

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Національний університет «Запорізька політехніка»

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

Кафедра комп'ютерних систем та мереж
(повне найменування кафедри)

Пояснювальна записка

до дипломного проекту (роботи)

магістра

(ступінь вищої освіти (освітній ступінь))

на тему МОБІЛЬНА СИСТЕМА СКЛАДСЬКОГО ОБЛІКУ
ТОВ «APCOR»

Виконав: студент 6 курсу, групи КНТз-511м
спеціальності _____

123 «Комп'ютерна інженерія»

(код і найменування спеціальності)

Освітня програма (спеціалізація) _____

«Комп'ютерні системи та мережі»

Авер'янов І.С.

(прізвище та ініціали)

Керівник Кудерметов Р.К.

(прізвище та ініціали)

Рецензент Єфименко М.В.

(прізвище та ініціали)

м. Запоріжжя
2022 рік

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

Інститут, факультет інформатики та радіоелектроніки, комп'ютерних наук і технологій
 Кафедра «Комп'ютерні системи та мережі»
 Ступінь вищої освіти (освітній ступінь) магістерський
 Спеціальність 123 Комп'ютерна інженерія
 (код і найменування)
 Освітня програма (спеціалізація) Комп'ютерні системи та мережі
 (назва освітньої програми (спеціалізації))

ЗАТВЕРДЖУЮ

Завідувач кафедри Кудерметов Р.К.

“ ” 2022 року

З А В Д А Н Н Я
НА ДИПЛОМНИЙ ПРОЕКТ (РОБОТУ) СТУДЕНТА

Авер'янова Івана Сергійовича

(прізвище, ім'я, по батькові)

1. Тема проекту (роботи) Мобільна система складського обліку ТОВ «АрСог»

Керівник проекту (роботи) Кудерметов Р.К., к. т. н., доцент

(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

затверджені наказом вищого навчального закладу від 07 листопада 2022 року № 367

2. Строк подання студентом проекту (роботи) 10 грудня 2022 року

3. Вихідні дані до проекту (роботи) Необхідно розробити мобільну систему складського обліку, що містить у собі не менше 1 мобільного пристрою марки Apple та сервер. Мінімальні вимоги до серверу: процесор – 3 ГГц, жорсткий диск – 512 Гб, ОП – 4 Гб. Прикладне програмне забезпечення розробити на високорівневих мовах програмування Objective C та C++ , у середовищі XCode з використанням бібліотек IOS, SDK.

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

1) Аналіз способів організації складського обліку

2) Розробка мобільної системи складського обліку

3) Результати розробки системи

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

Плакати 1. Архітектура мобільної системи складського обліку ТОВ «АрСог»

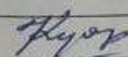
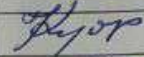
Плакати 2. Інтерфейси мобільної системи складського обліку ТОВ «АрСог»

Плакати 3. Інтерфейси мобільної системи складського обліку ТОВ «АрСог»

Плакати 4. Структура мобільної системи складського обліку ТОВ «АрСог»

Плакати 5. Блок-схема алгоритму додатку, який розроблюється

6. Консультанти розділів проекту (роботи)

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	прийняв виконане завдання
1-4	Кудерметов Р.К., зав. каф. КСМ		
Нормоконтроль	Щербак Н.В., ст. викл.		

7. Дата видачі завдання 30.09.2022 р.

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів дипломного проекту (роботи)	Строк виконання етапів проекту (роботи)	Примітка
1	Огляд та аналіз існуючих систем складського обліку	07.10.2022	
2	Аналіз вимог до мобільної системи	14.10.2022	
3	Вибір апаратного забезпечення мобільної системи	25.10.2022	
4	Проектування мобільної системи ТОВ «АрСог»	02.11.2022	
5	Вибір програмного забезпечення системи	09.11.2022	
6	Розробка прикладного програмного забезпечення	16.11.2022	
7	Розробка інтерфейсів програмного забезпечення	22.11.2022	
8	Налагодження роботи системи	27.11.2022	
9	Оформлення отриманих результатів у ПЗ	01.12.2022	
10	Оформлення графічного матеріалу	05.12.2022	
11	Оформлення допоміжного матеріалу	09.12.2022	

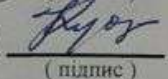
Студент


(підпис)

І.С. Авер'янов

(ініціали та прізвище)

Керівник проекту (роботи)


(підпис)

Р.К. Кудерметов

(ініціали та прізвище)

РЕФЕРАТ

МОБІЛЬНАЯ СИСТЕМА, OBJECTIVE-C, XCODE, ТРЬОХУРІВНЕВА
АРХІТЕКТУРА, MYSQL, PHPMYADMIN, ГРАФІЧНИЙ ІНТЕРФЕЙС

ПЗ: 58 с., 30 рис., 1 додаток, 15 посилань.

Об'єкт розробки – мобільна система складського обліку ТОВ «ApCor».

Мета цієї дипломної роботи – автоматизувати складський облік ТОВ «ApCor» на мобільній системі зі зручним інтуїтивно зрозумілим графічним інтерфейсом.

У першому розділі роботи проводитиметься аналіз існуючих систем складського обліку та формуються завдання до системи, що розробляється. Проводиться порівняння кількох мов програмування, таких як Java, C#, Objective-c, та середовищ розробки для iPhone, у результаті якого зроблено вибір на користь розробки власної мобільної системи складського обліку мовою Objective-C серед програмування XCode.

У другій частині диплома описано реалізацію мобільної системи системи складського обліку ТОВ «ApCor». В основі системи лежить трирівнева архітектура: база даних, серверна частина та клієнтська частина.

База даних програми написана мовою MySQL, для роботи з сервером MySQL через Інтернет використовувалася програма PhpMyAdmin. У пояснювальній записці наведено листинг запитів.

У другій частині також представлено алгоритм роботи програми; реалізовано поставлені завдання функціонування системи, код реалізації деяких функцій представлений у лістингах.

У третій частині детально представлений графічний інтерфейс користувача та посібник з використання програми.

ЗМІСТ

РЕФЕРАТ	4
ВСТУП	6
1 АНАЛІЗ СПОСОБІВ ОРГАНІЗАЦІЇ СКЛАДСЬКОГО ОБЛІКУ	7
1.1 Поняття системи управління складом	7
1.2 Використання існуючої системи складського обліку	11
1.2.1 Система EME.WMS	12
1.2.2 Система «IBS Trade & Warehouse»	13
1.2.3 Система керування складом Oracle.....	15
1.2.4 Мобільна система передачі даних.....	16
1.2.5 Система MobileSOP	19
1.3 Розробка системи складського обліку.....	22
1.3.1 Середовище розробки програм для iPhone	22
1.3.2 Мова програмування для розробки системи складського обліку.....	29
1.3.3 Засоби роботи з базою даних системи складського обліку.....	32
2 РОЗРОБКА МОБІЛЬНОЇ СИСТЕМИ СКЛАДСЬКОГО ОБЛІКУ	37
2.1 Структурна розробка мобільних додатків	37
2.2 Створення бази даних програми	38
2.2.1 Схема реляційної бази даних.....	43
2.2.2 Розробка запиту.....	45
2.3 Апаратне забезпечення системи	45
2.4 Функції системи.....	47
2.4.1 Реалізація системи	49
3 РЕЗУЛЬТАТИ РОЗРОБКИ СИСТЕМИ	49
3.1 Графічний інтерфейс користувача.....	49
3.2 Напрями удосконалення програми	54
ВИСНОВКИ	56
ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ	57
ДОДАТОК А	
ПЛАКАТ 1	
ПЛАКАТ 2	
ПЛАКАТ 3	
ПЛАКАТ 4	
ПЛАКАТ 5	

ВСТУП

Останнім часом обсяг інформації та обсяг транзакцій у всіх сферах людської діяльності, таких як економіка, фінанси, політика, психологія значно збільшилися. І процес накопичення, обробки та використання знань постійно прискорюється. Вчені стверджують, що кількість інформації подвоюється кожне десятиліття. У зв'язку з цим повинні використовуватися автоматизовані засоби, що дозволяють ефективно зберігати, обробляти та розповсюджувати накопичені дані [1, 2].

На даний момент актуальним є питання автоматизації складу. Програми автоматизації складу дозволяють як малим і середнім підприємствам, і великим компаніям максимально оптимізувати роботу складських комплексів. Тому обрані теми робіт є актуальними.

Мета роботи — автоматизувати складський облік ТОВ «ApCor» на мобільній системі із зручним інтуїтивно зрозумілим графічним інтерфейсом.

Для досягнення поставленої мети необхідно виконати такі завдання:

- провести аналіз методів організації складського обліку;
- розробити структуру мобільного додатка;
- створити базу даних програми;
- вибрати відповідне програмне та апаратне забезпечення;
- реалізувати мобільну систему складського обліку зі зручним інтуїтивно зрозумілим графічним інтерфейсом.

1 АНАЛІЗ СПОСОБІВ ОРГАНІЗАЦІЇ СКЛАДСЬКОГО ОБЛІКУ

1.1 Поняття системи управління складом

Архітектура автоматизованої інформаційної системи управління складом ґрунтується на трьох рівнях принципів. Згідно з ним, система складається з трьох компонентів: інтерфейсу, сервера бази даних та бізнес-логіки. Розглянемо кожен детальніше.

- інтерфейс – надає клієнтам можливість вводити, змінювати та видаляти дані, а також вимагати операції та підготовку звітів.

сервер бази даних, де дані зберігаються безпосередньо

- бізнес-логіка опрацьовує дані, запитані користувачем. Потім він повертає оброблені дані до бази даних і повідомляє користувача через екран клієнтської програми завершення запитаної обробки.

Основні функції системи управління складом:

- активне керування складом;
 - збільшити швидкість збирання продукції;
 - отримувати достовірну інформацію про місцезнаходження товару на складі;
 - ефективне керування продуктами з обмеженим терміном придатності.
- Отримайте інструменти для підвищення ефективності та розробки процесів обробки товарів на складі. Оптимізація використання складських площ [3].

Як правило, системи управління складом вирішують такі завдання:

- прийом товару в режимі реального часу з використанням бездротових терміналів або паперових носіїв;
- друк штрих-кодів;
- гнучка ідентифікація з або без замовлень на постачання або попередження про відвантаження, отримані від постачальників.
- прийняття відповідального зберігання
- перевірка придатності та виправлення даних.

Склад:

- автоматизовані склади чи склади під контролем персоналу;
- складські правила коригуються для максимального використання складських площ та/або продуктивності складських операцій.
- складні критерії побудови блоків зберігання;
- поставлено завдання на створення складу.
- організація зберігання невпакованих товарів від різних постачальників.

Автоматизуйте одноразове отримання та відвантаження товарів:

- перевантаження товарів, що надійшли до відправки замовнику.
- Перевезення продукції по складах.

Гнучке управління замовленнями та групами замовлень:

- складне угруповання замовлень;
- обробка та випуск замовлення групою з оптимізованими процесами та ресурсами;
- консолідація та дроблення партій продукції;
- ідентифікація продукту, що настроюється, по відвантажувальній і зворотній упаковці.

Поповнення:

- налаштувати параметри потреби у поповненні;
- поповнення неповних піддонів;
- спільне поповнення групи товарів на одній палеті.
- автоматичне створення та надсилання завдань на поповнення.
- коригування стратегій поповнення;
- різні варіанти поповнення (штуки, ящики, піддони).

Цілісність замовлення:

- автоматичне формування завдань на виконання замовлень та надсилання їх співробітникам.
- упаковка безпосередньо на піддони з урахуванням вимог ергономіки та габаритів продукції, ваги та інших параметрів.

- складання на конвеєрній стрічці;
- пакетна обробка товарів;
- підтримка відбору проб поштучно, ящиками та повними палетами;
- комплектація з використанням радіотерміналів або етикеток;
- пристрої, які йдуть голосовим командам, даним системою.
- упаковка;
- різні варіанти складання (індивідуальна, групова, комбінована);
- персоналізація замовлень при складанні.
- генерація та відстеження ідентифікаційних номерів відправлених контейнерів.

Завантажити:

- планування відвантажень товарів з урахуванням пріоритету.
- розстановка та комбінування товарів під час навантаження відповідно до порядку доставки.
- операції завантаження, перевірки та закриття, контрольовані бездротовими терміналами.
- підтвердження та завершення відвантажувальних операцій.
- визначення кар'єри (вибір);
- заяву про відповідність;
- оформлення супровідних документів.

Управління запасами:

- відстеження контейнерів;
- повний функціонал для роботи із важкими вантажами;
- гнучкість переміщення та коригування складських запасів;
- попередня часткова інвентаризація;
- повна інвентаризація із зазначенням ваги на вході та виході;
- контроль стану та отримання інформації про складські запаси в режимі реального часу;
- консолідація часток у всіх РЦ;

- локалізація інвентарю та конфігурації по складському регіону та зонування;

- відстеження атрибутів майна (партія, код, серійний номер);
- розраховуйте дати та відстежуйте умови реалізації товару;
- відстеження власників нерухомості;
- гнучка система переадресації, розбиття на партії та рухи інвентарю;
- гнучкі способи доставки LIFO, FIFO, FPFO, FEFO, BBD [4].

Управління кадровими операціями:

- автоматичне створення та відправлення завдань;
- отримувати;
- готель;
- рух запасів;
- розрахунок товарообігу;
- поповнення запасів;
- замовлення виконано;
- завантаження;
- перевезення;

План роботи розподільчого центру:

- заплануйте завдання, переставивши їх відповідно до пріоритету;
- диспетчеризація та ротація завдань;
- багато руху.

Управління контейнерами:

- застосувати ліцензійну/патентну інформацію;
- поміщення кількох різних товарів у контейнер;
- ідентифікація товару з упаковки при відвантаженні та поверненні;
- визначення обмежень на сумісне зберігання товарів.

Управління складськими та виробничими приміщеннями:

- визначення точного місця зберігання;
- оптимізація прогресивного сховища;
- автоматичне поповнення та передача на підсклади;

- рух усередині організації
- управління та оптимізація сховища відповідно до терміну придатності;
- управління та утилізація небезпечних вантажів;
- планування огляду та заправлення складської техніки.

Управління людськими ресурсами:

- розрахунок робочого часу;
- відстеження кадрових операцій;
- звітність із людських ресурсів;
- розробка нормативів трудових ресурсів;
- визначення планової продуктивності праці.

1.2 Використання існуючої системи складського обліку

Створення конкурентоспроможної, функціональної та простої у використанні системи складського обліку потребує аналізу існуючих можливостей.

Неможливо дати однозначну відповідь на питання, яка складська програма краща. Це з тим, що необхідно враховувати безліч чинників, як-от розмір компанії, її деталі тощо. Професійні компанії часто віддають перевагу індивідуальні програми стандартним програмам. Дорожче, але явно ефективніше.

Порівняння різних програм складського обліку може бути дуже складним як реальної оцінки швидкості, так зручності. У програму необхідно внести тисячі записів, а демо-версія програми, як правило, виключає таку можливість.

Можна вибрати деякі загальні положення щодо вибору програми.

Якщо вам потрібно придбати програму складського обліку, нагальних питань немає. Вихід є – купити найпростішу програму. Однак тут важливо не помилитись. Це пов'язано з тим, що дешеві та прості програми містять дуже обмежений функціонал. І велика ймовірність того, що ваші очікування від нової системи автоматизації не виправдаються. І було б непогано, якби це було очевидно, коли ви

почали його використовувати, а не тоді, коли ви вже запровадили тисячі даних. У цьому випадку ваша робота була витрачена даремно.

Програма проста, тому що вона проста, як калькулятор. Вона відповідає на кілька простих питань (наприклад, скільки продуктів є в наявності). Однак, якщо вам потрібні всілякі додаткові функції, такі як розрахований на багато користувачів режим, обмежені права користувача і можливість автоматичного резервного копіювання, ви, ймовірно, будете розчаровані [5].

На сучасному ринку програмних продуктів існує достатньо складських програм (програм складського обліку). Вони багато в чому схожі, але мають багато відмінностей [6]. Нині немає аналогічних додатків чи систем розробки, використовують звичайні мобільні телефони як терміналів. Тому я вирішив порівняти складну систему керування складом із простим робочим комп'ютером. Це допоможе вам зрозуміти функціональність та призначення цього програмного забезпечення та сформулювати уявлення про функціональність майбутніх додатків.

Для порівняння були обрані поширені системи EME.WMS, IBS Trade & Warehouse, Oracle Warehouse Management System, DataMobile, Мобільний склад.

1.2.1 Система EME.WMS

Програма EME.WMS – професійна система управління адресним складом. Розроблена для великих та середніх складів з активним переміщенням товарів, він успішно застосовується і на невеликих складах площею від 1200 до 3000 кв. метрів

Основні характеристики:

- отримання товару за допомогою сканера або паперу;
- твердження asn (попереднє повідомлення про відвантаження);
- прийом перекладів в автоматичному режимі з отриманням інформації зі складу відправника;
- розміщення товару безпосередньо під час прийому;
- друк штрих-кодів;
- автоматичне присвоєння адреси або визначення адреси вручну;
- розміщення товарів у осередках із суворою прив'язкою до відповідних кодів товарів чи груп розміщення товарів;

- пакетний розрахунок замовлень (бандл-розрахунок, запуск хвилі) для оптимізації процедури поповнення зон відбору;
- розрахунок замовлень за методами fefo, lefo;
- розрахунок товарів, замовлених із зазначеного місця;
- автоматичне формування завдань на поповнення вибраних зон на замовлення або на вимогу оператора;
- резервування товару;
- підтримка вибору частинами, блоками та коробками;
- вибір за допомогою терміналу збору даних;
- автоматичне поповнення зон комплектації;
- різні стратегії відбору (індивідуальний, груповий, комбінований);
- ABC/XYZ аналіз;
- опція «не брати, якщо є що колупати»;
- консолідація замовлень у транспортні засоби заздалегідь або під час відвантаження;
- підтвердження та завершення відвантажувальних операцій;
- підтвердження клієнтом отримання товару;
- визначення (підбір) перевізника, типу транспортного засобу та іншої транспортної інформації;
- транзитні перевезення з отриманням всіх даних про вступ в автоматичному режимі на інші склади від eme; wms (для трансфертних перевезень);
- підготовка супровідних документів, включаючи друк сертифікатів та сертифікатів якості [7].

1.2.2 Система «IBS Trade & Warehouse»

"IBS Trade & Warehouse" – спеціалізований програмний продукт на базі системи обліку та аналізу "IBS Enterprise Management".

Компанія «Інтелектуальні бізнес-системи» займається розробкою та впровадженням програмного забезпечення для обліку та аналізу, що відповідає останнім вимогам та стандартам систем управління підприємством. Основні особливості:

- гнучкий і настрюваний інтерфейс, що гарантує спрощення та зручність роботи.
 - довільна зміна розміру та розташування вікон документів та каталогів.
 - вимкніть вкладки документів і каталогів, які не використовуються.
- Фільтрування та сортування даних, що відображаються за будь-яким набором полів.
- увімкнути/вимкнути відображення стовпців табличних даних для документів та довідників.
 - довільна зміна ширини та порядку стовпців;
 - можливість виконувати ті ж дії за допомогою панелей інструментів, гарячих клавіш або контекстного меню.
 - використання майстрів для створення/копіювання документів та даних каталогів.
 - детальний довідник товарів та матеріалів з кількома рівнями класифікації.
 - детальна класифікація за принципом адміністративної території, на кшталт контрагента та її торгових точок (магазини, кіоски тощо.).
 - вести облік умов купівлі/постачання в рамках контракту або обмежень іншої сторони;
 - гнучка система ціноутворення та управління націнками (великий опт, опт, дрібний опт тощо).
 - призначайте менеджерів клієнтам та їх відділам з можливістю аналізу обсягу продажу, зобов'язань та прибутковості з погляду менеджера.
 - елементи транспортної логістики: формування транспортних маршрутів, що з'єднують клієнтів, автомобілі та водіїв.
 - повна інформація про залишки та рух товарів за період часу у будь-який момент з можливістю швидкої деталізації будь-якої цифри;
 - додаткові витрати на придбання шляхом розрахунку партій запасів, розрахунку закупівельних цін та розрахунку вартості кожної партії;
 - адреса складу та центральне сховище.

- аналіз потреб у продуктах та планування закупівель на основі статистичного аналізу продажів та товарних запасів.
- прийом замовлень від клієнтів, керування виконанням замовлень та аналіз кількісних та загальних відхилень, які можуть пояснити «дефекти».
- система резервування товару;
- управління дебіторською/кредиторською заборгованістю, простроченою заборгованістю;
- договірна бухгалтерія;
- повноцінний мультивалютний облік усіх бізнес-операцій.
- ведення різних взаєморозрахунків (передплата, післяплата, "продаж" тощо)
- платіжні системи з повним чи частковим закриттям документів, договорів;
- отримуйте складні звіти.
- інтеграція коїться з іншими популярними обліковими системами: 1С, Акцент та інших [8].

1.2.3 Система керування складом Oracle

Oracle Warehouse Management System – це спеціалізована система керування складом від Oracle, що забезпечує інформаційну та технічну підтримку складних процесів керування складом.

Oracle, найбільший виробник програмного забезпечення для електронного бізнесу, анонсувала Oracle Warehouse Management System – рішення для складів та розподільчих центрів. Цей новий додаток до E-Business Suite дозволяє пристроям введення штрих-кодів та мобільного зв'язку керувати складськими запасами та структурою складу, планувати завдання, підтримувати складні методи сортування та надавати додаткові послуги, призначені для надання. Oracle Warehouse Management System повністю інтегрована в Oracle E-Business Suite і постачається як частина готової до використання комплексної системи, що значно спрощує впровадження та підвищує загальну окупність інвестицій.

Oracle Warehouse Management System підтримує різні методи сортування, у тому числі цілеспрямований вибір та відхилення, автоматичне призначення завдань та маркування як у реальній виробничій системі. У поєднанні з потужним механізмом бізнес-правил ці методології полегшують адаптацію Oracle Warehouse Management System до індивідуальних вимог замовника. Стандартні функції, що забезпечують вбудовану підтримку штрих-кодів, використовують систему номерних знаків (LPN), а функції керування контейнерами значно знижують складність операцій та підвищують точність [9].

Основні можливості системи:

- керування бізнесом;
- організація праці;
- управління товарами у трейлерах/складах;
- графік надходження відвантажень;
- прийом та попередній розподіл з обробкою отриманої інформації;
- крос-докінг, перевалка та зберігання.
- повний функціонал попереднього повідомлення про відвантаження.
- управління запасами;
- зберігання, управління та профілювання одиниць складського обліку.
- додаткові послуги;
- розподіл та планування ресурсів
- управління завданнями [10].

1.2.4 Мобільна система передачі даних

Програмне забезпечення DataMobile автоматизує складські та роздрібні операції.

Програма DataMobile була створена для керування терміналами збору даних. На сьогоднішній день це зручний та доступний інструмент відстеження всіх товарних потоків для складів, магазинів, установ та організацій різного рівня та профілю.

Термінал збору даних – це спеціальний мобільний пристрій, здатний прочитувати інформацію штрих-коду. Вбудований сканер штрих-коду дозволяє

швидко збирати інформацію та фіксувати переміщення товарів за вашим складом або магазином. Це значно спрощує та прискорює процес обліку товарів на вашому підприємстві.

Програмне забезпечення DataMobile дозволяє швидко обробляти інформацію, отриману від терміналів збору даних, і надавати вам всю необхідну інформацію про ваш продукт. Програмне забезпечення DataMobile можна використовувати для вирішення різних завдань, таких як складський облік, перевезення, мерчандайзинг та управління виробничими процесами.

Мобільні термінали дозволяють вести точний та безпомилковий облік на складі. Термінали збору даних дозволяють значно прискорити весь процес прийому та доставки товарів на складі. Тому автоматизація складу використовується сьогодні великими та малими організаціями. Після зчитування штрих-коду інформація про кожен товар заноситься до загальної бази даних. Програми складського обліку не обмежуються записом інформації про надходження товару. Система автоматизації управління складом – Прискорений процес введення даних та підвищений рівень обробки та узагальнення інформації, що отримується. Автоматизація складу дозволяє правильно розподіляти будь-яку кількість товарів з урахуванням їх габаритів та ваги, а потім на основі цих даних раціонально формувати партії товарів для розвантаження. Програма складського обліку допомагає ефективно керувати такими процесами:

- прихід товарів (інформація про товари зчитується за штрих-кодом, програма складського обліку формує стратегію їх розміщення на складі);

- відпустка товару (на основі штрих-коду товару швидко формується серія відвантажень, розраховується загальна вага та складське становище кожної партії. Після завершення відвантаження автоматично коригується кількість складських позицій. Це забезпечує ведення складського обліку точно і без помилок);

- інвентаризація (Після зчитування штрих-коду мобільним терміналом збору даних всі товари, що зберігаються, швидко заносяться в базу даних. У найкоротші терміни створюється таблиця, що містить дані про наявність товарів на складі. Це

дозволяє вести все складський облік (ретельніше, скоротити час усіх складських операцій) та запобігти втратам складських товарів);

- швидке формування замовлень (точний облік усіх товарів збільшує швидкість виконання всіх замовлень);
- контролювати переміщення товарів за складом (періодично формуються звіти про рух товарів за складом).

Програмне забезпечення DataMobile має модульну структуру. Це дозволяє розширювати чи зменшувати функціональні можливості програми залежно від потреб конкретного користувача.

Базова версія програми DataMobile дозволяє:

- запис приходу, відправлення та руху товарів;
- створить інвентар;
- створює форму документа за заданим шаблоном;
- ідентифікувати товари та перевіряти відповідність маркуванню (функція перевірки цін).

Встановлення додаткових модулів дозволяє програмі виконувати такі функції:

- адресне зберігання складських товарів. Встановлений модуль зберігання адрес дозволяє відстежувати розміщення товарів по осередках;
- пакувальний лист. За допомогою цього модуля проводиться збирання товарів для відвантаження та/або зберігання;
- підбір за завданням. Перевіряє правильність оформлення та забирає товар за документами, завантаженими з бек-офісу;
- декілька штрих-кодів. Можливість включення кількох штрих-кодів на продукт, ідентифікація штрих-кодів контейнера/упаковки;
- серійний номер. Опис серійного номера при відвантаженні продукту та його наявності на складі;
- новий продукт. Додайте нові товари до номенклатури програми обліку товарів прямо з терміналу;

- мобільний друк. Можливість завантажувати шаблони етикеток на свій пристрій та роздруковувати їх на мобільному принтері.

Усі версії програми DataMobile мають такі особливості роботи, що підвищують зручність використання TDS.

- аутентифікація користувача;
- редактор шаблонів у терміналі;
- підтримка апаратних кнопок терміналу;
- звуковий супровід подій та повідомлень [11].

1.2.5 Система MobileSOP

MobileSOP: Мобільна комерція та автоматизація бізнесу – проєкт DigSee.

DigSee – компанія, яка надає корпоративні мобільні рішення. DigSee представляє групу топ-менеджерів та IT-фахівців.

Основним напрямком діяльності компанії є розробка різних мережевих корпоративних рішень, що розміщуються в настільних системах (настільних ПК та КПК) та мобільних пристроях. Саме два останні напрями розвитку системи сьогодні викликають найбільший інтерес у замовників. Тому DigSee приділяє особливу увагу розробці програм для КПК.

DigSee Inc. є партнером Microsoft Corp. Розробка програмного забезпечення для кишенькових комп'ютерів на платформі Pocket PC у рамках таких програм, як партнерська програма Microsoft Mobile Solutions та партнерська програма MS Windows Embedded.

DigSee просуває ринок інформаційних технологій за допомогою послуг доступу, зберігання та розповсюдження інформації в мобільних та веб-технологіях, корпоративних мережах, мобільних мережах та Інтернеті.

MobileSOP реалізує всі види автоматизації бізнесу та оптимізації бізнес-процесів на базі комунікаторів, комп'ютерів (ПК) та мобільних пристроїв. Мобільна торгівля та автоматизація роботи торгових агентів, сервісних інженерів, власників магазинів, страхових агентів, кур'єрів та іншого мобільного торгового персоналу. Автоматизуйте бізнес-моніторинг та віддалене управління бізнес-процесами торговими представниками на ходу.

Система автоматизації складу дозволяє кожному контрагенту вказати свої реквізити, спеціальні позначки (наприклад, не приймати товар, звернутися до товарознавця) і дозволяє вводити нових контрагентів, штрих-код або ряд символів (код, найменування). Це робиться за лічені секунди для категорії товарів. Як додаткові функції є функція прийому оплати та функція перегляду фото товару.

Програми для мобільного трейдингу дозволяють мобільному зв'язку автоматично передавати результати роботи співробітників прямо в офіс, в облікову систему клієнта швидко і недорого. Воно увійшло. Бізнес-процеси оптимізуються, торги автоматизуються – забираються «зайві» люди, такі як оператори. Торгові представники вводять дані безпосередньо у комунікатори та кишенькові комп'ютери, тому вони не потрібні. Існує також функція, яка дає вам повний контроль за вашими торговими представниками. Набір функцій автоматизації транзакцій, тобто автоматичне сканування та зчитування RFID-міток може однозначно ідентифікувати об'єкти, документи, торгових представників і навіть торгові точки. Управління бізнес-процесами, автоматизація бізнесу, щоб зробити його контрольованим та передбачуваним. Можливості друку на місці дозволяють роздруковувати рахунки-фактури, етикетки, звіти, заявки, замовлення на поставку та інші форми документів безпосередньо у клієнта.

Система автоматизації складу та складського обліку «Мобільний склад» пропонує можливість повної інтеграції рішення з існуючими офісними системами (такими як 1с), які працюють у клієнта. При цьому до офісної системи клієнта автоматично надходять усі виконані операції, пов'язані зі складським обліком, а також інша контрольна інформація від співробітників складу (час виконання операції, автоматичний розрахунок кількості тощо). Система автоматизації складського обліку «мобільний склад», аналогічна до системи автоматизації торгівлі, повністю управляється з офісної системи. Інакше кажучи, поточна офісна система визначає списки продуктів, упаковки, ціни тощо. Ці дані щодня автоматично передаються в систему автоматизації обліку «Мобільний склад» та можуть оновлюватись у будь-який час, щоб ви могли регулярно аналізувати продаж та відвантаження продукції.

Система автоматизації бізнесу та складського обліку «Склад у русі» має модульну структуру, що дозволяє вибудувати систему відповідно до необхідної бізнес-логіки та реалізує ефективність складського обліку та управління за рахунок оптимізації процесів складської логістики унікального для вашої компанії бізнесу. Простота та доступність інтерфейсу знижує витрати на навчання, забезпечує підвищення ефективності та швидкості виконання повсякденних операцій, спрощує управління складським персоналом.

Основні завдання системи автоматизації бізнесу MobileSOP (мобільної комерції) можна сформулювати так:

- повна автоматизація бізнесу;
- автоматизація бізнес-процесів;
- посилення циркуляції;
- управління людськими ресурсами;
- управління дистрибуцією;
- управління діяльністю торгових представників;
- удосконалене управління продуктом;
- автоматизація роботи торгових представників;
- оптимізація бізнес-процесів;
- скорочення етапів обробки замовлень;
- отримуйте аналітичну інформацію у центрі;
- ефективна мобільна торгівля;
- можливість централізовано збирати інформацію про всі замовлення, закупівлі, клієнтів, відмови, продаж та асортимент в одному місці;
- віддалений контроль товарів та маршрутів;
- підвищення операційної ефективності компанії та ефективності управління;
- збільшення інвестиційної вартості компанії [12].

1.3 Розробка системи складського обліку

1.3.1 Середовище розробки програм для iPhone

В даний час є кілька платформ для розробки під iOS. Розглянемо та проаналізуємо плюси та мінуси деяких відомих середовищ.

MonoTouch – це фреймворк, що дозволяє розробляти програми для iPhone та iPod Touch мовою C#. З MonoTouch ви можете писати програми на C#, компілювати для iPhone та використовувати бібліотеки .NET. Повний комплект ПЗ:

- Apple Mac OS X;
- встановлено MonoTouch для Mac OS X.
- моно розробка.

В даний час є кілька платформ для розробки під iOS. Розглянемо та проаналізуємо плюси та мінуси деяких відомих середовищ.

MonoTouch – це фреймворк, що дозволяє розробляти програми для iPhone та iPod Touch мовою C#.

З MonoTouch ви можете писати програми на C#, компілювати для iPhone та використовувати бібліотеки .NET.

Повний комплект ПЗ:

- Apple Mac OS X;
- встановлено MonoTouch для Mac OS X.
- моно розробка.

MonoTouch дозволяє розробникам створювати програми C# і .NET, компілювати їх та запускати на iPhone. У той же час програмісти можуть працювати у знайомому середовищі та писати знайомою мовою без необхідності вивчення Objective-C.

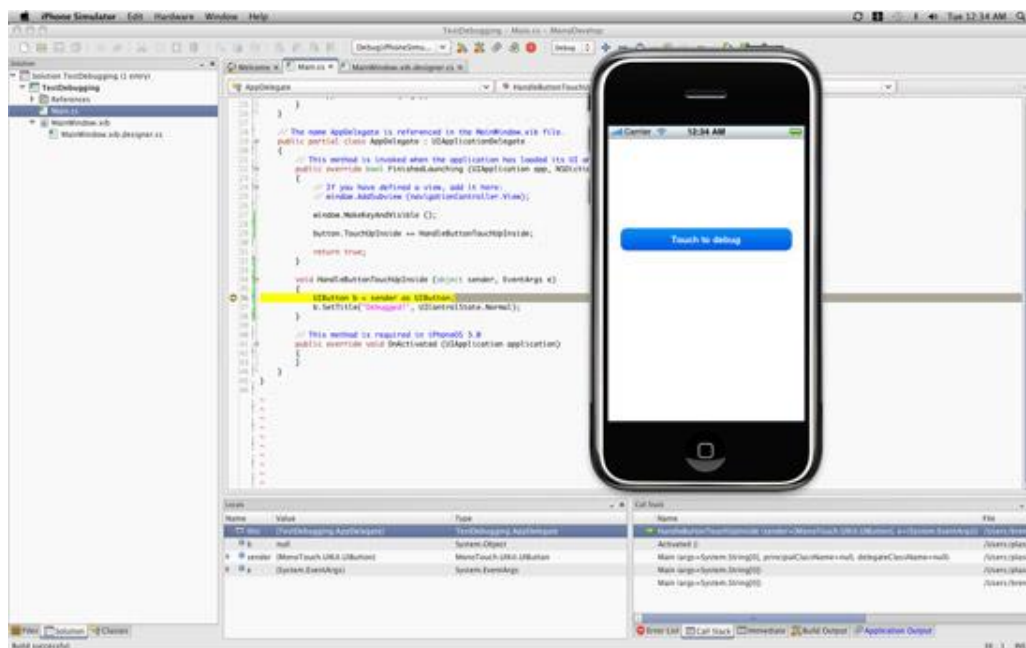


Рисунок 1.1 – Інтерфейс MonoTouch

Нові версії MonoTouch дозволяють налагоджувати програми на пристрої або в симуляторі, а налагодження програм на пристрої здійснюється через Wi-Fi. MonoTouch – платний продукт, але на офіційному сайті доступна пробна версія, яка працює з деякими обмеженнями (наприклад, у симуляторі можна лише налагоджувати програми).

Програми можна розробляти з використанням як Xcode, так і MonoDevelop, «рідного» середовища розробки Mono.

Однією з переваг MonoTouch є те, що програмістам, які знають C#, не потрібно вивчати нову мову для написання програм для iPhone.

До недоліків можна віднести високу вартість ліцензування (найпростіший коштує \$399) та наявність власного середовища розробки Apple.

Appcelerator Titanium – платформа для створення мобільних та десктопних кросплатформових додатків на JavaScript, HTML та CSS.

Ми підтримуємо використання HTML5, CSS3, JavaScript, Python, Ruby та PHP для розробки програм. Програми, створені на основі API Titanium, працюватимуть на всіх цільових платформах з універсальним рівнем кожної платформи. Платформа використовує стандартні системні виклики для створення вікон, системних меню, 2D/3D-анімації, відображення інформаційних повідомлень та доступу до файлової системи та обладнання. Titanium надає інтерфейс для

організації роботи з локальною базою даних (за допомогою SQLite) та дозволяє підключати додаткові розширення (плагіни). З точки зору користувача програма, створена на основі Titanium, нічим не відрізняється від нативної програми для конкретної програмної платформи.

Підготовка до програми Titanium розділена на чотири етапи.

- визначити логіку та зовнішній вигляд програми за допомогою html/css/javascript.
- спеціальний API, який дозволяє організувати доступ до функціоналу вашої цільової платформи.
- рівень, який не залежить від мови програмування, яка компілює веб-програми у власний код цільової платформи.
- оболонка, що дозволяє створювати пакети для кроссплатформного поширення вашої програми. Хмарні сервіси Appcelerator Network можна використовувати для пакування, тестування та розповсюдження програм.

Продукт позиціонується як альтернатива технології Adobe AIR, але, на відміну від AIR, розробники мають повний доступ до всіх системних ресурсів, а програми вимагають для роботи 1/10 частину пам'яті. Вихідний код Appcelerator Titanium доступний за ліцензією Apache і може вільно поширюватися в особистих та комерційних цілях. Крім того, доступний розширений комерційний пакет за ціною 199 доларів США на місяць на одного розробника. Комерційні пакети відрізняються наявністю послуг технічної підтримки, наданням аналітичних матеріалів, доступом до хмарних сервісів Appcelerator Network та участю у програмах тестування підтримки iPad та платформи Blackberry.

До недоліків цього продукту можна віднести те, що він зручний створення невеликих проєктів. Продуктивність швидко знижується зі зростанням програми. Це означає велике навантаження на ЦП, більше помилок та відхилень програми та більше енергоспоживання.

Phonegap – безкоштовна кроссплатформова бібліотека для мобільних платформ, що дозволяє відображати веб-сторінки з використанням HTML5, CSS3

та JavaScript. Це дозволяє запускати один і той же додаток на різних мобільних пристроях.

Помилково думати, що PhoneGap переводить код HTML, CSS та Javascript вашої сторінки у власний код вашої програми для конкретної платформи (Android, iOS, Windows Mobile тощо). Насправді рендеринг HTML+CSS здійснюється шляхом завантаження вмісту сторінки у спеціальний веб-подання. Цей webView також має розширений JavaScript (як правило, використовується WebKit) для обробки завдань, як звичайний браузер. Бібліотека складається з двох частин, які залежать одна від одної.

Нативна оболонка це оболонка функціональності конкретної платформи у вигляді набору класів. Інтерфейс класу бібліотеки однаковий усім платформам. В даний час оболонка Phonegap дозволяє працювати з наступними ресурсами та пристроями: компас, акселерометр, відеокамера, файлова система, підключення до мережі, геолокація, сповіщення, сховище (БД). Javascript Wrapper – оболонка для роботи з власними оболонками через Javascript. Фактично при виклику функції JavaScript-оболонки клієнтський сценарій викликає власну оболонку через функцію.

Програми, розроблені за допомогою PhoneGap, мають безліч переваг та недоліків. Почнемо з переваг.

- здійснювати спільну розробку програм дуже легко.
- код написаний на HTML, Javascript, CSS та може використовувати сторонні бібліотеки.
- кросплатформова підтримка;
- налагодження програм за допомогою браузера.
- можливість конвертувати вашу програму на будь-яку платформу за допомогою PhoneGap.

Недоліки розробки з PhoneGap:

- програма не підтримує багатопоточність.
- проблемна реалізація для довгих списків (1000 +);
- не всі програми можна оформити як веб-програми.

– апаратну частину мобільного пристрою адресовано по-різному, що може призвести до несподіваних результатів.

– складність налаштування програмного середовища для програмування за допомогою PhoneGap.

Xcode – це набір інструментів, розроблений Apple для розробки програм для Mac OS X та iPhone OS. Оновлення можна завантажити безкоштовно з офіційного сайту підтримки.

Xcode тісно інтегрований із інфраструктурою Cocoa. Він створює продуктивну та просту у використанні середовище розробки. Він також використовується для розробки Apple Mac OS X. Цей набір інструментів включає:

- Xcode IDE (для кодування, створення та налагодження додатків);
- Interface Builder (для розробки інтерфейсу користувача);
- Інструменти для аналізу поведінки та продуктивності;
- безліч додаткових інструментів.

Xcode – це інтегроване середовище розробки, що надає всі інструменти для створення та управління проєктами та вихідними файлами iPhone, а також запуску та налагодження коду на симуляторі або пристрої iPhone.



Рисунок 1.2 – Вікно вітання Xcode

Щоб створити новий додаток для iPhone, нам потрібно створити новий проєкт у Xcode. Проєкт керує всією інформацією, пов'язаною із додатком. Сюди входять вихідні файли, параметри конфігурації та правила, що поєднують усі ці частини. Основою всіх проєктів Xcode є вікно проєкту. З цього вікна ви маєте швидкий доступ до всіх основних елементів програми. Список «Групи та файли» – це область, в якій можна керувати файлами вашого проєкту, включаючи вихідні файли та об'єкти, створені з вихідних файлів. Панелі інструментів забезпечують доступ до часто використовуваних інструментів та команд. Інші аспекти вікна проєкту надають контекстну інформацію про ваш проєкт.

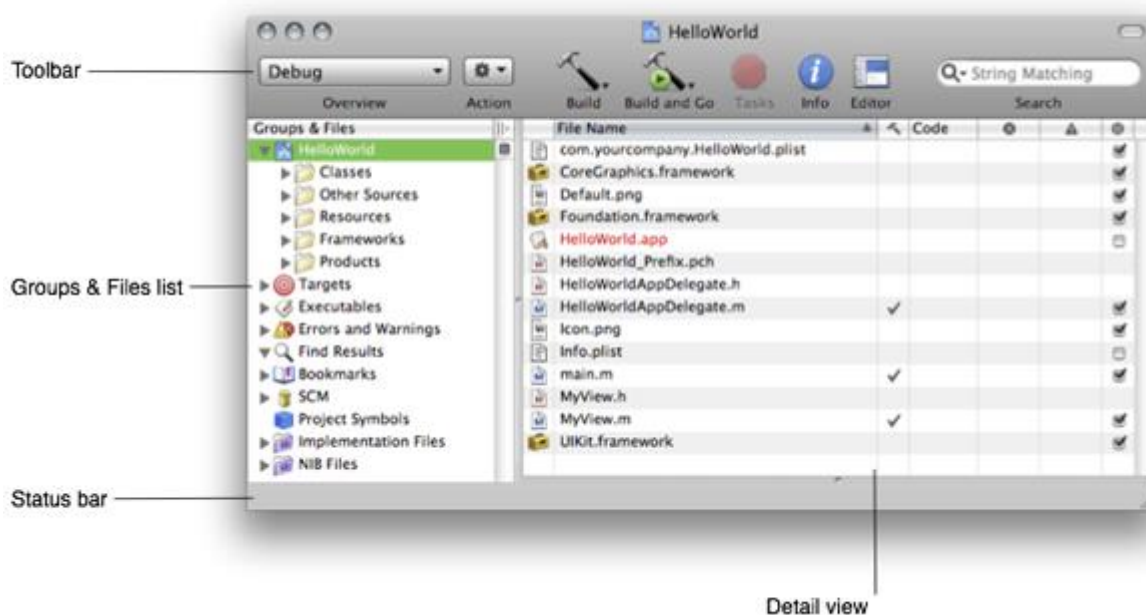


Рисунок 1.3 – Вікно налаштувань

Xcode поставляється з розширеним текстовим редактором, який підтримує такі функції, як завершення коду, виділення синтаксису, згортання коду (тимчасове приховування груп коду) та активні анотації помилок, попереджень та приміток. Я тут. Система розробки Xcode надає розумні значення за промовчанням та можливість налаштовувати ваше середовище та повідомляти про ваш контент. Якщо вам потрібна допомога, помічник надає контекстно-залежну документацію, дозволяючи вам переглядати та шукати інформацію у вікні документації Xcode.

При розробці програми в Xcode ви можете вибрати, як скомпілювати вашу програму: для емулятора iPhone або для пристроїв. Симулятор створює середовище, в якому ви тестуєте свою програму. Як тільки ви будете задоволені основною поведінкою вашої програми, ви можете вказати Xcode встановити та запустити його безпосередньо на iPhone або iPod, підключеному до вашого комп'ютера. Виконання на пристрої забезпечує ідеальне середовище тестування, а Xcode дозволяє підключати вбудований налагоджувач до коду, що працює на вашому пристрої.

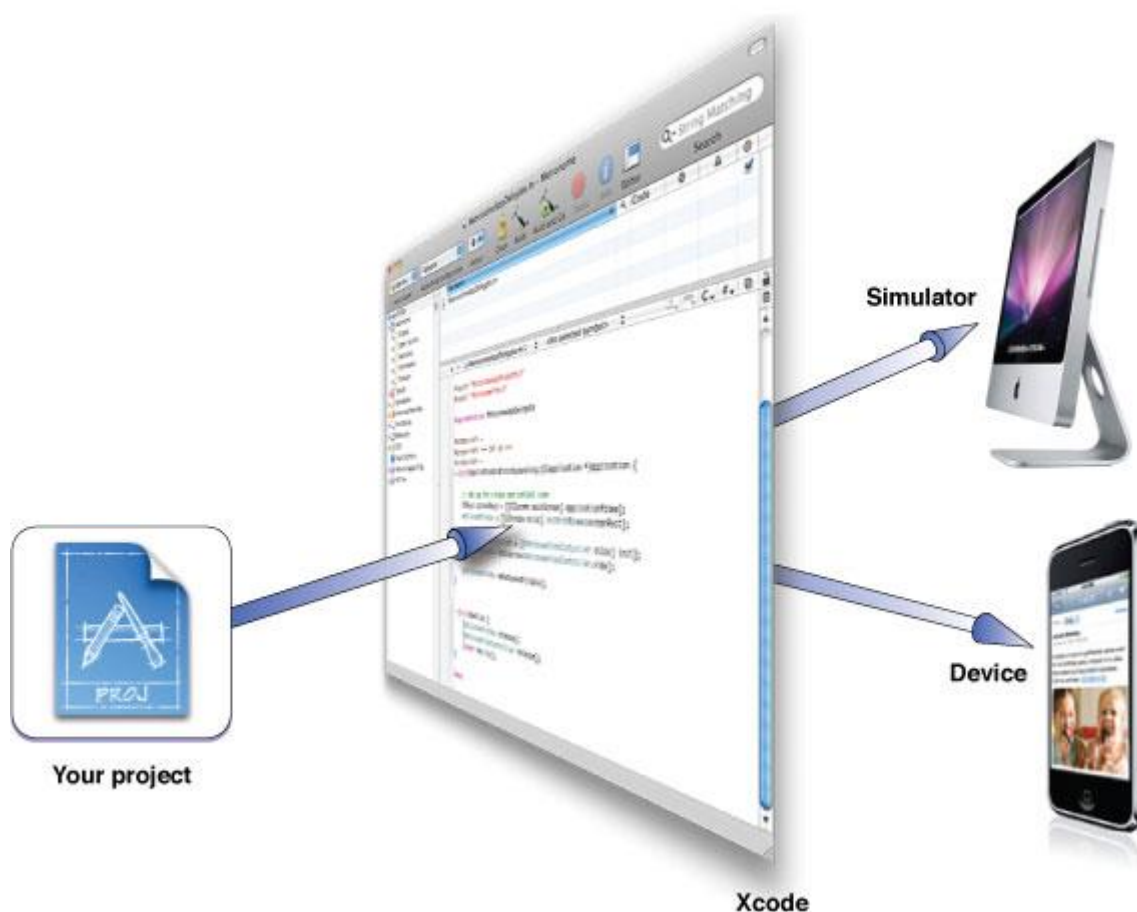


Рисунок 1.4 – Функціональна схема тестового середовища

Саме середовище Xcode використовується для розробки дипломної програми. Це тому, що це найбільш просунуте та відповідне середовище для розробки додатків для iOS, зі швидкою швидкістю компіляції додатків та зручним налагодженням коду.

1.3.2 Мова програмування для розробки системи складського обліку

У цьому розділі диплома розглядаються найпопулярніші мови програмування на даний момент, засновані на щорічному рейтингу TIOBE Software. До першої п'ятірки входять такі мови, як Java, C# і Objective-C.

Java – це мова програмування та обчислювальна платформа, вперше випущена компанією Sun Microsystems у 1995 році. Це базова технологія, на якій базується багато сучасних програм, включаючи службові програми, ігри та бізнес-програми. Java працює на понад 850 мільйонах персональних комп'ютерів і мільярдах пристроїв у всьому світі, включаючи мобільні телефони та телевізори.

Зараз Java розробляється Oracle. Є багато програм і веб-сайтів, які працюють, лише якщо у вас встановлено Java.

Java є основою для багатьох типів мережеских програм і стандартом для розробки та надання мобільних програм, ігор, веб-вмісту та корпоративного програмного забезпечення.

Програми Java, як правило, скомпільовані у власний байт-код, щоб вони могли працювати на будь-якій віртуальній машині Java (JVM), незалежно від архітектури комп'ютера.

Системи складського обліку часто реалізують клієнтську частину на цій мові.

C# (вимовляється як "she-sharp") є . Мова програмування, розроблена для розробки різних програм, призначених для роботи в середовищі .NET Framework. Мова C# проста, строго типізована та об'єктно-орієнтована. Завдяки багатьом інноваціям C# пропонує можливість швидкої розробки додатків, зберігаючи при цьому виразність і елегантність, властиві мові C.

C# належить до сімейства мов із C-подібним синтаксисом, який найбільш близький до C++ і Java. Мова є статично типізованою та анонімною з підтримкою поліморфізму, завантаження операторів (включно з явними та неявними операторами приведення типу), делегатів, атрибутів, подій, властивостей, загальних типів і методів, ітераторів, замикань Підтримує функції, LINQ, винятки та формальні коментарі . XML.

Objective-C зародився у 80-х як модифікація C для Smalltalk. Крім того, ця зміна полягала в додаванні нових синтаксичних конструкцій і спеціальних препроцесорів для них, а також бібліотеки часу виконання (ці виклики до блоку обробки). Тому Objective-C спочатку розглядався як доповнення до C. У певному сенсі це все ще так. Ви можете написати свою програму на чистому C і додати деякі конструкції Objective-C, якщо хочете. Або навпаки, вільно використовуйте C у програмах Objective-C. Більш того, все це стосується і програм на C++. У 1988 році NeXT (а пізніше Apple) ліцензували Objective-C і створили його компілятор і стандартну бібліотеку (по суті SDK). У 1992 році розробники проєкту GNU брали участь у розробці мови та компілятора в рамках проєкту OpenStep. Відтоді GCC підтримує Objective-C. Після придбання NeXT Apple взяла власний SDK (компілятор, бібліотеки, IDE) як основу для подальшого розвитку. IDE для коду називалася Xcode, а IDE для GUI – Interface Builder.

Файли модуля Objective-C мають розширення «.m» (якщо C++ і Objective-C змішані, розширення буде «.mm»). Файл заголовка – ".h". Усі об'єкти класу, створені в Objective-C, повинні бути розміщені в динамічній пам'яті. Тому тип `id`, який є вказівником на об'єкт будь-якого класу (загалом `void *`), представляє особливий інтерес. Нульовий покажчик називається нульовою константою. Таким чином, ви можете привести вказівник на будь-який клас до типу ідентифікатора. Використовуючи незмінний `isa`, ви можете знайти клас, до якого належить об'єкт типу `id`. Належить до зарезервованого типу класу. Об'єкти цього типу дозволяють вказувати імена ваших власних класів і базових класів, набір інваріантів класів, а також прототипи всіх методів, реалізованих цим об'єктом, і їх адреси (за допомогою локального списку селекторів). Усі зарезервовані слова Objective-C, які не є зарезервованими словами C, починаються із символу `@`. Для рядків Cocoa має дуже корисний клас `NSString`. Тип `BOOL` (по суті беззнаковий `char`) може приймати постійні значення `YES` і `NO`. Усі зарезервовані слова Objective-C (`objc/objc`, на відміну від мови C, знайдені в заголовку `h`) перераховані нижче.

`@interface` – починає оголошення класу або категорії (категорія є розширенням класу з додатковими методами без імітації).

`@implementation` – починається визначення класу або категорії.

`@protocol` – починає оголошення протоколу (аналогічно класам C++, що складаються з чистих віртуальних функцій).

`@end` – завершує оголошення\визначення будь-якого класу, категорії або протоколу.

`@private` – обмежує область застосування інваріантів класу методами класу (подібно до C++).

`@protected` є типовим - обмежує область інваріантів класу методами класу та методами похідних класів (подібно до C++).

`@public` – знімає обмеження області (подібно до C++).

`@try` – визначає блок, який може генерувати винятки (подібно до C++).

`@throw` – створює об'єкт винятку (подібно до C++).

`@catch()` – обробляє винятки, створені попередніми блоками `@try` (подібно до C++).

`@finally` – визначає блок після блоку `@try`, якому надається контроль незалежно від того, генерується виняткова ситуація чи ні.

`@class` – скорочення для оголошення класу (тільки назва (подібно до C++));

`@selector(method_name)` – повертає скомпільований селектор для імені методу `method_name`.

`@protocol(protocolname)` – повертає екземпляр класу протоколу під назвою `protocolname`.

`@encode(type_spec)` – Ініціалізація рядка, що використовується для шифрування даних типу `type_spec`.

`@synchronized()` – визначає блок коду, який виконується лише одним потоком у будь-який момент часу [14].

Тому, розглядаючи плюси і мінуси мов програмування для розробки мобільної системи складського обліку для ТОВ «АрСог», ми вибираємо Objective-C.

1.3.3 Засоби роботи з базою даних системи складського обліку

СУБД різних розробників слід розглядати як інструментарій для роботи з базою даних проєкту, що розробляється.

Мова структурованих запитів SQL – це універсальна комп’ютерна мова, яка використовується для створення, зміни та керування даними в реляційних базах даних. Всупереч поширеній думці, SQL є мовою інформаційної логіки, а не мовою програмування. SQL задумувався як інструмент кінцевого користувача, але з часом він став настільки складним, що став інструментом програміста. SQL заснований на реляційній алгебрі. Текст SQL-запиту, включаючи визначення даних і оператори обробки даних, можна легко перенести з однієї СУБД в іншу;

Об’єктно-реляційна СУБД Oracle. Єдиним недоліком цієї СУБД є складність адміністрування, але це окупається ефективною та надійною роботою. Можливості: Можливість розділення великих баз даних на розділи, орієнтація на Інтернет-технології, вільні таблиці (в інших СУБД всі таблиці заповнюються відразу після створення), розпаралелювання операцій в запитах, наявність широкого спектру засобів розробки, моніторингу та засоби управління, наявність універсальних засобів захисту інформації. Oracle займає почесне друге місце після СУБД MySQL, значно перевершуючи всі інші СУБД за показниками надійності та безпеки;

SQLite. SQLite, вбудований механізм бази даних, не є окремим запущеним процесом, з яким взаємодіє ваша програма, а надає бібліотеку, у яку компілюється ваша програма, і механізм стає компонентом вашої програми. Тому виклики функцій бібліотеки SQLite використовуються як протокол обміну. Такий підхід зменшує накладні витрати та час відповіді та спрощує програму. Архітектура двигуна дозволяє використовувати SQLite як у вбудованих системах, так і на спеціалізованих машинах з гігабайтними масивами даних;

Firebird (Firebird SQL). Компактна база даних, яка працює на GNU/Linux, Microsoft Windows і різних платформах Unix. Переваги Firebird включають багатoversійну архітектуру, яка забезпечує паралельну обробку запитів на операції та аналіз (це можливо, оскільки користувачі не блокують записувачів),

компактність (5 Мб розподілу), високу ефективність, простоту використання збережених процедур і тригерів. підтримка мови. Firebird використовується в різних промислових системах з 2001 року. Це комерційно незалежний проєкт програмістів на С і С++, технічних консультантів і розробників кросплатформної системи управління базами даних на основі вихідного коду. Деякі з недоліків включають кешування результатів запиту та відсутність повнотекстового індексу для повнотекстового пошуку;

MySQL – це реляційна база даних, яка використовує запити SQL для обробки даних. Основними перевагами цього типу бази даних є швидкість і простота використання. MySQL дозволяє виконувати операції з даними, які важко реалізувати в текстових файлах.

Особливості MySQL. MySQL підтримує стандартну мову запитів SQL ANSI 92, а також багато розширень цього стандарту, яких немає в інших СУБД. Короткий список функцій MySQL:

- підтримує необмежену кількість користувачів, які одночасно працюють з базою даних;
- кількість рядків у таблиці може досягати 50 мільйонів;
- швидке виконання команд;
- проста та ефективна система безпеки.

Недоліки MySQL:

MySQL справді є дуже швидким сервером, але він вимагав від розробників пожертвувати деякими вимогами реляційних СУБД.

MySQL не має:

- підтримка вкладених запитів типу `SELECT * FROM table1 WHERE id IN (SELECT id FROM table2)`.
- підтримка транзакцій не реалізована. Ми рекомендуємо замість цього використовувати `LOCK/UNLOCK TABLE`.
- зовнішні ключі не підтримуються.
- тригери та збережені процедури не підтримуються.
- перегляди (VIEW) не підтримуються.

Access – це система керування базами даних (СУБД).

База даних (БД) зберігає набір даних і об'єктів, що належать до конкретного завдання. Це допоможе вам упорядкувати вашу інформацію та дає змогу швидко створювати зразки. Даними може бути будь-яка інформація, що зберігається на диску комп'ютера. Комбінація символів, слів, файлів, зображень тощо. Спосіб введення та зберігання даних визначає структуру бази даних.

СУБД Access дозволяє створювати реляційні бази даних, у яких дані зберігаються як таблиць.

Основні компоненти MS Access:

- конструктор таблиць;
- конструктор екранних форм;
- будівник запитів SQL (мова SQL MS Access відповідає стандартам ANSI);
- побудовник звітів для друку.

Оскільки вони можуть викликати скрипти на мові VBA, MS Access дозволяє розробляти програми та бази даних практично з нуля, а також створювати оболонки для зовнішніх баз даних.

Ядро бази даних Microsoft Jet, що використовується як ядро бази даних MS Access, є СУБД файлового сервера і тому застосовується тільки до додатків, які обробляють невеликі обсяги даних і мають невелику кількість користувачів, що працюють з цими даними. в той же час. У Access відсутні багато механізмів, такі як тригери, необхідні базі даних, призначеної для багатьох користувачів.

PhpMyAdmin - програма, написана на PHP, призначена для управління серверами MySQL у всесвітньому павутинні. PhpMyAdmin підтримує широкий спектр операцій із MySQL. Найбільш часто використовувані операції підтримуються через інтерфейс користувача (керування базами даних, таблицями, полями, відносинами, індексами, користувачами, дозволами і т. д.), але ви можете виконувати будь-який SQL-запит безпосередньо.

phpMyAdmin також має велику документацію у вигляді книги, написаної одним з його розробників, а Mastering phpMyAdmin для ефективного управління MySQL доступний англійською, чеською, німецькою та іспанською мовами.

Можливості phpMyAdmin:

- інтуїтивно зрозумілий веб-інтерфейс;
- підтримка більшості функцій MySQL;
- перегляд та видалення баз даних, таблиць, полів та індексів;
- створювати, копіювати, видаляти, перейменовувати та змінювати бази даних, таблиці, поля та індекси;
- консультації щодо сервера, бази даних, управління таблицями та налаштування сервера;
- запускате, змінюєте та зберігаєте будь-які SQL-вирази, включаючи пакетні запити.
- управління користувачами MySQL та їх привілеями.
- робота зі процедурами, що зберігаються, і тригерами.
- підтримка імпорту даних із CSV та SQL.
- підтримка експорту до різних форматів CSV, SQL, XML, PDF, ISO/IEC 26300 — Текст та таблиці OpenDocument, Word, Excel, LATEX та інші.
- управління кількома серверами.
- створіть схему вашої бази даних у форматі PDF.
- створюйте складні запити за допомогою функції «Шаблони запитів».
- повний чи частковий пошук у базі даних.
- перетворення даних на будь-який формат за допомогою призначеного набору функцій, таких як відображення даних BLOB-об'єктів у вигляді зображень або посилань для скачування.

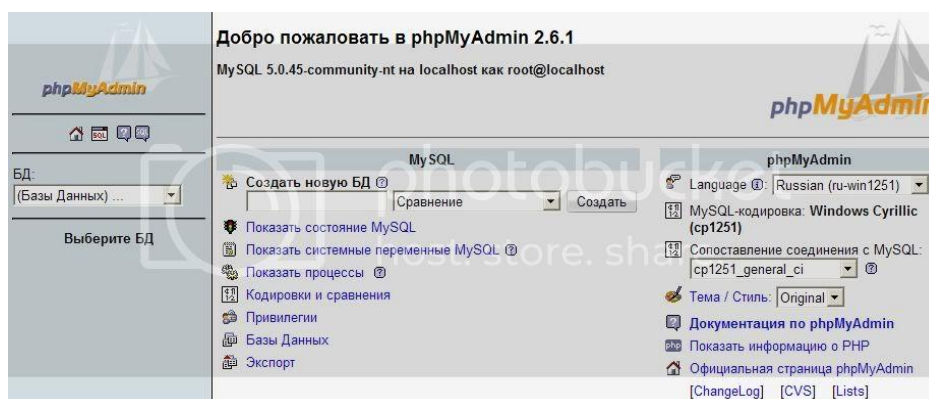


Рисунок 1.5 – Стартова сторінка PHPMyAdmin

Денвер використовувався як найбільш доступний та зручний варіант для розробки системи на локальній машині. PhpMyAdmin включено до списку інструментів цього набору.

Тому через його численні переваги у цій статті для розробки системи було обрано реляційну базу даних MySQL, а для роботи з сервером MySQL через Інтернет — phpMyAdmin.

2 РОЗРОБКА МОБІЛЬНОЇ СИСТЕМИ СКЛАДСЬКОГО ОБЛІКУ

2.1 Структурна розробка мобільних додатків

Розроблена програма ґрунтувалася на традиційній трирівневій архітектурі.

Архітектура системи складського обліку, що будується, складається з трьох взаємозалежних частин: бази даних, серверної частини та клієнтської частини.

Серверна частина програми виконує запит клієнтської частини. Серверна частина програми забезпечує зберігання даних та їх обробку, а клієнтська частина надсилає відповідні запити на сервер.

Клієнтська сторона реалізує функції генерації та розпізнавання QR-коду і, звичайно ж, сам інтерфейс користувача.

Багаторівнева архітектура має багато переваг:

- відсутність дублювання програмного коду сервера клієнтськими програмами;

- усі дані зберігаються на наших серверах. Сервери, як правило, набагато безпечніші, ніж більшість клієнтів. На сервері простіше забезпечити контроль дозволів, який дозволяє доступ до даних лише клієнтам із відповідними правами доступу.

- ви можете комбінувати різних клієнтів. Клієнти, що використовують різні апаратні платформи, операційні системи тощо, часто можуть використовувати ресурси одного і того ж сервера.

- знижує навантаження на мережу, оскільки між сервером та клієнтом передається лише невелика частина даних.

- незважаючи на безліч переваг, він має певні недоліки.

- відмова сервера може вивести з експлуатації всю комп'ютерну мережу. Непрацездатними серверами слід вважати сервери, продуктивність яких недостатня обслуговування всіх клієнтів, і навіть сервери, що у ремонті, профілактиці тощо.

– для підтримки роботи цієї системи потрібен окремий спеціаліст (системний адміністратор).

– висока вартість обладнання [15].

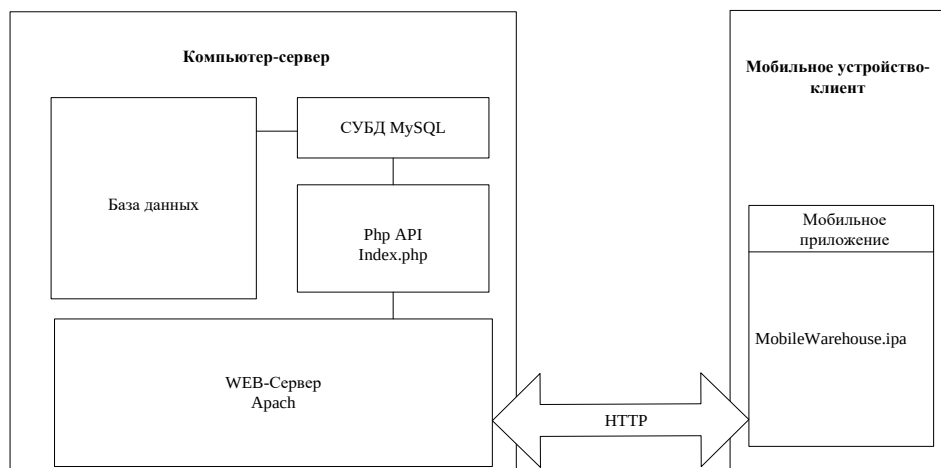


Рисунок 2.1 — Архітектура системи обліку мобільного складу ТОВ «АрСог»

Наступним етапом розробки стала розробка скриптів та програмної логіки. Сценарії повинні відображати вимоги користувача до вирішення конкретного завдання.

Структуру програми показано рис. 2.2.

2.2 Створення бази даних програми

Для роботи системи обліку мобільного складу необхідно було розробити базу даних (БД) складу, що розглядається.

По-перше, ви повинні самі побачити, яку інформацію містить база даних та які функції на неї покладено.

Інформаційні системи повинні зберігати таку інформацію:

- товари діляться на види.
- клієнт АрСог LLC;
- постачальник ТОВ «АрСог»;

- оператори, які працюють у цій компанії;

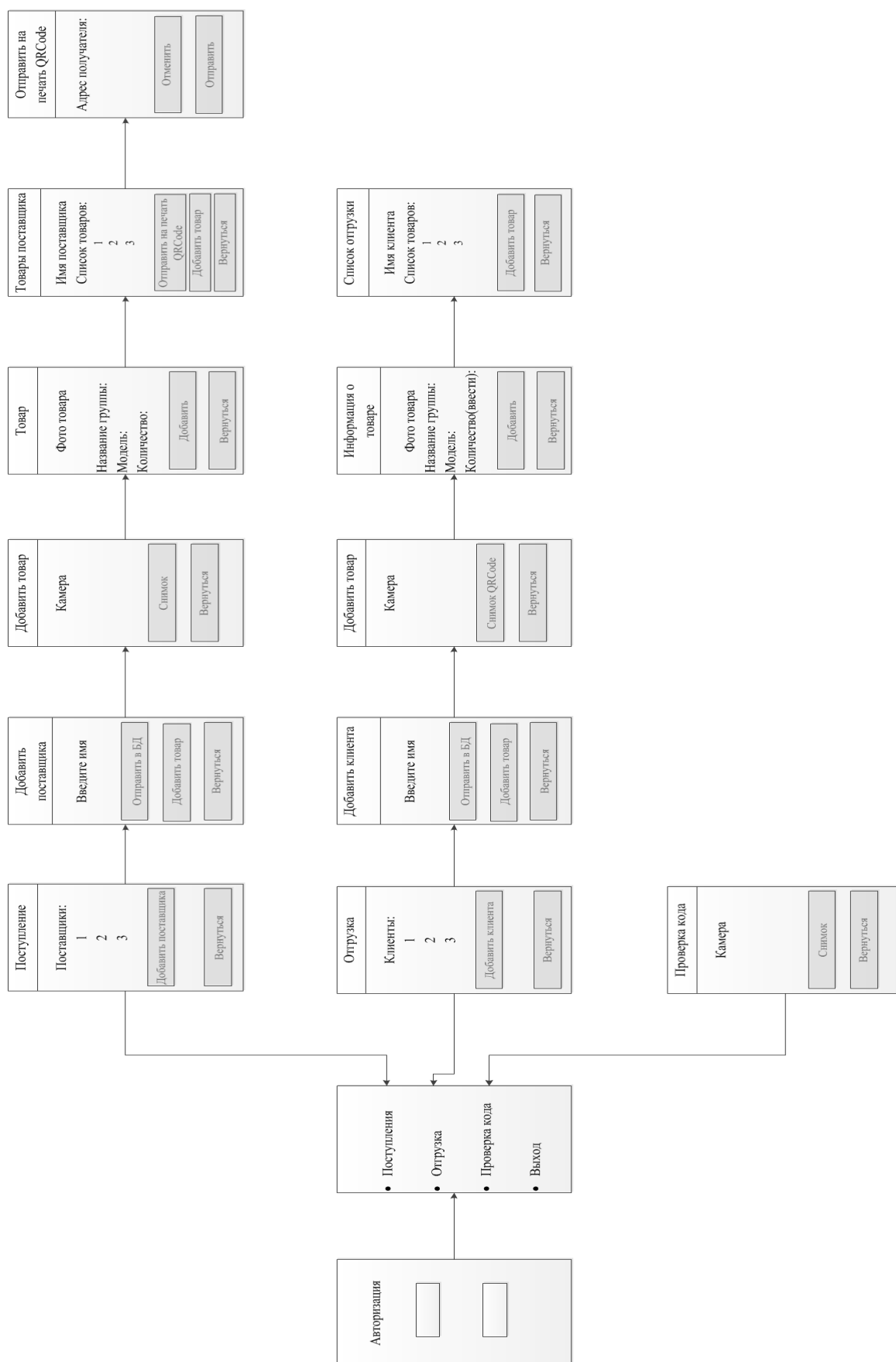


Рисунок 2.2 – Структура системи обліку мобільного складу ТОВ «АрСор»»

- список та комплектація відвантажень товарів;
- список та опис товарів, що надходять.

Створення бази даних починається з розробки структури її таблиць. Структура повинна бути такою, щоб менше даних потрібно було вводити до бази даних під час роботи з нею. Структура кожної таблиці розробляється окремо. Щоб зв'язок між таблицями працював надійно та записи в одній таблиці однозначно знаходили записи в іншій таблиці, таблиці мають бути забезпечені унікальним полем.

Одним із найскладніших етапів проектування бази даних є створення таблиць. Це з тим, що результати, формовані базою даних (звіти, форми виведення тощо), який завжди дають повне уявлення про структуру таблиці. Вам не потрібно використовувати СУБД для розробки таблиць. Я рекомендую спершу створити структуру на папері. Таким чином, ви повинні дотримуватися цих основних принципів при розробці своїх таблиць.

Інформація в таблиці не має дублюватися. Між столами не повинно бути повторень. Якщо певна інформація зберігається лише в одній таблиці, потрібно змінити її лише в одному місці. Це робить роботу більш ефективною, а також унеможливорює використання різної інформації в різних таблицях.

Кожна таблиця має містити інформацію лише з однієї теми. З даними на кожну тему набагато простіше працювати, якщо вони знаходяться в окремій таблиці. Наприклад, інформація про округ і результати голосування зберігаються в окремих таблицях, тому при повторному використанні бази даних інформація про округ залишається в базі даних.

Кожна таблиця містить інформацію про окрему тему, а кожне поле таблиці містить окрему інформацію про тему таблиці. Під час розробки полів кожної таблиці слід пам'ятати:

- кожне поле має ставитись до теми таблиці;
- не рекомендується включати дані формул у таблиці;
- таблиця має містити всю необхідну інформацію;
- інформація має бути поділена на найдрібніші логічні одиниці.

База даних, що розробляється, складається з восьми таблиць (рис. 2.1): «Клієнт», «Оператор», «Постачальник», «Відвантаження», «Склад відвантаження», «Надходження», «Склад-одержувач», «Товар» і «Тип» товару».

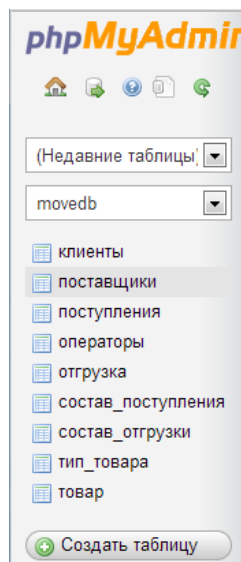


Рисунок 2.3 – Створені таблиці бази даних

На рисунках з 2.4 до 2.12 показано структуру таблиці додатків.

 Обзор	 Структура	 SQL	 Поиск	 Вставить	 Экспорт	 Импорт	
#	Имя	Тип	Сравнение	Атрибуты	Null	По умолчанию	Дополнительно
☐	1 <u>Код клиента</u>	int(11)			Нет	Нет	AUTO_INCREMENT
☐	2 ФИО	varchar(100)	utf8_general_ci		Нет	Нет	
☐	3 Адрес	varchar(200)	utf8_general_ci		Нет	Нет	
☐	4 Телефон	int(11)			Да	NULL	
☐	5 Расчетный счет	int(11)			Да	NULL	

Рисунок 2.4 – Таблица «Клієнти»

Обзор	Структура	SQL	Поиск	Вставить	Экспорт	Импорт	
#	Имя	Тип	Сравнение	Атрибуты	Null	По умолчанию	Дополнительно
☐	1 Номер поставщика	int(11)			Нет	Нет	AUTO_INCREMENT
☐	2 Название_поставщика	varchar(100) utf8_general_ci			Нет	Нет	

Рисунок 2.5 – Таблица «Постачальник»

Обзор Структура SQL Поиск Вставить Экспорт Импорт							
#	Имя	Тип	Сравнение	Атрибуты	Null	По умолчанию	Дополнительно
☐	1 <u>Номер партии</u>	int(11)			Нет	Нет	AUTO_INCREMENT
☐	2 Код_поставщика	int(11)			Нет	Нет	
☐	3 Дата	timestamp			Да	CURRENT_TIMESTAMP	
☐	4 Код_оператора	int(11)			Да	NULL	

Рисунок 2.6 – Таблица «Квитанция»

Обзор Структура SQL Поиск Вставить Экспорт Импорт							
#	Имя	Тип	Сравнение	Атрибуты	Null	По умолчанию	Дополнительно
☐	1 <u>Код оператора</u>	int(11)			Нет	Нет	AUTO_INCREMENT
☐	2 ФИО	varchar(50)	utf8_general_ci		Нет	Нет	
☐	3 Имя_в_системе	varchar(255)	utf8_general_ci		Нет	Нет	
☐	4 Пароль	varchar(8)	utf8_general_ci		Нет	Нет	

Рисунок 2.7 – Таблица «Операторы»

Обзор Структура SQL Поиск Вставить Экспорт Импорт							
#	Имя	Тип	Сравнение	Атрибуты	Null	По умолчанию	Дополнительно
☐	1 <u>Код договора</u>	int(11)			Нет	Нет	AUTO_INCREMENT
☐	2 Код_клиента	int(11)			Да	NULL	
☐	3 Дата	datetime			Да	NULL	
☐	4 Код_оператора	int(11)			Да	NULL	

Рисунок 2.8 – Таблица «Доставка»

Обзор Структура SQL Поиск Вставить Экспорт Импорт							
#	Имя	Тип	Сравнение	Атрибуты	Null	По умолчанию	Дополнительно
☐	1 <u>Код состава поступления</u>	int(11)			Нет	Нет	AUTO_INCREMENT
☐	2 Номер_партии	int(11)			Нет	Нет	
☐	3 Код_товара	int(11)			Нет	Нет	
☐	4 Количество	int(11)			Да	1	

Рисунок 2.9 – Таблица «Структура чека»

Обзор Структура SQL Поиск Вставить Экспорт Импорт							
#	Имя	Тип	Сравнение	Атрибуты	Null	По умолчанию	Дополнительно
☐ 1	<u>Код состава отгрузки</u>	int(11)			Нет	Нет	AUTO_INCREMENT
☐ 2	Код_договора	int(11)			Да	0	
☐ 3	Код_товара	int(11)			Да	NULL	
☐ 4	Количество	tinyint(4)			Да	1	

Рисунок 2.10 – Таблица «Конфігурація доставки»

Обзор Структура SQL Поиск Вставить Экспорт Импорт							
#	Имя	Тип	Сравнение	Атрибуты	Null	По умолчанию	Дополнительно
☐ 1	<u>код группы</u>	int(11)			Нет	Нет	AUTO_INCREMENT
☐ 2	Название_группы	varchar(50) utf8_general_ci			Нет	Нет	

Рисунок 2.11 – Таблица "Тип продукту"

Обзор Структура SQL Поиск Вставить Экспорт Импорт							
#	Имя	Тип	Сравнение	Атрибуты	Null	По умолчанию	Дополнительно
☐ 1	<u>Код товара</u>	int(11)			Нет	Нет	AUTO_INCREMENT
☐ 2	Код_группы	int(11)			Нет	Нет	
☐ 3	Модель	varchar(50) utf8_general_ci			Нет	Нет	
☐ 4	Дата_выпуска	year(4)			Нет	Нет	
☐ 5	Колво_на_складе	tinyint(4)			Да	NULL	
☐ 6	Путь_к_картинке	varchar(255) utf8_general_ci			Да	NULL	

Рисунок 2.12 – Таблица «Товары»

2.2.1 Схема реляційної бази даних

Реляційна база даних – це набір відносин, який містить всю інформацію, яку вам потрібно зберігати в базі даних, і, виходячи з того, що база даних, що розробляється, є реляційною, вона володіє наступними властивостями:

- кожна таблиця складається з рядків одного типу та має унікальне ім'я;
- рядки мають фіксовану кількість полів (стовпців) і значень, множинні поля та групи, що повторюються, не допускаються. Тобто в кожному місці таблиці, де перетинаються рядки і стовпці, завжди є одне значення або взагалі немає;

– рядки таблиці мають відрізнятись один від одного хоча б на одне значення.
Це дозволяє однозначно ідентифікувати будь-який рядок у такій таблиці;

– стовпці в таблиці мають унікальні імена і в кожному розміщуються однорідні значення даних (дата, прізвище, ціле число або грошова сума);

– повне інформаційне наповнення бази даних подано у вигляді явних значень даних. Цей спосіб уявлення є єдиним. Зокрема, відсутні спеціальні посилання або покажчики, що з'єднують одну таблицю з іншою;

– під час виконання операцій над таблицею її рядки та стовпці можуть оброблятися у будь-якому порядку, незалежно від інформаційного наповнення. Цьому сприяє наявність назви таблиці та її стовпців, і навіть можливість вибору рядка чи набору рядків із заданими характеристиками.

Схема бази даних показана на рис. 2.13 нижче містить такі об'єкти:

- Клієнт (Код_Клієнта, ПІБ (назва організації));
- Відвантаження (код_договору, код_клієнта, дата, код_оператора);
- Склад_відвантаження (Код_відвантаження_складу, Код_договору, Код_товару, Кількість);
- Товар (Код_Товара, Код_Групи, Модель, Кількість_в_наявності, Let_to_picture);
- Тип_Товара (Код_Групи, Назва_Групи);
- Склад_приймання (Код_складу_отримання, Номер_лоту, Код_товару, Кількість);
- чек (номер партії, код постачальника, дата, код оператора);
- Постачальник (номер постачальника, найменування постачальника);
- Оператор (код оператора, ПІБ, ім'я в системі, пароль).

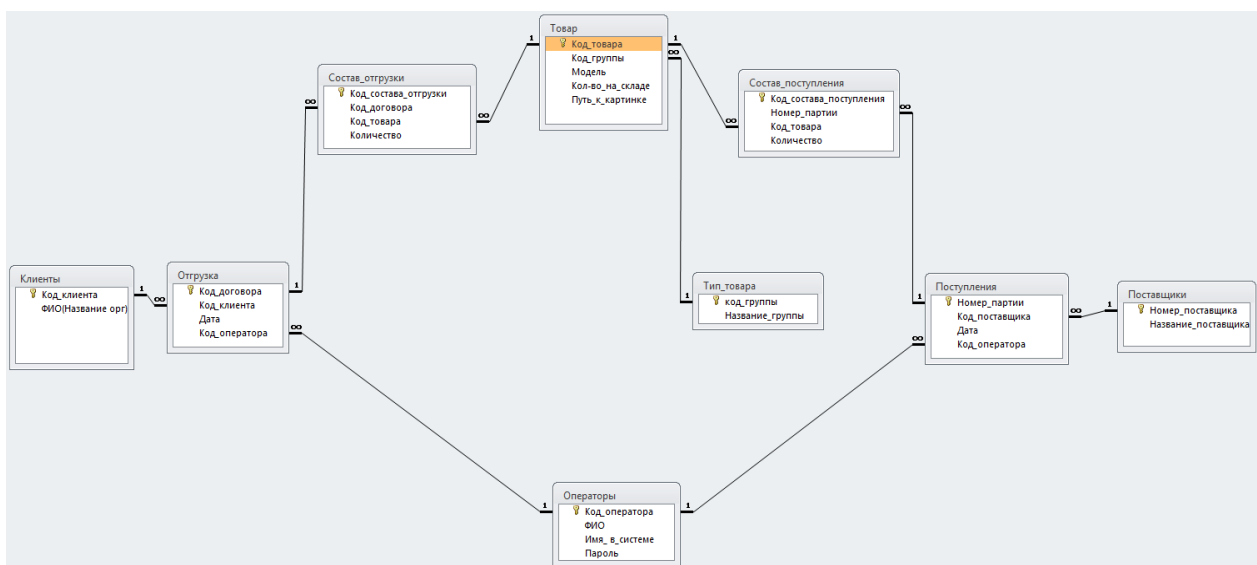


Рисунок 2.13 – Схема бази даних

2.2.2 Розробка запиту

Для того щоб створювана нами система обліку мобільного складу працювала, нам необхідно робити запити на стороні сервера. Для авторизації оператора потрібен запит на відповідність логіну та паролю. Можливість додавання нових продуктів передбачає відповідні запити. Запит на пошук товару створено для оптимізації та зручності роботи оператора. При відвантаженні зі складу вводиться запит на формування списку товарів, що відвантажуються для наочності.

Запит на затвердження надсилається лістингом А.1.

Запит пошуку продукту представлений у лістингу А.2.

Запит на додавання нового продукту представлений у лістингу А.3.

У лістингу А.4 показаний запит створення відвантажувального складу.

2.3 Апаратне забезпечення системи

Мінімальні вимоги до обладнання для системи обліку мобільного складу, що розробляється: ПК, мобільний телефон Apple моделі 4, 4S або 5.

Для роботи системи потрібен комп'ютер, сумісний із IBM PC.

Для операційних систем WINDOWS і Linux вам знадобляться як мінімум наступні конфігурації:

- процесор Intel Pentium IV.
- ОЗУ 512 МБ та більше;
- не менше 4 ГБ вільного місця на диску;
- монітор;
- клавіатура;
- маніпулятор "Миша".

Для операційної системи iOS:

- процесор Intel Core i3
- ОЗУ 512 МБ та більше;
- не менше 4 ГБ вільного місця на диску;
- маніпулятор "Миша".

Сервер можна використовувати замість стандартного персонального комп'ютера.

Обмін даними між мобільними пристроями та серверами здійснюється за технологією Wi-Fi стандартів 802.11a/b/g/n (802.11n, 2,4 ГГц та 5 ГГц).

Стандарт 802.11n для мереж Wi-Fi був ратифікований Інститутом інженерів з електротехніки та електроніки (IEEE) 11 вересня 2009 року.

Стандарт 802.11n включає безліч покращень порівняно з пристроями 802.11g.

Пристрої 802.11n можуть працювати в одному з двох діапазонів: 2,4 ГГц або 5,0 ГГц.

Фізично (PHY) реалізована поліпшена обробка і модуляція сигналу, додана можливість одночасної передачі сигналу через чотири антени.

Більш ефективне використання доступної пропускної спроможності реалізовано на мережевому рівні (MAC).

2.4 Функції системи

Проаналізувавши існуючу у вас систему складського обліку та визначивши, які завдання ви хочете покласти на систему, що розробляється, ви можете сформулювати її функціональні можливості.

1) У системі передбачена можливість надсилання даних на сервер для подальшої обробки та зберігання.

2) На основі фотографій та спрощених даних про товар система створює ідентифікатор для зберігання та обліку;

3) Ця система дозволяє отримати інформацію про продукт за допомогою лише ідентифікаційного коду;

4) У системі є можливість групувати товари за постачальниками;

5) У системі передбачено можливість ведення складських звітів;

6) Система об'єднує продукти на сервері за групуванням, загальним функціям;

7) У системі передбачено можливість редагування даних у разі виникнення помилок;

8) Буде згенерований QR-код з ідентифікатором продукту.

QR-код – це тип матричного коду (двовимірний штрих-код), створений японською корпорацією Denso Wave у 1994 році та використовується для швидкої передачі його вмісту користувачам. "QR" означає "Швидка відповідь".

Спочатку QR-коди застосовувалися в машинобудуванні для запису деталей, але тепер вони використовуються не тільки в комерційних системах, а й для швидкої доставки інформації користувачам мобільних телефонів. QR-коди можуть містити різноманітну інформацію, включаючи контактну інформацію, текст, адреси та гіпертекстові посилання. Більшість сучасних мобільних телефонів мають можливість розпізнавати QR-коди. Це позбавляє користувачів необхідності запам'ятовувати довгі посилання та адреси і дозволяє їм сфотографувати код і отримати потрібну інформацію.

Система працює за алгоритмом, представленим на блок-схемі рис. 2.18.

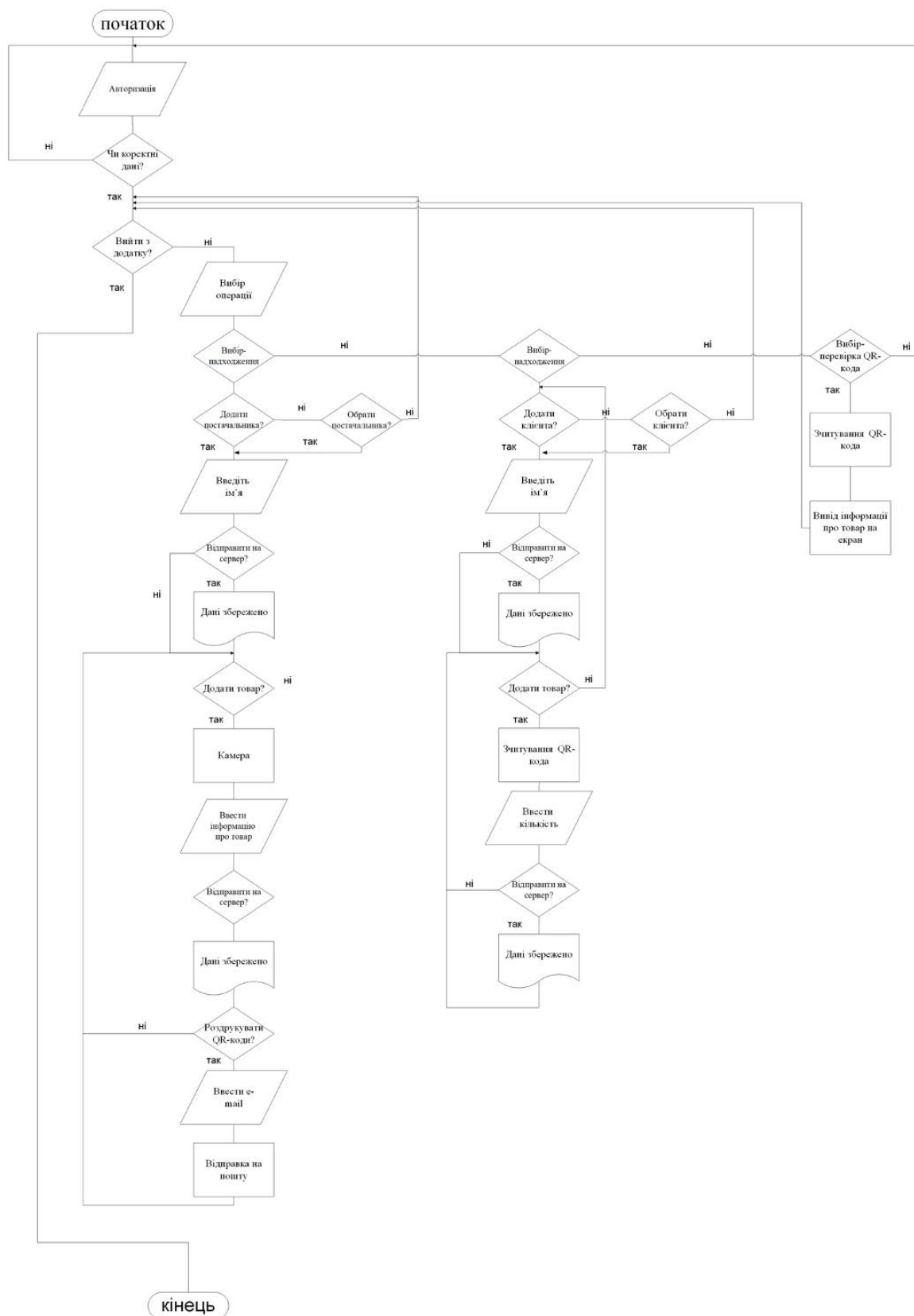


Рисунок 2.14 – Алгоритм програми

2.4.1 Реалізація системи

Деякі реалізації функціональних рішень представлені у вигляді списку. У лістингу А.5 показано код функції авторизації оператора.

Код створення інтерфейсу вибору дії показаний у лістингу 2.6.

Вам потрібно буде роздрукувати QR-код, згенерований для нового продукту, і вставити його в свій продукт. Для цього програма створює PDF-документ, що містить зображення QR-коду.

Окремо реалізовано можливість перетворення матриці QR-коду у формат зображення.

3 РЕЗУЛЬТАТИ РОЗРОБКИ СИСТЕМИ

3.1 Графічний інтерфейс користувача

Для запуску системи обліку мобільного складу вам необхідно натиснути на піктограму програми на екрані iPhone. Після запуску програми відкривається вікно аутентифікації оператора (рис. 3.1).

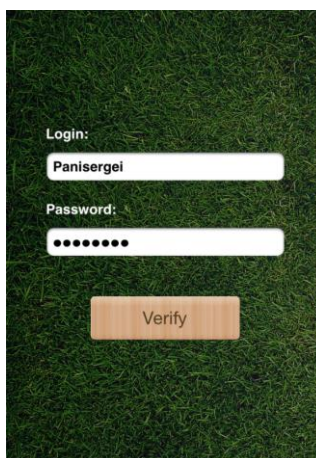


Рисунок 3.1 – Вікно аутентифікації оператора

Далі перейдіть у вікно "Вибір дії": "Доставка товару", "Відвантаження" або "Підтвердження QR-коду" (рис. 3.2). Натисніть відповідну кнопку, щоб вибрати

потрібну дію. Натисніть кнопку «Завершити», щоб завершити сеанс поточного оператора та відобразити вікно автентифікації іншого оператора.

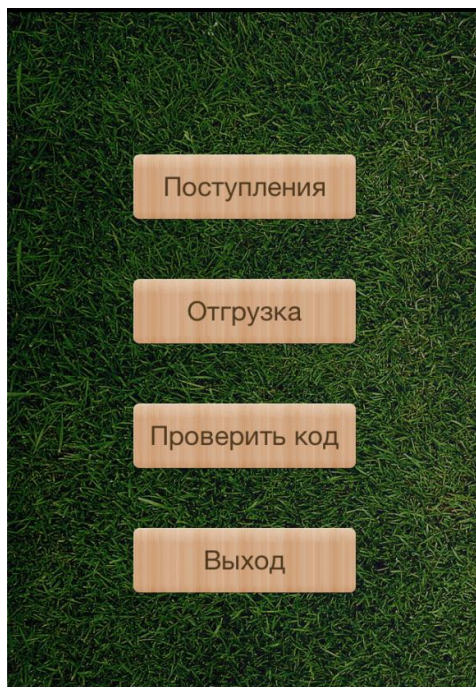


Рисунок 3.2 – Вікно вибору дії

Для додавання нового об'єкта (Постачальник, Клієнт, Товар) необхідно використовувати кнопку плюс у правому верхньому кутку.

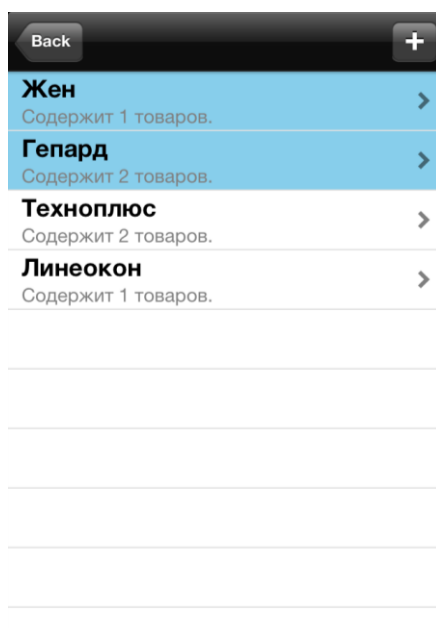


Рисунок 3.3 – Перегляд списку постачальників та замовників

Якщо ви додаєте нового постачальника або клієнта, потрібно ввести його ім'я.

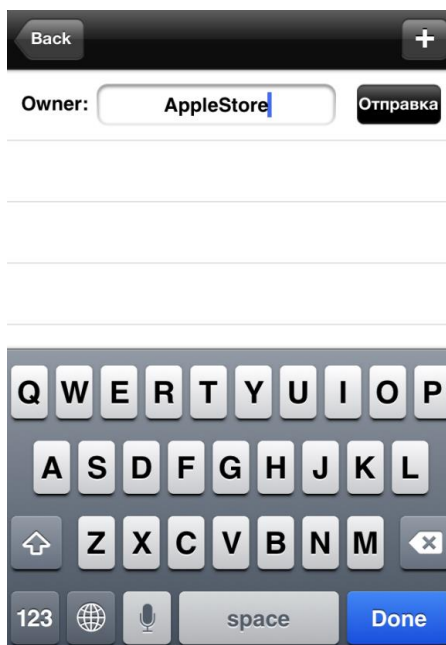


Рисунок 3.4 – Поле введення імені

Додавання нового продукту містить камеру. Натисніть кнопку зображення камери, щоб зробити знімок продукту.

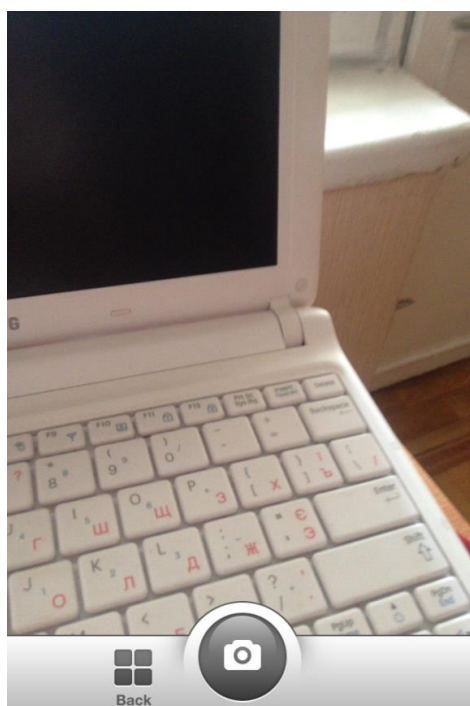


Рисунок 3.5 – Створити фото товару

Створення фотографій відкриває список усіх видів товарів.

Далі йде форма введення інформації про продукт.

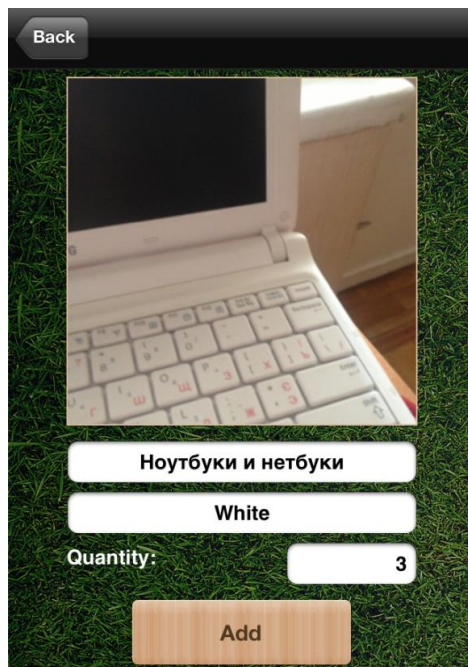


Рисунок 3.7 – Форма введення інформації про продукт

Натисніть кнопку «Надіслати», щоб надіслати дані на сервер. Дані вже виділені синім кольором на сервері.

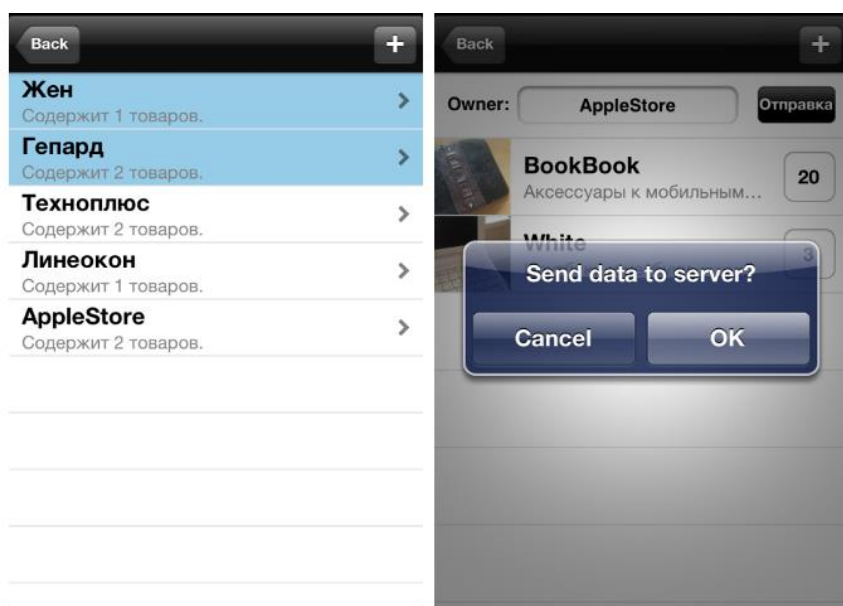


Рисунок 3.8 – Зміни після надсилання даних на сервер

Після натискання кнопки «Друк» згенерований QR-код товару буде надіслано на пошту для подальшого друку.

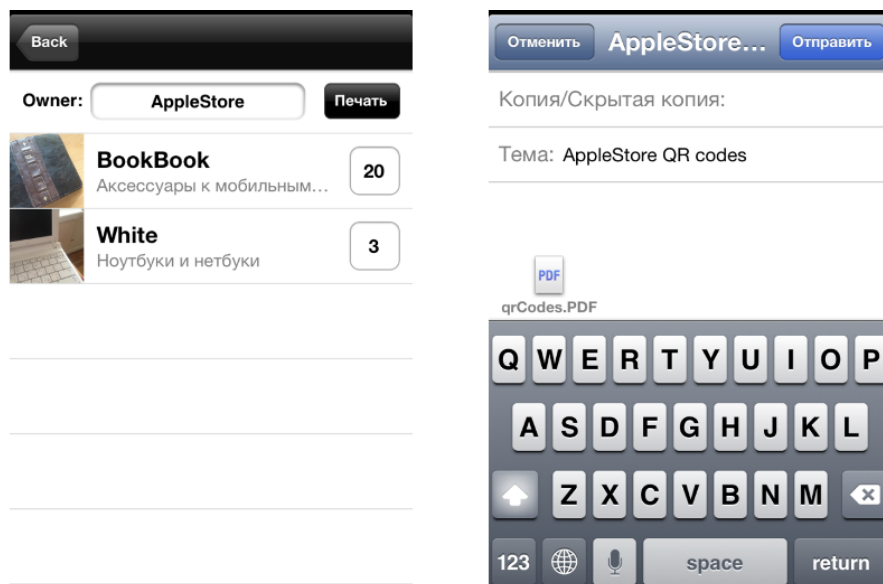


Рисунок 3.9 – Надсилання QR-коду електронною поштою у форматі PDF

Відскануйте QR-код, щоб відкрити камеру. Вам просто потрібно навести його на зображення QR-коду, і камера автоматично сфокусується та розпізнає код.

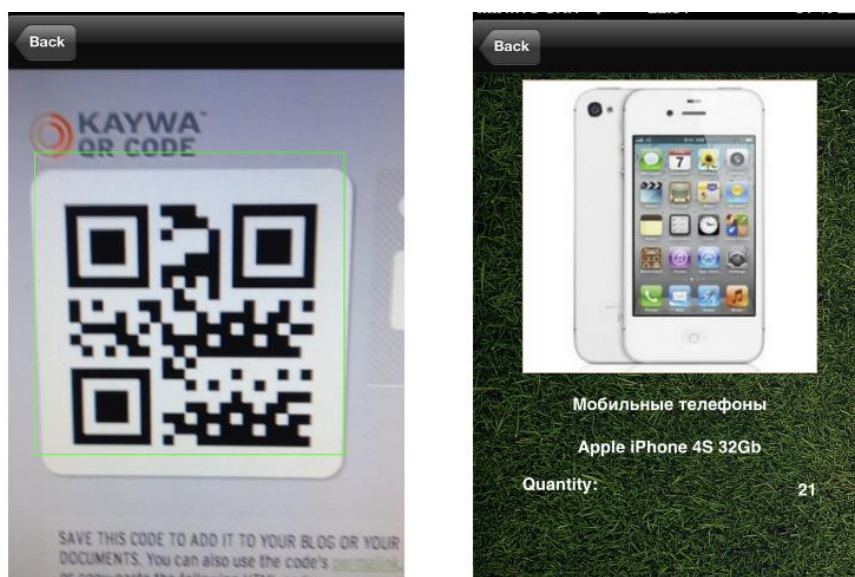


Рисунок 3.10 – Розпізнавання QR-коду

Щоб видалити поле у списку, потрібно провести пальцем по рядку праворуч наліво. Відобразиться кнопка Видалити.

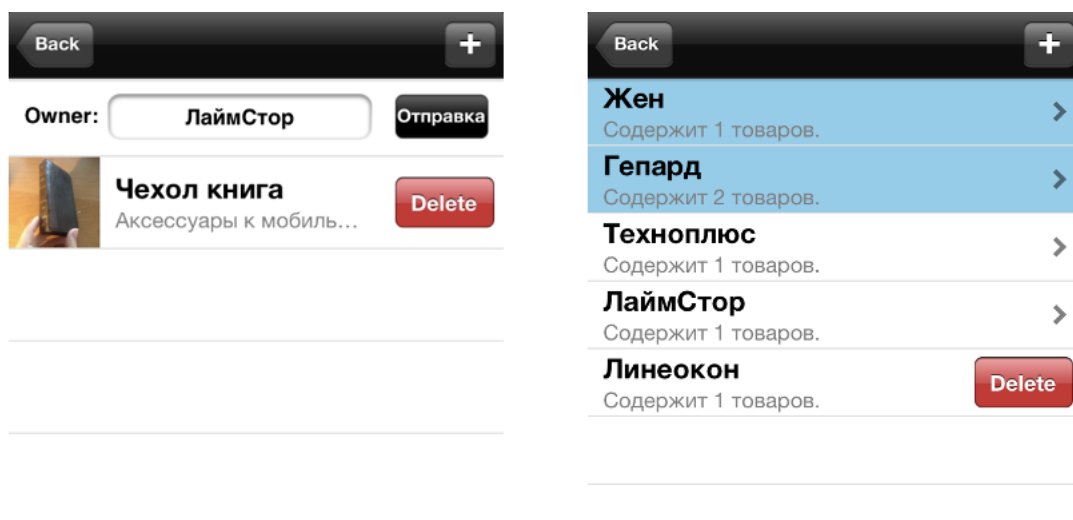


Рисунок 3.11 – Варіанти використання кнопки Видалити.

Також у будь-якому вікні програми є кнопка «Назад» у верхньому лівому кутку для повернення на попередню сторінку.

3.2 Напрями удосконалення програми

Розроблена система реалізує всі функції, необхідні для повноцінної системи складського обліку, але є портативною, що значно підвищує зручність використання.

Однак одержана система може бути додатково удосконалена.

По-перше, ви можете додати більше параметрів товару, таких як ціна, знижки та можливість входу до накопичувальної системи покупця.

По-друге, надати можливість оператору самостійно здійснювати контроль та формувати звіти по телефону.

По-третє, важливо додати можливість збирання статистики додатків для виявлення проблемних моментів та усунення недоліків.

По-четверте, нам потрібно надати можливість розвантаження історії, щоб з додатком було простіше працювати. Тому у разі «вильоту» програми чи обриву

мережі в історії збережеться інформація про попереднє відвідування, що заощадить багато часу на обробку доставки чи відвантаження.

По-п'яте, ви можете скористатися послугами дизайнерів для створення більш ергономічного та стильного інтерфейсу програми та оптимізації роботи із системою.

По-шосте, у статті не приділялося особливої уваги захисту даних за сервера. Тому, якщо ви хочете використовувати цю систему обліку мобільного складу у великій компанії, вам доведеться доробити цю частину системи.

По-сьоме, розроблена система складського обліку підтримується лише на мобільних телефонах Apple. У майбутньому також можна забезпечити підтримку інших мобільних ОС.

ВИСНОВКИ

Таким чином, в цій роботі розроблено мобільну систему складського обліку ТОВ «АрСог» із зручним та інтуїтивно зрозумілим графічним інтерфейсом. При цьому всі завдання, поставлені у дипломному проєкті, виконані.

Існуюча система складського обліку EME. За допомогою WMS, IBS Trade & Warehouse, Oracle Warehouse Management System, DataMobile та MobileSOP ми проаналізували можливості організації складського обліку та розробили власну систему складського обліку. В результаті аналізу було прийнято рішення щодо розробки унікальної системи, для чого були складені основні вимоги.

Розроблено структуру мобільного додатка. Він складається з функцій автентифікації оператора, реєстрації доставки або відвантаження та перевірки QR-коду.

Була створена база даних, що містить наступні вісім таблиць: Клієнти, Оператори, Постачальники, Відвантаження, Склад відвантаження, Надходження, Склад отримання, Продукт, Тип продукту.

Сформульовано мінімальні апаратні вимоги до системи мобільного складського обліку: комп'ютер сумісний з IBM PC мобільний телефон Apple Model 7.

Створено мобільну систему складського обліку, яка має зрозумілий, навіть для нетренованого користувача, інтерфейс, вона зручна у використанні, також у ній передбачена можливість зміни даних, у разі помилки. Система реалізована з використанням мови програмування Objective-C та середовища розробки XCode. База даних програми створювалася мовою MySQL, для роботи з БД через Інтернет використовувався PhpMyAdmin.

Створена система не є вузькоспеціальною і може бути використана на складських установах, що реалізують продукцію будь-яких типів.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Stephen Cass “The Top Programming Languages 2021” // Офіційна сторінка журналу IEEE Spectrum [Електронний ресурс]. – Режим доступу: <https://spectrum.ieee.org/top-programming-languages-2021>
2. Denisova L A and Meshcheryakov V A 2018 Control systems design: the technology of stochastic perturbations simulation J. Phys.: Conf. Ser. 1050 012020
3. X. Liang and J. Xu, “Control for networked control systems with remote and local controllers over unreliable communication channel,” Automatica, vol. 98, no. 2, pp. 86–94, December 2018.
4. Інформаційний ресурс Texas Instruments [Електронний ресурс]. – Режим доступу: <https://www.ti.com/>
5. Сайт з надання технічних визначень [Електронний ресурс]. – Режим доступу: <https://www.techopedia.com/definition/31065/automated-system-operations-aso>
6. Інформаційні системи в менеджменті: Навчальний посібник // Батюк А.Є., Двудіт З.П., Обельовська К.М., Огородник І.М. та ін. – К.: ІнтелектЗахід, 2004. – 520 с.
7. Інформаційне забезпечення менеджменту // Новак В.О., Макаренко Л.Г., Луцький І.Г. – К.: Кондор, 2006. – 462 с.
8. Тлумачний словник з інформатики / Г.Г.Півняк, Б.С.Бусигін, М.М.Дівізінюк та ін. – Дніпропетровськ: Нац. гірн. ун-т, 2008. – 599 с.
9. W. Hou, M. Fu, H. Zhang, and Z. Wu, “Consensus conditions for general second-order multi-agent systems with communication delay,” Automatica, vol. 75, no. C, pp. 293–298, January 2017.
10. D. Y. Meng, M. J. Du, and Y. M. Jia, “Interval bipartite consensus of networked agents associated with signed digraphs,” IEEE Transactions on Automatic Control, vol. 61, no. 12, pp. 3755–3770, December 2016.

11. X. Liang, J. Xu, and H. Zhang, "Optimal control and stabilization for networked control systems with asymmetric information," *IEEE Transactions on Control of Network Systems*, vol. 7, no. 3, pp. 1355–1365, February 2020.
12. Ismoilov, M.I., A.B. Nikolaev and A.V. Ostroukh, Petrochemical Enterprise.. *International Journal of 2013. Training and retraining of personnel and Advanced Studies*. Vol. 2. No 3. pp. 6. DOI: industrial enterprises of transport complexes using 10.12731/2227-930X-2012-3-6. mobile technology. Saint-Louis, MO, USA:
13. Ostroukh, A.V. and A.B. Nikolaev, 2013. Publishing House Science and Innovation Center, Development of virtual laboratory experiments in 166 p. ISBN 978-0-615-67111-6.
14. iLABS environment. *International Journal Of Applied* 27. Ostroukh, A.V., K.A. Barinov, A.B. Nikolaev and And Fundamental Research. 2013. No 2; URL: V.Yu. Stroganov, 2014. Formal methods for the www.science-sd.com/455-24313. synthesis of the organizational structure of the 19.
15. Ostroukh, A.V. and A.B. Nikolaev, 2013. management through the personnel recruitment at the Development of virtual laboratory experiments in industrial enterprises. *Journal of Applied Sciences iLabs. International Journal of Online Engineering (JAS)*. 14, No 5. pp. 474-481. DOI: (IJOE). Vol. 9, No 6. pp. 41-44. DOI: 10.3991/ 10.3923/jas.2014.474.481.

ДОДАТОК А

Лістинг А.1 – Запит на авторизацію

```

case "authorization" :
if (isset($_POST["login"]) && isset($_POST["password"]))
{
    $login = $_POST["login"];
    $password = $_POST["password"];
    $operatorCode = 0;
    $operatorName = "";

    $stmt = $this->db->prepare('SELECT Код_оператора, ФИО FROM
Операторы WHERE Имя_в_системе=? AND Пароль=?');
    $stmt->bind_param("ss", $login, $password);
    $stmt->execute();
    $stmt->bind_result($operatorCode, $operatorName);
    while ($stmt->fetch()) {
        break;
    }
    $stmt->close();

    if ($operatorCode <= 0) {
        sendResponse(400, 'Login and password doesn\'t exist');
        return false;
    }

    // Return unlock code, encoded with JSON
    $result = array(
        "operatorCode" => $operatorCode,
        "operatorName" => $operatorName,);
    sendResponse(200, json_encode($result));
    return true;
}
break;

```

Лістинг А.2 — Запит на пошук товару

```

case "getProduct" :
if (isset($_POST["productCode"]))
{
    $productCode = $_POST["productCode"];

    $groupName = '';
    $model = '';
    $imageUrl = '';
    $productAmount = -1;

    $stmt = $this->db->prepare('SELECT Тип_товара.Название_группы,
Товар.Модель, Товар.Колво_на_складе, Товар.Путь_к_картинке,

```

```

Товар.Код_товара FROM Товар LEFT JOIN Тип_товара ON
Тип_товара.код_группы=Товар.Код_группы WHERE Код_товара=?');
$stmt->bind_param("i", $productCode);
$stmt->execute();
$stmt->bind_result($groupName, $model, $productAmount,
$imageURL, $productID);
while ($stmt->fetch()) {
    break;
}
$stmt->close();

if ($productAmount <= 0 || $groupName == '' || $model == '' ||
$imageURL == '' || $productID == '') {
    sendResponse(400, 'Product doesn\'t exist');
    return false;
}

// Return unlock code, encoded with JSON
$result = array(
    "groupName" => $groupName,
    "model" => $model,
    "productAmount" => $productAmount,
    "imageURL" => $imageURL,
    "productID" => $productID
);
sendResponse(200, json_encode($result));
return true;
}
break;

```

Лістинг А.3 — Запит на додавання нового товару

```

case "saveNewProducts" :

    $products = $_POST["Products"];
    $operatorCode = $_POST["operatorCode"];
    $productOwner = $_POST["ProductCartOwner"];

    $uploadDir = 'upload/';

    if (!empty($_FILES)) {
        $count = 0;
        foreach($_FILES as $key => $value)
        {
            $fileTmp = $value['tmp_name'];
            $uploadfile = $uploadDir . basename($value['name']);
            move_uploaded_file($fileTmp, $uploadfile);
            $products[$count]["imageURL"] = $uploadfile;
            $count = $count + 1;
        }
    }

```

```

        $stmt = $this->db->prepare("INSERT INTO Поставщики
(Название_поставщика) VALUES (?)");
        $stmt->bind_param("s", $productOwner);
        $stmt->execute();
        $stmt->close();

        $stmt = $this->db->prepare("SELECT LAST_INSERT_ID()");
        $stmt->execute();
        $stmt->bind_result($ownerCode);
        while ($stmt->fetch()) {
        }
        $stmt->close();

        $stmt = $this->db->prepare("INSERT INTO Поступления
(Код_поставщика, Код_оператора) VALUES (?,?)");
        $stmt->bind_param("ii", $ownerCode, $operatorCode);
        $stmt->execute();
        $stmt->close();

        $stmt = $this->db->prepare("SELECT LAST_INSERT_ID()");
        $stmt->execute();
        $stmt->bind_result($packageId);
        while ($stmt->fetch()) {
        }
        $stmt->close();

        $productIDs = array();

        foreach($products as $key => $value)
        {
            $stmt = $this->db->prepare("SELECT код_группы FROM Тип_товара
WHERE Название_группы=?");
            $stmt->bind_param("s", $value["groupName"]);
            $stmt->execute();
            $stmt->bind_result($groupCode);
            while ($stmt->fetch()) {
            }
            $stmt->close();

            $stmt = $this->db->prepare("INSERT INTO Товар (Код_группы,
Модель, Колво_на_складе, Путь_к_картинке) VALUES (?, ?, ?, ?)");
            $stmt->bind_param("isis", $groupCode, $value["model"],
$value["productAmount"], $value["imageUrl"]);
            $stmt->execute();
            $stmt->close();

            $stmt = $this->db->prepare("SELECT LAST_INSERT_ID()");
            $stmt->execute();
            $stmt->bind_result($productId);
            while ($stmt->fetch()) {
            }
            $stmt->close();

```

```

array_push($productIDs,$productId);
}

foreach($productIDs as $key => $value)
{
    $stmt = $this->db->prepare("INSERT INTO Состав_поступления
(Номер_партии, Код_товара) VALUES (?,?)");
    $stmt->bind_param("ii",$packageId, $value);
    $stmt->execute();
    $stmt->close();
}
sendResponse(200, json_encode($productIDs));
return true;
}

sendResponse(400, 'Upload Failed');
return false;
break;

```

Лістинг А.4 — Запит на створення складу відвантаження

case "setProductCart" :

```

$product = $_POST["Products"];
$operatorCode = $_POST["operatorCode"];
$productCartOwner = $_POST["ProductCartOwner"];

$stmt = $this->db->prepare("INSERT INTO Клиенты (ФИО) VALUES
(?)");
$stmt->bind_param("s", $productCartOwner);
$stmt->execute();
$stmt->close();

$stmt = $this->db->prepare("SELECT LAST_INSERT_ID()");
$stmt->execute();
$stmt->bind_result($ownerCode);
while ($stmt->fetch()) {
}
$stmt->close();

$stmt = $this->db->prepare("INSERT INTO Отгрузка (Код_клиента,
Код_оператора) VALUES (?,?)");
$stmt->bind_param("ii", $ownerCode, $operatorCode);
$stmt->execute();
$stmt->close();
$stmt = $this->db->prepare("SELECT LAST_INSERT_ID()");
$stmt->execute();
$stmt->bind_result($packageId);
while ($stmt->fetch()) {
}
$stmt->close();

foreach($product as $key => $value)

```

```

    {
        $this->db->query("UPDATE Товар SET
Колво_на_складе=Колво_на_складе-? WHERE Код_товара=?");
        $stmt->bind_param("ii", $value["productAmount"],
$value["productID"]);
        $this->db->commit();
        $stmt = $this->db->prepare("INSERT INTO Состав_отгрузки
(Код_договора, Код_товара, Количество) VALUES (?, ?, ?)");
        $stmt->bind_param("iii", $packageId, $value["productID"],
$value["productAmount"]);
        $stmt->execute();
        $stmt->close();
    }
    sendResponse(200, 'Success');
    return true;

break;

```

Лістинг А.5 — Авторизація операторів

```

- (void) sendAuthorizationRequestWithLogin: (NSString*) login
password: (NSString*) password
{
    NSURL* serverAddress = [NSURL URLWithString:[NSString
stringWithFormat:@"http://%@", [[SettingsManager sharedManager]
serverAddress]]];
    AFHTTPClient* sharedClient = [AFHTTPClient
clientWithBaseURL:serverAddress];
    NSDictionary *params = [NSDictionary
dictionaryWithObjectsAndKeys:
@"authorization", @"actionType",
login, @"login",
password, @"password",
nil];
    NSMutableURLRequest *request = [sharedClient
requestWithMethod:@"POST" path:@"/index.php" parameters:params];
    AFJSONRequestOperation *operation = [AFJSONRequestOperation
JSONRequestOperationWithRequest:request success:^(NSURLRequest
*request, NSHTTPURLResponse *response, id JSON) {
        NSString *operatorCode = [JSON valueForKey:@"operatorCode"];
        NSString *operatorName = [JSON valueForKey:@"operatorName"];
        [[NSUserDefaults standardUserDefaults] setValue:operatorCode
forKey:@"operatorCode"];
        [[NSUserDefaults standardUserDefaults] setValue:operatorName
forKey:@"operatorName"];
        ActionTableViewController* vc = [ActionTableViewController
new];
        [self.navigationController pushViewController:vc
animated:YES];
    } failure:^(NSURLRequest *request, NSHTTPURLResponse *response,
NSError *error, id JSON) {
        if (error.code == -1004) {
            [[[UIAlertView alloc] initWithTitle:@"Сервер находится на

```

```

техническом обслуживании, попробуйте позже." message:nil
delegate:nil cancelButtonTitle:@"OK" otherButtonTitles:nil] show];
    }
    else
    {
        [[[UIAlertView alloc] initWithTitle:@"Проверьте Ваш логин и
пароль." message:nil delegate:nil cancelButtonTitle:@"OK"
otherButtonTitles:nil] show];
    }
    NSLog(@"[HTTPClient Error]: %@", error.localizedDescription);
}];
[operation start];
}

```

Лістинг А.6 — Інтерфейс вибору дії

```

(void)viewDidLoad
{
    [super viewDidLoad];
    UIImage* im = [UIImage imageNamed: IS_IPHONE_5 ?
@"background.jpg" : @"background5.jpg"];
    self.view.backgroundColor = [UIColor colorWithPatternImage:im];
    CGRect fullScreen = [UIScreen mainScreen].bounds;
    UIButton* takeButton = [UIButton
buttonWithType:UIButtonTypeCustom];
    takeButton.frame = CGRectMake(0.0, fullScreen.size.height*0.2,
150.0, 44.0);
    takeButton.center = CGPointMake(self.view.center.x,
takeButton.center.y);
    [takeButton setTitle:@"Поступления"
 forState:UIControlStateNormal];
    [takeButton addTarget:self action:@selector(buttonAction:)
 forStateEvents:UIControlEventTouchUpInside];
    takeButton.tag = GET_PRODUCT;
    [takeButton setWoodStyle];
    [self.view addSubview:takeButton];
    UIButton* giveButton = [UIButton
buttonWithType:UIButtonTypeCustom];
    giveButton.frame = CGRectMake(0.0,
CGRectGetMaxY(takeButton.frame)+ 40.0, 150.0, 44.0);
    giveButton.center = CGPointMake(self.view.center.x,
giveButton.center.y);
    [giveButton setTitle:@"Отгрузка"
 forState:UIControlStateNormal];
    [giveButton addTarget:self action:@selector(buttonAction:)
 forStateEvents:UIControlEventTouchUpInside];
    giveButton.tag = GIVE_PRODUCT;
    [giveButton setWoodStyle];
    [self.view addSubview:giveButton];
    UIButton* checkButton = [UIButton
buttonWithType:UIButtonTypeCustom];
    checkButton.frame = CGRectMake(0.0,
CGRectGetMaxY(giveButton.frame)+ 40.0, 150.0, 44.0);
    checkButton.center = CGPointMake(self.view.center.x,

```

```

checkButton.center.y);
    [checkButton setTitle:@"Проверить код"
forState:UIControlStateNormal];
    [checkButton addTarget:self action:@selector(buttonAction:)
forControlEvents:UIControlEventTouchUpInside];
    checkButton.tag = CHECK_QR;
    [checkButton setWoodStyle];
    [self.view addSubview:checkButton];
    UIButton* logoutButton = [UIButton
buttonWithType:UIButtonTypeCustom];
    logoutButton.frame = CGRectMake(0.0,
CGRectGetMaxY(checkButton.frame)+ 40.0, 150.0, 44.0);
    logoutButton.center = CGPointMake(self.view.center.x,
logoutButton.center.y);
    [logoutButton setTitle:@"Выход"
forState:UIControlStateNormal];
    [logoutButton addTarget:self action:@selector(buttonAction:)
forControlEvents:UIControlEventTouchUpInside];
    logoutButton.tag = LOGOUT;
    [logoutButton setWoodStyle];
    [self.view addSubview:logoutButton];
}

```

Лістинг А.7 — Створення PDF-документу

```

(void)printQRcodes
{
    int qrcodeImageDimension = 190;
    double pageWidth = 612.0;
    double pageHeight = 792.0;

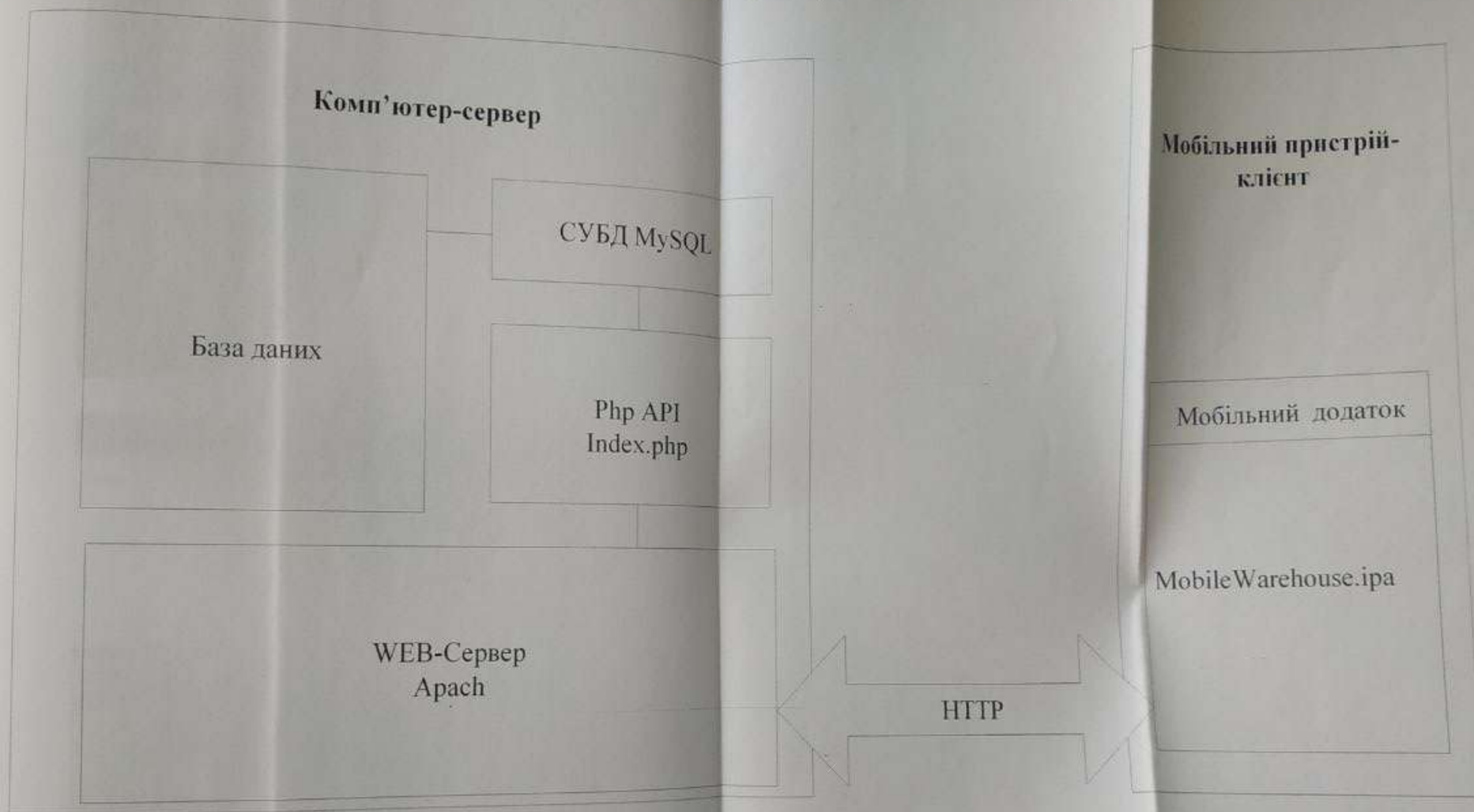
    NSMutableData *pdfFile = [[NSMutableData alloc] init];
    CGDataConsumerRef pdfConsumer =
    CGDataConsumerCreateWithCFData((__bridge
CFMutableDataRef)pdfFile);
    // The page size matches the image, no white borders.
    CGRect mediaBox = CGRectMake(0, 0, pageWidth, pageHeight);
    CGContextRef pdfContext = CGPDFContextCreate(pdfConsumer,
&mediaBox, NULL);
    CGContextBeginPage(pdfContext, &mediaBox);
    CGContextTranslateCTM(pdfContext, 0.0, mediaBox.size.height);
    CGContextScaleCTM(pdfContext, 1.0, -1.0);

    [_item.products enumerateObjectsUsingBlock:^(Product* item,
NSUInteger idx, BOOL *stop) {
        DataMatrix* qrMatrix = [QREncoder
encodeWithECLevel:QR_ECLEVEL_AUTO version:QR_VERSION_AUTO
string:[item.onlineIdentifier stringValue]];
        UIImage* qrcodeImage = [QREncoder renderDataMatrix:qrMatrix
imageDimension:qrcodeImageDimension];
        CGRect imageRect = CGRectMake((idx%3)*(qrcodeImageDimension +
14.0), (idx/3)*(qrcodeImageDimension + 14.0), qrcodeImageDimension,
qrcodeImageDimension);
    }
}

```

```
        CGContextDrawImage(pdfContext, imageRect, [qrcodeImage
CGImage]);
    }];
    CGContextEndPage(pdfContext);
    CGContextRelease(pdfContext);
    CGDataConsumerRelease(pdfConsumer);
    [self sendEmailWithPdfName:@"qrCodes.PDF" file:pdfFile];
}
```


Архітектура мобільної системи складського обліку ТОВ «АрСог»



						13.02070849.00017 Пл 1				
						Архітектура мобільної системи складського обліку ТОВ «АрСог»				
Зм.	Арх.	№ докум.	Підп.	Дата	Літ.				Маса	Масштаб
Розроб.	Авер'янов І.С.									
Перевір.	Кудерметов Р.К.									
Т.контр.										
Н.контр.	Щербак Н.В.					Арк. 1 Аркушів 5				
Затв.	Кудерметов Р.К.									
						НУ «Запорізька політехніка», КНТз-511м				

Інтерфейси мобільної системи складського обліку ТОВ «АрСог»

Login:

Paniserget

Password:

.....

Verify

Back

Owner: AppleStore

Отправка

Q W E R T Y U I O P

A S D F G H J K L

↑ Z X C V B N M ↓

123

space

Done

Login:

Сервер находится на техническом обслуживании, попробуйте позже.

OK

Verify



Поступления

Отгрузка

Проверить код

Выход

- Мобильные телефоны
- Аккумуляторы к телефонам
- Гарнитуры
- Мониторы
- Планшеты
- Мыши
- Ноутбуки и нетбуки
- Аксессуары к мобильным...
- Аксессуары к компьютерам
- Клавиатуры

Back

Жен

Содержит 1 товаров

Гепард

Содержит 2 товаров

Техноплюс

Содержит 2 товаров

Линекон

Содержит 1 товаров

Back

Ноутбуки и нетбуки

White

Quantity: 3

Add

13.02070849.00017 Пл 2

Зм	Арк.	№ докум.	Підп.	Дата
Розроб.	Дев'янов І.С.			
Перевір.	Кудерметов Р.К.			
Т.контр.				
Н.контр.	Щербак Н.В.			
Затв.	Кудерметов Р.К.			

Інтерфейси мобільної системи складського обліку ТОВ «АрСог»

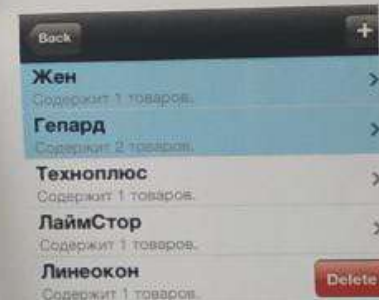
Літ.	Маса	Масштаб
Арк. 1	Аркуші 5	
НУ «Запорізька політехніка», КНТз-511		

Інтерфейси мобільної системи складського обліку ТОВ «АрСог»



Копия/Скрытая копия:

Тема: AppleStore QR codes



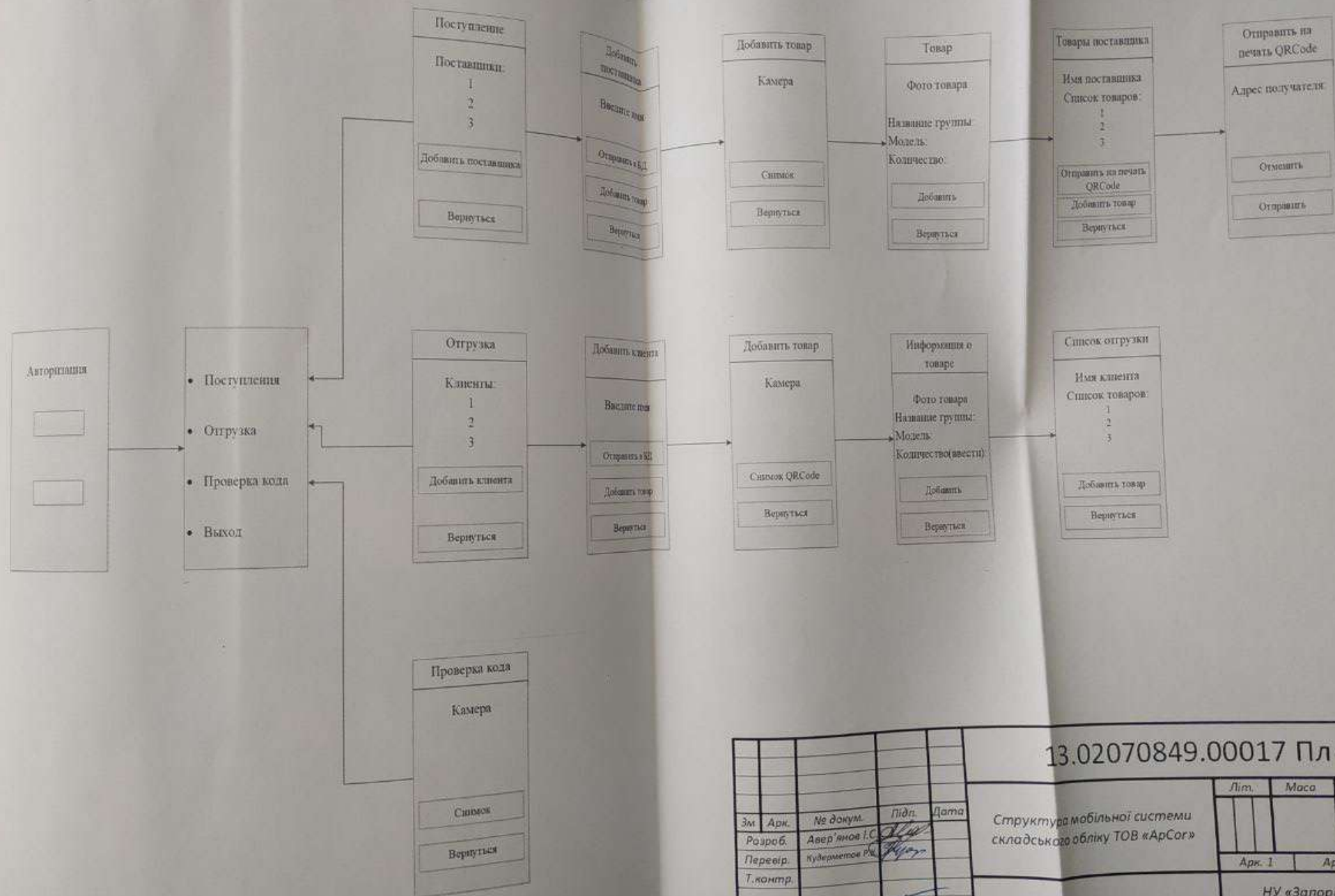
13.02070849.00017 Пл 3

Зм.	Арк.	№ докум.	Підп.	Дата
Розроб.		Авер'янов І.С.		
Перевір.		Кудерметов Р.А.		
Т.контр.				
Н.контр.		Щербак Н.В.		
Затв.		Кудерметов Р.А.		

Інтерфейси мобільної системи складського обліку ТОВ «АрСог»

Літ.	Маса	Масштаб
Арк. 1	Аркуші 5	
НУ «Запорізька політехніка», КНТз-511м		

Структура мобільної системи складського обліку ТОВ «АрСог»



13.02070849.00017 Пл 4

						13.02070849.00017 Пл 4		
</								

Структура мобільної системи складського обліку ТОВ «АрСог»