

Прикарпатський національний університет імені Василя Стефаника

Фізико-технічний факультет

Кафедра комп'ютерної інженерії та електроніки

Демчук Любомир Сергійович

Liubomyr Demchuk

УДК 004:42

Спеціальність 123 «Комп'ютерна інженерія»

Кваліфікаційна робота

на здобуття освітнього ступеня бакалавра

Інформаційна система пошуку та купівлі квитків на різні види транспорту
Information system for searching and buying tickets for various types of transport

Науковий керівник:

К. т.н., доцент

комп'ютерної інженерії та

електроніки, Володимир ГРИГА

Рецензент:

К.ф.-м.н, професор

кафедри фізики і хімії твердого

тіла, Любомир НИКИРУЙ

Івано-Франківськ

2024

Формат	Поз.	Позначення	Найменування	К-ть	Прим.
A4			Загальна блок-схема роботи Android-додатку.	1	
A4			Блок -схема підключення клієнта до сервера за допомогою Android Spring Library	1	
A4			Пояснювальна записка	118	

			123.КІ-41.04			
<i>Змн.</i>	<i>Арк.</i>	<i>№ докум.</i>	<i>Підпис</i>	<i>Дата</i>		
<i>Розробив</i>		Демчук Л.С.			<i>Літ.</i>	<i>Арк.</i>
<i>Перевірів</i>		Грига В.М.				<i>Аркушів</i>
					Специфікація	
<i>Н. Контр.</i>						
<i>Затвердив</i>						

АНОТАЦІЯ

Пояснювальна записка містить: 118 сторінку, 26 рисунків, 14 таблиць та 14 посилань.

Об'єкт розробки - створення програмної системи для продажу залізничних та автобусних квитків.

Реалізована система має клієнт-серверну архітектуру і побудована за допомогою паттерну програмування MVC. Серверна частина реалізована у вигляді веб-сервера TomCat Server Local, сервер прикладної логіки - Spring Web MVC, який надає засоби для доступу до серверної бази даних PostgreSQL. Клієнтська частина реалізована у вигляді прикладної підсистеми для мобільного обчислювального пристрою на платформі ОС Android і забезпечує продаж автобусних та залізничних квитків. Забезпечено можливість адаптації зміни даних у базі даних системи. Реалізовано можливість генерації оплаченого квитка у вигляді QR-коду. Розроблено архітектуру системи, діаграми класів, алгоритми роботи системи та інтерфейс користувача, як клієнтської частини так і серверної.

Ключові слова:

ANDROID-ДОДАТОК, КЛІЄНТ-СЕРВЕРНА АРХІТЕКТУРА, QR-КОД, АВТОБУСНІ ТА ЗАЛІЗНИЧНІ КВИТКИ, АЛГОРИТМ.

					123.КІ-41.04			
<i>Змн.</i>	<i>Арк.</i>	<i>№ докум.</i>	<i>Підпис</i>	<i>Дата</i>				
Розробив		Демчук Л.С.			Анотація	<i>Літ.</i>	<i>Арк.</i>	<i>Аркуші</i>
Перевірив		Грига В.М.					3	1
Н. Контр.								
Затвердив								

ABSTRACT

The explanatory note contains: 118 pages, 26 figures, 14 tables and 14 references
The object of development is the creation of a software system for the sale of railway and bus tickets.

The implemented system has a client-server architecture and is created using MVC template programming. The server part is implemented in the form of a web server TomCat Server Local, an application logic server - Spring Web MVC, which provides tools for accessing the server database PostgreSQL. The client part is implemented in the example of the subsystem for a mobile computing device on the OC Android platform and provides the sale of bus and train tickets. It is possible to adapt data changes in the system database. The possibility of generating a paid ticket in the form of a QR-code has been implemented. The system architecture, class diagrams, system operation algorithms and user interface, both client and server, have been developed.

Keywords:

ANDROID APPENDIX, CLIENT-SERVER ARCHITECTURE, QR CODE, BUS AND TRAIN TICKETS, ALGORITHM.

					123.KI-41.04			
<i>Змн.</i>	<i>Арк.</i>	<i>№ докум.</i>	<i>Підпис</i>	<i>Дата</i>				
Розробив		Демчук Л.С.			Abstract	<i>Лім.</i>	<i>Арк.</i>	<i>Аркушіє</i>
Перевірів		Грига В.М.					4	1
Н. Контр.								
Затвердив								

ПЕРЕЛІК ОСНОВНИХ СКОРОЧЕНЬ

ІС – Інформаційна система

БД – База даних

UI – User Interface (Користувацький інтерфейс)

UX – User Experience (Користувацький досвід)

API – Application Programming Interface (Інтерфейс прикладного програмування)

HTTP – HyperText Transfer Protocol (Протокол передачі гіпертексту)

HTTPS – HyperText Transfer Protocol Secure (Захищений протокол передачі гіпертексту)

XML – Extensible Markup Language (Розширювана мова розмітки)

JSON – JavaScript Object Notation (Нотація об'єктів JavaScript)

SQL – Structured Query Language (Мова структурованих запитів)

GUI – Graphical User Interface (Графічний інтерфейс користувача)

CRUD – Create, Read, Update, Delete (Створення, Читання, Оновлення, Видалення)

MVP – Minimum Viable Product (Мінімально життєздатний продукт)

SaaS – Software as a Service (Програмне забезпечення як послуга)

GPS – Global Positioning System (Глобальна система позиціонування)

RFID – Radio Frequency Identification (Радіочастотна ідентифікація)

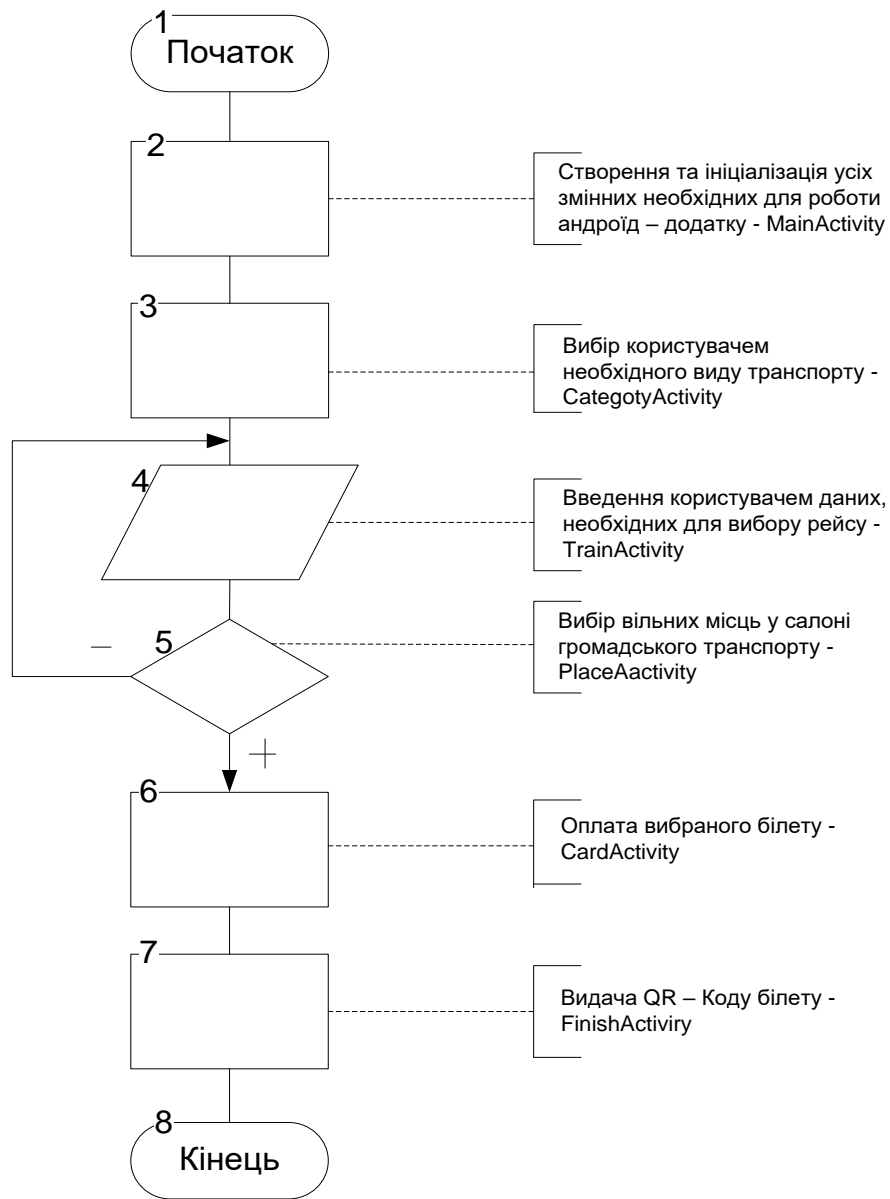
SSL – Secure Sockets Layer (Рівень захищених сокетів)

TSL – Transport Layer Security (Транспортний рівень безпеки)

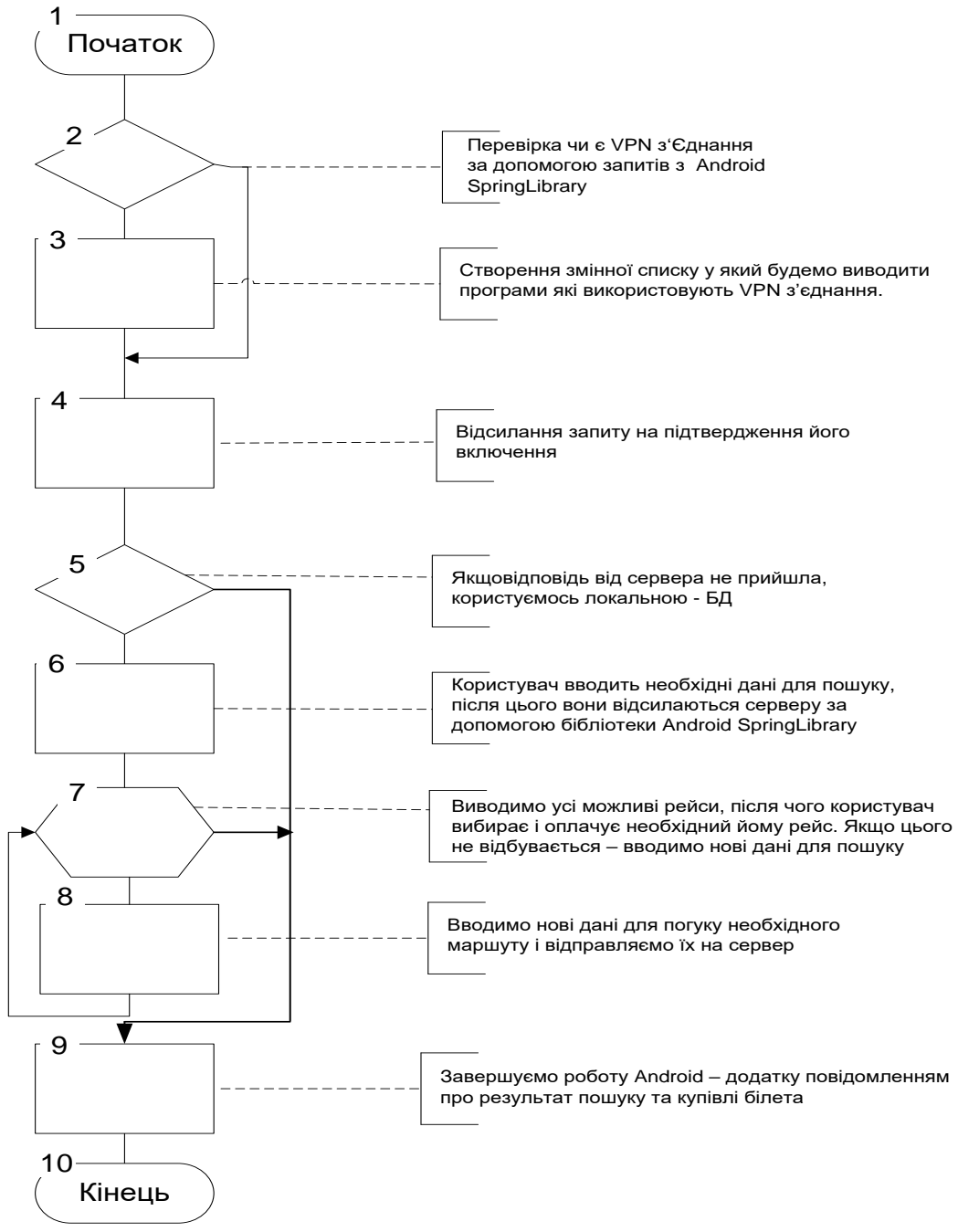
OTP – One-Time Password (Одноразовий пароль)

PNR – Passenger Name Record (Запис імені пасажирів)

					123.КІ-41.04			
<i>Змн.</i>	<i>Арк.</i>	<i>№ докум.</i>	<i>Підпис</i>	<i>Дата</i>				
Розробив		Демчук Л.С.			Abstract	<i>Літ.</i>	<i>Арк.</i>	<i>Аркуші</i>
Перевірив		Грига В.М.					4	1
Н. Контр.								
Затвердив								



					123.КІ-41.04		
Змн.	Арк.	№ докум.	Підпис	Дата			
Розробив		Демчук Л.С.			Літ.	Арк.	Аркушіє
Перевірив		Грига В.М.				5	1
Н. Контр.					Загальна блок-схема роботи Android-додатку		
Затвердив							



					123.КІ-41.04			
<i>Змн.</i>	<i>Арк.</i>	<i>№ докум.</i>	<i>Підпис</i>	<i>Дата</i>				
Розробив		Демчук Л.С.			Блок -схема підключення клієнта до сервера за допомогою Android Spring Library	<i>Літ.</i>	<i>Арк.</i>	<i>Аркушіє</i>
Перевірив		Грига В.М.					6	1
Н. Контр.								
Затвердив								

Пояснювальна записка
до кваліфікаційної роботи

на тему:

**«Інформаційна система пошуку та купівлі квитків на різні види
транспорту»**

					123.KI-41.04			
<i>Змн.</i>	<i>Арк.</i>	<i>№ докум.</i>	<i>Підпис</i>	<i>Дата</i>				
Розробив		Демчук Л.С.			Пояснювальна записка	<i>Літ.</i>	<i>Арк.</i>	<i>Аркушів</i>
Перевірив		Грига В.М.					7	118
Н. Контр.								
Затвердив								

ЗМІСТ

ВСТУП.....	4
РОЗДІЛ 1 ОГЛЯД ТА АНАЛІЗ ПРОГРАМНИХ СИСТЕМ ПОШУКУ ТА КУПІВЛІ КВИТКІВ НА РІЗНІ ВИДИ ТРАНСПОРТУ.....	6
1.1. Актуальність розроблення та впровадження програмних систем для продажу квитків на різні види транспорту.....	6
1.2. Огляд відомих програмних систем з продажу квитків на різні види транспорту ю.....	7
1.2.1. Інтернет-сервіс Booking.com.....	7
1.2.2. Інтернет-сервіс Busfor.ua	10
1.2.3. Інтернет-сервіс Tickets.ua.....	11
1.2.4. Інтернет-сервіс "Приват24".....	14
1.2.5. Інтернет-сервіс Укрзалізниці.....	15
1.3. Порівняльний аналіз відомих систем.....	17
1.4. Постановка задачі.....	19
РОЗДІЛ 2 ВИБІР ЗАСОБІВ РОЗРОБКИ ДЛЯ РЕАЛІЗАЦІЇ ПРОГРАМНОЇ СИСТЕМИ ПОШУКУ ДЛЯ ПРОДАЖУ КВИТКІВ.....	20
2.1 Обґрунтування вибору ОС Android.....	21
2.1.1 Android Studio	27
2.1.2 Android SDK та Android Virtual Device (AVD).....	28
2.1.3 Фрагментація	31
2.1.4 Особливості Android M та Android N. Material Design.....	36
2.1.5 Вибір апаратних засобів.....	38
2.2. Вибір засобів для реалізації серверної частини на основі REST-архітектури.....	39
2.2.1 IntelliJ IDEA.....	39
2.2.2 Apache Tomcat 9.0.....	40
РОЗДІЛ 3 РОЗРОБКА ПРОГРАМНОЇ СИСТЕМИ ПОШУКУ ДЛЯ ПРОДАЖУ КВИТКІВ НА РІЗНІ ВИДИ ТРАНСПОРТУ.....	42

3.1 Реалізація клієнтської частини спеціалізованої системи продажу білетів.....	42
3.1.1 Розробка основних функціональних вузлів Android – додатку....	42
3.1.2 Розробка графічного інтерфейсу користувача.....	49
3.2 Реалізація серверної частини спеціалізованої системи продажу білетів.....	57
3.2.1 Розробка REST – full сервісу	57
3.2.2 Розробка бази даних для REST – full сервісу.....	60
РОЗДІЛ 4. ОХОРОНА ПРАЦІ.....	66
4.1 Значення охорони праці для забезпечення безпечних умов праці.....	66
4.2 Правила охорони праці під час експлуатації ЕОМ.....	67
4.3 Забезпечення нормальних умов праці.....	69
4.4 Система електроживлення та заходи з електробезпеки.....	75
4.5 Пожежна безпека.....	77
РОЗДІЛ 5. ЕКОНОМІЧНА ЧАСТИНА.....	80
5.1 Розрахунок часу на виготовлення програмного продукту.....	80
5.2 Економічне обґрунтування розробки та впровадження проектного рішення.....	82
5.2.1 Розрахунок витрат на розробку та впровадження проектного рішення.....	83
5.2.2 Визначення експлуатаційних витрат.....	90
5.2.3 Розрахунок ціни споживання проектного рішення.....	92
5.2.4 Визначення показників економічної ефективності.....	93
ВИСНОВКИ.....	96
ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ.....	97
ДОДАТКИ.....	99

ВСТУП

Інформаційна система пошуку та купівлі квитків на різні види транспорту представляє собою – розробку удосконаленої інформаційної системи для продажу квитків на різні види транспорту на основі ОС Android. На сьогоднішній день, коли смартфони стали невід’ємною частиною будь-якої людини, вимоги до них кардинально виросли.

Операційна система Google Android є однією із найпопулярніших ОС на ринку інформаційних послуг, тому розробка системи для продажу білетів саме на цій платформі є виправданою, так як дана система має бути доступною широкому колу споживачів. Крім того система є розподіленою, а Android дозволяє легко взаємодіяти з віддаленими сервісами і базами даних [1].

Електронний квиток можна самостійно купити в мережі Інтернет. Переваги інтернет-купівлі квитка полягають в тому, що особа, яка бронює квиток отримує повну інформацію про всі можливі варіанти маршруту вибраного напрямку або запропоновані маршрути з пересадками, якщо немає прямого сполучення, запропонованих системою, а також бачить структуру утворення ціни квитка, із зазначенням плати за вартість квитка та додаткові сервіси. Покупка здійснюється набагато швидше стандартної покупки через каси. Купівля та оплата електронного квитка здійснюється на сайтах, які підключені до ресурсів продажу квитків. Для оплати приймається будь-яка платіжна банківська картка. Підтвердження про покупку надсилається на електронну адресу [2,3].

Актуальність розробки інформаційних програмних систем для продажу електронних квитків на проїзд в транспортних засобах має велике практичне застосування та може користуватися значним попитом серед клієнтів. Оскільки на сьогоднішній день більша половина смартфонів використовує операційну систему Android і розробка спеціалізованої програмної системи під дану платформу знайде чимало користувачів та широкий ряд застосувань. Розроблена серверна частина може бути застосованою у web-технологіях та мати підтримку в інших рішеннях. Дана спеціалізована система може розглядатись як повноцінний комерційний додаток і бути завантаженою у PlayMarket з певним цінником.

					123.KI-41.04	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		4

Основною метою дипломного проектування є реалізація інформаційної програмної системи для продажу автобусних та залізничних квитків на основі ОС Android з використанням серверної бази даних PostgreSQL.

					<i>123.KI-41.04</i>	<i>Арк.</i>
<i>Зм.</i>	<i>Арк.</i>	<i>№ докум.</i>	<i>Підпис</i>	<i>Дата</i>		5

1 ОГЛЯД ТА АНАЛІЗ ІНФОРМАЦІЙНИХ ПРОГРАМНИХ СИСТЕМ ДЛЯ КУПІВЛІ КВИТКІВ НА РІЗНІ ВИДИ ТРАНСПОРТУ

1.1. Актуальність розроблення та впровадження програмних систем для продажу квитків на різні види транспорту.

Актуальність розробки та впровадження програмної системи продажу квитків на різні види транспорту дуже висока. Люди намагаються зробити своє життя максимально комфортним в усіх сферах, а також у транспорті та подорожах. Система продажу та бронювання квитків через мобільний додаток вирішує багато проблем, а саме:

- просту покупку квитка через мобільний додаток;
- оплата за допомогою онлайн-банків;
- перегляд розкладу руху транспорту в електронному варіанті;
- перегляду кількості вільних місць в транспорті;
- збереження екології за допомогою зменшення паперових квитків;
- непотрібно стояти в живій черзі;
- змога зменшити кількість працівників на касо обслуговуванні.

Цей мобільний додаток простий у використанні. Простий інтерфейс і безліч порад допоможуть користувачам зареєструватися і використовувати його на свій смак. У майбутньому додаток зможе об'єднати всі види транспорту, створивши так звану мережу. Це дозволить вам шукати найвигідніший маршрут і вибрати вартість і клас поїздки, забронювати місце та оплатити, протягнувши картку, замість того, щоб використовувати QR-код, який ви зчитували, коли заходили в транспортний засіб замість квитка [4].

Користувач може придбати електронні квитки онлайн самостійно. Перевага покупки квитків онлайн полягає в тому, що якщо система не забезпечує пр'яме з'єднання, замовник отримує повну інформацію про всі можливі варіанти маршруту або запропоновані маршрути пересадки в обраному напрямку, а також бачить структуру тарифу, вказуючи вартість проїзду та комісію за додаткові

					123.KI-41.04	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		6

послуги. Купівля через касу відбувається набагато швидше, ніж стандартна покупка.

Купівля та оплата електронних квитків здійснюється на веб-сайтах, які посилаються на ресурси з продажу квитків. Приймає будь-які платежі банківською картою. Підтвердження покупки буде надіслано на електронну адресу.

1.2. Огляд відомих програмних систем з продажу квитків на різні види транспорту

1.2.1 Інтернет-сервіс Booking.com

Booking.com – це веб-сайт, розроблений компанією Priceline для онлайн-бронювання авіаквитків та проживання [21].

Booking.com інвестує в технології, які допомагають вам подорожувати з легкістю, надаючи мільйонам гостей чудові варіанти відпочинку, транспортні послуги та неймовірне житло: від приватних будинків до готелів тощо. Як найбільша у світі туристична платформа для визнаних брендів і підприємців усіх рівнів, Booking.com допомагає власникам будинків по всьому світу залучати гостей та розвивати їхній бізнес. Сайт для платформи Booking.com пропонує наступні послуги:

- пропозиція різноманітних варіантів проведення дозвілля від постачальників туристичних послуг;
- можливість обробляти платежі, що надходять з метою покупки або оренди туристичних продуктів і послуг, якщо обраний платіжний сервіс їх підтримує;
- служба підтримки клієнтів працюють цілодобово та без вихідних;
- надаємо інформативний і зручний у використанні платформ, а також гарантуємо найкращі доступні ціни.

Всю інформація про замовлення місць квитків для туристів Booking можна знайти на офіційному сайті за посиланням з списку літератури [21].

					<i>123.KI-41.04</i>	<i>Арк.</i>
<i>Зм.</i>	<i>Арк.</i>	<i>№ докум.</i>	<i>Підпис</i>	<i>Дата</i>		7

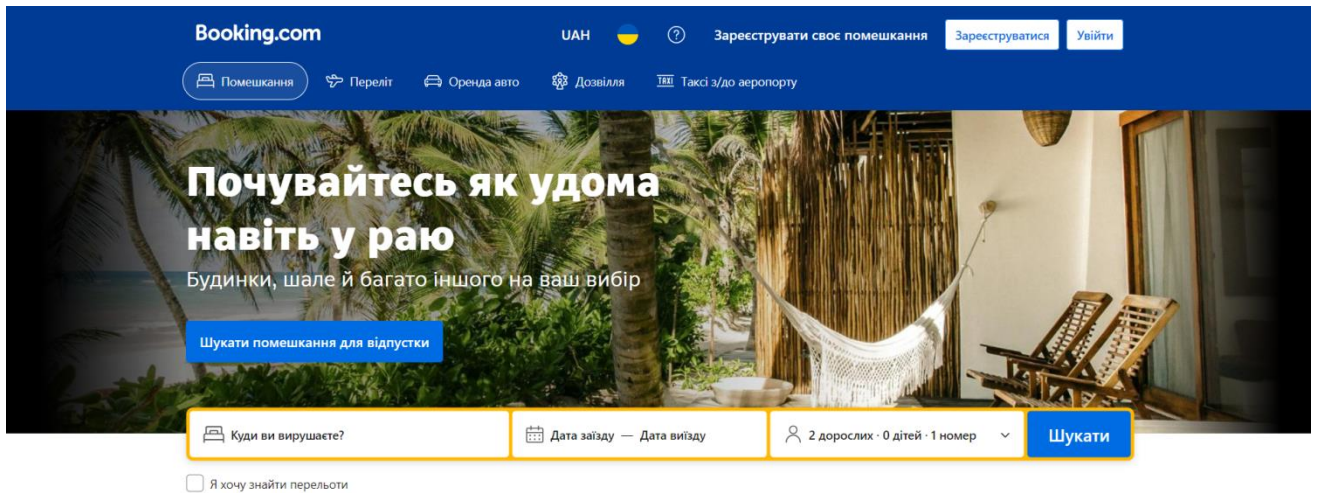


Рис. 1.1. Офіційний сайт для замовлення місць Booking.com

У розділах «Проживання», «Авіаквитки», «Прокат автомобілів», «Відпочинок» та «Таксі в аеропорт» користувачі можуть отримати всю інформацію про бронювання місць та авіаквитків. На цих сторінках розробники пропонують переглядати інформацію, надану у формі бронювання та продавати квитки на проїзд. Крім того, користувачі мають можливість завантажити інформацію у вигляді PDF-документа при купівлі авіаквитка [5].

Booking.com надає лише веб-додаток своїх продуктів, доступ до якого можна отримати в будь-якому інтернет-браузері.

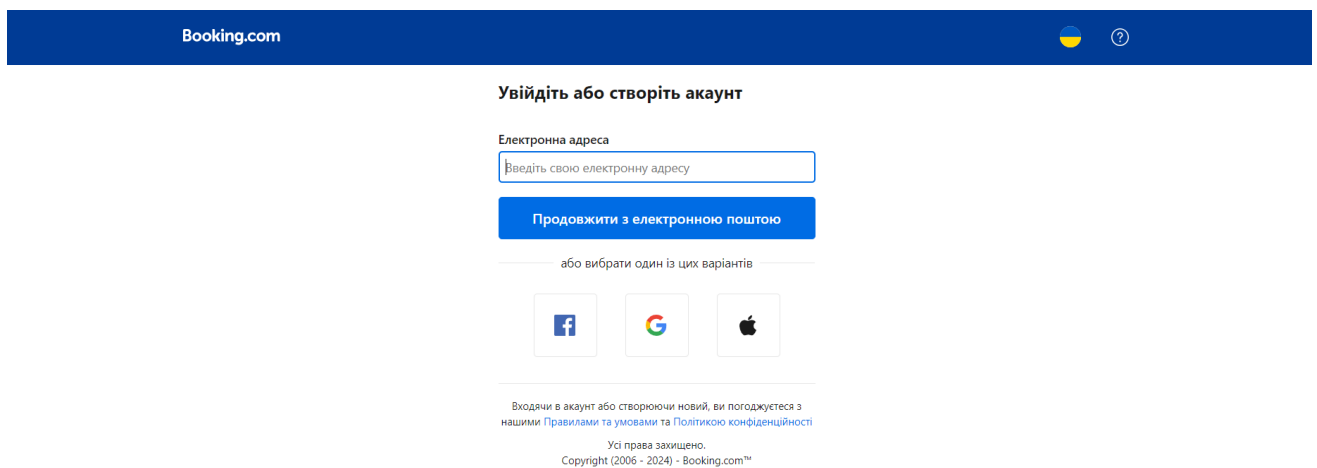


Рис.1.2 Реєстрація акаунта Booking.com.

					<i>123.KI-41.04</i>	<i>Арк.</i>
<i>Зм.</i>	<i>Арк.</i>	<i>№ докум.</i>	<i>Підпис</i>	<i>Дата</i>		8

Сайт створений у зручних для користувача відтінках сірого та жовтого, прості шрифти полегшують читання тексту, а великі елементи керування та значки роблять використання системи швидким та легким.

Немає можливості налаштувати положення або колір елементів, а також немає додаткового набору стилів дизайну.

Верхня панель містить посилання на основні сайти бронювання, а також інформацію про користувачів, які ввійшли, бронювання, відгуки та списки, центр підтримки, а також можна завантажити мобільний додаток. В робочій області відбувається робота із замовлення місць для різних видів транспорту, пошуку за типом помешкання, налаштування системи, адміністрування та управління користувачами, тощо.

На рисунку 1.3 зображено приклад бронювання квитків платформ booking.com.

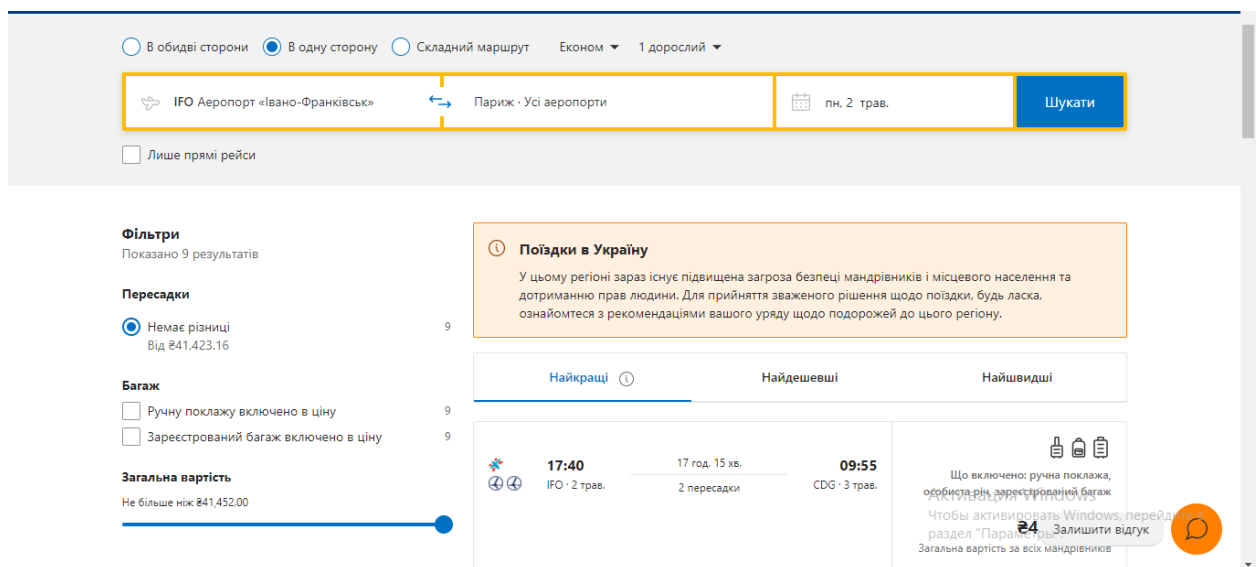


Рис.1.3 Приклад бронювання квиток на літак

Переваги платформ Booking.com:

- гарантує кращої ціни на розміщення, а також зобов'язує повернути різницю в разі виявлення клієнтом більш вигідної пропозиції;
- бронювання квитків та місць проводиться без комісії, тобто інтернет-сервіс не бере ні копійки.

Недоліки платформ Booking.com:

- незважаючи на величезну базу Booking.com, увазі інтернет-користувачів пропонуються не всі готелі в даному місті або на потрібному вам курорті.

1.2.2. Інтернет-сервіс Busfor.ua

Busfor - міжнародний онлайн-сервіс для пошуку та покупки квитків на автобуси на міжміські та міжнародні рейси [7]. Цей онлайн-сервіс є частиною однойменної холдингової компанії до 2015 року, коли був відомий як GillBus. Зараз працює в Росії, Білорусі, Україні, Польщі та Чехії. Офіси знаходяться в Москві, Києві, Варшаві. Компанія не є перевізником, а об'єднує послуги автобусних перевезень, які пропонуються на ринку.

Busfor прагне надати клієнтам легкий і зручний сервіс для пошуку та покупки дешевих автобусних квитків. В режимі онлайн користувачі можуть переглянути всі доступні варіанти подорожей та легко оплатити квитки банківською картою в Busfor, готівкою, кур'єрською службою, терміналами чи партнерськими касами.

Busfor працює на основі власної розробки – технології GDS. Це інноваційна система розподілу та продажу автобусних квитків. Оператори успішно використали цю технологію для швидкого розміщення безкоштовних місць на своїх автобусних маршрутах для продажу клієнтам онлайн. Автовокзали підключені до системи GDS, яка зацікавлена у покращенні якості своїх послуг, надаючи пасажиром можливість купувати квитки на автобус онлайн на сервісі Busfor.

Тарифи зазвичай встановлює сам перевізник. Busfor завжди намагається запропонувати клієнтам найдешевший квиток за такою ж ціною, що й автовокзал.

У результатах пошуку відображаються рейси на автобуси різних цінових категорій, серед яких клієнти можуть вибрати найдешевший квиток.

Сайт додатку Busfor можна надати доступ в будь-якому інтернет-браузері та мобільного додатку. Процес пошуку та відображення вже придбаних квитків є зручним способом:

					<i>123.KI-41.04</i>	<i>Арк.</i>
<i>Зм.</i>	<i>Арк.</i>	<i>№ докум.</i>	<i>Підпис</i>	<i>Дата</i>		10

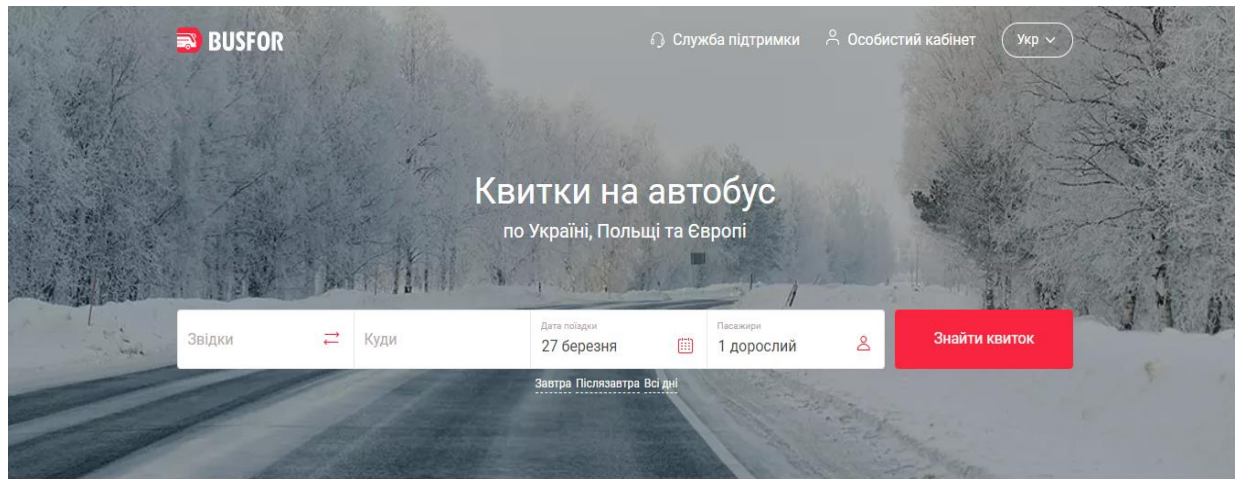


Рис.1.4 - Процес вибору маршруту продаж квитків на автобус на міжміські та міжнародні рейси за допомогою сайту Busfor.

Як можемо бачити інтерфейс доволі простий та інтуїтивно зрозумілий. Обираємо станцію відправлення, станцію прибуття, дату, час, кількість пасажирів і тиснемо на кнопку знайти квиток і знаходимо його.

Переваги продажу квитків Busfor:

- швидка покупка квитка з будь-якої точки;
- багато перевізників, бонуси, акційні пропозиції, можна бронювати конкретне місце;
- зручний мобільний додаток, електронні квитки, можливість придбати квитки онлайн;
- багато рейсів від різних перевізників, доступна вартість, можна платити картою, бонус 3% за купівлю квитків;

Недоліки продажу квитків Busfor:

- незрозумілою є ситуація з поверненням коштів за квиток.
- потрібно друкувати квиток, а в процесі перебування в дорозі це не завжди зручно робити.

1.2.3. Інтернет-сервіс Tickets.ua

Tickets.ua – український онлайн-оператор, який пропонує авіа, залізничні та автобусні електронні квитки, трансфери, квитки на події та концерти, бронювання

					<i>123.KI-41.04</i>	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		11

готелів та страхові поліси для подорожей до України та країн Європи [22]. За підсумками 2013 року Tickets.ua посів друге місце в рейтингу найбільших компаній журналу Forbes Україна.

Усі веб-сайти компанії створені в її R&D центрі у Львові. Технічну підтримку та розробку послуг забезпечує команда з понад 200 інженерів та менеджерів. Центр підтримки Холдингу доступний цілодобово і без вихідних для підтримки користувачів на всіх ринках. Допомога клієнту зробити оптимальний вибір усіх складових подорожі, дати можливість обирати серед максимальної кількості провайдерів, забезпечити прозоре порівняння цін та найкращий і доступний для кожного сервіс, піклуючись та підтримуючи користувача на кожному етапі мандрівки.

Проаналізувавши ціни перед встановленням цін, ведемо переговори з авіакомпаніями, знаходимо нових партнерів, досягаємо компромісів з усіма гравцями ринку, які формують тарифи – ми робимо все можливе, щоб мінімізувати вартість авіаквитків на наших ресурсах.

Tickets.ua – унікальна платформа, яка поєднує в собі систему бронювання, бізнес-логіку та платіжний шлюз. Щоб відповідати високим стандартам обслуговування клієнтів, ми впровадили інноваційні технологічні розробки.

Фактичні білінгові рішення дозволяють здійснювати електронні платежі через провідні банки-еквайери. Такий підхід економить витрати на фінансові операції та збільшує пропускну здатність картки.

Безпека Tickets.ua сертифікована міжнародним сертифікатом безпеки PCI DSS. Сервіс гарантує, що дані платіжних карток, які обробляються під час процесу бронювання Tickets.ua, повністю захищені.

Ліцензії Tickets.ua визнаються Міжнародною асоціацією повітряного транспорту (IATA). Її повноваження підтверджують, що агентство може продавати квитки і має всі ресурси, включаючи компетентного персоналу. Діяльність інтернет-агентства відповідає вимогам чинного законодавства України [6].

					<i>123.KI-41.04</i>	<i>Арк.</i>
<i>Зм.</i>	<i>Арк.</i>	<i>№ докум.</i>	<i>Підпис</i>	<i>Дата</i>		12

Співпраця Технологічними партнерами холдингу є лідери світової туристичної індустрії: Sabre, Amadeus, TravelPort, Sirena, ETS, національні авіалінії, національні залізниці та міжнародні автобусні оператори. Як і інші онлайн-агентства, підконтрольні ТТН, Tickets.ua створено з урахуванням особливостей кожного ринку. Ресурс постійно покращує обслуговування клієнтів і розширює портфель послуг.

Центр досліджень і розробок залучає стратегічних партнерів, що спеціалізуються на пасажирських перевезеннях та інших послугах для мандрівників, а також місцевих постачальників туристичних послуг.

Сайт Tickets.ua надає програмного продукту, доступ до якої можна отримати в будь-якому інтернет-браузері для різних країн світу. Процес пошуку квитків та відображення для зручнім та простим способом:

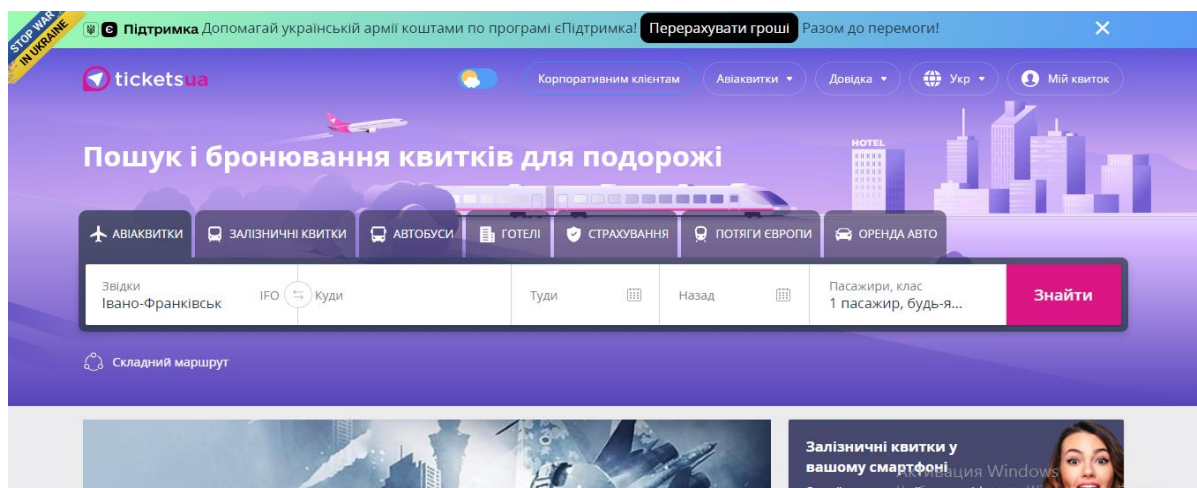


Рис.1.5 Процес вибору замовлення квитків на різні види транспорту для подорожі на сайті Tickets.ua

Як можемо бачити інтерфейс доволі простий та інтуїтивно зрозумілий. Обираємо будь-які види транспортних послуг, станцію відправлення, станцію прибуття, дату в одну сторону та дату в зворотню сторону, кількість пасажирів та тип класу натискаємо на кнопку знайти і знаходимо його.

Переваги продаж квитків Ticket.ua:

- швидке оформлення та бронювання квитків;
- швидка покупка квитка з будь-якої точки;
- відсутність можливості отримання повної інформації.

Недоліки продаж квитків Ticket.ua:

- немає повернення коштів при продажі квитків;
- потрібно друкувати квиток, а на ходу це не завжди зручно робити.

1.2.4. Інтернет-сервіс “Приват24”

«Приват24» — найпопулярніший інтернет-банкінг в Україні, який запущений з 2021 року [23]. Система дозволяє отримувати доступ та керувати платіжними картками клієнта в режимі реального часу.

Доступ до системи може бути здійснений, як через Web-версію, так і через мобільні додатки для операційних систем Android. Доступ до системи через Web-версію здійснюється за допомогою одноразових динамічних паролів, що надсилаються користувачеві через SMS.

Можливості системи додатку Приват24:

- отримання інформації щодо балансу та реквізитів своїх рахунків;
- оплата рахунків;
- оплата за постачання води, електроенергії, газу, надання послуг інтернету, зв'язку та інших;
- можливість використання API-функцій для веб-розробників;
- купівля квитків на авіа-, автобусні та залізничні перевезення, на спортивні заходи, концерти, кіно, бронювання готелів тощо;
- можливість роботи декількох користувачів;
- цілодобова підтримка онлайн-консультантами банку.

					<i>123.KI-41.04</i>	<i>Арк.</i>
<i>Зм.</i>	<i>Арк.</i>	<i>№ докум.</i>	<i>Підпис</i>	<i>Дата</i>		14

міжнародної безпеки руху та якісних залізничних перевезень, забезпечення ефективної роботи та розвитку залізничні перевезення [24].

У 2014 році компанія здійснила якісний перехід від традиційних паперових квитків до найсучасніших цифрових систем продажу та перевірки (до речі, значно пізніше європейських аналогів).

Тепер, якщо вам потрібно купити квиток, ви можете це зробити двома способами, перший спосіб — піти на місцевий вокзал, стати в чергу і отримати паперовий квиток, який потім не пропаде. Погодьтеся, що цей варіант не дуже привабливий, особливо для тих, у кого не має вільного часу через складний робочий графік, чи навіть часу на зустрічі з друзями, а не цілою епопеєю.

Другий спосіб – встановити на смартфон офіційний мобільний додаток, вибрати потрібний маршрут, заповнити пункти персональних даних (ім'я, номер телефону тощо), оплатити банківську картку через один із онлайн-сервісів та отримати свій пропуск Квитки електронною поштою. Сідайте на потяг. Тобто отриманий поштою електронний квиток вже є підставою для посадки.

Другий варіант більш привабливий. Заощаджуючи час і гроші, сервіс також економить папір і, отже, природні ресурси. Процес пошуку квитків і відображення придбаних квитків зручний і простий.

На рис 1.7 показано процес вибору маршруту та купівлі квитка на поїзд з використанням сайту Укрзалізниці.

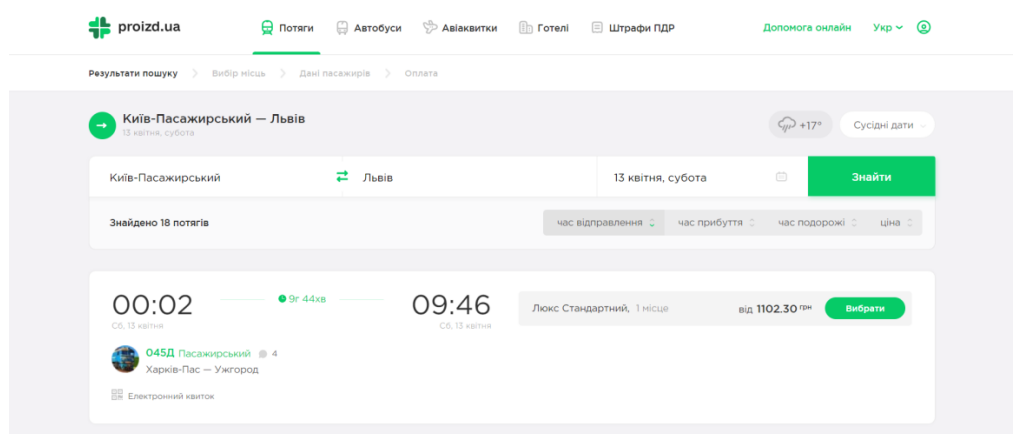


Рис.1.7 Процес вибору маршруту та купівлі квитка на поїзд з використанням сайту Укрзалізниці

										Арк.
										16
Зм.	Арк.	№ докум.	Підпис	Дата						

Як бачимо інтерфейс доволі простий та інтуїтивно зрозумілий. Обираємо станцію відправлення, станцію прибуття та час, тиснемо на кнопку знайти потяг і система автоматично знаходить його.

Переваги продажу квитків з використанням платформи Укрзалізниці:

- зручний та зрозумілий інтерфейс;
- зручний спосіб для перегляду придбаних квитків;
- електронний квиток навіть не обов'язково друкувати, при посадці достатньо пред'явити посвідчення особи і QR-код на екрані мобільного пристрою і провідник справить посадку на потяг;
- для купівлі електронного квитка не потрібна каса;
- надійність системи;
- система перевірки квитків є закритим корпоративним продуктом товариства «Укрзаліниця», тому її неможливо використовувати для створення власних квитків.

Недоліки продажу квитків з використанням платформи Укрзалізниці:

- паперовий квиток, який забезпечений засобами захисту від підробок (наприклад: папір з водяними знаками);
- перевіряючий квиток не завжди потрібно сканувати – потребують паперовий.

1.3. Порівняльний аналіз відомих систем

Практично всі сучасні сайти з продажу квитків на різні види транспорту пропонують приблизно однакові послуги, однак при детальному аналізі їх все-таки можна порівняти за рівнем поставлених завдань.

Попрацювавши з кожним з розглянутих веб-сайтів було виявлено деякі відмінності у зручності користування та швидкості реакції на необхідні користувачу операції (див. таблиця 1.1).

					123.KI-41.04	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		17

Таблиця 1.1 Порівняння розглянутих платформ

Функції	Voking.com	Приват 24	Укрзалізниця	Busform	Tickets.ua
Кросплатформність	+	+	+	+	+
Кросбраузерність	+	+	+	+	+
Режим Offline	-	-	+	-	+
Використання фреймворків при розробці	-	-	+	+	+
Використання баз даних	+	+	+	+	+
Використання АРІ	+	-	+	+	+
Можливість завантаження файлів на диск	+	+	+	+	+
Наявність ліцензії	Безкошт.	Безкошт.	Безкошт.	Безкошт.	Безкошт.
Країна	Нідерланди	Україна	Україна	Польща	Україна
Тип платформи	Система бронювання житла	Електронні платежі	Акціонерне товариство	Публічна компанія	Електронні квитки
Власник	The Priceline Group	Приватбанк	Кабінет Міністрів України	Gill Business Systes	Генеральний директор

Якщо немає труднощів з доступом до мережі Інтернет то всі із розглянутих платформ з продажу квитків однаково справляються з відтворенням різного роду композицій або файлів. Значною перевагою Booking.com, Приват24, Укрзалізниця, Busfor та Tickets.ua є можливість завантаження файлів із серверів на диск.

1.4. Постановка завдання

Головним завданням дипломного проектування є розроблення клієнт-серверної системи з продажу автобусних та залізничних квитків за допомогою програмної платформи під назвою ОС Android з можливостями відтворення, збереження, додавання інформації.

Для цього необхідно виконати наступні завдання:

- проаналізувати існуючі веб додатки;
- оволодіти навиками використання ОС Android;
- розробити структурну модель веб-додатку;
- розробити оригінальний дизайн веб-додатку;
- створити зручні і корисні функції для користувачів;
- розробити графічну структуру сайту;
- розробити схему розташування навігації та інформації на сторінках веб-додатку;
- проаналізувати всі можливі варіанти інформаційного наповнення веб-додатку;
- веб-додаток повинен сприяти виконанню наступних завдань;
- мати можливість відтворення продаж квитків, які за потребою завантажуватимуться на веб-сервер;
- містити всі необхідні дані про розробників веб-додатку.

РОЗДІЛ 2. ВИБІР ЗАСОБІВ РОЗРОБКИ ДЛЯ РЕАЛІЗАЦІЇ ПРОГРАМНОЇ СИСТЕМИ ДЛЯ ПРОДАЖУ КВИТКІВ

У зв'язку з розвитком мобільних пристроїв частина навантаження ПК перемістилася на них. В результаті вдосконалюються спеціалізовані операційні системи та засоби розробки додатків для них.

На сьогоднішній день існують три тенденції, що швидко розвиваються в операційних системах (ОС) для мобільних пристроїв. Перший – це Windows Phone 10, операційна система Microsoft, яка ще не набула широкого поширення. Друга — iOS, створена Apple для своїх пристроїв. Третя система — Android, розроблена Google. Система охоплює всі цінові категорії мобільних пристроїв, від бюджетних телефонів до дорогих планшетів. Тому мова йде про Android, як про операційну систему, яка стала найбільш поширеною [7-9].

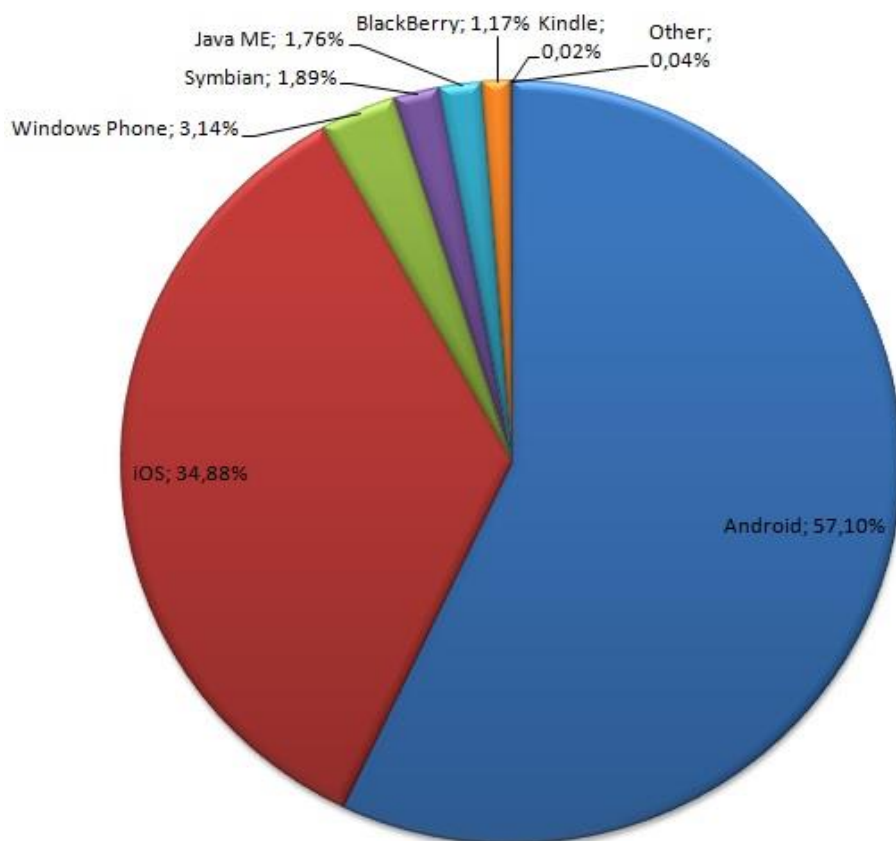


Рис. 2.1 Розподіл ОС на мобільних пристроях

2.1 Обґрунтування вибору ОС Android

Android — це операційна система для смартфонів, планшетів, електронних книг, цифрових плеєрів, годинників, ігрових приставок, нетбуків, смартбуків, Google Glass, телевізорів та інших пристроїв. У майбутньому планується підтримка автомобілів і домашніх роботів. На основі ядра Linux та власної реалізації віртуальної машини Java від Google. Спочатку розроблена компанією Android Inc., а пізніше придбана Google. Пізніше Google запустив Open Handset Alliance (ОНА) і зараз підтримує та розвиває платформу.

Android дозволяє створювати програми Java для керування пристроєм за допомогою бібліотек, розроблених Google. Android Native Development Kit дозволяє імпортувати (але не налагоджувати) бібліотеки та компоненти програми, написані на C та іншими мовами.

86% смартфонів, проданих у другому кварталі 2014 року, мали встановлену операційну систему Android. У 2014 році було продано понад 1 мільярд пристроїв Android [10-14].

Програми для Android — це програми для віртуальної машини Dalvik. На відміну від більшості віртуальних машин (наприклад, віртуальної машини Java), які орієнтовані на стек, Dalvik орієнтований на реєстр і не може бути названий стандартним рішенням. Система добре підходить для процесорів RISC, включаючи процесори ARM, що використовуються в мобільних пристроях. Dalvik розроблено для платформи Android. Віртуальні машини оптимізовані для економного споживання пам'яті та роботи на мобільному обладнанні.

Починаючи з Android 2.2, Dalvik використовує компіляцію JIT (just-in-time). Завдяки цим можливостям виходить продуктивна віртуальна машина, яка не може вплинути на роботу всього додатка. Dalvik використовує власний байт-код. При розробці програми для Android вихідний код програми перекладається компілятором у спеціальний машинно-незалежний низькорівневий код. Результатом є виконуваний пакет Android (APK).

					<i>123.KI-41.04</i>	<i>Арк.</i>
<i>Зм.</i>	<i>Арк.</i>	<i>№ докум.</i>	<i>Підпис</i>	<i>Дата</i>		21

Програмісти, які розробили програмне забезпечення для настільних комп'ютерів, спочатку рідко розробляють програмне забезпечення для Android. API Android вимагає від користувачів бути хорошими програмістами. Абстракції Android є більш цінними (роздуті, надмірно складні). Слід враховувати принципи роботи мобільних пристроїв та обмеження обчислювальних ресурсів: обмежена продуктивність процесора, обмежена оперативна пам'ять, відсутність файлу підкачки, пристрій працює від акумулятора з обмеженою потужністю.

У липні 2005 року Google придбала Android. 5 листопада 2007 року компанія офіційно оголосила про створення Open Handset Alliance (ОНА) та оголосила про відкриття мобільної платформи Android, а 12 листопада 2007 року альянс запустив першу версію Android Early Look SDK та Емулятор Android.

23 вересня 2008 року була офіційно випущена перша версія операційної системи разом із першим повним пакетом SDK 1.0 розробника, випуск 1. З моменту випуску першої версії платформи було кілька системних оновлень. Зазвичай ці оновлення стосуються виправлення помилок та додавання нових функцій до системи.

У 2009 році було випущено до чотирьох оновлень платформи. Тому в лютому була випущена версія 1.1 з різними виправленнями помилок. У квітні та вересні вийшли ще два оновлення — 1.5 «Кексики» та 1.6 «Пончики».

Оновлення Cupcake вносить серйозні зміни: віртуальна клавіатура, відтворення та запис відео, браузер тощо. Donut є першим, хто підтримує різні дозволи та щільності екрана, а також мережі CDMA. У жовтні того ж року вийшла версія Android 2.0 «Eclair», яка підтримує кілька облікових записів Google і підтримує HTML5 та інші нововведення. Після невеликого оновлення у версії «Eclair» (2.1) з'явилися «живі шпалери» та заблокований екран був змінений.

У середині 2010 року Google випустила Android 2.2 під назвою «Froyo», а наприкінці 2010 року — Android 2.3 «Gingerbread». Після оновлення Froyo можна використовувати смартфон як точку доступу, використовуючи традиційний смартфон або зміни, як-от алфавітно-цифрову систему блокування пароля, в той час як оновлення Gingerbread надає більше контролю над копіюванням і вставкою,

					123.KI-41.04	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		22

покращує керування живленням та програму керування, підтримка багатокамерного пристрою тощо. 22 лютого 2011 року була офіційно представлена платформа Android 3.0, орієнтована на Інтернет-планшет «Сті».

Випущена 19 жовтня 2011 року, Android 4.0 Ice Cream Sandwich стала першою універсальною платформою, розробленою для планшетів і смартфонів. Оновлення також представляє новий інтерфейс Holo, який використовувався до Android 4.4.4 Kitkat і був замінений на Material Design.

У червні 2012 року було випущено оновлення під назвою «Jelly Bean» із серійним номером 4.1, який було змінено на 4.2 через незначне оновлення в кінці жовтня та 4.3 після оновлення в липні 2013 року. 31 жовтня 2013 Google випускає наступну версію Android 4.4 під назвою Chocolate Bar «KitKat» за угодою з виробником Nestle. KitKat вперше з'явився на Nexus 5; ця версія Android оптимізована для роботи на більш широкому діапазоні пристроїв з мінімальним рекомендованим обсягом оперативної пам'яті 512 МБ. 25 червня 2014 року Google представив Android L, який тепер доступний для розробників.

У 2014 році була випущена операційна система для носимих пристроїв Android Wear. У Google I/O також є версії Android Auto (для автомобілів) і Android TV (для телевізорів), тому Android більше не є ОС лише для мобільних пристроїв. 15 жовтня 2014 року було офіційно випущено Android 5.0 Lollipop. Основним оновленням системи є новий дизайн. Крім того, якщо пристрій Android має код доступу або графічний ключ, а годинник користувача з Android Wear знаходиться поруч, пристрій автоматично розблокується. 9 грудня 2014 року Google замінив офіційне середовище розробки на основі Eclipse (adt-bundle) на Android Studio.

Додаток операційної системи Android — це нестандартна програма байт-коду віртуальної машини Dalvik, для якої розроблено формат інсталяційного пакету APK. Для обробки програм доступно багато бібліотек: Bionic (стандартна бібліотека, не сумісна з glibc); мультимедійна бібліотека PacketVideo OpenCORE (підтримує такі формати, як MPEG-4, H.264, MP3, AAC, AMR, JPEG і PNG); SGL (два 3D графічний движок); OpenGL ES 1.0 ES 2.0 (3D графічний движок); Surface Manager (забезпечує доступ до 2D/3D-додатків); WebKit (стандартний механізм

					<i>123.KI-41.04</i>	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		23

для веб-браузерів; обробляє HTML, JavaScript); FreeType (обробка шрифтів). двигун)); SQLite (легка база даних для всіх програм); SSL (протокол, що забезпечує безпечну передачу даних по мережі).

Порівняно з традиційними додатками Linux, програми Android підпорядковуються додатковим правилам: постачальники контенту - обмін даними між програмами; менеджер ресурсів - доступ до таких ресурсів, як файли XML, PNG, JPEG; менеджер сповіщень - доступ до рядка стану; менеджер активності - керування додатків діяльності.

Google надає безкоштовні комплекти для розробки програмного забезпечення для комп'ютерів x86 під керуванням Linux, Mac OS X (10.4.8 або новішої версії), Windows XP, Windows Vista та Windows 7. Для розробки потрібен JDK 5 або вище. Програми для Android можна розробляти на Java (не нижче Java 1.5). Eclipse має плагін - Android Development Tools (ADT), розроблений для Eclipse версій 3.3-3.7. Існує також плагін для IntelliJ IDEA, який полегшує розробку додатків Android, і плагін для середовища розробки NetBeans IDE, який більше не є експериментальним з NetBeans 7.0, хоча офіційно не випущений. Крім того, існує Motodev Studio для Android – всеосяжне середовище розробки на основі Eclipse, яке дозволяє працювати безпосередньо з Google SDK.

У 2009 році, крім ADT, був випущений Android Native Development Kit (NDK) — пакет інструментів і бібліотек, що дозволяє реалізувати частини вашої програми на C/C++. Рекомендується використовувати NDK для розробки важливих для швидкості частин коду. У Android 4.4 віртуальну машину Dalvik можна змінити на ART (Android Runtime). ART має вищу швидкість завантаження. У Android 5.0 опція автомобіля була втрачена, оскільки замість Dalvik було запущено ART.

					<i>123.KI-41.04</i>	<i>Арк.</i>
<i>Зм.</i>	<i>Арк.</i>	<i>№ докум.</i>	<i>Підпис</i>	<i>Дата</i>		24

21 жовтня 2008 року ОНА Alliance опублікував вихідний код для платформи Android: операційну систему, проміжне програмне забезпечення та основну наскрізну програму, написану на Java на Android з відкритим вихідним кодом. Загальний вихідний код Android становить 2,1 ГБ. «Концесійна ліцензія»

Вихідний код Android має ліцензію Apache 2.0. Після виходу Android 3.0 «Сті» Енді Рубін, президент мобільного підрозділу Google, заявив, що відкриття вихідного коду нової версії системи буде відкладено, оскільки система не готова до роботи на комунікаторі і потребує серйозна оптимізація. Це рішення викликало критику з боку аналітиків: оглядач ZDNet Крістофер Доусон, наприклад, назвав цей крок Google розчаруванням. Однак, згідно з обіцянкою компанії, восени 2011 року Google випустив вихідний код наступної версії системи Android 4.0 Ice Cream Sandwich.

21 жовтня 2008 року ОНА Alliance опублікував вихідний код для платформи Android: операційну систему, проміжне програмне забезпечення та основну наскрізну програму, написану на Java на Android з відкритим вихідним кодом. Загальний вихідний код Android становить 2,1 ГБ. «Концесійна ліцензія»

Вихідний код Android має ліцензію Apache 2.0. Після виходу Android 3.0 «Сті» Енді Рубін, президент мобільного підрозділу Google, заявив, що відкриття вихідного коду нової версії системи буде відкладено, оскільки система не готова до роботи на комунікаторі і потребує серйозна оптимізація. Це рішення викликало критику з боку аналітиків: оглядач ZDNet Крістофер Доусон, наприклад, назвав цей крок Google розчаруванням. Однак, згідно з обіцянкою компанії, восени 2011 року Google випустив вихідний код наступної версії системи Android 4.0 Ice Cream Sandwich.

Переваги:

- хоча спочатку забороняє встановлення програм з «неавтентифікованих джерел» (наприклад, з карт пам'яті), це обмеження стандартним чином вимкнено в налаштуваннях пристрою, дозволяючи встановлювати програми на телефони та планшети без підключення до Інтернету (наприклад, без Wi-Fi). -Точка доступу Fi і не хочеться витратити гроші на мобільний інтернет,

									123.KI-41.04	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата						25

- який часто коштує дорого), а також дозволяє будь-кому написати безкоштовний додаток для Android і протестувати його на своєму пристрої;
- android доступний для різних апаратних платформ, таких як ARM, MIPS, x86;
 - існують альтернативні Google Play магазини додатків: Amazon Appstore, Opera Mobile Store, Yandex.Store, GetUpps, F-Droid;
 - У версії 4.3 введений режим декількох користувачів.
 - Недоліки:
 - наявність в деяких Android-пристроях сервісів Google, що забезпечують можливість передачі ідентифікаційної інформації на сервери компанії, наприклад, інформацію про переміщення користувача в реальному часі;
 - у версії Android 1.6 розробники додали Native Development Kit, який дозволяє писати власні низькорівневі модулі для системи на мовах C/C ++, спираючись на стандартні Linux-бібліотеки. Хоча, наприклад, стандартна бібліотека мови C на платформі Android, відома як Bionic, якраз не є стандартною і повністю сумісною з libc;
 - для доступу до Google Play та інших сервісів від Google необхідно використовувати пропрієтарні додатки, які виробник телефону має право встановлювати на телефон тільки після укладення контракту з Google;
 - конкуренти Android виступили з критикою платформи, звинувачуючи її в надмірній фрагментації, що створює перешкоди розробникам. Google спростувала всі звинувачення, заявивши, що ніяких подібних проблем немає;
 - критикувати рішення Google не робити Android 3.0 Honeycomb загальнодоступним лише для членів Open Handset Alliance або на індивідуальний запит під час підписання угоди. Google мотивував це через погану підготовленість платформи та заходи щодо запобігання її недбалому впровадженню. Річард Столман каже, що все зрозуміло і просто: Android 3 не є безкоштовним програмним забезпеченням, за винятком ядра Linux, і хоча сучасні телефони Android не такі хороші, як смартфони Apple або Windows, ми не можемо сказати, що вони поважають вашу свободу.

						<i>123.KI-41.04</i>	<i>Арк.</i>
<i>Зм.</i>	<i>Арк.</i>	<i>№ докум.</i>	<i>Підпис</i>	<i>Дата</i>			26

Відповідно до GPL, Google тепер відкрив усі версії Android 4.4 і передав усі зміни до Linux.

- за даними Lookout Security Mobile у 2011 році користувачі смартфонів Android вкрали близько 1 мільйона доларів (наприклад, надсилаючи текстові повідомлення без відома власника телефону). Однак це часто трапляється з користувачами, які ненавмисно читають список дозволів.

2.1.1 Android Studio

IDE є важливим компонентом для розробки додатків для мобільних пристроїв, оскільки стає неможливим збирати та тестувати проекти, використовуючи лише редактори та компілятори. Існують автоматизовані системи складання (Gradle, Maven), інструменти тестування (JUnit, GreenHat), симулятори тощо, які допомагають створювати краще програмне забезпечення за менший час [15,16].

До 2012 року Android не мав власної IDE, а лише набір інструментів розробки, вбудованих у рішення від Oracle або Eclipse. Однак перша бета-версія рідної IDE Android була показана на Google IO 2012 –Android Studio.

Android Studio засновано на IntelliJ IDEA і Android Bundle і поглинає найкраще з обох продуктів. Підтримує всі інструменти синтаксису IDEA і має вбудовані інструменти створення інтерфейсу та відображення, емулятор і менеджер SDK з Android Bundle. Після виходу першої стабільної версії 1.0 у 2014 році Android Studio стала офіційно підтримуваною IDE Google для Android.

Графічний інтерфейс IDE Android Studio зображений на рис. 2.2.

					<i>123.KI-41.04</i>	<i>Арк.</i>
<i>Зм.</i>	<i>Арк.</i>	<i>№ докум.</i>	<i>Підпис</i>	<i>Дата</i>		27

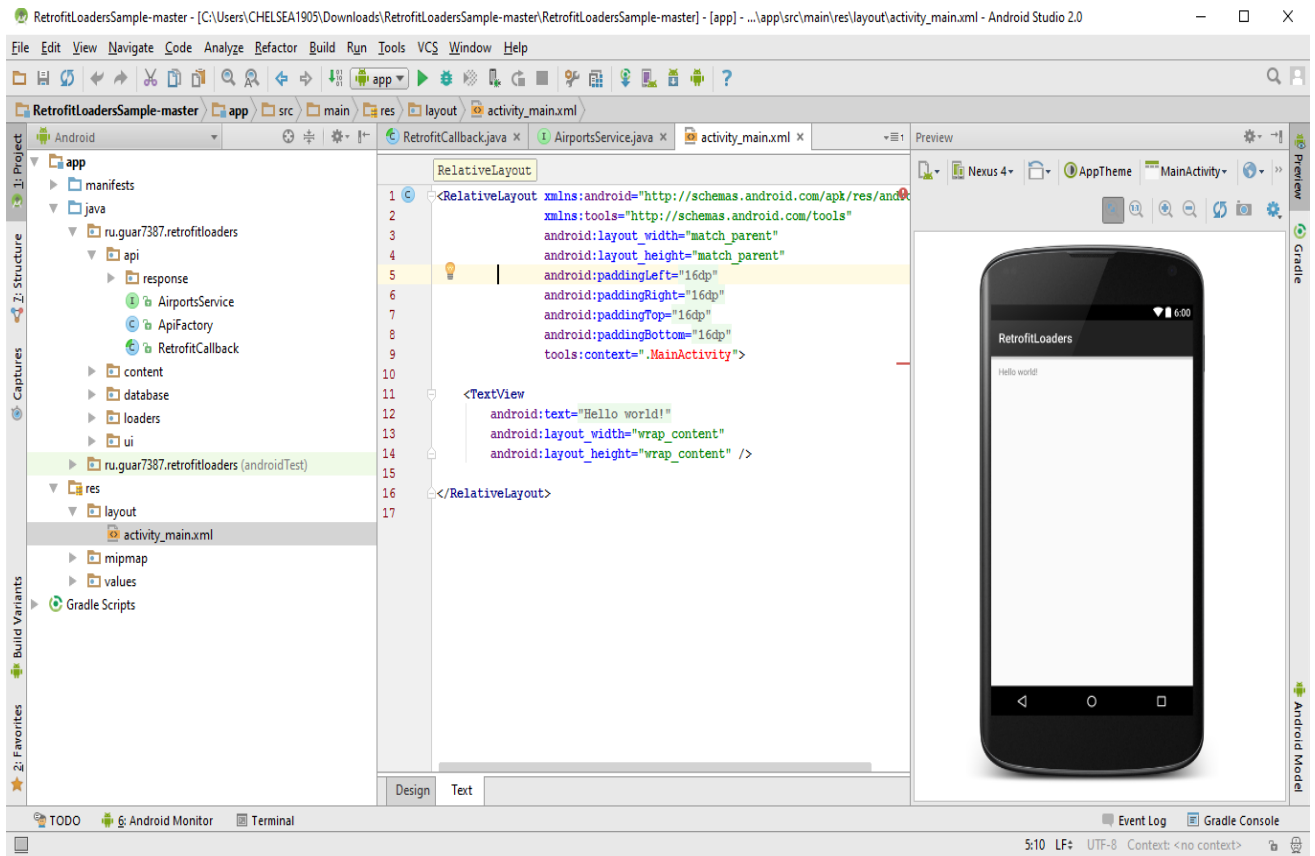


Рис.2.2 Графічний інтерфейс IDE Android Studio

2.1.2 Android SDK та Android Virtual Device (AVD)

Як і будь-яка інша мобільна операційна система, Android має набір інструментів розробки або SDK. Мета SDK полягає в тому, щоб полегшити написання програм для певної платформи, надаючи готові модулі або бібліотеки для доступу до функціональних можливостей системи, середовищ розробки або компілятора та інструментів тестування.

В Android на момент написання дипломної роботи був один з найбільш повних і функціональних SDK, який включає в себе популярне середовище розробки Android Studio, набір пакетів Android Bundle, ПЗ для локалізації, ПЗ для редагування SDK (рис 2.3) і емулятор для тестування додатків.

					<i>123.KI-41.04</i>	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		28

Отримати версію системи з додатковими функціями. Нові та розробники - пишуть більш загальні програми.

Важливою частиною SDK є емулятор Android. З його допомогою ви можете протестувати свою програму на різних версіях системи без фізичного пристрою. Однак важливо відзначити, що програми в ньому працюють набагато повільніше, відповідно, для тестування високонавантажуваних додатків емулятор не підходить. Ви також можете завантажити кращий користувацький емулятор (до Android Studio 2.0) або встановити програму Chrome за допомогою спеціального компонента. Незважаючи на це, Android SDK надає розробникам багато можливостей для тестування своїх програм. Приклад роботи стандартного емулятора з версії SDEA 2.0 на рис. 2.4.

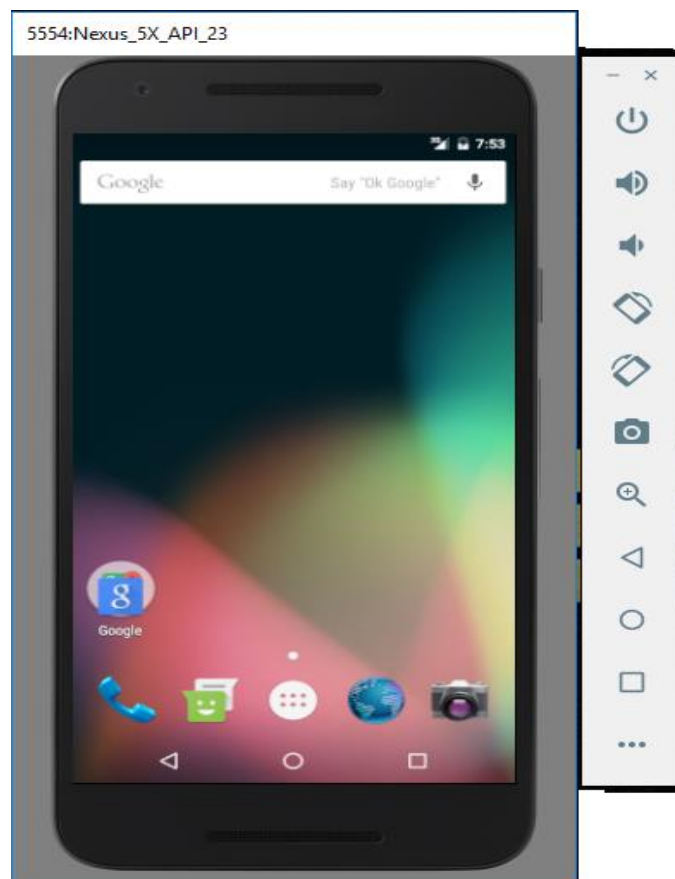


Рис.2.4 - Стандартний емулятор Android.

					<i>123.KI-41.04</i>	<i>Арк.</i>
<i>Зм.</i>	<i>Арк.</i>	<i>№ докум.</i>	<i>Підпис</i>	<i>Дата</i>		30

2.1.3 Фрагментація

Фрагментація - процес дроблення чогось на безліч дрібних розрізаних фрагментів. У разі Android - це присутність на ринку одночасно декількох версій системи крім останньої. Фрагментація за версіями на рис. 2.5.

Фрагментація пристрою – не єдина проблема, з якою стикаються розробники. Сама система дуже фрагментована і буде йти тільки далі в цьому напрямку. На графіку показано етапи фрагментації Android за версіями та постійне падіння популярності кожного етапу (білими лініями позначено спалахи).

Однією з переваг фрагментації Android є те, що вона забезпечує свободу пересування для виробників пристроїв, тож вони, у свою чергу, можуть надати споживачам пристрої, які повністю відповідають їхнім потребам. Цей фактор стає особливо важливим, коли Android замінює пристрої Nokia Symbian в менш економічно розвинених країнах.

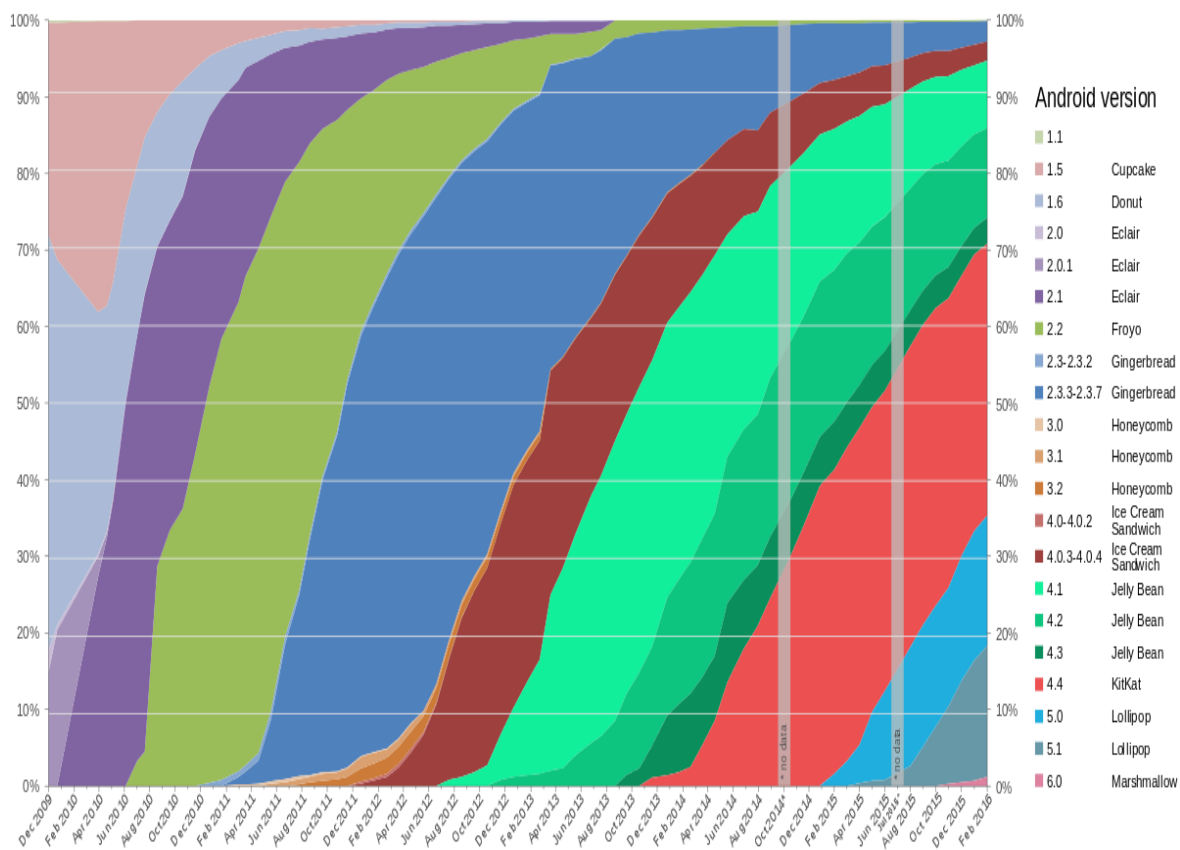


Рис.2.5 Діаграма поширення версій ОС Android

Зм.	Арк.	№ докум.	Підпис	Дата

Фрагментація API — це коли, старіші пристрої використовують старішу версію ОС, а новіші пристрої працюють гірше. На графіку показано співвідношення ВВП на душу населення до 5 найпопулярніших API Android на ринку (4 версії KitKat і 4.3.1 Jelly Bean), з більшими значеннями на осі Y, що означає меншу фрагментацію. Кореляція між цими двома показниками очевидна, і Катар виділяється з діаграми своїм високим ВВП, який не завжди відображає справжнє фінансове становище населення.

Поширення системи залежно від ВВП на душу населення на рис. 2.6. Цей графік являє окремо фрагментацію для країн з ВВП на душу населення більше і менше \$ 20 000, і ми бачимо, наскільки вона відрізняється.

Створити додаток для економічно розвинутого ринку набагато простіше, оскільки у цільовій аудиторії більше просунутих пристроїв на Android, що мають нові версії ОС.

Близько 35% пристроїв в більш економічно розвинених країнах мають версію KitKat, в менш розвинених - близько 12%.

Порівняння фрагментації Android з фрагментацією iOS на рис. 2.7. Фрагментація Android часто демонструється в порівнянні з iOS. Ці дві кругові діаграми показують фрагментацію API у двох конкуруючих ОС.

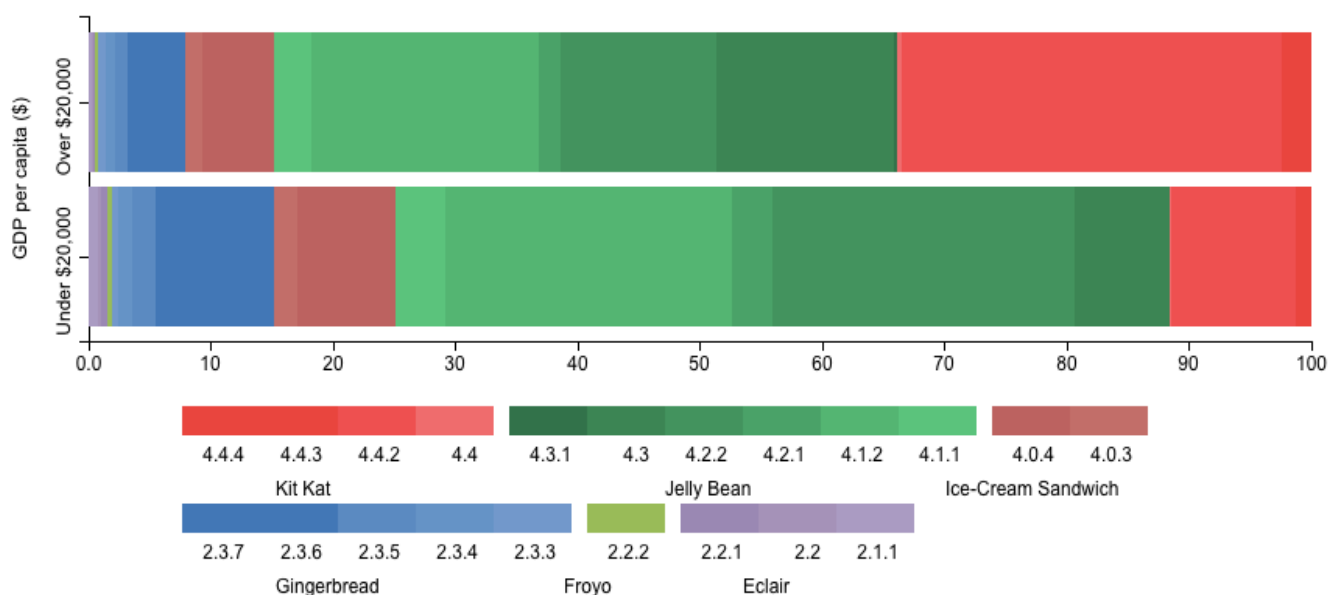


Рис.2.6 Фрагментація Android в залежності від ВВП на душу населення

Графіки фрагментації по сенсорах на рис. 2.8 і 2.9. Цей графік показує еволюцію пристроїв Android з точки зору кількості датчиків, використаних у прикладі серії Galaxy S. Лінійка смартфонів Galaxy показує, що виробники додають до своїх пристроїв все більш складні датчики. На діаграмі також показано протилежне явище — розробники створюють програми для певних датчиків (наприклад, вищезгаданий погодні сервіс використовує датчики вологості та температури), а потім перестають їх використовувати, додаючи розробникам фрагментацію.

Важливим фактом є те, що запорукою успіху будь-якої програми є правильне використання оболонки, і тут Android представляє дві фундаментальні проблеми для розробників. По-перше, виробники хочуть налаштувати стандартні інтерфейси (наприклад, TouchWiz від Samsung або HTC Sense). Друге - для різних розмірів екранів Android-пристроїв. Розробити схеми, які б адекватно працювали на екранах різного розміру, є дуже складним завданням.

Таблиця 2.1 Фрагментація Android - пристроїв

Версія	Кодове ім'я	Дата випуску	Рівень API	Розподіл
8.0	Marshmallow	5 жовтня, 2019	23	0.7 %
5.1.x	Lollipop	9 березня, 2015	22	15.7 %
5.0.0–5.0.2		3 листопада, 2014	21	16.9 %
4.4.0–4.4.4	KitKat	31 жовтня, 2013	19	36.1 %
4.3.x	Jelly Bean	24 липня, 2013	18	3.5 %
4.2.x		13 листопада, 2012	17	12.2 %
4.1.x		9 липня, 2012	16	9.0 %
4.0.3–4.0.4	Ice Cream Sandwich	16 грудня, 2011	15	2.7 %
2.3.3–2.3.7	Gingerbread	9 лютого, 2011	9	3.0 %
2.2	Froyo	20 травня, 2010	8	0.2 %

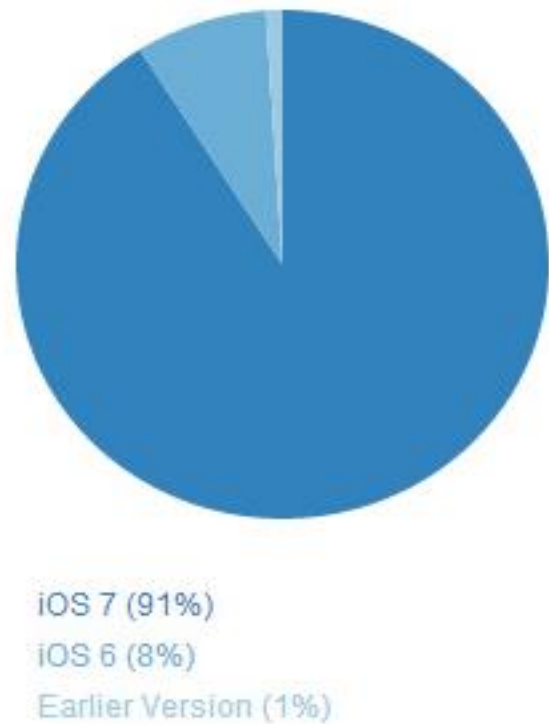
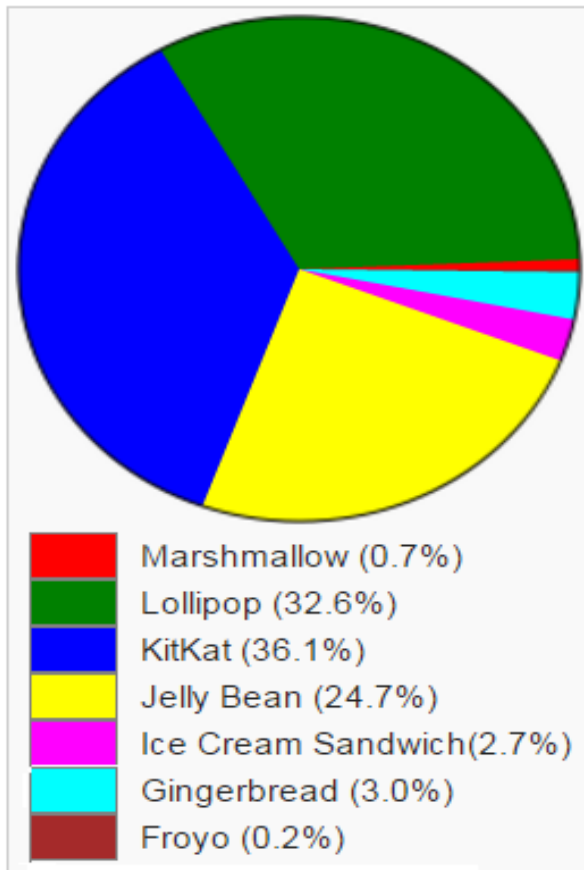


Рис.2.7 Порівняння фрагментації iOS та Android

На рис. 2.9 зображено відношення розмірів екранів мобільних телефонів під ОС Android. За даною діаграмою стає зрозумілим що сьогодні найбільш поширені телефони під ОС Android мають величину в 5-5,5 дюйма.

Розміри екранів Android девайсів в порівнянні з пристроями під IOS зображені на рис. 2.10, а iOS - 2.11. З даних малюнків чітко видно різниця між розвитком даних платформ та тимами фрагментації.

На відміну від цього, пристрої Apple мають лише чотири розміри екрана, оскільки компанія подвоїла щільність пікселів і вчетверо збільшила роздільну здатність, зберігаючи при цьому екран незмінним. На графіку нижче показано фрагментацію iOS за розміром екрана, яку легко порівняти з Android.

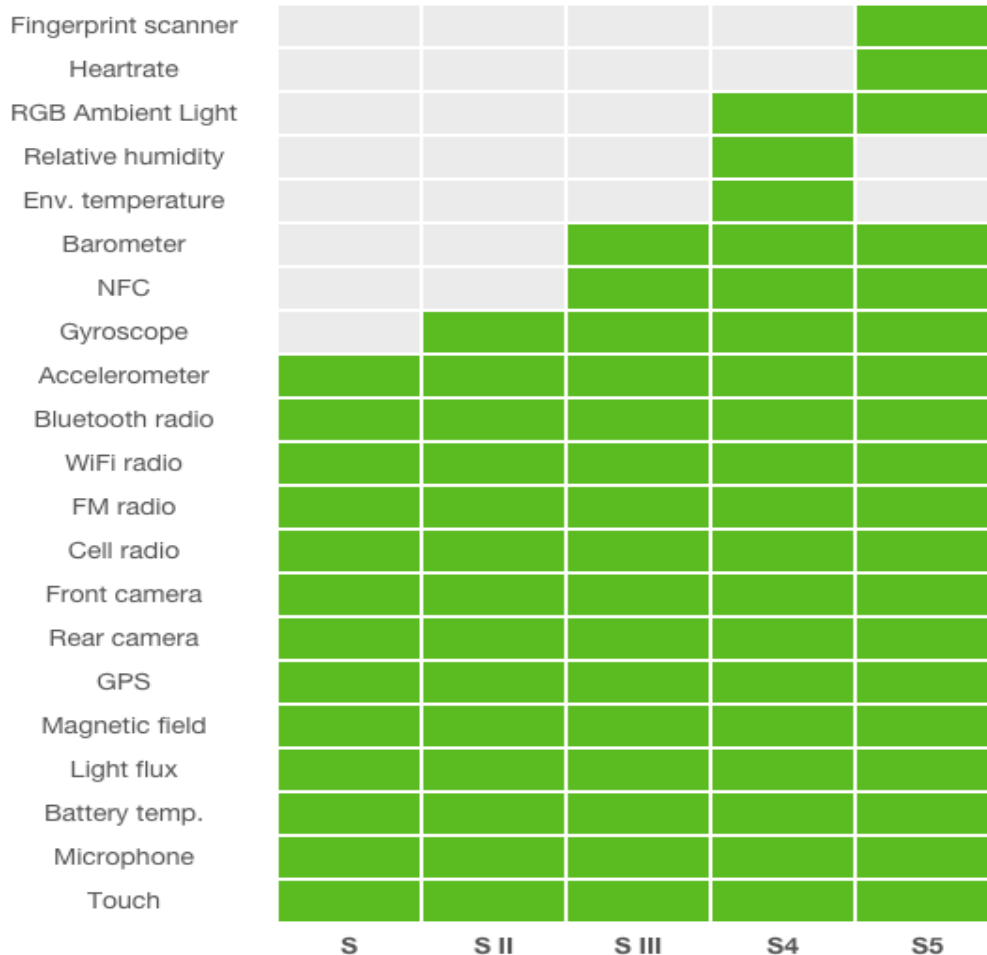


Рис.2.8 Фрагментація сенсорів на прикладі лінійки смартфонів Galaxy S

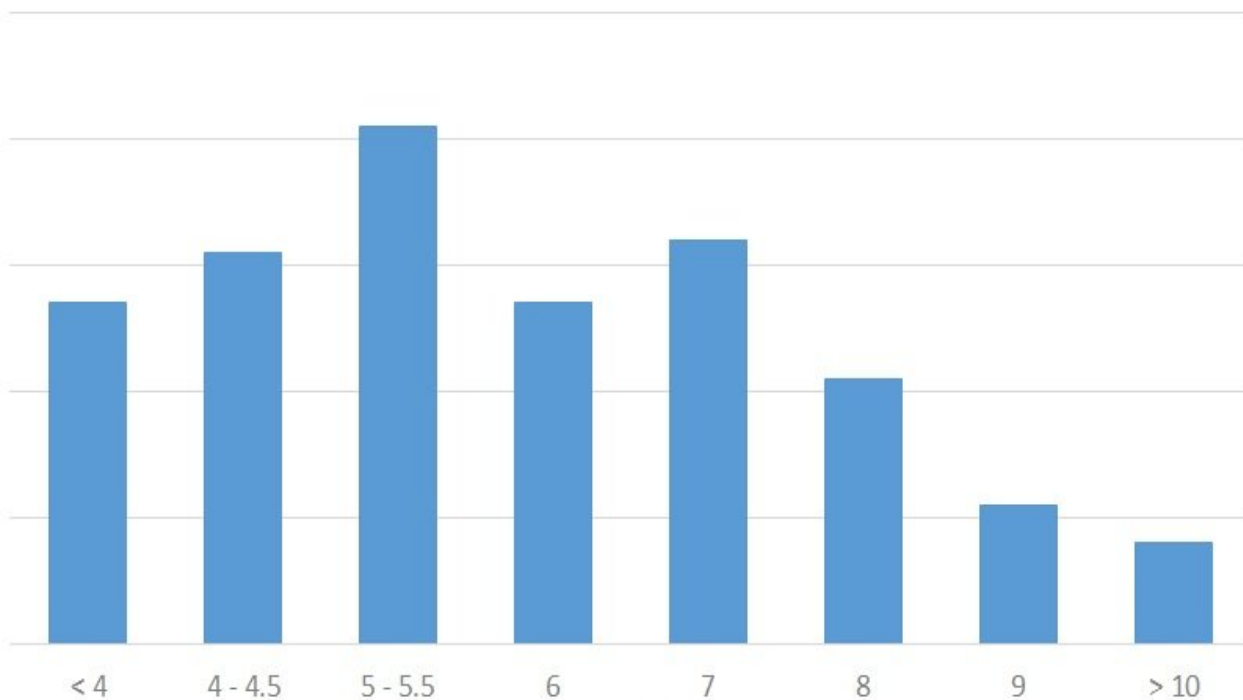


Рис.2.9 Відношення розмірів екранів мобільних телефонів під ОС Android

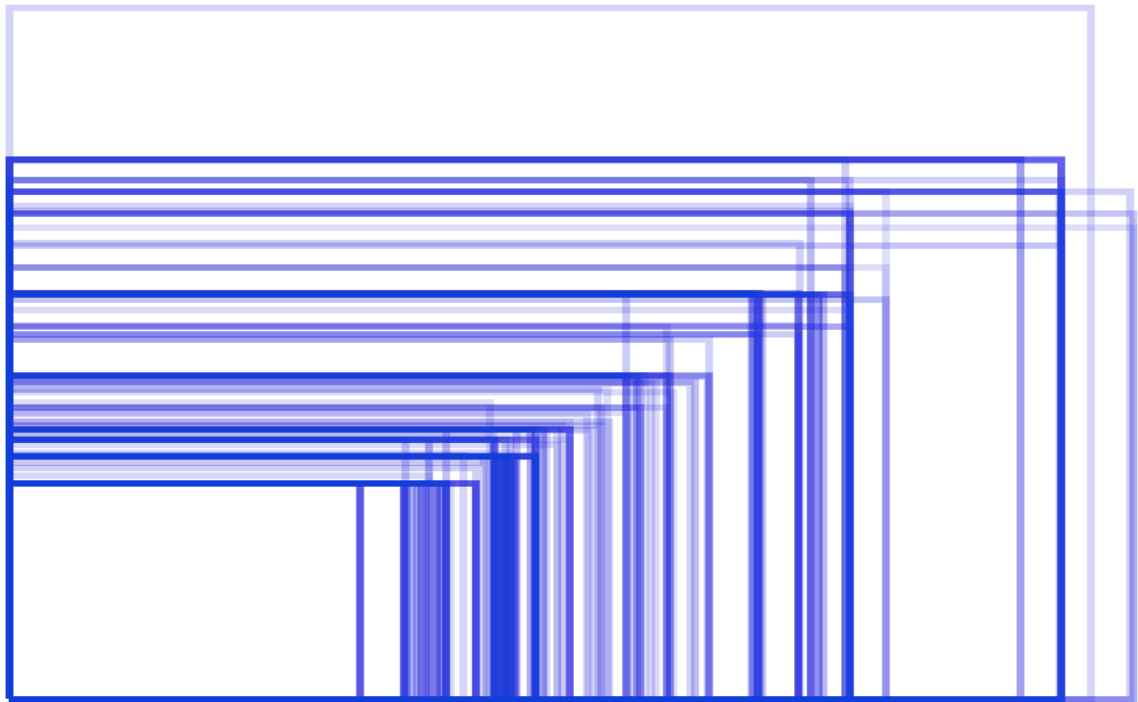


Рис.2.10 Всі можливі розміри дисплеїв Android пристроїв у порівнянні

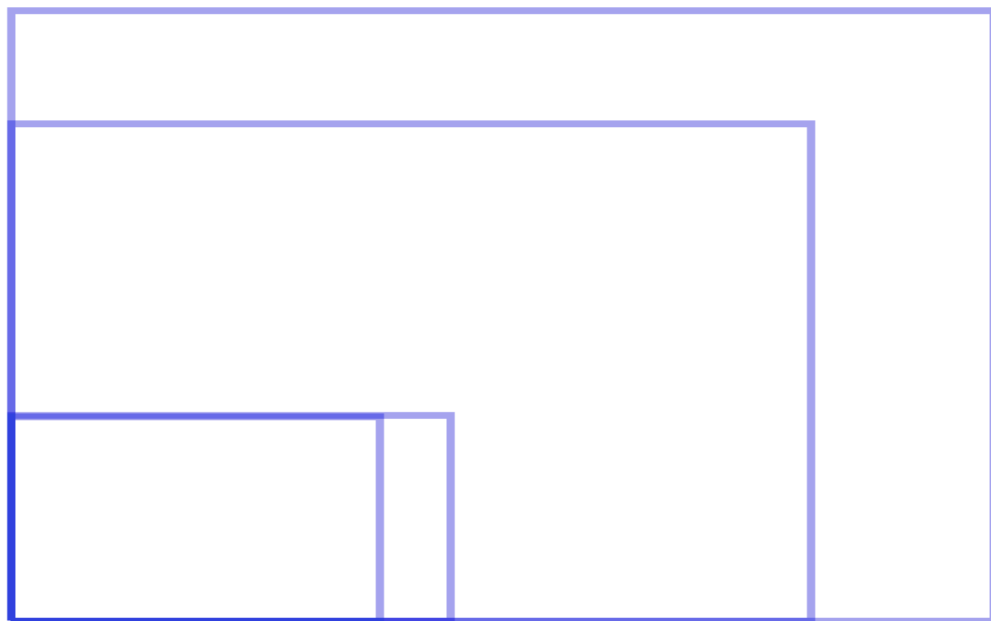


Рис.2.11 Всі можливі розміри дисплеїв iOS пристроїв у порівнянні

2.1.4 Особливості Android M та Android N. Material Design

Дата виходу: 28 травня 2018 року • Версія ядра Linux: 3.10 • Версія API: 23

Google оголосила про нову версію своєї мобільної операційної системи під кодовою назвою Android M. Віце-президент Google Сундар Пічаї представив новинку на конференції розробників Google I/O.

					123.KI-41.04	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		36

Основні функції цього випуску включають систему мобільних платежів Android Pay і стандартизовану перевірку відбитків пальців на рівні платформи. Пристрої зі сканерами відбитків пальців будуть доступні для розблокування гаджетів і авторизації покупок у Google Play, а розробники зможуть вставляти відповідні API в будь-який додаток Android.

Google також дуже зосереджена на збереженні заряду акумулятора – за що пристрої Android часто критикують. Функція відкладення працює через датчик руху: коли пристрій тривалий час стоїть на місці, активується режим «глибокого сну», а діяльність фонових програм суворо обмежується, що дозволяє економити заряд акумулятора [17].

Google також обіцяла користувачам більше контролю над тим, як додатки, що використовують дані, запити на роботу з інформацією будуть відправлятися не в момент установки, а безпосередньо у процесі використання програми.

Нова функція Chrome custom tabs покликана зробити більш зручним споживання веб-контенту на мобільному. Всього в Android M буде близько 100 нових функцій, заявив віце-президент Google Девід Бюрку .

З популярністю Android 6.0, Material Design став популярним. Material Design — це принцип, система, парадигма дизайну веб-сайту, програмного забезпечення та додатків Google, а також правила дизайну інтерфейсу операційної системи Android. Вперше показано на Google I/O 25 червня 2017 року. Концепція дизайну — це інтерфейс, поведінка та зовнішній вигляд якого відповідають правилам поведінки та зовнішнього вигляду реальних паперових карток. Material Design повністю застосовується в операційних системах Android 5.0, Android 5.1, Android 6.0.

Дата випуску: 16 березня 2019 р. • Версія ядра Linux: 3.10 • Версія API: 24. Зараз працює на пристроях серії Nexus (Nexus 6, Nexus 9, Nexus Player, Nexus 5X, Nexus 6P, Pixel C).

Однією з особливостей нової версії Android є багатокористувацький режим, режим накладання «збереження даних», перемикання фонових завдань, «нічний режим», покращені функції енергозбереження та багато інших покращень.

					<i>123.KI-41.04</i>	<i>Арк.</i>
<i>Зм.</i>	<i>Арк.</i>	<i>№ докум.</i>	<i>Підпис</i>	<i>Дата</i>		37

2.1.5 Вибір апаратних засобів

Основним мікропроцесором у більшості пристроїв Android сьогодні є мікропроцесор Cortex серії A. ARM Cortex-A7 — це суперскалярний 64-розрядний мікропроцесор від ARM Holdings [18].

Процесор забезпечує хороший баланс між продуктивністю та енергоспоживанням і може підтримувати 32-розрядні та 64-розрядні програми.

Cortex-A7 пропонує набагато вищу продуктивність, ніж успішний процесор Cortex-A53, і може використовуватися в автономному пристрої. Використовується на платформі Raspberry Pi версії 3.

Цей мікропроцесор має повну назву - MediaTek MT6589. Загальна схема MediaTek MT6589 приведена на рис.2.12.

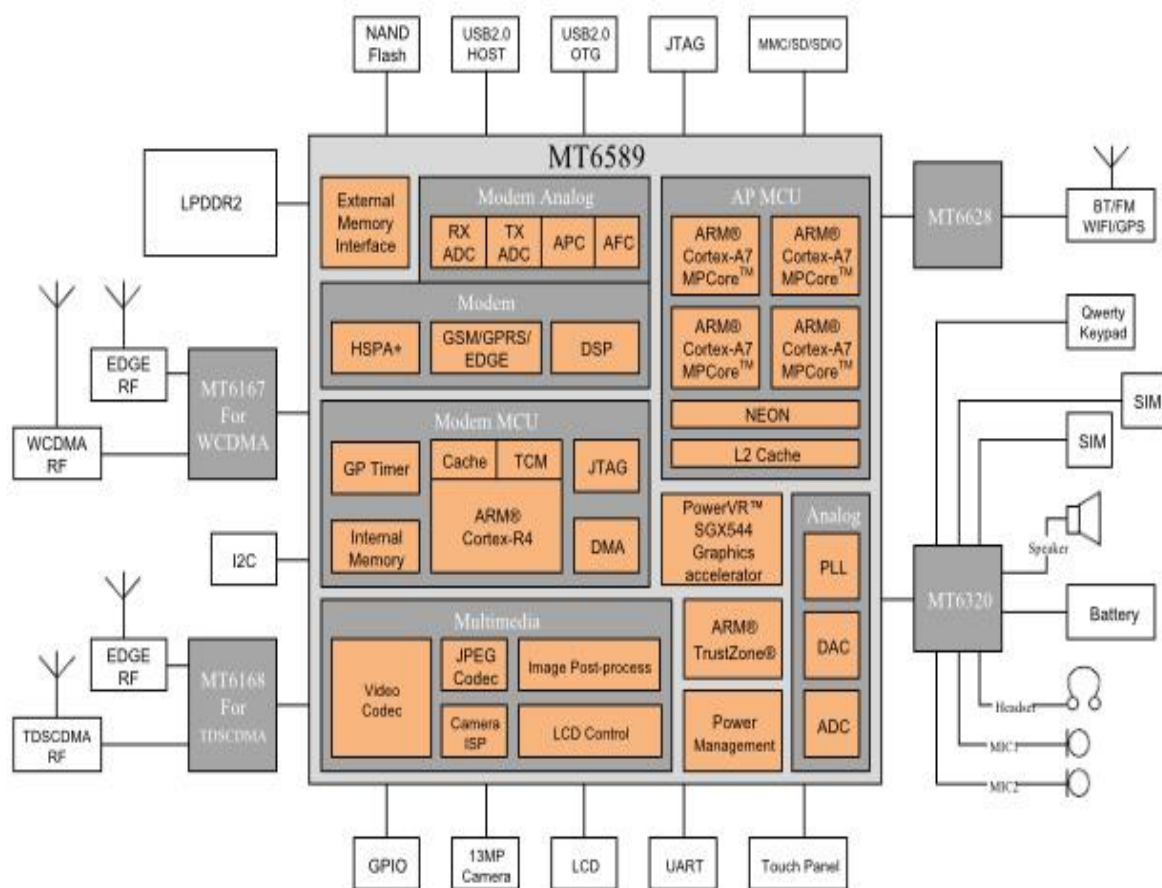


Рис.2.12 Схема мікропроцесора MediaТех з усіма інтерфейсами

Основними характеристиками є наявність 2 Гб оперативної пам'яті ,4 Гб вбудованої пам'яті, 4-ри ядра по 1,2 ГГц частоти та ряд інших особливостей.

Зм.	Арк.	№ докум.	Підпис	Дата

2.2. Вибір засобів для реалізації серверної частини на основі REST-архітектури

2.2.1 IntelliJ IDEA

Для написання системного сервера я буду використовувати IntelliJ IDEA - комерційне інтегроване середовище розробки від JetBrains для різних мов програмування (Java, Python, Scala, PHP та ін.). Система поставляється в урізаній формі безкоштовної версії «Community Edition» та повнофункціональної комерційної версії «Ultimate Edition», а активні розробники відкритих проектів мають можливість отримати безкоштовну ліцензію. Оригінальний текст видання Community Edition поширюється під ліцензією Apache 2.0. Бінарні випуски підготовлені для Linux, Mac OS X і Windows [19,20].

Перша версія IntelliJ IDEA з'явилася в січні 2001 року, і вона швидко стала популярною як перша IDE Java з широкими інтегрованими інструментами рефакторингу, що дозволило програмістам швидко реструктурувати оригінальні програми коду.

Середовище розроблено так, щоб зосередитися на продуктивності програмістів, дозволяючи їм зосередитися на розробці функцій, тоді як IntelliJ IDEA піклується про повсякденні операції.

Починаючи з версії 6 продукту, IntelliJ IDEA надає інтегровані інструменти для розробки графічних інтерфейсів користувача.

Версія 9.0 є безкоштовною версією Open Source Community Edition. Вихідний код відкритої версії IntelliJ IDEA Community Edition поширюється під ліцензією Apache 2.0. Бінарні пакети підготовлені для Linux, Mac OS X і Windows.

IntelliJ IDEA включає розробку, створену в партнерстві з Google, використовуючи IntelliJ IDEA як основу для свого нового відкритого середовища розробки, Android Studio. Ця співпраця значно розширює кадрові можливості IntelliJ IDEA для розробки додатків для платформи Android.

Community Edition IntelliJ IDEA підтримує інструменти (плагіни) для тестування TestNG і JUnit, CVS, Subversion, Mercurial і Git систем контролю версій, інструментів компіляції Maven, Ant, Gradle, мов програмування Java, Scala,

					123.KI-41.04	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		39

Clojure, Groovy і Dart. Підтримує розробку додатків для мобільної платформи Android. Система включає в себе модуль візуального дизайну для графічного інтерфейсу Swing UI Designer, редактор XML, редактор регулярних виразів, систему перевірки коду, систему моніторингу завдань і надбудови для імпорту та експорту проектів Eclipse. Доступні інструменти для інтеграції з системами відстеження помилок JIRA, Trac, Redmine, Pivotal Tracker, GitHub, YouTrack, Lighthouse.

Комерційна версія «Ultimate Edition» має підтримку інших мов програмування, підтримку технологій Java EE, діаграм UML, розрахунок покриття коду, використання фреймворків (Rails, Grails, Google Web Toolkit, Spring, Play Framework і Hibernate)), інтегрується з Perforce, Microsoft Team Foundation Server і Rational ClearCase.

2.2.2 Apache Tomcat 9.0

Для запуску розробленого сервера я викоистовуватиму Apache Tomcat 9.0. Цей продукт використовується досвідченими розробниками для створення комерційних проектів, оскільки це зручне, просте у використанні та надійне програмне забезпечення.

Apache Tomcat — це контейнер сервлетів, розроблений Apache Software Foundation. Повністю написаний мовою програмування Java, реалізує сервлет Sun Microsystems і специфікації Java Server Pages, які є стандартами для розробки веб-додатків на Java.

Веб-сервери можуть мати різні додаткові функції, наприклад:

- автоматизація роботи веб-сторінок;
- ведення журналу звернень користувачів до ресурсів;
- аутентифікація і авторизація користувачів;
- підтримка динамічно генеруються сторінок;
- підтримка HTTPS для захищених з'єднань з клієнтами.
- часто на комп'ютері разом з веб-сервером встановлюється також і поштовий.

					<i>123.KI-41.04</i>	<i>Арк.</i>
						40
<i>Зм.</i>	<i>Арк.</i>	<i>№ докум.</i>	<i>Підпис</i>	<i>Дата</i>		

В якості клієнтів для звернення до веб-серверів можуть використовуватися різні програми та пристрої:

- - веб-браузер, що працює на настільному комп'ютері або переносному пристрої (наприклад, кишеньковому ПК);
- - різноманітні програми, самостійно звертаються до веб-серверів для отримання оновлень або іншої інформації (наприклад, антивірус може періодично запитувати у певного веб-сервера поновлення своїх баз даних);
- - мобільний телефон, який одержує доступ до ресурсів веб-сервера за допомогою протоколу WAP;
- - інші цифрові пристрої або побутова техніка.

					<i>123.KI-41.04</i>	<i>Арк.</i>
<i>Зм.</i>	<i>Арк.</i>	<i>№ докум.</i>	<i>Підпис</i>	<i>Дата</i>		41

РОЗДІЛ 3. РОЗРОБКА ІНФОРМАЦІЙНОЇ ПРОГРАМНОЇ СИСТЕМИ ПОШУКУ ТА КУПІВЛІ КВИТКІВ НА РІЗНІ ВИДИ ТРАНСПОРТУ

3.1 Реалізація клієнтської частини спеціалізованої системи продажу білетів

3.1.1 Розробка основних функціональних вузлів Android - додатку

Незважаючи на те, що у клієнт – серверній системі клієнтська частина зазвичай не має багато логіки, оскільки вся логіка та дані збережені на серверній частині, у реалізованому додатку чимало програмний методів, які виконують ряд важливих дій.

Основним функціональним вузлом є методи призначені для підключення до інтернет з'єднання, передачі GET- та POST- запитів на серверну частину та оброблення відповідей сервера. Даний вузол реалізований бібліотекою Android Spring Library, оскільки дана бібліотека є найбільш сучасною, простою у користуванні та найбільш дієспособною і швидкою в роботі з інтернет підключенням. Порівняння з іншими Андроїд – бібліотеками, які призначені для роботи з інтернетом зображено у таблиці 3.1.

Таблиця 3.1 - Порівняльна таблиця Андроїд-бібліотек

Бібліотека	Один запит	7 запитів	25 запитів
AsynkTask	<i>941ms</i>	<i>4,539ms</i>	<i>13,957ms</i>
Volley	<i>560ms</i>	<i>2,202ms</i>	<i>4,275ms</i>
Retrofit	<i>312ms</i>	<i>889ms</i>	<i>1,059ms</i>
Spring Library	<i>298ms</i>	<i>669ms</i>	<i>998ms</i>

Отже, Android Spring Library – це бібліотека написана на Java за допомогою Spring Framvork компанією Oracle і на даний момент є найбільш популярною бібліотекою роботи з андроїд – аплікацією та інтернетом.

Для роботи із серверною частиною було створено та використовувались наступні класи каталогу adapter та TicketDTO.class. Частина структури проекту, що відповідає за роботу з сервером зображена на рис. 3.1 та 3.2.

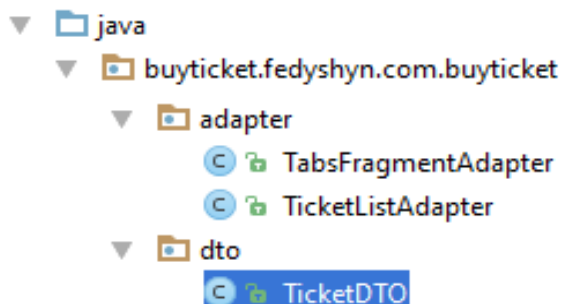


Рис.3.1 Класи з проекту призначені для взаємодії із серверною частиною

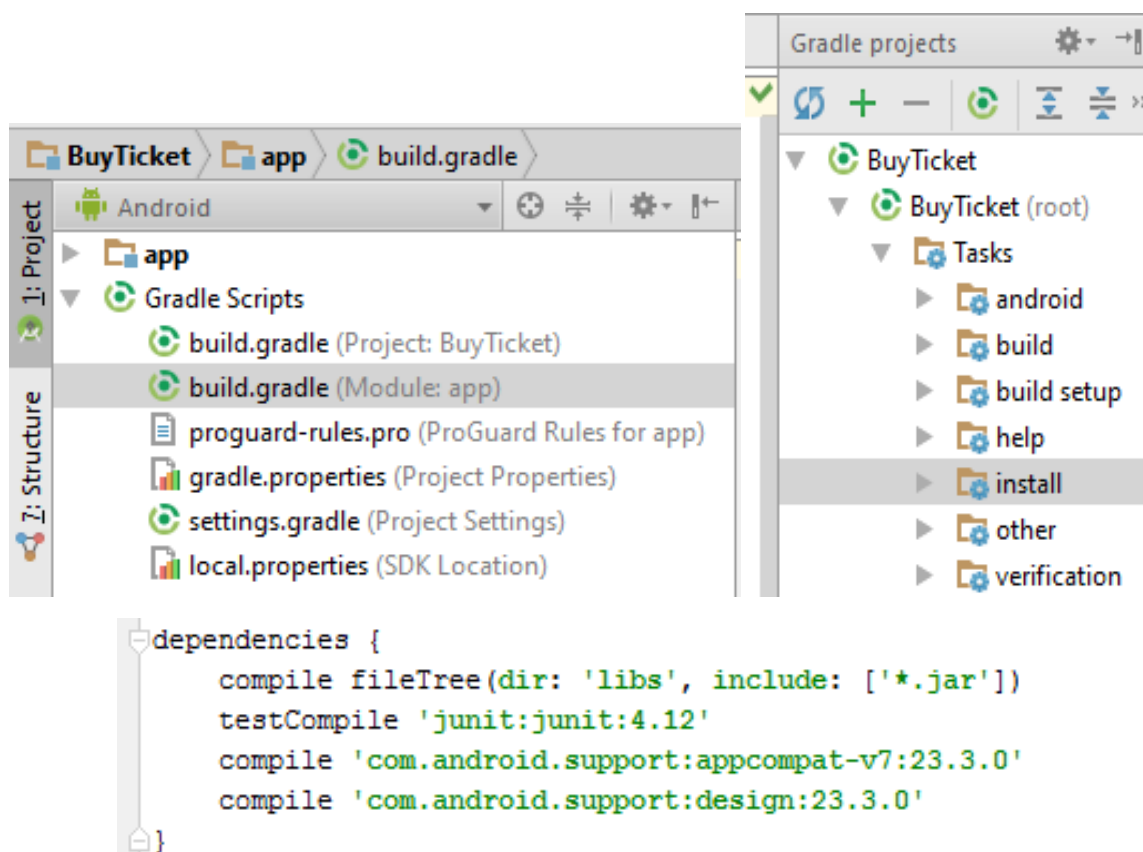


Рис.3.2 Бібліотеки, підтягнуті Gradle

На рис. 3.3 зображено загальну структуру проекту Android – додатку у середовищі розробки Android Studio. Усі ресурси необхідні для роботи андроїд – додатка знаходяться в папці ресурсів (res). Зображення у підкаталозі /mipmap/ та /drawable/. Усі графічні настройки додатку зображені знаходяться у підкаталозі /layout/, а усі настройки тексту та кольорів у підкаталозі /values/.

Файл настройки та конфігурацій (AndroidManifest.nml) знаходиться у окремому підкаталозі. Такий самий принцип і у зберіганні пакетів з класами. Окреме місце у проекті займаються файли та бібліотеки підкачані системою збирання додатків – Gradle.

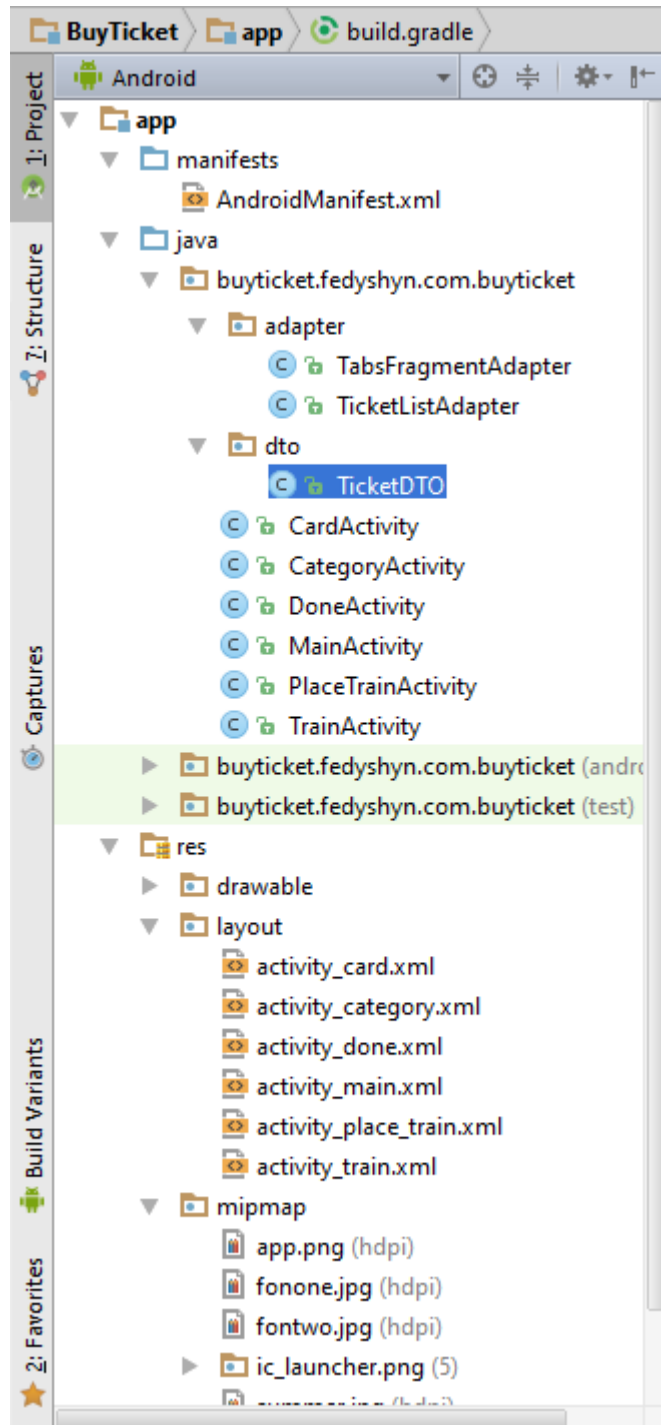


Рис.3.3 Загальна структура проекту у Android Studio

Другим важливим функціональним вузлом розробленого додатку є активності, що відповідають за взаємодію з користувачем, тобто графічна оболонка реалізованого Андроїд – додатку.

На рис. 3.4 зображена загальна блок - схема роботи розробленого Android – додатку з використанням усіх основних функціональних вузлів.

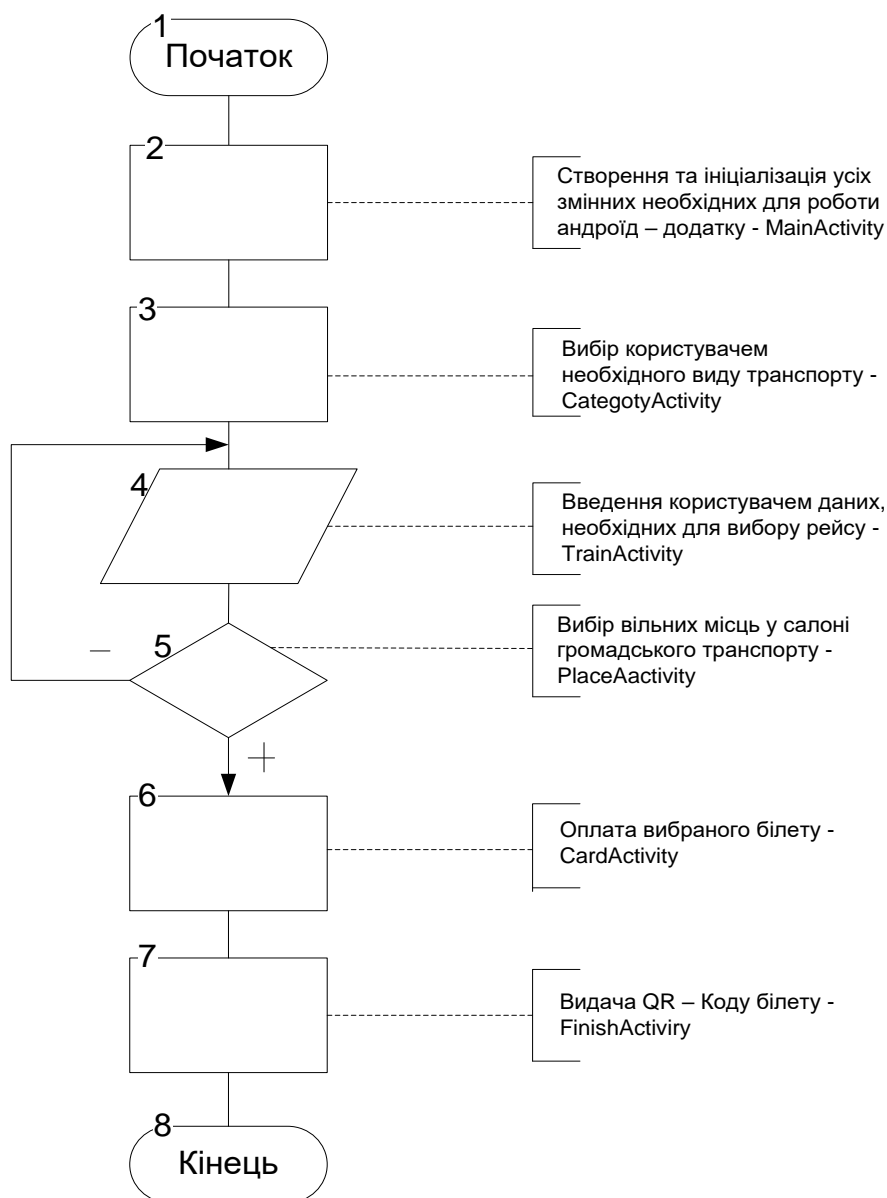


Рис.3.4 Загальна блок-схема роботи Android-додатку

Важливо зазначити, що алгоритм вибору найбільш підходящих маршрутів за введеними даними користувача реалізований на серверній частині розробленої спеціалізованої системи. Усі введені дані відправляються POST – запитом на сервер, де одробляються і на їх підставі сервер повертає з бази даних усі відповідні розклади маршрутів.

На рис. 3.5 зображена блок-схема реалізації підключення клієнтської частини (android – додатку) до серверної частини (REST – сервісу) за допомогою Android Spring Library.

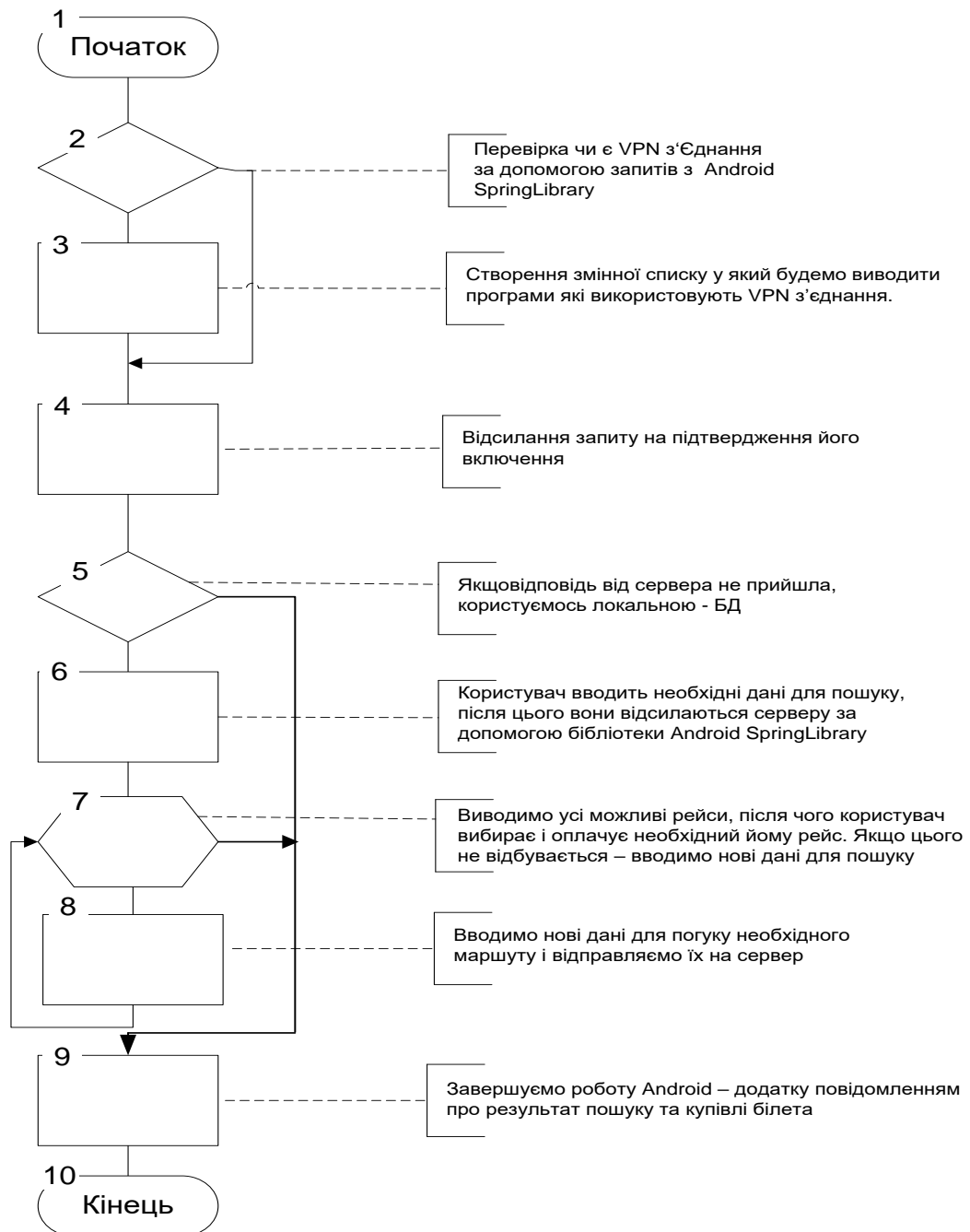


Рис. 3.5 Блок -схема підключення клієнта до сервера за допомогою Android Spring Library

У фоновому режимі працюють функції, зображені на рис. 3.6.

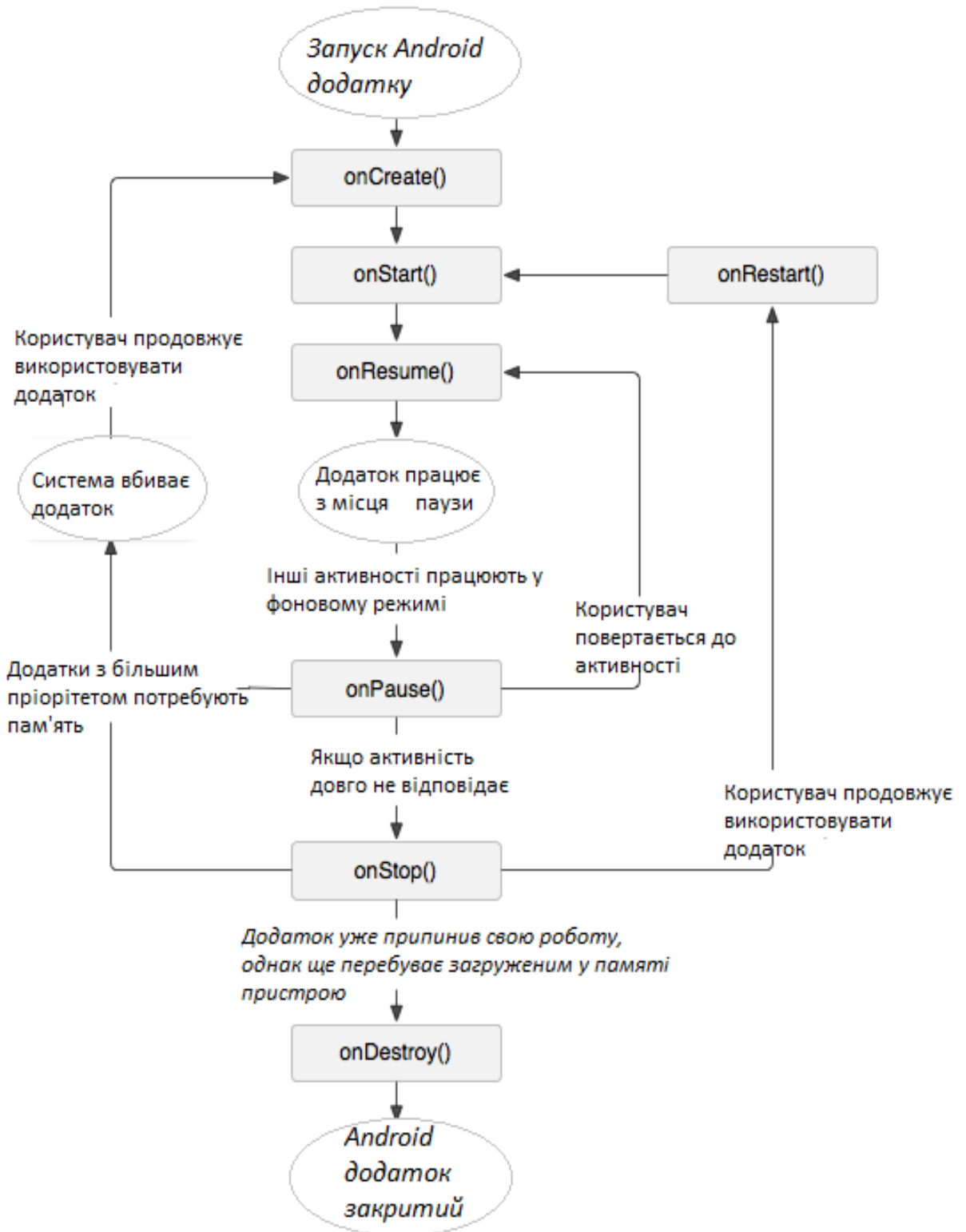


Рис.3.6 Функції запущені у фоновому режимі роботи Android – додатку

Зм.	Арк.	№ докум.	Підпис	Дата

Серед основних таких функцій є наступні:

Метод **onAttach** викликається, коли примірник Fragment асоціюється з примірником Activity Це ще не означає повної ініціалізації Activity

Метод **onCreate** викликається, коли створюється або відтворюється примірник Fragment Якщо відбувається відновлення після передував руйнування фрагмента або активності, в якій утримувався, то аргумент bundle виявиться ненульовим, якщо було збережено стан.

Метод **onCreateView** викликається, коли примірник Fragment повинен створити ієрархію об'єктів View, які в ньому містяться Роль фрагмента в активності досить незвичайна У чомусь фрагмент нагадує по поведінці ViewGroup, але не відноситься до ієрархії класу View Можна сказати, що фрагмент дозволяє екземпляру активності містити кілька наборів видів У нашому прикладі завантажується виключно простий макет, містить єдиний ViewHolder.

Метод **onActivityCreated** викликається, коли активність, яка містить екземпляр фрагмента, вже створена, а також створені об'єкти View, що містяться у Fragment На цьому етапі ви можете вільно шукати об'єкти View, наприклад, по їх ID:

- Метод **onResume** викликається, коли фрагмент стає видимим і працює.
- Метод **onStart** викликається, коли фрагмент стає видимим і працює.
- Метод **onPause** викликається в тих же умовах, що і метод onPause, що відноситься до активності, коли система збирається відвести фрагмент з переднього плану.
- Метод **onStop** викликається, перед тим як фрагмент повинен зупинити роботу.
- Метод **onSaveInstanceState** викликається, коли необхідно зберегти стан екземпляра, щоб, якщо цей екземпляр буде знищений (фактично будуть видалені вказують на нього посилання), будь призначене для збереження стан, специфічне для класу, можна було зберегти в об'єкті Bundle, переданому цього виклику.

Активність (activity) в Android – це і елемент для взаємодії з користувачем (зазвичай активність цілком займає екран мобільного пристрою з Android), і елемент виконання Коли ви створюєте інтерактивну програму.

Об’єкти-фрагменти не є компонентами. Фрагмент можна вважати способом розбиття Activity на кілька об’єктів, що містяться в Activity. Кожен фрагмент має власну ієрархію видів View, і ця ієрархія діє так, як якщо б вона розташовувалася усередині Activity.

Коли система запускає цю активність, викликається конструктор для TestActivity підкласу Activity, після чого викликається його метод onCreate. Таким чином, завантажуються і відображаються ієрархія видів, описана у файлі main.xml.

Клас Activity – це один з найважливіших класів у системі Android. Клас забезпечує модульність додатків і дозволяє спільно використовувати функції. Активність (Activity) взаємодіє з середовищем часу виконання Android для реалізації ключових аспектів життєвого циклу програми. Кожна активність також може незалежно конфігуруватися за допомогою класу Context.

Перехід між активностями у додатку здійснюється за допомогою намірів або Intent. Кожному наміру можна передати певні об’єкти, щоб наступна активність могла з ними працювати і обробляти необхідні дані та запити. Для цього використовується метод putExtra(), в якому якості параметра передається той об’єкт, який має бути доступний наступній активності.

3.1.2 Розробка графічного інтерфейсу користувача.

При завантаженні та запуску додатку на Android-пристрій користувач має змогу бачити головне вікно додатку. Дане вікно зображена на рис. 3.7. Для його побудови використано EmptyActivity (Пусту Активність) для якого використано властивість background та додано клавiшу - зображення (ImageButton), яка служить для переходу на наступну активність – вибір виду транспорту (CategoryActivity).

Усі налаштування графічного інтерфейсу виконувались у файлах типу layout з розширенням .xml, які призначені для візуалізації активностей та взаємодії з користувачами.

										123.KI-41.04	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата							49

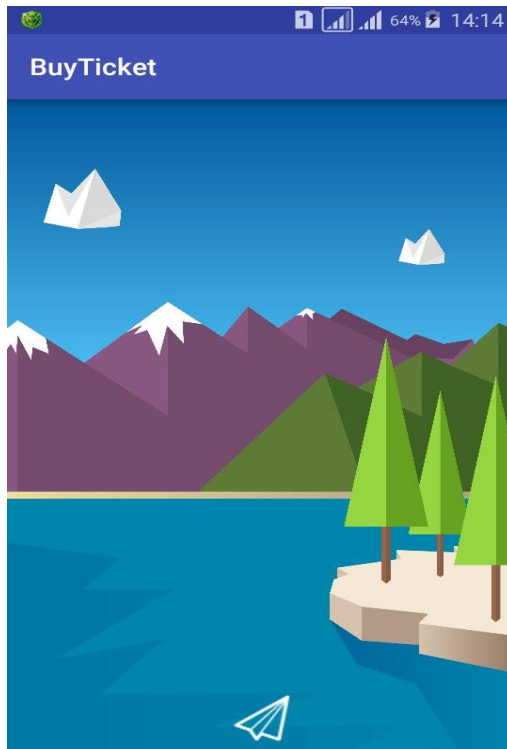


Рис.3.7 Початкове вікно Android–додатку (клієнтська частина)

Наступне вікно додатку надає користувачеві можливість вибрати необхідний вид транспорту та подальшу купівлю квитка даного типу. Вікно цієї активності зображено на рис. 3.8. Налаштування даного графічного інтерфейсу відбувались у файлі `layout_category.xml`. На цьому ж екрані розташовується список доступних видів транспортних засобів та можливість відкриття меню налаштування. Також на цьому екрані можна відкрити бокову панель меню, на якій зображені усі відомості про користувача, а саме – його персональні дані профілю, активні та неактивні покупки білетів користувача, баланс платіжної карточки, яка є прикріпленою до даного профілю та можливість виходу з облікового запису.

Даний тип меню є досить популярним сьогодні не лише в ОС Android але і у інших ОС. Даний тип меню є зручним тому, що не забирає вільного місця на екрані та не використовує додаткових фрагментів.

					<i>123.KI-41.04</i>	<i>Арк.</i>
<i>Зм.</i>	<i>Арк.</i>	<i>№ докум.</i>	<i>Підпис</i>	<i>Дата</i>		50

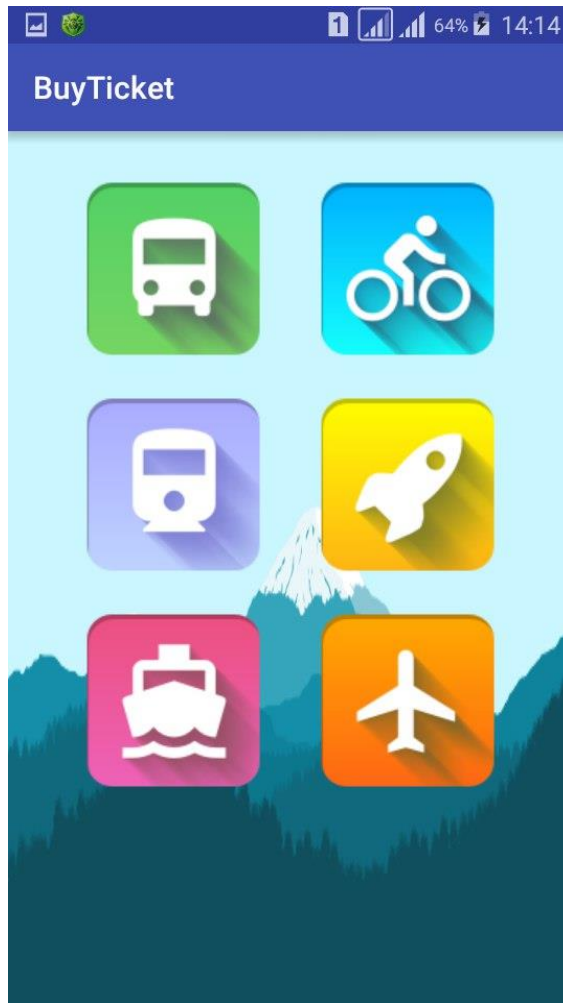


Рис.3.8 Вибір необхідного виду транспорту

Після вибору необхідного виду транспорту (це може бути поїзд, автобус, літак, паром чи інший) користувачеві надається можливість вказати дані про рейс, після чого сервер надає користувачеві усі підходящі рейси і користувач вже вибирає найбільш йому зручний. Користувач повинен вказати такі дані, як місце відправлення та пункт прибуття а також необхідне число подорожі. Дана активність зображена на рис. 3.9.

					<i>123.KI-41.04</i>	<i>Арк.</i>
<i>Зм.</i>	<i>Арк.</i>	<i>№ докум.</i>	<i>Підпис</i>	<i>Дата</i>		51

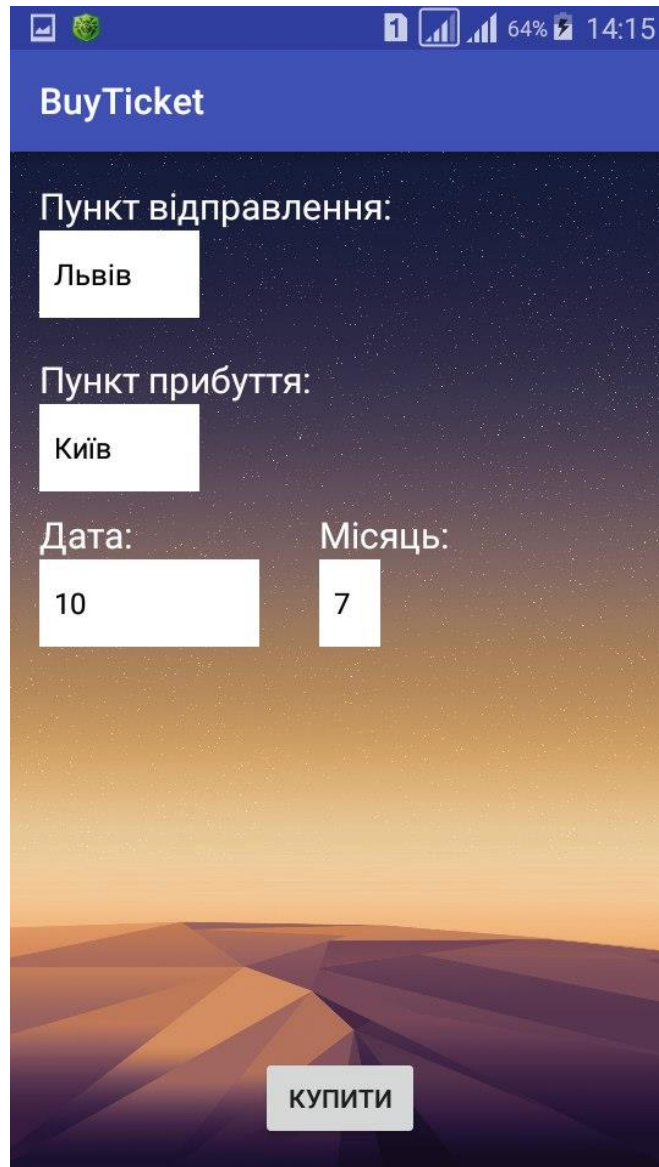


Рис.3.9 Вибір необхідного рейсу

Після того як користувач вибрав необхідний рейс, вибирається місце у салоні. На рис. 3.10 зображений приклад вибору вільного місця у салоні потягу. Усі заброньовані або куплені місця зображені сірим кольором на схемі вагону, а вільні місця є активними, та зображені чорним кольором.

На рис. 3.10 зображена дана активність з відкритою клавіатурою, для вводу бажаного одного місць.

					<i>123.KI-41.04</i>	<i>Арк.</i>
<i>Зм.</i>	<i>Арк.</i>	<i>№ докум.</i>	<i>Підпис</i>	<i>Дата</i>		52

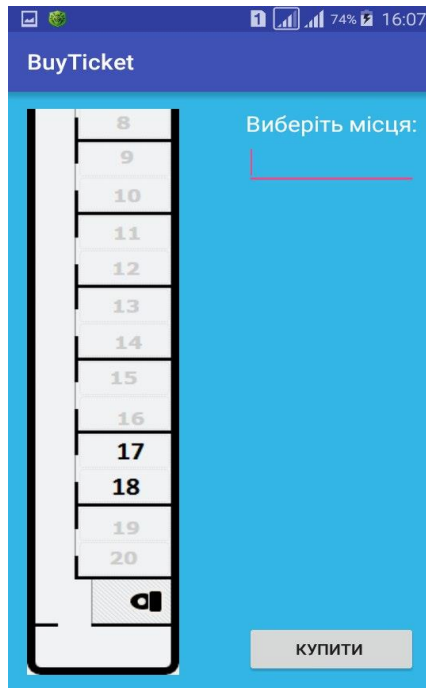


Рис.3.10 Вибір одного місця в салоні транспортного засобу

На рис. 3.11 зображена активність з можливістю вибору декількох місць. Якщо користувач хоче придбати більше одного білета, то йому необхідно лише, через кому, вводити номери вільних місць у салоні транспортного засобу.



Рис.3.11 Вибір декількох місць в салоні транспортного засобу

У TextView користувач вводить бажані вільні місця та нажимає кнопку купівлі білета. Коли користувач вибрав необхідний рейс та місце у салоні, йому пропонується оплатити вибрані білети чи білет за допомогою виклику онлайн інтернет-банкінгу, а саме MasterCard – API, яке надається у вільний доступ усім розробникам програмного забезпечення і дане API є захищеним та безпечним у використанні інтернет-ресурсів.

Дана активність представлена на рис. 3.11 у заповненому вигляді та на рис. 3.12 у незаповненому вигляді.

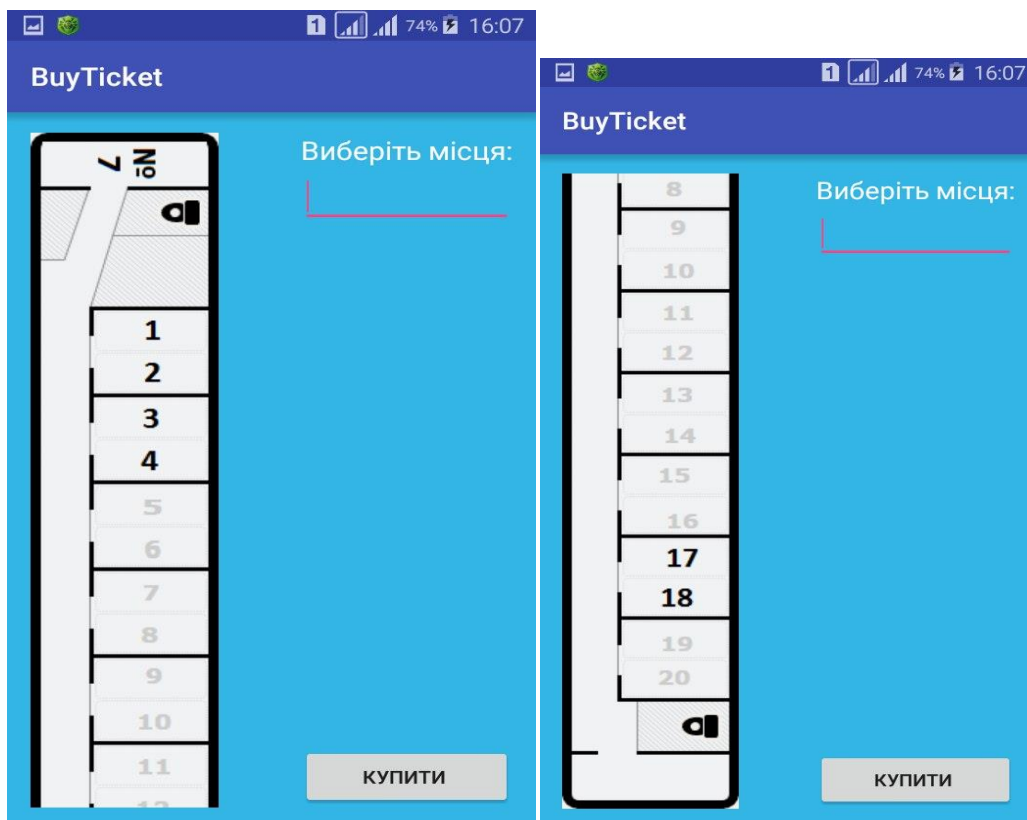


Рис.3.12 Купівля одного та декількох місць

На рис. 3.13 показано встановлений Android-додаток (BuyTicket) на смартфоні.

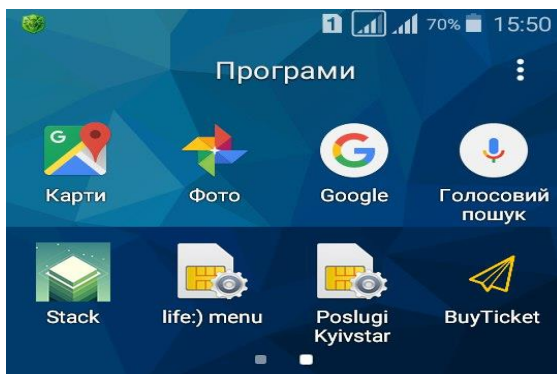


Рис.3.13 Андроїд додаток у меню пристрою

На рис. 3.14 показано процес оплати за придбаний квиток на транспортний звіс із використанням інтернет-банкінгу.

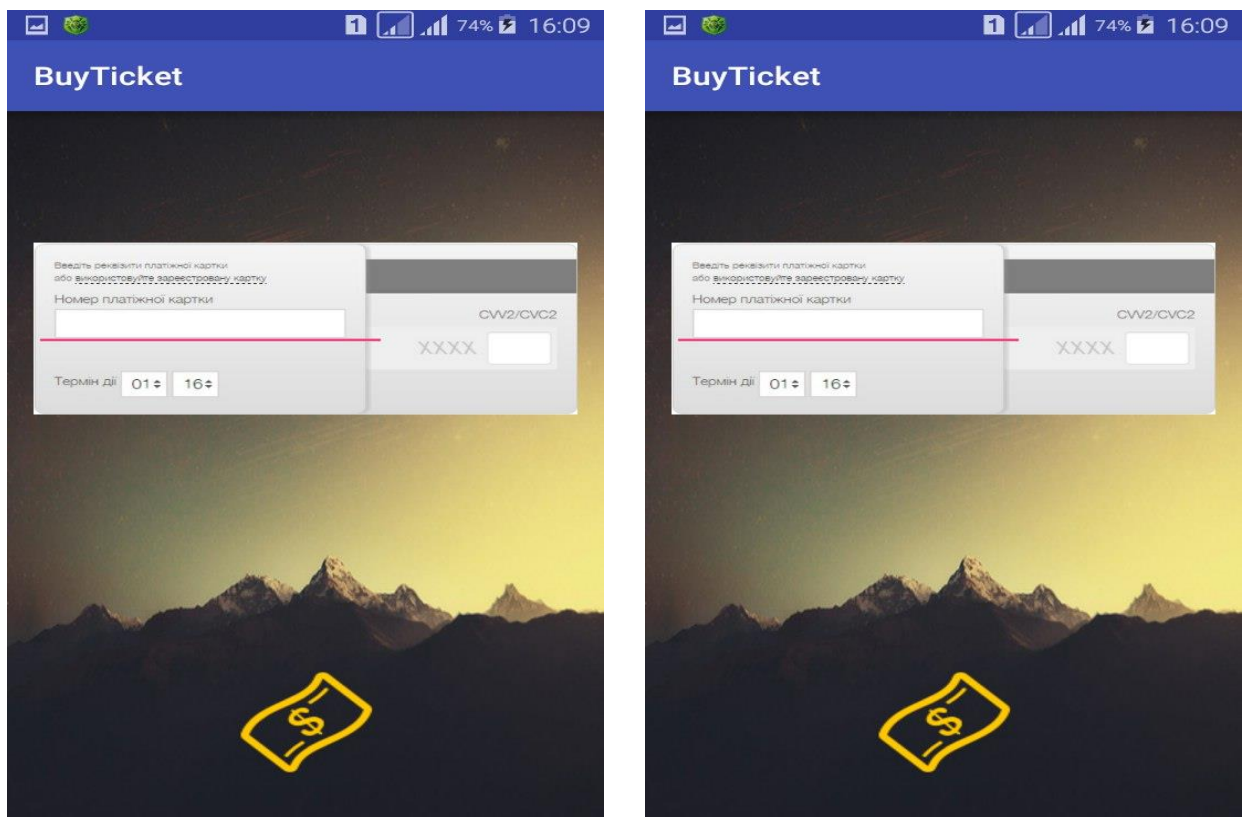


Рис.3.14 Процес оплати вибраного білета

Після успішної оплати вибраних білетів, користувач переходить на останню активність розробленого android -додатку, на якому зображено QR – код білета придбаного та інформація про успішну купівлю необхідного білета. При перевірці білетів кондуктором ,користувач відкриває даний QR – код, після чого його сканує кондуктор і підтверджує достовірність перебування особи в салоні транспортного засобу. Дана активність зображена на рис. 3.15.

					123.KI-41.04	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		55



Рис.3.15 QR -код придбаного білету

Після успішного придбання білету, користувач закриває додаток і продовжує свою діяльність.

На основі виконаної роботи було написано наукову статтю, яка детально висвітлює основні результати та висновки дослідження. Ця стаття була подана для розгляду на науково-технічній дослідницькій конференції, яка відбулася у 2024 році. Під час конференції стаття пройшла етап рецензування, де була високо оцінена провідними фахівцями у відповідній галузі.

Стаття була включена до збірника наукових праць конференції, який був виданий у вигляді книги [25]. Ця книга, що містить найважливіші наукові досягнення, є цінним джерелом інформації для дослідників, науковців та фахівців у галузі. Публікація в книзі підтверджує значущість та актуальність проведеного дослідження, а також сприяє подальшому поширенню знань та результатів роботи серед наукової спільноти.

					123.KI-41.04	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		56

3.2 Реалізація серверної частини спеціалізованої інформаційної системи продажу білетів

3.2.1 Розробка REST – full сервісу

Для того щоб розробити повноцінний REST – full сервіс необхідно насамперед підключити ряд бібліотек, які нададуть можливість роботи з інтернетом та підключення сервера до нього. Для цього з репозиторію Maven було додано ряд бібліотек, а саме Spring Core, Spring Web MVC, для обгортки об'єктів в JSON – Jackson, для роботи із базою даних - Hibernate Tool та ряд інших не менш важливих бібліотек.

Підключення основних бібліотек для побудови REST – архітектури приведені фрагментом коду нижче.

```
<dependencies>
```

```
<dependency>
```

```
<groupId>org.springframework</groupId>
```

```
<artifactId>spring-core</artifactId>
```

```
<version>${spring.version}</version>
```

```
</dependency>
```

```
<!--
```

```
Web-->
```

```
<dependency>
```

```
<groupId>org.springframework</groupId>
```

```
<artifactId>spring-webmvc</artifactId>
```

```
<version>${spring.version}</version>
```

```
</dependency>
```

```
<dependency>
```

```
<groupId>javax.servlet</groupId>
```

```
<artifactId>javax.servlet-api</artifactId>
```

```
<version>${javax.servlet-api.version}</version>
```

```
</dependency>
```

					123.KI-41.04	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		57


```

<!--          JSON          Converter          -->
<dependency>
  <groupId>com.fasterxml.jackson.core</groupId>
  <artifactId>jackson-databind</artifactId>
  <version>${jackson-databind.version}</version>
</dependency>

```

```

<!--          Database          -->
<dependency>
  <groupId>org.springframework.data</groupId>
  <artifactId>spring-data-jpa</artifactId>
  <version>${spring-data-jpa.version}</version>
</dependency>

```

Усі бібліотеки були додані у файл pom.xml, після чого вони були автоматично підтягнуті з інтернету за допомогою Maven збірки проекту та додані у проект у вигляді jar – файлів.

Наступним важливим кроком у побудові веб – серверу було створення та налаштування контролерів. Діаграма послідовності роботи контролера із мого веб – додатку та взаємодія із додатком і базою даних зображена на рис. 3.16.

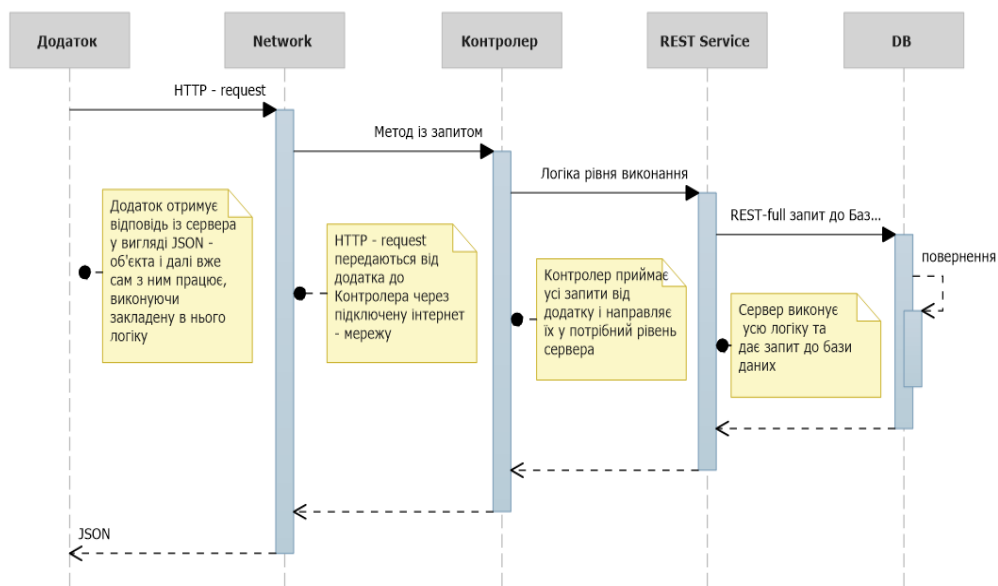


Рис.3.16 Діаграма послідовності взаємодії додатка із сервером

Надзвичайно важливе місце займають анотації, оскільки саме за їх допомогою було налаштовано не лише контролери але й методи для взаємодії з базами даних та інтнет – ресурсами.

Далі наведено приклад основних анотації та їх опис.

```
@RestController
```

```
public class TicketsController {

    @Autowired

    private TicketService service;

    ...
}
```

Дана анотація призначена для визначення контролера в класі та надання йому додаткових методів – контролю.

```
@RequestMapping(value = "/tickets", method =
RequestMethod.GET)
@ResponseBody
public List<Ticked> getAllReminders() {
    return service.getAll();
}
```

Дана анотація призначена для мапінгу (перехоплення) запитів і направлення їх на певний функціонал. Так, запит з /reminders направить у певний метод, відмінний від методу для обробки /reminders/1

```
@Id
@GeneratedValue(generator = "increment")
@GenericGenerator(name = "increment", strategy =
"increment")
private long id;
```

Дана анотація призначена прийому об'єкта із бази даних та перетворення його у зрозумілий для вебу формат - Json. Оскільки база даних не може повертати об'єкти такого типу – це дуже важлива анотація.

Самий сервер побудований за принципом патерну MVC і діаграма послідовності взаємодіє елементів у сервері зображена на рис. 3.17.

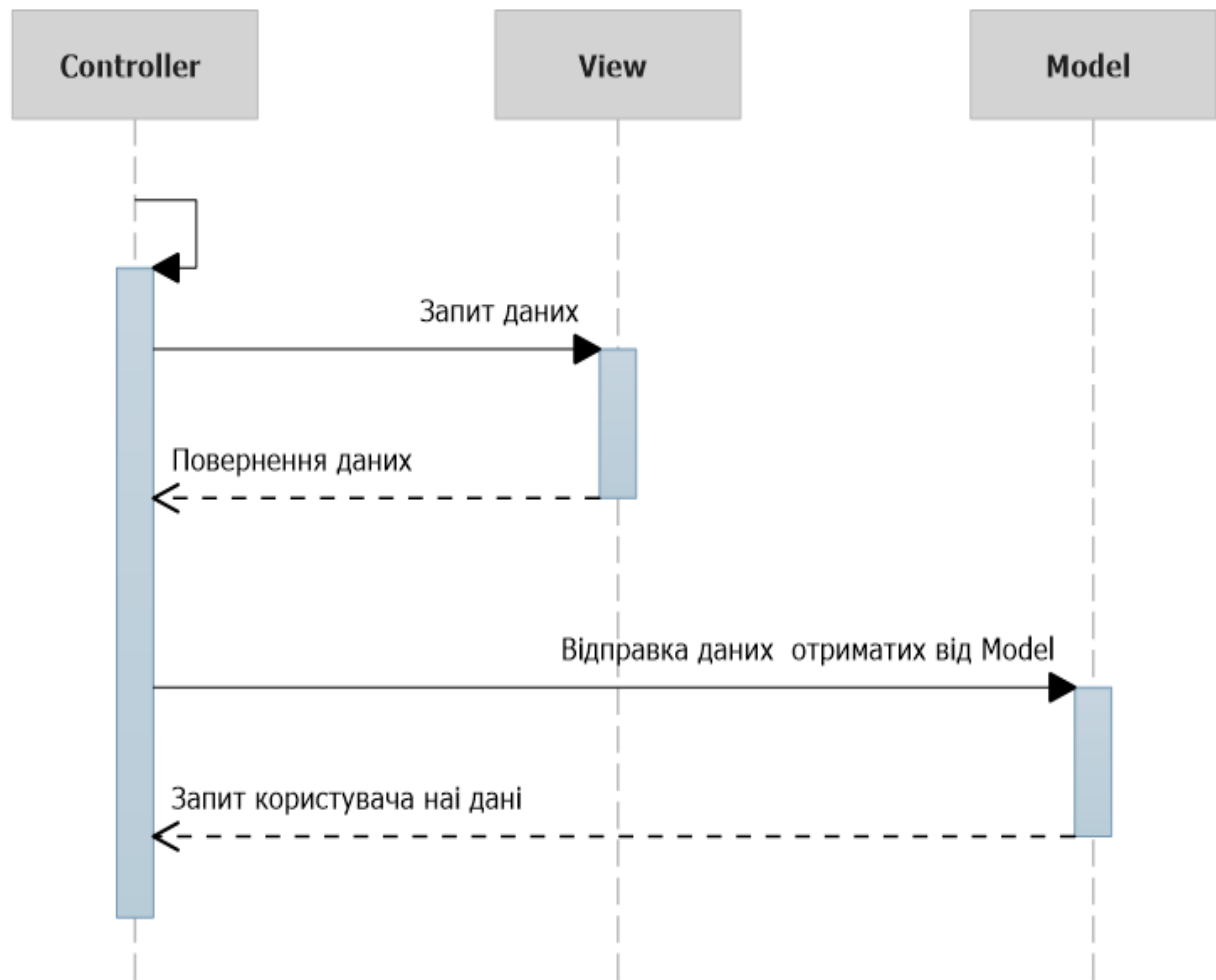


Рис.3.17 Діаграма послідовності взаємодії елементів в сервері

3.2.2 Розробка бази даних для REST – full сервісу

Для повноцінної роботи даного проекту, достатньо реалізувати просту базу даних, яка складається із чотирьох таблиць – можливі транспортні засоби, розклад руху певного виду транспортного засобу, інформація про певний рейс та особисті дані користувача та білета. Кожна із таблиць має свій унікальний ID, PrimaryKey та ForeignKey, які пов’язують таблиці між собою. На рис. 3.18 зображена схема взаємодії таблиць у базі даних між собою.

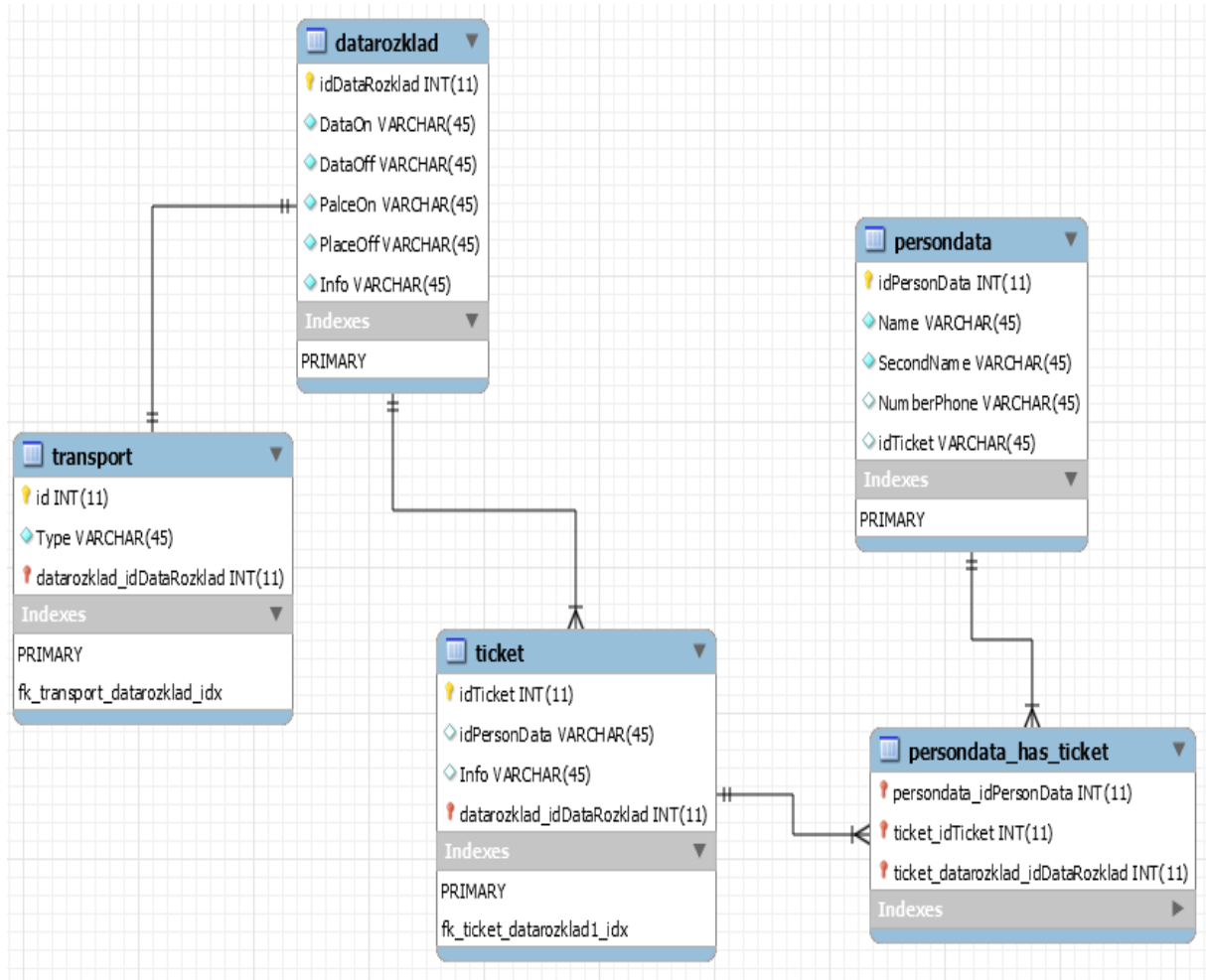


Рис.3.18 Схема взаємодій таблиць у базі даних

Далі зображено інформацію, що міститься в кожній із таблиць. Дану інформацію може змінити лише адміністратор бази даних, тобто людина яка має доступ прав адміністратора. На рис. 3.19 зображено таблицю “Transport”, на рис.3.20 – “Ticket”, на рис.3.21 – “DataRozklad” і на рис. 3.22 зображено таблицю, яка має назву “PersonData”.

id	Type
1	Train
2	Bus
3	Plain
4	Board
5	Velo
6	Onter
NULL	NULL

Рис.3.19 Таблиця бази даних “Transport”

	idTicket	idPersonData	Info
▶	1	1	Ticket_436
	2	2	Ticket_332
	3	3	Ticket_362
★	NULL	NULL	NULL

Рис.3.20 Таблица бази даних “ Ticket ”

	idDataRozklad	DataOn	DataOff	PalceOn	PlaceOff	Info
▶	1	11.06.16	12.06.16	Lviv	Kyiv	1
	2	11.06.16	12.06.16	Odessa	Kyiv	1
	3	11.06.16	12.06.16	Lviv	Kharkiv	1
	4	11.06.16	12.06.16	Odessa	Kharkiv	2
	5	11.06.16	12.06.16	Kharkiv	Kyiv	3
	6	11.06.16	12.06.16	Lviv	Kyiv	4
	7	11.06.16	12.06.16	Lviv	Odessa	2
	8	11.06.16	12.06.16	Lviv	Kyiv	2
★	NULL	NULL	NULL	NULL	NULL	NULL

Рис.3.21 Таблица бази даних “ DataRozklad”

	idPersonData	Name	SecondName	NumberPhone	idTicket
▶	1	Volodymyr	Fedyshyn	0993207981	3
	2	Ivan	Ivanov	0937627090	1
	3	Peter	Drogba	0503621762	2
★	NULL	NULL	NULL	NULL	NULL

Рис.3.22 Таблица бази даних “PersonData”

Процес створення бази даних зображений на рис. 3.23 - рис. 3.29.

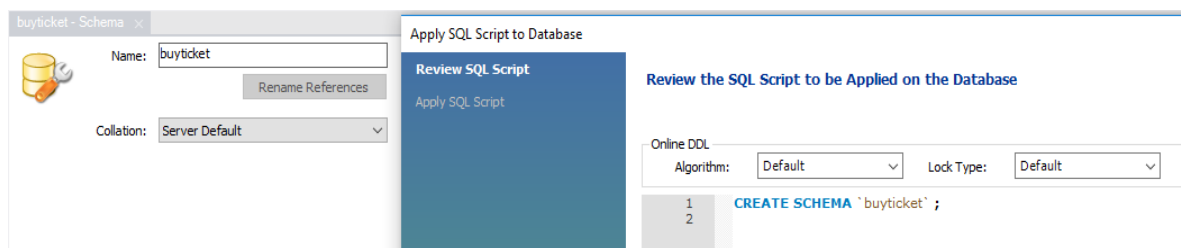


Рис.3.23 Створення бази даних “buyticket”

Online DDL

Algorithm: Lock Type:

```

1 CREATE TABLE `buyticket`.`transport` (
2   `id` INT NOT NULL,
3   `Type` VARCHAR(45) NOT NULL,
4   PRIMARY KEY (`id`))
5 COMMENT = 'Data about tupe of transport.';
6

```

Рис.3.24 Створення таблиці “transport” у базі даних “buyticket”

buyticket - Schema Transport - Table

Table Name: Transport

Column Name	Datatype	PK	NN	UQ	B	UN	ZF	AI	G	C
id	INT	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Type	VARCHAR(45)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

Рис.3.25 Створення таблиці “transport” у базі даних “buyticket”

Online DDL

Algorithm: Lock Type:

```

1 CREATE TABLE `buyticket`.`persondata` (
2   `idPersonData` INT NOT NULL,
3   `Name` VARCHAR(45) NOT NULL,
4   `SecondName` VARCHAR(45) NOT NULL,
5   `NumberPhone` VARCHAR(45) NULL,
6   `idTicket` VARCHAR(45) NULL,
7   PRIMARY KEY (`idPersonData`));
8

```

Рис.3.26 Створення таблиці “persondata” у базі даних “buyticket”

Output

Action Output

	Time	Action
✓	1 08:37:44	Apply changes to BuyTicket
✓	2 08:42:07	Apply changes to Transport
✓	3 08:44:37	Apply changes to PersonData
✓	4 08:45:43	Apply changes to Ticket
!	5 08:46:15	Apply changes to ticket
✓	6 08:47:55	Apply changes to DataRozklad

Рис.3.27 Усі запити до сервера відбулись успішно

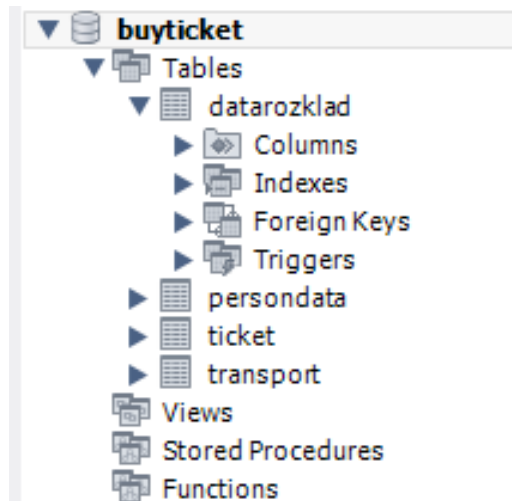


Рис.3.28 Структура бази даних “buyticket”

	idPersonData	Name	SecondName	NumberPhone	idTicket
▶	1	Volodymyr	Fedyshyn	0993207981	3
	2	Ivan	Ivanov	0937627090	1
	3	Peter	Drogba	0503621762	2
*	NULL	NULL	NULL	NULL	NULL

Рис. 3.29 Заповнена таблиця ”persondata” даними у базі даних “buyticket”

З’єднання бази даних та сервера відбувалося за допомогою Hibernate Framework та графічного менеджера DBConnector у IDEA. Процес підєднання БД до сервера зображений на рис. 3.30.

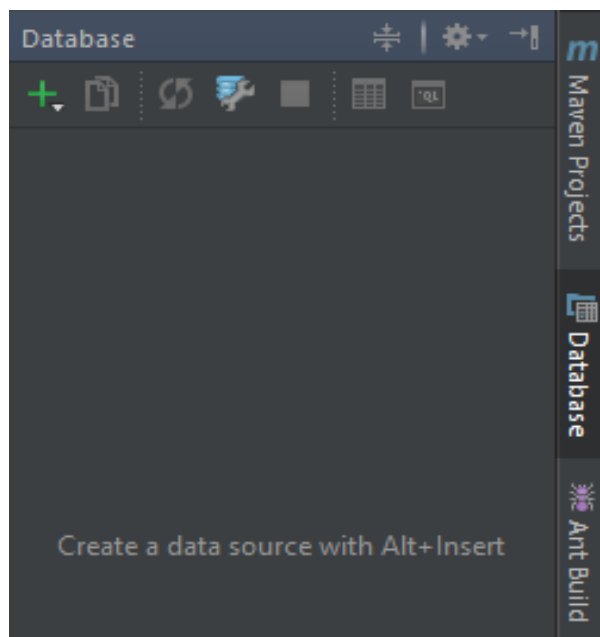


Рис.3.30 Створення з’єднання із сервером та базою даних

Усі драйвера та параметри настроювання бази даних було взято із офіційного сайту підтримки даної бази даних. Наступним кроком було здійснено перевірку успішного підключення до самої бази даних та необхідних портів, вони виявились успішними та не завантаженими іншими потоками даних. На рисунках 3.31 і 3.32 показано процес створення з'єднання із сервером та базою даних.

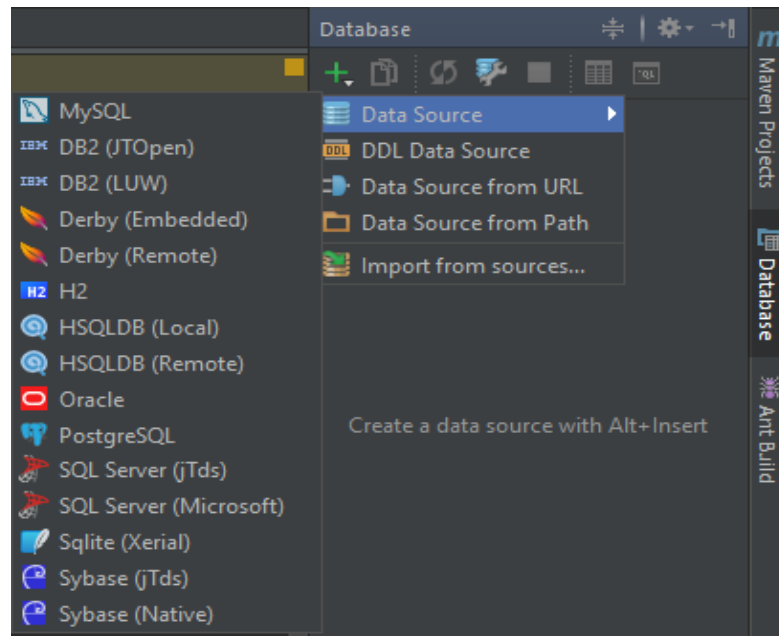


Рис.3.31 Створення з'єднання із сервером та базою даних

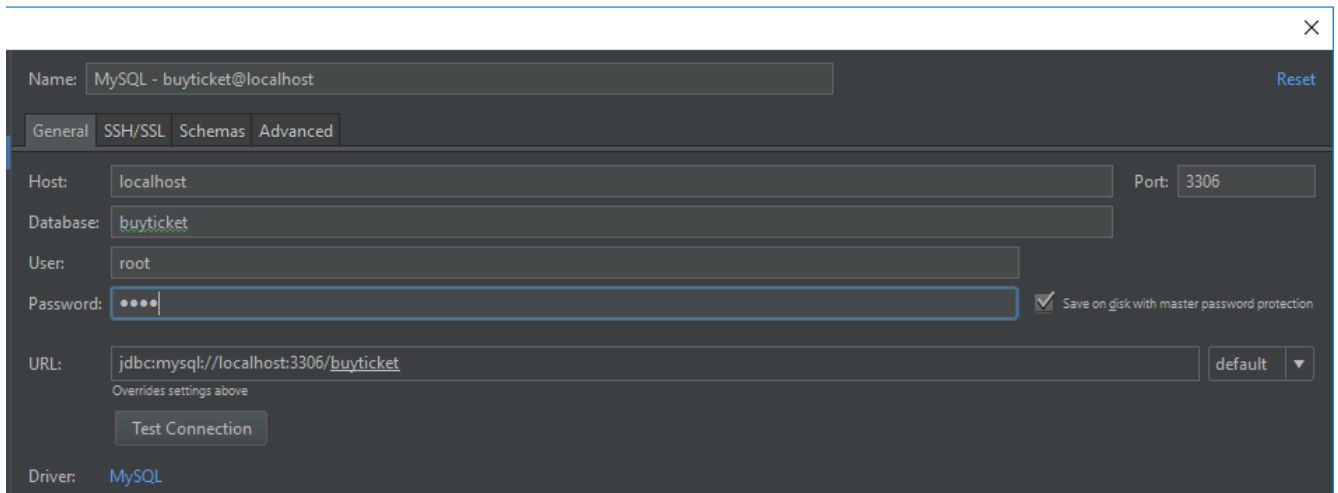


Рис.3.32 Створення з'єднання із сервером та базою даних

Крім цього у папку з ресурсами було додано файли property з налаштуванням драйверів необхідних для підключення бази даних до сервера. У даному файлі було також вказано необхідний діалект, вимоги які представлені до бази даних та ряд інших, менш важливих налаштувань з'єднання бази даних до сервера.

РОЗДІЛ 4. ОХОРОНА ПРАЦІ

4.1 Значення охорони праці для забезпечення безпечних умов праці.

Охорона праці вирішує питання щодо забезпечення здорових і безпечних умов праці та відповідних соціально-економічних, організаційних, технічних та гігієнічних заходів.

Повністю безпечного і нешкідливого виробництва не існує. Завдання охорони праці – мінімізувати можливість травматизму або захворювання, забезпечуючи комфорт і максимальну продуктивність.

Важливість безпеки життєдіяльності залежить від того, що вона спрямована на вирішення важливої соціальної проблеми – збереження здоров'я людини. Покращення умов праці та підвищення їх безпеки має важливі наслідки для навколишнього середовища. Це впливає на економічний результат виробництва - продуктивність, якість, собівартість виготовленої продукції.

Поліпшення умов і безпеки праці може зменшити виробничий травматизм, захворювання, інвалідність, захищаючи тим самим здоров'я працівників, одночасно зменшуючи витрати на пільги та компенсації за роботу у шкідливих умовах для покриття наслідків такої роботи, з причин, пов'язаних з умовами праці, внаслідок персоналу Лікування та перепідготовка працівників виробництва у зв'язку з плинністю кадрів по причинах, що пов'язані з умовами праці.

Охорона праці визначає і вивчає ймовірні причини нещасних випадків на виробництві, професійних захворювань, аварій, вибухів, пожеж, розробляє систему заходів і вимог щодо усунення цих причин та створення безпечних і сприятливих умов праці. Вона повинна захищати працівників від шкідливих і небезпечних факторів виробництва, забезпечуючи максимально сприятливі умови праці, що підвищують продуктивність праці. Тому кожен бізнес повинен розробляти заходи щодо покращення умов та безпеки праці.

У процесі життєдіяльності та виробничої діяльності людина неминує вносить в середовище, що її оточує, певні зміни. Вони стосуються хімічного та біологічного складу навколишнього середовища та його фізичних властивостей. Серед факторів, які можуть негативно впливати на здоров'я людини, є етіологія,

					123.KI-41.04	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		66

тобто причинно-наслідковий зв'язок, що безпосередньо визначає розвиток і тяжкість патологічного процесу, і фактори ризику, які не є причиною захворювання, але сприяють посиленню дії збудника. фактор. Основним нормативним документом з охорони праці для користувачів є ЕОМ ДНАОП 0.00-1.31-99.

4.2 Правила охорони праці під час експлуатації ЕОМ.

У процесі праці людина взаємодіє з предметом праці, засобами виробництва, іншими людьми. Крім того, на нього впливають різні параметри виробничого середовища, в якому відбувається робота - температура, вологість, шум, рух повітря, вібрація, інфразвук, вплив електромагнітних полів і різних видів випромінювання. Від умов праці значною мірою залежать здоров'я і працездатність людини, настрої до праці та результати праці. При поганих умовах створюються передумови для виникнення травм і професійних захворювань.

Серед факторів виробництва прийнято розрізняти шкідливі та небезпечні фактори. Небезпеки – це фактори виробництва, вплив яких на працівників може призвести до травм або раптового та різкого погіршення здоров'я. Фактор виробництва вважається шкідливим, якщо викликає захворювання або знижує ефективність.

До груп шкідливих продуктивних факторів трудового процесу належать фізичні перевантаження (статичні, динамічні), нервово-психічні перевантаження (психічна напруга, сенсорне напруження, монотонність роботи, емоційні перевантаження).

Процес праці здійснюється за певних умов виробничого середовища. Це низка факторів, які впливають на фізичні, хімічні та біологічні властивості людини разом із соціально-економічними факторами в процесі роботи людини. Фактори виробничого середовища та трудового процесу, також відомі як психофізіологічні чинники, разом становлять умови праці.

До найважливіших шкідливих фізичних факторів належать: підвищена запиленість повітря робочої зони, мікроклімат промислових об'єктів, підвищений рівень інфрачервоного випромінювання, підвищений рівень ультрафіолетового

					<i>123.KI-41.04</i>	<i>Арк.</i>
<i>Зм.</i>	<i>Арк.</i>	<i>№ докум.</i>	<i>Підпис</i>	<i>Дата</i>		67

випромінювання, підвищений рівень вібрації, шуму, інфразвуку та ультразвукових хвиль, підвищення або зниження тиск повітря, підвищений рівень статичної електрики, небезпечні рівні напруги в електричних ланцюгах, підвищена або знижена іонізація повітря, нестача або відсутність природного світла, відсутність або широкі зміни в освітленні робочих місць, підвищена яскравість, знижена контрастність, пульсація світла.

Використання комп'ютерної техніки характеризується шкідливим впливом електромагнітного випромінювання (ЕМВ) від комп'ютерних екранів та дисплеїв смартфонів. Ступінь впливу такого випромінювання на організм людини залежить від частотного діапазону, інтенсивності, тривалості та характеру випромінювання (безперервного чи модульованого), розміру опромінюваної поверхні та особливостей організму. Біологічна дія низькочастотної ЕРС, що виробляється екраном та дисплеєм смартфона, може призвести до порушення функцій центральної нервової та серцево-судинної систем організму, а також певних змін у складі крові, особливо при високій напруженості електромагнітного поля.

Тривалий вплив низькоінтенсивного ЕМВ може викликати різні неврологічні та серцево-судинні розлади (головний біль, втома, порушення сну, біль у серці тощо). Можливі ендокринні порушення та зміни складу крові.

Крім того, оператори ЕОМ виконують велику сидячу роботу у вимушеному положенні гнучких рук. Надмірне напруження очей, що може призвести до болю в очах, головного болю. Результатом сидячої роботи є болі в спині.

Під час роботи з телефоном на організм людини впливають різноманітні фактори фізичної природи, більшість з яких відповідно до чинних нормативних актів знаходиться в межах норми. Дані вимірювання шкідливого випромінювання екрану та дисплею наведені в таблиці 4.1.

					<i>123.KI-41.04</i>	<i>Арк.</i>
<i>Зм.</i>	<i>Арк.</i>	<i>№ докум.</i>	<i>Підпис</i>	<i>Дата</i>		68

Таблиця 4.1 Аналіз потенційно-небезпечних виробничих факторів

Вид випромінювання	Діапазон	Фактичні (середні дані замірів)	Нормативні значення
Рентгенівське	Понад 1,2 Кев	12-13 мкр/год	75 мкр/год
Ультрафіолетове випромінювання	220-280 нм	0	0,01 Вт/м ²
			0,01 Вт/м ²
Видимий діапазон	280-320 нм	0-0,03	10 Вт/м ²
Інфрачервоне випромінювання	320-700 нм	0,05-4,2	100 Вт/м ²
Електростатичне поле	0 Гц	16 кВм	20-60 кВ/м ²
Електрообладнання		U=220В, I=2А, f=50Гц	42 В, I < 0.1 А, f=50 Гц
Яскравість		60 кД/м ²	>30 кД/м ²

Негативний вплив екранів та дисплеїв на зір проявляється сльозотечею, печією, болем в очах, ломоти в надбрівній області, розмитими межами зображення предметів, які викликані тимчасовим порушенням світлочутливого пристрою ока.

Ці явища зазвичай супроводжуються головним болем, важкою головою, загальною втомою, сонливістю і млявістю. Щоб цього не сталося, необхідно дотримуватися правил при виборі приміщення, робочого місця, дотримуватися рекомендацій з гігієни та освітлення, викладених у документі про безпечне поводження з електронно-обчислювальними машинами.

4.3 Забезпечення нормальних умов праці.

Згідно з гігієнічною класифікацією праці умови праці поділяються на чотири класи:

- I клас - оптимальні умови - такі умови не тільки захищають здоров'я працівників, а й створюють передумови для підтримки високого рівня працездатності. Оптимальні критерії факторів виробництва встановлені для

мікроклімату та факторів трудового процесу. Щодо інших факторів, оптимальними умовами праці є виробниче середовище, яке не є більш ворожим, ніж те, що вважається безпечним.

- II клас – прийнятні умови – характеризуються факторами виробництва та трудовими процесами, які не перевищують встановлених гігієнічних норм і, якщо це можливо, зміни фізичного функціонального стану відновлюються під час регулярних перерв або до наступного негативного впливу на здоров'я їх нащадків.
- III клас — шкідливі умови — характеризуються такими рівнями шкідливих виробничих факторів, які перевищують гігієнічні нормативи і здатні чинити несприятливий вплив на організм працюючого та (або) його майбутніх нащадків. Шкідливі умови за показниками перевищення гігієнічних нормативів та вираженості можливих змін в організмі працюючих поділяються на чотири ступеня:
- Клас III – шкідливі умови – характеризуються рівнем шкідливих виробничих факторів, що перевищують гігієнічні норми та здатні негативно впливати на організм працівника та/або його майбутні покоління. Небезпечні умови поділяються на чотири стадії за показниками перевищення гігієнічних норм і тяжкості можливих змін в організмі працівників:
- 1-й ступінь – умови праці, що характеризуються таким високим рівнем шкідливих факторів у виробничому середовищі та в процесі праці, що це часто викликає функціональні зміни за межі фізіологічних коливань (останні відновлюються при тривалішій, ніж початок наступної зміни перерві, контакту зі шкідливими факторами) та збільшують ризик погіршення здоров'я;
- 2-й ступінь — умови праці характеризуються такими рівнями шкідливих факторів виробничого середовища і трудового процесу, які здатні викликати стійкі функціональні порушення, призводять переважно до зростання професійно зумовленої захворюваності, виникнення окремих ознак або легких форм патології (як правило, без втрати професійної працездатності);

					<i>123.KI-41.04</i>	<i>Арк.</i>
<i>Зм.</i>	<i>Арк.</i>	<i>№ докум.</i>	<i>Підпис</i>	<i>Дата</i>		70

- 3-й ступінь — умови праці характеризуються такими рівнями шкідливих факторів виробничого середовища і трудового процесу, що призводять, крім зростання професійно зумовленої захворюваності, до розвитку професійних захворювань, як правило, легкого та середнього ступенів важкості (із втратою працездатності в період трудової діяльності);
- 4-й ступінь — умови праці характеризуються такими рівнями шкідливих факторів виробничого середовища і трудового процесу, які здатні призводити до значного зростання хронічної патології та рівнів захворюваності з тимчасовою втратою працездатності, а також до розвитку важких форм професійних захворювань із втратою загальної працездатності.

Забезпечення здорових та безпечних умов праці у виробничій сфері досягається коштом дотримання чинних нормативних документів, для чинних об'єктів шляхом встановлення значень, що не суперечать нормативним.

Вимоги до мікроклімату. Трудова діяльність людини завжди здійснюється за певних метеорологічних умов, які спільно визначаються температурою повітря, швидкістю вітру, відносною вологістю, атмосферним тиском, тепловим випромінюванням нагрітої поверхні.

Всі види робіт у виробничих приміщеннях поділяють на 3 категорії по важкості робіт. Робота оператора ЕОМ відноситься до категорії робіт з затратами енергії до 150 ккал/год, це згідно з ДСТУ-Н Б А. 3.2-1:2016 відповідає категорії – легка – 1 б.

Мікроклімат виробничих приміщень або метрологічні умови визначаються такими параметрами: температурою повітря t , °С; Відносною вологістю ϕ ,%; швидкість руху повітря на робочому місці V , м/с.

Параметри мікроклімату, іонного складу повітря, вмісту шкідливих речовин на робочих місцях, що комплектуються, повинні відповідати вимогам пункту 2.4. ДСН 3.3.6.042-99 „Санітарні норми мікроклімату виробничих приміщень”; ДСанПіН 5.5.2.008-01 „ Санітарно- гігієнічні норми допустимих рівнів іонізації повітря виробничих та громадських приміщень” і наведені в таблиці 4.2 та таблиці 4.3.

					<i>123.KI-41.04</i>	<i>Арк.</i>
<i>Зм.</i>	<i>Арк.</i>	<i>№ докум.</i>	<i>Підпис</i>	<i>Дата</i>		71

Таблиця 4.2 Нормовані параметри мікроклімату для приміщень з ЕОМ для категорії робіт 1-б

Пора року	Оптимальна температура повітря, °С	Оптимальна відповідна вологість повітря, %	Оптимальна швидкість руху повітря, м/с
Холодний	21-23	40-60	0,1
Теплий	22-24	35-50	0,1

Таблиця 4.3 Рівні іонізації повітря при роботі в приміщенні з ЕОМ

Рівні	Кількість іонів в 1 см ³ повітря, n	
	n+	n-
Мінімально необхідні	400	600
Оптимальні	1500-300	3000-5000
Максимально допустимі	50000	50000

Вимоги до освітлення. Відповідно до ДНАОП 0.00-1.31-99 освітлення в приміщенні з ЕОМ має бути суміщеним.

Для оцінки освітленості використовуються наступні значення: люкс - рівень освітленості поверхні площею 1 квадратний метр, на яку падає, розподіляючи світловий потік рівномірно на 1 люмен; люмени - від джерела світла контрольної точки до міжнародної свічки Luminous потік, у верхній частині тіла, кут 1 стерadian; Болт - Яскравість світної поверхні, світло рухається у вертикальному напрямку із силою 1 свічка і 1 квадратний метр; Свічка - Потужність точкового джерела світла свічки, що випромінює світловий потік 1лм, Рівномірно розподілений посередині тіла під кутом 1 сферичний кут.

Відчуття світла при дії на око людини викликають електромагнітні хвилі так званого оптичного діапазону.

Область оптичних електромагнітних випромінювань розташована між областю рентгенівського випромінювання та областю радіовипромінювання.

					123.KI-41.04	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		72

Гігієнічні вимоги до виробничого освітлення ґрунтовані на психофізичних особливостях сприйняття світла і його впливу на організм людини.

Освітлення робочого місця може здійснюватися за допомогою природного і штучного освітлення. Якщо природного освітлення недостатньо, використовуйте загальне освітлення. Останнім є освітлення, при якому використовується природне та штучне освітлення. Зазвичай називають освітленням, світильники, які висвітлюють всю площу кімнати.

Природне освітлення повинно проникати через бічні світло прорізи, зорієнтовані як правило на північ, чи північний схід і забезпечувати коефіцієнт природної освітленості не нижче 1,5%. Розрахунки проводять згідно ДБН В.2.5-28-2018.

При виробничій потребі дозволяється експлуатувати ЕОМ у приміщеннях без природного освітлення за узгодженням з органами державного нагляду за охороною праці та органами і уповноваженими санітарно епідеміологічної служби.

Рівень освітлення в області документа робочого столу повинен бути в діапазоні 300-500 люкс. Якщо такий рівень освітленості забезпечити неможливо, системи загального освітлення припускають використання світильників точкового освітлення, але на поверхні екрану та дисплею не повинно бути відблисків і збільшити яскравість екрану та дисплея до понад 300 люкс.

Розрахунок системи освітлення виконується з використанням коефіцієнта використання світлового потоку, який дорівнює відношенню світлового потоку, що падає на розрахункову поверхню, до загального потоку всіх світильників. Його величина залежить від характеристик світильника, розміру приміщення, кольору стін і стель, характеризується коефіцієнтом відбиття стін і стель.

					<i>123.KI-41.04</i>	<i>Арк.</i>
<i>Зм.</i>	<i>Арк.</i>	<i>№ докум.</i>	<i>Підпис</i>	<i>Дата</i>		73

Для дорослих користувачів робота за комп'ютером чи смартфоном не повинна перевищувати шести годин на день. Для будь-якого дорослого користувача монітор не можна використовувати більше двох годин безперервно з наступною 15-20-хвилинною перервою.

Деякі відомості про шкідливу дію ЕОМ на людину приведені в таблиці 4.4 по таблиці видно, що із збільшенням тривалості роботи на ЕОМ число скарг на самопочуття зростає.

Таблиця 4.4 Відомості про шкідливу дію ЕОМ на людину

Симптоми впливу комп'ютера	Відсоток операторів, які повідомили про симптоми			
	Неповна зміна, робота за дисплеями до 12 міс.	Повна зміна, робота за дисплеями до 12 міс.	Робота за дисплеями більше 12 міс.	Робота за дисплеями більше 2 років
Головна біль та біль в очах	10	37	53	78
Перевтома, запаморочення	7	34	43	71
Підвищена дратівливість	5	13	24	53
Депресія	5	18	24	52

Оскільки електромагнітне випромінювання виходить від всіх частин екрана (багато вимірювань показали, що рівень випромінювання з боків й позаду екрана вище, ніж з переду), найбільш безпечно встановити комп'ютер в кутку кімнати або в такому місці, де ті, хто на ньому не працює, не знаходяться з боку або ззаду від машини.

Приміщення, де знаходяться комп'ютери, повинне бути достатньо просторим і добре провітрюватись. Мінімальна площа на один комп'ютер - 6 м², мінімальний об'єм - 20 м³.

Ергономічні, гігієнічні й фізкультурно-оздоровчі заходи дозволяють всім хто працює з комп'ютерами зменшити майже до нешкідливого рівня негативну дію комп'ютера на здоров'я.

Основним заходом профілактики захворювань, пов'язаних з комп'ютером, є оптимальна організація робочого місця, тобто використання таких засобів, як комп'ютерні столи, стільці, які можуть змінювати висоту та кут нахилу спинки, пристрої введення тексту тощо. типу. Екран чи дисплей повинен бути забезпечений підставкою, яка може змінювати кут нахилу. Екран чи дисплей повинен знаходитися на відстані більше 45 см від ока, бажано нахилити, щоб користувач міг бачити поверхню екрана чи дисплея під кутом 5-35 градусів.

Рекомендується встановити кут між клавіатурою і робочим столом в межах 5-10 градусів. Серед заходів профілактики захворювань м'язів і опорно-рухового апарату в основному виділяють: зручність сидіння, тобто висота сидіння і відповідність природному вигину хребта. Під час роботи на комп'ютері потрібне правильне місце користувача. Це означає, що потрібно сидіти прямо, не нахилитися, рівень зап'ястя повинен бути на відстані витягнутої руки, стопи повинні бути на підлозі або стояти прямо, а не схрещені.

Кут нахилу стоп до поверхні підлоги повинен становити 5°-15°. Вологість у приміщенні повинна бути в межах 40-60%. На завершення цього розділу можна сказати, що причиною проблем зі здоров'ям користувачів є не стільки сам комп'ютер, скільки недотримання ергономічних принципів і гігієнічних вимог.

4.4 Система електроживлення та заходи з електробезпеки.

Під час проектування систем електропостачання, монтажу силового електрообладнання та електричного освітлення будівель та приміщень для ЕОМ необхідно дотримуватися вимог ПВЕ, ПБЕ, ДСТУ-Н Б В. 2.5-83:2018 „Інструкція по проектуванню силового освітительного обладнання промислових підприємств”, ДСТУ Б А. 3.2-13:2018 Електробезпека. Захисне заземлення, занулення”, ДСТУ 7237:2011, „ССБП Електробезпека. Загальні вимоги і номенклатура видів захисту. Норми захисту Правил пожежної безпеки в Україні, а також розділів СНиП, що стосуються штучного освітлення і електротехнічних

					123.KI-41.04	Арк.
						75
Зм.	Арк.	№ докум.	Підпис	Дата		

пристроїв, та вимог нормативно – технічної і експлуатаційної документації заводу-виробника ЕОМ.

Лінії живлення комп'ютерної периферійної комп'ютерної техніки та комп'ютерної техніки з обслуговування, ремонту та налагодження виконуються як окремий комплекс трипровідних мереж шляхом прокладання фазних, нульових і нульових захисних провідників. Для заземлення використовується нульовий захисний провідник.

У приміщенні, де одночасно працює більше п'яти персональних комп'ютерів, електроживлення приміщення, крім освітлення, має бути встановлено на видному та легкодоступному місці.

Крім фазних і нульових контактів для підключення нульового захисного провідника, штепсельні з'єднання та розетки повинні мати спеціальні контакти. Вони повинні бути сконструйовані таким чином, щоб підключення нульового захисного провідника відбувалося до з'єднання фазного провідника та нульового робочого провідника.

Порядок роз'єднання при відімкненні повинен бути зворотнім. Необхідно унеможливити з'єднання контактів фазових провідників з контактами нульового захисного провідника.

Неприпустимим є підключення ЕОМ, периферійних пристроїв ЕОМ та устаткування для обслуговування, ремонту та налагодженням ЕОМ до звичайної двопровідної, в тому числі з використання переходних пристроїв.

Є недопустимим:

- експлуатація кабелів та проводів з пошкодженою або такою, що втратила захисні властивості за час експлуатації, ізоляцією;
- залишення під напругою кабелів та проводів з неізольованими провідниками;
- застосування саморобних подовжувачів, що не відповідають вимогам ПВЕ до переносних електропроводок;
- застосування для опалення приміщення нестандартного (саморобного) електронагрівального обладнання або ламп розжарення;

					<i>123.KI-41.04</i>	<i>Арк.</i>
<i>Зм.</i>	<i>Арк.</i>	<i>№ докум.</i>	<i>Підпис</i>	<i>Дата</i>		76

- користування пошкодженими розетками, розгалужуваними чи з'єднувальними коробками, вимикачами чи іншими електровиробами, а також лампами, скло яких має сліди затемнення або випинання;
- підвішування світильників безпосередньо на струмопроводних проводах, обгортання світильників чи ламп папером, тканиною та іншими горючими матеріалами;

У відповідності до ДСТУ 3635-98 засоби і системи контролю доступу відносяться по класифікації технічних засобів до категорії 4 по за умовами експлуатації до групи Е. Рівень допустимих радіоперешкод при роботі засобів і систем контролю доступу повинен відповідати ДСТУ Б В.2.5-30:2018.

4.5 Пожежна безпека.

Першочерговим завданням протипожежної профілактики є запобігання пожежі. Протипожежний захист — це комплекс організаційно-технічних заходів, призначених для забезпечення безпеки людей, запобігання пожежам, обмеження їх поширення, створення умов для успішного гасіння пожежі. Заходи протипожежного захисту розробляються і впроваджуються в комплексі з усіма проектними, будівельними та експлуатаційними роботами.

Приміщення, надане для технічних працівників, обладнане протипожежним обладнанням (вогнегасник вуглекислотний типу ВВ-2). Прохід між рядом і розеткою не повинен бути перекритий. У разі пожежі перше, що потрібно зробити, це відключити електроживлення та повідомити пожежну частину.

Евакуювати сторонніх людей, які могли опинитися в небезпечній зоні й лише після цього приступити до гасіння пожежі та рятування цінного обладнання.

Вогнегасник CO2 ВВ-2 розрахований на площу 40-50 кв. Для гасіння невеликих пожеж можна використовувати деякі порошкоподібні матеріали (хлориди лужних металів, соду, пісок тощо), які за допомогою порошкового вогнегасника подавати на місце горіння. Будівля, в якій розташована ділянка, повинна містити запасний вихід на випадок екстреної евакуації працівників і неможливості використання основного виходу.

Електрична пожежна сигналізація використовується для швидкого оповіщення пожежної охорони у разі виникнення пожежі в приміщенні. Електричні системи пожежної сигналізації виявляють пожежі та повідомляють про їх місце на початковому етапі та автоматично включають стаціонарні системи пожежогасіння.

Автоматичні сповіщувачі при ознаках пожежі здійснюють посилення сигналу. Сповіщувачі типу АТИП-1, АТИП-3 і АТИП ЗМ спрацьовують внаслідок теплової деформації (при 80—100 °С) біметалевих пластинок і мають розраховану площу обслуговування в приміщеннях до 15 м². Комбіновані теплові і димові сповіщувачі типу КИ-1 мають чутливий елемент у вигляді іонізуючої камери (реагування на дим) і терморезистори (реагування на теплоту). Температура спрацювання цих сповіщувачів 50—80 °С, площа обслуговування 100 м².

Передбачені заходи з охорони праці відповідають вимогам нормативних документів і актів та забезпечують нормальну, ефективну і безпечну для здоров'я людини виробничу діяльність.

Профілактичні заходи проти загорання:

- організаційні — правильна експлуатація приладів, інструменту, паяльників і належне утримання робочих місць та лабораторії в цілому, протипожежний інструктаж;
- технічні — додержання протипожежних правил та норм при проектуванні будівлі, при влаштуванні електропроводки, опалення, вентиляції, освітлення, доцільне розміщення обладнання;
- режимного характеру — заборона куріння в невідповідних місцях;
- експлуатаційні — своєчасні профілактичні огляди, ремонти та випробування.

Автоматичні та кнопкові сповіщувачі розташовані в кожній лабораторії, а також у коридорах біля пожежних шаф. Для гасіння невеликих пожеж підприємства використовують основні методи пожежогасіння: вогнегасники, азбестові ковдри, вогнегасники внутрішнього протипожежного водопостачання.

					<i>123.КІ-41.04</i>	<i>Арк.</i>
<i>Зм.</i>	<i>Арк.</i>	<i>№ докум.</i>	<i>Підпис</i>	<i>Дата</i>		78

Протипожежні стовбури розміщені в протипожежних шафах. На кожному поверсі по дві шафи. Крім того, до кожної шафи йдуть окремі сходові клітки.

Шляхи евакуації дозволили всім працівникам вчасно покинути зону пожежі. Слід зазначити, що для даної категорії виробництва при ступені вогнетривкості 1 або 2 для багатопверхового будинку віддалі до евакуаційного виходу не нормуються.

Охорона праці дуже важлива для забезпечення безпечних і здорових умов праці. Тому врахування вимог фізичного, психологічного та професійного здоров'я має велике значення у створенні оптимальних умов для охорони та вдосконалення виробничих процесів. Забезпечити безпеку праці, зменшити надмірне навантаження на працівників та допомогти більш ефективно використовувати людські ресурси, тим самим підвищивши ефективність.

Складність завдання, що стоїть перед охороною праці, вимагає використання результатів і висновків багатьох наукових дисциплін, прямо чи опосередковано пов'язаних із завданням створення здорових і безпечних умов праці. Насамперед це стосується суспільних наук, а також досліджень у галузі наукової організації праці, технічної естетики, ергономіки, соціальної та інженерної психології.

Охорона праці є суттєвим елементом процесу виробництва. Тому вона має як організаційно-технічні, так і соціальні аспекти. Вона покликана вберігати персонал від впливу небезпечних і шкідливих факторів, забезпечувати найсприятливіші умови праці, забезпечувати підвищення продуктивності праці.

					<i>123.KI-41.04</i>	<i>Арк.</i>
<i>Зм.</i>	<i>Арк.</i>	<i>№ докум.</i>	<i>Підпис</i>	<i>Дата</i>		79

РОЗДІЛ 5. ЕКОНОМІЧНА ЧАСТИНА

5.1 Розрахунок часу на виготовлення програмного продукту.

В таблиці 5.1 наведено перелік основних видів робіт по розробці інформаційної програмної системи для продажу квитків на різні види транспорту на базі ОС Android.

Таблиця 5.1 Перелік робіт

№	Робота	Тривалість, днів	Виконавець
1	Написання та аналіз технічного завдання	1	Консультант
2	Аналіз аналогів	3	Розробник
3	Проектування системи класів та залежностей	3	Розробник
4	Реалізація серверної частини спеціалізованої системи продажу білетів	3	Розробник
5	Проектування бази даних	3	Розробник БД
6	Розробка web-додатку	6	Дизайнер
7	Розробка елементів web-додатку	4	Дизайнер
8	Написання основних класів роботи системи	5	Розробник
9	Розробка модулів користувача	2	Розробник
10	Розробка модулів ядра	16	Розробник
11	Налаштування android-додатку	3	Розробник
12	Тестування android-додатку	5	Тестувальник

Всього на розробку проекту було витрачено 54 дні. Кількість днів, відпрацьованих кожним виконавцем, зведені в таблиці 5.2.

					123.KI-41.04	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		80

Таблиця 5.2 Кількість днів відпрацьованих кожним виконавцем

Працівник	Кількість відпрацьованих днів
Розробник	35
Дизайнер	10
Тестувальник	5
Розробник БД	3
Консультант	1

Для планування і керування розробкою проекту побудовано мережевий діаграм, який відповідає переліку робіт в таблиці 5.1. В мережевому графіку детально показано в якій послідовності, коли та за який час відбувається реалізація проекту.

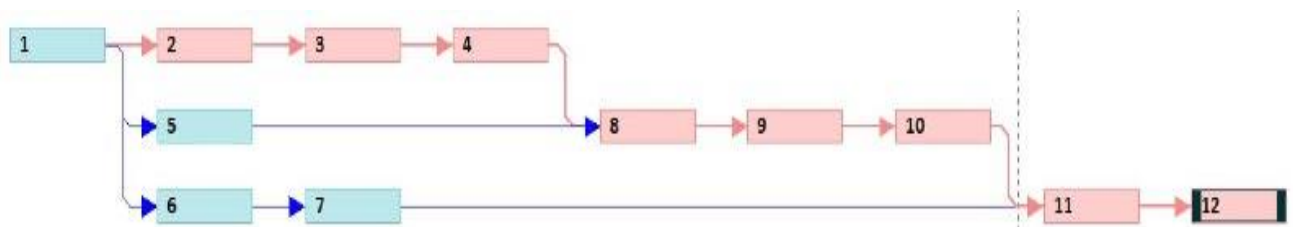


Рис.5.1 Мережевий граф

Критичний шлях – це найбільш тривала по часу послідовність робіт, які ведуть від вихідної до завершальної події. Величина критичного шляху визначає термін виконання всього комплексу планування робіт. Зміна тривалості будь-якої роботи, що лежить на критичному шляху, відповідним чином змінює термін настання завершальної події, тобто дату досягнення кінцевої мети.

Як видно з рис. 5.1 критичним шляхом є роботи під номером 1, 2, 3, 7, 10 та 15, так як вони мають найдовший час виконання.

Розгорнутий мережевий графік з ранніми, пізніми термінами та резервами зображено на рис. 5.2, рис. 5.3 та рис. 5.4.

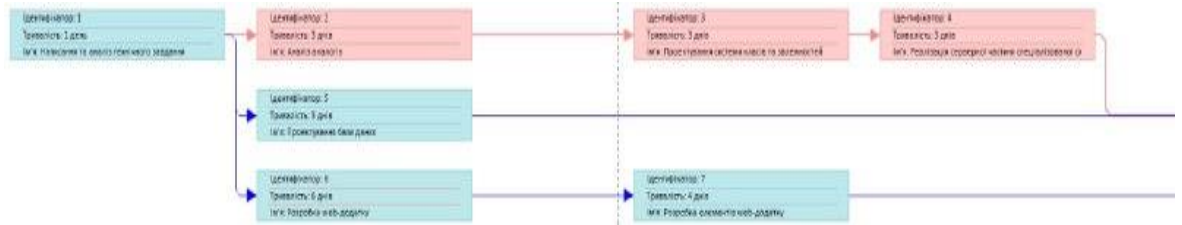


Рисунок 5.2 Розгорнутий мережевий графік

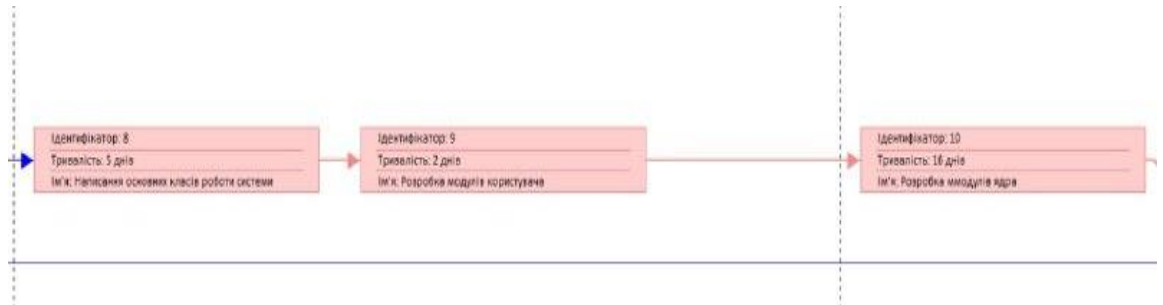


Рис.5.3 Продовження розгорнутого мережевого графіку

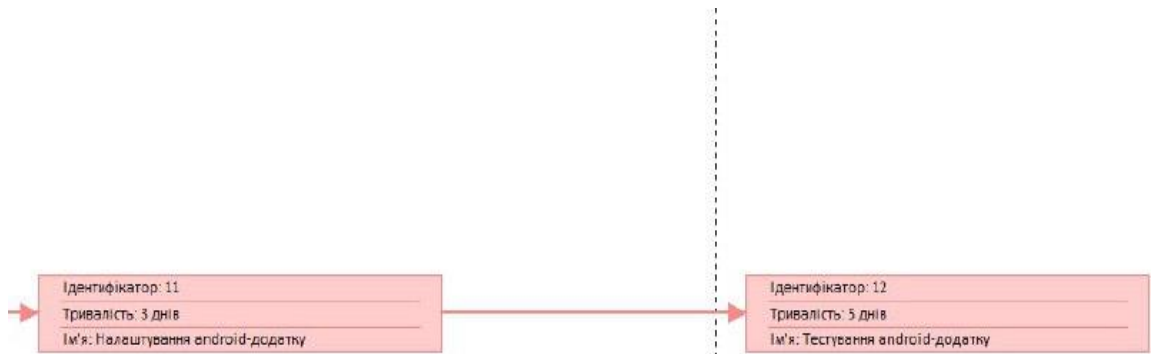


Рис.5.4 Продовження розгорнутого мережевого графіку

Пізній час розробки проекту обчислюється сумою тривалості усіх робіт що лежать на критичному шляху.

$$T_{\Pi} = T_1 + T_2 + T_3 + T_{10} + T_{12} = 34 \text{ (дні)}.$$

5.2 Економічне обґрунтування розробки та впровадження проектного рішення.

Для економічного обґрунтування розробки та впровадження проектного рішення обчислюється ряд економічних показників:

- витрати на розробку та впровадження проектного рішення.

Розрахунок витрат на розробку та впровадження проектного рішення включає в себе наступні складові:

- витрати на оплату праці розробників;
- єдиний соціальний внесок;

- вартість додаткових виробів, що закуповуються;
- транспортно-заготівельні витрати;
- накладні витрати;
- інші витрати;
- витрати на відлагодження і дослідну експлуатацію програмного забезпечення на ЕОМ.

- експлуатаційні витрати;
- ціна споживання проектного рішення;
- показники економічної ефективності.

5.2.1 Розрахунок витрат на розробку та впровадження проектного рішення.

Розрахунок витрат на розробку та впровадження проектного рішення, що визначається за формулою (5.1) та включає в себе наступні складові:

- витрати на оплату праці розробників (B_{on});
- єдиний соціальний внесок ($B_{св}$);
- вартість додаткових виробів, що закуповуються (B_{ϕ});
- транспортно-заготівельні витрати (B_{mp});
- накладні витрати (B_n);
- інші витрати (B_{in}).

Формула має наступний вигляд:

$$K_{заг} = K_1 + K_2 \quad (5.1)$$

де K_1 – витрати на розробку програмного засобу, грн.;

K_2 – витрати на відлагодження і дослідну експлуатацію програмного забезпечення на ЕОМ, грн.

Визначення витрат на оплату праці та відрахувань на соціальні заходи.

Відповідно до Закону України «Про оплату праці» заробітна плата – це «винагорода, обчислена, як правило, у грошовому виразі, яку власник або уповноважений ним орган виплачує працівникові за виконану ним роботу».

					123.KI-41.04	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		83

Розмір заробітної плати залежить від складності та умов виконуваної роботи, професійно-ділових якостей працівника, результатів його праці та господарської діяльності підприємства. Заробітна плата складається з основної та додаткової оплати праці.

Основна заробітна плата нараховується на виконану роботу за тарифними ставками, відрядними розцінками чи посадовими окладами і не залежить від результатів господарської діяльності підприємства.

Додаткова заробітна плата – це складова заробітної плати працівників, до якої включають витрати на оплату праці, не пов’язані з виплатами за фактично відпрацьований час. Нараховують додаткову заробітну плату залежно від досягнутих і запланованих показників, умов виробництва, кваліфікації виконавців. Джерелом додаткової оплати праці є фонд матеріального стимулювання, який створюється за рахунок прибутку.

Для обчислення середньоденної заробітної плати кожного з працівників можна використати наступну формулу:

$$ЗП_{Дi} = \frac{ЗП_i}{\Phi_M}$$

де $ЗП_i$ – основна місячна заробітна плата розробника 1-ої спеціальності, грн.;

Φ_M – місячний фонд робочого часу, днів (24 дні).

Обчислення витрат на оплату праці усіх розробників проекту можна здійснити за формулою:

$$B_{OP} = \sum_{i=1}^N n_i \cdot t_i \cdot ЗП_{Дi}$$

де n_i – чисельність розробників проекту і-ої спеціальності, чол.;

t_i – час, витрачений на розробку проекту працівником і-ої спеціальності, дні;

$ЗП_{Дi}$ – денна заробітна плата розробника і-ої спеціальності, грн.;

Таблиця 5.3 Заробітна плата робітників

					123.KI-41.04	Арк.
						84
Зм.	Арк.	№ докум.	Підпис	Дата		

Працівник	$ЗП_i$
Розробник	$ЗП_{Д1} = \frac{7000}{24} = 291,66$
Дизайнер	$ЗП_{Д2} = \frac{6800}{24} = 283,33$
Тестувальник	$ЗП_{Д3} = \frac{6700}{24} = 279,16$
Розробник БД	$ЗП_{Д4} = \frac{6600}{24} = 275$
Консультант	$ЗП_{Д5} = \frac{6500}{24} = 270,83$

Тоді витрати на оплату праці усіх розробників проекту становлять:

$$В_{ОП} = (1 * 291,66 * 35) + (1 * 283,33 * 10) + (1 * 279,16 * 5) + (1 * 275 * 3) + (1 * 270,83 * 1) = 15533,03 \text{ грн.}$$

Додаткова заробітна плата становить 15% від суми основної заробітної плати.

$$З_{дод.} = З_{осн.} \cdot K_{дод.}$$

де, $K_{дод.}$ – коефіцієнт додаткових виплат працівникам, 0.15.

Розробник:

$$З_{осн} = 279,16 * 35 = 10208,1 \text{ грн.}$$

$$З_{дод.} = 9770,6 * 0,15 = 1531,21 \text{ грн.}$$

Дизайнер:

$$З_{осн} = 254,17 * 10 = 2833,3 \text{ грн.}$$

$$З_{дод.} = 2541,7 * 0,15 = 424,99 \text{ грн.}$$

Тестувальник:

$$З_{осн} = 262,5 * 5 = 1395,8 \text{ грн.}$$

$$З_{дод.} = 1312,5 * 0,15 = 209,37 \text{ грн.}$$

Розробник БД:

$$З_{осн} = 256,25 * 3 = 825 \text{ грн.}$$

$$З_{дод.} = 768,75 * 0,15 = 123,75 \text{ грн.}$$

					<i>123.KI-41.04</i>	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		85

Консультант:

$$З_{осн} = 258,3 * 1 = 270,83 \text{ грн.}$$

$$З_{дод.} = 258,3 * 0,15 = 40,62 \text{ грн.}$$

Звідси загальні витрати на оплату праці ($B_{o.n.}$) визначаються за формулою:

$$B_{o.n.} = Z_{осн.} + Z_{дод.}$$

Розробник : $B_{o.n.1} = 10208,1 + 1531,21 = 11739,31 \text{ грн.}$

Дизайнер: $B_{o.n.2} = 2833,3 + 424,99 = 3258,29 \text{ грн.}$

Тестувальник: $B_{o.n.3} = 1395,8 + 209,37 = 1605,17 \text{ грн.}$

Розробник БД: $B_{o.n.4} = 825 + 123,75 = 948,75 \text{ грн.}$

Консультант: $B_{o.n.5} = 270,83 + 40,62 = 311,45 \text{ грн.}$

Всього: $B_{o.n.} = 11739,31 + 3258,29 + 1605,17 + 948,75 + 311,45 =$
 $17862,97 \text{ грн.}$

Витрати на оплату праці працівникам тягнуть за собою додаткові зобов'язання підприємства по сплаті ЄСВ ($B_{есв}$).

Розробник: $B_{есв} = 11739,31 * 0,22 = 2582,64 \text{ грн.}$

Дизайнер: $B_{есв} = 3258,29 * 0,22 = 716,82 \text{ грн.}$

Тестувальник: $B_{есв} = 1605,17 * 0,22 = 353,13 \text{ грн.}$

Розробник БД: $B_{есв} = 948,75 * 0,22 = 208,72 \text{ грн.}$

Консультант: $B_{есв} = 311,45 * 0,22 = 68,51 \text{ грн.}$

Загальний $B_{есв} = 17862,97 * 0,22 = 3929,64 \text{ грн.}$

Нарахування єдиного внеску складають 22 відсотки до бази нарахування згідно із Законом України «Про внесення змін до Податкового кодексу України та деяких законодавчих актів України щодо забезпечення збалансованості бюджетних надходжень у 2023 році» № 909-VIII від 24 грудня 2023 року.

					123.KI-41.04	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		86

Проведені розрахунки витрат на оплату праці зведені у таблиці 5.1.

Таблиця 5.4 Зведені розрахунки витрат на оплату праці

№ п/п	Категорія працівників	Основна заробітна плата, грн.			Додатков а заробітна плата, грн.	Нарахування на ФОП, грн.	Всього витрати на оплату праці, грн.
		Денна заробітна плата розробника, грн.	К-сть відпрацьов. днів	Фактично нарахов./пл., грн.			
1	Розробник	291,66	35	10208,1	1531,21	2582,64	14321,95
2	Розробник БД	275	3	825	123,75	208,72	1157,47
3	Дизайнер	283,33	10	2833,3	428,99	716,82	3979,11
4	Консультант	270,83	1	270,83	40,62	68,51	379,96
5	Тестувальник	279,16	5	1395,8	209,37	353,13	1958,3
Разом		1399,98	54	15533,03	2333,94	3929,64	21796,79

Визначення вартості додаткових виробів, що закуповуються.

Витрати на додаткові вироби, що закуповуються (B_0) визначаються за їхніми фактичними цінами з урахуванням найменування, номенклатури та необхідної їх кількості в проекті. Вихідні дані та результати розрахунків оформляються у таблиці «Розрахунок витрат на куповані вироби».

Таблиця 5.5 Розрахунок витрат на куповані вироби

Найменування купованих виробів	Марка, тип	Кількість на розробку, шт.	Ціна за одиницю, грн.	Сума витрат, грн.	Сума транспортно-заготівельних витрат, грн.
Маркерна дошка	Вигомax	1	329,15	329,15	36,53
USB-накопичувач	Transcend, 128gb	1	299	299	32,89
Папір офісний	Папір Zoom A4 80 г/м2, 500 арк/пач.	1	330	330	36,3
Всього		3	958,15	958,15	105,72

Визначення транспортно-заготівельних витрат.

Транспортно-заготівельні витрати ($B_{тр}$) для обрахунку даного типу витрат складають 11% від суми придбання купованих виробів.

$$B_{тр1} = 329,15 * 0,11 = 36,53 \text{ грн.}$$

$$B_{тр2} = 299 * 0,11 = 32,89 \text{ грн.}$$

$$B_{тр3} = 330 * 0,11 = 36,3 \text{ грн.}$$

Визначення накладних витрат.

Накладні витрати (B_n) проектних організацій передбачають витрати на управління, загальногосподарські, невиробничі витрати. Вони становлять 25% витрат на оплату праці.

$$B_n = 17862,97 * 0,25 = 4465,74 \text{ грн.}$$

Визначення інших витрат.

Інші витрати ($B_{ін}$) — це витрати, які не враховані в попередніх статтях витрат. Їх розраховують за встановленими відсотками до витрат на оплату праці (6%).

$$B_{ін} = 17862,97 * 0,06 = 1071,77 \text{ грн.}$$

					<i>123.KI-41.04</i>	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		88

Визначення витрат на розробку проектного рішення.

Після вирахування всіх витрат, можна виконати підрахунок витрат на розробку проектного рішення, використовуючи формулу:

$$K_1 = B_{on} + B_{ссв} + B_{o} + B_{н} + B_{н} + B_{ин}.$$

$$K_1 = 17862,97 + 3929,64 + 958,15 + 105,72 + 4465,74 + 1071,77 \\ = 28393,99 \text{ грн.}$$

Визначення витрат на відлагодження і дослідну експлуатацію програмного забезпечення.

Витрати на відлагодження і дослідну експлуатацію програмного забезпечення визначають згідно з формулою:

$$K_2 = S_{м.г.} \cdot t_{від} \quad (5.7)$$

де $S_{м.г.}$ – вартість однієї години роботи ПК, грн./год.

$t_{від}$ – кількість годин роботи ПК на відлагодження програми, год.

Загальна кількість днів роботи на комп'ютері рівна 54. Середній щоденний час роботи на ЕОМ - 2 год., тому:

$$t_{від} = 54 * 3 = 162 \text{ год.}$$

Експлуатаційні витрати, що припадають на 1 год. машинного часу, можуть бути визначені за витратою електроенергії:

$$e_{г} = P_{сн} * C_{вод} \quad (5.8)$$

де $P_{сн} = 360$ Вт - споживана потужність ЕОМ;

$C_{вод} = 1.44$ - вартість 1 кВт/год електроенергії для населення.

Отже, за формулою (5.8)

$$e_{г} = 0.360 * 2.47 = 0.60 \text{ грн/год.}$$

Тоді:

$$K_2 = 0,60 * 162 = 97,2 \text{ грн.}$$

Виходячи з формули розрахунок витрат на розробку та впровадження проектного рішення отримуємо наступні результати:

$$K_{заг} = 28393,99 + 97,2 = 28491,19 \text{ грн.}$$

					123.KI-41.04	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		89

Підготовка кошторису на розробку проектного рішення.

Результати розрахунків для формування кошторису зведені у таблиці 8.5.

Таблиця 5.6 Кошторис витрат на розробку проектного рішення

Найменування елементів витрат	Сума витрат, грн.
Витрати на розробку проектного рішення, у т.ч.:	28393,99
Витрати на оплату праці	17862,97
Єдиний соціальний внесок	3929,64
Витрати на додаткові вироби, що закуповуються	958,15
Транспортно-заготівельні витрати	105,72
Накладні витрати	4465,74
Інші витрати	1071,77
Витрати на відлагодження і дослідну експлуатацію системи	97,2
Всього	28491,19

5.2.2 Визначення експлуатаційних витрат.

Визначення одноразових експлуатаційних витрат проектного рішення та аналогу.

При порівнянні програмних засобів в експлуатаційні витрати включають вартість підготовки даних (E_1) і вартість годин роботи ПК (E_2). Одноразові експлуатаційні витрати визначаються за формулою:

$$E_{П(А)} = E_{1П(А)} + E_{2П(А)} \quad (5.9)$$

Вартість підготовки даних для експлуатації проектного рішення (аналогу) (E_1) визначаються за формулою:

$$E_1 = \sum_{i=1}^N n_i \cdot t_i \cdot 3Пг_i \quad (5.10)$$

Середньогодинна ставка робітника визначається за формулою:

$$3Пг_i = \frac{3Пг_{0i}(1+b)}{\Phi} \quad (5.11)$$

b – коефіцієнт відрахувань на соціальні заходи, становить 0.22;

Φ – фонд робочого часу (год).

					123.KI-41.04	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		90

Отже:

Середньогодинна ставка робітників над проектним рішенням:

Системний адміністратор: $ЗП_{Г1} = \frac{6400 * 1,22}{22 * 8} = 44,36$ грн/год.

Витрати на відлагодження і дослідну експлуатацію аналога становлять:

$P_{cn} = 400$ Вт - споживана потужність ЕОМ;

$C_{вод} = 1,44$ - вартість 1 кВт/год електроенергії для споживачів;

$t_{від} = 1$ год. - кількість годин роботи ПК.

$e_r = 0,400 * 1,44 = 0,57$ грн/год.

Одноразові експлуатаційні витрати на розробку становлять:

$$E_{П} = 44,36 + 0,57 = 44,93 \text{ грн.}$$

Визначення річних експлуатаційних витрат проектного рішення та аналогу.

Експлуатаційні витрати – витрати виробництва, пов'язані з підтриманням у працездатному стані використовуваних систем, машин та устаткування. В інформаційній сфері це поточні витрати об'єкту необхідні для надання обсягу послуг за певний період часу (рік, квартал).

Річні експлуатаційні витрати визначаються за формулою:

$$B_{(e)П(A)} = E_{П(A)} * N_{П(A)} \quad (5.12)$$

де $N_A = 12$ (міс.); $N_{П} = 251$ днів.

$$B_{(e)П} = 44,93 * 251 = 11277,43 \text{ грн.}$$

Вихідні дані та результати розрахунків витрат на підготовку даних для експлуатації проектного рішення зведені у таблицю 5.7.

					<i>123.KI-41.04</i>	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		91

Таблиця 5.7 Розрахунок витрат на підготовку даних для роботи на ЕОМ

Категорія персоналу	Чисельність співробітників і-ої категорії, чол.	Час роботи співробітників і-ої категорії, год.	Середньогодин на ЗП співробітника і-ої категорії, грн.	Витрати на підготовку даних, грн.
Проектне рішення				
Розробник	1	251	44.93	44.93
Всього	1	251	44.93	11277,43

Проектне рішення було розроблено із меншими фінансовими затратами порівняно із досліджуваним аналогом, тому можна зробити висновок, що розроблена модель є більш рентабельною і доступнішою ніж аналог.

5.2.3 Розрахунок ціни споживання проектного рішення.

Ціна споживання ($Ц_C$) – це витрати на придбання і експлуатацію проектного рішення за весь термін його служби:

$$Ц_{C(П)} = Ц_{П} + B_{(E)NPV} \quad (5.13)$$

де $Ц_{П}$ – ціна придбання проектного рішення, грн.;

$B_{(E)NPV}$ – теперішня вартість витрат на експлуатацію проектного рішення (за весь час його експлуатації), грн.

Визначення ціни придбання проектного рішення.

$$Ц_{П} = K \times \left(1 + \frac{P}{100}\right) + K_0 + K_k \quad (5.14)$$

де P – норматив рентабельності 26%;

K – капітальні витрати;

K_0 – витрати на прив'язку та освоєння проектного рішення на конкретному об'єкті, грн.;

K_k – витрати на доукомплектування технічних засобів на об'єкті, грн.;

$$Ц_{П} = 28491,19 * (1 + 0,26) = 35898,66 \text{ грн.}$$

Визначення теперішньої вартості витрат на експлуатацію проектного рішення (за весь час його експлуатації).

Теперішня вартість витрат на експлуатацію проектного рішення розраховується за формулою:

$$B_{(e)NPV} = \sum_{t=1}^T \frac{B_{(E)Nt}}{(1+R)^t} \quad (5.15)$$

де $B_{(E)Nt}$ – річні експлуатаційні витрати в t-ому році, грн.;

При умові, що впродовж всього строку експлуатації $B_{(E)Nt} = const$, тоді:

$$B_{(E)NPV} = \sum_{t=1}^T B_{(T)nt} \cdot \frac{1}{(1+R)^t} = pvB_{(E)n} \quad (5.16)$$

де pv – ставка дисконта на період T (при $T=4$), яка визначається залежно від процентної ставки R (при $R=0,14$) і періоду експлуатації T .

Таблиця 5.8 Теперішня вартість витрат на експлуатацію

T	0	1	2	3	4
PV	1	0,88	0,77	0,68	0,59

$$PV = 0,88 + 0,77 + 0,68 + 0,59 = 2,92.$$

Тоді:

$$Ц_{C(П)} = 35898,66 + 11277,43 * 2,92 = 68828,755 \text{ грн.}$$

$$Ц_{C(A)} = 15000 * 4 = 60\ 000 \text{ грн.}$$

5.2.4 Визначення показників економічної ефективності.

Економічний ефект в сфері експлуатації (грн.):

$$E_{екс} = B(e)a - B(e)n \quad (5.17)$$

$$E_{екс} = 19\ 000 * 4 - 11277,43 * 2,92 = 60000 - 32930,09 = 27069,90 \text{ грн.}$$

Економічний ефект в сфері проектування (грн.):

$$E_{np} = Цa - Цn \quad (5.18)$$

$$E_{np} = 60000 - 32930,09 = 27069,91 \text{ грн.}$$

Отже, $E_{np} > 0$ та $E_{екс} > 0$, тоді розраховується:

Додатковий економічний ефект в сфері експлуатації (грн.):

$$E_{\text{екс Д}} = \sum_{i=1}^T E_{\text{екс}} (1 + R)^{T-i}, \quad (5.19)$$

$$E_{\text{екс Д}} = 27069,90 * (1 + 0,14)^3 + 27069,90 * (1 + 0,14)^2 + 27069,90 * (1 + 0,14)^1 = 40105,24 + 35180,04 + 30859,90 = 106144,97 \text{ грн.}$$

Додатковий економічний ефект в сфері проектування (грн.):

$$E_{\text{пр Д}} = E_{\text{пр}} * (1 + R)^T \quad (5.20)$$

$$E_{\text{пр Д}} = 27069,91 * (1 + 0,14)^4 = 45519,99 \text{ грн.}$$

Термін окупності витрат на проектування рішення (років):

$$T_{\text{ок}} = \frac{K}{E_{\text{екс}}}; \quad (5.21)$$

$$T_{\text{ок}} = \frac{28393,99}{45519,99} \approx 0,6 \text{ року або 7 місяців}$$

Результуючі показники економічної ефективності зводяться у таблицю.

Таблиця 5.9 Показники економічної ефективності проектного рішення

Найменування показників	Одиниці вимірювання	Значення показників	
		Аналог	Проектне рішення
1. Капітальні вкладення	грн.	-	28393,99
2. Ціна придбання	грн.	50 600	35898,66
3. Річні експлуатаційні витрати	грн.	19 000	11277,43
4. Ціна споживання	грн.	76 000	68828,755
5. Економічний ефект в сфері експлуатації	грн.	-	27069,90
6. Додатковий економічний ефект в сфері експлуатації	грн.	-	106144,97
7. Економічний ефект в сфері проектування	грн.	-	45519,99 1
8. Додатковий економічний ефект в сфері проектування	грн.	-	45519,99
9. Термін окупності витрат на проектування рішення	роки	-	0,6

Висновок.

В результаті проведеного економічного аналізу проектного рішення було розраховано витрати на розробку, ціну придбання, експлуатаційні витрати, економічний ефект та термін окупності. Значення були порівняні із значеннями існуючого аналогу, що довело переваги над готовими рішеннями. Даний проект є економічно доцільний, це пояснюється меншими затратами на розробку і експлуатацію без втрати якості та надійності продукту, великим економічним ефектом та коротким терміном окупності.

					123.KI-41.04	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		95

ВИСНОВКИ

В результаті виконання дипломного проекту розроблено програмну інформаційну систему для продажу квитків на різні види транспорту на базі ОС Android. Реалізована система дозволяє виконувати покупку білета на певний вид транспорту за допомогою додатку не виходячи з дому, виконувати бронювання місця та переглядати наявність вільних місць в транспорті.

Під час роботи на дипломним проектом було виконано наступні завдання:

- описано актуальність розробленої системи;
- здійснено огляд та аналіз відомих інформаційних систем для продажу квитків на різні види транспорту;
- проведено порівняльний аналіз відомих систем;
- обгрунтовано вибір програмних засобів для реалізації серверної та клієнтської частини системи;
- реалізовано серверну та клієнтську частини програмної системи;
- розроблено базу даних системи;
- розроблено діаграми класів та алгоритм роботи системи;
- розроблено графічний інтерфейс системи та проведено його тестування.

Реалізована система має клієнт-серверну архітектуру і побудована за допомогою паттерну програмування MVC. Серверна частина реалізована у вигляді веб-сервера TomCat Server Local, сервер прикладної логіки - Spring Web MVC, який надає засоби для доступу до серверної бази даних PostgreSQL. Клієнтська частина реалізована у вигляді прикладної підсистеми для мобільного обчислювального пристрою на платформі ОС Android і забезпечує продаж автобусних та залізничних квитків.

По темі дипломної роботи опубліковано тези на міжнародній науково-практичній конференції «Прикладні науково-технічні дослідження» [26].

В майбутньому можна розширити функціонал додатку та викорисати його для роботи у компаніях, які здійснюють продаж і бронювання квитків на різні види транспорту.

					123.KI-41.04	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		96

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. "Effective Java" (3rd Edition) by Joshua Bloch- програмування Java видавництво Addison-Wesley Profesional 2018. – 261p.
2. Дарвін Я. Android. Збірник рецептів. Завдання та рішення для розробників додатків / Ян Дарвін., 2016. – 432с.
3. "Java: The Complete Reference" (11th Edition) by Herbert Schildt – комплексний посібник з усіма спекти мови, від основ до передових тем. 2022. – 734р.
4. "Java Concurrency in Practice" by Brian Goetz, Tim Peierls, Joshua Bloch, Joseph Bowbeer, David Holmes, and Doug Lea – основні проблеми та рішення Java Addison-Wesley Profesional 2021. – 121p.
5. Booking.com – Режим доступу: <https://www.booking.com/> (дата звернення 24.04.2024).
6. Пасічник В.В., Пасічник О.В., Угрин Д.І. Веб-технології. Підручник – Львів: Магнолія-2006, 2015. – 520 с.
7. busfor.ua – Режим доступу: <https://busfor.ua/> (дата звернення 24.04.2024).
8. Tickets.ua – Режим доступу: <https://tickets.ua/> (дата звернення 24.04.2024).
9. Пасічник О.Г., Пасічник О.В., Стеценко І.В. Основи веб-дизайну. Навчальний посібник – К.: Вид. група ВНУ. - 2009, – 336 с.
10. Приват 24 – Режим доступу: <https://bilet.privatbank.ua/> (дата звернення 24.04.2024).
11. Укрзалізниця– Режим доступу: <https://uz.gov.ua/> (дата звернення 24.04.2024).
12. Кормен, Т. Алгоритми: побудова та аналіз / [Текст] Т. Кормен, Ч. Лейзерсан, Р. Рівест. - М.: МУНМО, 2001. – 960 с.
13. О. М. Ткаченко., В.А. Капчук. Об'єктно-орієнтоване програмування мовою JAVA. Навчальний посібник.- Вінниця: ВНТУ, 2006. -107с.
14. Герберт Шілдіт. Java. Повне керівництво, 8-е вид.: Переклад з англ. - М.: ТОВ "І.Д. Вільямс",2012. – 1104с.

					<i>123.KI-41.04</i>	<i>Арк.</i>
Зм.	Арк.	№ докум.	Підпис	Дата		97

15. Патрік Ноутон, Герберт Шилдт – Java2. – Видавництво: BHV, 2005. – 455с.
16. «PHP and MySQL Web Development (4th Edition)», Luke Welling, Laura Thomson 848 стр., с ил.; ISBN 978-5-8459-1574-0, 978-0-672-32916-6.
17. Лазарева С. Ф. Економіка та організація інформаційного бізнесу : навч. посібн. / С. Ф. Лазарева. - К. : КНЕУ, 2012.- 213 с.
18. Закон України "Про охорону праці" Законодавство України про охорону праці , т.1.- К. - 1995. - 558 с.
19. Державний реєстр міжгалузевих і галузевих нормативних актів про охорону праці (реєстр ДНАОП). – К.: 1998. — 240 с.
20. Android Studio- <https://developer.android.com/>
21. Пакети SDK – Режим доступу: <https://developer.microsoft.com/> (дата звернення 24.04.2024).
22. IntelliJ IDEA – Режим доступу: <https://www.jetbrains.com/> (дата звернення 24.04.2024).
23. IntelliJ IDEA – Режим доступу: <https://www.youtube.com/watch?v=yefmcX57Eyg> (дата звернення 24.04.2024).
24. Android Studio – Режим доступу: https://www.youtube.com/watch?v=zzV2aML_zNg (дата звернення 24.04.2024).
25. SDK – Режим доступу: <https://poandroidam.ua/installation/chtotakoe-sdk-i-kak-ustanovit.html> (дата звернення 24.04.2024).
26. Демчук Л., Грига В. Інформаційна система пошуку та купівлі квитків на різні види транспорту // Матеріали VI міжнародної науково-практичної конференції “Прикладні науково-технічні дослідження” 14-16 травня 2024 року. Івано-Франківськ, Україна. - С. 26-28.

					123.KI-41.04	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		98

ДОДАТКИ

Додаток А.

Цей код налаштовує веб-застосунок на базі Spring.

ApplicationInitializer.class

```
package com.goobico.ticket.server;

import com.goobico.remindme.server.config.WebConfig;
import org.springframework.web.WebApplicationInitializer;
import org.springframework.web.context.ContextLoaderListener;
import
org.springframework.web.context.support.AnnotationConfigWebApplicati
onContext;
import org.springframework.web.servlet.DispatcherServlet;

import javax.servlet.ServletContext;
import javax.servlet.ServletException;
import javax.servlet.ServletRegistration;

public class ApplicationInitializer implements
WebApplicationInitializer {

    private final static String DISPATCHER = "dispatcher";

    public void onStartup(ServletContext servletContext) throws
ServletException {
        AnnotationConfigWebApplicationContext ctx = new
AnnotationConfigWebApplicationContext();
        ctx.register(WebConfig.class);
        servletContext.addListener(new ContextLoaderListener(ctx));

        ServletRegistration.Dynamic servlet =
servletContext.addServlet(DISPATCHER, new DispatcherServlet(ctx));
        servlet.addMapping("/");
    }
}
```

					123.KI-41.04	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		99

```
        servlet.setLoadOnStartup(1);
    }
}
```

TicketService.class

```
public interface TicketService {

    List<Ticked> getAll();
    Ticked getByID(long id);
    Ticked save(Ticked ticked);
    void remove(long id);

}
```

TicketServiceImpl.class

```
@Service
public class TicketServiceImpl implements TicketService {

    @Autowired
    private TicketRepository repository;

    public List<Ticked> getAll() {
        return repository.findAll();
    }

    public Ticked getByID(long id) {
        return repository.findOne(id);
    }

    public Ticked save(Ticked ticked) {
        return repository.saveAndFlush(ticked);
    }

    public void remove(long id) {
        repository.delete(id);
    }
}
```

					<i>123.KI-41.04</i>	<i>Арк.</i>
<i>Зм.</i>	<i>Арк.</i>	<i>№ докум.</i>	<i>Підпис</i>	<i>Дата</i>		100

```

    }
}

Ticket.class
@Entity
@Table(name = "remind")
public class Ticked {

    @Id
    @GeneratedValue(generator = "increment")
    @GenericGenerator(name = "increment", strategy = "increment")
    private long id;

    @Column(name = "title", nullable = false, length = 50)
    private String title;

    @Column(name = "remind_date", nullable = false)
    @Temporal(TemporalType.TIMESTAMP)
    private Date remindDate;

    public Ticked() {
    }

    public long getId() {
        return id;
    }

    public void setId(long id) {
        this.id = id;
    }

    public String getTitle() {
        return title;
    }
}

```

					<i>123.KI-41.04</i>	<i>Арк.</i>
<i>Зм.</i>	<i>Арк.</i>	<i>№ докум.</i>	<i>Підпис</i>	<i>Дата</i>		101

```

public void setTitle(String title) {
    this.title = title;
}

public Date getRemindDate() {
    return remindDate;
}

public void setRemindDate(Date remindDate) {
    this.remindDate = remindDate;
}
}

@RestController
public class TicketsController {

    @Autowired
    private TicketService service;

    @RequestMapping(value = "/tickets", method = RequestMethod.GET)
    @ResponseBody
    public List<Ticked> getAllReminders() {
        return service.getAll();
    }

    @RequestMapping(value = "/tickets/{id}", method =
RequestMethod.GET)
    @ResponseBody
    public Ticked getReminder(@PathVariable("id") long remindID) {
        return service.getByID(remindID);
    }

    @RequestMapping(value = "/tickets", method = RequestMethod.POST)
    @ResponseBody
    public Ticked saveRemider(@RequestBody Ticked ticked) {

```

					<i>123.KI-41.04</i>	<i>Арк.</i>
<i>Зм.</i>	<i>Арк.</i>	<i>№ докум.</i>	<i>Підпис</i>	<i>Дата</i>		102

```

        return service.save(ticked);
    }

    @RequestMapping(value = "/tickets/{id}", method =
RequestMethod.DELETE)
    @ResponseBody
    public void delete(@PathVariable long id) {
        service.remove(id);
    }
}

```

WebController.class

```

@Configuration
@EnableWebMvc
@ComponentScan("com.goobico.remindme.server")
public class WebConfig extends WebMvcConfigurerAdapter {

    @Override
    public void
configureMessageConverters(List<HttpMessageConverter<?>> converters)
{
        MappingJackson2HttpMessageConverter converter = new
MappingJackson2HttpMessageConverter();
        converter.setObjectMapper(new ObjectMapper());

        converter.setSupportedMediaTypes(Collections.singletonList(MediaType
.APPLICATION_JSON));

        converters.add(converter);
    }
}

```

DBController.class

					123.KI-41.04	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		103

```

@Configuration
@EnableJpaRepositories("com.qoobico.remindme.server.repository")
@EnableTransactionManagement
@PropertySource("classpath:db.properties")
@ComponentScan("com.qoobico.remindme.server")
public class DatabaseConfig {

    @Resource
    private Environment env;

    @Bean
    public LocalContainerEntityManagerFactoryBean
entityManagerFactory() {
        LocalContainerEntityManagerFactoryBean em = new
LocalContainerEntityManagerFactoryBean();
        em.setDataSource(dataSource());

em.setPackagesToScan(env.getRequiredProperty("db.entity.package"));
        em.setJpaVendorAdapter(new HibernateJpaVendorAdapter());
        em.setJpaProperties(getHibernateProperties());

        return em;
    }

    @Bean
    public DataSource dataSource() {
        BasicDataSource ds = new BasicDataSource();
        ds.setUrl(env.getRequiredProperty("db.url"));
        ds.setDriverClassName(env.getRequiredProperty("db.driver"));
        ds.setUsername(env.getRequiredProperty("db.username"));
        ds.setPassword(env.getRequiredProperty("db.password"));

ds.setInitialSize(Integer.valueOf(env.getRequiredProperty("db.initia
lSize")));
    }
}

```

					<i>123.KI-41.04</i>	<i>Арк.</i>
<i>Зм.</i>	<i>Арк.</i>	<i>№ докум.</i>	<i>Підпис</i>	<i>Дата</i>		104

```

ds.setMinIdle(Integer.valueOf(env.getRequiredProperty("db.minIdle"))
);

ds.setMaxIdle(Integer.valueOf(env.getRequiredProperty("db.maxIdle"))
);

ds.setTimeBetweenEvictionRunsMillis(Long.valueOf(env.getRequiredProp
erty("db.timeBetweenEvictionRunsMillis")));

ds.setMinEvictableIdleTimeMillis(Long.valueOf(env.getRequiredPropert
y("db.minEvictableIdleTimeMillis")));

ds.setTestOnBorrow(Boolean.valueOf(env.getRequiredProperty("db.testO
nBorrow")));

ds.setValidationQuery(env.getRequiredProperty("db.validationQuery"))
;

    return ds;
}

@Bean
public PlatformTransactionManager transactionManager() {
    JpaTransactionManager manager = new JpaTransactionManager();

manager.setEntityManagerFactory(entityManagerFactory().getObject());

    return manager;
}

public Properties getHibernateProperties() {

    return properties;
} catch (IOException e) {

```

					<i>123.KI-41.04</i>	<i>Арк.</i>
<i>Зм.</i>	<i>Арк.</i>	<i>№ докум.</i>	<i>Підпис</i>	<i>Дата</i>		105


```

        throw new IllegalArgumentException("Can't find
'hibernate.properties' in classpath!", e);
    }
}
}

```

ДОДАТОК Б

Цей код створює Android-додаток з вкладками, навігаційним меню та можливістю завантажувати дані з сервера.

AndroidManifest.xml

```

<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="buyticket.fedyshyn.com.buyticket">

    <application
        android:allowBackup="true"
        android:icon="@mipmap/app"
        android:label="@string/app_name"
        android:supportsRtl="true"
        android:theme="@style/AppTheme">
        <activity android:name=".MainActivity">
            <intent-filter>
                <action android:name="android.intent.action.MAIN" />

                <category
android:name="android.intent.category.LAUNCHER" />
            </intent-filter>
        </activity>
        <activity android:name=".CategoryActivity" />
        <activity android:name=".TrainActivity" />
        <activity android:name=".PlaceTrainActivity" />
        <activity android:name=".CardActivity" />
        <activity android:name=".DoneActivity"></activity>
    </application>

```

					<i>123.KI-41.04</i>	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		106

```
</application>
</manifest>
```

MainActivity.xml

```
package buyticket.fedyshyn.com.buyticket;

import android.content.Intent;
import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;
import android.view.View;
import android.widget.ImageButton;

public class MainActivity extends AppCompatActivity {

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        setTheme(R.style.AppDefault);
        super.onCreate(savedInstanceState);
        setContentView(LAYOUT);

        initToolbar();
        initNavigationView();
        initTabs();
    }

    private void initToolbar() {
        toolbar = (Toolbar) findViewById(R.id.toolbar);
        toolbar.setTitle(R.string.app_name);
        toolbar.setOnMenuItemClickListener(new
Toolbar.OnMenuItemClickListener() {
            @Override
            public boolean onMenuItemClick(MenuItem menuItem) {
                return false;
            }
        });
    }
}
```

					<i>123.KI-41.04</i>	<i>Арк.</i>
<i>Зм.</i>	<i>Арк.</i>	<i>№ докум.</i>	<i>Підпис</i>	<i>Дата</i>		107

```

        toolbar.inflateMenu(R.menu.menu);
    }

    private void initTabs() {
        viewPager = (ViewPager) findViewById(R.id.viewPager);
        adapter = new TabsFragmentAdapter(getApplicationContext(),
getSupportFragmentManager(), new ArrayList<TicketDTO>());
        viewPager.setAdapter(adapter);

        new RemindMeTask().execute();

        TabLayout tabLayout = (TabLayout)
findViewById(R.id.tabLayout);
        tabLayout.setupWithViewPager(viewPager);

    }

    private void initNavigationView() {
        drawerLayout = (DrawerLayout)
findViewById(R.id.drawer_layout);

        ActionBarDrawerToggle toggle = new
ActionBarDrawerToggle(this, drawerLayout, toolbar,
R.string.view_navigation_open, R.string.view_navigation_close);
        drawerLayout.setDrawerListener(toggle);
        toggle.syncState();

        NavigationView navigationView = (NavigationView)
findViewById(R.id.navigation);
        navigationView.setNavigationItemSelectedListener(new
NavigationView.OnNavigationItemSelectedListener() {
            @Override
            public boolean onNavigationItemSelectedListener(MenuItem

```

					123.KI-41.04	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		108

```

menuItem) {
    drawerLayout.closeDrawers();
    switch (menuItem.getItemId()) {
        case R.id.actionNotificationItem:
            showNotificationTab();
    }
    return true;
}

});
}

private void showNotificationTab() {
    viewPager.setCurrentItem(Constants.TAB_TWO);
}

private class RemindMeTask extends AsyncTask<Void, Void,
TicketDTO> {

    @Override
    protected TicketDTO doInBackground(Void... params) {
        RestTemplate template = new RestTemplate();
        template.getMessageConverters().add(new
MappingJackson2HttpMessageConverter());

        return template.getForObject(Constants.URL.GET_REMIND,
TicketDTO.class);
    }

    @Override
    protected void onPostExecute(TicketDTO remindDTO) {
        List<TicketDTO> data = new ArrayList<>();
        data.add(remindDTO);

        adapter.setData(data);
    }
}

```

					<i>123.KI-41.04</i>	<i>Арк.</i>
<i>Зм.</i>	<i>Арк.</i>	<i>№ докум.</i>	<i>Підпис</i>	<i>Дата</i>		109

```

    }
}
main_activity_layout.xml
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout
xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:paddingBottom="@dimen/activity_vertical_margin"
    android:paddingLeft="@dimen/activity_horizontal_margin"
    android:paddingRight="@dimen/activity_horizontal_margin"
    android:paddingTop="@dimen/activity_vertical_margin"

tools:context="buyticket.fedyshyn.com.buyticket.CategoryActivity"
    android:background="@mipmap/fontwo">

    <ImageButton
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:id="@+id/imageButton2"
        android:src="@drawable/q1"
        android:layout_alignParentTop="true"
        android:layout_alignParentLeft="true"
        android:layout_alignParentStart="true"
        android:layout_marginBottom="10dp"
        android:layout_marginLeft="25dp"
        android:layout_marginTop="8dp"
        android:background="#00ffffff" />

    <ImageView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:id="@+id/imageView"
        android:src="@drawable/circ"

```

					<i>123.KI-41.04</i>	<i>Арк.</i>
<i>Зм.</i>	<i>Арк.</i>	<i>№ докум.</i>	<i>Підпис</i>	<i>Дата</i>		110

```
android:layout_alignParentTop="true"
android:layout_alignParentRight="true"
android:layout_alignParentEnd="true"
android:layout_marginRight="25dp"
android:layout_marginTop="8dp" />
```

<ImageView

```
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:id="@+id/imageView2"
android:layout_below="@+id/imageView"
android:layout_alignParentRight="true"
android:layout_alignParentEnd="true"
android:src="@drawable/fntypo"
android:layout_marginRight="25dp"
android:layout_marginTop="10dp" />
```

<ImageButton

```
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:id="@+id/imageButton3"
android:layout_below="@+id/imageButton2"
android:layout_alignParentLeft="true"
android:layout_alignParentStart="true"
android:src="@drawable/train"
android:background="#00ffffff"
android:layout_marginLeft="25dp" />
```

<ImageButton

```
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:id="@+id/imageButton4"
android:src="@drawable/board"
android:background="#00ffffff"
android:layout_below="@+id/imageButton3"
```

					<i>123.KI-41.04</i>	<i>Арк.</i>
<i>Зм.</i>	<i>Арк.</i>	<i>№ докум.</i>	<i>Підпис</i>	<i>Дата</i>		111

```

        android:layout_alignParentLeft="true"
        android:layout_alignParentStart="true"
        android:layout_marginLeft="25dp"
        android:layout_marginTop="10dp" />

<ImageButton
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:id="@+id/imageButton5"
    android:layout_alignBottom="@+id/imageButton4"
    android:layout_alignParentRight="true"
    android:layout_alignParentEnd="true"
    android:src="@drawable/plain"
    android:background="#00ffffff"
    android:layout_marginRight="25dp" />

</RelativeLayout>

```

Contact.class

```

public class Constants {

    public static final int TAB_TWO = 1;

    public static class URL {
        private static final String HOST =
"http://192.168.1.100:8080/";
        public static final String GET_REMIND = HOST +
"reminders/1";
    }

}

```

TicketDTO.class

```

public class TicketDTO {

    private long id;

```

					<i>123.KI-41.04</i>	<i>Арк.</i>
<i>Зм.</i>	<i>Арк.</i>	<i>№ докум.</i>	<i>Підпис</i>	<i>Дата</i>		112

```

private String title;
private Date remindDate;

public TicketDTO(String title) {
    this.title = title;
}

public TicketDTO() {
}

public long getId() {
    return id;
}

public void setId(long id) {
    this.id = id;
}

public Date getRemindDate() {
    return remindDate;
}

public void setRemindDate(Date remindDate) {
    this.remindDate = remindDate;
}

public String getTitle() {
    return title;
}

public void setTitle(String title) {
    this.title = title;
}
}

```

					<i>123.KI-41.04</i>	<i>Арк.</i>
<i>Зм.</i>	<i>Арк.</i>	<i>№ докум.</i>	<i>Підпис</i>	<i>Дата</i>		113

Types.class

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout
xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:card="http://schemas.android.com/apk/res-auto"
    android:orientation="vertical"
android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:layout_marginTop="5dp"
    android:layout_marginBottom="5dp"
    android:layout_marginLeft="12dp"
    android:layout_marginRight="12dp">

    <android.support.v7.widget.CardView
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:id="@+id/cardView"
        card:cardElevation="2dp"
        card:cardUseCompatPadding="true">

        <LinearLayout
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:padding="16dp"
            android:orientation="vertical">

            <TextView
                android:layout_width="wrap_content"
                android:layout_height="wrap_content"
                android:id="@+id/title"
                android:textSize="8pt"
                android:textColor="@color/colorBlack"
                android:paddingBottom="5dp"/>

            <LinearLayout
```

					123.KI-41.04	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		114

```

        android:layout_width="match_parent"
        android:layout_height="wrap_content">
        <TextView
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:text="TEST"/>
    </LinearLayout>
</LinearLayout>
</android.support.v7.widget.CardView>

</LinearLayout>

```

TicketListAdapter.class

```

public class TicketListAdapter extends
RecyclerView.Adapter<TicketListAdapter.RemindViewHolder> {

    private List<TicketDTO> data;

    public TicketListAdapter(List<TicketDTO> data) {
        this.data = data;
    }

    @Override
    public RemindViewHolder onCreateViewHolder(ViewGroup parent, int
viewType) {
        View view =
LayoutInflater.from(parent.getContext()).inflate(R.layout.remind_ite
m, parent, false);

        return new RemindViewHolder(view);
    }

    @Override
    public void onBindViewHolder(RemindViewHolder holder, int
position) {
        TicketDTO item = data.get(position);

```

					<i>123.KI-41.04</i>	<i>Арк.</i>
<i>Зм.</i>	<i>Арк.</i>	<i>№ докум.</i>	<i>Підпис</i>	<i>Дата</i>		115

```

        holder.title.setText(item.getTitle());
    }

    @Override
    public int getItemCount() {
        return data.size();
    }

    public void setData(List<TicketDTO> data) {
        this.data = data;
    }

    public static class RemindViewHolder extends
RecyclerView.ViewHolder {
        CardView cardView;
        TextView title;

        public RemindViewHolder(View itemView) {
            super(itemView);
        }
    }
}

```

TabsFragmentAdapter.class

```

public class TabsFragmentAdapter extends FragmentPagerAdapter {

    private Map<Integer, AbstractTabFragment> tabs;
    private Context context;

    private List<TicketDTO> data;

    private HistoryFragment historyFragment;

    public TabsFragmentAdapter(Context context, FragmentManager fm,
List<TicketDTO> data) {
        super(fm);
        this.data = data;
    }
}

```

					123.KI-41.04	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		116

```

        this.context = context;
        initTabsMap(context);
    }

    @Override
    public CharSequence getPageTitle(int position) {
        return tabs.get(position).getTitle();
    }

    @Override
    public Fragment getItem(int position) {
        return tabs.get(position);
    }

    @Override
    public int getCount() {
        return tabs.size();
    }

    private void initTabsMap(Context context) {
        tabs = new HashMap<>();
        historyFragment = HistoryFragment.getInstance(context,
data);
        tabs.put(0, historyFragment);
    }

    public void setData(List<TicketDTO> data) {
        this.data = data;
        historyFragment.refreshList(data);
    }
}

```

colors.xml

					<i>123.KI-41.04</i>	<i>Арк.</i>
<i>Зм.</i>	<i>Арк.</i>	<i>№ докум.</i>	<i>Підпис</i>	<i>Дата</i>		117

```
<?xml version="1.0" encoding="utf-8"?>
<resources>
  <color name="colorPrimary">#0097A7</color>
  <color name="colorPrimaryDark">#00838F</color>
  <color name="mainBackground">#e2e2e2</color>
  <color name="colorWhite">#e2e2e2</color>
  <color name="colorBlack">#191919</color>
</resource>
```

					123.KI-41.04	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		118