

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ
федеральное государственное бюджетное образовательное учреждение высшего образования
«Тольяттинский государственный университет»

Институт Математики, физики и информационных технологий
(наименование института полностью)

Кафедра «Прикладная математика и информатика»
(наименование)

09.03.03 «Прикладная информатика»
(код и наименование направления подготовки, специальности)

Бизнес-информатика
(направленность (профиль)/специализация)

ВЫПУСКНАЯ КВАЛИФИКАЦИОННАЯ РАБОТА (БАКАЛАВРСКАЯ РАБОТА)

на тему «Разработка программного комплекса для учета данных в
продуктовых автоматах»

Обучающийся

А.В. Коновалов
(Инициалы Фамилия)

(личная подпись)

Руководитель

Н.Н. Казаченок
(ученая степень, звание, Инициалы Фамилия)

Тольятти 2022

Аннотация

Бакалаврская работа посвящена разработке программного комплекса для автоматизации учета вендинг-бизнеса с использованием мобильных устройств и веб-приложения.

Актуальность работы заключается в том, что программный комплекс позволяет избавиться от рутинности бумажного процесса, оперативно получить статистическую информацию об обороте товаров и сэкономить время и силы персонала по обслуживанию торговых автоматов.

Благодаря проведению анализа существующих систем было выявлено, что программных комплексов, выполняющих поставленную задачу не так много, и они имеют значительные недостатки. Соответственно создание нового комплекса является актуальной задачей.

Целью бакалаврской работы является создание комплекса, автоматизирующего процесс ввода данных персоналом с использованием мобильного приложения и работающего с единой системой удаленного хранения.

Программный комплекс включает в себя веб-сервер, сервер базы данных, мобильное приложение под OS Android для персонала и веб-сайт для администратора либо владельца бизнеса. В ВКР подробно рассматривается анализ, моделирование и разработка каждой части данной системы. При создании программного комплекса использовались технологии разработки: Java, PHP, JavaScript, JQuery, AJAX, HTML, CSS, SQL.

Система была испытана в реальных условиях, подтвердила возможность автоматизации учета и получила положительные отзывы.

Бакалаврская работа представлена на 49 страницах, включает 15 иллюстраций, 12 таблиц, 7 приложений и список используемой литературы, состоящий из 28 источников.

Оглавление

Введение	4
Глава 1 Программный комплекс для автоматизации учета вендинг-бизнеса ..	6
1.1 Концептуальное моделирование предметной области	6
1.2 Обсуждение функционала и структуры программного комплекса ..	8
1.3 Сравнительный анализ существующих систем	12
Глава 2 Проектирование и реализация программного комплекса	18
2.1 Моделирование и разработка базы данных	18
2.2 Разработка серверной части	24
2.3 Разработка клиентской части на выбранной платформе	29
Глава 3 Тестирование созданной системы и расчет ее экономической эффективности	34
3.1 Тестирование и разбор недочетов, выявленных по его завершению	34
3.2 Определение экономической эффективности проектных мероприятий	34
Заключение	39
Список используемой литературы	40
Приложение А Листинг создания таблиц на языке SQL	43
Приложение Б Листинг файла mobile.php	46
Приложение В Листинг файла site.php	47
Приложение Г Примеры запросов	48
Приложение Д Скриншоты сайта	50
Приложение Е XML-формат для сохранения данных пакетов в мобильном приложении	53
Приложение Ж Скриншоты приложения	54

Введение

В современном мире автоматизация является важной частью торговли. Повсюду начинают появляться вендинг-автоматы [10], которые продают товары через специальное оборудование с большим количеством опций, оснащенное устройством для купюр и кассовым аппаратом. Во всем мире это уже признано высокотехнологичным и прибыльным бизнесом, но в России вендинговый бизнес находится только в начале своего развития. Особенно сильно вендинг привлекает начинающих бизнесменов и предпринимателей, так как требует его осуществление возможно при наличии минимального состава персонала, небольших вложениях капитала и времени.

Вендинговый бизнес способен приносить своему владельцу высокий и стабильный доход при правильной организации всех процессов, в том числе и качественном учете оборота товаров. Необходимо создать такой порядок организации, при котором было бы легко получать оперативную информацию о наличии товара на складе или в каждом торговом автомате.

Предложенный в данной выпускной квалификационной работе вариант решения поставленной проблемы заключается в создании системы учета через разработку программного комплекса.

Актуальность работы заключается в том, что программный комплекс позволяет избавиться от рутинности бумажного процесса, оперативно получить статистическую информацию об обороте товаров и сэкономить время и силы персонала по обслуживанию торговых автоматов.

Объектом исследования являются бизнес-процессы ввода и контроля данных оборота товаров в вендинговых автоматах

Предметом исследования является автоматизация исследуемых бизнес-процессов.

Целью бакалаврской работы является создание комплекса, автоматизирующего процесс ввода данных персоналом с использованием

мобильного приложения и работающего с единой системой удаленного хранения.

Для достижения цели были обозначены следующие задачи:

- осуществить постановку задачи, выделить требования к программному комплексу,
- провести анализ существующих систем,
- изучить современные средства для разработки мобильного приложения, веб-сайта и серверной части программного комплекса,
- определить модель программного комплекса и разработать его,
- определить экономическую эффективность проекта,
- провести тестирование комплекса и выявить направления дальнейшего развития функционала системы.

Структура работы включает в себя введение, три главы, заключение, приложение, список литературы.

В первой главе бакалаврской работы определяется общий функционал и структура программного комплекса. Проведено моделирование будущего веб-сайта и разработан порядок работы в системе. В главе так же приведен сравнительный анализ существующих систем и их оценка

Во второй главе проведены проектирование и реализация программного комплекса. Смоделирована база данных. Выделены основные сущности базы данных, рассмотрены их отношения, обозначены необходимые атрибуты и проведен выбор СУБД. Рассмотрено взаимодействие клиентов и сервера, подробно описаны планируемые возможности каждого класса. Разработана серверная часть. Проведен выбор платформы и разработка клиентской части. Расписаны разделы, которые будут содержать программный комплекс.

В третьей главе проведено тестирования и разбор ошибок, которое оно выявило. Так же в этой главе проведено определение экономической эффективности описанных выше проектных мероприятий.

Глава 1 Программный комплекс для автоматизации учета вендинг-бизнеса

1.1 Концептуальное моделирование предметной области

Для создания концептуальной модели предметной области необходимо провести структуризацию предметной области, для которой проектируется программный комплекс. Для решения данной задачи применим методологии семейства IDEF, а именно частную методологию, основанную на графическом представлении систем, IDEF0. Данный язык можно генерировать с помощью программы Ramus Educational.

Рассмотрим деятельность предпринимателя, владеющего одним и более торговыми автоматами. У него в подчинении имеются операторы, отмечающие количество проданных товаров, и пополняющие автомат новой продукцией. Оператор подсчитывает количество продукции в автомате и вносит информацию на бумажный или электронный носитель. Администратор (им может являться и сам владелец торговым автоматом) делает запрос оператору об остатках продукции в автомате, чтобы, исходя из полученных данных, сделать новую закупку продукции.

После определения деятельности сотрудников необходимо составить контекстную диаграмму «как есть» (as is). Диаграмма представлена на рисунке 1.

Для более детального раскрытия процесса составим диаграмму декомпозиции, представленную на рисунке 2. Исходя из этой диаграммы, выявим основные недостатки существующей системы. Процесс подсчета товаров и необходимость переносить данные с бумажного или электронного носителей на компьютер для дальнейшего хранения усложняют процесс получения актуальных данных администратором. На рисунке 3 приведем модель «как должно быть», применив автоматизацию, с помощью программного комплекса.



Рисунок 1 – Контекстная диаграмма процесса «Учет товаров в торговом автомате», модель «как есть»

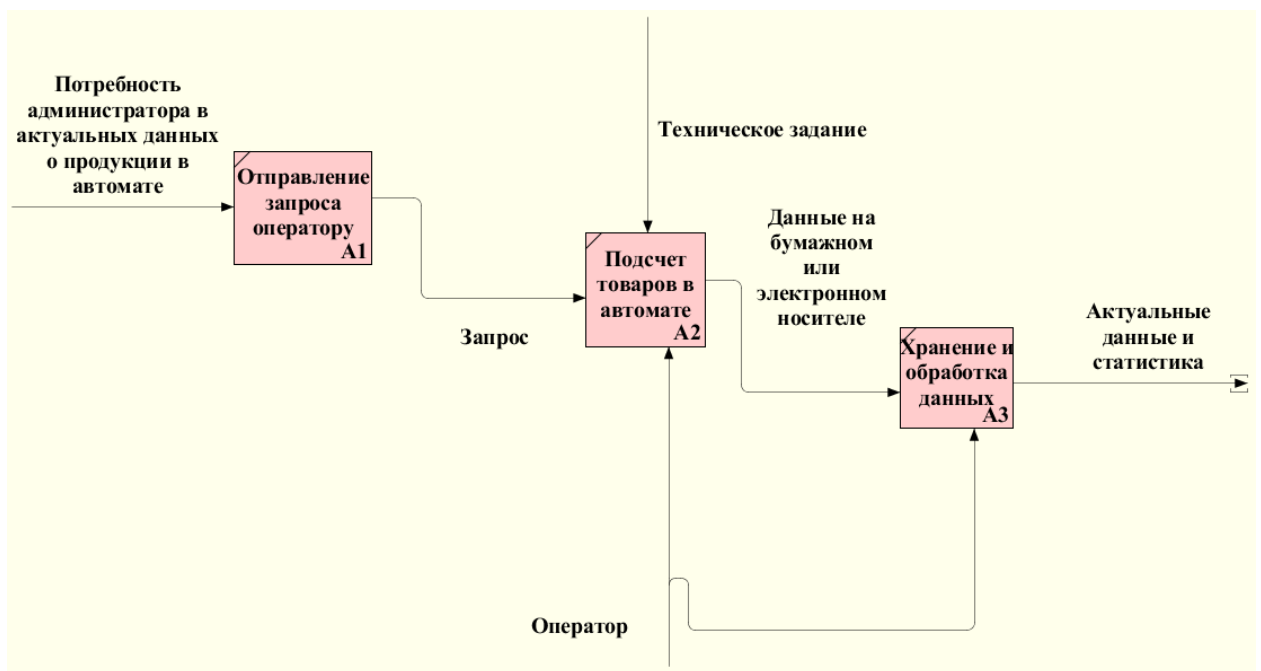


Рисунок 2 – Декомпозиция контекстной диаграммы процесса «Учет товаров в торговом автомате», модель «как есть»

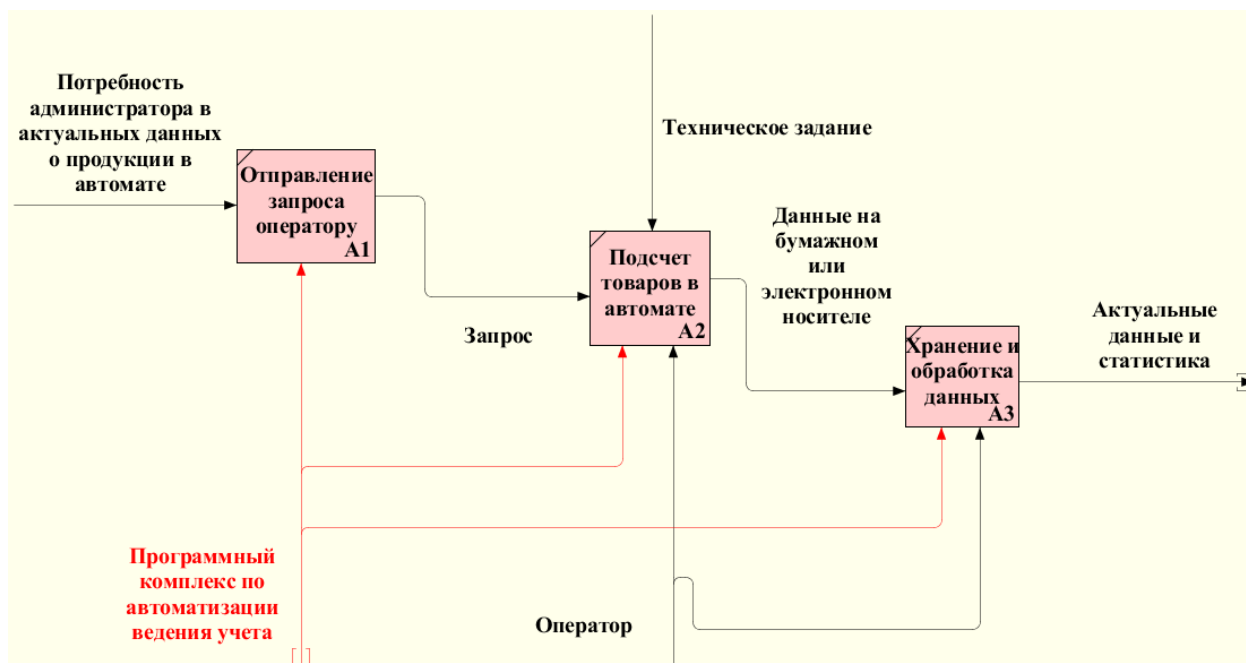


Рисунок 3 – Декомпозиция контекстной диаграммы процесса «Учет товаров в торговом автомате», модель «как должно быть»

Таким образом, применение программного комплекса позволит администратору в любое время мгновенно получить актуальные данные без дополнительного привлечения оператора. Так же это упростит работу оператора по подсчету товара и внесению данных, так как исчезнет необходимость вносить данные несколько раз.

1.2 Обсуждение функционала и структуры программного комплекса

Вендинговый автомат – это удобный способ покупки товаров, но для его обслуживания нужен человек (оператор). Оператор отмечает количество проданных товаров, пополняет автомат необходимым количеством товарных единиц, устанавливает и изменяет цены в настройках автомата.

Работу оператора можно автоматизировать, используя специальное мобильное приложение. Такое приложение позволит сохранять данные о количестве проданных товаров, информировать оператора об изменении

цены на товар, а также о необходимости изъятия или добавления нового товара в автомат.

Но если данные хранятся на мобильном устройстве, то возникает проблема переноса информации на компьютер. К тому же, нужно уведомлять оператора о ее изменениях как можно скорее. Также может появиться новый оператор, может смениться мобильное устройство, один автомат могут обслуживать несколько операторов. Все это приводит к мысли, что лучше обеспечить работу мобильного приложения с единым удаленным хранилищем. В этой роли мог бы выступать сервер, а синхронизация данных осуществляться по сети Интернет. Так как сервер с информационной базой системы постоянно доступен через Интернет, то управление последней можно сделать с помощью веб-сайта.

Доступ к данным одновременно будут иметь несколько групп пользователей. Среди них: администратор для управления системой через веб-сайт, и операторы, фиксирующие данные и получающие уведомления через мобильное приложение. То есть нужно продумать систему безопасной авторизации и разграничение прав пользователей.

Порядок работы в системе будет следующим:

- администратор системы авторизуется на веб-сайте и вносит первоначальные данные об автоматах, товарах и операторах;
- оператор системы авторизуется в мобильном приложении и получает уведомления о необходимых действиях к исполнению;
- оператор системы собирает данные о товарных единицах автомата, заполняет автомат и отправляет полученные данные на сервер;
- администратор системы авторизуется на веб-сайте, просматривает статистику, и управляет учетом товаров и цен.

Определив варианты использования программного комплекса администратором и операторами, построим логическую модель системы, используя нотацию языка UML. Получившаяся в результате диаграмма представлена на рисунке 4.

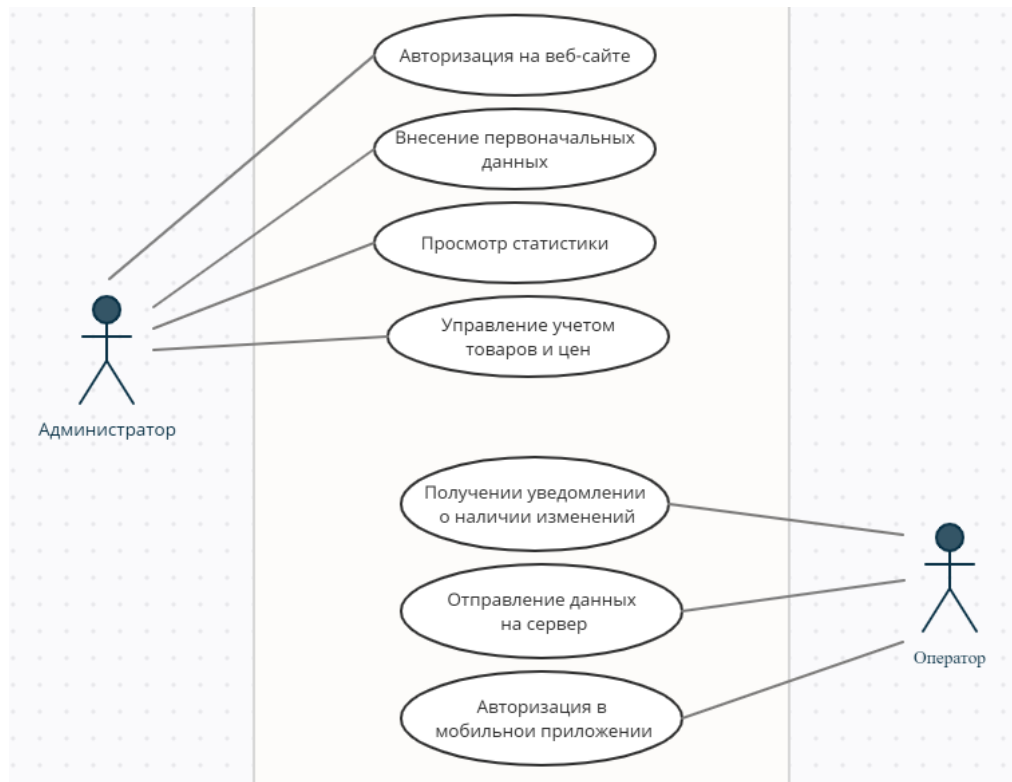


Рисунок 4 - Диаграмма вариантов использования проектируемого комплекса

Таким образом система представляет собой программный комплекс, в который входит веб-сайт для обеспечения работы администратора, мобильное приложение для оператора, и сервер в роли удаленного хранилища и связующего звена между ними [18].

Мобильное приложение будет иметь несколько экранов интерфейса:

- окно авторизации,
- список автоматов,
- список товаров,
- экран выбора количества товарных единиц.

Отправляться данные будут при нажатии на кнопку, если есть Интернет-соединение, иначе сохраняться в памяти устройства для отправки при следующей попытке.

Веб-сайт будет иметь страницы:

- авторизация,
- статистика,
- информация об автоматах,
- информация о товарах,
- информация об операторах.

Определив функционал комплекса, составим его структурную схему на рисунке 5.

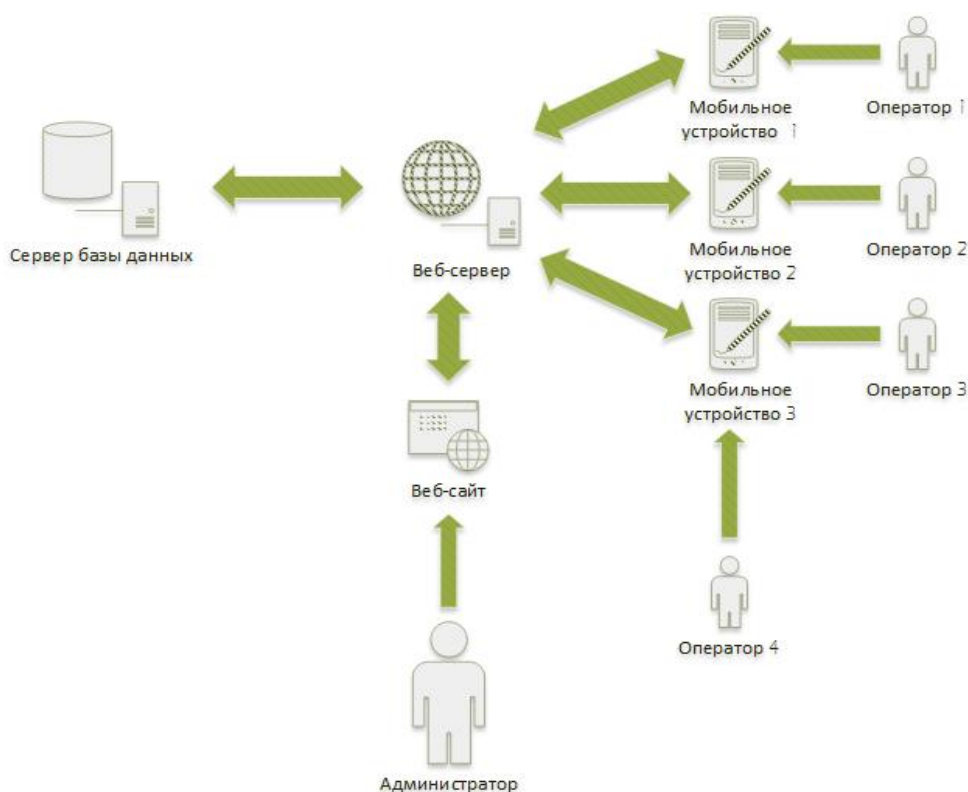


Рисунок 5 – Структурная схема комплекса

Все пользователи взаимодействуют с системой через веб-сервер. Это говорит о том, что он должен иметь единый программный интерфейс (API), позволяющий работать как с мобильным приложениям, так и с веб-сайтом.

1.3 Сравнительный анализ существующих систем

В настоящее время существует несколько программных комплексов, которые способны справиться с поставленной задачей. У каждой из этих систем есть свои преимущества и недостатки. Рассмотрим приложения, которые являются самыми популярными среди владельцев вендингового бизнеса на данный момент времени, и проведем их сравнительный анализ.

Приложение «KiT Vending» [21] позволяет полностью контролировать работу торговых автоматов. Интерфейс данного мобильного приложения приведен на рисунке 6. В качестве примера взят раздел, содержащий данные о деталях загрузки автомата с горячими напитками.

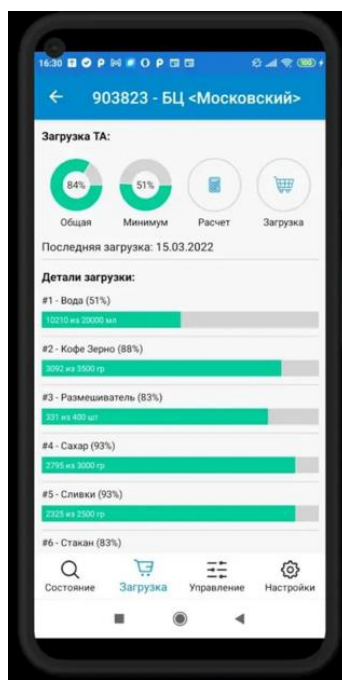


Рисунок 6 – Пример интерфейса приложения «KiT Vending»

Возможности, которые предоставляет данное приложение:

- отслеживание количества денежных средств в автомате;
- контроль количества обслуживаний и загрузок;
- формирование различных отчетов;

- управление настройками автоматов.

Однако, не смотря на такую широкую многофункциональность, при более детальном изучении приложения были выявлены следующие недостатки:

- фильтры приложения не позволяют просматривать количество проданного товара по типу, что значительно затрудняет проведение анализа по оставшейся продукции в автомате;
- нет возможности просматривать детализацию продаж в одном конкретном автомате за определенный период времени, так как все продажи сортируются только по дням;
- отсутствует поиск автомата в мониторинге, соответственно приходится пролистывать весь список автоматов, чтобы найти нужный (особенно это может быть неудобно пользователям, у которых во владение очень большое количество торговых автоматов);
- не является полноценным программным комплексом, а только мобильным приложением.

Далее рассмотрим мобильный комплекс «Telemetron» [22], который позволяет проводить мониторинг и обслуживание вендинговых автоматов. Интерфейс данного мобильного приложения приведен на рисунке 7. В качестве примера взят раздел, содержащий статистику о всех торговых автоматах владельца.

Приложение позволяет:

- вести контроль продаж за определенный промежуток времени;
- просматривать последние события на автомате;
- контролировать количество наличных в автомате;
- вести учет с помощью приложения «Telemetron Teleport».



Рисунок 7 – Пример интерфейса приложения «Telemetron Teleport»

Какие недостатки были выявлены при более детальном анализе данного комплекса:

- в приложении «Telemetron Teleport» отображается не вся статистика и информация по торговым автоматам, в отличие от веб-версии, соответственно приложением пользоваться менее удобно и информативно;
- высокая цена.

Одной из наиболее популярных систем управления вендингом является система «NayaxVend»[23]. Данное программное обеспечение позволяет удаленно проводить управление и мониторинг вендинговыми автоматами. Интерфейс данного мобильного приложения приведен на рисунке 8. В качестве примера взята приборная панель приложения.

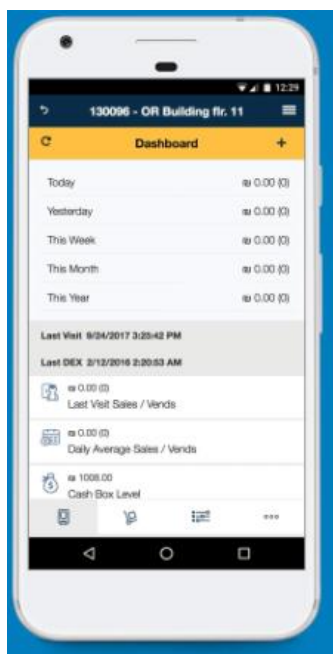


Рисунок 8 – Пример интерфейса приложения «MoMa»

Какими программными функциями оснащен «NayaхVend»:

- система оповещений о неблагоприятных событиях или сценариях;
- удобная отчетность;
- удаленное управление автоматами, в том числе проверка остатков продукции;
- мобильное приложение «MoMa», синхронизирующееся с веб-сайтом.

При всех этих значительных плюсах данное программное обеспечение оказывается не очень полезным для начинающих предпринимателей, так как имеет высокий ценник.

«Vendon»[24] - облачная система управления вендингом. Разработчики обещают, что данный комплекс позволит просматривать состояние автоматов в режиме реального времени, получать отчетность для анализа данных об автоматах и автоматическое обновление данных о необходимости пополнения запасов продукции. У системы также есть удобное мобильное приложение. Главной проблемой «Vendon» является сложный интерфейс,

начинающему предпринимателю тяжело разобраться в управлении данным комплексом. Пример интерфейса «Vendon» приведен на рисунке 9.



Рисунок 9 – Пример интерфейса программного комплекса «Vendon»

Каждый из этих комплексов был оценен по нескольким критериям. Данные анализа приведены в таблице 1.

Таблица 1 – Сравнительный анализ аналогов

Программный комплекс	«KiT Vending»	«Telemetron»	«NayaxVend»	«Vendon»
Наличие мобильного приложения	+	+	+	+
Наличие веб-сервиса	-	+	+	+
Работа без интернет соединения	-	-	+	+
Понятный и удобный интерфейс	+	+	+	-
Ценовая доступность	+	+	-	+

Проведя детальный анализ популярных программных комплексов для автоматизации учета вендинг-бизнеса, можно сделать вывод, что в каждом из них имеются какие-нибудь существенные недостатки. Соответственно, разработка комплекса, решающего данную задачу, является актуальной.

Выводы по первой главе.

В первой главе работы проведено концептуальное моделирование предметной области, в ходе которого выявлены основные недостатки существующей системы. Было определено, каким образом можно повысить ее эффективность.

После этого составлены функционал и структура программного комплекса: определены основные страницы веб-сайта, экраны интерфейса мобильного приложения и порядок работы в системе для администратора и операторов.

В ходе проведения анализа существующих систем были выявлены преимущества и недостатки в программных комплексах или мобильных приложениях конкурентов.

Глава 2 Проектирование и реализация программного комплекса

2.1 Моделирование и разработка базы данных

При проектировании базы данных, будем использовать семантический метод моделирования [14], который представляет собой моделирование структуры данных, основанное на смысле самой информации. В качестве инструмента семантического моделирования используем диаграмму сущность-связь (ER - Entity-Relationship [27]).

Исходя из вышеописанного функционала, в качестве основных сущностей обозначим такие как:

- пользователь,
- автомат,
- товар,
- цена закупки,
- цена продажи,
- загруженный товар.

Рассмотрим отношения сущностей на рисунке 10.

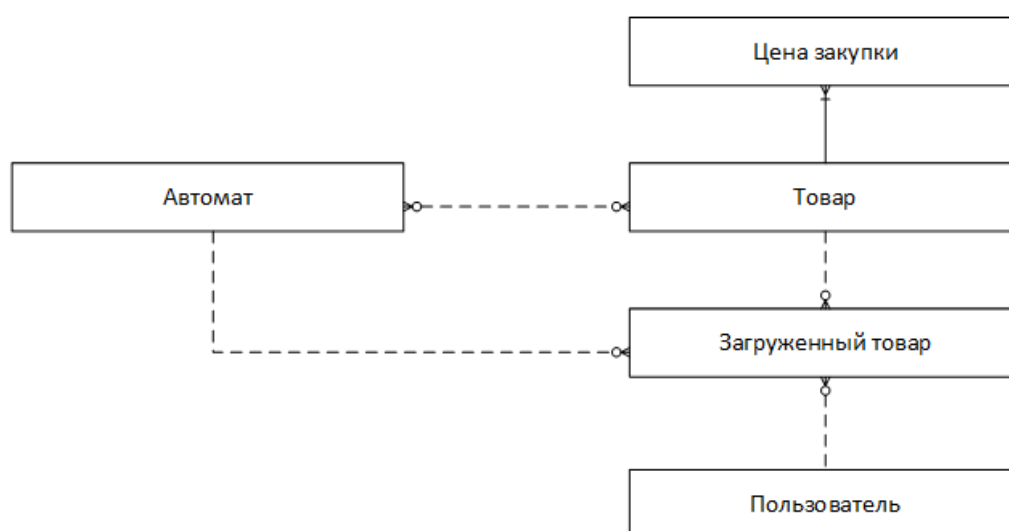


Рисунок 10 – Диаграмма взаимосвязи сущностей

Рисунок 9 содержит связь «многие ко многим» между сущностями «автомат» и «товар», которая допускается лишь на начальном этапе проектирования базы данных, но должна быть преобразована к типу «один ко многим». Проведя такое преобразование, получим следующую диаграмму, представленную на рисунке 11.

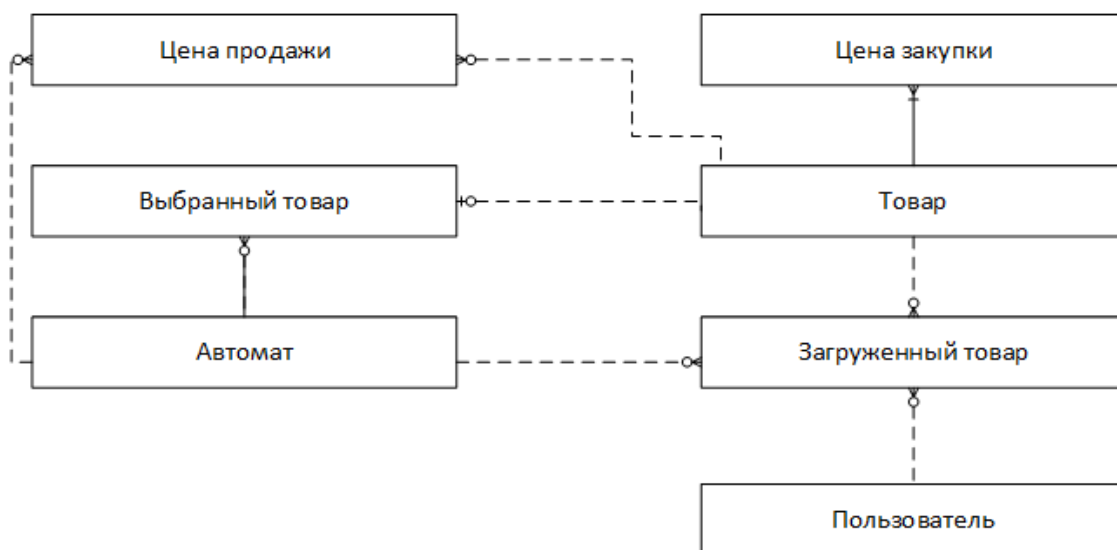


Рисунок 11 – Преобразованная диаграмма взаимосвязи сущностей

На этой диаграмме автомат с товаром связываются через сущности «цена продажи», «выбранный товар». Добавив эти сущности, получили четыре связи «один ко многим» вместо одной «многие ко многим».

Проанализировав сущности диаграммы, обозначим для каждой определяющие ее атрибуты. Для организации связей будем использовать специальные атрибуты - первичные ключи (id - primary key) и внешние ключи (automate_id, product_id, user_id - foreign key), приведенные на рисунке 12.

Так как в работе в дальнейшем будет использована для хранения данных реляционная БД [11], то каждый класс сущности будет представляться таблицей, в которой все поля, соответствующие атрибутам, будут иметь заданный тип. Рассмотрим таблицу 2.

Таблица 2 – Типы полей таблицы пользователей

Тип поля	Поле
INT	Id
TEXT	Fio
VARCHAR(20)	Login
VARCHAR(40)	Password
BYTE	Type
BYTE	Status

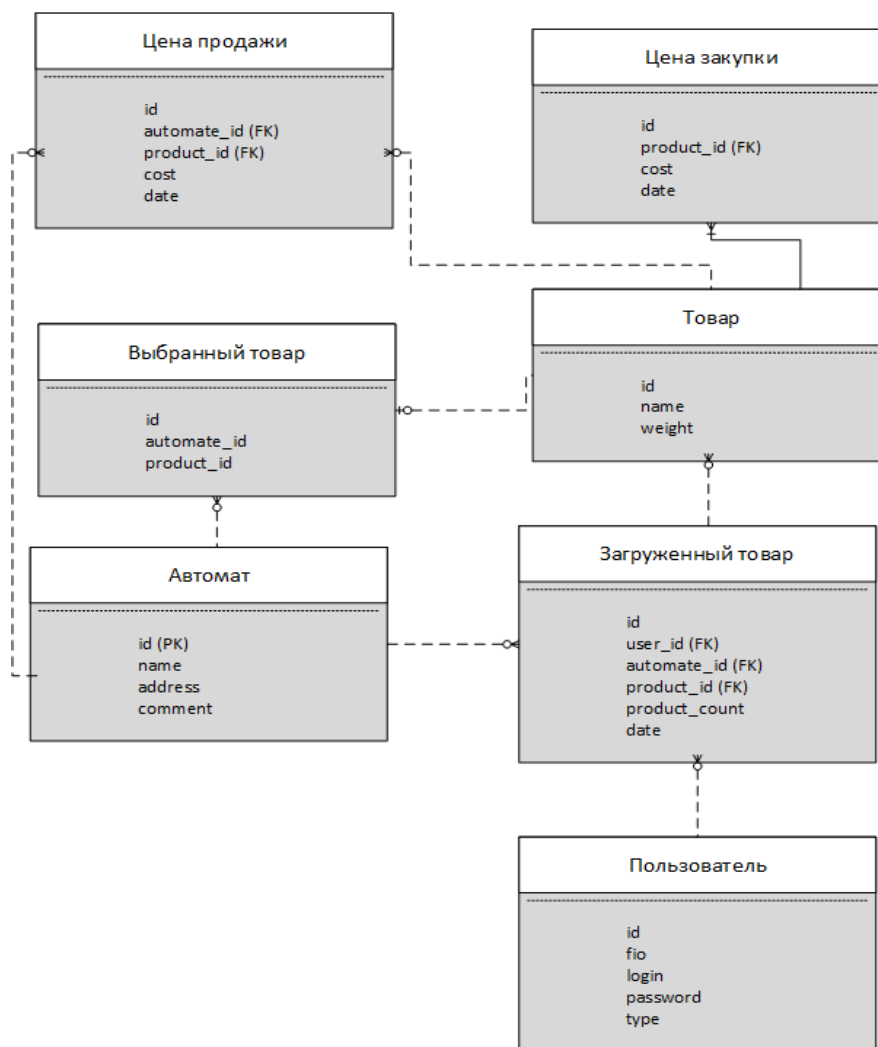


Рисунок 12 – Атрибуты сущностей

Поле type используем для разграничения прав пользователей, значение 0 будет соответствовать оператору, 1 - администратору. Поле status необходимо для обозначения состояния (активен/удален - соответствует 0/1), так как после удаления пользователя должна остаться информация для ведения статистики. Конечно, в полях type и status можно использовать тип boolean, но укажем byte для возможности дальнейшего расширения. В таблицах 3 и 4 поле status будет нести аналогичную функцию.

Таблица 3 – Типы полей таблицы автоматов

Тип поля	Поле
INT	Id
VARCHAR(20)	Name
TEXT	Address
TEXT	Comment
BYTE	Status

Таблица 4 – Типы полей таблицы товаров

Тип поля	Поле
INT	Id
VARCHAR(40)	Name
FLOAT	Weight
BYTE	Status

В связи с тем, что закупочная цена может измениться в течение времени, будем каждый раз добавлять новую запись в таблицу с указанием

даты, благодаря которой сможем легко выбрать цены на нужный период времени. Для этих целей создадим таблицу 5.

Таблица 5 – Типы полей таблицы закупочных цен

Тип поля	Поле
INT	Id
INT	product_id
FLOAT	Cost
DATETIME	Date

Таблица 6 аналогична таблице 5. Поле cost (стоимость) дает возможность установить независимую цену для каждого автомата.

Таблица 6 – Типы полей таблицы цен для продаж

Тип поля	Поле
FLOAT	Cost
DATETIME	Date
INT	Id
INT	automate_id
INT	product_id

Поля automate_id, product_id таблицы 7 и таблицы 6 нужны для реализации связи многие ко многим. Связь таблицы 6 с таблицей 3 показывает принадлежность товаров к автомату.

Таблица 7 – Типы полей таблицы текущих активных товаров в автомате

Тип поля	Поле
INT	Id
INT	automate_id
INT	product_id

Поле `product_count` в таблице 8 хранит количество пополненных продуктов в автомате, `date` - дату пополнения. Данные таблицы 8 будут участвовать в формировании статистики.

Таблица 8 – Типы полей таблицы, которая хранит присылаемую оператором информацию

Тип поля	Поле
INT	Id
INT	user_id
INT	automate_id
INT	product_id
INT	product_count
DATETIME	Date

После завершения этапа проектирования выберем СУБД. Создаваемая система будет использовать веб-сервер, также СУБД должна предоставляться многими хостинг-провайдерами, быть реляционной и прекрасно подходит для работы малых и средних систем. Поэтому остановимся на СУБД MySQL[5], которая имеет все эти возможности.

SQL запросы, создающий таблицы базы данных и администратора для начала работы системы представлен в приложении А.

2.2 Разработка серверной части

Сервер - ядро всей системы, потому что это связующее звено в передаче информации между клиентами и базой данных.

«Общение» клиентов с веб-сервером будет происходить через унифицированный интерфейс (API) [2]. Особенности каждого клиента повлияют на построение API. Поэтому ниже более подробно рассмотрим взаимодействие клиентов и сервера.

Мобильный клиент будет отправлять данные в формате, удобном, для обработки сервером. Таковыми являются популярные форматы XML и JSON. Последний имеет преимущество в сокращении объема передаваемых данных, поэтому в работе сделан выбор в пользу использования JSON. Отправляемый клиентом запрос должен содержать: имя, данные, логин и токен (подробно процедура авторизации будет реализована позже). Возвращаемый ответ сервера будет содержать токен и данные.

Клиент-сайт, использующий JavaScript и библиотеку JQuery, по умолчанию работает с форматом JSON (JavaScript Object Notation), который был специально разработан для JS. Отправляемый запрос должен содержать имя, данные. Возвращаемый ответ содержит данные.

Опираясь на общие части запросов, спроектируем API. Отличия запросов сайта от мобильного приложения незначительны, и заключаются лишь в передаче авторизационных данных и получении токена.

В целом при получении запроса необходимо преобразовать данные, обработать их, выполнить соответствующий sql запрос к базе данных (БД) и вернуть ответ.

Работа сервера чем-то похожа на работу офиса. Служба безопасности проверяет входящих людей и выдает пропуска. Секретарь встречает клиентов и общается с ними, передает инструкции далее в отделы (например оператору) для выполнения. Архивариус выдает запрошенные данные, если таковые требуются отделам. Организация работы системы подобно

вышеописанной схеме позволяет с легкостью, при необходимости, расширять внутренний функционал системы, не нарушая ее целостности.

Делегируем работу каждой из офисных служб соответствующим классам: AgentSecretar, AgentSecurity, AgentOperator и AgentDB. Эти классы будут унаследованы от класса Agent, имеющего общий механизм для работы с ошибками.

Алгоритм взаимодействия объектов на схеме, представленной на рисунке 13, следующий:

- секретарь получает тип клиента и его запрос (Приложение В), который преобразует из JSON-формата в ассоциативный массив;
- секретарь передает авторизационные данные службе безопасности, которая возвращает ответ о возможности продолжения работы с клиентом;
- если работа с клиентом возможна, секретарь передает данные оператору, иначе возвращает ошибку;
- оператор обрабатывает данные и посылает запрос архивариусу;
- архивариус формирует специальный запрос к базе данных и возвращает массив данных;
- оператор формирует и возвращает полученный от архивариуса ответ секретарю;
- секретарь преобразует полученные данные в JSON-формат и отправляет клиенту (Приложение В).

Теперь более подробно опишем планируемые возможности каждого класса. Секретарь будет получать данные от клиентов и возвращать им ответ. Весь процесс будет реализован с помощью системы API, в состав которой входят именованные команды для общения клиента с сервером:

- «product.add», «product.edit», «product.delete», «products.get», «product.addCost» - команды для работы с товаром;

- «operator.add», «operator.edit», «operator.delete», «operators.get», - команды для работы с оператором;
- «automate.add», «automate.edit», «automate.delete», «automates.get», - команды для работы с автоматом.

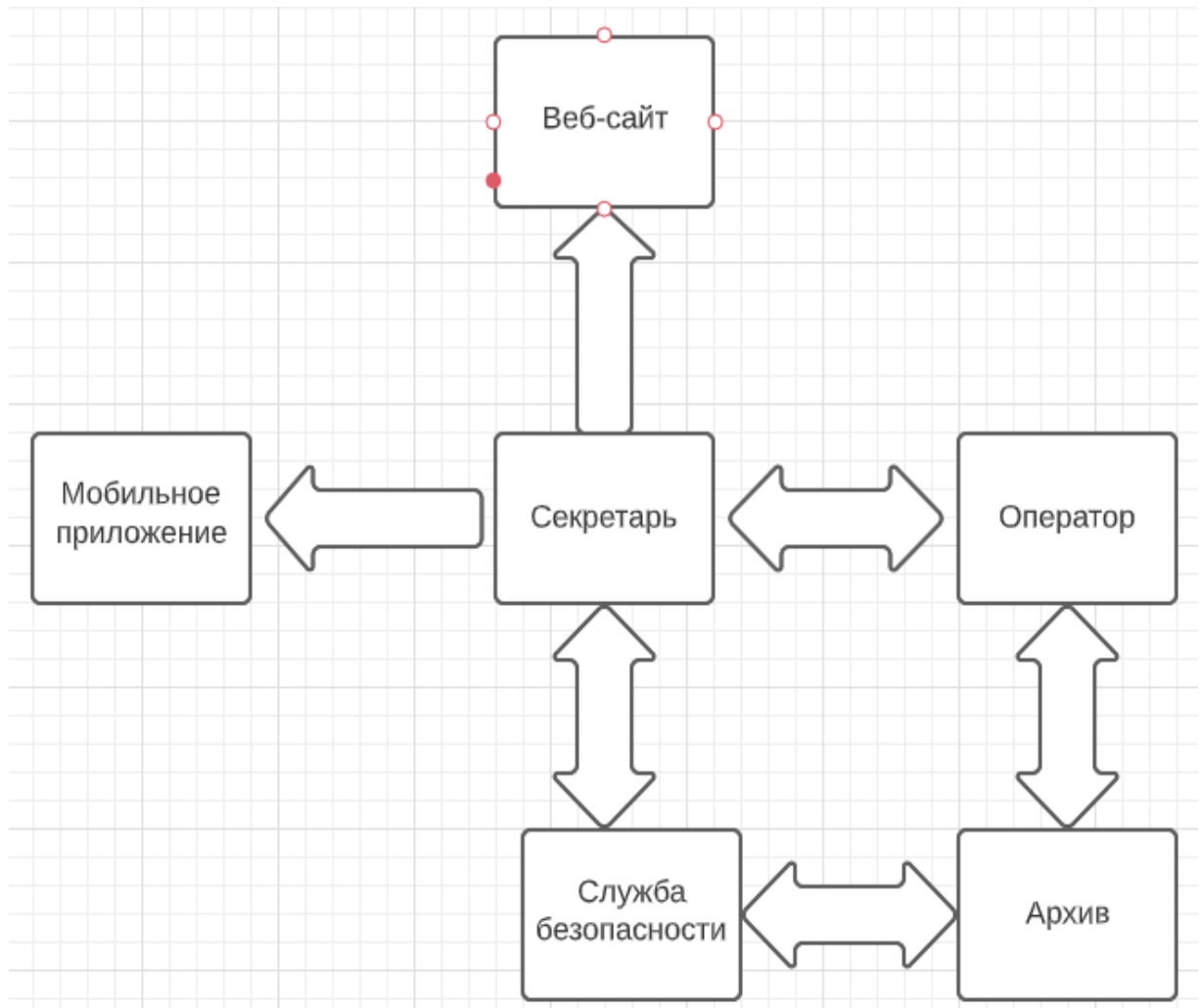


Рисунок 13 – Диаграмма взаимодействия объектов

Таким образом, это позволит в дальнейшем расширить возможности клиентов, дополняя API системы новыми командами.

Объект секретарь будет взаимодействовать только со службой безопасности и оператором. Обязанность секретаря - распознавать

полученные данные (согласно командам API), перенаправлять оператору и формировать ответ (методом `getAnswer`).

Служба безопасности взаимодействует с архивариусом, чтобы проверить переданные секретарем авторизационные данные от разных клиентов и возвращать так называемые пропуски (токены). В обязанности службы безопасности входит:

- проверять сайт клиента (`checkSiteLogin`);
- проверять авторизацию мобильного приложения (`checkLogin`, `checkPasscard`).

Оператор будет подготавливать данные для передачи архивариусу и формировать ответ для запрошенных секретарем данных. Оператор сможет:

- формировать список товаров (`getProducts`);
- добавлять товар (`addProduct`);
- удалять товар (`deleteProduct`);
- редактировать информацию о товаре (`editProduct`);
- устанавливать новую цену товару (`addCostOfProduct`);
- формировать список автоматов (`getAutomates`);
- добавлять автомат (`addAutomate`);
- удалять автомат (`deleteAutomate`);
- редактировать информацию о автомате (`editAutomate`);
- формировать список операторов (`getOperators`);
- добавлять оператора (`addOperator`);
- удалять оператора (`deleteOperator`);
- редактировать информацию о операторе (`editOperator`);
- получать статистику (`getStatistic`).

Возможности архивариуса будут заключаться лишь в поиске и возвращении информации оператору и службе безопасности из базы данных MySQL. Таким образом в этом классе инкапсулируются особенности работы через запросы на языке SQL с базой данных. Архивариус сможет:

- получать товар из БД (getProduct);
- добавлять товар в БД (addProduct);
- редактировать информацию о товаре (editProduct);
- устанавливать новую цену товару (addCostOfProduct);
- получать автомат из БД (getAutomate);
- добавлять автомат в БД (addAutomate);
- редактировать информацию о автомате (editAutomate);
- получать оператора из БД (getOperator);
- добавлять оператора в БД (addOperator);
- редактировать информацию о операторе (editOperator);
- формировать статистику (getStatistic).

Все описанные классы будут располагаться в отдельной библиотеке `vendbook.inc.php`, с которой клиенты будут взаимодействовать через точки входа в систему - скрипты `site.php` и `mobile.php` представленные в приложении Б. В данных скриптах будут создаваться экземпляры классов, связь между которыми происходит через конструкторы согласно схеме взаимодействия, представленной на рисунке 14.

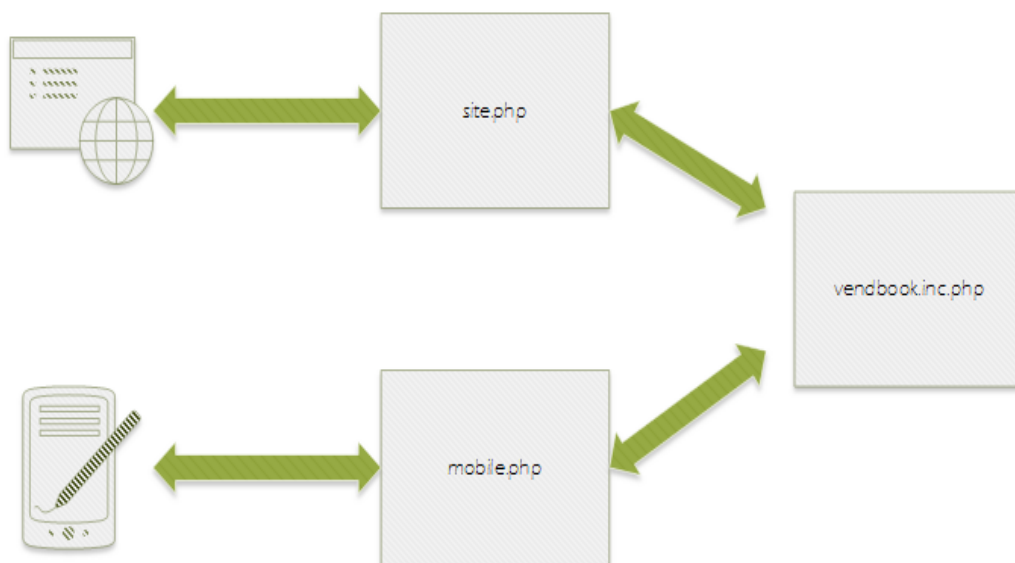


Рисунок 14 – Точки входа в систему для клиентов

В разработке веб-сервера необходимо применения языка программирования, который должен обладать следующими возможностями:

- поддерживаться многими хостинг-провайдерами;
- иметь возможность работы с СУБД MySQL;
- быть простым в освоении;
- быть независимым от платформы;
- быть бесплатным в использовании.

Под эти критерии подходит язык программирования PHP [5].

2.3 Разработка клиентской части на выбранной платформе

Как было описано выше, система будет иметь два типа клиентов - сайт и мобильное приложение. Рассмотрим их более подробно.

Клиентом для управления системой будет сайт, так как браузеры для работы с веб-платформой доступны на любой операционной системе.

В связи с обозначенным функционалом системы, сайт будет содержать разделы:

- авторизация (по логину и паролю, который будем хешировать алгоритмом sha1 для безопасности);
- статистика (просмотр статистики за выбранный период времени по прибыли, обороту каждого автомата, информации о количестве проданных товаров);
- автоматы (добавление, редактирование, удаление автомата, переход на товары автомата);
- все товары (добавление, редактирование, удаление товара, установление цены закупки на определенную дату);
- товары автомата (выбор товаров автомата из всех товаров, добавление цены продажи на определенную дату);

- операторы (добавление, редактирование, удаление оператора, смена пароля).

Для удобства использования, эти данные можно представить в табличной форме. Так как оформление всех страниц сайта будет в едином стиле, создадим шаблон, описанный на языке верстки веб-страниц HTML [12] и каскадных таблиц стилей CSS [26], в который будет входить меню и блок для содержимого. Данный шаблон представлен на рисунке 15.

меню	Логотип
Содержание	
Подвал	

Рисунок 15 – Модульная сетка сайта

Современные веб-сайты постепенно переходят в веб-приложения, что подразумевает динамичность и отзывчивость интерфейса, а также экономию трафика при переходе от страницы к странице. Для того чтобы это сделать, будем применять технологию AJAX (Asynchronous Javascript and XML)[17] и скриптовый язык JavaScript с библиотекой JQuery[6], которая выносит повторяющийся код из функций и позволяет не беспокоиться о совместимости в различных браузерах.

Разделы автоматов, товары и операторы будут иметь похожие функции, такие как добавление, редактирование и удаление объектов (Приложение Г). В связи с этим элементы интерфейса будут похожи.

Добавление и редактирование объекта будет происходить в диалоговых окнах. Также в разделе автоматы будет элемент навигации для перехода на выбранные товары данного автомата.

Для каждого раздела сайта создадим свой php-скрипт, который подгружает специальный JS-скрипт раздела и единый шаблон сайта. В функционал JS-скрипт будет входить асинхронная передача запросов серверу, формирование таблиц с данными, и, при добавлении и редактировании объекта - создание диалоговых окон и динамическое обновление содержимого таблицы при изменении данных.

Статистика будет формировать отчет за выбранный период времени по прибыли, обороту каждого автомата, информацию о количестве проданных товаров в приложении Г.

При выборе платформы мобильного приложения сначала нужно исследовать рынок. Примерные данные статистики показывают, что 70% рынка смартфонов в России занимает OS Android и 10% - iOS. Поэтому приложение для работы оператора сначала нужно создавать для OS Android[3].

Для разработки мобильного приложения под OS Android будем использовать язык программирования Java в IDE Eclipse, Android Studio[8][13][2].

В связи с обозначенным функционалом системы, мобильное приложение для оператора будет содержать окна:

- авторизация (по логину и паролю);
- список автоматов (с возможностью перехода на товары автомата);
- полный список товаров автомата (изменение количества определенного товара, уведомления для оператора);
- сводный список товаров автомата (для удобства подготовки к заполнению автомата и для отправки данных на сервер).

Для авторизации пользователю нужно ввести логин и пароль. Для реализации безопасности системы пароль будем передавать в зашифрованном виде, то есть создадим первичный токен:

- генерируем число от 10 до 99;
- хешируем пароль алгоритмом sha1 несколько раз;
- в определенные места хеша добавляем соль;
- получаем первичный токен.

На сервер будем отправлять логин и первичный токен.

После прохождения авторизации пользователь увидит список автоматов, по нажатию на которые перейдет на полный список товаров автомата. Этот список создадим с помощью сдвигающихся слайдов для удобства навигации, так как все товары не уместятся на одном экране. При нажатии на товар будет появляться окно с вариантами выбора количества, нужного для пополнения.

Сводный список товаров будет показывать выбранные товары, то есть количество которых больше нуля. Данная функция поможет оператору в подготовке товаров для пополнения автомата.

При отправке данных о количестве пополненных товаров на сервер, создадим диалоговое окно для выбора даты отправки, которая будет учитываться в статистике. Также следует предусмотреть случай, когда устройство оператора не имеет интернет-соединения в момент отправки. Для решения этой проблемы будем создавать офлайн-пакеты данных в XML формате (Приложение Д), которые можно будет отправить позже и чтобы можно было дальше продолжить обслуживание автомата. Также предусмотрим синхронизацию данных, при прохождении которой будем сверять полученную информацию с имеющейся и, при появлении новых данных, будем добавлять следующие уведомления:

- оператору необходимо добавить новый товар в автомат;
- оператор должен изменить цену на ценнике автомата;

– оператор должен изъять удаленный товар из автомата.

Скриншоты созданного мобильного приложения представлены в приложении Д, а исходный код, в связи с его большим объемом, в приложении на электронном носителе.

Выводы по второй главе.

Во второй главе проведены моделирование и разработка базы данных. Построена диаграмма взаимосвязи сущностей для дальнейшего определения первичных и внешних ключей базы данных.

Проведена разработка серверной части, в ходе которой подробно описано взаимодействие клиентов и сервера. Спроектирована API и описаны возможности каждого класса.

При создании клиентской части составлен шаблон веб-сайта и продуманы разделы, которые он содержит. Определены и описаны окна, которые содержит мобильное приложение. Разработана система для авторизации пользователя. Пошагово описана работа оператора с мобильным приложением.

Глава 3 Тестирование созданной системы и расчет ее экономической эффективности

3.1 Тестирование и разбор недочетов, выявленных по его завершению

Созданная система проходила тестирование работоспособности как в ходе процесса разработки, так и по его завершению, но при апробации программного комплекса выявились следующие недочеты:

а) в приложении:

- при создании анимации перелистывания списка товаров возникала ошибка исполнения;
- при генерировании случайного цвета для обозначения товара в списке обнаруживалось зависание программы;
- сгенерированный sha1-хеш приложения отличался от серверного sha1-хеша;
- на сайте:
- отсутствие контроля ввода данных пользователем, что приводило к созданию пустых записей в БД;

б) на сервере:

- ошибка в построении запроса при расчете статистики за выбранный период времени.

Данные ошибки программного комплекса были успешно устранены.

3.2 Определение экономической эффективности проектных мероприятий

Стоимость разработки программного комплекса напрямую зависит от сложности проекта и от количества его составляющих. В случае данного

комплекса стоимость трудозатрат на разработку мобильного приложения, сайта, веб-сервера и базы данных учитываться не будет. Соответственно, на стоимость готового продукта в большей степени будет влиять выбор платформ, через которые он будет размещаться, и составление механизма вовлечения аудитории.

После проведения окончательной отладки программного комплекса и финального тестирования необходимо сделать регистрацию будущего сайта и размещение приложения в необходимых магазинах мобильных приложений. Мобильное приложение в данном программном комплексе разработано для платформы Android. Android-приложения можно размещать для скачивания прямо с официального сайта программного комплекса, но большей вовлеченности аудитории можно добиться при распространении приложения через магазин мобильных приложений «Google Play».

После размещения сайта и приложения на всех необходимых платформах необходимо привлечение специалистов технической поддержки, чтобы клиент, воспользовавшись комплексом имел возможность сообщать о создавшихся проблемах в работе комплекса для дальнейшего устранения этих неполадок. Все перечисленные выше затраты описаны в таблице 9.

Таблица 9 – Наименование и стоимость затрат на разработку и внедрение

Наименование	Стоимость
Обслуживание веб-сервера	5 000 руб./мес.
Публикация приложения в «Google Play»	1500 руб.
Продвижение сайта и мобильного приложения для привлечения аудитории	50000 руб.

Таким образом, для реализации данного проекта необходимо потратить 111500 рублей инвестиционных вложений в первый год работы.

Теперь проведем расчет экономической эффективности проекта. Для этого необходимо использовать данные о затратах на реализацию проекта и прогноз продаж услуг программного комплекса.

Выручка будет складываться от продажи подписок на пользование данного комплекса. Будет реализовываться продажа двух видов подписок, описание которых приведено в таблице 10. Продажа подписок будет происходить через веб-сайт, после оплаты происходит регистрация пользователя. Используя свои логин и пароль клиент может зайти в личный кабинет в мобильном приложении после его скачивания. Это позволит избежать выплаты дополнительных налогов платформе «Google Play».

Таблица 10 – Разновидности подписок на пользование продуктом

Наименование подписки	Услуги, предоставляемые по подписке	Стоимость подписки в месяц за подключение одного автомата
Стандарт	Доступны все функции программного комплекса	150 руб.
Премиум	Доступны все функции программного комплекса и круглосуточная техническая поддержка	200 руб.

Сделаем предположение, что 60% пользователей данным программным комплексом захотят приобрести подписку «Стандарт», а оставшиеся 40% - «Премиум». Тогда проведя необходимые расчеты можно получить минимальное число покупок данных подписок в месяц для полного погашения затрат на разработку, поддержку и продвижение программного комплекса. Данные приведены в таблице 11.

Предположим, что рост приобретения подписок на пользование данным программным комплексом будет составлять 40%. Проведем расчет экономической эффективности проекта на ближайшие 5 лет. Данные приведены в таблице 12.

Таблица 11 – Минимальное количество приобретений подписки в год

Наименование подписки	«Стандарт»	«Премиум»	Сумма
Количество приобретений подписок	540	264	804
Выручка	53520 руб.	80280 руб.	133800 руб.

Таблица 12 – Экономическая эффективность проекта

Показатели	Год				
	1	2	3	4	5
Обслуживание веб-сервера	60000 руб.	60000 руб.	60000 руб.	60000 руб.	60000 руб.
Публикация приложения в «Google Play»	1500 руб.	-	-	-	-
Продвижение сайта и мобильного приложения для привлечения аудитории	50000 руб.	50000 руб.	50000 руб.	50000 руб.	50000 руб.
Всего затраты	111500 руб.	110500 руб.	110500 руб.	110500 руб.	110500 руб.
Количество подписок	804	1125	1575	2205	3087
Прибыль до налогообложения	133800 руб.	191250 руб.	267750 руб.	374850 руб.	524790 руб.
Налог на прибыль	22300 руб.	38250 руб.	53550 руб.	74970 руб.	104958 руб.
Чистая прибыль	111500 руб.	153000 руб.	214200 руб.	299880 руб.	419832 руб.
Экономическая эффективность	0 руб.	42500 руб.	103700 руб.	189380 руб.	309332 руб.

Прибыль от продажи подписок на пользование программным комплексом через веб-сайт в первый год продаж будет составлять 133800 рублей. К пятому году она увеличится до 524790 рублей.

В первый год реализации проекта инвестиционные затраты будут состоять из обслуживания веб-сервера, публикации приложения в «Google Play» и рекламы для привлечения аудитории. В последующие года затраты будут составлять эту же сумму за вычетом затрат на публикацию приложения.

Чистая прибыль от данного программного комплекса в первый год составит 111500 рублей и к пятому году достигнет 419832 рублей. Соответственно, экономическая эффективность проекта к пятому году достигнет 309332 рублей. Из этого можно сделать вывод, что проект имеет положительный эффект и является экономически эффективным.

Выводы по третьей главе.

В третьей главе работы проведено тестирование созданной системы. Выявлены и устранены ошибки в работе приложения и сервера.

Посчитаны затраты на разработку и внедрение комплекса. Разработаны две разновидности клиентских подписок на пользование данным продуктом. Данные результаты использованы в проведении расчета экономической эффективности проекта.

Заключение

Целью данной бакалаврской работы было создание комплекса, автоматизирующего процесс ввода данных персоналом о продукции, хранящейся в торговом автомате. В ходе выполнения были поставлены и выполнены следующие задачи:

- разработка структуры будущего программного комплекса;
- проведение анализа уже существующих систем, решающих данную задачу;
- изучены все необходимые технологии разработки для создания запланированного комплекса;
- моделирование и разработка базы данных для хранения информации о каждом автомате;
- разработка серверной части;
- разработка клиентской части для веб-сайта и мобильного приложения;
- тестирование приложения, выявление и устранение существующих недочетов.

После выполнения поставленных задач цель работы была достигнута: разработан достаточно функциональный программный комплекс для ведения учета в вендинг-бизнесе, который позволяет оперативно получать статистическую информацию, поможет избавиться от рутинности бумажного процесса и сэкономит время и силы как оператора, так и администратора, а следовательно, увеличит рентабельность бизнеса.

Программный комплекс постоянно совершенствуется, и следующим этапом разработки будут добавление новых видов статистики на сайт, создание приложения для платформы iOS, улучшение интерфейса, повышение отзывчивости мобильного приложения и сайта, исходя из пожеланий пользователей.

Список используемой литературы и используемых источников

1. Бенкен Е. С. AJAX: программирование для Интернета. – БХВ-Петербург, 2012. 436 с.
2. Голощапов А. Л. Android. Создание приложений для смартфонов и планшетных ПК. Наиболее полное руководство. – БХВ-Петербург, 2013. 832 с.
3. Дон Гриффитс, Дэвид Гриффитс Head First Программирование под Android, 2-е изд.: Пер. с англ.: Уч.пос. – «Питер», 2016. 694 с.
4. П.Дейтел, Х.Дейтел Android для программистов. Создаем приложения, 2-е изд.: Пер. с англ.: Уч.пос. – «Питер», 2016. 694 с.
5. Дронов Владимир PHP 5/6, MySQL 5/6 и Dreamweaver CS4. Разработка интерактивных Web-сайтов; БХВ-Петербург- Москва, 2009. 544 с.
6. Жилияк, Н. А. Введение в JavaScript : учебно-методическое пособие для студентов учреждений высшего образования по направлению специальности 1-40 05 01-03 "Информационные системы и технологии (издательско-полиграфический комплекс)" - Минск : БГТУ, 2014. 239 с.
6. Заяц А. М., Васильев Н. П. Проектирование и разработка WEB-приложений. Введение в frontend и backend разработку на JavaScript и node. Js // Лань - 2019. – 120 с.
7. Казарин С. А., Клишин А. П. Среда разработки Java-приложений Eclipse (ПО для объектно-ориентированного программирования и разработки приложений на языке Java) //М.: Федеральное агентство по образованию. – 2008. – 77 с.
8. Кахриманова Д. Г., Магомедов Г. Д. Вендинговый бизнес и сервис услуг //Инновационная наука. – 2015. – №. 6-1. – С. 86-88.
9. Качан Н. А., Шумская Л. И. Розничная торговля товарами через вендинг-автоматы: организация и особенности учета // Вестник Белгородского университета кооперации, экономики и права. – 2015. – №. 3. – С. 210-214.

10. Кириллов В. В. Введение в реляционные базы данных. – БХВ-Петербург, 2009. – 464 с.
11. Лыткина Е. А., Глотова А. Г. Основы языка HTML // Северный (Арктический) федеральный университет им. М.В. Ломоносова (Архангельск) – 2014. – 112 с.
12. Майер Р. Android 2. Программирование приложений для планшетных компьютеров и смартфонов. – Litres, 2022. – 671 с.
13. Массель Л. В., Массель А. Г. Семантические технологии на основе интеграции онтологического, когнитивного и событийного моделирования //Открытые семантические технологии проектирования интеллектуальных систем. – 2013. – №. 3. – С. 247-250.
14. Моргунов Е. П., Рогова Е. В., Лузанова П. В. PostgreSQL. Основы языка SQL // – БХВ-Петербург, 2018. - 336 с.
15. Наумов Р. В. Android программирование. Начало программирования //Academy. – 2016. – №. 1 (4). – С. 46-48.
16. Пауэрс Ш. Добавляем Ajax. – БХВ-Петербург, 2009. – 448 с.
17. Приходько А. П. Анализ и выбор инструментария для разработки веб-сайта //Актуальные научные исследования в современном мире. – 2020. – №. 7-1. – С. 124-127.
18. Розанов И. А. Удалённая автоматизированная диагностика вендинговых автоматов // Новые информационные технологии в автоматизированных системах. – 2016. – №. 19. – С. 283-287.
19. Сухорослов О. В. Унифицированный интерфейс доступа к алгоритмическим сервисам в Web //Труды Института системного анализа Российской академии наук. – 2009. – Т. 46. – С. 60-82.
20. Kit Vending – комплексное решение для вендинга [Электронный ресурс]. URL: <https://vending.kit-invest.ru/>(дата обращения: 11.07.2022)
21. Мониторинг торговых (вендинговых) автоматов, эквайринг, телеметрия и кассы для вендинга | Telemetron [Электронный ресурс]. URL: <https://telemetron.ru/>(дата обращения: 11.07.2022)

22. Home – Nayax [Электронный ресурс]. URL: <https://www.nayax.com/>(дата обращения: 11.07.2022)
23. Global telemetry, IoT, payments for Vending [Электронный ресурс]. URL: <https://vendon.net/>(дата обращения: 11.07.2022)
24. Daniel A. Begun Amazing Android Apps For Dummies; Вильнюс: Vaga - Москва, 2011. - 256 с.
25. Duckett J. HTML & CSS: design and build websites. – Indianapolis, IN : Wiley, 2011. – Т. 15. – 512 с.
26. Li Q., Chen Y. L. Entity-relationship diagram //Modeling and analysis of enterprise and information systems. – Springer, Berlin, Heidelberg, 2009. – С. 125-139.
27. Nixon R. Learning PHP, MySQL & JavaScript: With jQuery, CSS & HTML5. – " O'Reilly Media, Inc.", 2014. – 806 с.
28. Philip J. Pratt, Mary Z. Last A Guide to MySQL; БХВ-Петербург, Арлит - Москва, 2012. - 304 с.

Приложение А

Листинг создания таблиц на языке SQL

```
CREATE TABLE `automates` (  
  `id` int(11) NOT NULL AUTO_INCREMENT,  
  `code` varchar(20) NOT NULL,  
  `address` text NOT NULL,  
  `comment` text NOT NULL,  
  `status` int(11) DEFAULT NULL,  
  PRIMARY KEY (`id`)  
)
```

```
CREATE TABLE `fillings` (  
  `id` int(11) NOT NULL AUTO_INCREMENT,  
  `user_id` int(11) NOT NULL,  
  `automate_id` int(11) NOT NULL,  
  `product_id` int(11) NOT NULL,  
  `product_count` int(11) NOT NULL,  
  `date` datetime DEFAULT NULL,  
  PRIMARY KEY (`id`)  
)
```

```
CREATE TABLE `products` (  
  `id` int(11) NOT NULL AUTO_INCREMENT,  
  `name` varchar(40) NOT NULL,  
  `weight` float NOT NULL,  
  `spiral_type` int(11) NOT NULL,  
  `status` int(11) NOT NULL,  
  PRIMARY KEY (`id`)  
)
```

Продолжение Приложения А

```
CREATE TABLE `products_state` (  
  `id` int(11) NOT NULL AUTO_INCREMENT,  
  `automate_id` int(11) NOT NULL,  
  `product_id` int(11) NOT NULL,  
  PRIMARY KEY (`id`)  
)
```

```
CREATE TABLE `trade_prices` (  
  `id` int(11) NOT NULL AUTO_INCREMENT,  
  `product_id` int(11) DEFAULT NULL,  
  `cost` float DEFAULT NULL,  
  `date` datetime DEFAULT NULL,  
  PRIMARY KEY (`id`)  
)
```

```
CREATE TABLE `users` (  
  `id` int(11) unsigned NOT NULL AUTO_INCREMENT,  
  `fio` text,  
  `login` varchar(20) NOT NULL,  
  `password` varchar(40) NOT NULL,  
  `type` tinyint(3) unsigned NOT NULL,  
  `token` varchar(40) DEFAULT NULL,  
  `salt` varchar(12) DEFAULT NULL,  
  `token_active` datetime DEFAULT NULL,  
  `token_expire` datetime DEFAULT NULL,  
  `new_salt` varchar(12) DEFAULT NULL,  
  `status` int(11) NOT NULL,  
  PRIMARY KEY (`id`)  
)
```

Продолжение Приложения А

```
CREATE TABLE `vend_prices` (  
  `id` int(11) NOT NULL AUTO_INCREMENT,  
  `automate_id` int(11) DEFAULT NULL,  
  `product_id` int(11) DEFAULT NULL,  
  `cost` float DEFAULT NULL,  
  `date` datetime DEFAULT NULL,  
  `price_active` int(11) DEFAULT NULL,  
  PRIMARY KEY (`id`)  
)
```

```
INSERT INTO `users` (`id`, `fio`, `login`, `password`, `type`, `status`)  
VALUES (1, 'Alex', 'alex@yandex.ru',  
'cWuJydQYaFv9pMJNSCRJabQtSDGP8pnlgmV', 1, 1);
```

Приложение Б

Листинг файла mobile.php

```
<?php
    if (isset($_POST['json'])) {
        include_once("../lib/vendbook.inc.php");
        $agentDB = new AgentDB("mysql.example.com", "dbname", "login",
"password");
        $agentSecurity = new AgentSecurity($agentDB);
        $agentOperator = new AgentOperator($agentDB);
        $agentSecretar = new AgentSecretar($_POST['json'], $agentSecurity,
$agentOperator, 'Mobile');
        echo $agentSecretar->getAnswer();
    }
?>
```

Приложение В

Листинг файла site.php

```
<?php
    if (isset($_POST['json'])) {
        include_once("../lib/vendbook.inc.php");
        $agentDB = new AgentDB("mysql.example.com", "dbname", "login",
"password");
        $agentSecurity = new AgentSecurity($agentDB);
        Продолжение Приложения Б
        $agentOperator = new AgentOperator($agentDB);
        $agentSecretar = new AgentSecretar($_POST['json'], $agentSecurity,
$agentOperator, 'Site');
        echo $agentSecretar->getAnswer();
    }?>
```

Приложение Г

Примеры запросов

авторизоваться в мобильном приложении -

```
{“request”:“authMobile.get”, “login”:“a@yandex.ru”,  
“token”:“12sd4az16d22s”}
```

ответ - {“token”:“2sada231df”} или ошибка - {“errors”:[5]}

получить список автоматов с продуктами - {“request”:“automates.get”,
“login”:“a@yandex.ru”, “token”:“2sada231df”}

ответ - {“token”:“2sda212da2a231df”, “automates”:[{“id”:“2”,
“name”:“DCITY”, “address”:“Mojas”, “products”:[{“name”:“Apple”, “id”:“1”,
“cost”:“12”}, {“name”:“Orange”, “id”:“12”, “cost”:“23”}]}]}

Приложение Д

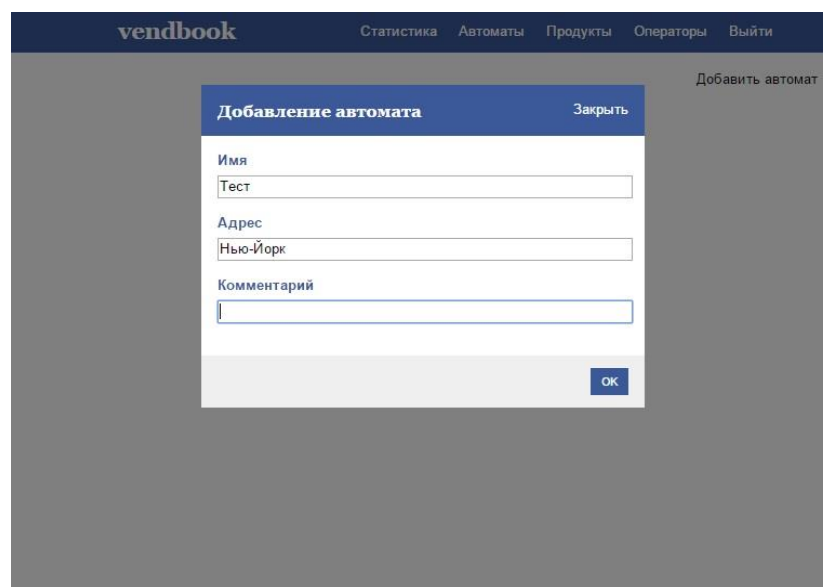
Скриншоты сайта



The screenshot shows the 'vendbook' website interface. At the top, there is a navigation bar with the logo 'vendbook' and menu items: 'Статистика', 'Автоматы', 'Продукты', 'Операторы', and 'Выйти'. Below the navigation bar, there are two date input fields: '01.11.2014' and '20.12.2014', followed by a 'Показать' button. The main content area displays a table with the following data:

Продукт	Количество	Прибыль	Оборот
KitKat	8	16	192
Nesquik Crispy	2	4	56
Круассан	6	12	84
Читос	7	7	133
Чипсы Lay's	3	3	54

Рисунок Д.1 – Скриншот сайта – статистика



The screenshot shows the 'vendbook' website interface with a modal form titled 'Добавление автомата'. The form has a 'Закреть' button in the top right corner. It contains three input fields: 'Имя' with the value 'Тест', 'Адрес' with the value 'Нью-Йорк', and 'Комментарий' which is empty. At the bottom right of the form is an 'ОК' button. In the top right corner of the page, there is a link 'Добавить автомат'.

Рисунок Д.2 – Скриншот сайта - добавление автомата

Продолжение Приложения Д

The screenshot shows a web application interface for 'vendbook'. At the top, there is a dark blue navigation bar with the logo 'vendbook' on the left and menu items 'Статистика', 'Автоматы', 'Продукты', 'Операторы', and 'Выйти' on the right. Below the navigation bar, there is a light gray area with a link 'Добавить оператора' on the right. A modal dialog box titled 'Добавление оператора' is centered on the screen. The dialog has a blue header with the title and a 'Закрыть' button. It contains three input fields: 'ФИО' with the value 'Романов Александр Павлович', 'E-mail' with the value 'alexandr1@empire.ru', and 'Пароль' with the value 'Боже, царя храни!'. At the bottom right of the dialog is an 'OK' button.

Рисунок Д.3 – Скриншот сайта - добавление оператора

The screenshot shows the 'Operators' section of the 'vendbook' application. It features the same dark blue navigation bar as in the previous screenshot. Below the navigation bar, there is a light gray area with a link 'Добавить оператора' on the right. A table with two columns and two rows is displayed. The table has a header row with columns: '№', 'ФИО', 'E-mail', 'Пароль', and two small icons (a pencil and an 'x'). The first row contains the values: '1', 'Романов Николай Павлович', 'nick1@empire.re', '*****', and the two icons. The second row contains the values: '2', 'Романов Александр Павлович', 'alexandr1@empire.re', '*****', and the two icons.

№	ФИО	E-mail	Пароль		
1	Романов Николай Павлович	nick1@empire.re	*****		
2	Романов Александр Павлович	alexandr1@empire.re	*****		

Рисунок Д.4 – Скриншот сайта – операторы

Продолжение Приложения Д

vendbook							Статистика	Автоматы	Продукты	Операторы	Выйти
							Добавить продукт				
№	Продукт	Вес	Цена покупки								
1	Чипсы Lay's	0.35	17 руб.	2014-12-01 00:00:00							
2	Читос	0.35	18 руб.	2014-12-01 00:00:00							
3	Сухарики, 40 г	0.04	0 руб.	0000-00-00 00:00:00							
4	Сухарики+соус	0.06	0 руб.	0000-00-00 00:00:00							
5	Круассан	0	12 руб.	2014-12-01 00:00:00							
6	Nesquik, 100 г	0.1	0 руб.	0000-00-00 00:00:00							
7	Nesquik Crispy	0.014	26 руб.	2014-12-01 00:00:00							
8	KitKat	0.032	22 руб.	2014-12-01 00:00:00							
9	M&M's	0.03	0 руб.	0000-00-00 00:00:00							
10	Skittles	0.03	0 руб.	0000-00-00 00:00:00							
11	Зверюшки	0.04	0 руб.	0000-00-00 00:00:00							
12	Бонди	0.02	0 руб.	0000-00-00 00:00:00							
13	Snickers	0.055	0 руб.	0000-00-00 00:00:00							
14	Twix	0.055	0 руб.	0000-00-00 00:00:00							
15	Bounty	0.055	0 руб.	0000-00-00 00:00:00							

Рисунок Д.5 – Скриншот сайта – продукты

vendbook		Статистика	Автоматы	Продукты	Операторы	Выйти
						Добавить автомат
Редактирование автомата						Заккрыть
Имя						
<input type="text" value="VND002"/>						
Адрес						
<input type="text" value="DCITY"/>						
Комментарий						
<input type="text" value="LED"/>						
<input type="button" value="OK"/>						

Рисунок Д.6 – Скриншот сайта - редактирование автомата

Продолжение Приложения Д

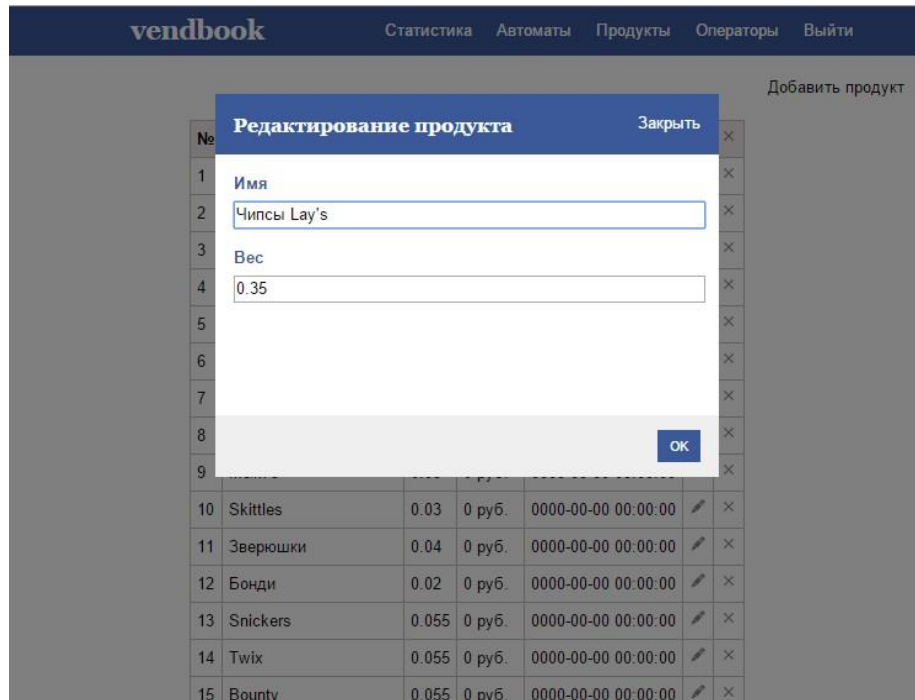


Рисунок Д.7 – Скриншот сайта - редактирование продукта

Приложение Е

XML-формат для сохранения данных пакетов в мобильном приложении

<points>

<point id="1" name="RO" address="DCITY" date="12.06.2014">

<product id="1" name="Apple" cost="12" count="4"/>

<product id="2" name="Banan" cost="35" count="12"/>

<product id="3" name="Orange" cost="24" count="8"/>

</point>

<point id="2" name="RO" address="DCITY2" date="14.06.2014">

<product id="1" name="Apple" cost="12" count="4"/>

<product id="2" name="Banan" cost="35" count="12"/>

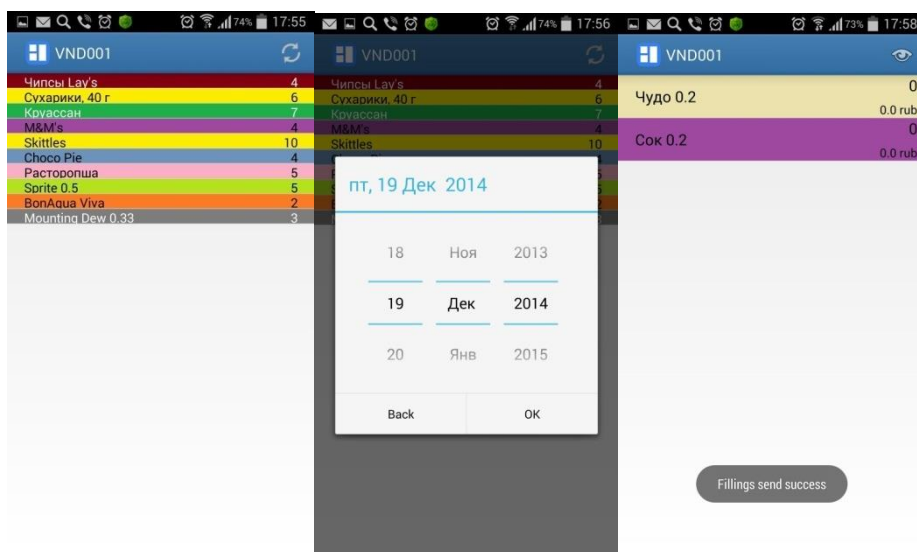
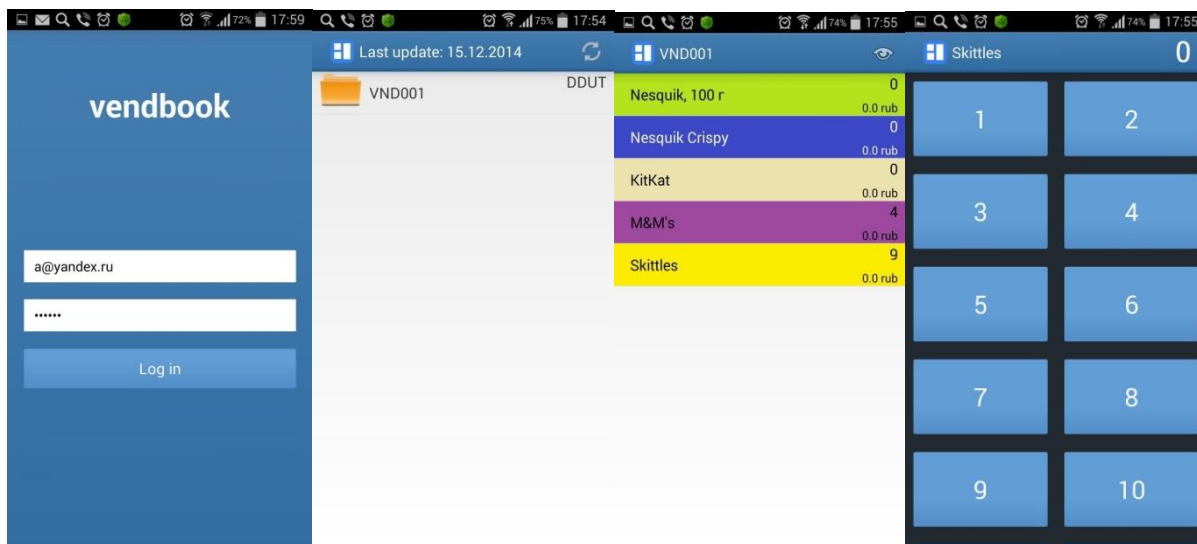
<product id="3" name="Orange" cost="24" count="8"/>

</point>

</points>

Приложение Ж

Скриншоты приложения



Рисунки Ж.1-7. скриншоты созданного приложения