

МИНОБРНАУКИ РОССИИ
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ
ВЫСШЕГО ОБРАЗОВАНИЯ
“ВОРОНЕЖСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ”

Факультет компьютерных наук

Кафедра программирования и информационных систем

Система обработки полей от вредителей квадрокоптером

Курсовой проект

09.03.02 Информационные системы и технологии

Допущен к защите

Обучающийся _____ П.Н. Парамонов, 3 курс, д/о

Обучающийся _____ А.Ю. Сигарев 3 курс, д/о

Воронеж 2020

Оглавление

Введение.....	4
1. Постановка задачи.....	5
2. Анализ предметной области.....	6
2.1. Описание предметной области.....	6
2.2. Сравнение с аналогами.....	6
2.3. Пользователи системы.....	7
2.5. Графическое описание работы системы.....	10
2.5.1. Диаграмма idеf0.....	11
2.5.2. Диаграмма прецедентов.....	12
2.5.3. Диаграмма последовательностей.....	14
2.5.4. Диаграмма коммуникаций.....	16
2.5.5. Диаграмма состояний.....	16
2.5.6. Диаграмма активностей.....	20
2.5.7. Диаграмма классов.....	22
2.5.8. Диаграмма объектов.....	23
2.5.9. ER-диаграмма.....	24
2.5.10. Диаграмма развертывания.....	27
3. Реализация приложения.....	29
3.1. Анализ средств реализации.....	29
3.2. Разработка Frontend.....	30
3.2.1. Начальная страница.....	30
3.2.2. Создание заказа.....	31
3.2.3. Регистрация/Авторизация пользователя.....	31
3.2.4. Авторизированный пользователь.....	32
3.2.5. Роль менеджера.....	33
3.2.6. Роль оператора.....	38
3.2.7. Блокировка реквизитов.....	41
3.3. Разработка backend.....	41

3.4. Тестирование.....	42
3.4.1. Дымовое тестирование.....	42
3.4.2. Интеграционное тестирование серверной части.....	43
3.4.3. Юзабилити тесты.....	45
Заключение.....	46
Список литературы.....	47

Введение

Несмотря на то, что Российское общество находится на стадии постиндустриального развития, значительную часть экономики занимает сельское хозяйство. Оно является основным поставщиком продуктов питания и сырья для многих отраслей промышленности. Нормальное функционирование сельского хозяйства обеспечивает большую часть физических потребностей человека. Развитие сельского хозяйства обеспечивало развитие общества, наглядно это можно увидеть во времена аграрных революций, например «Зеленой». В конце XX в. в сельском хозяйстве многих развивающихся стран происходила техническая и социальная революция, чему способствовали научнотехнические достижения и инновации, внесенные в аграрный сектор. Результаты сельскохозяйственного производства все в большей степени стали определяться уровнем технического оснащения отрасли. В последние десятилетия выросло количество тракторов и сельхозтехники, увеличилось применение удобрений и средств защиты растений, улучшилась агротехника земледелия и животноводства, постепенно модернизируется структура хозяйств.

Решение проблем сельского хозяйства может быть сконцентрировано в следующем направлении: инвестиции в отрасли, обеспечивающие рост производительности (создание инфраструктуры доступа на рынки, производство сельскохозяйственной техники и химикатов, развитие биотехнологий, развитие агротехники и науки и т. д.) В частности, имеет место быть применение беспилотных летательных средств для проведения рутинных сельскохозяйственных операций, таких например, как обработка от вредителей. В связи с этим возникает потребность в средствах информатизации компаний-посредников, занимающихся операциями данного рода.

1. Постановка задачи

Цель курсовой работы: реализовать веб приложение, которое будет отвечать следующим требованиям:

- Понятный пользователю интерфейс
- Возможность делать заказ на обработку поля от вредителей
- Возможность просматривать историю заказов
- Возможность просматривать текущие заказы
- Возможность для менеджера просматривать текущие пестициды на склады, а так же удалять их
- Возможность для операторов дронов смотреть свои текущие заказы, а так же завершать их и обновлять прогресс.

Для достижения этих целей выделены задачи:

- Разработка Back-end часть приложения, которая отвечает за бизнес логику проекта и развернута на сервере приложения
- Разработка Базы Данных, для хранения больших объемов данных о пользователях, заказах, пестицидов и т.д
- Разработка Front-end части приложения, которая отвечает за предоставления интерфейса пользователю

2. Анализ предметной области

2.1. Описание предметной области

Для решения некоторых сельхоз проблем в сфере обрабатывания полей от вредителей, такие как заправка техники топливом, содержание самой техники, ее обслуживание, неудобность довольно таки большой техники на малых земельных участках и т.д, было решено отказаться от этой всей техники в пользу беспилотных дронов. Ведь они не нуждаются в топливе, нужно лишь их зарядить, следовательно, не загрязняется окружающая среда, а также идет экономия на топливе. Ведь куда дешевле полностью зарядить дрон, нежели полностью заправить трактор. Так как дроны очень маленькие по сравнению с тракторами, то содержание гораздо легче их и обслуживание так же проще. Еще из-за своего размера, проще обрабатывать маленькие участки.

Для всех этих решений было придуман сервис “Agrin”. Который включается в себя, возможность сделать заказ на обработку полей дронами, а также просмотр этих заказов, для клиента. Просмотр списка пестицидов, возможность удаления и добавления пестицидов на склад для менеджера. А также просмотр списка доступных заказов, обновление и завершение заказа для оператора дронов.

2.2. Сравнение с аналогами

Среди аналогов на просторах интернета мы смогли выделить, при беглом просмотре первых страниц поисковика, только один web сервис:

Дронцентр (<https://dronecentre.ru/services/for-agriculture/>). На этом сервисе предлагают такие услуги как покупка дронов разных категорий, обучение пилотированию на дронах, фото и видеосъемка с дронов. Что касается сель-хоз промышленности, на этом сервисе предоставляется такая услуга как, фото-съемка земельного участка, и тем самым получение такой информации как:

- точный объем обрабатываемых и/или пригодных к обработке полей;
- территория пастбищ, сенокосов и других угодий;
- количество, размер, расположение пустых участков;

Из плюсов можно выделить:

- Возможность покупки и обучение пилотирования дрона. Тем самым можно в дальнейшем самому обрабатывать свой участок.
- Съемки различных категорий

Из минусов выделим и самый главный:

- Отсутствие обработки поля, а только лишь расчеты для этой обработки, что является одной из основных целей нашего проекта.

2.3. Пользователи системы

Система предназначена для работы:

- Менеджера системы
- Оператора системы
- Клиента системы

2.4. Анализ задач

Задачи, связанные для гостя:

1. Задача регистрации пользователя

Задача регистрации пользователя разбивается на следующие этапы:

- Неавторизированный пользователь нажимает на кнопку “Регистрация”
- Появляется форма для регистрации
- Пользователь заполняет эту форму

- Данные отправляются на сервер

2. Задача авторизации пользователя

Данная задача разбивается на следующие этапы:

- Неавторизованный пользователь нажимает на кнопку “Войти” 2. Появляется форма для ввода данных
- Пользователь вводит свои данные и нажимает на кнопку “Войти”.
- Данные отправляются на сервер система авторизует пользователя

3. Задача создания заказа

Данная задача разбивается на следующие этапы:

- Пользователь нажимает в навигационном меню кнопку “Заказ”
- Появляется форма заказа
- Пользователь вводит требуемые данные и отправляет их
- Новый заказ становится в режиме обработки

Задачи для авторизованного пользователя:

1. Задача просмотра заказов

Данная задача разбивается на следующие этапы:

- Пользователь нажимает на кнопку “Посмотреть историю своих заказов”
- Система выдает пользователю историю его заказов

2. Задача просмотра текущих заказов

Данная задача разбивается на следующие этапы:

- Пользователь нажимает на кнопку “Посмотреть текущие заказы”
- Система выдает пользователю текущие его заказ

3. Задача редактирования профиля

Данная задача разбивается на следующие этапы:

- Пользователь нажимает на кнопку “Редактировать профиль”
- Пользователь вводит новые данные
- Приложение проверяет данные на валидность, если это так, отправляет данные на сервер
- Сервер меняет данные пользователя

Задачи для менеджера:

1. Задача просмотра пестицидов

Данная задача разбивается на следующие этапы:

- Менеджер нажимает на кнопку “Посмотреть пестициды”
- Система выдает список текущих пестицидов

2. Задача удаления пестицидов

Данная задача разбивается на следующие этапы:

- Менеджер нажимает на кнопку “Посмотреть пестициды”
- Система выдает список пестицидов
- Менеджер выбирает нужные пестициды и удаляет их

3. Задача создания аккаунта оператора

Данная задача разбивается на следующие этапы:

- Менеджер нажимает на кнопку “Создать аккаунт оператора”
- Появляется форма для ввода данных
- Менеджер вводит данные
- Данные отправляются на сервер

4. Задача принятия/отклонения/блокировки заказа клиента

Данная задача разбивается на следующие этапы:

- Менеджер выбирает заказ из списка заказов

- Менеджер выбирает действие с заказом: Принять, отклонить, заблокировать.

Задачи для оператора:

1. Задача просмотра текущих заказов оператора

Данная задача разбивается на следующие этапы:

- Оператор нажимает на кнопку “Текущие заказы”
- Система выдает список текущих заказов

2. Задача обновления прогресса текущего заказа

Данная задача разбивается на следующие этапы:

- Оператор нажимает на кнопку “Текущие заказы”
- Система выдает список текущих заказов
- Оператор выбирает нужный заказ
- Оператор обновляет прогресс заказа

3. Задача завершения текущего заказа

Данная задача разбивается на следующие этапы:

- Оператор нажимает на кнопку “Текущие заказы”
- Система выдает список текущих заказов
- Оператор выбирает нужный заказ
- Оператор завершает заказ

4. Задача создания отчета текущего заказа

Данная задача разбивается на следующие этапы:

- Оператор нажимает на кнопку “Текущие заказы”
- Система выдает список текущих заказов
- Оператор выбирает нужный заказ
- Оператор указывает хим средство во время работы и кол-во потраченного его объема
- Оператор отправляет отчет на сервер

2.5. Графическое описание работы системы

Для удобства описания работы системы была использована графическая нотация UML.

2.5.1. Диаграмма **idef0**

Для описания работы системы в целом была разработана диаграмма IDEF0. Контекстная диаграмма представлена на Рисунке 1.

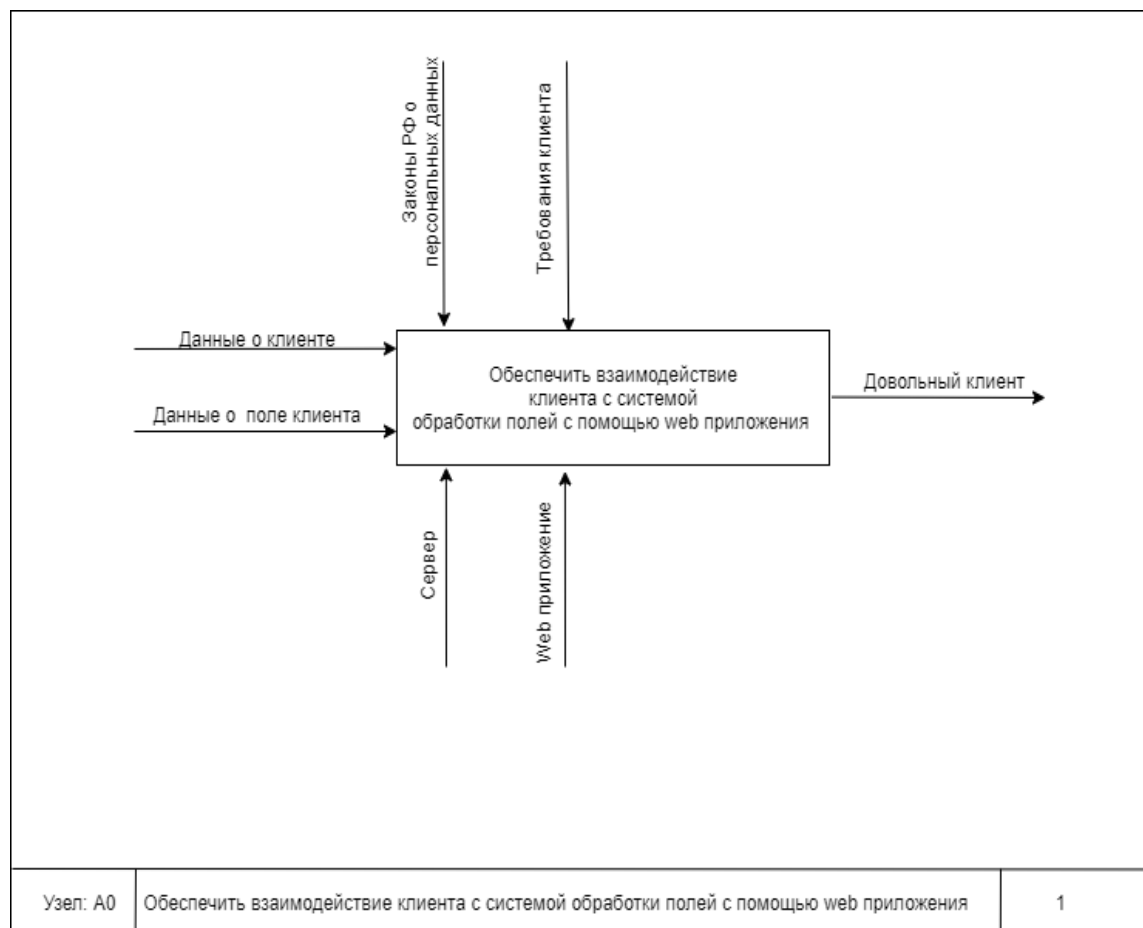


Рис. 1 Диаграмма IDEF0

По ней мы видим основную цель “Обеспечить взаимодействие клиента с системой обработки полей с помощью web приложения”.

Механизмы:

- Сервер
- Web приложение

Входные данные:

- Данные о клиенте
- Данные о поле клиента

Управляющие средства:

- Закон РФ о персональных данных
- Требования клиента

Выход:

- Довольный клиент

На втором уровне декомпозиции диаграммы у нас будут 5 функциональных блоков:

- Блок управления
- Создать заказ клиента
- Подготовить пестициды
- Обработать поле
- Завершить заказ

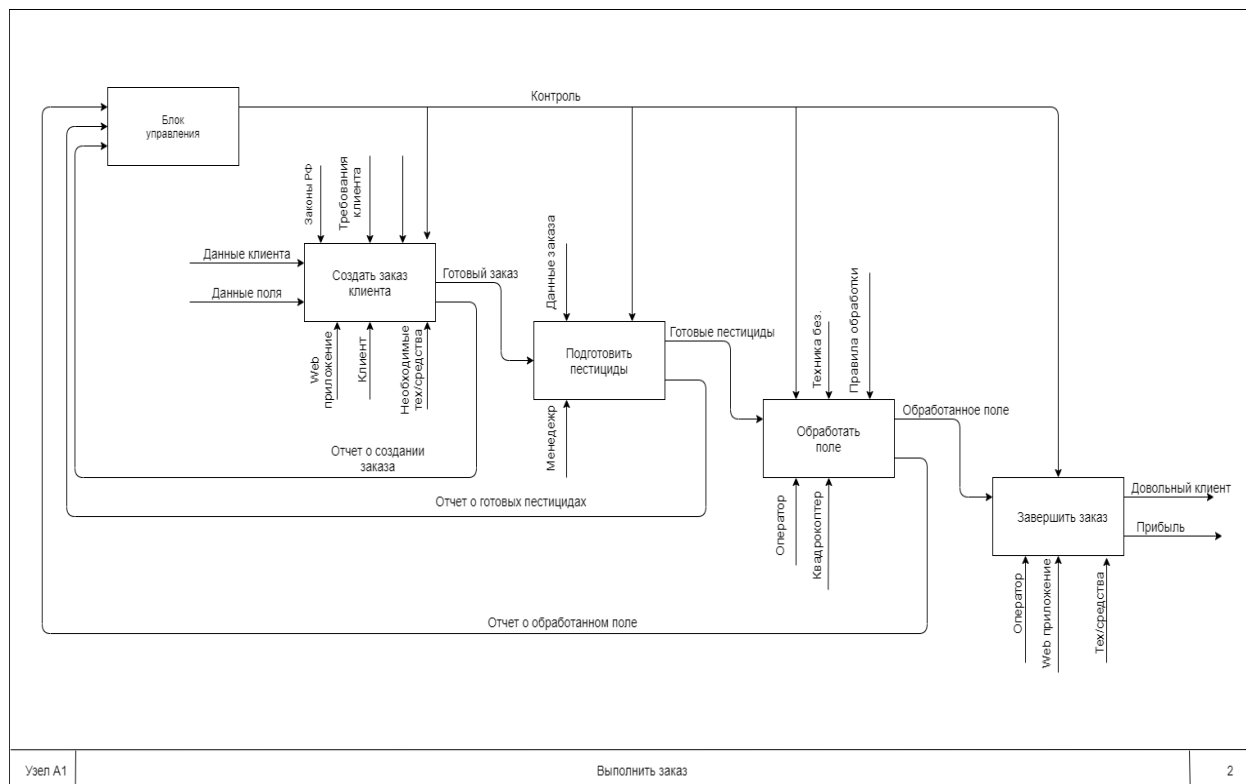


Рис.2 IDEF0 на втором уровне декомпозиции

Блок управления выступает здесь за контроль над всеми блоками, который посылает каждому блоку свои руководства, а также получает отчет о выполнении каждого функционального блока.

2.5.2. Диаграмма прецедентов

Для описания системы на концептуальном уровне была спроектирована диаграмма прецедентов, представленная на Рисунке 3.

Из основных участников можно выделить:

- Менеджер
- Оператор
- Авторизированный пользователь
- Неавторизированный пользователь

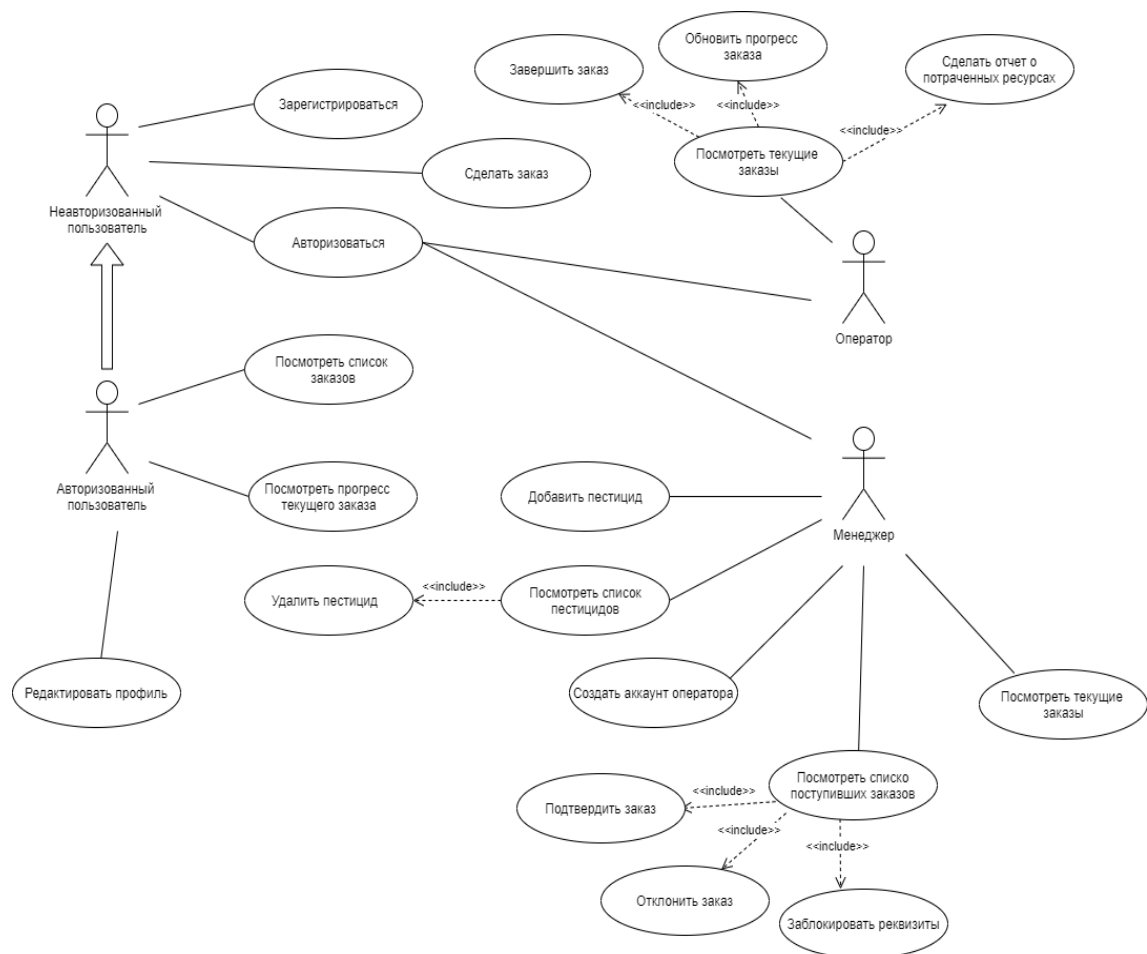


Рис.3 UseCase

У актера “Неавторизированный пользователь” следующие варианты использования:

- Зарегистрироваться
- Авторизоваться
- Сделать заказ

Актер “Авторизированный пользователь” наследует варианты использования от “неавторизированный пользователь” и у него так же есть и свои:

- Посмотреть список заказов
- Посмотреть список текущих заказов
- Редактировать профиль

У актера “Менеджер” следующие варианты использования:

- Добавить пестицид
- Добавить аккаунт оператора
- Посмотреть текущие заказы
- Посмотреть поступившие заказы
 - Заблокировать реквизиты
 - Принять заказ
 - Отклонить заказ
- Посмотреть список пестицидов
 - Удалить пестицид

У последнего актера “Оператор” следующие варианты использования:

- Посмотреть текущие заказы
 - Обновить прогресс заказа
 - Завершить заказ
- Сделать отчет о потраченных ресурсах

2.5.3. Диаграмма последовательностей

Для демонстрации взаимодействия пользователя, администратора и программного продукта была создана диаграмма последовательностей, представленная Рисунке 4.

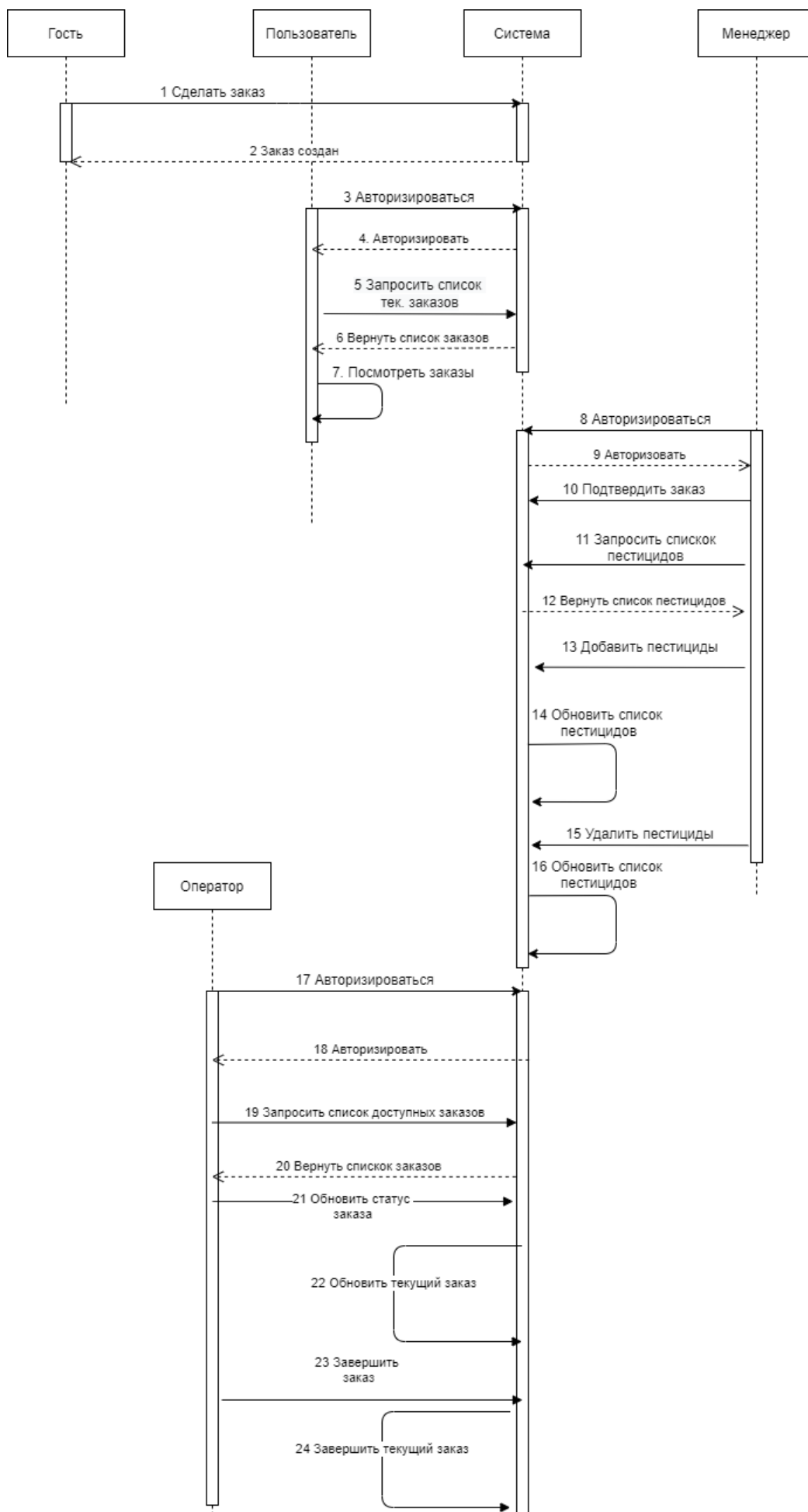


Рис 4. Диаграмма последовательности

2.5.4. Диаграмма коммуникаций

Для визуализации взаимодействия Пользователя, Менеджера, Оператора с системой, была спроектированная диаграмма коммуникаций на Рисунке 5.

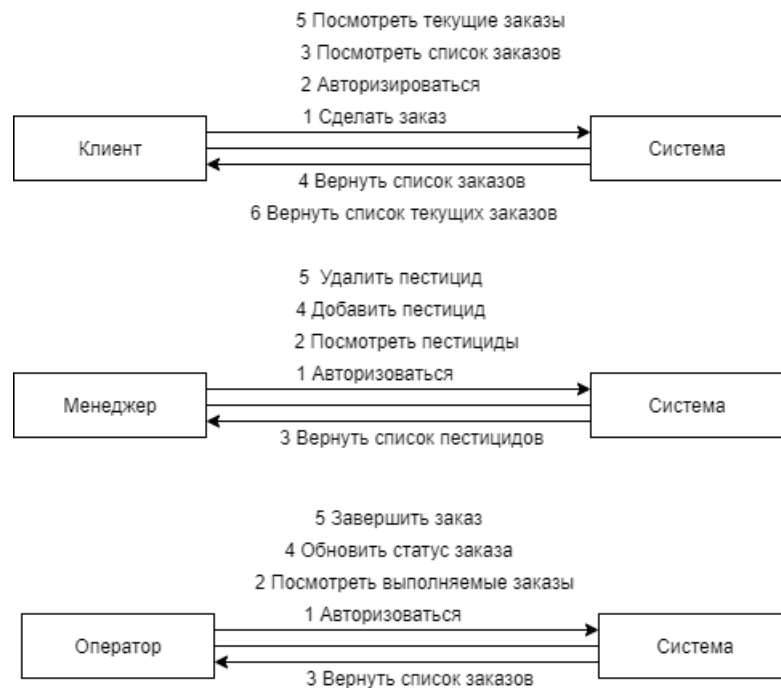


Рис.5 Диаграмма коммуникаций

2.5.5. Диаграмма состояний

Диаграмма состояний, изображенная на Рисунке 6, отражает возможные состояния системы. При запуске приложения система находится в ожидании выбора действия.

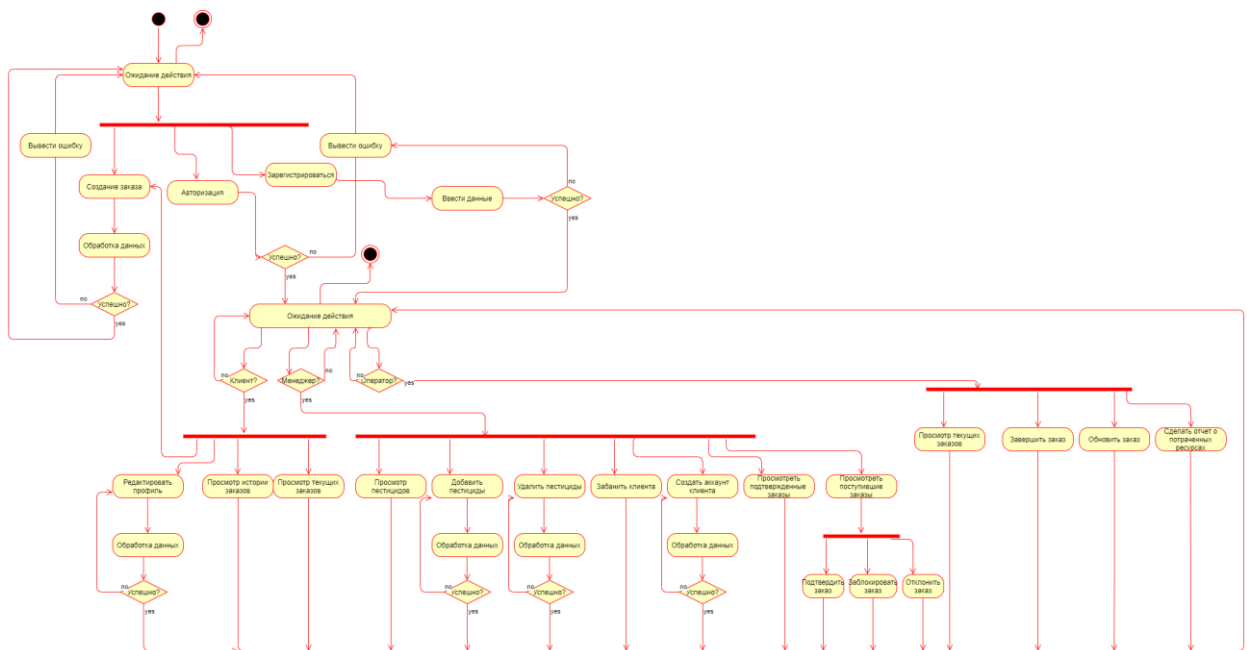


Рис.6 Диаграмма состояний

На данной диаграмме мы видим, что когда пользователь заходит на web приложение, приложение ждет от пользователя выбор:

- Авторизоваться
- Зарегистрироваться
- Создать заказ

Если пользователь выберет создание заказа, приложение потребует ввести необходимые данные, далее пользователь отправляет свои данные. Далее идет обработка полученных данных от пользователя. Если они валидны, то происходит создание заказа. Если же введенные данные не валидны, то приложение сначала показывает информацию об этом, потом приложение опять ожидает действие от пользователя.

Второе состояние, которое может быть – это Авторизация пользователя или же Регистрация пользователя. Пользователь вводит необходимые данные, если они валидны, то система авторизует пользователя с доступной ей ролью: пользователь, менеджер, оператор.

Далее приложение опять ждет от пользователя действий. Если он клиент:

- Просмотр истории заказов
- Просмотр текущих заказов
- Создание заказа
- Редактировать профиль

При просмотре истории заказов, пользователю приложение выдает список всех проведенных заказов и опять ждет действие от пользователя.

При просмотре текущих заказов, пользователю приложение выдает список всех текущих на данный момент заказов и ждет дальнейшее действие.

При создании заказа, пользователь проводит те же самые действия, описанные при неавторизованном пользователе.

При редактировании профиля, пользователь меняет необходимые ему данные своего профиля, если данные корректны приложение меняет данные, иначе выдает сообщение о ошибке, и ждет дальнейшие действия.

Если пользователь Менеджер:

- Просмотр пестицидов
- Добавление пестицидов
- Удаление пестицидов
- Посмотреть список заказов
- Посмотреть поступившие заказы
- Принять заказ
- Отклонить заказ
- Заблокировать заказ
- Создать аккаунт оператора

При просмотре пестицидов, менеджеру приложение выдает список всех пестицидов на складе и ждет дальнейшее действие.

При добавлении пестицидов, менеджер вводит данные о добавляющихся пестицидах, если данные валидны приложение добавляет пестициды, иначе выводит сообщение о ошибке и ждет дальнейшее действие.

При удалении пестицидов, менеджер выбирает необходимые пестициды для удаления, если данные валидны приложение удаляет выбранные пестициды, иначе выводит сообщение о ошибке и ждет дальнейшее действие.

При просмотре поступивших заказов, менеджер может принять заказ, отклонить его или заблокировать реквизиты заказа, далее когда приложение сделает текущее действие оно ждет дальнейшие действия

При просмотре текущих заказов, менеджер видит список текущих заказов, а приложение ждет дальнейшие действия.

При создании аккаунта оператора, менеджер вводит новые данные оператора в поле его регистрации, если данные корректны приложение создает профиль нового оператора, иначе выдает сообщение о ошибке, и ждет дальнейшие действия.

Если пользователь Оператор:

- Просмотр текущих заказов
- Обновление прогресса заказа
- Завершение заказа
- Создание отчета о потраченных ресурсах

При просмотре текущих заказов, оператору приложение выдает список всех доступных ему заказов и ждет дальнейшее действие.

При обновлении прогресса, оператор вводит данные которые необходимо обновить, приложение обновляет прогресс заказа и ждет дальнейшее действие.

При завершении заказа, оператор завершает заказ приложение завершает заказ и ждет дальнейшее действие.

При создании отчета о потраченных ресурсах, оператор выбирает нужный ему заказ, выбирает потраченные ресурс и его количество, приложение делает отчет и ждет дальнейших действий от пользователя.

Если пользователю больше нечего делать в приложении, при ожидании действия, пользователь может выйти.

2.5.6. Диаграмма активностей

С помощью диаграммы активности, представленной на Рисунке 7, была продемонстрирована работа клиента с приложением.

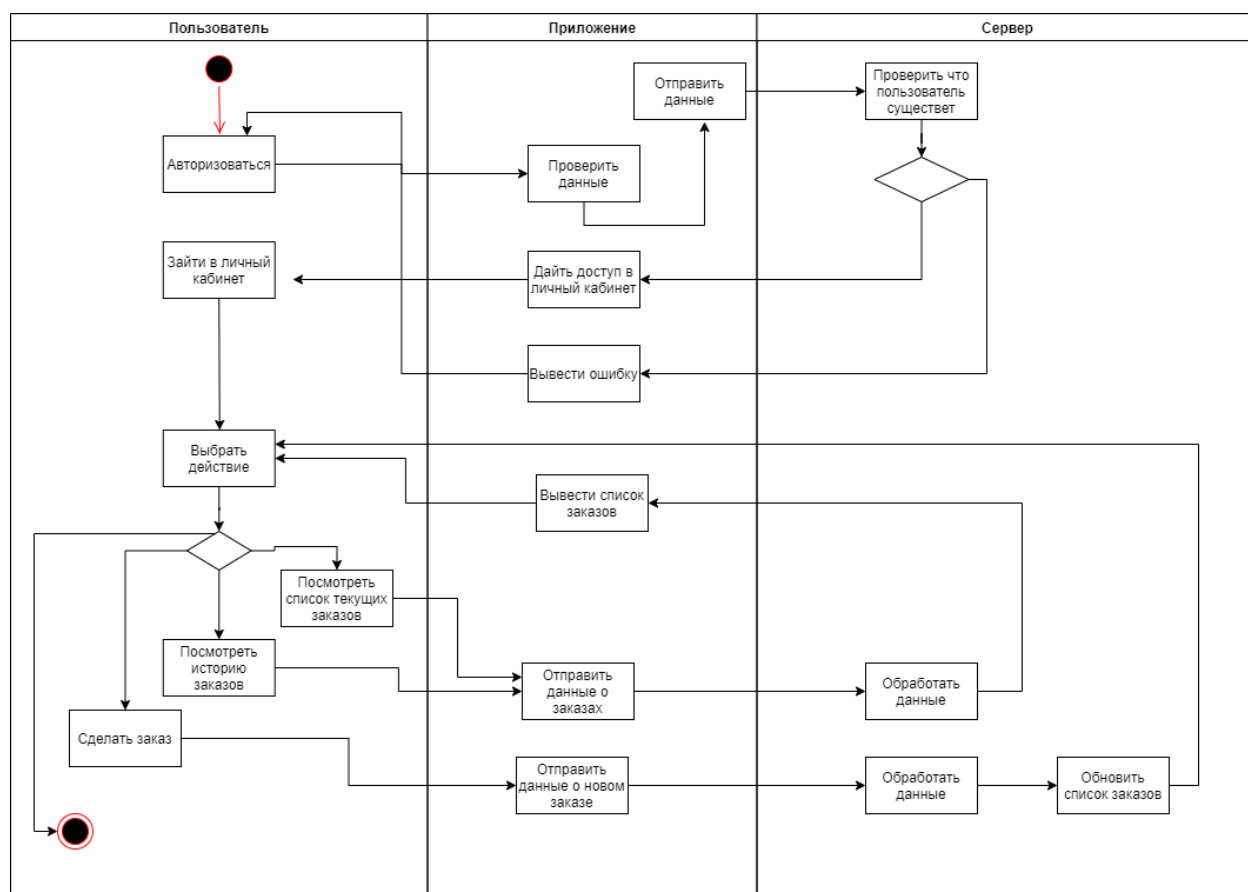


Рис.7 Диаграмма активностей клиента

С помощью диаграммы активности, представленной на Рисунке 8, была продемонстрирована работа менеджера с приложением.

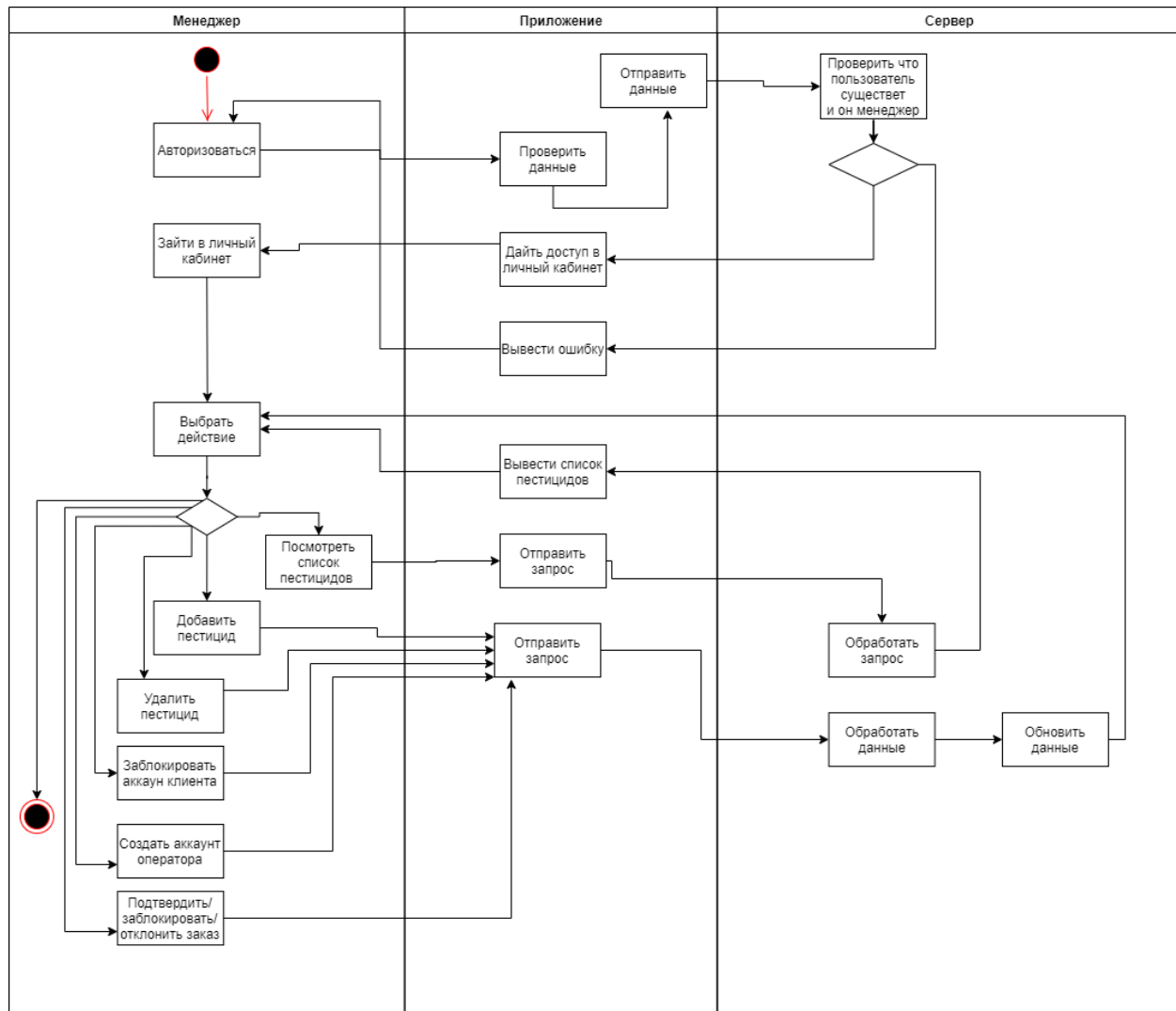


Рис. 8 Диаграмма активностей менеджера

С помощью диаграммы активности, представленной на Рисунке 9, была продемонстрирована работа оператора с приложением.

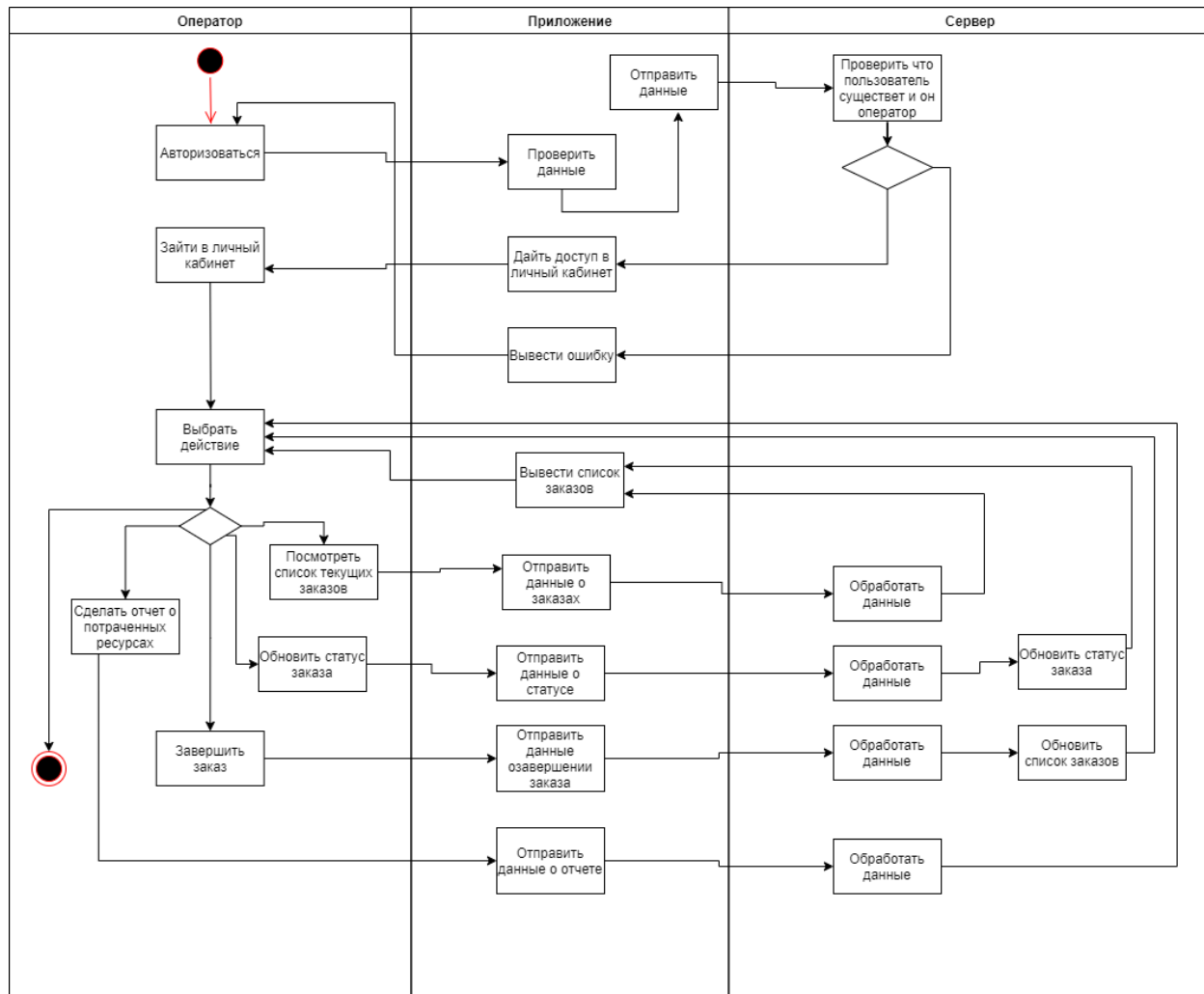


Рис. 9 Диаграмма активностей оператора

2.5.7. Диаграмма классов

В ходе разработки приложения была составлена данная диаграмма сущностей на Рисунке 10

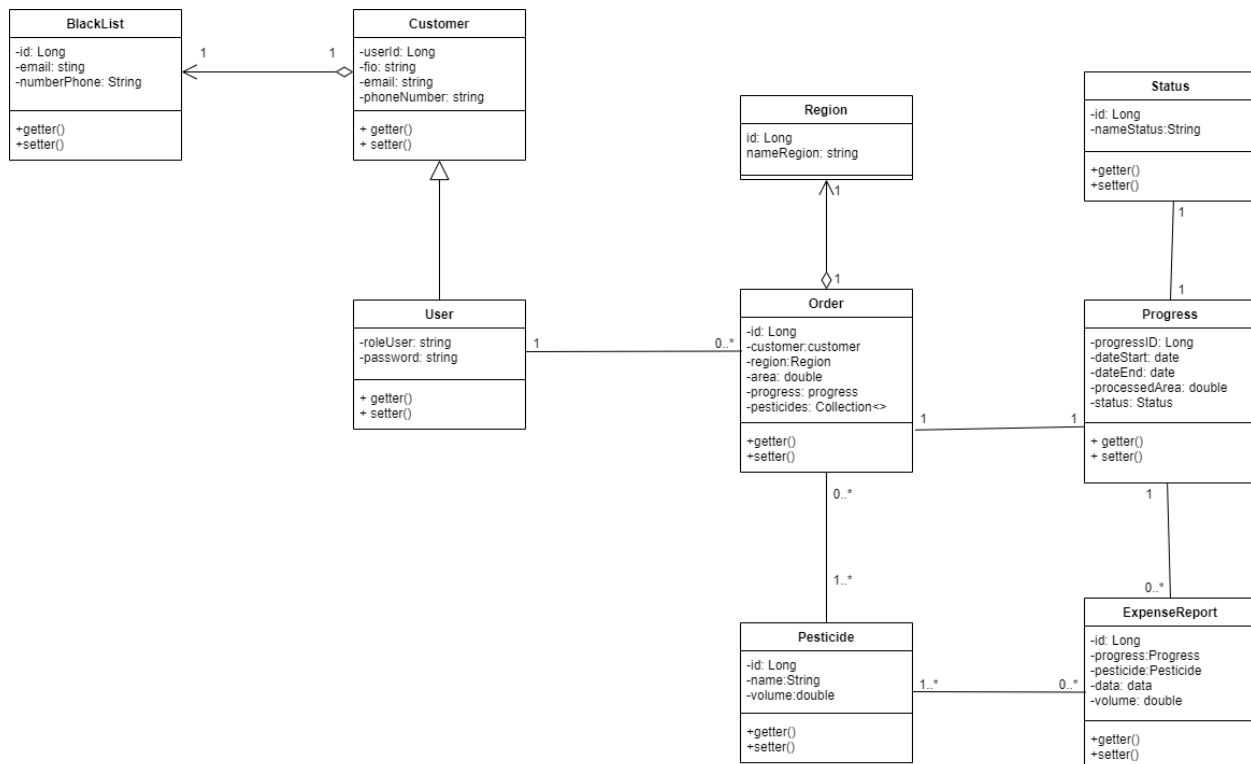


Рис.10 Диаграмма классов

Это классы:

- Customer. С полями id, ФИО, почты, номера телефона.
- User. С полями от Customer, а так же его роль, и пароль
- Order. С полями id, customer, площадь, прогресс, пестициды
- Region. С полями id, название региона.
- Progress. С полями id, дата начала, дата окончания, выполненная площадь, статус.
- Pesticide. С полями id, название, кол-во
- Status. С полями id, название
- ExpenseReport. С полями id, progress, pesticide, дата, кол-во потраченного
- BlackList. С полями id, email, phone

Так же у каждого класса, есть для всех полей get() и set().

2.5.8. Диаграмма объектов

Для лучшего понимания диаграммы классов была составлена диаграмма объектов, представленная на Рисунке 11.

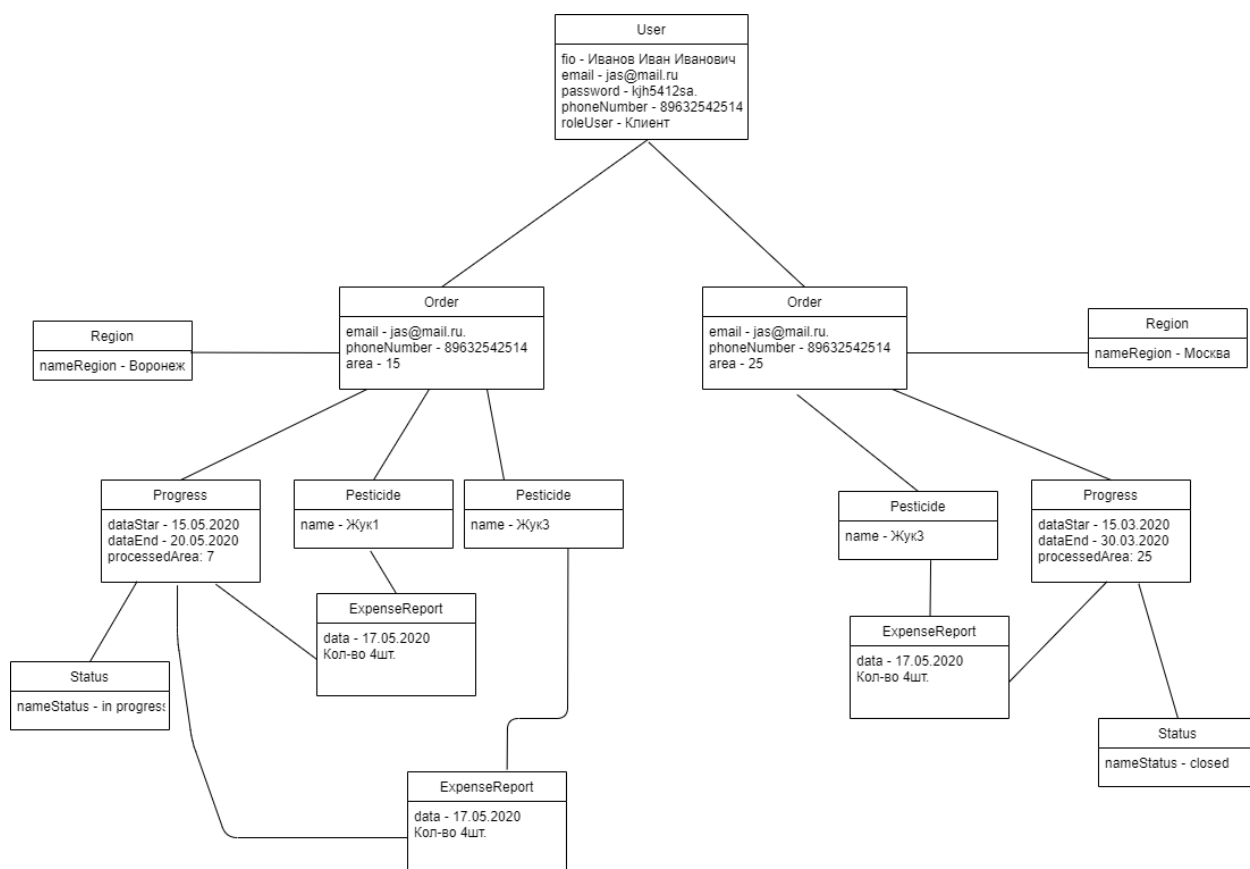


Рис.11 Диаграмма объектов

2.5.9. ER-диаграмма

Для работы с базой данной была спроектирована ER-диаграмма, представленная на Рисунке 12.

На ней представлены 8 сущностей:

- User – пользователь

Атрибуты:

- User_id – PK
- Role – роль пользователя
- Fio – ФИО пользователя

- Email –почта пользователя
- Pass- пароль пользователя
- Phone – номер телефона пользователя

- Order – заказ

Атрибуты:

- Order_id – PK
- Progress_id – FK на таблицу Progress
- Region_id – FK на таблицу Region
- User_id – FK на таблицу User
- Area – площадь

- Progress – прогресс заказа

Атрибуты:

- Progress _id – PK
- Status_id – FK на таблицу Status
- Data_start –Дата начала заказа
- Data_end –Дата окончания заказа
- Processed_area – в для заказа
- Area – площадь
- Status – статус заказа

- Region – регион заказа

Атрибуты:

- Region _id – PK

- Name –Название региона
- Pesticide – пестициды заказа

Атрибуты:

- Pesticide_id – PK
- Name – название
- value – кол-во
- Status – статус заказа

Атрибуты:

- Status_id – PK
- Name – название

- ExpenseReport – отчет заказа

Атрибуты:

- Report_id – PK
- Progress_id – FK на таблицу Progress
- Data – дата
- value – кол-во

- BlackList – черный список клиентов

Атрибуты:

- user_id – PK
- email – почта
- phone - телефон

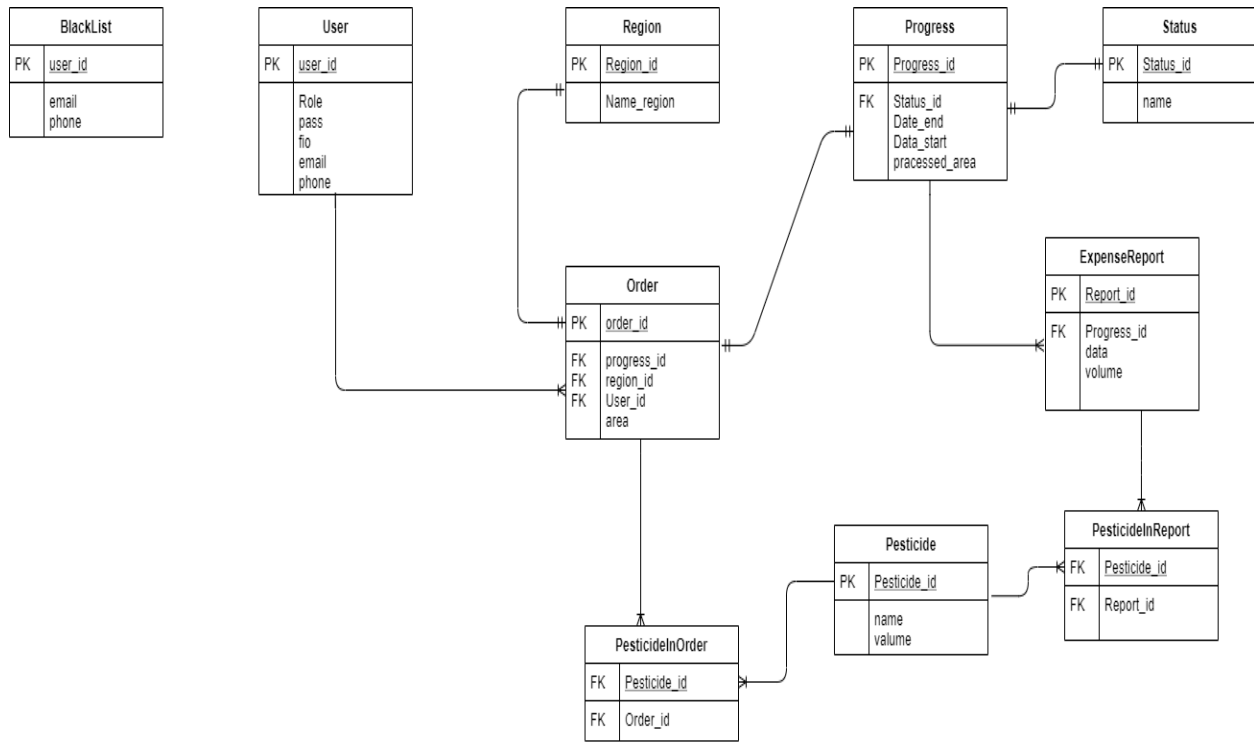


Рис. 12 ER-диаграмма

2.5.10. Диаграмма развертывания

На Рисунке 13 приведена диаграмма развертывания, визуализирующая элементы и компоненты разрабатываемой системы, которые существуют на этапе ее исполнения.

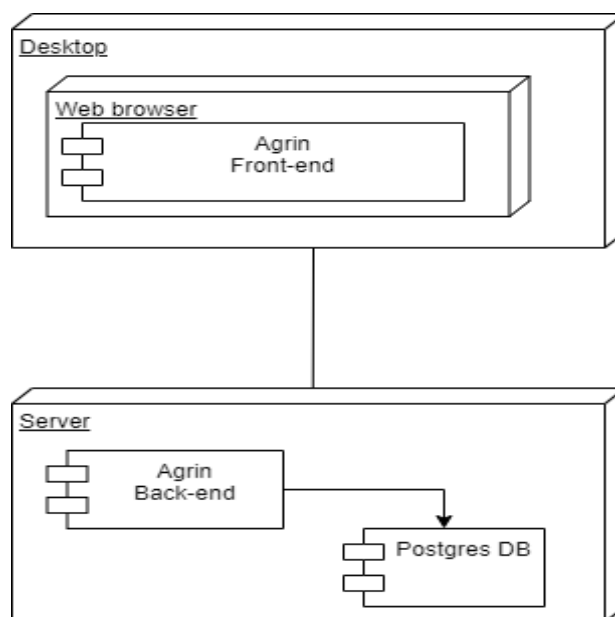


Рис. 13 Диаграмма разворачивания

3. Реализация приложения

3.1. Анализ средств реализации

В качестве средств реализации, были выбраны следующие технологии:

- Spring Boot - универсальный фреймворк с открытым исходным кодом для Java-платформы. Обладает большим функционалом, но его наиболее значимыми особенностями являются: управление зависимостями, автоматическая конфигурация и встроенные контейнеры сервлетов.
- Angular - это фреймворк, построенный на основе HTML и JavaScript, двух технологий, давно используемых в веб-разработке. Он позволяет использовать привычные редакторы и расширения для браузеров. Версия Angular CLI (Command Line Interface) стандартизирует структуру, позволяет создать сущности внутри приложения, а также автоматизировать его сборку.
- В качестве СУБД была выбрана PostgreSQL в силу относительной легковесности и высокой производительности, а также за счёт поддержки данной СУБД schema-less данных, такие как JSON.
- Tomcat - контейнер сервлетов с открытым исходным кодом. Реализует спецификацию сервлетов, спецификацию JavaServer Pages (JSP) и JavaServer Faces (JSF). Написан на языке Java. Tomcat позволяет запускать веб-приложения. Tomcat используется в качестве самостоятельного веб-сервера, в качестве сервера контента в сочетании с веб-сервером Apache HTTP Server.

- Для реализации Android-приложения был выбран высокоуровневый язык программирования Java, так как он является проверенным временем выбором.

3.2. Разработка Frontend

3.2.1. Начальная страница

На Рисунке 14 и Рисунке 15 представлена главная страница приложения.

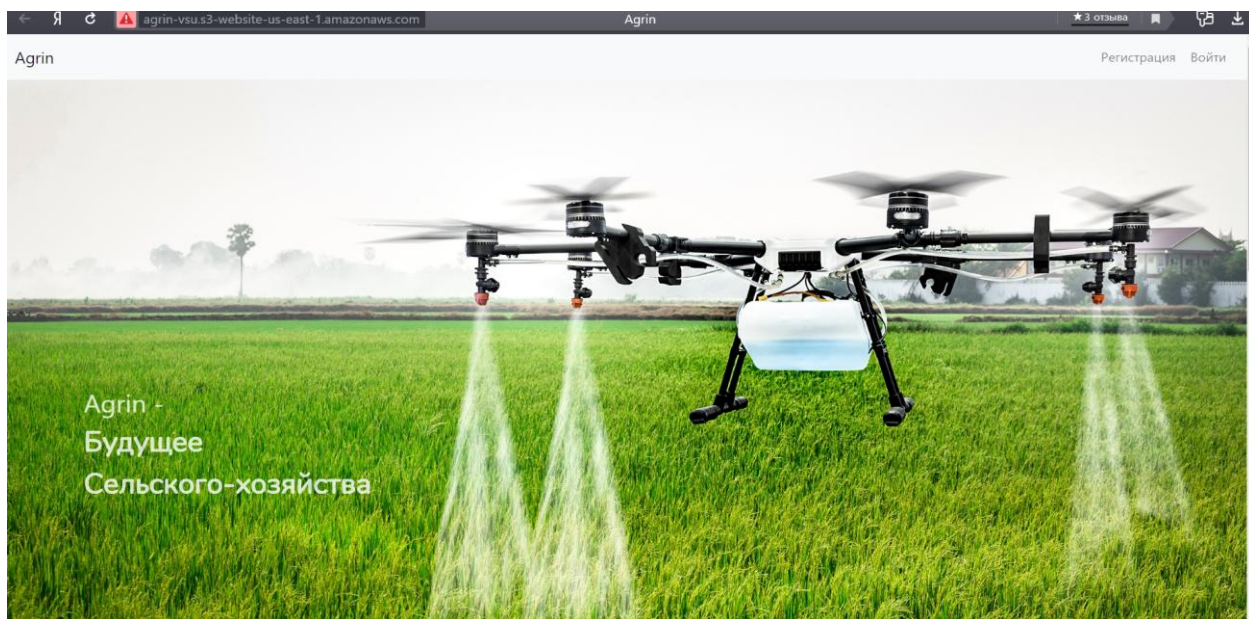


Рис. 14 Начальная страница

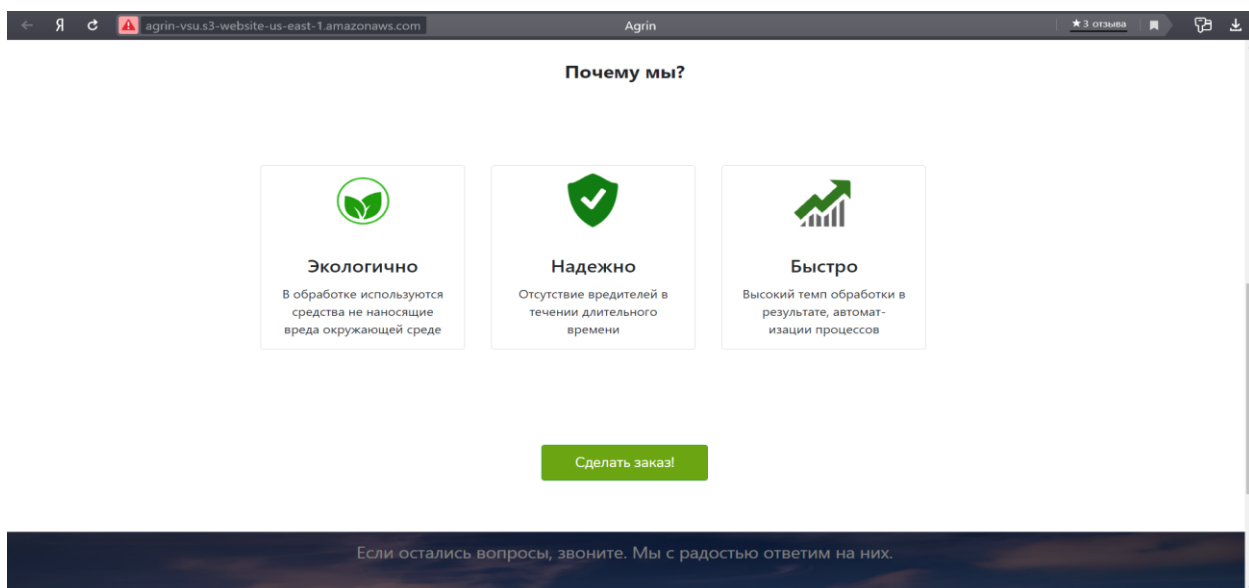


Рис. 15 Начальная страница

3.2.2. Создание заказа

При нажатии кнопки “Сделать заказ”, пользователю показывается форма, где он вносит свои данные. (Рисунок 16).

The screenshot shows a web browser window with the URL `agrin-vsus3-website-us-east-1.amazonaws.com/create/order`. The page title is "Agrin" and there are links for "Регистрация" (Registration) and "Войти" (Login). The main heading is "Оформление заказа" (Order Form). The form contains the following fields:

- ФИО:** Иван Иванович Иванов
- Телефон:** +7 9405531732 (with a note: "10 цифр, без начальной 8/+7")
- Mail:** @ cust@gmail.com
- Размер:** 10 (with a note: "Размер земельного участка, который нужно обработать, укажите в гектарах")
- Регион:** Тамбовская область (dropdown menu)
- Дата:** 2020-06-17 (with a note: "Желательная дата для начала обработки" and a calendar icon)

A green button labeled "Заказать" (Order) is located at the bottom of the form.

Рис.16 Создание заказа

3.2.3. Регистрация/Авторизация пользователя

При нажатии на навигационной панели “Регистрация”, пользователю показывается форма для ввода своих данных для регистрации. (рисунок 17)

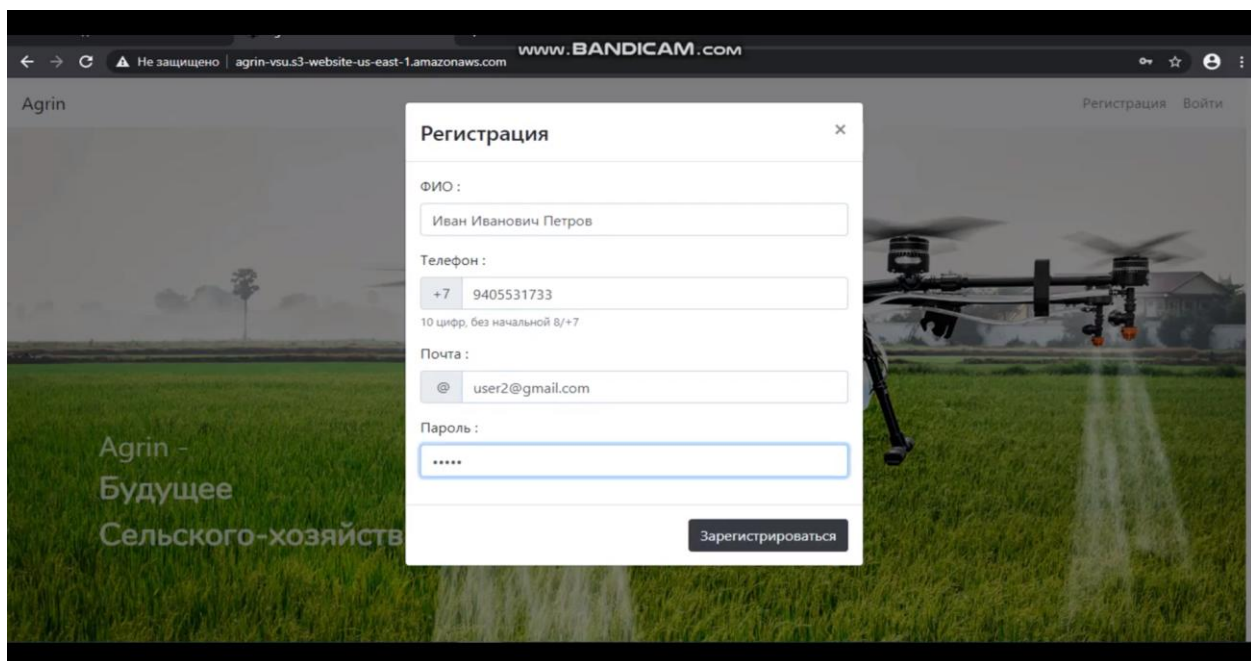


Рис. 17 Регистрация пользователя

Далее пользователь вводит свои данные, которые он только что зарегистрировал, в поля для входа. И система авторизует пользователя.

3.2.4. Авторизованный пользователь

Пользователь может редактировать свой профиль, введя новые данные.

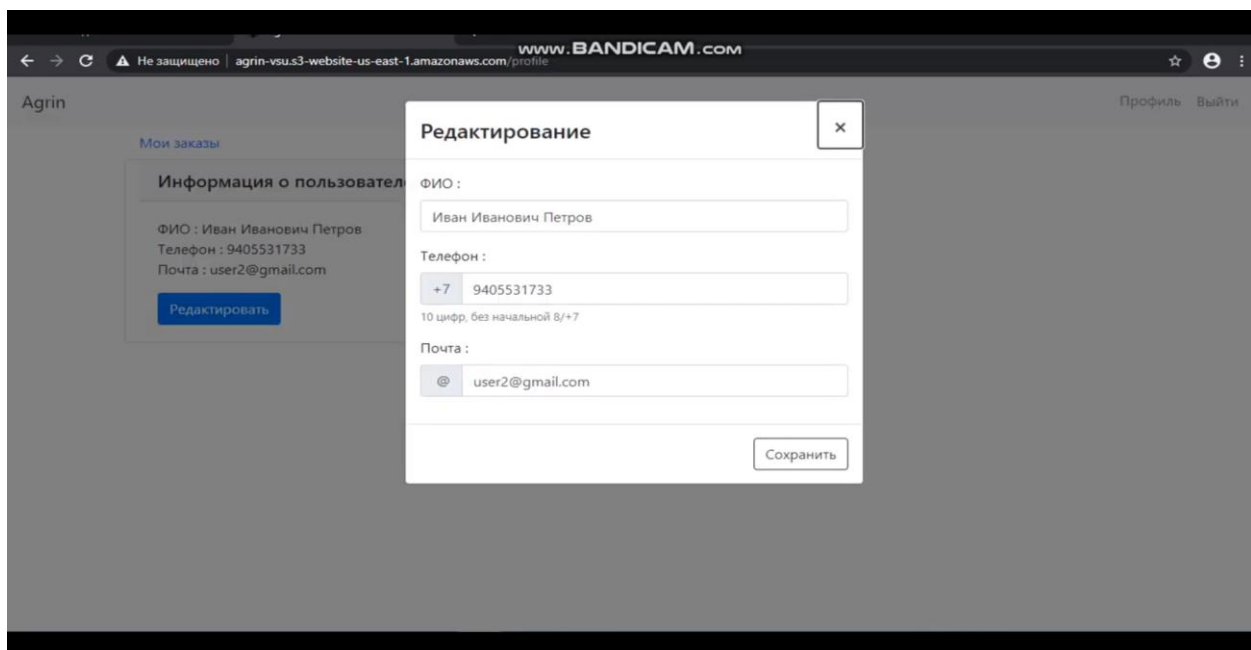


Рис.18 Редактирование профиля

Если пользователь авторизован, он точно так же сможет сделать заказ, по такому же принципу как и не авторизованный. Далее в профиле он сможет посмотреть свои текущие заказы (Рисунок 19).

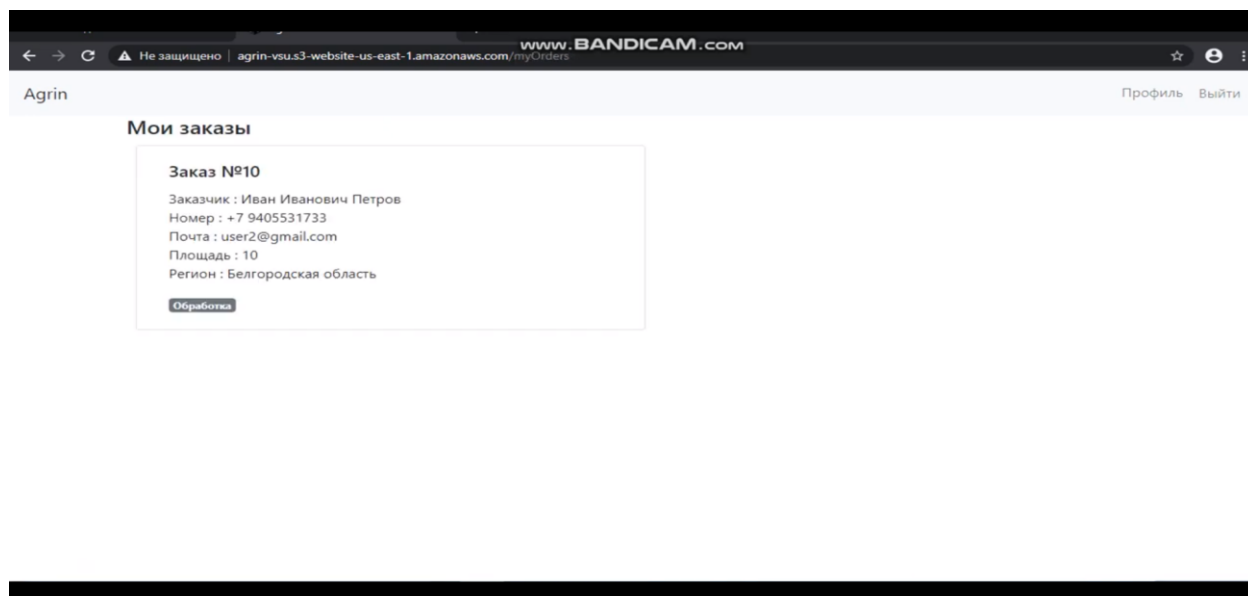


Рис.19 Просмотр заказов

3.2.5. Роль менеджера

Менеджер вводит в поля авторизации свой логин и пароль admin, и авторизуется с ролью менеджера.

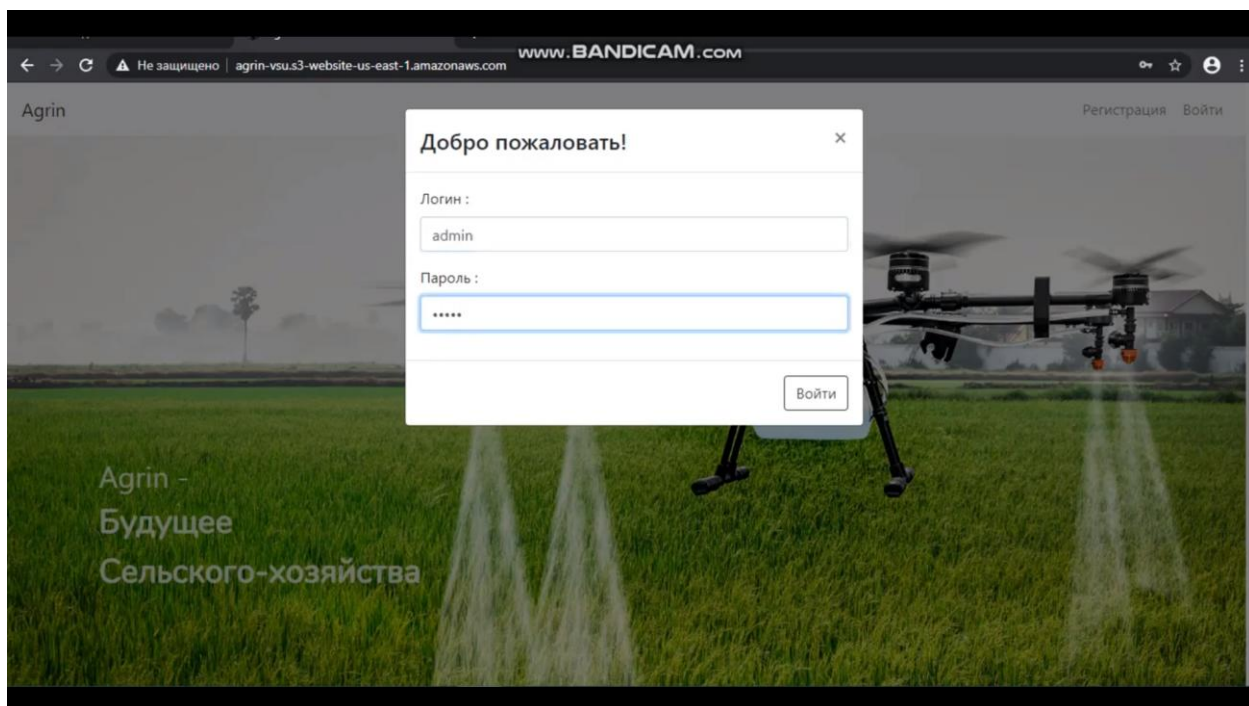


Рис.20 Вход менеджера

Как только менеджер авторизуется, его перебросит на главную страницу менеджера. Его навигационная панель будет содержать такие функции:

- Текущие заказы
- Пестициды
- Заказы
- Профиль
- Операторы
- Выйти

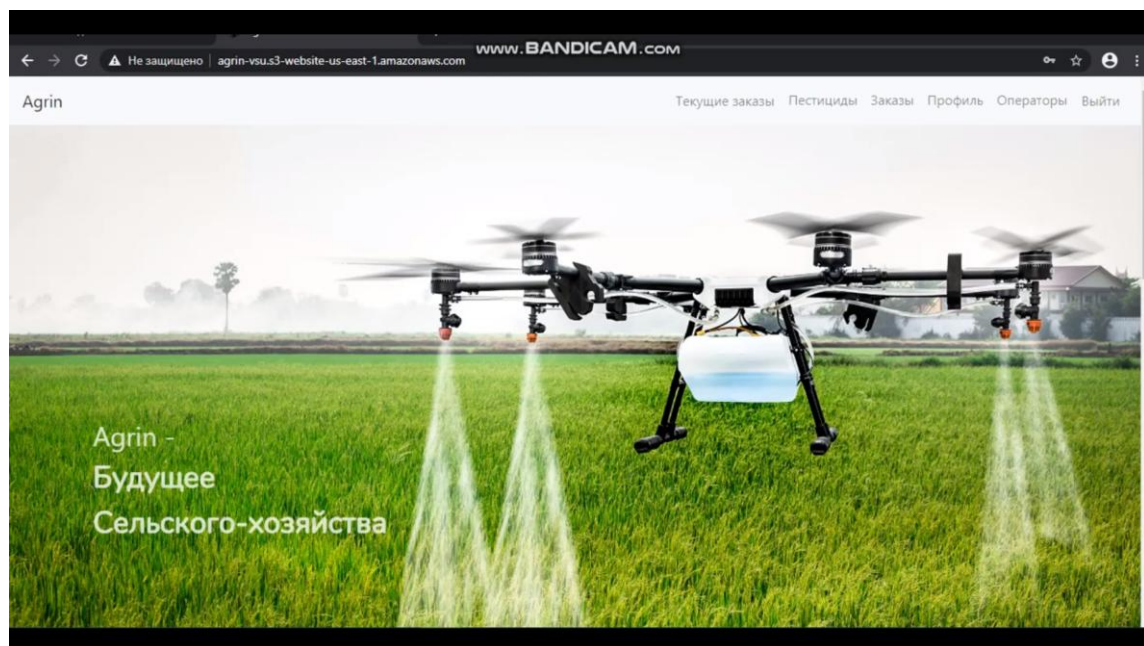


Рис. 21 Главная страница менеджера

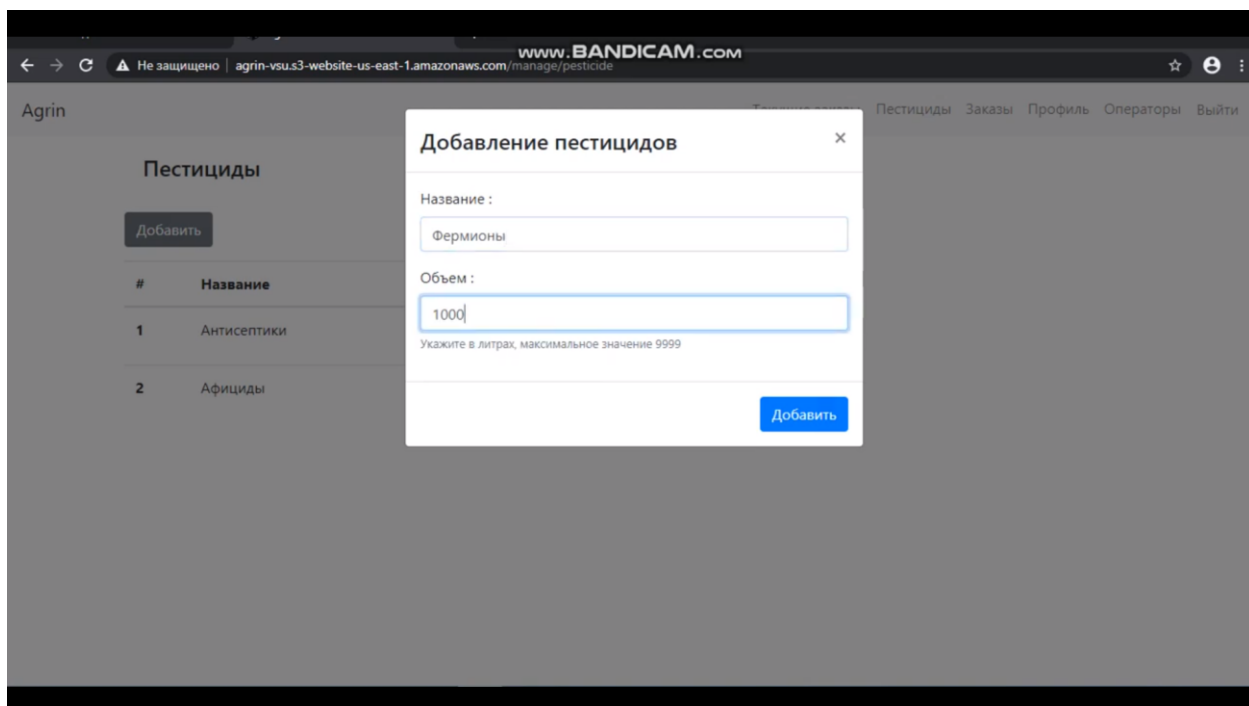
Менеджер может посмотреть список текущих пестицидов

 The screenshot shows the 'Пестициды' (Pesticides) section of the Agrin manager. It includes a 'Добавить' (Add) button and a table with the following data:

#	Название	Объем	Действие
1	Антисептики	1000	Изменить
2	Афициды	880	Изменить

Рис. 22 Список пестицидов

Так же после этого менеджер может добавить в этот список необходимые пестициды



Менеджер так же может смотреть поступившие заказы. Он может либо принять его, либо отклонить, или же и вовсе заблокировать клиента.

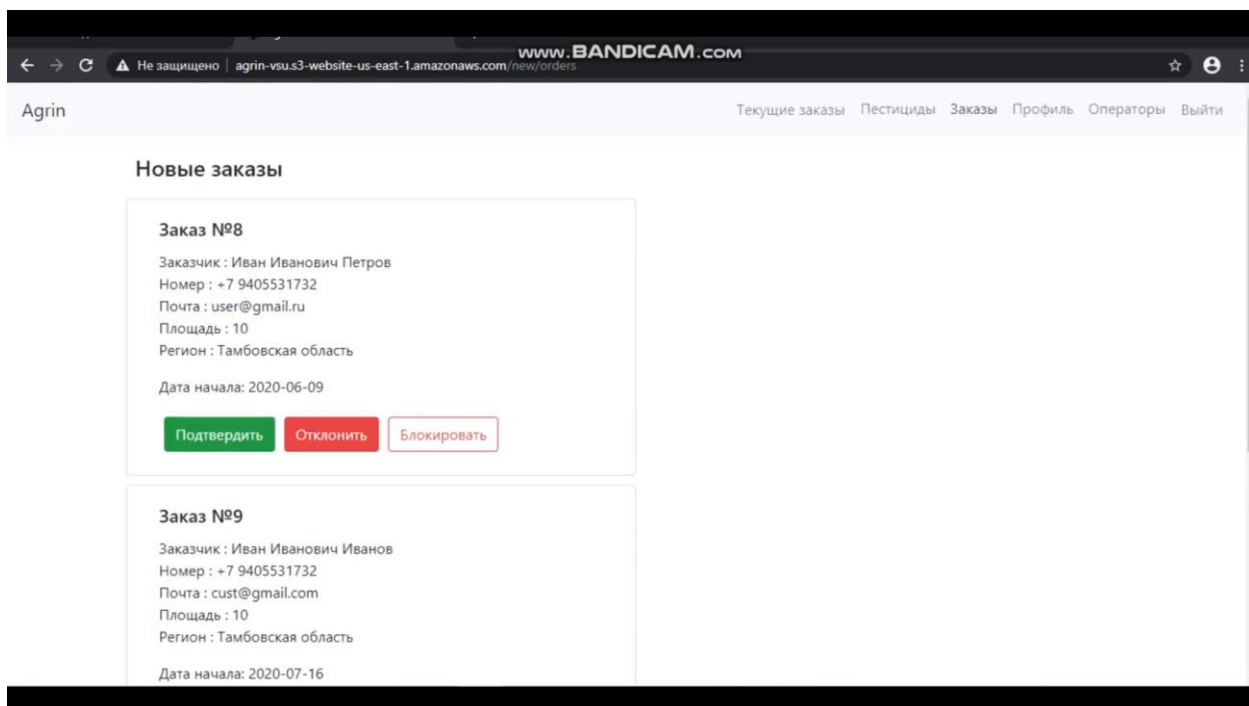


Рис. 22 Просмотр поступивших заказов

При подтверждении заказа, менеджер выбирает пестициды для этого заказа, дату начала заказа и дату завершения.

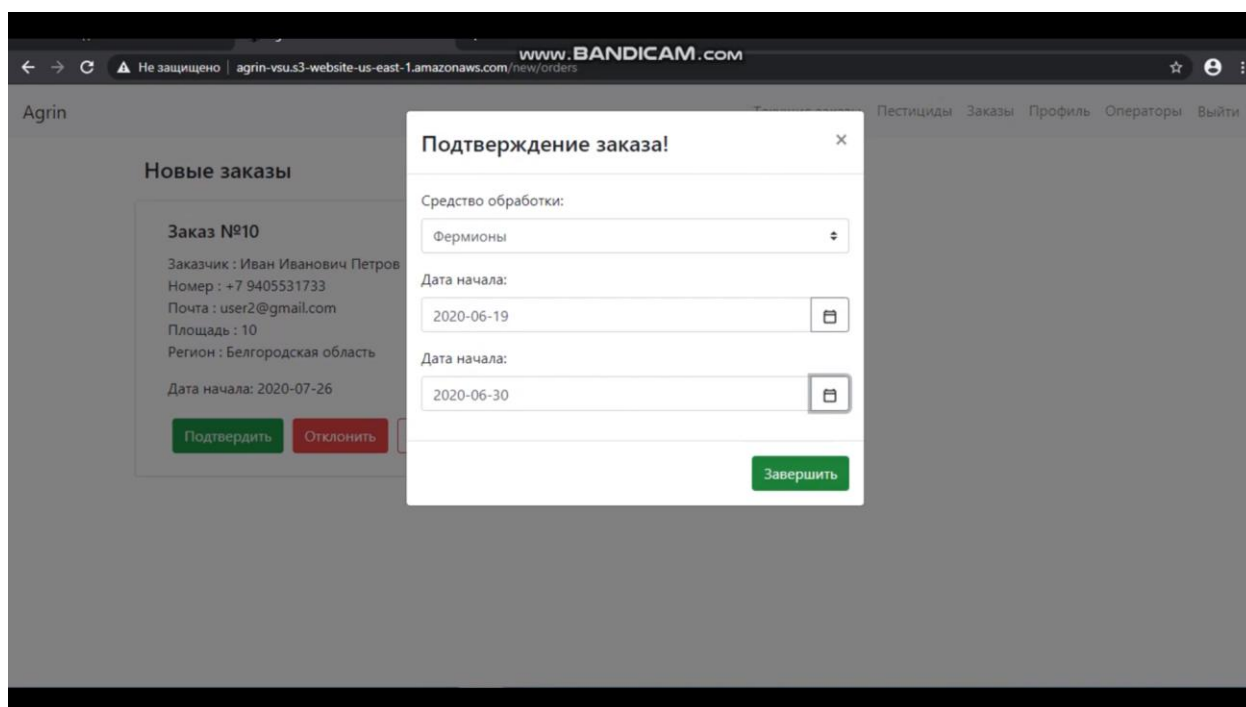


Рис. 23 Подтверждение заказа

Далее этот заказ будет в списке текущих заказов.

Так же менеджер может создавать аккаунты операторам, кликнув по “Операторы”

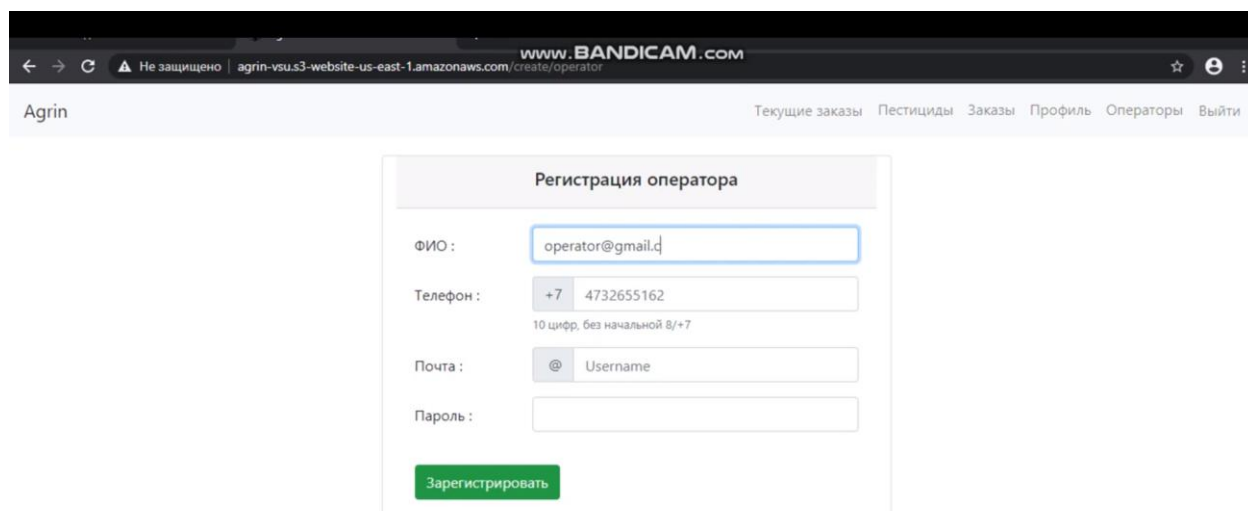


Рис. 24 Регистрация оператора

Далее высветится сообщение о успешной регистрации оператора.

3.2.6. Роль оператора

Оператор вводит принадлежащие ему данные и входит в сервис с ролью оператора. У него следующие функции в навигационной панели:

- Текущие заказы
- Профиль
- Выйти

Кликнув по текущим заказам, оператор видит все его заказы.

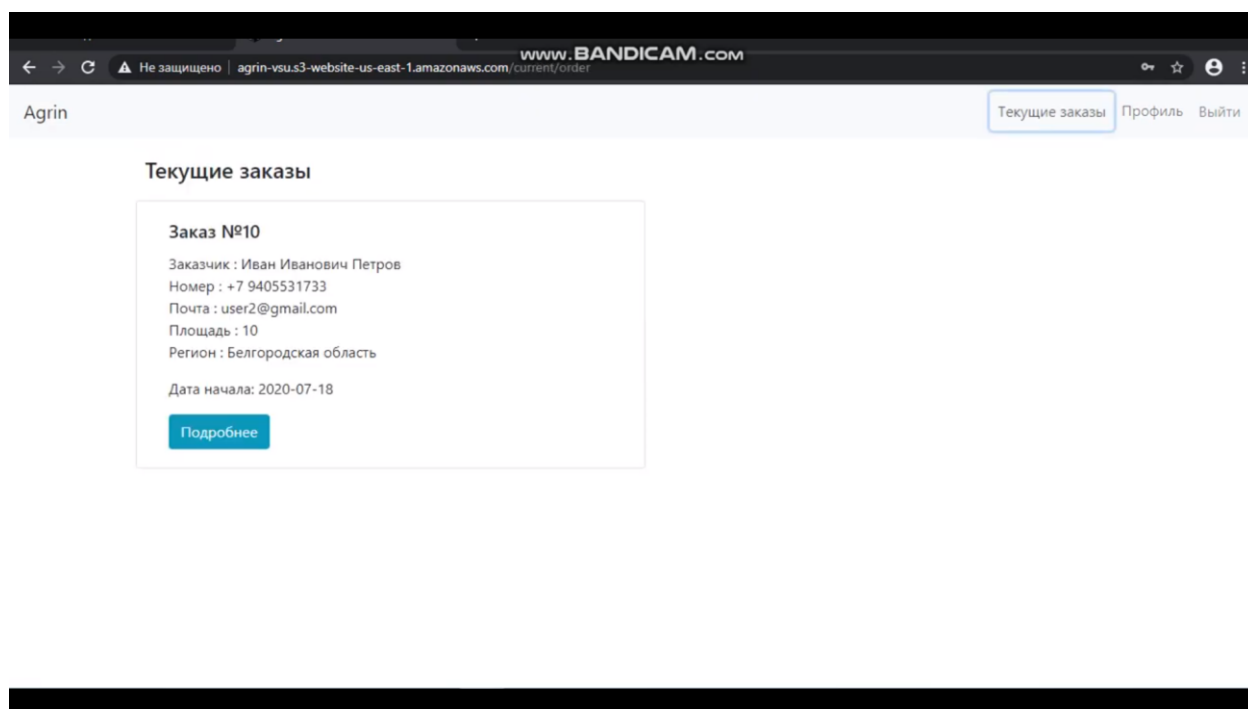


Рис. 25 Текущие заказы оператора

Оператор может выбрать какой-либо заказ(в данном случае он только один) и сделать по нему отчет о затраченных ресурсах.

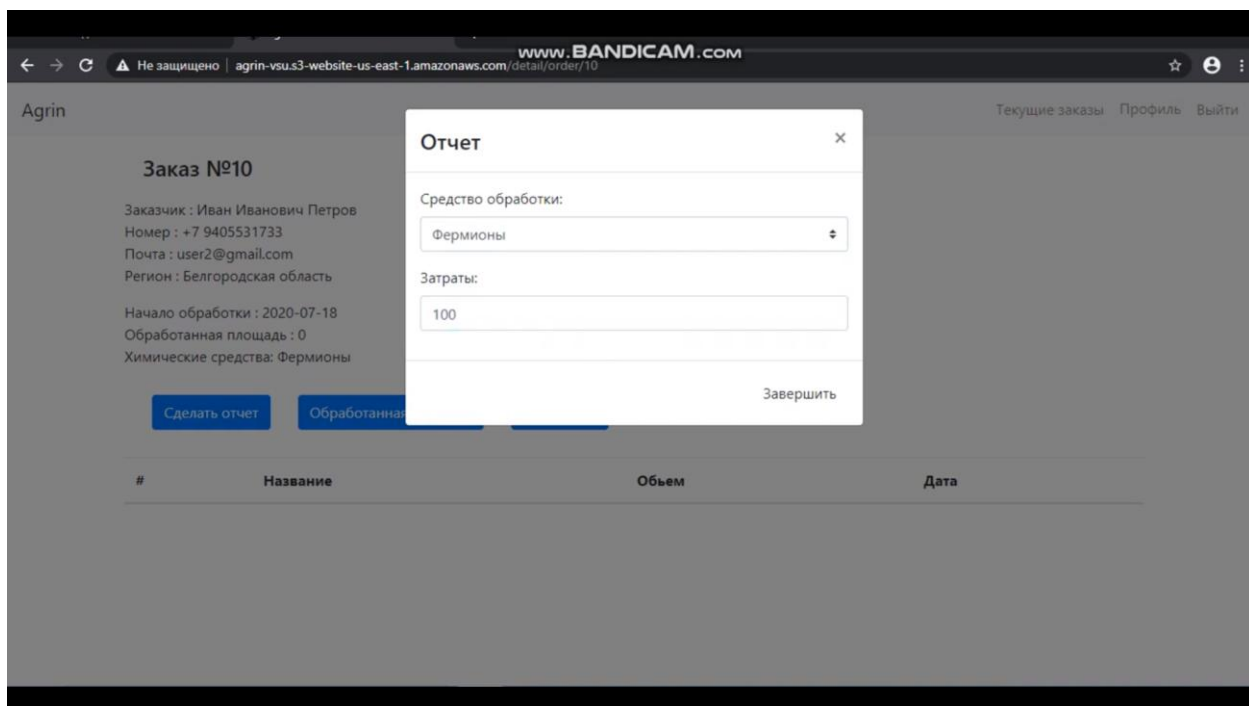


Рис. 26 Отчет о потраченных ресурсах

Внизу будет отображаться все отчеты об этом заказе. И оператор может указывать прогресс заказа (обработанную площадь).

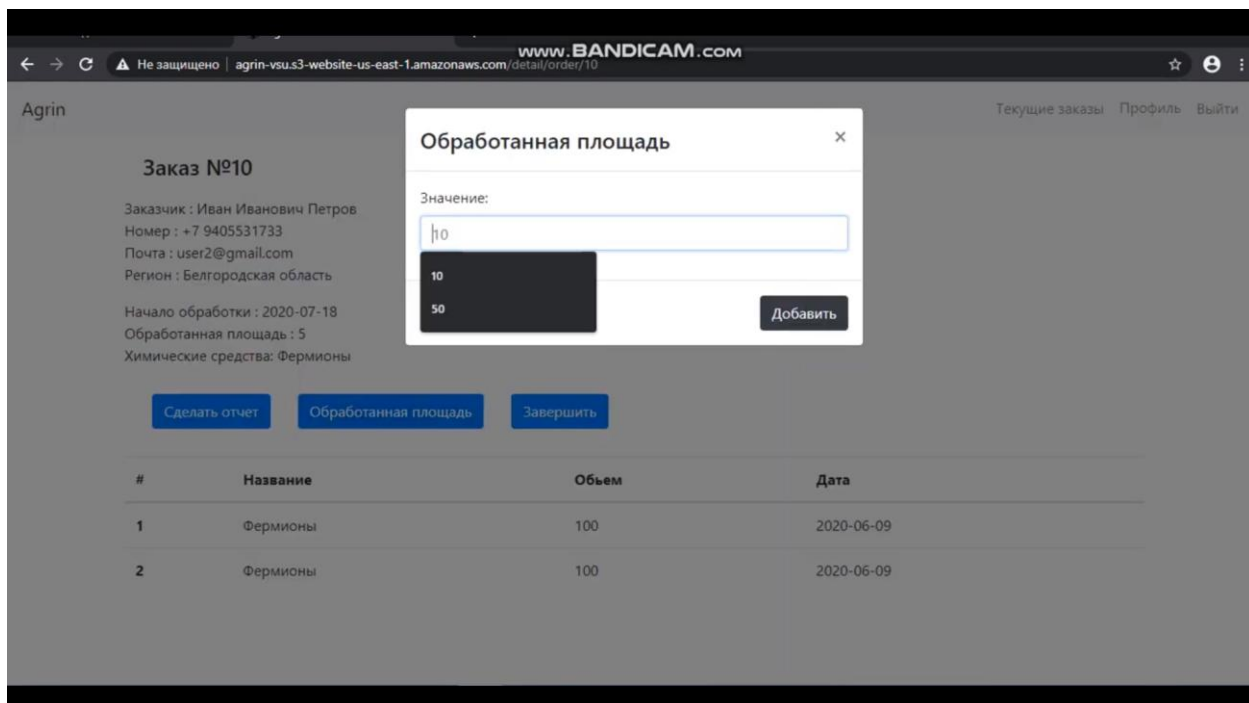


Рис. 27 Обработанная площадь

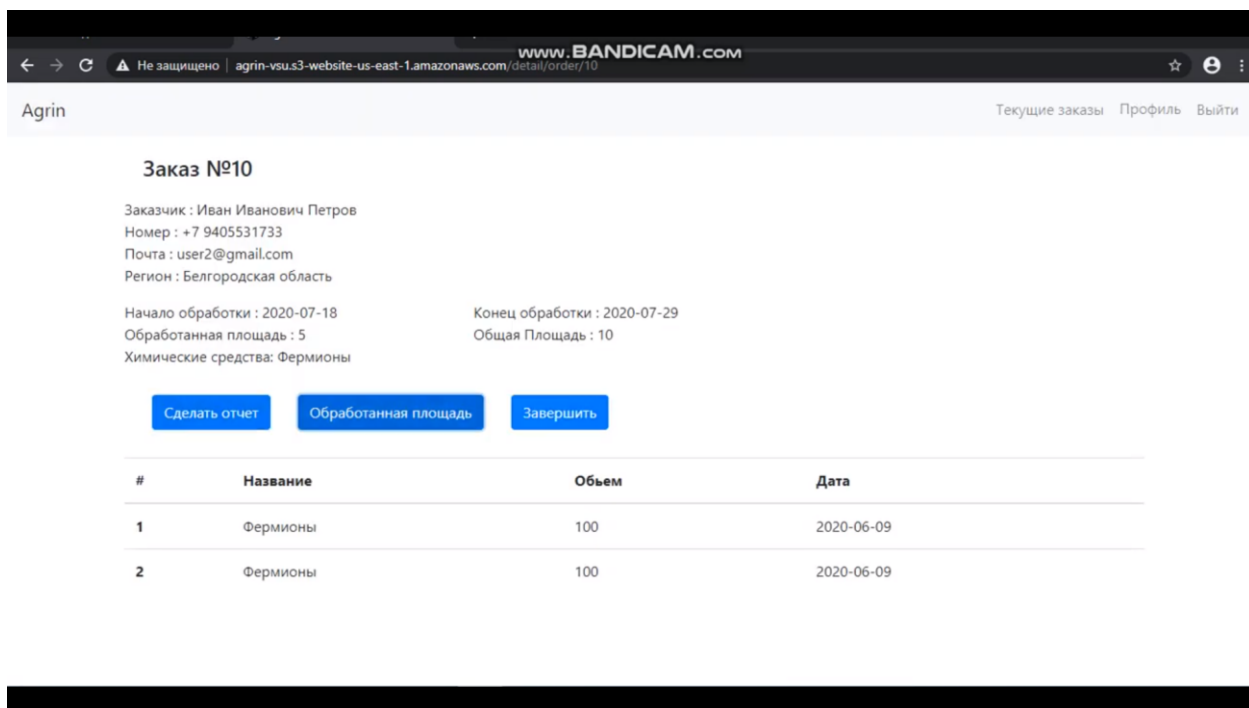


Рис. 28 Список всех отчетов

И в конце оператор может завершить заказ, и статус заказа изменится на “Завершен”. И если зайти за пользователя, который сделал этот заказ, то он увидит, что его заказ завершен.

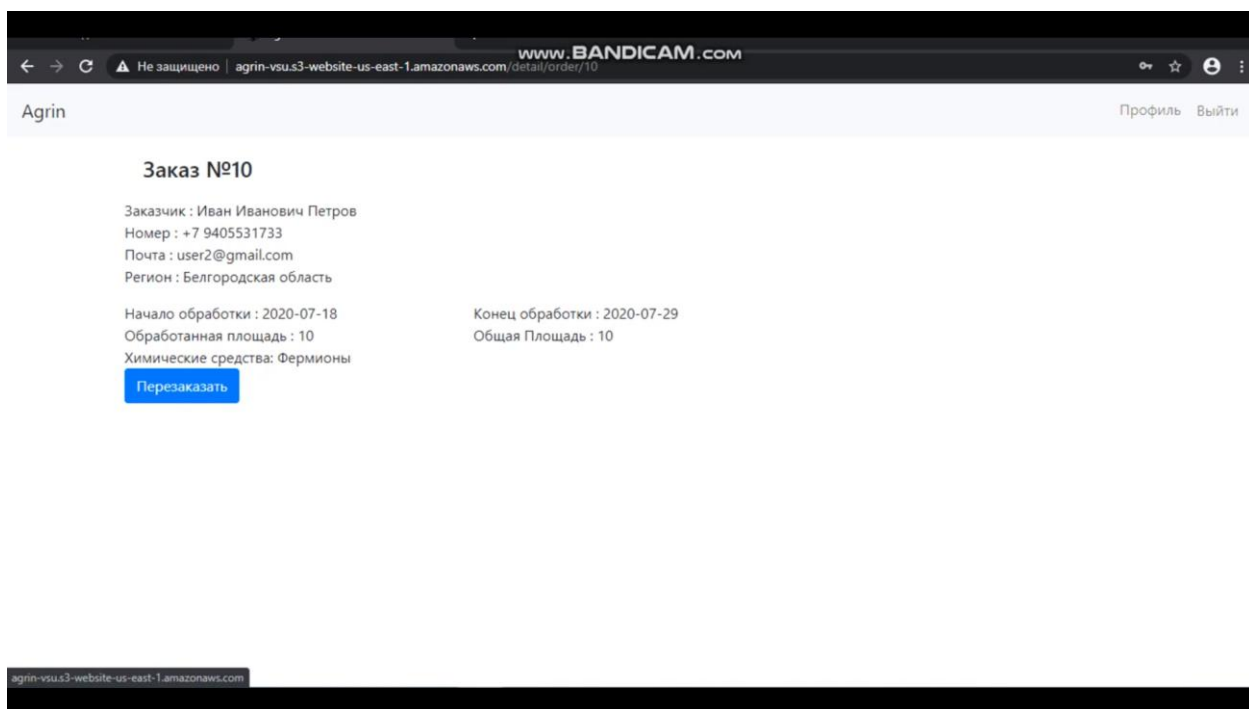


Рис. 30 Завершенный заказ пользователя

3.2.7. Блокировка реквизитов

Менеджер при просмотре поступивших заказов, может заблокировать реквизиты заказа. И пользователь, если введет эти данные при заказе не сможет сделать его и увидит соответствующее сообщение.

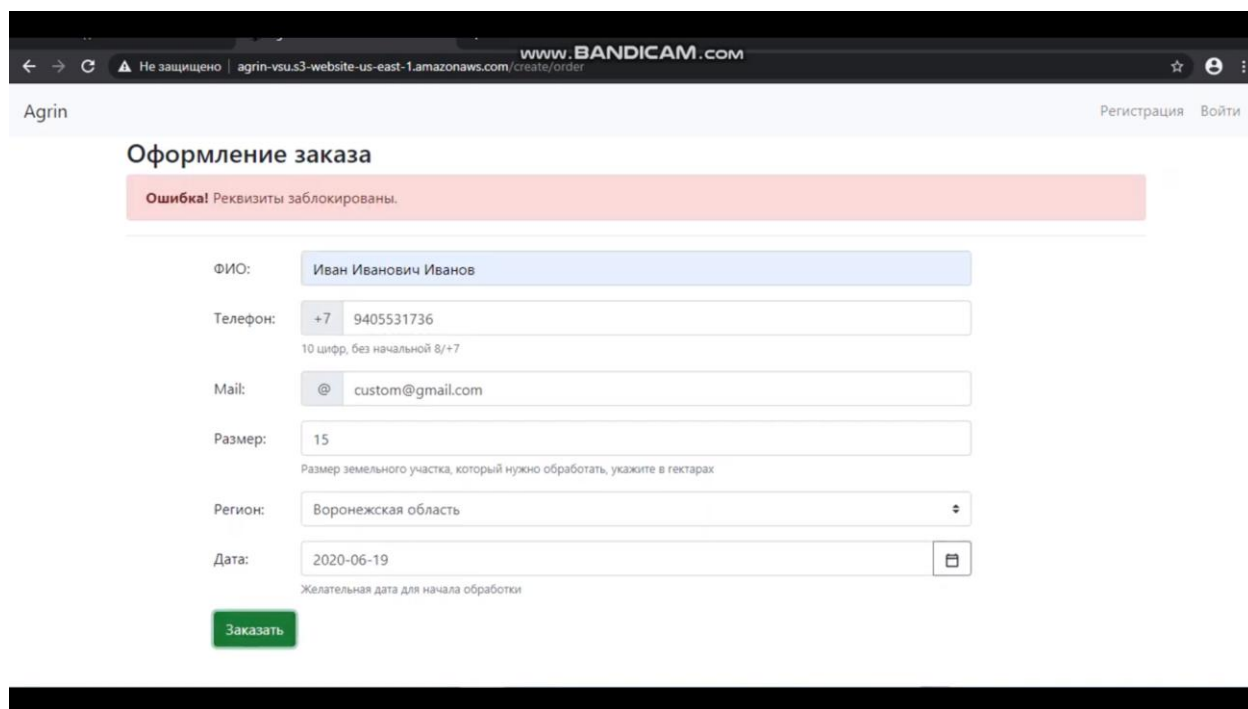


Рис 31 Блокировка реквизитов

3.3. Разработка backend

Серверная часть приложения написана на языке Java с использованием фреймворка Spring. Этот фреймворк был выбран по следующим причинам:

- Предоставление каркаса приложения
- Слабосвязанность объектов в следствии поддержки внедрения зависимости

Архитектура сервера состоит из компонентов:

- Контролер
- Сервис
- Репозиторий

Компонент контроллер используется для принятия и обработки запросов, создания модели и ее передача в виде ответа. Каждому контроллеру

присваивается уникальный url. При поступлении запроса DispatcherServlet проверит все контроллеры на соответствия с url, указанным в запросе, и при совпадении передаст обработанные данные контроллеру. Сервис представляет бизнес-логику приложения. Реализованные сервисы: Репозиторий – это хранилище данных. Он представлен в виде интерфейсов, которые обеспечивают операции поиска, сохранения и удаления данных каждой сущности в базе данных. Каждая конечная точка в приложении, исключая публичные, защищена с использованием технологии JWT.

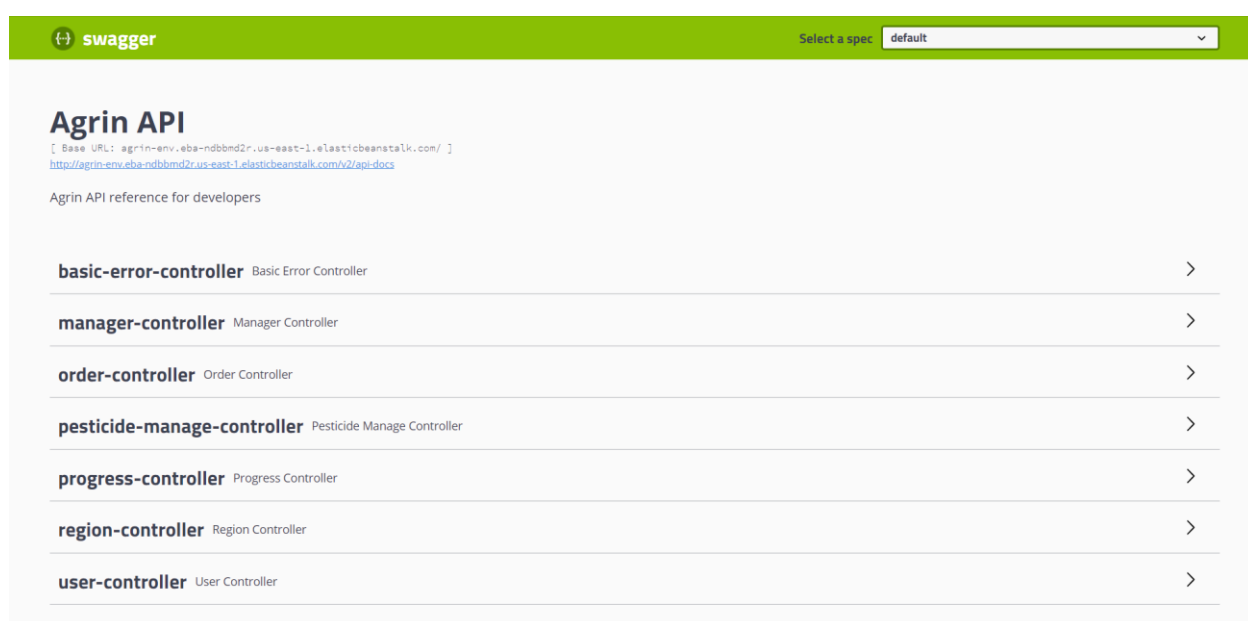


Рис 32 Панель swagger

3.4. Тестирование

3.4.1. Дымовое тестирование

Для данного типа тестирования необходимо было проверить работоспособность приложения на следующих основных сценариях:

- Авторизация пользователя
- Получение информации о заказах
- Получение информации о профиле пользователя

- Получение информации о поступивших заказах
- Получение информации о подтвержденных заказах
- Получение информации о пестицидах
- Получение информации текущих заказах оператора
- Получение информации о прогрессе заказа
- Выбор пестицидов и даты при подтверждении заказа

Результаты, полученные в ходе тестирования, представлены в Таблицы

1.

Таблица 1 – результаты дымового тестирования

Авторизация пользователя	Пройден
Получение информации о заказах	Пройден
Получение информации о профиле пользователя	Пройден
Получение информации о поступивших заказах	Пройден
Получение информации о подтвержденных заказах	Пройден
Получение информации текущих заказах оператора	Пройден
Получение информации о прогрессе заказа	Пройден
Выбор пестицидов и даты при подтверждении заказа	Пройден
Получение информации о пестицидах	Пройден

Таблица 1 Дымовое тестирование

3.4.2. Интеграционное тестирование серверной части

Для данного типа тестирования необходимо было проверить работоспособность приложения на следующих основных сценариях:

- Регистрация пользователя
- Создание заказа
- Редактирование профиля
- Добавление пестицидов
- Удаление пестицидов
- Подтверждение заказа
- Блокировка реквизитов
- Отклонение заказа
- Регистрация оператора
- Создание отчета оператором
- Изменение прогресса заказа оператором
- Завершение заказа оператором

Результаты, полученные в ходе тестирования представлены в Таблице

2.

Регистрация пользователя	Пройден
Создание заказа	Пройден
Редактирование профиля	Пройден
Добавление пестицидов	Пройден
Удаление пестицидов	Пройден
Подтверждение заказа	Пройден
Блокировка реквизитов	Пройден
Отклонение заказа	Пройден
Регистрация оператора	Пройден
Создание отчета оператором	Пройден
Изменение прогресса заказа оператором	Пройден
Завершение заказа оператором	Пройден

Таблица 2 Интеграционное тестирование серверной части

3.4.3. Юзабилити тесты

Для проведения юзабилити тестирования было отобрано 3 человека, не пользовавшиеся заранее приложением. Для данного тестирования необходимо проверить следующие основные сценарии взаимодействия пользователя с приложением:

- Регистрация пользователя
- Авторизация пользователя
- Просмотр профиля
- Редактирование профиля
- Создание заказа
- Получение информации о своих заказах

Юзабилити тесты представлены на таблице 3.

Сценарий	Пользователь 1	Пользователь 2	Пользователь 3
Регистрация пользователя	Пройден	Пройден	Пройден
Авторизация пользователя	Пройден	Пройден	Пройден
Просмотр профиля	Пройден	Пройден	Пройден
Редактирование профиля	Пройден	Пройден	Пройден
Создание заказа	Пройден	Пройден	Пройден
Получение информации о своих заказах	Пройден	Пройден	Пройден

Таблица 3 Юзабилити тесты

Заключение

В результате работы было реализовано web приложение, которое позволяет создавать и отслеживать заказы обработки поля для клиента, для менеджера оно позволяет принимать заказы, контролировать объем пестицидов, создавать учетные записи операторов, а для оператора, просматривать текущие заказы, делать отчеты о работе, изменять прогресс заказа и завершать его.

Были выполнены следующие задачи:

1. Разработана Front-end часть приложения, которая удобная в использовании пользователю;
2. Разработана Back-end часть приложения, развернутая на удаленном сервере приложений;
3. Была создана связь между Front-end и Back-end частями приложения с помощью REST API;
4. Разработана база данных, расположенная на удаленном сервере;

Список литературы

1. JavaRush – Онлайн справочник по Java [сайт] – URL: <https://javarush.ru/> (дата обращения: 2.04.2020 – 8.06.2020)
2. Уроки по разработке Android приложений [сайт] – URL: <https://www.fandroid.info> (дата обращения: 28.08.2019 – 17.09.2019)
3. Мэтт Вайсфельд. Объектно-ориентированное мышление – Санкт-Петербург: Питер, 2014. – 304 с.
4. Spring Framework Documentation [сайт] – URL: <https://docs.spring.io/spring/docs/current/spring-framework-reference/> (дата обращения: 30.04.2020 – 01.06.2020)
5. Форум stackoverflow Помощь в решениях возникающих проблем [сайт] – URL: <https://stackoverflow.com> (дата обращения: 2.04.2020 – 31.05.2020)