

МИНОБРНАУКИ РОССИИ

Федеральное государственное бюджетное образовательное  
учреждение высшего образования

**«САРАТОВСКИЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ  
ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ ИМЕНИ Н.Г. ЧЕРНЫШЕВСКОГО»**

Кафедра Математического и компьютерного моделирования

**Разработка и реализация ИС**

**предприятия агропромышленного комплекса**

АВТОРЕФЕРАТ БАКАЛАВРСКОЙ РАБОТА

студента 4 курса 441 группы

направления 09.03.03 — Прикладная информатика

механико-математического факультета

Гиевого Даниила Сергеевича

Научный руководитель

доцент, к.э.н.,

Ю.В. Мельникова

Зав. кафедрой

зав.каф., д.ф.-м.н., доцент

Ю.А. Блинков

**Введение.** В современных условиях информационные технологии занимают особое место в развитии бизнеса. Их использование позволяет улучшить процессы продажи товаров, уменьшить расходы на содержание торговой точки и оптимизировать поставку товаров на предприятие. В условиях экономических санкций, покупка оборудования для сельхозтехники зарубежного производства стала практически невозможна для рядового фермера. Поэтому востребованы компании, способные заниматься нахождением, завозом и продажей такого оборудования внутри страны посредством современных информационных технологий. В рамках данной бакалаврской работы разрабатывается информационная система, позволяющая осуществить эти действия.

Цель работы заключается в разработке и реализации информационной системы для агропромышленного комплекса (АПК), которая позволит управлять эффективностью продаж, контролировать движение товаров на складе, ускорить процессы поставки товаров в магазин. Для решения этой цели поставлены следующие задачи:

1. Провести анализ производственных процессов предприятия АПК;
2. Спроектировать информационную систему с помощью UML диаграмм;
3. Спроектировать базу данных;
4. Разработать серверную часть информационной системы с помощью PHP;
5. Разработать мобильное приложение с помощью Swift.

Данная информационная система будет полезна всем сторонам, ее использующим: как покупателю товара для удобного и быстрого приобретения необходимого оборудования, так и руководителям и менеджерам, ответственным за продажи и контроль его использования на предприятии. Внедрение данной системы позволит повысить эффективность работы, сократить расходы механизации и амортизации и оптимизировать бизнес-процессы на предприятии АПК.

**Первый раздел** посвящен разработке следующих диаграмм:

- диаграмма вариантов использования;
- диаграмма последовательности;
- диаграмма активности.

Унифицированный язык моделирования UML – это метод объектно-ориентированного анализа и проектирования информационных систем. UML относится к стандартным языкам моделирования и содержит стандартный набор диаграмм и нотаций разнообразных видов:

Использование UML не ограничивается моделированием программного обеспечения. Его также используют для моделирования бизнес-процессов, системного проектирования и отображения организационных структур.

UML позволяет также разработчикам программного обеспечения достигнуть соглашения в графических обозначениях для представления общих понятий, таких как класс, компонент, обобщение, объединение и поведение, и сконцентрировать больше времени и ресурсов на проектировании и архитектуре.

Диаграмма вариантов использования включает в себя два компонента:

1. Варианты использования – это последовательность событий транзакций, выполняемых системой в ответ на события, инициализируемые некоторым внешним объектом. Вариант использования описывает типичные взаимодействия между пользователем и системой. В простейшем случае варианты использования определяются в процессе обсуждения с пользователями технических функций, которые необходимо реализовать;
2. Действующее лицо – это роль, которую пользователь играет по отношению к системе. Действующее лицо может также быть внешней системой, которой необходима некоторая информация данной системы.

Скомпилированная диаграмма вариантов использования для информационной системы представлена в соответствии с рисунком 1.



Рисунок 1

**Второй раздел** посвящен проектированию базы данных ИС. База данных – это упорядоченный набор структурированной информации или данных, которые обычно хранятся в электронном виде в компьютерной системе. База данных обычно управляется системой управления базами данных (СУБД).

При проектировании ИС выбрана СУБД – PostgreSQL. PostgreSQL – это объектно-реляционная система управления базами данных, наиболее развитая из открытых СУБД в мире. Она имеет открытый исходный код и является альтернативой коммерческим базам данных.

Преимущества PostgreSQL заключаются в следующем:

1. Объектно-реляционная модель. Традиционно популярные СУБД – реляционные. Это значит, что данные, которые в них хранятся, представляются в виде записей, связанных друг с другом отношениями. Существует еще одна популярная модель – объектная. Данные представляются в виде объектов, их атрибутов, методов и классов. PostgreSQL – объектно-реляционная СУБД. Это значит, что она поддерживает и объектный, и реляционный подход;

2. Поддержка сложных запросов. Система справляется с задачами разбора и выполнения трудоемких операций, которые подразумевают и чтение, и запись, и валидацию одновременно. Она медленнее аналогов, если речь заходит только о чтении, но в других аспектах превосходит конкурентов;
3. Одновременная модификация базы. Имеется возможность одновременного доступа к базе с нескольких устройств. В СУБД реализована клиент-серверная архитектура, когда база данных хранится на сервере, а доступ к ней осуществляется с клиентских компьютеров;
4. Соответствие ACID. ACID – это набор принципов для обеспечения целостности данных. Аббревиатура расшифровывается как Atomicity, Consistency, Isolation, Durability – атомарность, согласованность, изолированность, прочность. Если база данных соответствует этим принципам, она ведет себя максимально предсказуемо и надежно. В ней низок риск конфликта или непредвиденного поведения системы;
5. Высокая мощность и широкая функциональность. PostgreSQL – возможно, единственная бесплатная СУБД с открытым исходным кодом, которая рассчитана на работу с объемными и сложными проектами. Она мощная, производительная, способна эффективно работать с большими массивами данных;

Схема «сущность-связь» (также ERD или ER-диаграмма) – это разновидность блок-схемы, где показано, как разные «сущности» связаны между собой внутри системы. ER-диаграммы чаще всего применяются для проектирования и отладки реляционных баз данных в сфере образования, исследования и разработки программного обеспечения и информационных систем для бизнеса. ER-диаграммы (или ER-модели) полагаются на стандартный набор символов, включая прямоугольники, ромбы, овалы и соединительные линии, для отображения сущностей, их атрибутов и связей.

Для разрабатываемой ИС спроектирована база данных на основе ER-диаграммы, в соответствии с рисунком 2.

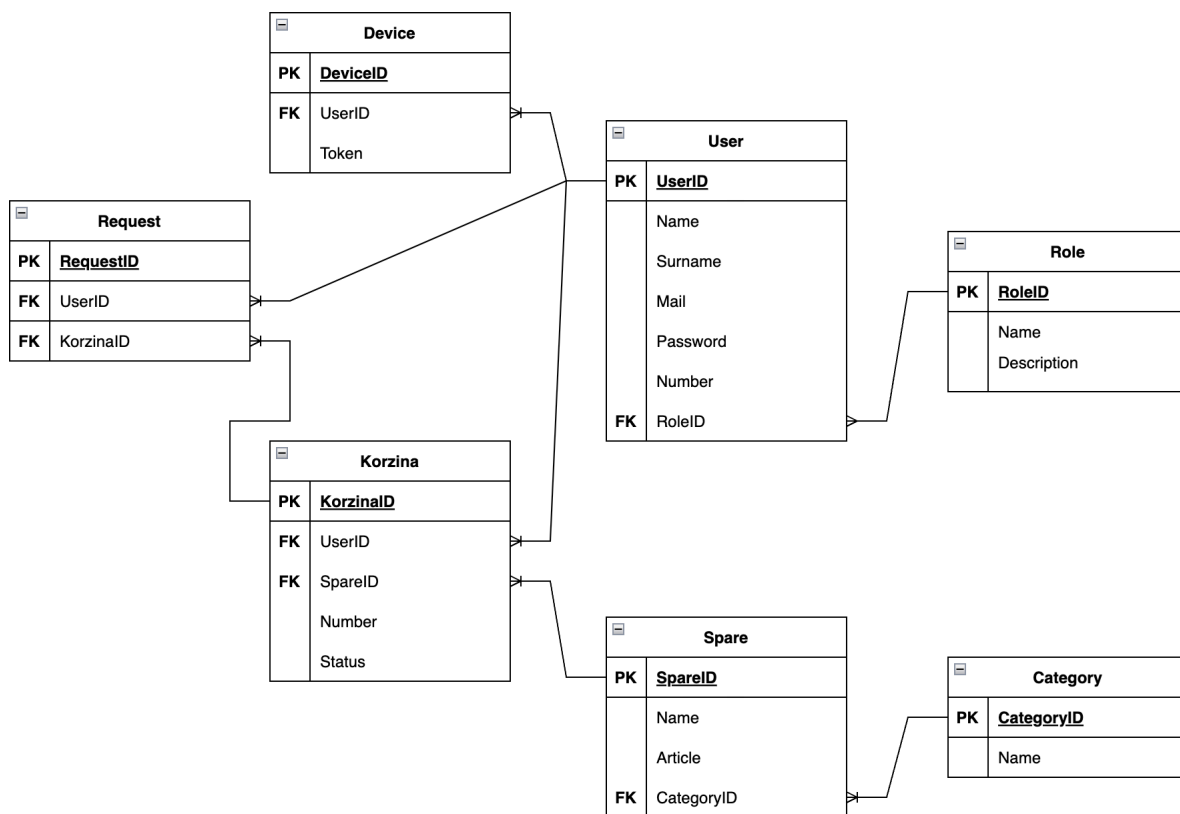


Рисунок 2

**Третий раздел** посвящен разработке ИС. Разработанное приложение – это пользовательский интерфейс, через который пользователь взаимодействует с приложением, серверная часть, содержащая бизнес-логику, и уровень персистентности, в котором хранятся данные и в качестве которого выступает реляционная база данных.

PHP является интерпретируемым языком программирования, который используется для написания серверной части ИС. В данном контексте разработки, PHP используется для создания, обработки и отправки данных на клиентскую сторону.

В данной работе PHP используется в сочетании с фреймворком Symfony для создания более эффективных, безопасных и надежных приложений. Symfony – фреймворк для языка программирования PHP, позволяющий разрабатывать сложные веб-приложения. Наиболее важные преимущества – это простота освоения, внушительная скорость работы, а также способность поддерживать больше, чем одну парадигму программирования.

Symfony имеет поддержку множества баз данных, таких как MySQL, MariaDB, PostgreSQL, SQLite. В основном реляционные базы данных и классические классы в объектно-ориентированном программировании предполагают разные подходы к хранению данных, а также механизмы управления этими данными, поэтому возникает задача синхронизации изменений между этими уровнями.

Структура серверной части ИС строится на трех основных (Entity, Repository, Controller) и нескольких вспомогательных классах в соответствии с рисунком 3, которые реализуются для каждой таблицы, на основе ER-диаграммы.

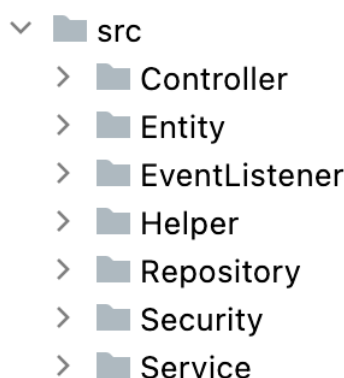


Рисунок 3

Entity в общем случае является объектом, который представляет конкретный элемент данных в приложении. Обычно Entity соответствует таблице в базе данных и хранит информацию о связанных сущностях.

Entity в Symfony являются основными строительными блоками приложения, так как они обычно представляют различные типы данных, которые передаются между компонентами, что позволяет более легко работать с данными и уменьшает связность в приложении.

Класс Repository в общем случае является шаблоном проектирования, предназначенным для управления доступом к данным в приложении. Обычно репозиторий отвечает за запросы к базе данных или другим источникам данных, хранение моделей данных и их поиск в соответствии с заданными критериями.

В Symfony Repository – это класс, который используется для доступа к данным, связанным с сущностями (т.е. моделями данных). Классы репозито-

рия позволяют выполнять запросы к базе данных через сущности. Репозитории могут включать методы для поиска, сортировки и фильтрации данных, а также для выполнения операций вставки, обновления и удаления данных.

Controller обычно используется для обработки и маршрутизации запросов от пользователя к соответствующей модели и представлению. Контроллер часто содержит методы, которые вызываются в зависимости от типа запроса и которые получают или обрабатывают данные и передают их в представление.

В Symfony Controller – это класс, который отвечает за обработку HTTP-запроса и взаимодействие с сервисами и моделями данных для генерации нужной HTTP-ответа.

Контроллеры в Symfony содержат методы, которые обрабатывают запросы и возвращают HTTP-ответы. Классы контроллеров могут содержать логику обработки форм, роутинга и работы с базой данных. Контроллеры могут использовать сервисы и зависимости приложения для обработки запросов и генерации ответов.

Мобильное приложение написано на языке программирования Swift с использованием фреймворка SwiftUI.

Swift – это мощный и простой язык программирования, представленный Apple в 2014 году, благодаря которому открывается возможность создания приложений для iOS и MacOS. Swift – это современный язык программирования со стабильной релизной версией, который достаточно прост в освоении. Он предотвращает массу ошибок на этапе компиляции, ускоряет процесс разработки и повышает качество кода.

SwiftUI – это фреймворк с набором готовых библиотек для создания пользовательского интерфейса в iOS-приложениях, который позволяет декларативно разрабатывать приложения, то есть позволяет сообщать обо всех возможных состояниях приложения сразу. Этот инструмент, разработанный Apple, пришёл на смену UIKit – последний почти во всём устраивал разработчиков, но создание элементов в части UI занимало много времени.

Разработанное мобильное приложение состоит из четырех основных экранов:

1. AuthorizationView – Экран авторизации в соответствии с рисунком 4;



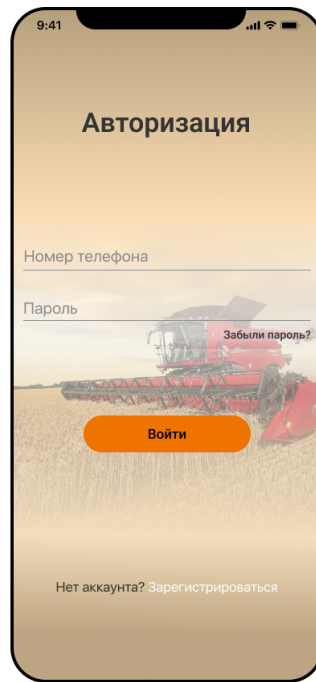


Рисунок 4

2. ProductListView – Экран списка продуктов;
3. BasketView – Экран корзины;
4. ProfileView – Экран профиля в соответствии с рисунком 5;

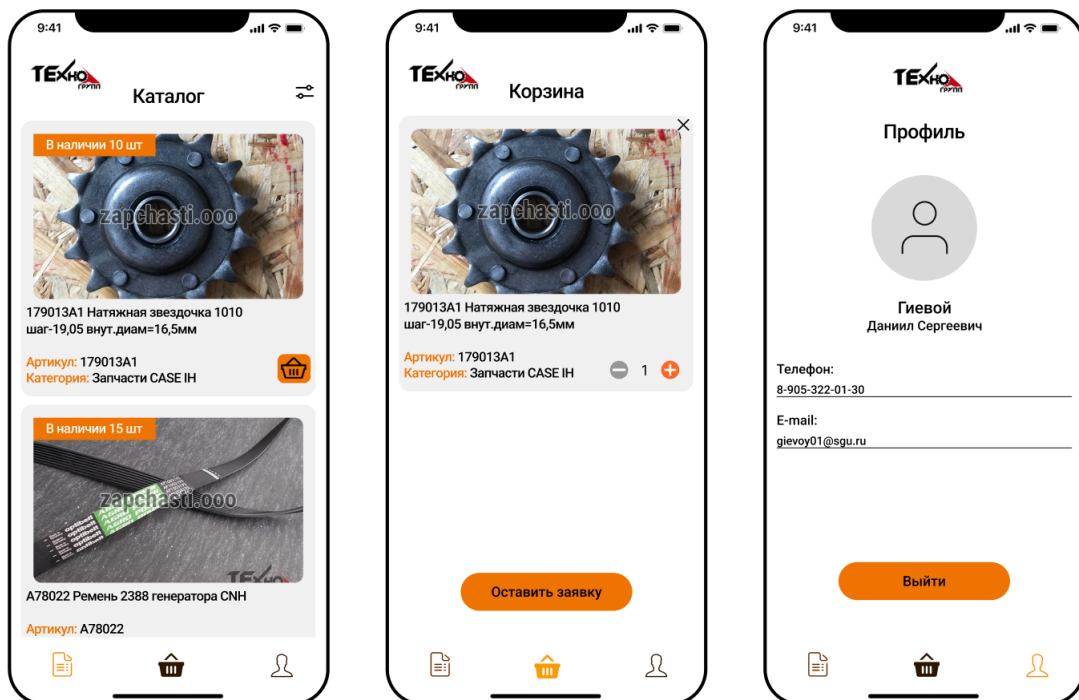


Рисунок 5

Аутентификация является, пожалуй, наиболее распространенным требованием любого приложения. Возможность быстро и легко зарегистрироваться или войти на сайт играет огромную роль для пользователя. В разработанном приложении это делается с помощью сервера, который отслеживает сеансы в той или иной форме.

Одно из лучших правил заключается в том, чтобы предоставить пользователям достаточную свободу действий, чтобы они могли уничтожить свои собственные данные, но не чужие. Очень важно помнить о том, что строится защита приложения от злоумышленников, а не попытка необоснованно ограничить пользователей.

Процесс аутентификации должен включать в себя сервер. Самое простое средство аутентификации работает в основном следующим образом:

1. Клиент направляет пользователя к процессу аутентификации на стороне сервера;
2. Сервер проверяет подлинность пользователя с помощью пароля, цифровых подписей или других средств;
3. Сервер создает случайно сгенерированный токен (ключ) и связывает его с уже прошедшим аутентификацию;
4. Сервер передает токен обратно клиенту;
5. Клиент использует предоставленный токен при последующих запросах к серверу в качестве документа, удостоверяющего личность, предоставляющего пользователю доступ к защищенным ресурсам.

Поскольку токен генерируется в момент входа в систему случайным образом, его присутствие служит достаточным доказательством того, что запрос исходит от пользователя, которому был выслан токен.

Стоит обратить внимание на то, что каждый, кто предъявляет токен, немедленно получает доступ к данным. Поэтому жизненно важно, чтобы любое сообщение, которое включает токен, проходило через соединение, защищенное SSL-шифрованием. SSL-сертификат – это цифровой сертификат, удостоверяющий подлинность веб-сайта и позволяющий использовать зашифрованное соединение.

Не менее важным моментом, связанной с защитой персональных данных, является хеширование паролей пользователей. Хеширование пароля -

это процесс перевода пароля в необратимый формат, называемый хешем. Хеш-сумма представляет собой строку фиксированной длины, которая генерируется на основе входных данных. Хеширование паролей используется для обеспечения безопасности паролей, так как защищает пароли от нежелательного доступа и хранит их в надежной форме.

Хеширование пароля выполняется с использованием хеш-функции, которая однозначно отображает исходный пароль в хеш-сумму. Как только произведена хеш-функция, полученная хеш-сумма сохраняется в базе данных вместо самого пароля. Когда пользователь вводит свой пароль, программа выполняет хеш-функцию с введенным паролем и сравнивает с хеш-суммой, сохраненной в базе данных. Если хеш-суммы совпадают, программа позволяет пользователю получить доступ к ресурсам.

При реализации ИС в качестве алгоритма хеширования паролей выбран один из распространенных алгоритмов хеширования `bcrypt`. Он широко используется для хеширования паролей в базах данных и является одним из самых безопасных алгоритмов, применяемых в настоящее время.

Использование `bcrypt` для хеширования паролей – это надежный метод защиты персональных данных и предотвращение возможных угроз веб-безопасности.

**Заключение.** Разработка и реализация информационной системы для агропромышленного комплекса является крайне важным и перспективным направлением в развитии современной сельскохозяйственной отрасли.

Разработанная в рамках данной работы информационная система позволяет как покупателю удобно и быстро приобретать необходимое оборудование, так и руководителям и менеджерам, ответственным за продажи и контроль его использования, повысить эффективность работы и оптимизировать бизнес-процессы.

В ходе работы решены следующие задачи:

1. Проанализированы производственные процессы предприятия АПК;
2. Спроектирована информационная система с помощью UML диаграмм;
3. Спроектирована база данных;
4. Разработана серверная часть информационной системы с помощью PHP;

5. Разработано мобильное приложение с помощью Swift.

Итоговая версия информационной системы, разработанная в данной работе, станет хорошей основой для дальнейшего развития и улучшения процессов в агропромышленном комплексе. Результаты бакалаврской работы внедрены в ООО «Техногрупп» и используются в компании, что обеспечивает повышение эффективности работы, сокращение расходов на механизацию и амортизацию оборудования, и оптимизацию бизнес-процессов.

Таким образом, все цели и задачи, поставленные в данной работе, полностью выполнены.