

**ДЕРЖАВНИЙ УНІВЕРСИТЕТ ТЕЛЕКОМУНІКАЦІЙ
НАВЧАЛЬНО-НАУКОВИЙ ІНСТИТУТ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ
КАФЕДРА ІНЖЕНЕРІЇ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ**

Пояснювальна записка

до бакалаврської роботи
на ступінь вищої освіти «Бакалавр»

**на тему: «РОЗРОБКА ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ ДЛЯ
АВТОМАТИЗОВАНОЇ СИСТЕМИ УПРАВЛІННЯ ЛОГІСТИЧНИХ
ЗАМОВЛЕНЬ МОВОЮ JAVA»**

Виконав студент 5 курсу, групи ППЗ-52
спеціальності

121

_____ Інженерія програмного забезпечення _____

(шифр і назва спеціальності)

Зозуля В.В.

_____ (прізвище та ініціали) _____

Керівник _____

Залива В.В.

_____ (прізвище та ініціали) _____

Рецензент _____

_____ (прізвище та ініціали) _____

ДЕРЖАВНИЙ УНІВЕРСИТЕТ ТЕЛЕКОМУНІКАЦІЙ**НАВЧАЛЬНО-НАУКОВИЙ ІНСТИТУТ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ**

Кафедра Інженерії програмного забезпечення

Ступінь вищої освіти «Бакалавр»

Спеціальність підготовки – 121 «Інженерія програмного забезпечення»

ЗАТВЕРДЖУЮ

Завідувач кафедри

Інженерії програмного забезпечення

Негоденко О.В.

“ ____ ” _____ 2021 року

**З А В Д А Н Н Я
НА БАКАЛАВРСЬКУ РОБОТУ СТУДЕНТУ**

ЗОЗУЛІ ВЛАДИСЛАВУ ВАСИЛЬОВИЧУ

(прізвище, ім'я, по батькові)

1. Тема бакалаврської роботи: «Розробка програмного забезпечення для автоматизованої системи управління логістичних замовлень мовою JAVA»

Керівник бакалаврської роботи Залива В.В., аспірант, асистент кафедри ПІЗ
(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

Затверджені наказом вищого навчального закладу від «12» березня 2021 року №65.

2. Строк подання студентом бакалаврської роботи «01» червня 2021 року

3. Вихідні дані до бакалаврської роботи: Вихідні дані по формуванню логістичних замовлень компанії. Загальна структура замовлень. Середовище програмування Android Studio 4.1.3. Технічна документація Google. Інформаційний портал Javarush. XML documentation. Науково-технічна література з питань програмування мобільних пристроїв на операційній системі Android об'єктно-орієнтованою мовою програмування JAVA.

4. Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити)

4.1 Актуальність розробки програмного забезпечення для автоматизованої системи управління логістичних замовлень.

4.2	Проектування автоматизованої системи розподілу замовлень.
4.3	Програмна реалізація автоматизованої системи управління логістичних замовлень мовою JAVA.
5.	Перелік графічного матеріалу
5.1.	Тема бакалаврської роботи.
5.2.	Актуальність теми, мета, об'єкт та предмет, основні завдання дослідження.
5.3.	Розгляд логістичних проблем та можливість їх вирішення.
5.4.	Дослідження основних мобільних операційних систем.
5.5.	Аналіз та порівняння основних мов програмування.
5.6.	Порівняльний аналіз основних принципів, та підходів до розробки програмного забезпечення.
5.7.	Рекомендації щодо розробки дизайну мобільного додатку.
5.8.	Функціональні можливості розробленої програми.
5.9.	Висновки за результатами роботи.

6. Дата видачі завдання «19» квітня 2021

КАЛЕНДАРНИЙ ПЛАН

№ зп	Назва етапів бакалаврської роботи	Строк виконання етапів бакалаврської роботи	Примітка
1.	Постановка цілей та задач, план роботи.	19.04-22.04	Виконано
2.	Збір інформації.	23.04-28.04	Виконано
3.	Аналіз актуальності та затребуваності продукта даної області розробки.	29.04-03.05	Виконано
4.	Розробка інтерфейсу користувача та функціональної частини мобільного додатка.	05.05-10.05	Виконано
5.	Вступ, висновки, реферат. Оформлення дипломної роботи.	15.05-19.05	Виконано
6.	Тестування розробленого програмного продукту.	20.05-25.05	Виконано
7.	Підготовка доповіді до захисту.	01.06.2021	

Студент _____

(підпис)

Зозуля В.В.

прізвище та ініціали

Керівник бакалаврської роботи _____

(підпис)

Залива В.В.

прізвище та ініціали

РЕФЕРАТ

Текстова частина бакалаврської роботи: 58 сторінок, 10 рисунків, 1 таблиця, 19 джерел.

Об'єкт дослідження – розробка програмного забезпечення по автоматизації логістичних замовлень.

Предмет дослідження – технологічні аспекти розробки програмного забезпечення для операційної системи Андроїд мовою програмування JAVA.

Мета роботи – розробити мобільний додаток котрий буде мати основний функціонал по створенню логістичних замовлень, тим самим підвищуючи ефективності та надійність їх функціонування у відповідності до встановлених стандартів підприємства.

Методи дослідження – в роботі були використані загальноприйняті методи при розробці програмного забезпечення та мобільних додатків, методи оптимального управління, обробка та аналіз отриманих результатів, що дало змогу проаналізувати актуальність роботи, її цілісність та функціональні можливості.

В роботі проведено аналіз затребуваності програмного забезпечення по управлінню логістичними замовленнями. Здійснено аналіз потреб, які диктує ринок, та на основі виконаних результатів розроблено програму на операційній системі Андроїд, що дозволяє користувачам застосовувати її в своїх цілях.

Галузь використання – інформаційні технології.

ІНФОРМАЦІЙНІ ТЕХНОЛОГІЇ, РОЗРОБКА ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ, РОЗРОБКА МОБІЛЬНОГО ДОДАТКУ ПІД ОПЕРАЦІЙНУ СИСТЕМУ АНДРОЇД, ОБРОБКА ЛОГІСТИЧНИХ ЗАПИТІВ, СТВОРЕННЯ БЛАНКІВ ДЛЯ ЛОГІСТИЧНИХ ЗАМОВЛЕНЬ.

ЗМІСТ

	Стор.
ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ	10
ВСТУП.....	11
1 АКТУАЛЬНІСТЬ РОЗРОБКИ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ ДЛЯ АВТОМАТИЗОВАНОЇ СТСТЕМИ УПРАВЛІННЯ ЛОГІСТИЧНИХ ЗАМОВЛЕНЬ	13
1.1. Вступ	13
1.2. Актуальність розробки системи управління логістичними замовленнями..	16
1.3. Аналіз основних логістичних проблем та задач	20
1.4 Висновки до розділу 1	23
2 ПРОЕКТУВАННЯ АВТОМАТИЗОВАНОЇ СИСТЕМИ РОЗПОДІЛУ ЗАМОВЛЕННЬ	24
2.1. Аналіз ринку мобільних операційних систем	24
2.2. Аналіз основних характеристик сучасних систем управління логістичними замовленнями	29
2.3. Дослідження основних задач та труднощів при проектуванні під операційну систему Андроїд.....	37
2.4. Алгоритм вибору мови програмування.....	41
2.5 Висновки до розділу 2	44
3 ПРОГРАМНА РЕАЛІЗАЦІЯ АВТОМАТИЗОВАНОЇ СИСТЕМИ РОЗПОДІЛУ ЛОГІСТИЧНИХ ЗАМОВЛЕНЬ МОВОЮ JAVA.....	46
3.1. Визначення основних цілей та задач	46
3.2. Розробка дизайну мобільного додатку	51
3.3. Аналіз потреб щодо функціональних можливостей програмного забезпечення та подальша програмна реалізація	58
3.4 Висновки до розділу 3	64
ВИСНОВКИ	65

ПЕРЕЛІК ПОСИЛАНЬ.....	66
ДЕМОНСТРАЦІЙНІ МАТЕРІАЛИ	68

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ

EOM	– електронна обчислювальна машина
БД	– база даних
СУБД	– система управління базами даних
XML	– <i>eXtensible Markup Language</i>
ОС	– операційна система
ANSI	– American National Standards Institute
ВЛС	– внутрішньовиробнича логістична система
АРМ	– автоматизоване робоче місце
ЛС	– логістична система
ПК	– персональний комп'ютер
ЛОМ	– локально-обчислювальна машина
ПК	– Public Key Infrastructure
АСОІ	– автоматизована система обробки інформації
VPN	– Virtual Private Network
ДП	– дипломний проект
ПП	– програмний продукт
ТЗ	– технічне завдання
HTML	– <i>HyperText Markup Language</i>
GPS	– Global Positioning System
ISO	– International Organization for Standardization
ROM	– <i>Read Only Memory</i>
RAM	– Random Access Memory
LTE	– Long-Term Evolution
GSM	– Groupe Spécial Mobile
KPI	– Key Performance Indicators
ID	– Identifier

ВСТУП

Актуальність дослідження. Більшість підприємств в наш час починають використовувати інформаційні технології упродовж виробничих процесів. В бізнесі все частіше зустрічається впровадження електронного документообігу та обробка даних з використанням ЕОМ. Набуває все більшої актуальності використання прикладних програм для багатьох видів підприємницької діяльності, щоб пришвидшити робочі процеси, мати змогу більш детально контролювати різні аспекти діяльності. На даний момент в логістичній галузі використовуються додатки, побудовані на веб інфраструктурі, що найчастіше використовуються великими підприємствами та міжнародними організаціями, що мають велику розгалужену структуру. Також в пріоритеті програми, які розроблені під ПК на тій чи іншій ОС, що також містять в собі досить велику кількість функціоналу, яка далеко не завжди потрібна початківцям, котрі тільки починають свою діяльність в бізнесі, продажі товарів та послуг, та логістиці.

Уникнути цієї проблеми можна за допомогою використання ПЗ, що розроблене спеціально для мобільних приладів, а саме те, що спроектовано під ОС Андроїд, оскільки смартфони наразі мають дуже велику популярність, абсолютна більшість людей вже перейшла на цей вид телефонів. Це дає змогу використовувати їх у всіх аспектах життя, адже вони є багатофункціональними приладами і можуть служити не тільки в побутових потребах, але і в спеціальних, таких як бухгалтерський облік, навчальні матеріали, графічний дизайн, моделювання та конструювання, відеомонтаж, графічне ПЗ та вирішення логістичних питань.

Актуальність даної теми дипломної роботи спричинена тим, що велика кількість людей сьогодні має можливість та хоче вирішувати логістичні задачі, використовуючи власний смартфон, що й надихнуло ідею розробки ПЗ автоматизованого управління логістичними замовленнями. Основний зміст

роботи становлять аналіз та дослідження вищезгаданих логістичних задач, а також вироблення практичних рекомендацій на основі отриманих результатів.

Об'єкт дослідження – розробка програмного забезпечення по автоматизації логістичних замовлень.

Предмет дослідження – технологічні та загальні аспекти розробки програмного забезпечення для операційної системи Андроїд мовою програмування JAVA. *Мета роботи* – розробити мобільний додаток, котрий 4буде мати основний функціонал по створенню логістичних замовлень, тим самим підвищуючи ефективність та надійність їх функціонування у відповідності до встановлених стандартів підприємства.

Наукові завдання:

дослідити затребуваність та актуальність даного ПЗ;

проаналізувати, які особливості та функціональні можливості має містити дане ПЗ ;

розробити мобільний додаток, який буде мати достатній функціонал для ефективної роботи підприємства по логістичним замовленням.

Методи дослідження – в роботі були використані загальноприйняті методи при розробці програмного забезпечення та мобільних додатків, методи оптимального управління, обробка та аналіз отриманих результатів, що дало змогу проаналізувати актуальність роботи, її цілісність та функціональні можливості.

Практичне значення одержаних результатів полягає у розробці мобільного застосунку, який буде мати достатній функціонал для обробки логістичних замовлень організації, а також у створенні практичних рекомендацій щодо ефективної реалізації його на підприємствах у відповідності до встановлених підприємствами вимог.

1 АКТУАЛЬНІСТЬ РОЗРОБКИ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ ДЛЯ АВТОМАТИЗОВАНОЇ СТТЕМИ УПРАВЛІННЯ ЛОГІСТИЧНИХ ЗАМОВЛЕНЬ

1.1. Вступ

На сьогоднішній день однією із найбільш затребуваних галузей є розвиток інформаційних технологій, що в свою чергу приносить новизну, прогрес до великої кількості різноманітних інших напрямків діяльності людини. Автоматизація є частиною робочих процесів які полегшують управління складними бізнес-процесами. Коли спрацьовує заповнення форми, дія користувача або внутрішній сигнал, автоматизовані робочі процеси можуть переміщувати або перетворювати дані відповідно до ваших інструкцій. Цей практичний підхід допомагає впорядкувати повторювану та трудомістку роботу. Правильно підібраний програмний продукт для автоматизації робочого процесу може допомогти команді пришвидшити прогрес у виконанні завдань, або й навіть включати вбудовану аналітику, яка допоможе визначити неефективність. Більшість інструментів автоматизації робочих процесів надають візуальний аспект, за допомогою якого можливо створювати робочі процеси без великих технічних знань.

Автоматизація робочих процесів стосується проектування, виконання та автоматизації процесів, заснованих на правилах робочого циклу, де людські завдання, дані або файли передаються між людьми або системами на основі заздалегідь визначених бізнес-правил.

Чому люди використовують автоматизацію робочих процесів? Завдяки автоматизації робочих процесів, особливо тих процесів, які переважно обробляються працівниками вручну, можна внести суттєві вдосконалення в:

- ефективність
- продуктивність
- точність

- підзвітність
- задоволення від роботи

Побудувати автоматизований робочий процес можна використовуючи інтерфейс перетягування для додавання та упорядкування завдань, можна планувати та візуально проектувати автоматизовані робочі процеси, щоб імітувати або вдосконалювати існуючі. У більшості випадків процес ілюструється у статичному форматі та перевіряється усіма зацікавленими сторонами на предмет точності. Після того, як статична конструкція доопрацьована, починається робота над розробкою фактичного процесу, включаючи форми, завдання, одержувачів, попередження / сповіщення тощо. Це робиться за допомогою програмного забезпечення для автоматизації робочих процесів, що включає заздалегідь побудовані завдання, які можна організувати та підключити за потреби.

Автоматизація робочих процесів - це проектування та виконання автоматизованих процесів для виконання завдань або маршрутизації даних та файлів між людьми та системами на основі бізнес-правил. Це особливо корисно для зменшення часу, витраченого на певні дії, що виконуються вручну, такі як запит інформації або автоматичне заповнення форм. Використання програмного забезпечення та інструментів автоматизації робочих процесів компанії, які прагнуть автоматизувати процеси, часто звертаються до ІТ-інструментів, до яких входить програмне забезпечення для управління робочими процесами. Правильні інструменти можуть скеровувати інформацію та виконувати дії без затримки. Зрештою, вони дозволяють знизити витрати, підвищити продуктивність та зменшити людські помилки.

Критерії для прийняття рішення про те, коли потрібно використовувати автоматизацію робочого процесу, включають наступне:

- Завдання повторюється.
- Завдання потрібно виконувати точно, без жодних шансів на людські помилки.

- Ряд простих завдань можна зробити більш ефективними, якщо їх автоматизувати.

Автоматизація робочих процесів повинна полегшити організацію роботи на підприємстві задля підвищення його ефективності. Наприклад, деякі автоматизовані завдання робочого циклу можуть бути управлінням електронними таблицями або електронною поштою. Важливість автоматизації робочого процесу полягає в тому, що робочі процеси повинні бути автоматизовані, коли це можливо, з багатьох причин, включаючи більш швидкі операції та підвищення ефективності та точності. Інші вдосконалення включають наступні критерії.

- Це походить від підвищення ефективності виконання завдань, що дозволяє працівникам працювати над іншими неавтоматизованими завданнями.
- Економія витрат. Економія зумовлена підвищенням продуктивності праці.
- Видимість. Якщо впроваджено відображення робочих процесів, то процеси автоматизації повинні бути більш прозорими, даючи організації перегляд її робочих процесів зверху вниз.
- Поліпшення комунікації. Якщо видимість збільшується, то спілкування для працівників може бути більш точним.
- Краще обслуговування клієнтів. Це можна забезпечити, наприклад, автоматизуючи відповіді на скарги клієнтів.
- Потенціал для збільшення залучення клієнтів. Клієнти можуть швидше реагувати за допомогою засобів автоматизації.
- Позбавлення від надмірностей. Надмірність робочого процесу - будь-яке непотрібне завдання, яке можна вирішити легшим способом.
- Покращений загальний кінцевий продукт. Помилка людини виведена з рівняння.
- Цифровий робочий процес можна відстежувати. Це дозволяє організації переглянути, наскільки добре функціонує її бізнес.

Переваги автоматизації робочого процесу включають наступне:

- зменшення циклів робочого процесу;
- менша потреба в ручній праці;
- менша потреба в ручному поводженні з продуктами;
- більша видимість, що призводить до легшого часу виявлення оперативних вузьких місць;
- поліпшення задоволеності споживачів, коли акцент робиться на клієнтах;
- загальне задоволення співробітників, що виключає необхідність у потенційно нудних, повторюваних завданнях;
- покращення рівня задоволеності співробітників шляхом надання інструментів аналізу робочого процесу, які можуть включати інформаційні панелі та ключові показники ефективності (KPI);
- кращі внутрішні та зовнішні комунікації;
- більша відповідальність в організації, що є результатом кожного кроку в робочому процесі, який чітко призначається одній дії;
- надання часу працівникам для виконання інших завдань;
- збільшення виробництва;
- економія витрат;
- масштабованість, оскільки автоматизацію робочого процесу можна змінювати та додавати за необхідності;
- більш ефективне управління завданнями, включаючи інформаційні панелі, календарі та інші інструменти, які можна зробити доступними за допомогою програмних засобів автоматизації робочих процесів.

1.2. Актуальність розробки системи управління логістичними замовленнями

Логістика – галузь або функція в корпорації, завданням якої є забезпечення переміщення та зберігання продукції та сировини для покращення виробництва та продажу.

Логістика – являє собою детальну організація та здійснення складної діяльності, як правило, у загально прийнятому розумінні логістика – це менеджмент потоком речей між пунктом споживання та пунктом походження з метою виконання та задоволення вимог клієнтів або організацій. Логістика керується такими ресурсами, які можуть включати матеріальні товари, обладнання та матеріали, а також витратні матеріали та продукти харчування.

Логістика у військовій науці вже практикувалась у стародавньому світі, і так як сучасна армія має значну потребу в логістичних рішеннях, розроблені вдосконалені варіанти впровадження. У військовій логістиці логістичні працівники визначають, як і коли переміщати ресурси і з'ясовувати де вони потрібні.

Управління логістикою – це проведення певних дій ланцюгами поставок та частина яка вконує інжиніринг ланцюгів поставок, яка планує, застосовує та контролює ефективний, зворотний та прямий потік зберігання послуг, товарів, та інших матеріальних благ, відповідної інформації між пунктом походження та пунктом споживання для задоволення кінцевого споживача. Складність логістики можна оптимізувати, моделювати, візуалізувати, аналізувати та за допомогою спеціального програмного забезпечення для моделювання. Загальною мотивацією у всіх сферах є мінімізація використання ресурсів логістики. В галузі управління логістикою, працює професіонал якого називають логістом.

Логістика – це процес руху товарів по ланцюгу поставок компанії. Однак цей процес складається з різних функцій, якими необхідно належним чином керувати, щоб забезпечити ефективність та дієвість ланцюга поставок організації.

Обробка замовлень – це важливе завдання у функціях логістичних операцій. Замовлення на купівлю, розміщене покупцем постачальнику, є важливим юридичним документом операцій між двома сторонами. Цей документ включає опис або технічні деталі товару щодо постачання, ціни, терміну доставки, умов оплати, податків та інших комерційних умов, як це було узгоджено. Обробка цього документа є важливою, оскільки вона має прямий зв'язок із замовленням або часом циклу виконання, який вказує час отримання замовлення та отримання

матеріалів замовником. Діяльність з обробки замовлень складається з наступних етапів:

1. Замовлення перевірки на будь-які відхилення в узгоджених або обговорених умовах.
2. Ціни, оплата та умови доставки.
3. Перевірка наявності матеріалів на складі.
4. Виробництво та планування матеріалів для нестачі.
5. Визнання наказу із зазначенням відхилень, якщо такі є.
6. Контроль запасів.

Управління запасами полягає у веденні достатньої кількості запасів для задоволення вимог замовника, і в той же час його балансова вартість повинна бути найнижчою. Це, в основному, знаходження балансу між службою споживачів, щоб не втратити ринкову можливість та витрати на її задоволення. Товарні запаси є найбільшим винуватцем у загальному ланцюжку поставок фірми через її величезну балансову вартість, яка побічно з'їдає прибуток. Він складається з витрат на фінансування запасів, страхування, зберігання, збитків. Середня вартість товарно-матеріальних цін варіюється від 10 до 25 відсотків загальної кількості запасів на рік залежно від продукції.

Складування - це зберігання готової продукції до її продажу. Воно відіграє життєво важливу роль у логістичних операціях фірми. Ефективність маркетингу організації залежить від відповідного рішення щодо складування. У сучасному контексті складське господарство трактується як комутаційна установка, а не як сховище неналежного управління складом. Складування є ключовою сферою прийняття рішень у логістиці.

Основними рішеннями щодо складування є:

- Розташування складських приміщень;
- Кількість складів;
- Розмір складу;
- Склад макета;
- Дизайн будівлі;

- Право власності на склад.

Транспортування. Для переміщення товарів від постачальника до покупця перевезення є найважливішою та найважливішою складовою логістики. Коли розміщується замовлення, транзакція не виконується доти, доки товар фізично не буде переміщений до місця клієнта. Фізичний рух товарів здійснюється за допомогою різних видів транспорту. У логістичних витратах його частка коливається від 65 до 70 відсотків у випадку масової споживання продукції з дуже низькою ціною. Фірми обирають спосіб перевезення залежно від інфраструктури перевезень у країні чи регіоні. Вартість є найважливішим фактором при виборі певного виду транспорту. Однак іноді терміновість товару для кінцевого споживача перевершує розгляд витрат.

Система обробки та зберігання матеріалів включає в себе швидкість руху товарних запасів по ланцюгу постачання залежить від методів обробки матеріалів. Неправильний спосіб поводження з матеріалом додасть продукту збитків та затримок у поставках та випадкових накладних витрат.

Механізація та автоматизація обробки матеріалів підвищують продуктивність логістичної системи. Іншими міркуваннями щодо вибору системи поводження з матеріалами є обсяги, які потрібно обробити, швидкість, необхідна для переміщення матеріалу, і рівень обслуговування, що пропонується замовнику. Система зберігання є важливою для максимального використання простору (підлогового та кубічного) на заданих розмірах складу. Система поводження з матеріалами повинна підтримувати систему зберігання для швидкого переміщення (зберігання та вилучення) товарів на склад та з нього.

Логістична або промислова упаковка є критично важливим елементом фізичного розподілу товару, який впливає на ефективність логістичної системи. Вона відрізняється від упаковки товару, яка базується на маркетингових цілях. Хоча матеріально-технічне пакування відіграє важливу роль у захисті від пошкоджень, у випадку з поводженням з матеріалами та економією місця зберігання використання вантажу має велике значення для логістичної упаковки з урахуванням вартості упаковки.

Завдання логістики, як науки, для бізнесу - підібрати кращий маршрут для переміщення товару, виконавши доставку ічасно і мінімізувавши витрати. Затрати на перевезення, на персонал та інші логістичні затарти, можуть досягати 20 відсотків від загальної суми. Потрібно брати до уваги, що для малого бізнесу логістичні витрати часто виявляються куди більш затратними і доводиться вдаватися до специфічних методів пошуку більш дешевих транспортних ланцюжків, при яких йде інтеграція декількох учасників ринку.

1.3. Аналіз основних логістичних проблем та задач

Не зважаючи на досить довге існування логістики як широкого поняття, попри всі існуючі рішення та моделі які створені, більшість із них націлені на великі компанії та підприємства, які мають значні фінансові можливості та змогу використовувати цілі відділи, що займаються питаннями цього напрямку, мають можливість підтримувати роботу цілих офісів де працює суттєва кількість персоналу на ПК. При цьому залишається актуальною підприємницька діяльність малого бізнесу, де немає таких фінансів, кількість працівників варіюється від одного до невеликої групи людей. На першому етапі діяльності вони не мають змогу орендувати офіси, придбати велику кількість офісної техніки, тому в такому разі стає актуальним ведення бізнесу в смартфоні.

Можливо відмовитись від бізнес-ноутбука на користь смартфона. Смартфони стали досить розвиненими за останнє десятиліття. За даними досліджень, використання мобільних цифрових застосунків за останні два роки зросло приблизно на 50%, причому понад 75% цього зростання «безпосередньо пов'язано» з мобільним додатком. «Мобільні пристрої зросли настільки швидко, що зараз це провідна цифрова платформа, загальна активність якої на смартфонах і планшетах становить 62% витраченого часу на цифрові медіа, а лише додатки зараз представляють більшість часу цифрових медіа на рівні 54%». Технології торкнулися і мобільних додатків, так що підприємець або власник невеликої компанії керує 100% свого бізнесу на додатках для смартфонів. Деякі платформи

надійно «сидять» над існуючими корпоративними системами та визначають важливі ділові події та зміни, щоб співробітники могли легко працювати зі своїх мобільних пристроїв. Насправді, проводити ефективне управління можна малим, середнім бізнесом із мобільними додатками за допомогою програмного забезпечення. Будь-де, починаючи від складання договорів, обробки платежів і закінчуючи програмами електронного підпису, мобільні додатки є головною опорою для масштабування малого та середнього бізнесу.

Малий бізнес може бути більш спритним, оскільки він, швидше за все, немає такої кількості складних застарілих технологічних систем, як ключові частини інфраструктури, порівняно з великими компаніями, що дозволяє їм мати простіший і коротший шлях до прийняття підходу лише для мобільних пристроїв. Зрозуміло, що замість того, щоб постійно потребувати поглибленого доступу до різних систем, програмного забезпечення та додатків, важливою є наявність легко доступних фрагментів інформації. По суті, однохвилинний досвід роботи з додатком на мобільному пристрої може бути набагато продуктивнішим, ніж використання кількох систем на робочому столі. Тепер можливо керувати організацією зі свого мобільного пристрою. Управління бізнесом з телефону дозволяє бути більш продуктивним.

Але є і ризики. Недоліком є те, що додатки не зовсім чудово справляються з бізнес-орієнтованими завданнями, такими як зберігання величезних обсягів даних про компанію або проведення зустрічей на рівні компанії. Крім того, існує ризик кібер-крадіжок, коли власники компанії покладаються виключно на програми для смартфонів. Проте переваги значно перевищують недоліки. З огляду на це, бізнес може зазнати підвищеного ризику крадіжки даних через втрату або викрадені смартфони. На щастя, цим ризиком можна ефективно управляти. Ці занепокоєння можуть зникнути, коли власники бізнесу отримують вигоди, отримані завдяки мобільному керуванню своїми компаніями. По-перше, швидка обробка потоків замовлення позбавляє підприємства та їх працівників від повторюваної та втомливої канцелярської роботи. Також є безболісні юридичні контракти, управління заробітною платою та виплатами, проста обробка платежів та

виставлення рахунків - все це сьогодні може бути в мобільному пакеті. Ці програми працюють добре.

Електронна комерція – термін, що використовується для опису широкого спектру інструментів та методів, що використовуються для ведення бізнесу в безпаперовому середовищі. Електронна комерція включає електронний обмін даними, електронну пошту, електронні перекази коштів, електронну публікацію, обробку зображень, електронні дошки оголошень, спільні бази даних та магнітний / оптичний збір даних. Компанії здатні автоматизувати процес переміщення документів в електронному вигляді між постачальниками та замовниками.

Електронний обмін даними (EDI) стосується обміну між комп'ютером ділових документів у стандартному форматі. EDI описує як можливості, так і практику передачі інформації між двома організаціями в електронному вигляді замість традиційної форми пошти, кур'єра та факсу.

Переваги EDI:

1. Швидкий процес отримання інформації.
2. Краще обслуговування клієнтів.
3. Скорочення паперової роботи.
4. Підвищення продуктивності праці.
5. Покращене трасування та експедиція.
6. Ефективність витрат.
7. Конкурентна перевага.
8. Покращена оплата.

Використання EDI-партнерів з ланцюга поставок може подолати викривлення та перебільшення інформації про попит та пропозицію, вдосконалюючи технології для полегшення обміну фактичним попитом та інформацією про пропозицію в реальному часі.

1.4 Висновки до розділу 1

Отже, в даному розділі було проаналізовано та досліджено ринок логістичних послуг та існуюча інфраструктура, яка є інструментом управління цієї галузі. Помічені тенденції в загальній потребі розвитку інформаційних технологій в цій сфері, при отриманні вихідних даних стало зрозуміло на якому етапі логістичної ланки потрібен розвиток та автоматизація. Розглянуто існуючі рішення, що пропонує ІТ для логістики великого, середнього та малого бізнесу. Виявлено, що підприємці-початківці, невеликі компанії та загалом малий бізнес потребує мало затратних рішень для обробки логістичних замовлень, зокрема рішень, які не потребують закупівлі дорогих офісних приладів, ПК, ноутбуків, принтерів та іншої техніки. Виявлено значну кількість основних логістичних особливостей, що допомогли провести аналіз в багатьох подальших сферах діяльності. Оцінено переваги використання смартфона як інструменту по веденню бізнесу, що має низький поріг входження, відносно невелику вартість. Розглянуто недоліки та ризики при використанні мобільного пристрою як основного засобу зберігання, передачі та обробки інформації, незважаючи на них загальна ідея проявила себе більш ніж об'єктивно затребуваною. Попри всі деталі, що були розглянуті вище, загальна картина розробки програмного забезпечення управління логістичними замовленнями компанії є актуальною та затребуваною, що і призвело до подальшої розробки даного програмного продукту.

2 ПРОЕКТУВАННЯ АВТОМАТИЗОВАНОЇ СИСТЕМИ РОЗПОДІЛУ ЗАМОВЛЕНЬ

2.1. Аналіз ринку мобільних операційних систем

На сьогоднішній день існує дві основні мобільні операційні системи, Android та IOS, також є багато менш популярних, що займають близько одного відсотку : Tizen, Kai OS, Fire OS, Flyme OS, Lineage OS, Remix OS, Sailfish OS.

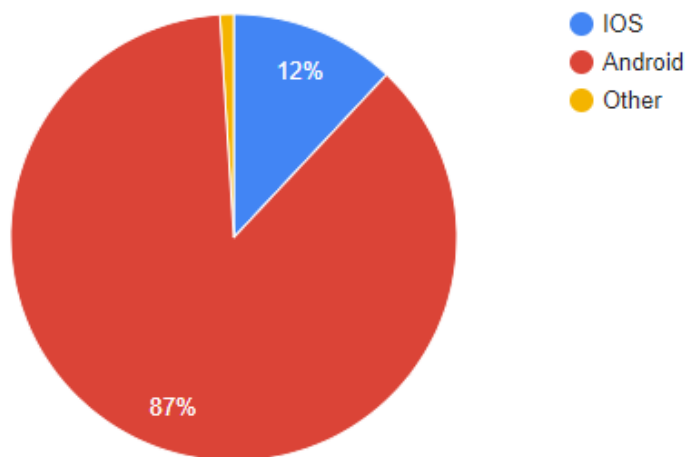


Рис. 2.1. Співвідношення ОС

Давно минули часи, коли мобільні телефони були лише пристроєм для здійснення телефонних дзвінків та надсилання випадкових текстових повідомлень. Сучасні смартфони ближче до портативних комп'ютерів, які дозволяють нам надсилати електронні листи, грати в ігри, дивитися новини та здійснювати відеодзвінки коханим. І багато іншого. Операційні системи, також відомі як мобільні ОС, - це програмне забезпечення, яке запускає та керує нашими настільними комп'ютерами та ноутбуками та управляє їх ресурсами і пам'яттю, коли вони використовуються для багатозадачності. Але вже деякий час смартфони також використовують операційні системи, і саме ця розробка

принесла розширені функції для мобільних телефонів, які раніше були доступні лише на наших комп'ютерах. Це також платформа, завдяки якій розробники можуть створювати додатки (застосунки, розроблені для смартфонів, які можуть виконувати певні функції). Доступні сотні тисяч програм, які постійно розробляються, кожна зі своїм призначенням. Наприклад, ви можете завантажити додаток для погоди, який повідомляє вам про поточну температуру або ймовірність дощу у вашому місті, додаток для новин або віджет, який надсилає останні заголовки прямо на головний екран вашого пристрою, або гру, щоб просто витратити час. Є кілька прикладів операційних систем для мобільних пристроїв, які включають Apple iOS, Google Android та ОС Windows Phone від Microsoft. Оскільки операційні системи для смартфонів настільки інтегровані із зовнішнім виглядом, відчуттями та функціями, можливостями мобільного телефону, багато людей базують свій вибір пристрою, навколо якої операційної системи він працює. Деякі мобільні операційні системи – являють собою програмне забезпечення з відкритим кодом, що означає, що немає обмежень щодо того, що ви можете завантажити на нього, або хто може розробляти його програмне забезпечення (часто існує «спільнота» розробників). Операційні системи з відкритим кодом можна повністю налаштувати, тоді як інші мають обмеження щодо типів програмного забезпечення, яким дозволено працювати на пристрої. Багатозадачна операційна система Apple iOS працює на iPhone, iPad та iPod від Apple. Спеціальна версія програмного забезпечення - це те, що також керує Apple Watch. iOS реагує на дотик користувача - дозволяє вам натиснути на екран, щоб відкрити програму, стискати пальці, щоб мінімізувати або збільшувати зображення, або проводити пальцем по екрану, щоб змінити сторінки. Apple iOS не можна використовувати в сторонніх системах, тому ви зможете використовувати його лише на продуктах, виготовлених Apple. Він постачається з веб-браузером Safari для використання в Інтернеті, програмою iPod для відтворення музики та поштою Apple для управління вашими електронними листами. Ви можете завантажити мільйони програм, які зараз доступні в App

Store, безпосередньо на будь-який пристрій під управлінням iOS, будь то iPhone або iPad. Вони охоплюють все - від книг рецептів до підручників з гітари до ігор.

ОС Android належить Google і працює на базі ядра Linux, яке можна знайти на широкому діапазоні пристроїв від бюджетних до флагманських пристроїв. Android - це операційна система, головна особливість якої відкритий код, яка дозволяє розробникам отримувати доступ до розблокованого обладнання та розробляти нові програми за власним бажанням. Це означає необмежений доступ кожному, хто хоче розробляти програми для телефону, і обмежує його ліцензування, тому користувачі отримують вигоду від маси безкоштовного контенту. В даний час Android є домінуючою платформою для смартфонів завдяки своїй величезній тязі з широким колом користувачів. Деякі з найкращих функцій Android включають можливість налаштування кількох головних екранів за допомогою корисних віджетів та додатків, які дають вам швидший та простіший доступ до вмісту та функцій, які вас найбільше турбують. Він також має чудову здатність до багатозадачності - з можливістю закривати програми, просто проводячи їх. І останнє, але не менш важливе: Google Play, який є еквівалентом Android App Store від Apple, є місцем знаходження мільйонів програм, багато з яких абсолютно безкоштовні. Windows Phone – також операційна система для смартфонів - Microsoft випустила надзвичайно оновлену версію своєї платформи для мобільних телефонів Windows наприкінці 2010 року, після того, як її програмне забезпечення відстало від iOS та Android.

Ми обрали ОС Андроїд, так як вона є найбільш розповсюдженою, тому розробка додатку для неї буде мати більшу кількість потенційних користувачів. Полегшує розробку також те, що у власному користуванні маємо смартфон під управлінням даної ОС. Також розробка проводиться мовою програмування Java, яка є досить популярною та розглядається як пріоритетна до вивчення. Особливістю Андроїд також є більш легке розповсюдження програми після її написання, а саме завантаження її до Google Play - онлайн магазину додатків та ігор, порівняно з іншими ОС.

Android - операційна система для смартфонів, планшетів та телевізорів, смарт годинників. Компанія Google придбала на той момент дуже перспективного розробника програмного забезпечення Android inc. в 2005 році. Операційна система Android розроблена на модифікованому та зміненому ядрі Linux. В результаті, Google та інші учасники співпрацюють для спільної розробки та вдосконалення цієї нової операційної системи. Далі проекту Андроїд з відкритим вихідним кодом Android (AOSP) було надано подальший розвиток та підтримку платформ. В Android є велика кількість співтовариств розробників, які розширюють набір функціональності пристроїв.

ОС Android має свій офіційний магазин з продажу додатків - Google Play Store. Включає в себе як платні застосунки, так і безкоштовні. Так як ОС Android відкрита, користувач пропонує можливість завантаження програм, ігор і т.д. з інших ресурсів. Розробники Андроїд, в основному, створюють додатки на мові Java, використовуючи ресурси та бібліотеки через розроблені портали Google. Офіційно про ОС Android стало відомо 05.11.2007 р. коли було оголошено основу Open Headset Alliance - об'єднання від великої кількості компаній (близько 80). Спочатку більшу частину коду Андроїд було видано під ліцензією вкористання Apache. Додатки для Android вміщують у собі Java-додатки та бібліотеки, які запускаються завдяки віртуальній машині Dalvik з компілятором JIT. Бібліотеки включають в себе систему управління, графіку OpenGL ES 2.0, движок WebKit, графічний движок SGL, SSL та бібліотеки Bionic. ОС Android - це 13 мільйонів кодів, у тому числі 4-х мільйонних XML, 3,9 мільйонів на C, 2,5 мільйона на Java та 1,9 мільйонів на C ++.

Android ОС, що встановлена в 85% проданих смартфонах станом на другий квартал 2020 року. Найсильніший конкурент Apple програв у гонці за технологічне домінування. Є дуже багато причин чому Андроїд краще, список досить довгий, основні причини приведені нижче.

1. Набагато більший вибір телефонів – це дуже очевидна перевага Android. Користувачі Apple мають обмежений вибір телефонів, доступних в продажу для оновлення, тоді як користувачі Android мають на вибір кілька марок, моделей,

форм факторів телефонів на різних рівнях цін. Сотні та тисячі телефонів інші, свідчать про широкий вибір, доступний для клієнтів. Різні телефони на вибір - це не просто перевага з точки зору бюджету, але також кожен пристрій може пропонувати різні зручності та функції. Цей рівень є черговою причиною того, чому Android кращий за iPhone.

2. Знімна картка пам'яті та акумулятор. Збільшення обсягу пам'яті в iPhone або iPad вимагає дорогого оновлення та змушує купувати дорожчу модель. Однак, у порівнянні плюсів і мінусів iPhone та Android, багато пристроїв Android мають розширювані слоти для карт microSD. Деякі акумулятори Android також можна знімати, що дозволяє замінити лише її, а й інші деталі телефону.

3. Доступ до найкращих віджетів Android. Віджети використовуються для швидкої взаємодії з застосунком, що додає функціональності та гнучкості пристроям Android. Ця функція є однією з найбільших причин, чому Android краще, ніж Apple. Але на Apple iPhone вони також почали з'являтися. Деякі з найкращих віджетів Android, такі як Battery Reborn та Circle Android Launcher, є популярними варіантами для покращення досвіду роботи Android.

4. Краще обладнання. Деякі флагманські телефони Андроїд добре конкурують з iPhone завдяки кращим параметрам. Наприклад, Samsung Galaxy S9 випереджає iPhone XS завдяки швидшому процесору, більшому обсягу оперативної пам'яті, кращій роздільній здатності екрана та збільшеній ємності акумулятора .

5. Кращі варіанти зарядки. Швидка зарядка Android подобається користувачам, коли справа доходить до того, чому Android кращий за Apple. Зарядити 50% батареї за 30 хвилин дуже зручно. Але є ще один доданий заряд для Android - бездротовий. Можливість бездротової зарядки пристроїв є основною перевагою Android вже деякий час, тоді як Apple лише нещодавно надала цю функцію.

6. Інфрачервоний порт. Чому б ви були раді, що ваш Android має інфрачервоний діапазон? Подумайте про зручну бездротову взаємодію між пристроями, і що більш важливо, він може виконувати функції пульта

дистанційного керування кондиціонером, телевізором та іншим. Це не обов'язково перше, про що ви думаєте, але це, безумовно, має свої переваги.

7. З використанням пристроїв Android ви можете купувати програми та інші застосунки за межами Play Store у таких місцях, як Amazon. Таким чином, такі речі, як десятки програм для Android, не обмежуються лише однією платформою. Це дає людям значно ширший доступ до програм та більше можливостей для задоволення їх інтересів.

8. Спеціальні клавіатури. Якщо вам не подобається ваша клавіатура Apple, ваші можливості обмежені. Swype одна з кращих Андроїд клавіатур. Це знову область, де з точки зору плюсів і мінусів iPhone проти Android, Android виходить на перше місце, оскільки пропонує альтернативні варіанти, такі як Swiftkey.

9. Google Play Store є більш зручним для користувачів. Як Apple App Store, так і Google Play для Android пропонують понад мільйон завантажуваних застосувань. Однак App Store має обмеження, такі як обов'язковий інтерфейс iTunes, що є не досить зручним, та являє собою вимогу відтворювати завантажені фільми на пристроях Apple. Натомість Google Play від Android є очевидною перевагою Андроїд, оскільки він використовує більш відкритий веб-інтерфейс для завантаження додатків і пропонує фільми, які можна відтворити на будь-якому пристрої з веб-браузером.

2.2. Аналіз основних характеристик сучасних систем управління логістичними замовленнями

Системи та способи управління логістикою які використовують великі компанії, організації та підприємства, використовують ПЗ для різних цілей, наприклад складу, транспорту або розподілу, вони являються досить багатофункціональними та важкими для розуміння. З поглядом на те, що така значна функція існує в повсякденній діяльності багатьох організацій, логістика так само, ще помітно досить повільно змінюється. Хоча роздрібна торгівля та E-commerce зіштовхуються з порушеннями після розривів і продовжують

виходити на перше місце, 50% вантажівок їдуть порожніми у зворотному напрямку, а склади переповнені товаром або простоюють.

Використання цифрової інформації може дозволити операціям із складування та транспортування підвищити досвід споживачів, надати більшу цінність компаньонам, а отже - створити ефективну екосистему різних частин, постачальників, ланцюгів поставок: виробників, перевізників, експедиторів тощо. Цифровізація часто зводиться до двох основних підходів: автоматизації та використання даних. Наше завдання описати, як системи управління логістикою можуть надати користь шляхом автоматизації більшості процесів та використання даних для прийняття обґрунтованих рішень. Система, що управляє логістикою як правило, працює у двох напрямках - прямому і зворотному. Коли ми говоримо про логістику, ми зазвичай маємо на увазі перший потік, який включає такі операції, перевірка та підготовка товарно-матеріальних цінностей, як отримання та обробка замовлення, упаковка та вибір товару, відправка його та вибір маршруту транспортування, який доставить товар чи послугу замовнику як швидко та ефективно. Зворотний потік означає будь-які операції з управлінням неправильними або пошкодженими вантажами, ремонтом складових частин та повторним використанням або переробкою. У відцифрованому світі для управління тими чи іншими процесами в обох напрямках підприємства використовують управління системою логістики - поєднання програмних інструментів, які оптимізують проходження всіх етапів від замовлення та доставки до домівки отримувача. Впровадження та інтеграція LMS у організації може здійснюватися по-різному: 1. Створення, придбання та управління програмним забезпеченням LMS самостійно.

2. Аутсорсинг - залучення сторонньої логістичної компанії (3PL), яка буде виконувати будь-які логістичні операції за працівників, використання 3PL дає швидкий термін впровадження, не потребуючи навчання своїх співробітників щодо нових процесів, пристроїв та програмного забезпечення.

Основні модулі системи управління логістикою. Управління замовленнями. Отримання та обробка замовлення в Інтернеті зазвичай включає операції

створення та редагування товарних запасів, управління обслуговуванням клієнтів, приймання платежів, перевірку на шахрайство та обробку документації між виробниками, постачальниками, складами та транспортними компаніями.

Для ефективної організації управління замовленнями та уникнення додаткових повторних помилок обробки заявки, товарні запаси, дані перевізників та клієнтів повинні бути одночасно відцифровані в одній системі. ПЗ для організації заявками одержує замовлення від майже всіх ваших каналів продажів і адмініструє їх з моменту їх розміщення до моменту погодження замовником доставки. Це ваш основний рівень підключення, де показується вся інформація про заявку: відповідні маршрути, місцезнаходження, інвентар, підключення до складу та інтеграція з бухгалтерською діяльністю для створення рахунків-фактур та прийому фінансів тощо.

Інтегрована OMS підключена до вашої CRM та бази даних про запаси, щоб доповнити послідовність продажів та надати інформацію для бухгалтерії та маркетингових відділів. Додаткові функції включають підключення до популярних торгових майданчиків, запаси постачальників та мультивалютні опції.

Управління запасами є життєво важливою ланкою ланцюгами поставок, відповідальним за контроль та документування кількості продукції, що продається. Отримання, зберігання та відстеження товарно-матеріальних цінностей, маючи справу з її швидкими та постійними змінами, вимагає надзвичайно точного управління інформацією про товар. Отже, автоматизація від класичних електронних таблиць до систем управління запасами (IMS) забезпечує необхідну чіткість, централізуючи всі дані в одному місці за допомогою системи управління запасами.

IMS відстежує та повідомляє про будь-які зміни на складі. Він підтримує баланс при поповненні, уникаючи запасів та надлишків запасів. Система відстежує низькі рівні запасів і встановлює автоматичне повторне замовлення для кожного товару. Тим часом це зменшує ризик замовлення надлишків, оцінюючи попит на продукцію. Управління складськими запасами - це сукупність процесів,

що підтримують, контролюють і автоматизують складські операції. Цей процес має в собі управління складським персоналом за допомогою числових показників діяльності, отримання предметів, зміни їх локації, підтримання безпечних умов праці та використання різноманітних програмних та апаратних застосунків для пошуку та знаходження предметів. Система організації складу складається з інструментів, які впорядковують робочий процес адміністрування товарів від прибуття на склад через розміщення та огляд в межах місця розташування до подальшого управління замовленнями та відправлення. Програмне забезпечення для управління складами може дати величезну перерву у виконанні та відстеженні запасів, у випадку з компаніями 3PL вони також надають величезну площу для зберігання продуктів організації. Як правило, WMS відповідає за наступні операції.

Дизайн складського приміщення. Забезпечуючи керовану тривимірну карту локації складу, ця функція дозволяє збільшити простір для збереження, та управляти розміщенням запасів та покращити розподіл предметів та робочої сили, надаючи більшої уваги ділянкам черги відвантаження, які вимагають додаткового нагляду.

Збір даних. Програма, синхронізована із деяким мобільним пристроєм, який являє собою сканер допомагає точно визначати предмети по всьому периметру складу. Відстежуючи штрих-коди, присвоєні виробам, WMS направляє виконавців замовлення до необхідних продуктів за допомогою оптимізованого маршруту збору. Щоб переконатися, що вибрано правильний лот, збирачі двічі перевіряють його: при скануванні вкладеного штрих-коду система перевіряє його. Що стосується швидкості, функція збору партії може заощадити час, оскільки вона дозволяє вибрати багато замовлень за один пробіг.

Упаковка. Замовлення можуть мати конкретні вимоги до упаковки, щоб гарантувати безпечну доставку або покращити досвід розпакування. WMS допомагає забезпечити упаковку у правильному порядку та максимально ефективно, відповідно до правил, встановлених бізнесом.

Управління над працею. Нагляд за людською стороною над складськими операціями передбачає використання менеджменту адміністрування працею, інтегрованої в систему. По-перше, система назначає завдання фірми окремим працівникам складу, застосовуючи можливості планування та освоєння робочої сили. Це дозволяє слідкувати за продуктивністю праці, знаходити неефективні заміни або працівників. Історичне середовище завдань показує всю історію діяльності співробітників, яку можна використовувати для аналізу максимальної праці, оптимізації робочих процесів та знаходження рішень там, де виникають труднощі. У дальнюгляжній перспективі це спонукатиме до підвищення продуктивності та ефективності при одночасному зменшенні затрат на заробітну плату працівникам.

Стратегічне транспортне планування Після того, як замовлення зібрано, останнє, що залишилось до виходу зі складу, - це оптимізація його відвантаження, а саме: вибір способу доставки. LMS з різних варіантів транспортної логістики визначає той, який найкраще відповідатиме вашим вантажним потребам. Методи доставки варіюються від сухих вантажних перевезень для продуктів, які не потребують охолодження, до вантажів, що зберігаються в холодильнику для товарів, чутливих до температури; від вантажу меншого вантажу (LTL) для ефективною доставки менших вантажів до важкого вантажу для негабаритних вантажів та об'ємних вантажів для великої кількості сировини тощо. Підключення до мережі оператора. LMS з'єднує вантажовідправників із перевізниками, забезпечуючи видимість для всіх зацікавлених осіб, щоб знайти найкращий варіант доставки.

Для вибору найдешевшого перевізника є інструменти закупівель транспорту. Ці інструменти допомагають налаштувати запит клієнта на пропозицію, відповісти на запити, прийняти пропозиції та оцінити заявки. У випадку 3PL, вони, як правило, не мають власних активів, а скоріше відповідають вимогам постачальників щодо доставки кваліфікованим постачальникам автопарку. Вважається, що логістика, яка не базується на активах, є більш ефективною: «3PL мають велику базу даних як компаній, так і приватних осіб, які

пропонують послуги з доставки. Оскільки вони не обмежуються власними активами, вони можуть запропонувати більшу ймовірність отримати роботу. " Визначення митних зборів та документації для глобального виконання. Хоча міжнародне судноплавство передбачає діяльність з великою кількістю паперової документації та постійно мінливими правилами контролю експорту, LMS може це зробити набагато простіше. Підтримуючи кілька мов і валют, система обчислює податки на додану вартість, транскордонні збори та транспортні експедиції. Крім того, LMS управляє необхідними документами з доставки, забезпечуючи дотримання як внутрішніх, так і транскордонних програм регулювання.

Управління транспортом. Основною загальною справою програмного забезпечення являється те, що відповідає потребам вантажним перевезенням і керує всіма деталями поставки, є програмне забезпечення по управлінню перевезеннями (TMS). Впроваджуючи рішення TMS, компанії шукають такі можливості:

Управління та планування поставок. TMS дозволяє користувачам системи планувати свої поставки через свій персональний онлайн-рахунок. Оцінка надання послуг базується на таких вкрай важливих параметрах, як щільність проживання населення, тип транспортного засобу та пропускна спроможність, прогнозована аналіз трафіку тощо. TMS забезпечує надшвидку доставку за допомогою автоматичного відстеження активів та швидкого оповіщення як вантажовідправникам, так і користувачам, якщо завантаження відстає від прийнятого графіка.

Перехресний док. Ця функція визначає тип доставки, який позбавляє потреби зберігати продукцію на складі, оскільки вона відвантажується безпосередньо від виробника до замовника. Це може скоротити час доставки, заощаджуючи при цьому робочу силу та складські площі. Однак перехресне стикування ризикує втратити контроль над запасами. Отже, для ефективного використання перехресного "докінгу" потрібні потужні процеси контролю запасів.

Логістика останньої милі. Останній етап шляху доставки - від розподільчого центру / об'єкта до остаточного споживача - може складати від декількох блоків до 50 або 100 миль. Для доставки замовлень споживачам компанія “логістика останньої милі” залучає перевізників посилок або малих пакетів. Якщо ви хочете дізнатись про інтеграцію з доставкою електронної комерції та перевізниками.

Відстеження замовлення. Щоб отримати повну видимість руху товарів, клієнтам надається доступ до Інтернет-порталу, який відстежує відвантаження в дорозі, попереджаючи про будь-які винятки з транзиту або непередбачені затримки. Зазвичай відстеження місцезнаходження передбачає подальші звіти у базі даних про місцезнаходження автомобіля в режимі реального часу.

Інший метод відстеження пакунків - повідомляти про прибуття або виїзд об'єкта та фіксувати ідентифікацію об'єкта, місце, де його спостерігали, час та статус. Не тільки постачальники, але, звичайно, клієнти потребують розуміння прогресу доставки. Надаючи ідентифікатор замовлення, вони можуть відстежувати просування товарів у відповідній заявці.

Таблиця 2.1. – Основані проблеми та їх рішення в логістичній галузі

Проблема	Рішення
Висока вартість ПЗ	Безкоштовний мобільний додаток
Затрати на велику кількість офісної техніки	Смартфон під управлінням ОС Android
Відсутність чіткої структури	Чітко структурований PDF документ
Втрата даних замовлень та файлів	Замовлення зберігаються на пристрої де встановлена програма

Транспортний облік. TMS інформує клієнтів про витрати на доставку, оскільки збирає всі документи, пов'язані з відвантаженням, в одному місці. Система спрощує бухгалтерський облік, розподіляючи витрати, призначаючи коди рахунків для підзвітності та бюджетних цілей, формуючи та оплачуючи рахунки за вантажі тощо.

Зворотна логістика. Управління поверненнями від споживача назад до виробника є складним для логістики процесом. Отже, варто застосовувати систему зворотної логістики (RLS), яка допоможе впорядкувати процеси ремонту, повернення та перерозподілу продукції.

Інформаційне відстеження. Коли здійснюється повернення, потрібно відстежувати інформацію про місцезнаходження товару та причину повернення. RLS може допомогти в управлінні цією інформацією, тому ви знаєте, які товари були повернуті, причину повернення та якщо вони були відправлені назад замовнику. Повертає статус. Надаючи клієнтам видимість їхньої віддачі, ви збільшите їх задоволення. Хорошим способом цього є надання миттєвого зворотного зв'язку щодо дій щодо повернення замовлення на порталі повернення або за допомогою сповіщень через емейл / SMS.

Відстеження забезпечення якості. Коли виникають проблеми з якістю, а продукцію потрібно поставити на карантин, відкликати або відремонтувати, система повідомить відповідних людей, щоб вони могли вирішити проблему.

Логістична аналітика. Використовуючи аналітичні та великі дані, ПЗ може досліджувати історію доставки та покращувати роботу клієнтів, що мінімізує витрати на логістику та терміни доставки вантажу зменшуються. Застосовуючи бізнес аналітику, програмні застосунки можуть розробляти показники ефективності та KPI, а також візуалізувати інші комп'ютерні моделі для прогнозування проблем у ланцюгу доставок. Оскільки ЛМС збирає значний спектр інформації, її слід ефективно використовувати. Можливості подання звіту програмного забезпечення для логістичної аналітики сприяють кращій можливості огляду та контролю над майбутніми логістичними результатами.

2.3. Дослідження основних задач та труднощів при проектуванні під операційну систему Андроїд

Операційна система Android досить молода, адже вона зародилася в час коли популярності почали набувати телефони з сенсорним управлінням, та їх дисплеї почали збільшувати свої діагоналі. Але вона не поступається по кількості технологій, можливостям розробки на різних мовах програмування, та має безліч середовищ розробки, які програмісти можуть використовувати як зручний інтерфейс для створення різноманітних ігор, бізнес додатків, програм по розрахунку бюджету, веб браузерів, соціальних мереж, перекладачів, месенджерів, навчальних програм, карт, графічних та відео редакторів.

Тому може постати питання яку про те, яку мову програмування використовувати, та в якому середовищі розробки розпочати конструювати графічний дизайн програми, прописувати код, тестувати на наявність збоїв та інших помилок.

Розглянемо мову Java. По-перше, Джава була офіційно застосованою мовою для розробки додатків для Андроїд (але тепер її замінив Kotlin), а отже, це також найбільш вживана мова. Багато програм у Google Play побудовані на Java, і це також найбільш підтримувана мова Google. На додаток до всього, Java має чудове Інтернет-співтовариство для підтримки у разі виникнення проблем, які скоріше за все будуть .Однак Java - це складна мова для початківців, оскільки вона містить складні теми, такі як конструктори, винятки з нульовим покажчиком, паралельність, перевірені винятки тощо. Крім того, Андроїд Software Development Kit (SDK) підвищує складність на новий рівень! Загалом, Java - чудова мова, щоб випробувати всі радості розробки додатків для Android. Однак це може бути трохи складно з початківцями, які вважають за краще почати з чогось простішого, а потім повернутися до цього.

Котлін. Зараз Kotlin являє собою офіційну мовою для розробки застосунків для Андроїд, оголошеною Google з 2019 року. Kotlin - це міжплатформна мова програмування, яка може використовуватися як альтернатива Java для розробки

застосунків для Андроїд. Він також був представлений як вторинна “офіційна” мова Java у 2017 році. Kotlin може взаємодіяти з Java, і він працює на віртуальній машині Java. Єдина значна відмінність полягає в тому, що Kotlin видаляє зайві функції Java, такі як винятки з нульовим показником. Це також усуває необхідність закінчення кожного рядка крапкою з комою. Коротше кажучи, Kotlin набагато простіший для входу в програмування у порівнянні з Java, і його також можна використовувати як "точку входу" для розробки додатків для Android.

C ++ можна використовувати для розробки програм для Android за допомогою Андроїд Native Development Kit (NDK). Однак програму неможливо створити повністю, використовуючи C ++, а NDK використовується для реалізації частин програми у власному коді C ++. Це допомагає використовувати бібліотеки коду C ++ для програми за необхідності. Хоча C ++ в деяких випадках корисний для розробки додатків для Android, його набагато складніше налаштувати та набагато менш гнучкий. Це також може призвести до більшої кількості помилок через підвищену складність. Отже, краще використовувати Java порівняно з C ++, оскільки вона не забезпечує достатнього виграшу, щоб компенсувати необхідні зусилля.

C # дуже схожий на Java, тому ідеально підходить для розробки додатків для Android. Як і Java, C # також реалізує збір сміття, тому є менше шансів на витік із пам'яті даних. І C # також має більш чистий і простий синтаксис, ніж Java, що робить кодування з ним порівняно простішим. Раніше найбільшим недоліком C # було те, що він міг працювати лише в системах Windows, оскільки використовував .NET Framework. Однак цією проблемою займався Xamarin. Android (з використанням Mono для Android), який є крос-платформною реалізацією інфраструктури спільної мови. Тепер інструменти Xamarin. Android можна використовувати для написання авторських програм для Андроїд та обміну програмним кодом на декількох платформах.

Python можна використовувати для розробки додатків для Андроїд, навіть якщо Андроїд не підтримує власну розробку Пайтон. Це можна зробити за допомогою різних інструментів, які перетворюють програми Пайтон у пакети

Андроїд, які можуть працювати на пристроях Андроїд. Прикладом цього є Kivu - бібліотека Пайтон з відкритим кодом, що використовується для розробки мобільних додатків. Він підтримує Андроїд, а також заохочує швидку розробку додатків (що, на мою думку, є безпрограшним варіантом!). Однак недоліком цього є те, що Kivu не матиме власних переваг, оскільки він не підтримується у власній формі.

Corona - це набір для розробки програмного забезпечення, який можна використовувати для розробки програм для Андроїд за допомогою Lua. Він має два режими роботи, а саме Корона Simulator та Корона Native. Корона Simulator використовується для безпосереднього створення застосунків, тоді як Корона Native використовується для інтеграції коду Lua з проектом Андроїд Studio для створення програми з використанням власних функцій. Хоча Lua трохи обмежений у порівнянні з Java, він також набагато простіший і має простішу криву навчання. Більше того, є вбудовані функції монетизації, а також різні активи та плагіни, які збагачують досвід розробки додатків. Корона в основному використовується для створення графічних додатків та ігор, але аж ніяк не обмежується цим.

HTML, CSS, JavaScript Програми Андроїд можна створювати за допомогою HTML, CSS та JavaScript, використовуючи фреймворк Adobe PhoneGap, який працює на базі Apache Кородова. Структура PhoneGap в основному дозволяє використовувати навички веб-розробки для створення гібридних додатків, які відображаються через "WebView", але упаковані як програма. Хоча фреймворку Adobe PhoneGap достатньо для базових завдань у сфері розробки додатків для Android, навряд чи він вимагає великого програмування, крім JavaScript. А оскільки для створення гідної програми потрібно багато попрацювати, краще використовувати інші мови у цьому списку, якщо ви хочете, щоб вас називали справжнім розробником Андроїд.

На ОС Андроїд припадає понад 81,7% світової частки ринку смартфонів. Жодна інша операційна система не наближається до цих смартфонів. І тенденція, здається, не уповільнюється. У 2017 році ця кількість становила 75%. Все більше

активних користувачів смартфонів обирають Android. Він простий у використанні, більшість додатків створені саме для нього, і люди мають вибір використовувати будь-яку марку телефону, яку вони хочуть. Різноманітність та свобода - це те, що найбільше приваблює користувачів. Через усе це більшість розробників сприймають Android як найкращу операційну систему.

Найпоширенішими видами написання програмного забезпечення є редагування вихідного коду, побудова виконуваних файлів та налагодження. IDE - це набір програм, де ви можете виконувати всі ці дії в одному місці. Ви можете виконати весь життєвий цикл розробки програмного забезпечення в IDE. В результаті було створено багато різних інтегрованих середовищ розробки (IDE) для розробки мобільних додатків. Багато нових з'являється, а старі еволюціонують. Сьогодні ми говоримо про найкращі IDE для Андроїд для розробників. Ми почнемо зі списку, а потім поговоримо про них детально. IDE для розробників Android:

- AIDE
- DroidScript
- CppDroid
- Android Web Developer (AWD)
- Python Suite
- Java Suite
- Eclipse
- Arduino
- Visual Studio
- Unreal Engine
- Android Studio
- Corona
- PhoneGap

Android Studio - офіційна IDE для Андроїд. Це набір програм та функцій, створений Google і має всі вбудовані інструменти для створення високоякісної програми для Android. Андроїд Studio в основному відома своєю здатністю

прискорювати процес розробки, не втрачаючи при цьому жодної якості. Новачок у розробці Андроїд, при використанні Android Studio матиме безліч підручників із створення додатків.

Ось найкращі особливості:

- Застосувати зміни - можна вносити зміни в програму без необхідності перезапускати програму.

- Інтелектуальний редактор коду - отримуйте пропозиції щодо кращого коду під час набору тексту.

- Швидкий та багатофункціональний емулятор - Дозволяє дуже швидко встановлювати та запускати тестові програми для різних пристроїв Android Шаблони коду та зразки програм - знайдіть шаблони з проектів, подібних до вашого власного.

- Інтелект - 365 різних перевірок ворсу у вашому додатку Інструменти та фреймворки для тестування - запускайте тести на своєму пристрої чи емуляторі, у середовищі безперервної інтеграції або в тестовій лабораторії Firebase.

- Серед інших функцій Android Studio дозволяє підключатись до інших файлів проекту, таких як C / C ++. Також існує інтеграція хмар Google.

2.4. Алгоритм вибору мови програмування

Алгоритм вибору мови для програмування може залежити від досить багатьох факторів. Здебільшого вибір припадає на ту мову програмування, яку ви вже вивчали та маєте основні знання про синтаксис мови, її функціональні можливості, загальні стилі програмування цією мовою.

Є різні мови програмування для розробки додатків для Android - це платформа, що використовує відкритий код для розробки програмного забезпечення, керована Гугл. Хоча Гугл має власні мобільні пристрої для телефонів та планшетів, інші виробники, такі як Samsung, Huawei та інші, також виробляють телефони та планшети, що працюють на ОС Android.

Щоб створити додаток для Android, вам потрібно отримати набір інструментів для розробки Android, який містить налагоджувачі, емулятори та необхідний SDK. Найкращим інтегрованим середовищем розробки (IDE) для розробки додатків для Android є Android Studio. Доступні й інші варіанти, але Android Studio, безумовно, є найпопулярнішим. Зазвичай IDE для Android можна запускати в будь-якій ОС, включаючи Windows, Mac та Linux.

З моменту офіційного запуску Android у 2008 році Java стала мовою розробки за замовчуванням для написання програм для Android. Ця об'єктно-орієнтована мова була спочатку створена ще в 1995 році. Незважаючи на те, що Java має неабияку частку несправностей, вона все ще є найпопулярнішою мовою для розробки Android. Більшість інших мов Android вважаються версією Java або результатом розвитку Java. Google оголосив, що почне підтримувати мову програмування Котлін у 2017 році. Це альтернативна мова традиційній Java для розробки Android. Навіть як нова мова, вона дуже популярна.

Kotlin та Java є взаємодіючими, тобто вони можуть використовувати одну і ту ж інформацію. До всіх ваших бібліотек Java можна отримати доступ за допомогою Kotlin. З точки зору виконання, мова Kotlin відповідає Java Bytecode. Загалом, це вважається більш охайною та чистою версією Java. Мови програмування для рідного розвитку Як я вже говорив раніше, усі рідні мови прикладного програмування мають свої плюси і мінуси. Незалежно від того, використовуєте ви Objective-C або Swift для Apple, або Java, або інший смак Java (наприклад, Kotlin) для Android, це переваги та недоліки. Плюси власного програмування: Найбільший контроль над пристроєм Низькорівневе кодування для передових технологій, доданих до пристрою. Найшвидший доступ до найновіших і найкращих функцій за допомогою вашої мови, найшвидший у виконанні нижній рядок, рідні мінуси програмування: Повільно розвивається найдорожчий метод розробки. Для створення iOS та Android потрібні найкваліфікованіші та спеціалізовані розробники додатків.

Незважаючи на те, що рідні мови програмування дають вам найбільший контроль над вашим додатком, їх важко вивчити, і вони довго розробляються.

Якщо ви не створюєте вузькоспеціалізований застосунок, вам, ймовірно, не потрібно буде йти рідним шляхом. Мови програмування для гібридних програм. Гібридні програми розробляються один раз, але написані мовою програмування, яка працює на декількох платформах.

Найчастіше одна розробка буде працювати як для iOS, так і для Android. Хоча деякі гібридні мови поширюють свою функціональність на інші платформи, такі як PWA (прогресивні веб-програми). Це приємно мати тим, хто переходить у більш зручне для Інтернету середовище. Створюючи гібридний додаток, ви зазвичай маєте справу з якоюсь мовою, фреймворком чи набором інструментів на основі JavaScript. Давайте детальніше розглянемо деякі з цих варіантів нижче. Ксамарин та C # Розроблений Microsoft, C # (вимовляється як C sharp) - ще одна об'єктно-орієнтована мова програмування. Зрештою Microsoft придбала фреймворк Xamarin, який дозволяє розробникам програм програмувати за допомогою C # для інших фреймворків. Інші типи альтернатив із низьким кодом, такі як OutSystems та Kony, мають SDK, який можна використовувати з різними мовами, а не лише з однією. Використовуючи IDE для гібридної розробки, код C # скомпільований для власної роботи на пристроях iOS та Android.

Мови JavaScript Найпопулярніші гібридні мови використовують фреймворки JavaScript. Це своєрідна мова програмування загального призначення для різних випадків використання. React Native Appcelerator Кордова / PhoneGap Це лише декілька найкращих варіантів. У Google навіть є кілька розробників ніш під назвою Dart and Flutter. React Native та Appcelerator I Reactive Native, і Appcelerator використовують JavaScript для спілкування із заздалегідь побудованою функціональністю, яка є рідною для їх фреймворку. Це дозволяє вам маніпулювати інтерфейсом користувача, збирати дані та отримувати дані, щоб ви могли їх представити користувачеві. По суті, це означає, що ви сильно покладаетесь на JavaScript для маніпулювання власними компонентами. Програмування програми таким чином має свої плюси і мінуси. Реагуйте на Native та плюси Appcelerator: Використання Java для маніпулювання чимось рідним Доступ до власних функціональних можливостей безпосередньо з

JavaScript React Native і мінуси Appcelerator: Не зачіпає таких речей, як HTML5 або CSS, які є технологіями, які зазвичай використовуються з будь-ким, хто розробляє в JavaScript, щоб вільно маніпулювати своїм власним інтерфейсом користувача Перешкода для вивчення цього методу програмування знаходиться на середньому рівні.

Кордова / PhoneGap Фреймворки типу Cordova / PhoneGap та Ionic насправді просто побудовані поверх мови програмування Apache Cordova. Гібридні програми, що використовують цю мову, будуються шляхом перенесення веб-інтерфейсу на рідну версію. Цей метод дозволяє створювати так само, як і для веб-сайту. Тож якщо ви веб-розробник, ви почуватиметесь тут як вдома. Він використовує Javascript, HTML та CSS. Це веб-середовище переноситься спочатку на iOS та Android. Плюси: Низький бар'єр для в'їзду Той, хто має досвід веб-розробки, може легко розпочати програмування таким чином Крива навчання легка Мінуси: Втрата частоти кадрів Якщо ви розробляєте програму для ігор або доповненої реальності (AR), це рішення, мабуть, не найкраща мова для вас. Обидва вони потребують вищої частоти кадрів

2.5 Висновки до розділу 2

Отже, в даному розділі здійснено аналіз та порівняння основних мобільних операційних систем, а саме Android та IOS. Розгляд загальної інформації, що доступна в широкому доступі в мережі Інтернет показав переваги Android та його більшу популярність серед користувачів мобільних девайсів. Відповідно було доведено, що розробка ПЗ на цю ОС потенційно може мати більшу популярність та затребуваність серед користувачів.

Було досліджено характеристики сучасних мов програмування, які можуть бути використаними при розробці ПЗ для мобільної ОС Android. Шляхом вивчення документації, яка розроблена самою компанією Google, що наразі являється власником найпопулярнішої мобільної ОС було прийнято рішення вести розробку системи управління логістичними замовленнями компанії на мові

програмування JAVA, що й досі являється пріоритетною мовою при написанні програмного коду. Виявлено перспективи розвитку відносно нової мови Kotlin, яка є спрощеною відносно мови JAVA та використовує багато ресурсів споріднених зі своєю спадкоємицею.

Проаналізовано основні середовища розробки. Таким чином, з основних офіційних джерел Google були взяті рекомендації щодо використання Android Studio, був опанований функціонал даної програми, налаштовані відповідні робочі вікна для подальшої зручної роботи. Також перевірено емулятор Android телефонів, де в живому режимі, динамічно можна слідкувати за виконанням розробленої програми. Перевірено можливості аналогів, серед яких можна відзначити середовище розробки Eclipse, що має менш навантажений інтерфейс та схожу загальну функціональність в порівнянні з Android Studio.

3 ПРОГРАМНА РЕАЛІЗАЦІЯ АВТОМАТИЗОВАНОЇ СИСТЕМИ РОЗПОДІЛУ ЛОГІСТИЧНИХ ЗАМОВЛЕНЬ МОВОЮ JAVA

3.1. Визначення основних цілей та задач

Перед початком розробки ПЗ потрібно розробити план дій, список послідовності, який приведе нас від ідеї створення мобільного додатку, до фінального запуску та повноцінного використання програмного продукту, який може мати наступний вигляд (Рис. 3.1.).

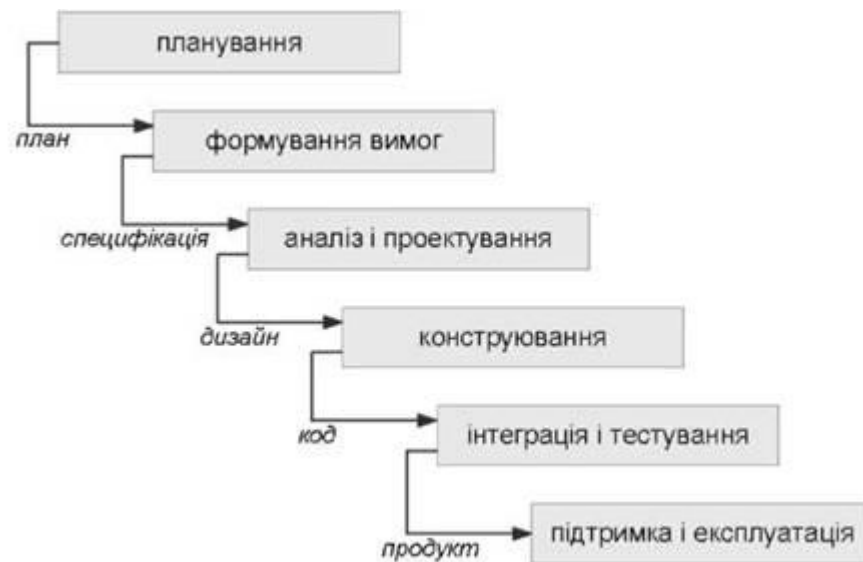


Рис. 3.1. Послідовна модель

Фізична топологія буде дещо відрізнятись і мати такий вигляд (Рис. 3.2.).

Етапи життєвого циклу які є в розробці програмного забезпечення. Є шість етапів життєвого циклу розробки програмного забезпечення, і вони дотримуються певного порядку, крім певних обставин. Ці етапи - це планування, аналіз, проектування, впровадження / розробка, тестування / інтеграція та технічне обслуговування. Дотримуючись цього етапу розвитку, програмні компанії можуть розробляти програмне забезпечення передбачуваним та керованим способом.

Планування. Всі проекти з розробки програмного забезпечення починаються на етапі планування. Тут формується початкова ідея програмного

забезпечення. Часто виникає проблема в бізнесі чи галузі, яку потрібно вирішити. Замість того, щоб звертатися до доступних опцій програмного забезпечення, компанії, що розробляє програмне забезпечення, доручено створити нове рішення проблеми. Етап планування зосереджуються на виявленні проблеми, зборі інформації, необхідної для планування рішення, та перегляді всіх наявних даних. Це найважливіша частина проекту, оскільки ефективне планування може усунути більшість проблем. Є кілька важливих речей, про які слід пам'ятати, починаючи фазу планування: Ця фаза нагадує мозковий штурм - починаючи з усіх ідей на столі і звужуючи до конкретного плану атаки до кінця фази (а також фільтрації через безліч підфаз).

Як і на будь-якій стадії будь-якого процесу, спілкування є першорядним. Зокрема, прозорі розмови між усіма, хто бере участь у розробці програмного забезпечення. тобто зацікавлені сторони, кінцеві користувачі та команда проекту - повинні бути частими. На етапі планування починаємо збирати вагомі докази, щоб прийняти обґрунтоване рішення про те, як діяти далі. Важливо знайти докази того, що потрібно обґрунтувати свої ідеї. Чим більше поінформованих та впродовж вашого дослідження на цій стадії, тим більше шансів уникнути серйозних проблем у майбутньому. Проводити співбесіди та розповсюджувати опитування для уточнення та підтвердження вимог зацікавлених сторін та користувачів

Системний аналіз. Системний аналіз - це, по суті, техніко-економічне обґрунтування, щоб перевірити, чи є ваша ідея життєздатною. Мета полягає в тому, щоб дослідити ідею вашого програмного забезпечення з об'єктива керівника бізнесу, який намагається уникнути поганих інвестицій. Вам доведеться розробити решту ідеї та знайти шляхи, щоб виправдати її розвиток. Значна частина роботи на цьому етапі окреслить етапи проектування та розробки пізніше. Починаючи процес системного аналізу, важливо пам'ятати: Крок 2 є подальшою деталізацією вищезазначеного кроку: буде визначена вся система програмного забезпечення, включаючи план кожного етапу розробки програмного забезпечення. Цей «обсяг роботи» розбиває проект на менші частини, щоб легше

визначати завдання на кожному етапі та забезпечувати належне управління ними, щоб нічого не прослизнуло через тріщини. Залучивши стільки сторін - розробників, тестувальників, дизайнерів, співпраці з клієнтами, керівників проектів, тестувальників тощо - неважко зрозуміти, чому навіть тим, хто працює над проектом на пізніх стадіях, такий розподіл відповідальності буде корисним.

Дизайн системи. На етапі проектування системи створюється повністю розроблений дизайн вашого програмного забезпечення. Тут відбуваються всі проектні роботи, щоб команда розробників могла працювати над проектом. У багатьох випадках обидві команди працюють на даний момент, оскільки команда розробників може розпочати створення систем для подальшого розвитку та координації ресурсів. Наприкінці процесу проектування компанія повинна скласти корисний план програмного забезпечення, включаючи його вигляд, функціонування та побудову. Починаючи етап проектування, пам'ятайте про те, що цей крок багато в чому говорить сам за себе. На цьому етапі розробники та кодери розробляють програмне забезпечення як для інтерфейсу, так і для внутрішнього функціоналу. Системні аналітики також сприятимуть розробці, щоб забезпечити належну функціональність протягом усього досвіду роботи користувача.

Кодування / Розробка. На етапі розробки відбувається збір програмного забезпечення. Це включає різноманітні процеси, включаючи кодування, налаштування інфраструктури та створення документації про те, як працює система. Розробники можуть співпрацювати з дизайнерами, щоб переконатися, що їх робота відповідає дизайну. Якщо є проблема, команда розробників може співпрацювати з командою розробників, щоб знайти рішення. На етапі розробки та кодування основне: з розробленим програмним забезпеченням для каркасних програм кодери можуть вільно зануритися в дрібницю. Це основна частина проекту; де побудовано серце програмного забезпечення. Іншими словами, саме тут ідеї, викладені на перших 3 кроках, стають реальністю, коли команда програмістів втілює їх у життя. Завдання кодування розподіляються відповідно до делегованих обсягом роботи, створеної на етапах 1 і 2 в процесі, який називається

"розподіл завдань". Цей підрозділ праці гарантує, що всі програмісти знають, за які коди відповідають, за що вони здатні максимізувати ефективність.

Тестування. Коли основна частина роботи буде завершена, вона може бути відправлена на тестування системи. Все програмне забезпечення ретельно перевіряється перед оприлюдненням. Команда QA використовує такі інструменти, як автоматизовані тестери, для швидкого випробування сценаріїв, щоб вони могли знайти проблеми в програмному забезпеченні. Важливо пам'ятати: Після того, як програмісти поставили крапки над їх і і перекреслили їх. Програмне забезпечення надсилається до відділу контролю якості команди для ретельного тестування продукту на наявність помилок. У успішних тестах певна кількість збоїв обов'язково буде виявлена. А потім ремонтуються, перш ніж будь-який користувач коли-небудь отримає повідомлення про можливу помилку. Як тільки кожен тест можна пройти без помилок, програмне забезпечення відправляється для впровадження.

Впровадження (запуск та підтримка) Коли все знаходиться у готовому для продажу стані, продукт запускається. Під час запуску групи підтримки дизайнерів та розробників працюють разом, щоб підтримати клієнтів. Це включає збір даних про проблеми, а також про те, що хочуть користувачі. Таким чином, команда може знайти рішення та внести виправлення за потреби. На етапі впровадження важливо пам'ятати, що: Заключна фаза життєвого циклу цього програмного продукту. Тут програмне забезпечення відкриває бета-фазу, яка передбачає реальне використання клієнтами та компаніями, які адаптують його після запуску. Якщо виникає якась помилка, про це повідомляється членам команди з контролю якості, які працюють із програмістами для виправлення ремонту. Після заздалегідь встановленого періоду часу, протягом якого програмне забезпечення може працювати без помилок, воно готове до офіційного запуску для всіх користувачів.

Судити про кінець проекту розвитку може бути складно. Має бути наданий широкий обсяг робіт, укомплектований графіками, завданнями та еталонами для досягнення успіху. Назвемо це найменшим мінімумом. Важливо не зупинятися на

досягнутому, особливо якщо співпраця була приємною - сильні лідери будуть шукати можливості для просування свого зоряного проекту у постійне взаємовигідне партнерство. Тут виникає „циклічність” або „вища спіраль” життєвого циклу (програмного забезпечення): постійне вдосконалення минулого та створення майбутнього.

Також буває спіральна модель розробки ПЗ зображена далі(Рис. 3.2.).

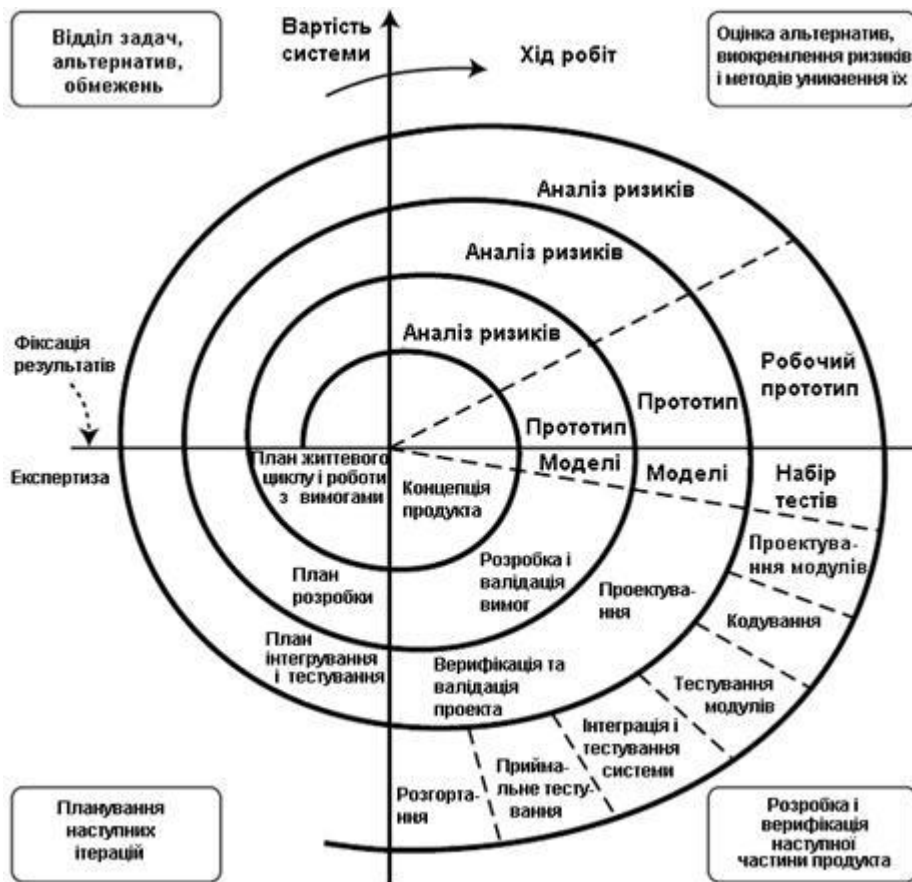


Рис. 3.2. Спіральна модель

Ми можемо розглянути за приклад ітераційну модель розробки ПЗ, проаналізувавши переваги та недоліки зі спіральною, та послідовною. Наглядно можемо побачити її особливості на наступній візуалізації. (Рис. 3.3.).

Інкрементна модель



Рис. 3.3. Інкрементна модель

3.2. Розробка дизайну мобільного додатку

Дизайн для планшетів або мобільних пристроїв сильно відрізняється від інших типів дизайну, таких як плакати, мультфільми, журнали чи навіть веб-сайти. У наш час люди дуже звикли до смартфонів і їх дуже важко здивувати. Тому дизайнери стикаються з більшими проблемами для створення інноваційних, красивих та унікальних інтерфейсів. Люди хочуть залишити програми, якими вони дійсно користуються або які їм подобаються, інакше програми видаляються. Навіть гірше, користувачі можуть опублікувати поганий відгук і вплинути на ефективність програми в магазинах. Дизайн відіграє дуже важливу роль у процесі розробки мобільних додатків. На них покладено обов'язок зробити додаток візуально привабливим та зручним для користувачів.

Проста анімація, компоненти, мінімалізм, прості кольорові схеми та поетапні дії. Елементи інтерфейсу повинні бути інтерактивними. Крім того, шрифти, кольори, піктограми, анімація та всі інші елементи повинні бути представлені просто. Природні взаємодії зосереджені на, адаптація відео / зображення може бути корисною, щоб привернути коротку увагу.

Тому розробка дизайну являється дуже важливою складовою при розробці ПЗ в цілому. Від початку до кінця я використовував середовище розробки Android Studio 4.1.3. Основними задачами до дизайну мобільного додатку було використання стандартних застосунків Android, та загальний мінімалістичний стиль, що не містив би в собі нічого зайвого, а лише практичність до використання. Головний екран на який потрапляє користувач містить коротке привітання, та кнопку для початку роботи (Рис. 3.3.).

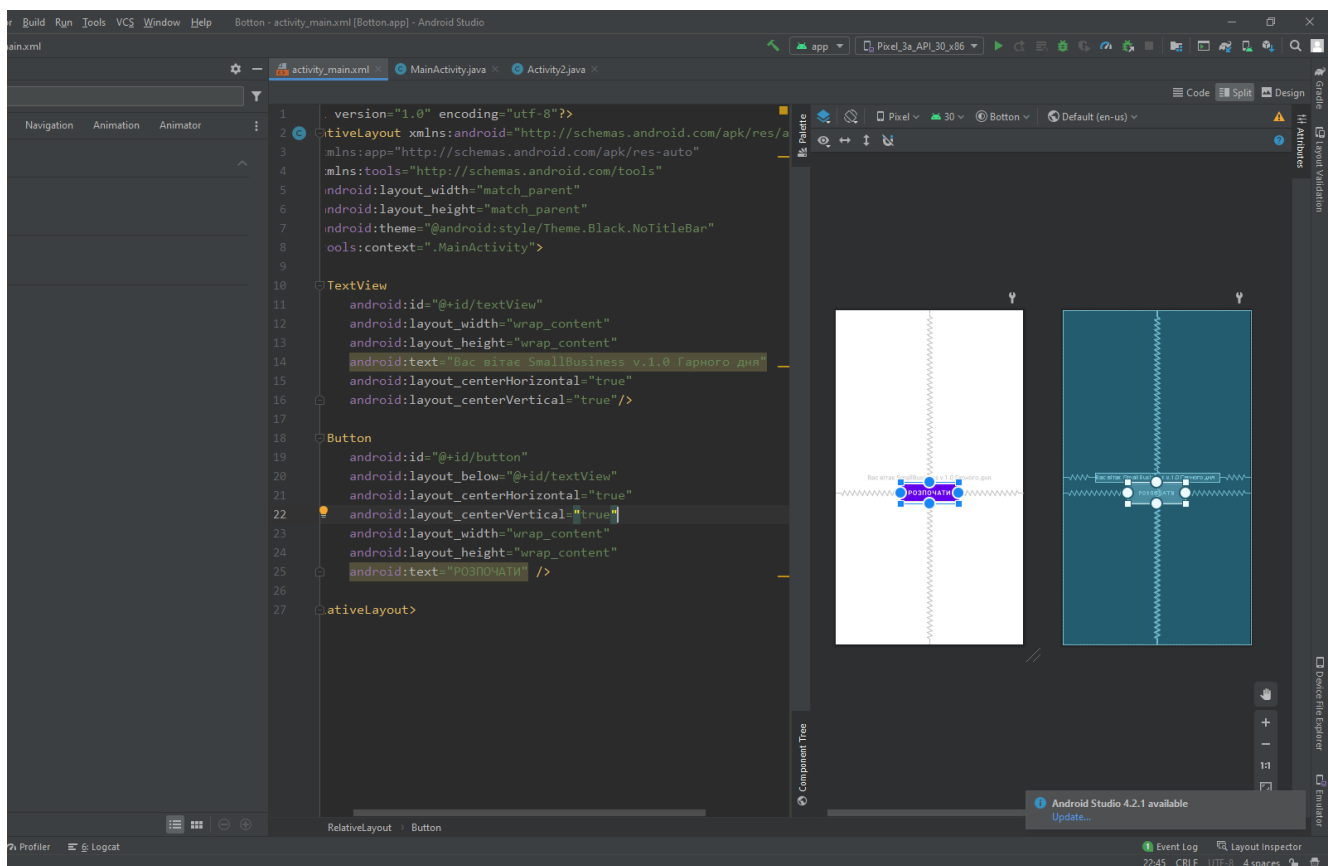


Рис 3.3. Головний екран

Після ознайомлення з сторінкою привітання, користувач натискає на кнопку “Розпочати” та переходить на наступний екран, де переходить саме до оформлення логістичного замовлення. Там він може перейти до вводу даних, а ввести основні дані логістичного замовлення, ПІБ замовника, його номер телефону, та електронну пошту, цих даних достатньо для того, щоб підтримувати оперативний зв’язок з замовником, та в разі потреби терміново з ним зв’язатись.

Далі йде поле для вводу даних про одержувача так само нам потрібно ввести ПІБ, номер телефону, електронну пошту, також додаємо дату доставки та коментарі до замовлення. Цього набору даних достатньо для оформлення логістичного замовлення, яке буде виконано з найменшими ресурсними затратами, фінансовими, логістичними та затратами часу. Після вводу всіх даних користувач має натиснути кнопку “створити замовлення”.

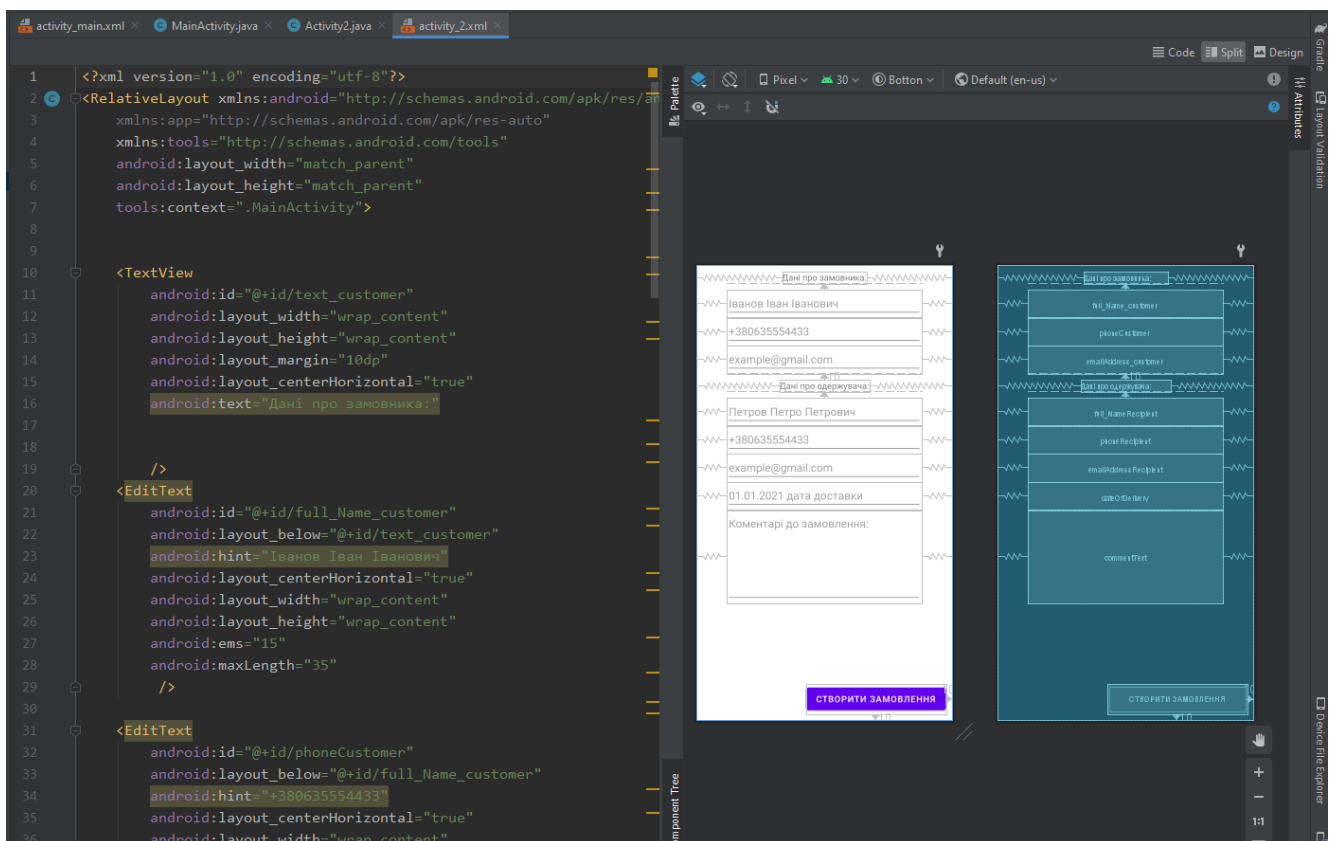


Рис 3.4. Екран створення замовлення

Середовище розробки Android Studio надає можливість тестувати як графічну складову так і функціональну в спеціальному емуляторі, який імітує

роботу звичайного смартфона, на якому встановлена ОС Android, тому ми можемо в режимі реального часу переглянути чи коректно відображається наш додаток (Рис. 3.5.).

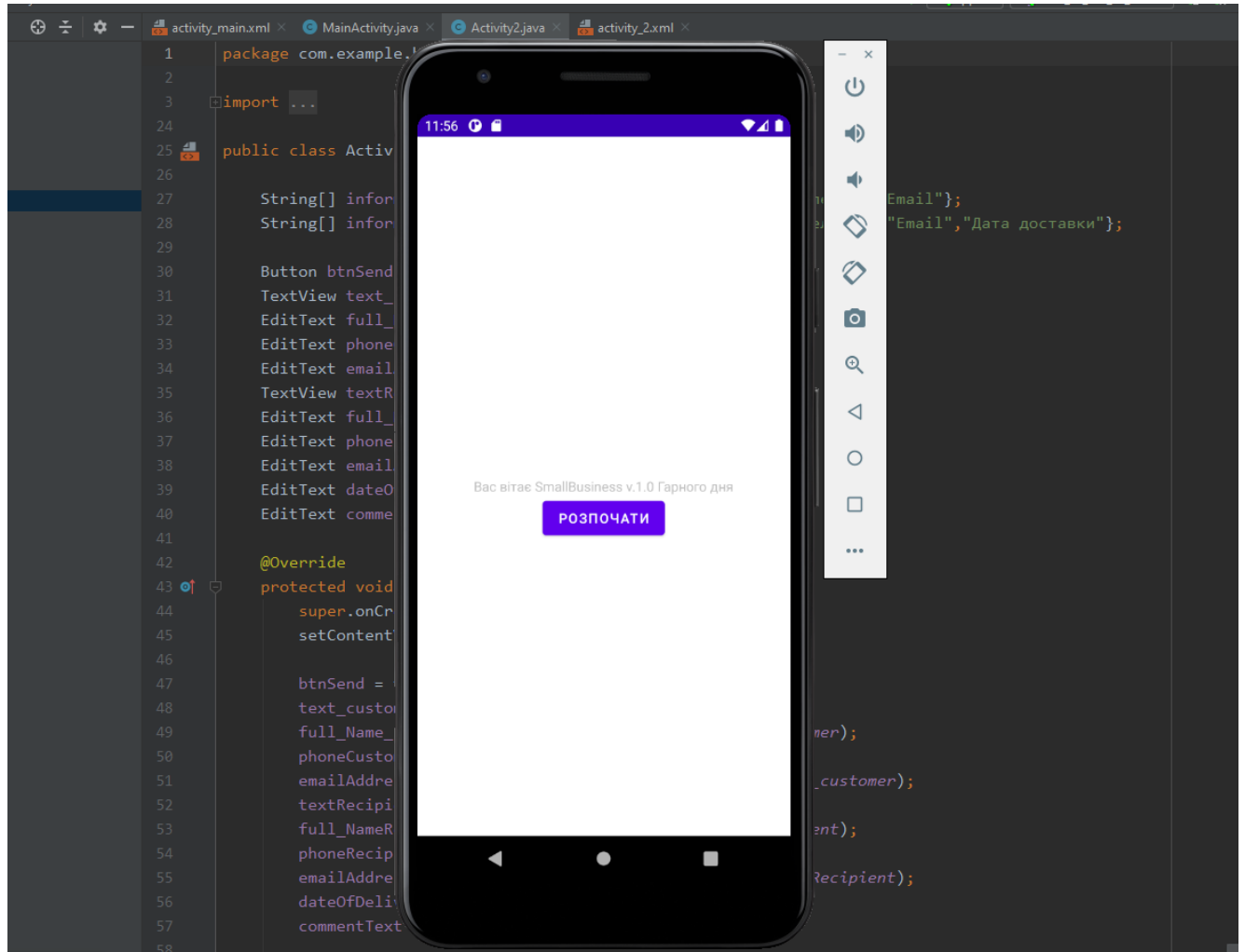


Рис. 3.5. Робота емулятора на головній сторінці

Для побудови дизайну в Android Studio використовується XML – являє собою розшифровану мову розмітки. Це набір кодів та тегів, що описують текст у цифровому документі. Найвідомішою мовою розмітки є мова розмітки гіпертексту (HTML), яка використовується для форматування веб-сторінок. XML, більш гнучкий двоюрідний брат HTML, дозволяє вести складний бізнес через Інтернет. У той час як HTML повідомляє додатку браузера, як повинен виглядати документ, XML описує, що в документі. Іншими словами, XML стосується того,

як організована інформація, а не як вона відображається. Форматування XML здійснюється за допомогою окремих таблиць стилів.

Гнучкість XML має багато переваг. Це дозволяє передавати дані між корпоративними базами даних та веб-сайтами, не втрачаючи при цьому важливу описову інформацію. Це дозволяє автоматично налаштовувати подання даних, а не відображати одну і ту ж сторінку для всіх бажаючих. І це робить пошук більш ефективним, оскільки пошукові системи можуть сортувати точні теги, а не довгі сторінки тексту.

Оскільки XML привносить на веб-сайти складне кодування даних, це допомагає компаніям інтегрувати свої інформаційні потоки. Створюючи єдиний набір тегів XML для всіх корпоративних даних, інформація може безперешкодно обмінюватися веб-сайтами, базами даних та іншими внутрішніми системами. Але революційна сила XML полягає в підтримці транзакцій між бізнесом. Коли компанія продає товар або послугу іншій компанії, необхідно обмінятися великою кількістю інформації - про ціни, терміни, технічні характеристики, графіки доставки тощо. Універсальний характер HTML робить такі обміни важкими, а то й неможливими через Інтернет. За допомогою XML вся необхідна інформація може обмінюватися в електронній формі, що дозволяє укласти складні угоди без будь-якого втручання людини. Форма даних замовлення представлена на рисунку. (Рис 3.6.).

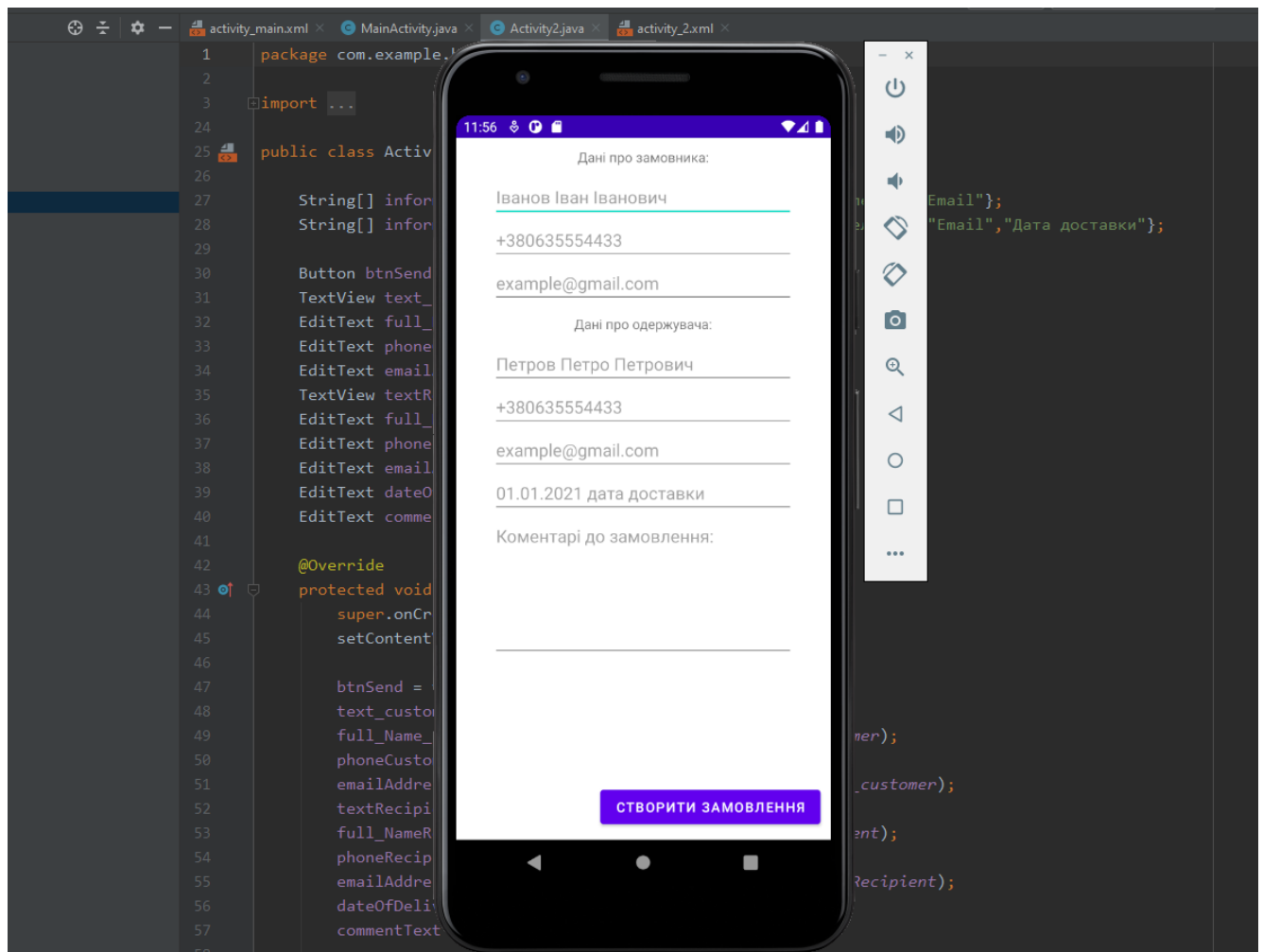


Рис. 3.6 Сторінка вводу даних замовлення

Загальна структура мобільного додатку побудована на `RelativeLayout` представляє об'єкт `ViewGroup`, який має в своєму розпорядженні дочірні елементи щодо позиції інших дочірніх елементів розмітки або щодо області самої розмітки `RelativeLayout`. Використовуючи відносне позиціонування, ми можемо встановити елемент по правому краю або в центрі або іншим способом, який надає даний контейнер. Для установки елемента в файлі XML ми можемо застосовувати різні атрибути по розміщенню в різних місцях окремих блоків. (Рис 3.7.).


```

1  |<?xml version="1.0" encoding="utf-8"?>
2  |<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
3  |   xmlns:app="http://schemas.android.com/apk/res-auto"
4  |   xmlns:tools="http://schemas.android.com/tools"
5  |   android:layout_width="match_parent"
6  |   android:layout_height="match_parent"
7  |   tools:context=".MainActivity">
8  |
9  |
10 |   <TextView
11 |       android:id="@+id/text_customer"
12 |       android:layout_width="wrap_content"
13 |       android:layout_height="wrap_content"
14 |       android:layout_margin="10dp"
15 |       android:layout_centerHorizontal="true"
16 |       android:text="Дані про замовника:"
17 |
18 |   />
19 |   <EditText
20 |       android:id="@+id/full_Name_customer"
21 |       android:layout_below="@+id/text_customer"
22 |       android:hint="Іванов Іван Іванович"
23 |       android:layout_centerHorizontal="true"
24 |       android:layout_width="wrap_content"
25 |       android:layout_height="wrap_content"
26 |       android:ems="15"
27 |       android:maxLength="35"
28 |
29 |   />
30 |
31 |   <EditText
32 |       android:id="@+id/phoneCustomer"
33 |       android:layout_below="@+id/full_Name_customer"
34 |       android:hint="+380635554433"
35 |       android:layout_centerHorizontal="true"
36 |       android:layout_width="wrap_content"
37 |       android:layout_height="wrap_content"
38 |
39 |   />
40 |
41 | </RelativeLayout>

```

Рис. 3.7. Структура XML документу

В XML документі використані такі компоненти, як кнопки, поля для вводу тексту, поля для відображення тексту, Button, EditText, TextView кожен із яких налаштовується по різним параметрам. Всі вони мають свій ID, налаштування по розмірам, по місцезнаходженню, та відступах від сусідніх об'єктів, наприклад: layout_centerHorizontal, layout_width, layout_height.

3.3. Аналіз потреб щодо функціональних можливостей програмного забезпечення та подальша програмна реалізація

Вимоги до програмного забезпечення - це опис особливостей та функціональних можливостей цільової системи. Вимоги передають очікування користувачів від програмного продукту. Вимоги можуть бути очевидними або прихованими, відомими чи невідомими, очікуваними чи несподіваними з точки зору клієнта. Процес збору вимог до програмного забезпечення від клієнта, їх аналізу та документування відомий як розробка вимог. Метою розробки вимог є розробка та підтримка складного та описового документа «Специфікація системних вимог».

Інженерний процес вимог, це процес в чотири етапи , який включає:

- Техніко-економічне обґрунтування
- Збір вимог
- Специфікація вимог до програмного забезпечення
- Перевірка вимог до програмного забезпечення

Техніко-економічне обґрунтування. Коли клієнт звертається до організації з метою отримання бажаного продукту, він приходить з приблизним уявленням про те, які всі функції повинно виконувати програмне забезпечення та які функції очікуються від програмного забезпечення. Посилаючись на цю інформацію, аналітики детально вивчають, чи можливо розробити бажану систему та її функціональність. Це техніко-економічне обґрунтування спрямоване на досягнення мети організації. У цьому дослідженні проаналізовано, чи можна програмний продукт практично матеріалізувати з точки зору впровадження, внеску проекту в організацію, обмежень витрат та відповідно до цінностей та цілей організації. Він досліджує технічні аспекти проекту та продукту, такі як зручність використання, ремонтпридатність, продуктивність та можливості інтеграції. Результатом цього етапу повинен бути звіт з техніко-економічного обґрунтування, який повинен містити адекватні коментарі та рекомендації керівництву щодо того, чи слід здійснювати проект.

Збір вимог Якщо звіт про техніко-економічне обґрунтування позитивний щодо реалізації проекту, наступний етап починається зі збору вимог від користувача. Аналітики та інженери спілкуються з клієнтом та кінцевими користувачами, щоб знати їх ідеї щодо того, що програмне забезпечення має надавати та які функції вони хочуть включати. Збір вимог - Розробники обговорюють з клієнтом та кінцевими користувачами та знають їхні очікування від програмного забезпечення. Вимоги до організації - Розробники розставляють пріоритети та упорядковують вимоги в порядку важливості, терміновості та зручності. Переговори та обговорення - якщо вимоги неоднозначні або існують певні суперечності у вимогах різних зацікавлених сторін, якщо вони є, тоді це обговорюється та обговорюється із зацікавленими сторонами. Потім вимоги можуть бути пріоритетними та обґрунтовано скомпрометованими. Вимоги надходять від різних зацікавлених сторін. Щоб усунути двозначність та конфлікти, їх обговорюють для ясності та правильності. Нереальні вимоги порушуються розумно.

Документація – це всі офіційні та неофіційні, функціональні та нефункціональні вимоги задокументовані та надані для обробки на наступному етапі.

Методи виявлення вимог. Вимога – це процес з'ясування потреб до передбачуваної програмної системи шляхом спілкування з клієнтом, кінцевими користувачами, користувачами системи та іншими, хто бере участь у розробці програмної системи. Існують різні способи виявити вимоги. Інтерв'ю є потужним засобом для збору вимог. Організація може проводити кілька типів співбесід, таких як: Структуровані (закриті) інтерв'ю, де кожна інформація, яку потрібно зібрати, вирішується заздалегідь, вони чітко дотримуються шаблону та предмету обговорення. Неструктуровані (відкриті) співбесіди, де інформацію для збору не вирішують заздалегідь, більш гнучкі та менш упереджені. Усні інтерв'ю, письмові інтерв'ю індивідуальні співбесіди, які проводяться між двома особами за столом. Групові інтерв'ю, які проводяться між групами учасників. Вони допомагають виявити будь-які відсутні вимоги, оскільки до цього причетні численні люди.

Опитування. Організація може проводити опитування серед різних зацікавлених сторін, ставлячи запитання щодо їх очікувань та вимог від майбутньої системи. Анкети. Документ із заздалегідь визначеним набором об'єктивних питань та відповідними варіантами передається всім зацікавленим сторонам для відповіді, які збираються та складаються. Недоліком цієї методики є те, що якщо варіант якоїсь проблеми не згадується в анкеті, проблема може залишитися без уваги. Аналіз завдань Команда інженерів та розробників може проаналізувати роботу, для якої потрібна нова система. Якщо у клієнта вже є деяке програмне забезпечення для виконання певної операції, воно вивчається та збираються вимоги запропонованої системи. Мозковий штурм Проводиться неофіційна дискусія між різними зацікавленими сторонами, і всі їх вклади реєструються для подальшого аналізу вимог. Прототипування створює користувальницький інтерфейс без додавання детальних функцій для інтерпретації користувачем особливостей передбачуваного програмного продукту. Це допомагає дати краще уявлення про вимоги. Якщо в кінці клієнта не встановлено програмне забезпечення для довідки розробника, і клієнт не знає про власні вимоги, розробник створює прототип на основі спочатку згаданих вимог. Прототип демонструється клієнтові та зазначаються відгуки. Відгуки клієнтів служать вхідними даними для збору вимог. Спостереження. Команда експертів відвідує організацію або робоче місце клієнта. Вони спостерігають за фактичною роботою існуючих встановлених систем. Вони спостерігають за робочим процесом в кінці клієнта та за тим, як вирішуються проблеми з виконанням. Команда сама робить деякі висновки, які допомагають сформулювати вимоги, які очікуються від програмного забезпечення.

Характеристики щодо вимог до програмного забезпечення. Збір вимог до програмного забезпечення є основою всього проекту з розробки програмного забезпечення. Отже, вони повинні бути чіткими, правильними та чітко визначеними.

В повній мірі специфікації в до ПЗ мають бути:

Ясні

Правильні
Послідовні
Зрозумілі
Модифіковані
Перевірені
Пріоритетні
Однозначні
Відслідковуванні
Мати достовірне джерело

Ми повинні спробувати зрозуміти, які вимоги можуть виникнути на початковому етапі формування вимог і які вимоги очікуються від програмної системи. Загалом вимоги до програмного забезпечення повинні бути розділені на декілька категорій. Функціональні вимоги. До цієї категорії належать вимоги, що стосуються функціонального аспекту ПЗ. Вони визначають функції та функціональні можливості всередині та із програмної системи. Приклади - Опція пошуку, надана користувачеві для пошуку за різними рахунками-фактурами. Користувач повинен мати можливість надіслати будь-який звіт керівництву. Користувачів можна розділити на групи, а групам можна надати окремі права. Повинен відповідати діловим правилам та адміністративним функціям. Програмне забезпечення розробляється, зберігаючи цілісність сумісності донизу. Нефункціональні вимоги – вимоги, які не пов'язані з функціональним аспектом програмного забезпечення, належать до цієї категорії. Вони є неявними або очікуваними характеристиками програмного забезпечення, які користувачі припускають.

До нефункціональних вимог належать - Безпека Лісозаготівля Зберігання Конфігурація Продуктивність Вартість Сумісність Гнучкість Аварійного відновлення Доступність Вимоги класифікуються логічно як обов'язкові – без них програмне забезпечення не можна вважати функціонуючим. Повинні мати: Покращення функціональних можливостей програмного забезпечення. Може

мати: Програмне забезпечення все ще може нормально функціонувати відповідно до цих вимог.

Розробка програмного забезпечення для автоматизованої системи управління логістичних замовлень мовою JAVA містить в собі ідею полегшити невеликим компаніям, малим підприємствам, та новому лише щойно відкритому бізнесу вирішувати питання щодо формування логістичних замовлень. Із всієї логістичної ланки виокремлено основну функцію ПЗ яка відповідає за оформлення замовлення, формування бланку на відправку з даними про відправника та одержувача, що полегшує процес систематизації, та відслідковування замовлень. Готовий бланк представлений на Рис. 3.8. що на даний момент версії програми 1.0 являється кінцевим етапом використання мобільного додатку.

Замовлення №1

Інформація про замовника

ПІБ	Зозуля Владислав Васильович
Телефон	+380638333221
Email	bi.zozulia@gmail.com

Інформація про одержувача

ПІБ	Іванов Іван Іванович
Телефон	+380505005050
Email	ivanov.i@gmail.com
Дата доставки	15.06.2021

Коментарі: Самовивіз з магазину.

Підпис одержувача _____

Рис. 3.8. Готовий бланк замовлення створений програмою

Автоматично він зберігається в пам'яті телефону та готовий до друку, якщо потрібен підпис одержувача, або може знаходитись в пам'яті пристрою та сортуватись основними параметрами ОС Android.

Дана програма має великий потенціал в розробці, після якого може бути доданий набагато більший функціонал, в наступних версіях, такий як збереження бланків замовлення в хмарних базах даних, більша кількість макетів, можливість кастомізувати бланк, додаючи власних брендівих логотипів, провести інтеграцію з соціальними мережами, або месенджерами з використанням API.

3.4 Висновки до розділу 3

Отже, в даному розділі було розглянуто основні підходи до розробки програмного забезпечення. Було обрано послідовну модель, що включає в себе багато етапів, від ідеї, створення графічної частини, та розробки функціоналу програми до її тестування та виправлення помилок.

В розділі про розробку дизайну було описано та наглядно показано як відбувалось налагодження графічної частини мобільного додатку, згідно якої логіки та яким чином групувались елементи управління. Показано можливості середовища розробки Android Studio та вбудованого емулятора, при використанні котрого можливо в реальному часі відслідковувати зміни в роботі програми.

Проаналізовано загальні вимоги та потреби до функцій, які мають бути в даній розробці, розписано загальноприйняті положення щодо розробки ПЗ, що має входити в будь-яку розроблену програму. Прийнято рішення про концепцію мінімалізму, головний екран з привітанням та версією встановленою на мобільному приладі, смартфон, планшет тощо, під управлінням ОС Android. Наступним кроком описано функції, які можливі в другому вікні, основному, з можливістю вводу даних та оформленням логістичного замовлення. Описано формування документу в форматі PDF, який і являє собою бланк логістичного замовлення, готовий до використання на самому девайсі або для друку на принтері з подальшою передачею по логістичній ланці організації.

ВИСНОВКИ

У роботі проведено аналіз та дослідження щодо шляхів спрощення праці в сфері автоматизації логістичних операцій для підвищення ефективності та надійності, кращого функціонування у відповідності до встановлених вимог компаній та підприємств.

У ході розв'язання поставлених задач були отримані наступні результати:

1) Досліджено ринок логістичних послуг, проведено аналіз потреб потенційних користувачів, представників бізнесу, компаній та організацій, в використанні програмного забезпечення по автоматизації.

2) Проаналізовано особливості ОС Android, IOS та інших, з акцентом на подальшу розробку мобільного додатку та виявленням цільової аудиторії, для можливості охоплення максимальної кількості користувачів.

3) Проведено огляд мов програмування, які можуть бути оптимальними для розробки ПЗ для мобільних пристроїв на базі ОС Android.

4) Розроблено програмне забезпечення для автоматизованої системи управління логістичних замовлень мовою JAVA. Програма готова до використання та завантаження, має можливість подальшого впровадження додаткового функціоналу в наступних версіях.

Отримані результати досліджень ринку логістичних послуг, аналіз мобільних операційних систем, огляд мов програмування можуть бути використані фахівцями, що починають свій шлях в розробці програмного забезпечення в даній сфері. Водночас, готовий програмний продукт може бути використаний потенційними користувачами, яким потрібно створювати логістичні замовлення з подальшою передачею їх в логістичну ланку.

ПЕРЕЛІК ПОСИЛАНЬ

1. Аникин Б. А. Логистика / Б.А. Аникин. – М.: Проспект, 2013. – 406 с.
2. Бауэрсокс Д. Логистика. Интегрированная цепь поставок/ Д. Бауэрсокс, Д Клосс. – Олимп-Бизнес, 2010. – 640 с.
3. Кальченко А. Г. Логістика: підручник / А.Г. Кальченко. – К.: КНЕУ, 2005. – 284 с.
4. Дибська В.В., Анікін Б.А. Логістика / В.В. Дибська, Б.А. Анікін. – М., 2012. – 367с.
5. Шилдт Г. Java 8. Полное руководство. 9-е изд.: пер. с англ. / Герберт Шилдт. – Москва : ООО "И.Д. Вильямс", 2015. – 1376 с.
6. Sierra K., Bates V., Head First Java (изучаем Java) : пер. с англ. / Kathy Sierra, Bert Bates. – Москва : «Эксмо», 2012. – 718 с.
7. Философия Java. Библиотека ппрограмиста. 4-е изд. : пер. с англ. / Брюс Еккель : «Питер», 2009. – 640 с.
8. Java 2. Библиотека профессионала, том 1. Основы. 8-е издание : пер. с англ. / Хорстманн Кей С., Корнелл Гари. – Москва : ООО "И.Д. Вильямс", 2012. – 816 с.
9. Основи програмування на Java : методичні вказівки до лабораторного практикуму та самостійної роботи, частина перша / Бивойно П. Г., Бивойно Т. П. – Чернігів: ЧНТУ, 2014. – 79 с.
10. Освоюємо Java. Вікіпідручник [Електронний ресурс]. – Режим доступу до ресурсу: https://uk.wikibooks.org/wiki/Освоюємо_Java.
11. Java. ProgLang. Самоучитель [Електронний ресурс]. – Режим доступу до ресурсу:<http://proglang.su/java>.
12. Java Course (рос.) [Електронний ресурс]. – Доступний з <http://javacourse.ru/begin/operations>.
13. Android Studio [Електронний ресурс]. – 2019 – Режим доступу до ресурсу: https://uk.wikipedia.org/wiki/Android_Studio.

14. Рик Роджерс. Android. Разработка приложений / Роджерс Рик. – М.: Эком, 2010. – 130 с.
15. Цехнер М. Программирование под Android / Марио Цехнер. – М.: Питер, 2012. – 688 с.
16. Дейтел П. Android для программистов. Создаем приложения / П.Дейтел – М.: Питер, 2012. – 560 с.
17. Документація Android [Электронний ресурс]. – Режим доступу до ресурсу: <https://developer.android.com>.
18. Android – Википедия [Электронний ресурс]. – Режим доступу до ресурсу: <https://ru.wikipedia.org/wiki/Android>.
19. Activity. [Электронний ресурс]. – Режим доступу до ресурсу: <http://metanit.com/java/android/2.1.php>.

ДЕМОНСТРАЦІЙНІ МАТЕРІАЛИ

ДЕРЖАВНИЙ УНІВЕРСИТЕТ ТЕЛЕКОМУНІКАЦІЙ
 НАВЧАЛЬНО-НАУКОВИЙ ІНСТИТУТ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ
 КАФЕДРА ІНЖЕНЕРІЇ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ



Бакалаврська робота

на тему

«РОЗРОБКА ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ ДЛЯ АВТОМАТИЗОВАНОЇ СИСТЕМИ УПРАВЛІННЯ ЛОГІСТИЧНИХ ЗАМОВЛЕНЬ МОВОЮ JAVA»



Виконав
 студент групи ППЗ-52
 Зозуля В.В.
 Керівник: Залива В.В.

АКТУАЛЬНІСТЬ РОБОТИ

Все частіше в різні сфери нашої діяльності приходить на допомогу електронний документообіг та автоматизація функцій, що раніше виконувалась вручну, та представляля собою рутинну справу. Уникнути проблеми можна завдяки використанню автоматизованої системи управління замовленнями мовою програмування JAVA, яка спрощує процес оформлення логістичних замовлень, збереження їх в зручній для користувача формі та передачі замовлення далі в логістичну ланку компанії.

НАУКОВІ ЗАВДАННЯ

Дослідити	Проаналізувати	Розробити
заотребуваність та актуальність даного ПЗ;	які особливості та функціональні можливості має містити дане ПЗ	мобільний додаток, який буде мати достатній функціонал для ефективної роботи підприємства по логістичним замовленням.

3

МЕТА РОБОТИ

ОБ'ЄКТ ДОСЛІДЖЕННЯ

ПРЕДМЕТ ДОСЛІДЖЕННЯ

розробити мобільний додаток котрий буде мати основний функціонал по створенню логістичних замовлень, тим самим підвищуючи ефективності та надійності їх функціонування у відповідності до встановлених стандартів підприємства.

розробка програмного забезпечення по автоматизації логістичних замовлень.

технологічні аспекти розробки програмного забезпечення для операційної системи Андроїд мовою програмування JAVA.

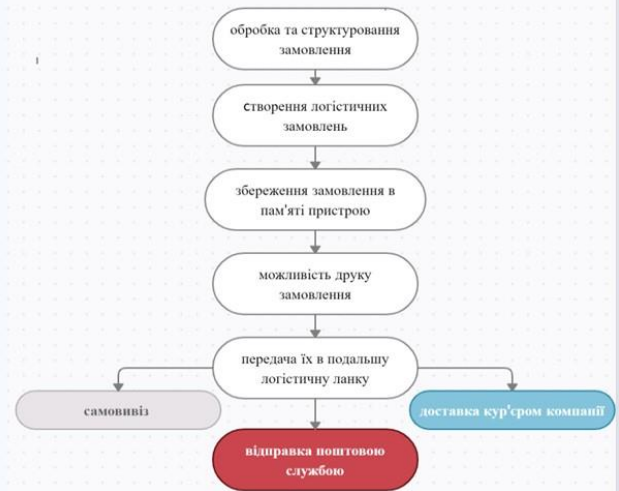
4

АНАЛІЗ РОЗРОБКИ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ АВТОМАТИЗАЦІЇ ЛОГІСТИЧНИХ ЗАМОВЛЕНЬ МОВОЮ JAVA

Саме програмне забезпечення автоматизації логістичних замовлень є досить важливим та покриває собою основні функції які потрібні для компаній початківців та малого бізнесу. Надає можливість більш гнучкою, швидко та ефективно проводити різні операції над запаленням, що в першу чергу відражається на рівні задоволення кінцевого споживача послуг.

ОСНОВНІ ФУНКЦІЇ:

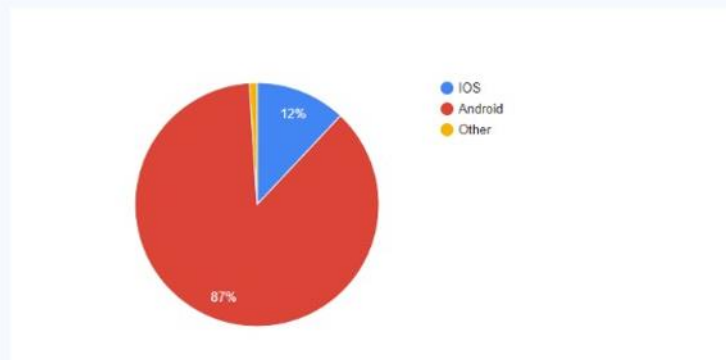
- Обробка та структурування замовлення
- Створення логістичних замовлень
- Збереження замовлення в пам'яті пристрою
- Можливість друку замовлення
- Передача їх в подальшу логістичну ланку



5

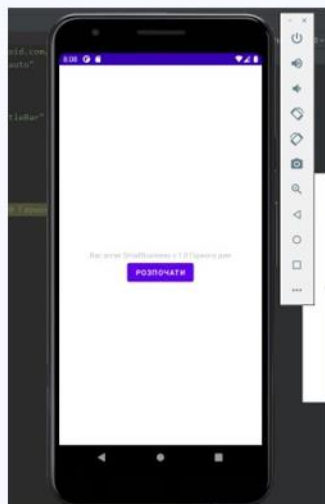
АНАЛІЗ РИНКУ МОБІЛЬНИХ ОС

На сьогоднішній день існує дві основні мобільні операційні системи, Android та IOS, також є багато менш популярних, що займають близько одного відсотку: Kai OS, Lineage OS, Fire OS, Flyme OS, Sailfish OS, Tizen, Remix OS.

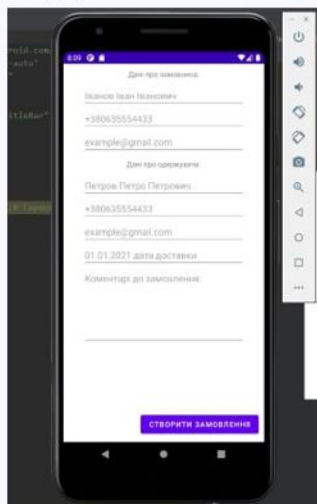


6

ВІЗУАЛЬНА СКЛАДОВА МОБІЛЬНОГО ДОДАТКУ



СТАРТОВИЙ ЕКРАН



ГОЛОВНИЙ ЕКРАН

Замовлення №1	
Інформація про замовника	
ПІБ	Зозуля Владислав Васильович
Телефон	+380638333221
Email	bi.zozulia@gmail.com
Інформація про одержувача	
ПІБ	Іванов Іван Іванович
Телефон	+380505005050
Email	ivanov.i@gmail.com
Дата доставки	15.06.2021
Коментарі: Самовивіз з магазину.	
Підпис одержувача _____	

СТВОРЕНИЙ PDF ДОКУМЕНТ

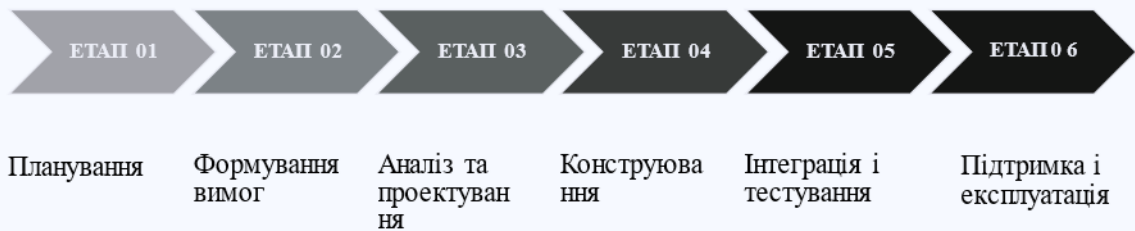
7

ОСНОВАНІ ПРОБЛЕМИ В ЛОГІСТИЧНІЙ ГАЛУЗІ

ПРОБЛЕМА	РІШЕННЯ
Висока вартість ПЗ	Безкоштовний мобільний додаток
Затрати на велику кількість офісної техніки	Смартфон під управлінням ОС Android
Відсутність чіткої структури	Чітко структурований PDF документ
Втрата даних замовлень та файлів	Замовлення зберігаються на пристрої де встановлена програма

8

ПРОЙДЕНІ ЕТАПИ



9

ВИСНОВКИ

У роботі проведено аналіз та дослідження щодо шляхів спрощення праці в сфері автоматизації логістичних операцій для підвищення ефективності та надійності, кращого функціонування у відповідності до встановлених вимог компаній та підприємств.

У ході розв'язання поставлених задач були отримані наступні результати:

Досліджено ринок логістичних послуг, проведено аналіз потреб потенційних користувачів, представників бізнесу, компаній та організацій, в використанні програмного забезпечення по автоматизації.

Проаналізовано особливості ОС Android, IOS та інших, з акцентом на подальшу розробку мобільного додатку та виявленням цільової аудиторії, для можливості охоплення максимальної кількості користувачів.

Проведено огляд мов програмування, які можуть бути оптимальними для розробки ПЗ для мобільних пристроїв на базі ОС Android.

Розроблено програмне забезпечення для автоматизованої системи управління логістичних замовлень мовою JAVA. Програма готова до використання та завантаження, має можливість подальшого впровадження додаткового функціоналу в наступних версіях.

10

ДЯКУЮ ЗА УВАГУ

