value;

        let arrivalCitySelector =
document.getElementById("createArrivalCity");
        let arrivalCityId = cityId[arrivalCitySelector.selectedIndex];
        let arrivalCityName =
arrivalCitySelector.options[arrivalCitySelector.selectedIndex].value;

        let departureDateAndTime =
document.getElementById("createDepartureDateAndTime").value;
        let flightDuration =
document.getElementById("createFlightDuration").value;
        let _class = document.getElementById("createClass").value;
        let airplaneId =
document.getElementById("createAirplaneId").value;
        let price = document.getElementById("createPrice").value;

        if(departureCityId === arrivalCityId) {
            messageLabel.empty().append("Создание: город отбытия(" +
departureCityName + ") и город прибытия(" + arrivalCityName + ") не могут быть
одинаковыми.");
            event.preventDefault();
            return;
        }

$.ajax({
    async: false,
    url: "../php/create.php",
    type: "POST",
    dataType: "html",
    data: {
        id: id,
        departureCityId: departureCityId,
        arrivalCityId: arrivalCityId,
        departureDateAndTime: departureDateAndTime,
        flightDuration: flightDuration,
        class: _class,
        airplaneId: airplaneId,
        price: price
    },
    beforeSend: () => console.log("creating..."),
    success: (data) => {
        console.log(data);
        if(data === "1") {
            $(createRow).replaceWith("<tr>" +
"<td>" + id + "</td>" +
"<td class='changeable'>" + departureCityName +
"</td>" +
"<td class='changeable'>" + arrivalCityName +

```

```

"</td>" +
                                "<td class='changeable'>" +
departureDateAndTime.replace("T", " ") + ":00" + "</td>" +
                                "<td class='changeable'>" + flightDuration +
"</td>" +
                                "<td class='changeable'>" + _class + "</td>" +
                                "<td class='changeable'>" + airplaneId +
"</td>" +
                                "<td class='changeable'>" +
(Number.isInteger(price) ? (price + ".00") : price) + "</td>" +
                                "<td class='delete'>-</td>" +
                                "</tr>");
                                isInput = false;
                                addDeleteOnTable();
                                addChangeOnTable();
                                } else {
                                messageLabel.empty().append("Создание: запись с
кодом " + id + " уже существует.");
                                event.preventDefault();
                                }
                                }
                                });

                                event.preventDefault();

                                console.log("-----");

                                });

                                $("#declineCreation").click(function () {

                                messageLabel.empty();

                                createRow.remove();
                                isInput = false;

                                });

                                });
                                }

function fillCitySelector(selectorId, titleText) {
    cityId = [];
    $.ajax({
        async: false,
        url: "../php/fillCitySelector.php",
        type: "POST",
        dataType: "html",
        beforeSend: () => console.log("filling city selector..."),
        success: (data) => {
            let result = "";
            if(!(titleText === "")) result = "<option class='title' disabled
selected style='display: none'>" + titleText + "</option>";
            $.each(JSON.parse(data), (index, item) => {
                let city = new City(item['id'], item['name']);

```

```

        result += "<option>" + city.getName() + "</option>";
        cityId.push(city.getId());
    })
    $(selectorId).empty().append(result);
}
});

console.log("-----");

}

function fillAirplaneSelector(selectorId, titleText) {
    $.ajax({
        async: false,
        url: "../php/fillAirplaneSelector.php",
        type: "POST",
        dataType: "html",
        beforeSend: () => console.log("filling airplane selector..."),
        success: (data) => {
            let result = "";
            if(!(titleText === "")) result = "<option class='title' disabled
selected style='display: none'>" + titleText + "</option>";
            $(selectorId).empty().append(result + data);
        }
    });

    console.log("-----");

}

});

```