



**ФЕДЕРАЛЬНОЕ АГЕНТСТВО МОРСКОГО И РЕЧНОГО ТРАНСПОРТА**

**Федеральное государственное бюджетное образовательное учреждение высшего образования  
«Государственный университет морского и речного флота имени адмирала С.О. Макарова»  
(ФГБОУ ВО «ГУМРФ имени адмирала С.О. Макарова»)  
Воронежский филиал**

Кафедра математики, информационных систем и технологий

Направление подготовки 09.03.02 Информационные системы и технологии

(код, наименование направления подготовки/специальности)

Форма обучения очная

«К ЗАЩИТЕ ДОПУЩЕН(А)»  
Заведующий кафедрой

(подпись)

Черняева С. Н.

(ФИО)

2024

**Выпускная квалификационная работа**

Обучающегося Скогорева Ильи Михайловича

(фамилия, имя, отчество)

Вид работы выпускная квалификационная работа бакалавра

(выпускная квалификационная работа бакалавра, специалиста, магистра)

**Пояснительная записка**

Тема Разработка АИС по планированию снабжения сырьем и материалами  
транспортного предприятия (на примере ООО «Производственная компания  
МАГС»)

(полное название темы квалификационной работы, в соответствии с приказом об утверждении тематики ВКР)

Руководитель работы Черняева С.Н., 21.06.2024

(должность, подпись, фамилия, инициалы, дата)

Консультант \_\_\_\_\_

(при наличии)

(должность, подпись, фамилия, инициалы, дата)

Консультант \_\_\_\_\_

(должность, подпись, фамилия, инициалы, дата)

Обучающийся Скогорев И.М.

(подпись, фамилия, инициалы, дата)

Воронеж

2024

**ФЕДЕРАЛЬНОЕ АГЕНТСТВО МОРСКОГО И РЕЧНОГО ТРАНСПОРТА**

**Федеральное государственное бюджетное образовательное учреждение высшего образования  
«Государственный университет морского и речного флота имени адмирала С.О. Макарова»  
(ФГБОУ ВО «ГУМРФ имени адмирала С.О. Макарова»)  
Воронежский филиал**

Кафедра математики, информационных систем и технологий

Направление подготовки 09.03.02 Информационные системы и технологии  
(код, наименование направления подготовки/специальности)

Форма обучения очная

УТВЕРЖДАЮ  
Заведующий кафедрой

(подпись)

Черняева С. Н.

(ФИО)

2024

**Задание  
на выпускную квалификационную работу**

Вид работы ВКР бакалавра  
(ВКР бакалавра, ВКР специалиста, ВКР магистра)

Обучающемуся Скогореvu Илье Михайловичу  
(фамилия, имя, отчество)

Тема Разработка АИС по планированию снабжения сырьем и материалами  
транспортного предприятия (на примере ООО «Производственная компания МАГС»)

Утверждена приказом ректора университета от \_\_\_\_\_ 20\_\_\_\_, № \_\_\_\_\_

Срок сдачи законченной работы \_\_\_\_\_ 20\_\_\_\_

Исходные данные (или цель ВКР):

Разработать АИС по планированию снабжения сырьем и материалами транспортного  
предприятия.

Перечень подлежащих исследованию, разработке, проектированию вопросов (краткое  
содержание ВКР):

*(актуальность темы, цели и задачи ВКР; аналитический обзор литературных источников;  
постановка задачи исследования, разработки, проектирования; содержание процедуры  
исследования, разработки, проектирования; обсуждение результатов; дополнительные  
вопросы, подлежащие разработке; заключение – выводы по работе в целом, оценка степени  
решения поставленных задач, практические рекомендации; и др.)*

- Введение. Актуальность выбранной темы, цель и задачи ВКР  
(наименование вопроса, раздела и его краткое содержание)
- Исследовательский раздел. Характеристика предприятия, Методы планирования и управления материально-техническим обеспечением, Сущность и виды процессов в складском учете, Анализ типовых программных решений для складского учета  
(наименование вопроса, раздела и его краткое содержание)
- Проектный раздел. Разработка системного решения, Разработка алгоритма работы программного обеспечения, Разработка серверной компоненты  
(наименование вопроса, раздела и его краткое содержание)
- Заключение. Выводы по работе в целом. Оценка степени решения поставленных задач  
(наименование вопроса, раздела и его краткое содержание)

Практические рекомендации

Перечень графического материала (или презентационного материала):

1. Титульный лист
2. Цель и задачи ВКР
3. Исследование предприятия
4. Разработка системного решения
5. Разработка структурной схемы ПО
6. Алгоритм работы авторизации пользователя
7. Основной интерфейс ПО
8. Интерфейс создания нового проекта
9. Этап изготовления
10. Этап отгрузки
11. Интерфейс оператора склада
12. Алгоритм взаимодействия с клиентом
13. Результаты ВКР

Консультанты по разделам ВКР (при наличии):

1. \_\_\_\_\_  
(наименование раздела, ученая степень, ученое звание и должность, ФИО консультанта)
2. \_\_\_\_\_  
(наименование раздела, ученая степень, ученое звание и должность, ФИО консультанта)
3. \_\_\_\_\_  
(наименование раздела, ученая степень, ученое звание и должность, ФИО консультанта)

Дата выдачи задания: \_\_\_\_\_ 20\_\_

Задание согласовано и принято к исполнению: \_\_\_\_\_ 20\_\_

Руководитель ВКР: Черняева С.Н., к.ф.-м.н. доцент  
(должность, ученая степень, ученое звание, ФИО) (подпись)

Обучающийся: ИТ-4-1 Скогорев Илья Михайлович  
(учебная группа, ФИО) (подпись)

## Содержание

ВВЕДЕНИЕ5

1 ИССЛЕДОВАТЕЛЬСКИЙ РАЗДЕЛ6

1.1 Характеристика предприятия6

1.2 Методы планирования и управления материально-техническим обеспечением7

1.3 Сущность и виды процессов в складском учете10

1.4 Анализ типовых программных решений для складского учета16

2 ПРОЕКТНЫЙ РАЗДЕЛ21

2.1 Разработка системного решения21

2.1.1 Анализ языков программирования23

2.1.2 Структурная схема ПО27

2.2 Разработка алгоритма работы программного обеспечения28

2.3 Разработка серверной компоненты41

2.3.1 Структура базы данных41

2.3.1 Разработка алгоритма взаимодействия с клиентом46

2.3.3 Разработка алгоритма взаимодействия с аппаратным блоком47

ЗАКЛЮЧЕНИЕ48

СПИСОК ИСПОЛЬЗУЕМОЙ ЛИТЕРАТУРЫ49

## ВВЕДЕНИЕ

Одним из ключевых условий успешной работы транспортной организации является оценка его финансово-экономических показателей. В последнее время роль финансового анализа существенно изменилась. Деятельность каждого хозяйствующего субъекта стала объектом внимания различных участников рынка, которые также заинтересованы в результатах своей деятельности. Основываясь на имеющихся данных, они могут оценить финансовое положение компании.

Объект исследования или разработки – автоматизированная транспортная система для планирования снабжения сырьем и материалами транспортного предприятия на примере ООО "Производственная компания МАГС".

Цель работы – разработка информационной системы для планирования снабжения сырьем и материалами транспортного предприятия на примере ООО "Производственная компания МАГС", включающую в себя разработку клиентского программного обеспечения, серверной компоненты системы и базы данных системы.

Метод исследования и аппаратура – математические расчеты, программирование, моделирование и тестирование макета устройства.

Полученные результаты и их новизна – полностью работоспособные и исправно функционирующие макет информационной подсистемы.

Область применения – управление хозяйственными запасами транспортной организации.

# 1 ИССЛЕДОВАТЕЛЬСКИЙ РАЗДЕЛ

## 1.1 Характеристика предприятия

Предприятие ООО «Производственная компания «МАГС» ИНН 3662998234 является производственной компанией, занимающейся производством мебели для офиса и предприятий торговли. Юридический адрес компании 394029, Воронежская Область, г.о. Город Воронеж, г Воронеж, ул Циолковского, д. 34/1, офис 203. Уставной капитал компании 10 000 р. Помимо производства мебели компания занимается полиграфической деятельностью, производством изделий из пластмасс, производством строительных металлических конструкций и изделий. Таким образом можно сделать вывод, что компания имеет несколько видов цехов. И оказывает услуги по производству изделий замкнутого цикла. Для выполнения всех видов работ необходимо программное обеспечение, позволяющее осуществить контроль изготовления изделий и контроль складских запасов. Следует отметить, что компания является клиент ориентированной и использует проектный подход для более качественного оказания услуг заказчикам.

Таким образом цель работы является разработка автоматизированной информационной системы по планированию снабжения сырьем и материалами транспортного предприятия (на примере ООО "Производственная компания МАГС")

## 1.2 Методы планирования и управления материально-техническим обеспечением

Методы планирования и управления запасами напрямую влияют на ценообразование и показатели производительности компании. Равномерность и ритмичность производства готовой продукции, а также затраты на ее приобретение и хранение во многом зависят от слаженности работы и своевременности поступления материальных ресурсов.

Характер и виды материальных ресурсов. С точки зрения производственной деятельности материальные ресурсы — это предметы труда, необходимые для производства готовой продукции. К ним относятся:

- сырье, направляемое в производство для первичной переработки;
- основные материалы и компоненты, входящие в готовое изделие;
- вспомогательные материалы обеспечивают работу основного оборудования;

Для обеспечения стабильной работы оборудования и организации логистики необходима энергия и топливо. Эти ресурсы позволяют обеспечить нормальное функционирование производственных процессов. С финансовой точки зрения физические ресурсы являются оборотными активами. В ходе производственного цикла ресурсы переносят свою ценность в готовый продукт.

Расход сырья, расходных материалов и деталей на единицу продукции характеризуется показателем материалоемкости, который может составлять до 70-80% структуры затрат. Снижение материалоемкости за счет рационального использования ресурсов может способствовать снижению затрат, повышению эффективности использования основных фондов и снижению стоимости жизни на производстве.

Материально-техническое обеспечение — это система, которая организует, циркулирует и использует ресурсы для удовлетворения производственных потребностей.

К задачам МТО можно отнести следующие:

- определение логистических потребностей и планирование компании;
- организация поставки сырья, расходных материалов и комплектующих в установленные сроки;
- установка периода покупки точно в соответствии с периодом потребления;
- поиск поставщиков с минимальными ценами закупок и оптимальные уровни запасов;
- соблюдение требований к качеству сырья, материалов и комплектующих;
- рациональное распределение ресурсов и контроль расходов;
- сокращение затрат на закупку и хранение за счет оптимизации процессов управления логистикой.
- повышение эффективности использования ресурсов, оборачиваемости активов и т.д.

По мере движения материальных ресурсов по производственной цепочке формируются материальные средства, которые подразделяются на следующие виды:

- текущие. К ним относятся сырье, материалы и комплектующие, которые еще не вошли в производственный процесс и присутствуют на складе между поставками для обеспечения непрерывности производства.
- незавершенного производства. К ним можно отнести ресурсы, находящиеся на различных стадиях производственного процесса и еще не полностью переработанные для производственного процесса.
- страховые. Эти запасы создаются для снижения риска возникновения нехватки сырья, материалов и комплектующих в случае возникновения непредвиденных обстоятельств, связанных с перебоями в поставках или изменением рыночного спроса. Если текущие запасы постоянно меняются,

уровень страховых резервов в стандартном производственном процессе обычно остается прежним.

— готовой продукции. Это готовая продукция для реализации, находящаяся на складе компании.

Эффективная система управления запасами должна быть направлена на минимизацию запасов, не влияя на весь процесс. Это позволяет высвободить замороженные оборотные средства и снизить затраты на содержание неиспользуемых в настоящее время резервов.

Основные этапы планирования материально-технического снабжения

— планирование производственной логистики – это процесс, состоящий из следующих этапов:

— анализ потребностей. Исходной информацией для анализа являются данные планов производства и продаж, норм потребления и запасов на начало планового периода.

— план закупок. На этом этапе определяется состав необходимых ресурсов, определяется количественный характер спроса и формулируются соответствующие условия закупок.

— корректировка планов. В течение периода планирования могут произойти изменения, которые потребуют корректировки логистического плана. Кроме того, планы закупок должны корректироваться исходя из фактических запасов в целях предотвращения образования избыточных запасов на складах и устранения дефицита отдельных видов материальных ресурсов при выявлении дефицита.

### 1.3 Сущность и виды процессов в складском учете

Успешное функционирование предприятия – это совокупность влияния различных факторов и эффективное выполнение основных функций. При этом следует отметить, что правильное размещение товаров – одно из главных условий стабильной работы компании. Грамотно организованный процесс учета складских материалов является очень важной и необходимой частью организационной деятельности.

Прием товара на склад – важная часть технического процесса. Это делает лицо, ответственное за финансы, в соответствии с внутренними правилами «приемки».

Началом приемки считается день поступления материалов и оборудования на склад. Своевременное получение оборудования обеспечивает точный расчет и отчетность. Если полученный материал ненадлежащего качества или его количество не соответствует заказу, поставщику предъявляется претензия. Если при приемке обнаружены дефекты, дается определенный срок для предъявления претензии поставщику. Несоблюдение этих сроков может привести к причинению ущерба, так как претензии поставщика не будут удовлетворены.

При приеме материалов и оборудования на склад организации необходимо обеспечить соответствие полученных материалов и оборудования сведениям, содержащимся в счете-фактуре поставщика, спецификациям, уровням запасов, упаковочных этикетках и т.п., а также условиям договора.

В течение установленного срока управляющий складом передает все поступающие товарно-материальные запасы в бухгалтерию. Как правило, документы отправляются ежедневно, но не реже 1 раза в неделю.

Для облегчения группировки первичных документов передача входящих и исходящих документов может регистрироваться в отдельных регистрах. В этом случае реестр выдачи документов постепенно нумеруется заведующим складом с начала года: нечетные (01, 03, 05 и т.д.) вносятся в реестр выдачи

Входящий документ четного номера (02, 04, 06 и т.д.). В этом случае записи о доставке документов нумеруются заведующим складом поэтапно, начиная с начала года: нечетными номерами (01, 03, 05 и т.д.). Поступившие документы заносятся в журнал под четными номерами (02, 04, 06 и т.д.). Выдача расходных квитанций. В реестре номера документов должны быть сгруппированы по субсчетам, а в случае субсчетов - по источнику поступления (от поставщика, ответственного лица, безвозмездно по договору дарения, за счет амортизации основных средств и т.п.) и по направлению затрат: производство продукции, оказание услуг, выполнение работ, ремонт основных средств, общепроизводственные и общехозяйственные цели, реализация, безвозмездные передачи и т.п.) и направлению затрат: производство продукции, оказание услуг, выполнение работ, обслуживание оборудования (новые фонды, общие производственные и общехозяйственные цели, реализация, свободные передачи и т.д.) и направлению затрат: производство продукции, оказание услуг, производительность труда, содержание основных фондов, общепроизводственные и общехозяйственные цели, реализация, бесплатный трансфер и т.д.

Бухгалтерия проверяет правильность и полноту записей в карточке складского учета не реже одного раза в неделю на основании основных документов, направляемых в бухгалтерию заведующим складом. Факт проверки подтверждается подписью проверяющего на карточке складского учета.

Оборудование размещается на складе, соблюдаются условия хранения и контроль за его состоянием и перемещением, принимаются меры по предотвращению повреждений. Условия хранения и особые требования к обработке можно найти в соответствующих документах.

Затем оборудование и материалы отправляются на склад.

Выезд со склада является завершающим этапом технического складского процесса и включает в себя передачу оборудования заказчику.

Отгрузка оборудования со склада является заключительным этапом работы склада, завершающим процесс выполнения заказа.

На рисунке 1 показан логический процесс работы склада.



Рисунок 1 — Логический процесс работы склада

Одной из основных проблем при организации складов является необходимость ускорить процесс формирования очередей, подтверждения заказа, отгрузки и транспортировки товара, что существенно сокращает время от заказа до доставки.

Сокращая «время отклика», вы можете сократить запасы, свести к минимуму отходы и, следовательно, снизить затраты на техническое обслуживание. Повышайте удовлетворенность и лояльность клиентов.

Продемонстрировать необходимость и цель использования программного обеспечения на складе.

Многие коммерческие и промышленные компании каждый день продают разнообразную продукцию.

Именно поэтому представители малого и среднего бизнеса стремятся добиться полноценной работы складов и в конечном итоге управлять всей хозяйственной деятельностью, связанной со складскими товарами.

Этот сложный и кропотливый анализ трудоемкий, так как требует обработки больших объемов информации и тщательного контроля наличия товара на складах. Ежедневное получение товара и ручная маркировка также отнимают немало времени. Поэтому каждому бизнесу необходим план отслеживания запасов. На основе ежедневного анализа продаж можно более точно спланировать следующие шаги компании, чтобы оптимизировать продажи материалов и повысить эффективность производства.

В настоящее время руководители многих коммерческих организаций приобретают специальное программное обеспечение для настройки систем отчетности и своевременной отметки поступления и реализации товаров. С помощью компьютерных программ вы сможете организовать правильное функционирование вашей организации, в том числе учет товарно-материальных запасов.

Каждая программа управления запасами разработана в соответствии с федеральными законами и стандартами бухгалтерского учета, чтобы защитить компанию от несоблюдения.

Современные программы инвентарного учета позволяют систематизировать все операции. Созданные программистами шаблоны мастер-документов помогают работникам склада тратить меньше времени на оформление документов по перевозке товаров.

Бухгалтерское программное обеспечение позволяет компаниям и новым филиалам с максимальной точностью учитывать затраты на запасы на единицу продукции.

Менеджеры торговых и промышленных компаний в любой момент могут узнать о количестве товара на складе.

Программное обеспечение позволяет анализировать все процессы внедрения и определять, какие продукты пользуются наибольшей популярностью у потребителей.

Автоматизация может повысить производительность и качество, а также сэкономить дополнительное оборудование, материалы и затраты.

Рассматривая преимущества автоматизации складов, менеджеры розничных магазинов часто в первую очередь думают о потенциальной внутренней экономии затрат, но это не единственная ценность.

На рынке представлено множество складских программ: разные типы хранения информации, разные интерфейсы, как платные, так и бесплатные. Однако при выборе программы важно помнить, что вложения в платный софт быстро окупаются. Затраты, возникающие из-за ненужных человеческих и механических работ, простоев, просроченных товаров и неправильной оценки складских функций.

К критериям выбора программного обеспечения можно отнести:

Интуитивно понятный пользовательский интерфейс. Важно то, что пользователи могут самостоятельно вносить информацию в систему и получать необходимые отчеты без помощи программиста. Все действия в программе должны быть интуитивно понятны;

Круглосуточная техническая поддержка. После покупки программы сотрудничество с компанией будет прекращено, так как может потребоваться консультация с программистом для понимания функционала программы и устранения возможных ошибок;

Базу данных можно обновлять онлайн. Программное обеспечение должно загружать информацию о состоянии продукта сразу после внесения изменений.

Отображение статуса продукта. Рядом с товаром стоит пометка (акция, предзаказ и т.п.). Это позволяет избежать ситуации, когда оплата завершена и покупатель ожидает размещения заказа, но товара нет в наличии.

Возможность обмена базами данных с интегрированными системами. Для проведения комплексного анализа программ лояльности и скидок важно загружать и скачивать информацию совместно с CRM-системой.

Если цена покупки изменится, цена продажи также изменится автоматически.

Заказ должен быть принят автоматически. Все поставщики и склады должны быть интегрированы, чтобы можно было быстро и автоматически обмениваться информацией между заинтересованными сторонами.

Возможность сортировки товаров в отчетах. При написании отчетов товары необходимо разделить на популярные и нерыночные локации, чтобы быстро реагировать на потребительский спрос и корректировать предложения. Например, ввести скидки и расширить ассортимент продукции.

Сегментировано по объему. Чтобы отслеживать свои доходы и расходы, вам необходимо отдельно отслеживать стоимость товаров, курьерские расходы и расходы на техническое обслуживание.

Автоматически учитывать возвраты товаров. В программе должна быть реализована функция полного или частичного возврата. Содействовать системе самостоятельно отслеживать заказы, не дошедшие до потребителя, отмечать их в карточке движения груза и добавлять в карточку груза сопроводительные документы.

Импорт прайс-листов в различных форматах. Чем больше количество форматов, тем лучше. Для программы требуется минимум 2 формата (XLS и CSV). Возможность импортировать только новые проекты или выбранные проекты по отдельности также очень полезна.

Возможность автоматического создания файлов. Возможность автоматической настройки платежных поручений на банки или счета покупателей может существенно ускорить вашу работу.

Поэтому основным критерием выбора решения должно стать упрощение складских операций.

## 1.4 Анализ типовых программных решений для складского учета

Современный рынок программного обеспечения предлагает широкий спектр типовых программных решений и продуктов для малого бизнеса и государственных компаний.

Недостаточная автоматизация и неправильное, неэффективное управление складом и логистикой могут привести к значительным потерям. Одной из причин может быть неправильный выбор программного обеспечения.

Но сбои в автоматизации происходят не только из-за плохих решений, но и из-за неправильных процессов внедрения компаниями. Оперативная информационная система должна обеспечивать следующее содержание:

Управляйте целостностью информации во всех аспектах системы. Надежность определяется как взаимосвязь между полученной информацией и информацией, запрошенной или управляемой запросом. Поскольку наши знания относительно, мы не можем достичь 100% полноты информации. Кроме того, следует отметить, что стремление к повышению целостности информации приводит к увеличению административных расходов и снижению ее эффективности.

Кроме того, автоматизированные информационные системы должны отвечать многим техническим требованиям, таким как:

- Скорость – скорость, с которой информация вводится, ищется и обрабатывается;
- надежно предотвратить несанкционированный доступ к данным;
- создать действие человека;
- интеграция с модулями, используемыми в системах передачи данных;
- удобный интерфейс рабочего места;
- ранее он использовался в новых системах;
- возможность конвертировать данные

- возможность развития системы;
- высокая эксплуатационная надежность.

В современном мире доступно множество различных программных продуктов для оптимизации деятельности компании.

При выборе автоматизированных складских информационных систем ИС «1С:УНФ» и «ЭМКАМ» для малых и средних предприятий учитывались функции существующих систем, а также их преимущества и недостатки [11].

Самая известная и часто используемая программа – «СТАЛ», которая является одним из лидеров ниши российского рынка и занимает высокие места в рейтингах.

Облачная программа направлена на полную автоматизацию малого бизнеса в коммерческом секторе и секторе услуг. С помощью ЕКАМ вы можете автоматизировать:

- управление запасами;
- объем продаж;
- купить;
- отчет о состоянии группы;
- создавать программы лояльности и скидок;
- прием и выполнение заказов в кафе и ресторанах;
- поток денег.

Кроме того, программное обеспечение инвентаризации ЕКАМ интегрируется с бухгалтерскими приложениями, упрощая создание налоговых отчетов.

Преимущества программного обеспечения для инвентаризации ЕКАМ:

Быстро и точно контролируйте уровень запасов.

Автоматически устанавливает цену продажи на основе указанной маржи.

Инвентаризация происходит быстро и легко.

Интеллектуальная система закупок.

Используйте эту программу для удаленной работы.

Широкий спектр настраиваемых аналитических, финансовых, продуктовых и управленческих отчетов.

Счета-фактуры на товары от нескольких структурных подразделений могут формироваться одновременно.

Загрузка прайс-листа поставщика в программу упрощает создание номенклатуры.

Кроссплатформенность: Программу можно использовать на компьютерах через браузер и на мобильных устройствах через специальное приложение.

Данный тарифный план приемлем для малого бизнеса.

Подключите свой интернет-магазин.

Удобный и интуитивно понятный интерфейс позволяет настроить программу и быстро обучить ее использованию новых сотрудников.

Предлагает удобный модуль группы клиентов, который можно привязать к дисконтным картам.

Мобильная техническая поддержка доступна 24/7.

Автоматически отправлять продающие материалы в 1С.

Нарисовать технические схемы предприятий общественного питания.

Высокая стабильность

Открытый API, можно настроить под нужды клиента.

Недостатки Экама:

Нет возможности управлять доставкой.

Нет календаря и панели задач.

1. Одним из преимуществ ЕКАМ является своевременная и профессиональная техническая поддержка. Ее специалисты решают проблемы посредством командной строки и удаленного подключения к компьютерам пользователей.

Подбор персонала в «ЭКАМ» осуществляется сотрудниками компании. Предприниматели смогут воспользоваться всеми преимуществами приложения.

Большое количество успешных внедрений системы подтверждают, что ее можно эффективно применять на складах любого размера и типа: от небольших складов до крупных распределительных центров или складов готовой продукции промышленных компаний различных отраслей.

1С:План УНФ — это стандартный план, разработанный специально для малого бизнеса. Эта система имеет два основных преимущества: надежность и обширный функционал. В данной системе реализованы только самые необходимые функции по учету, анализу, контролю и планированию внебюджетных операций. Это решение поможет вам сделать ваш бизнес более эффективным, предоставляя владельцам и менеджерам широкий спектр бизнес-инструментов, а сотрудникам — новые способы повышения производительности. Система не имеет лишних функций и поэтому легко адаптируется к административным и бухгалтерским функциям компании. Это позволяет «приступить к работе» и делает повседневные задачи удобными и комфортными.

1С:УНФ ведет складской учет и выполняет все складские операции:

Возможность просмотра количества и цены всех поступивших на склад товаров. В зависимости от типа склада товар поступает на склад по накладной или заказу. Квитанция – документ, оформляемый на обычном (нерегистрируемом) складе, содержащий перечень поставщиков, договоров, складов, видов работ, сроков, количества и цен. После сохранения счета-фактуры вы можете распечатать этикетки и ценники. В разделе «1С:УНФ» также можно предоставить чеки за оказанные услуги и ранее заявленные авансовые платежи.

Приходной заказ – документ, оформленный на складе заказов, в котором фиксируется поступивший на склад товар: номенклатуры и количества.

Возможность перемещать товарно-материальные запасы компании между складами, отделами и подразделениями. Передача запасов – это

документ, позволяющий оформить передачу, списание расходов, перевод в хозяйственную деятельность и возврат из хозяйственной деятельности.

Передача деталей — это документ, используемый для оформления перемещения товаров между деталями в пределах одного склада. Записывает перемещения из одной ячейки в несколько ячеек или из нескольких ячеек в одну ячейку. Эта программа позволяет передавать товары, которые ранее были зарезервированы для заказов клиентов.

Продажа товаров подразумевает возможность организации продажи товаров со склада компании с учетом предоплаты или предоставленного Ski-Doc.

Накладная — документ, оформляемый на обычном (неформальном) складе на основании заказа, счета-фактуры или товарного чека.

Инвентаризация на складе. Возможность использовать утилиты для своевременного формирования счетов за товарные запасы.

Уровень запасов отражает результаты проверки фактического наличия товара на складе. По результатам инвентаризации подготовить документы учета запасов (в которых фиксируются излишки запасов, определенные по результатам инвентаризации) и списания запасов (в которых отражаются выявленные недостатки).

Помимо фактически обнаруженных дефектов, программа также может вести учет материалов на складе, например, по сроку их годности.

## 2 ПРОЕКТНЫЙ РАЗДЕЛ

### 2.1 Разработка системного решения

Основной целью разрабатываемой системы является разработка программного продукта, позволяющего планировать и доставлять сырье и вспомогательные материалы транспортным компаниям. Одной из задач также была разработка системы, способной отслеживать местонахождение транспортных средств организации. Для отображения информации на карте местности необходима информация о текущем местонахождении автомобиля. Чтобы решить эту проблему, необходимо было разработать устройство, которое могло бы отправлять координаты серверному ПО. В этом случае время задержки должно быть минимальным. Полученная информация о местонахождении автомобиля должна храниться в базе данных. Для отображения данных требуется отдельный программный клиент, который устанавливается на автоматизированное рабочее место (АРМ). Блок-схема разрабатываемой системы представлена на рисунке 2.

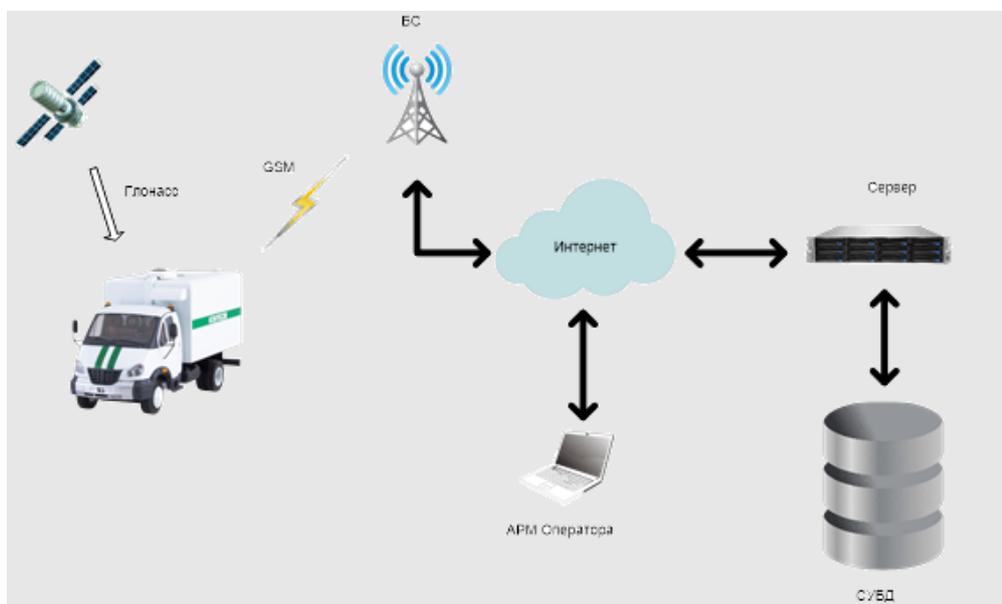


Рисунок 2 – Структурная схема разрабатываемой системы.

Следует отметить, что для определения координат будет использоваться система спутниковой навигации ГЛОНАСС. Для передачи данных будет

использоваться мобильные сети связи второго поколения. Скорость передачи данных необходимая для отправки данных необходима не большая, а на территории России в настоящее время наиболее распространены 2G сети. Эти два факта определяют выбор способа передачи данных с использованием 2G сетей. Таким образом была разработана структурная схема аппаратного блока, представленная на рисунке 3.



Рисунок 3 – Структурная схема аппаратного блока

Для разработки программного обеспечения используют двухуровневую архитектуру ПО, включающую серверную часть с БД и клиентскую. Следует отметить, что клиентское программное обеспечение устанавливается на ПК ответственного сотрудника по подразделению. Серверная же компонента отвечает за хранение, обновление и предоставление доступа к данным. Структурная схема двухуровневой архитектуры представлена на рисунке 4.

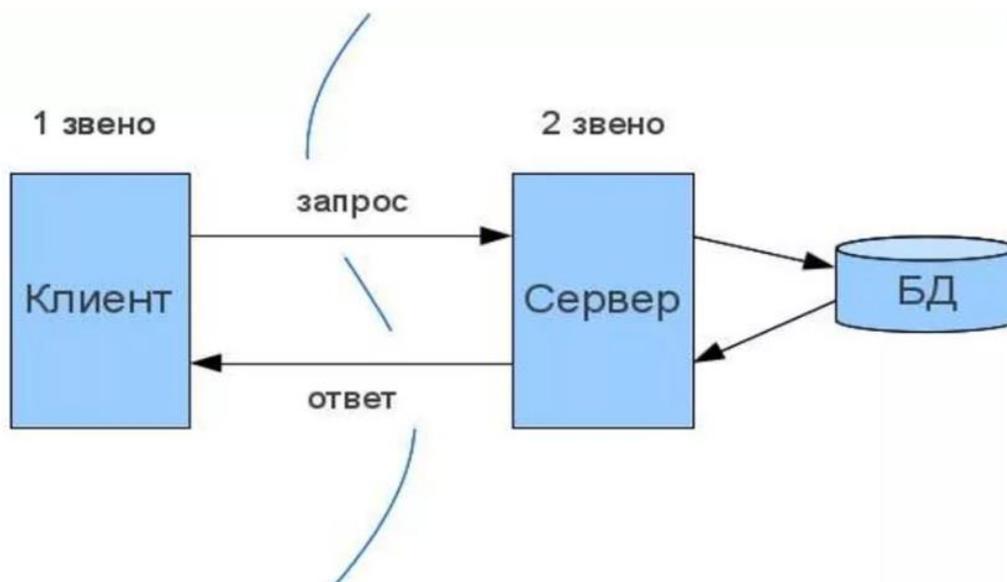


Рисунок 4 – Структурная схема разрабатываемой системы.

Из рисунка 4 видно, что разрабатываемая система будет работать по средством локальных вычислительных сетей. Что позволит обеспечить бесшовную интеграцию в существующие программные системы на предприятии.

### 2.1.1 Анализ языков программирования

Целью дипломного проекта является разработка специального программного обеспечения, которое может удаленно подключаться к серверу через Интернет и аутентифицировать, получать и просматривать информацию.

Выбор языка программирования не является тривиальным, но важно учитывать как скорость разработки, оптимизации, так и скорость выполнения программы. После анализа сравнения языков программирования по потребляемой процессором энергии, скорости выполнения и занимаемому объему программы была составлена сводная таблица. Результаты приведены в таблице. Но стоит сразу отметить, что не все языки соответствуют предъявляемым требованиям. Мы можем выбрать группу языков, которые наиболее популярны сегодня и подходят для выполнения задач, это: C++, Java,

JavaScript, Python и Go. Каждый из них имеет свои преимущества и недостатки. Сравнение языков представлено в таблице 5.

Таблица 5 - Сравнение языков программирования

Языки	Затрачиваемая энергия	Языки	Время Выполнения	Языки	Занимаемый объём
C	1	C	1	Pascal	1
Rust	1,03	Rust	1,04	Go	1,05
C++	1,34	C++	1,56	C	1,17
Java	1,98	Java	1,89	Fortran	1,24
Swift	2,79	C#	3,14	PHP	2,57
C#	3,14	Swift	4,2	Swift	2,71
Go	3,23	Fortran	4,2	Python	2,8
Python	75,88	Lua	82,91	Lua	6,72

В этом плане выделяются 2 конкурента, это C++ и Java. Чтобы решить этот вопрос, также стоит рассмотреть существующую структуру.

Фреймворк — это программная платформа, которая определяет структуру программной системы и различные компоненты большого программного проекта. Фреймворк позволяет библиотеке выполнять процедуры аналогичных функций, не накладывая на нее ограничений, не влияя на архитектуру программного продукта. Набор фреймворков определяет правила построения архитектуры приложения, но устанавливает поведение по умолчанию на ранних стадиях разработки - "фреймворк", который должен быть расширен и изменен в соответствии с указанными требованиями.

Для java существуют следующие фреймворки:

- Grails;
- Spring;
- Vaadin;

— Spark.

Однако для поставленной цели ни один из них не подходит, так как все они предназначены для написания веб приложений.

— C++ фреймворки;

— Juce;

— Qt.

Среди них особенно можно выделить Qt, позволяющий создавать приложения не только для Windows, но и для Android, iOS и Linux. Вы также можете использовать набор наборов инструментов Qt для создания развертываемых приложений и установщиков для различных платформ. Он наиболее полно охватывает требования языка, среды разработки и разрабатываемых программ.

Пакет Qt SDK включает в себя набор библиотек компиляторов для программ и, конечно же, различные компоненты, упрощающие работу с Qt, такие как QtCreator Qt Creator — это среда для разработки программ на c++. В нем есть редактор кода, а также редактор форм и отладчик. На рисунке 24 показана среда Qt Creator.

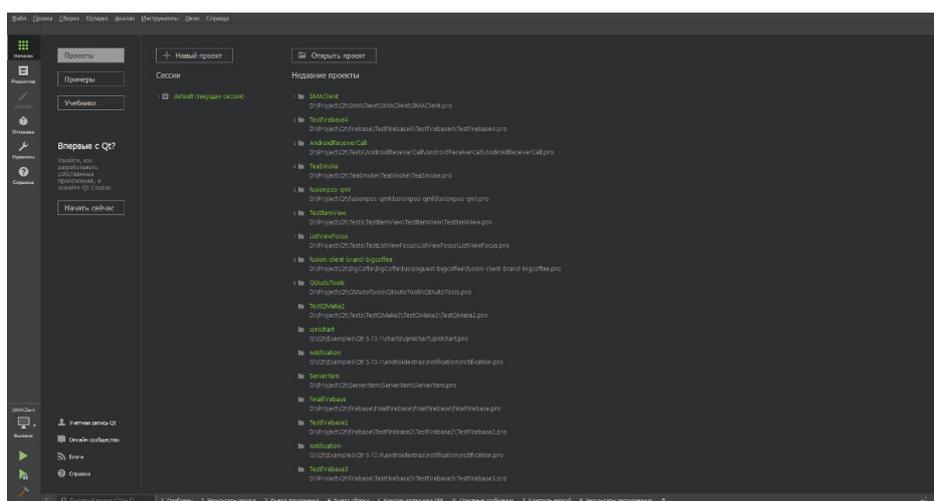


Рисунок 24 – Среда QtCreator

Для решения поставленной задачи был выбран язык C++ для разработки клиентского программного обеспечения.



## 2.1.2 Структурная схема ПО

Для описания клиентского программного обеспечения была выбрана классическая пятиуровневая архитектура.

Уровень доступа к данным состоит из TCP-сокетов, которые требуют подключения, авторизации и получения данных от системных серверов.

Бизнес-уровень включает в себя все блоки приема, хранения и обработки входящих данных, причем вы можете выбирать блоки обработки данных с сервера.

Уровень представления — это набор виджетов, которые используются для отображения информации.



Рисунок 25 – Структура пятислойного шаблона проектирования

## 2.2 Разработка алгоритма работы программного обеспечения

После запуска программной системы пользователь увидит интерфейс авторизации в системе. Алгоритм, осуществляющий эту процедуру представлен на рисунке 26.

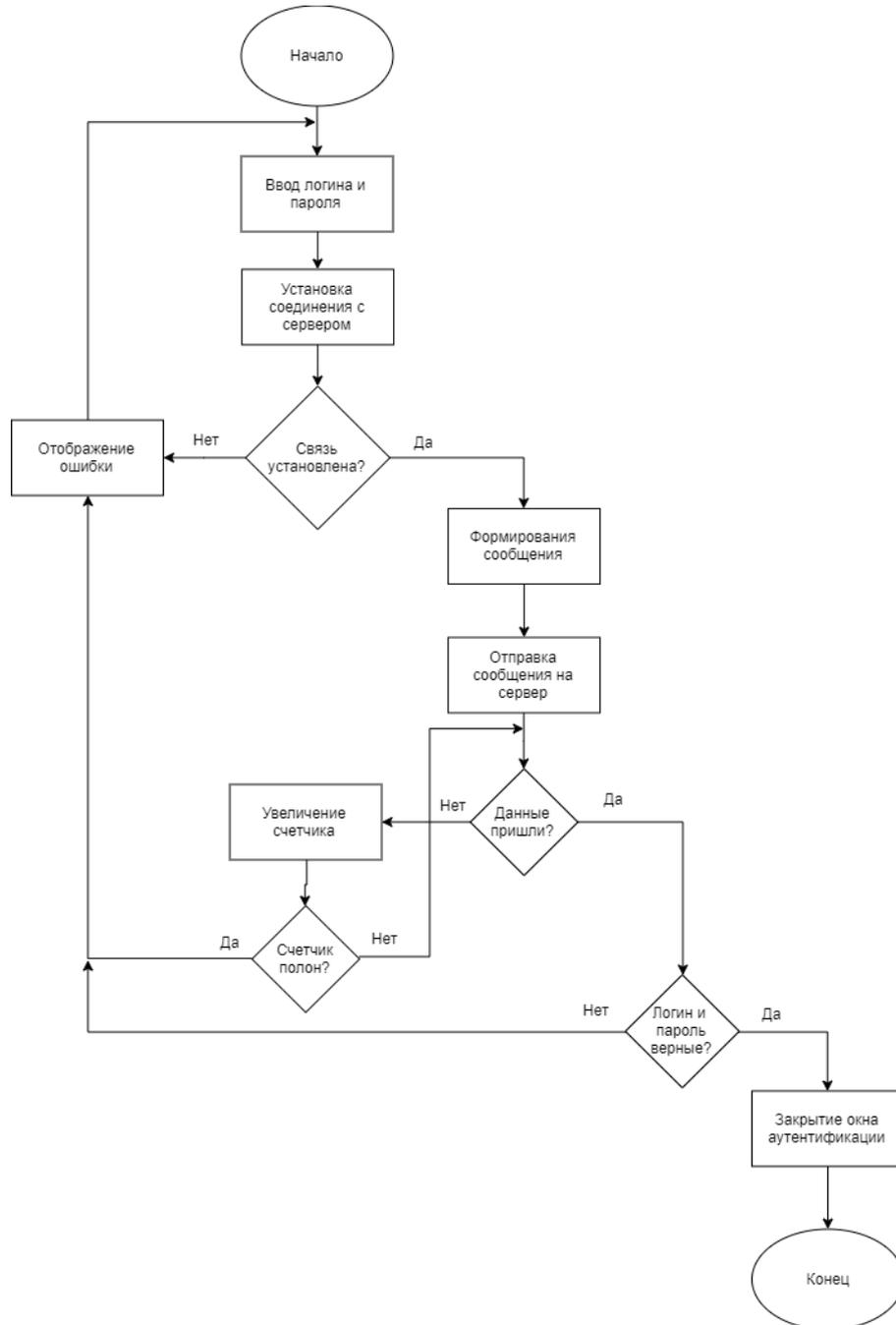


Рисунок 26 – Алгоритм аутентификации оператора

Следует отметить, что ввода оператором логина и пароля инициализирует процесс установки связи с сервером. В случае не удачной попытки, пользователь увидит сообщение, что сервер не доступен.

В случае удачного установления связи, программный модуль аутентификации формирует сообщение на сервер и отправляет его. На клиентской стороне запускается таймер 1000 мс, если сервер в течении это времени не ответит, пользователь уведет сообщение об ошибке. В случае удачного получения ответа, запускается основной интерфейс программной компоненты.

Окно аутентификации изображено на рисунке 27.

Рисунок 27 – Окно ввода логина и пароля

После ввода логина и пароля данные отправляются на сервер и осуществляется процедура аутентификации и авторизации пользователя.

После удачной авторизации перед пользователем откроется интерфейс, представленный на рисунке 28.



Рисунок 28 – Основной интерфейс программы

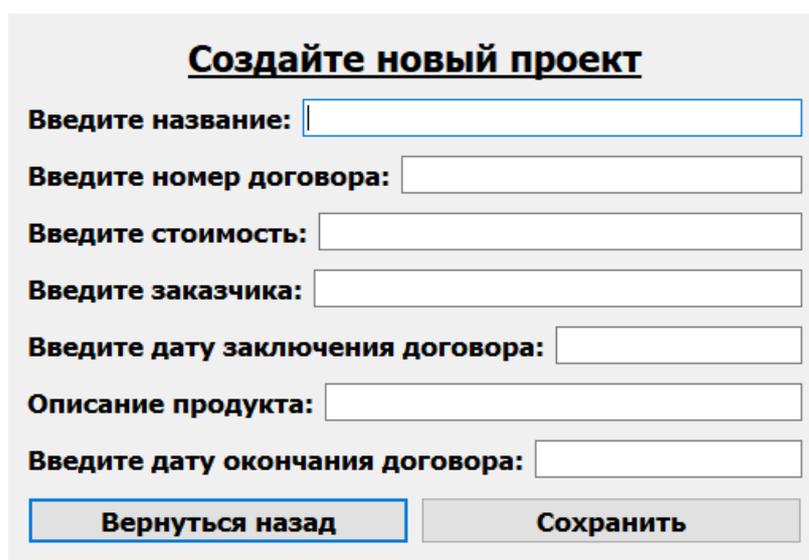
Стоит отметить, что разработанная информационная система подчеркивает проектный подход в производстве. Этот проект имеет следующие особенности:

1. Временный характер. Каждый проект имеет четкие временные рамки. Без такой структуры деятельность называется операцией и может длиться столько, сколько необходимо.

2. Уникальные продукты, услуги, результаты. Проект должен давать уникальные результаты, достижения и продукты. В противном случае такой бизнес превратится в массовое производство.

3. Постоянное развитие. Каждый проект развивается с течением времени и проходит заранее определенные фазы или этапы. В этом случае создание технического задания проекта строго ограничивается тем, что было установлено вначале. Таким образом в программном обеспечении можно выделить три этапа, а именно предварительный этап, этап изготовления и этап отгрузки изделия.

На предварительном этапе формируется проект. Интерфейс введения данных о проекте представлен на рисунке 29.



**Создайте новый проект**

Введите название:

Введите номер договора:

Введите стоимость:

Введите заказчика:

Введите дату заключения договора:

Описание продукта:

Введите дату окончания договора:

Рисунок 29 – Интерфейс создания нового проекта

Для создания проекта необходимо ввести название проекта, т.е. как его будет идентифицировать компания. Далее необходимо ввести номер договора и дату заключения договора, стоимость проекта. Под описание продукта понимается дополнительное требование, которое поступило от заказчика. Для контроля исполнения проекта, так же необходимо ввести дату окончания по договору.

После создания проекта приложение позволяет осуществить просмотр информации о проекте, при нажатии на кнопку «Информация о проекте» открывается интерфейс отображения информации о проекте.

**Инофрмация о проекте**

**Номер договора: 123123213**

**Название: Стул**

**Дата сделки: 12.02.2013**

**Стоимость проекта: 8000 руб.**

**Заказчик: АО "Концерн Созвездие"** Подробнее

**Описание: Отличная стремянка**

**Дата окнчания: 12.03.2014**

Закрыть

Рисунок 30 – Интерфейс просмотра информации о проекте

Следующим этапом является сопоставление проекта и материалов необходимых для изготовления изделия. Для этого необходимо выбрать в списке проект и далее нажать кнопку добавить материал.

**Название продукта**

<b>Материалы на складе</b>		<b>Материалы для проекта</b>	
Добавить	Удалить	Удалить	
Извание материала	личество на складе	Извание материала	Количество
1 Брус	250		
2 Клей	250		

**Добавить в проект** **Убрать на склад**

Количество:  Количество:

Добавить Убарть

Вернуться назад

Рисунок 31 – Интерфейс сопоставления материалов склада и материалов проекта

Открывшийся интерфейс представляет собой основное рабочее окно и кнопки управления, которые находятся внизу. Основной интерфейс состоит из двух частей. Левая часть отображает материалы, представленные на складе. Правая часть материалы, используемые в проекте. Для настройки материалов, используемых в проекте, необходимо выбрать существующие материалы на складе, указать количество нажать «Добавить». В случае, если произошла ошибка добавления материала необходимо нажать «Убрать на склад». После ввода всех материалов необходимо нажать «Вернуться назад». После чего интерфейс добавления материалов закроется, и пользователь окажется в интерфейсе основного окна. Таким образом после ввода всей информации можно передавать проект на этап изготовления. Для этого необходимо в левой части основного интерфейса выбрать проект и нажать «->».

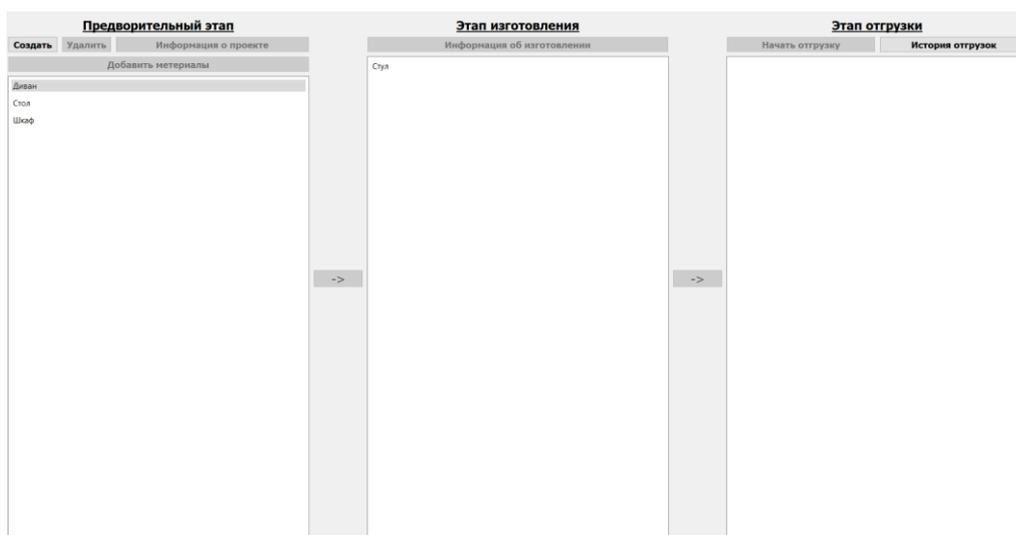


Рисунок 32 – Интерфейс перевода проекта на этап изготовления

Далее осуществляется процесс изготовления проекта. Для контроля изготовления проекта необходимо выбрать проект и нажать «Информация о проекте».



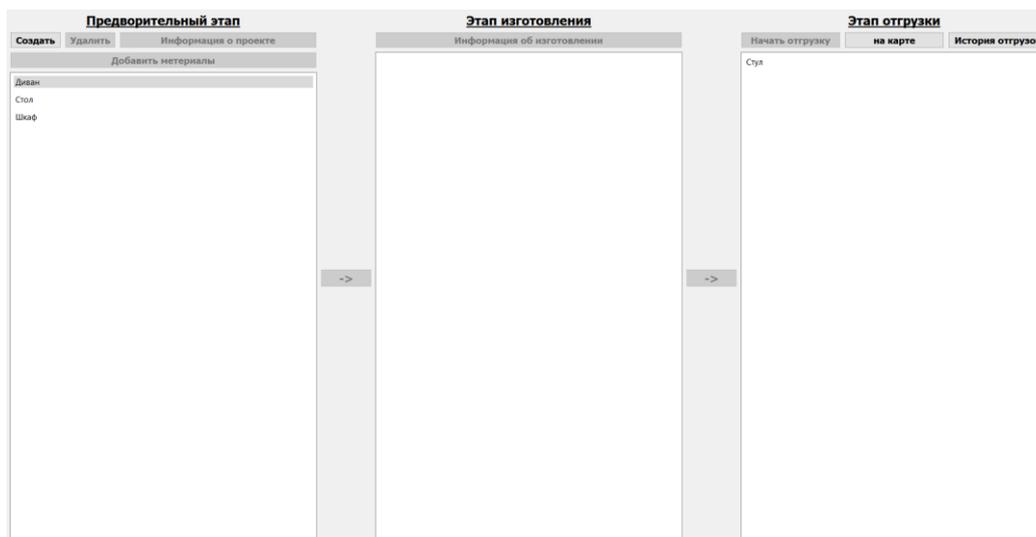


Рисунок 34 – Интерфейс этапа отгрузки проекта

Интерфейс этапа отгрузки представляет из себя список проектов, ожидающих отгрузки.

Для начала этапа отгрузки необходимо нажать «Начать отгрузку» и откроется интерфейс, представленный на рисунке 35.

**Отгрузка проекта**

**Название:** Стул

**Введите дату отгрузки:**

**Номер транспортного средства** A777AAA136

**Введите код сотрудника, отгрузившего товар:**

Рисунок 35 – Интерфейс отгрузки проекта

Для отгрузки проекта необходимо выбрать транспортное средство, указать дату отгрузки и ввести код сотрудника, отгрузившего товар и нажать «Отгрузить товар».

После сформированной информации об отгрузке проекта. Далее в основном интерфейсе можно отследить этап доставки проекта заказчику. Для этого необходимо нажать «на карте».

Интерфейс представлен на рисунке 36.

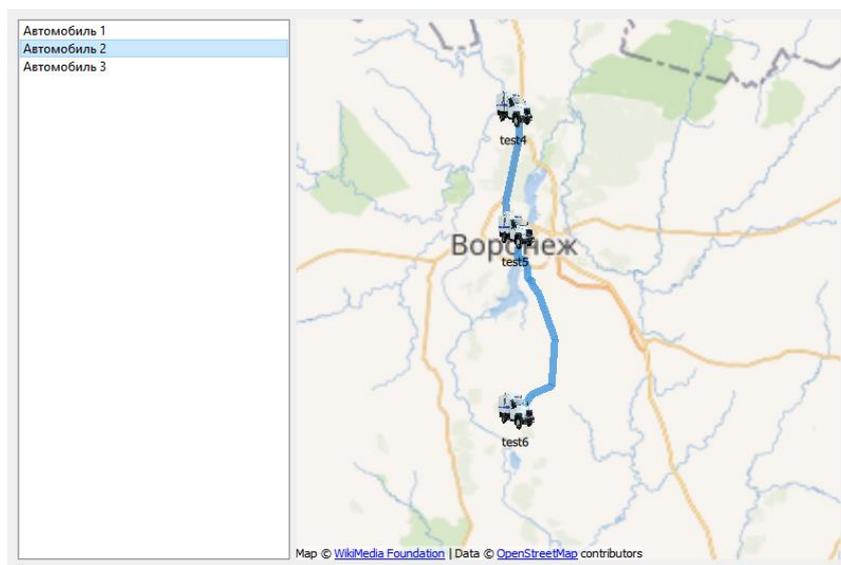


Рисунок 36 – Интерфейс отображения на карте автотранспортных средств

Открывшийся интерфейс представляет список автотранспортных средств и графическую составляющую картографии, где отображается информация о маршруте движения автотранспортного средства. Следует отметить, что места длительных остановок отображаются красными метками.

Функциональная схема блока мониторинга представлена на рисунке 37.



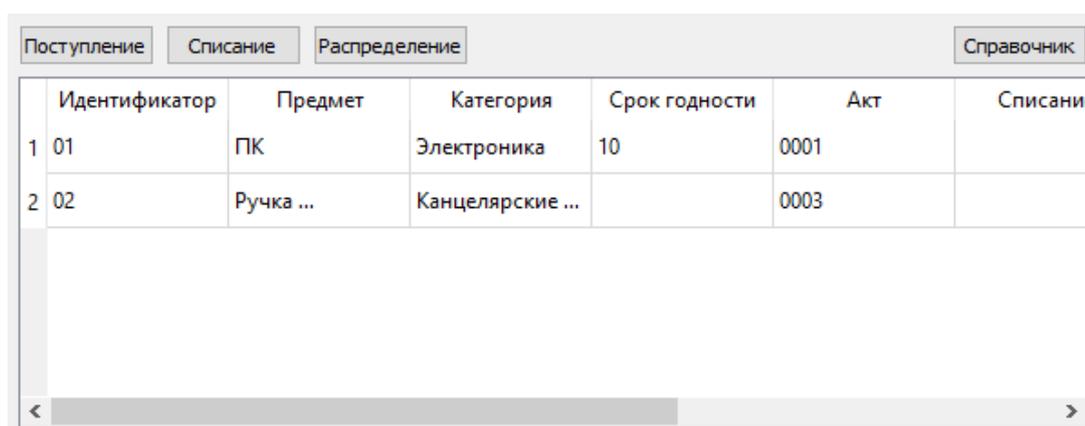
Рисунок 37 – Функциональная схем клиента

На рисунке 37 представлена функциональная схема клиентского программного обеспечения. Каждый блок схемы отвечает за определенную функцию клиентского ПО и может быть легко заменен или расширен. В этом ПО модули отправки и приема данных реализуют обмен данными посредством потоков. Это гарантирует, что модель OSI будет работать, когда нижние уровни не знают о передаваемых данных. Далее идет блок формирования и анализа запросов; этот блок отвечает за протокол обмена между клиентом и сервером. Если вы меняете договор, просто отредактируйте этот блок. Далее идет программный интерфейс, который отправляет запросы к серверу; он предлагает

те же функции для получения данных с сервера. Далее идет блок обработки данных, который делит данные на определенные категории. Это может быть информация о полях, датчиках или устройствах. При добавлении нового класса устройства необходимо создать новый класс, описать его логику и добавить условия в блок обработки данных.

Все ранее описанные блоки объединены в один блок API. Этот блок реализует набор функций для взаимодействия графического интерфейса пользователя с бизнес-логикой программы. Также слева имеется раздел для хранения настроек приложения. Это могут быть логин и пароль клиента, адрес сервера, выбранный тип карты и т. д.

Для заполнения данных о материалах и предметах на складе разработан модуль управления хранением. Основной интерфейс этого модуля представляет из себя общую информацию о всех предметах хозяйственной деятельности. Предметы хозяйственной деятельности представлены в табличной форме, которая имеет следующие поля идентификатор записи предмета, наименование предмета, категория предмета, срок годности, номер акта поступления, номер акта списания.



	Идентификатор	Предмет	Категория	Срок годности	Акт	Списани
1	01	ПК	Электроника	10	0001	
2	02	Ручка ...	Канцелярские ...		0003	

Рисунок 38 – Интерфейс оператора склада

Основной интерфейс представляет из себя общую информацию о всех предметах хозяйственной деятельности. Предметы хозяйственной деятельности представлены в табличной форме, которая имеет следующие поля

идентификатор записи предмета, наименование предмета, категория предмета, срок годности, номер акта поступления, номер акта списания.

Перечень предметов представлен в модуле справочник. Следует отметить, что Справочник в программном обеспечении (ПО) является важным инструментом, который предоставляет пользователю быстрый доступ к информации, которая может потребоваться в ходе работы с программой. Это могут быть:

1. Справочные материалы. В справочнике могут быть размещены справочники каталогов, справочники предметов, алфавитные указатели и т.д.

2. Технические данные. В справочнике могут быть размещены технические данные о ПО, такие как максимально возможные параметры работы программы, ограничения на использование определенных функций и т.д.

3. Ссылки на интернет-ресурсы. В справочнике могут быть размещены ссылки на интернет-ресурсы, где пользователь может найти дополнительную информацию о предмете.

5. Функции поиска. Справочник может содержать функции поиска, которые позволяют быстро находить необходимую информацию в большом объеме данных.

Интерес программного модуля справочника представлен на рисунке 39.

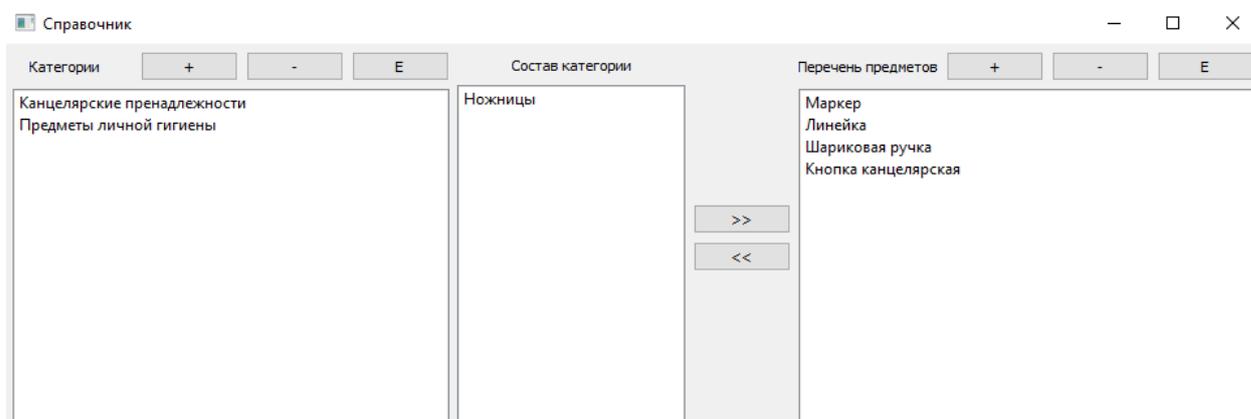


Рисунок 39 – Программный интерфейс модуля справочник

Следует отметить, что интерфейс справочника имеет горизонтальную структуру, а именно можно выделить 3 компоненты: категории, состав категорий и перечень предметов. Для добавления категории необходимо нажать кнопку добавить и откроется интерфейс, представленный на рисунке 40.

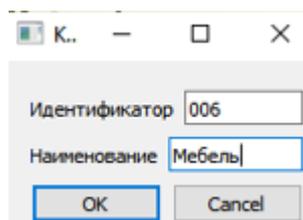


Рисунок 40 - Интерфейс добавления категорий.

Для добавления перечня предметов необходимо в левой части интерфейса нажать на «+» и в открывшемся интерфейсе добавить информацию: идентификатор предмета, наименование предмета и при необходимости ввести описание предмета. На рисунке 41 представлен интерфейс добавления предмета.

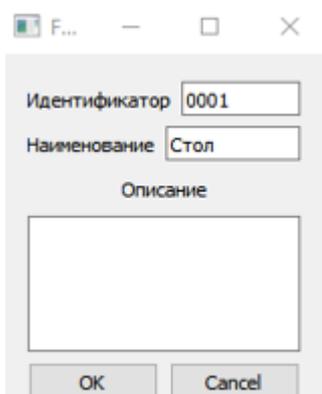


Рисунок 41 – Интерфейс добавления предмета

После добавления всех предметов и категорий необходимо распределить все предметы по категориям. Для добавления предмета в категорию необходимо в правой части выбрать предмет и нажать кнопку «<<<», после чего произойдет добавления предмета в категорию, путем отправки запроса на сервер. Для того, чтобы убрать предмет из категории необходимо в

соответствующем поле (среднее поле) выбрать предмет и нажать на кнопку «>>», что позволит убрать предмет из выбранной категории.

Следует отметить, что модуль справочник заполняется единожды и доступ к этому модулю ограничен. Данные заполненные и привязанные по категориям будет возможность выбирать на этапе поступления предметов. Для добавления данных о поступлениях предметов необходимо, в основном интерфейсе пользователя нажать поступления. Откроется интерфейс добавления поступивших предметов. А именно в открывшемся интерфейсе, представленном на рисунке 34, необходимо выбрать категорию предмета, за предмет хозяйственной деятельности и указать идентификатор. Далее нажать «Добавить акт» и в открывшемся окне выбрать или накладную или приказ, согласно которому заполняются данные.

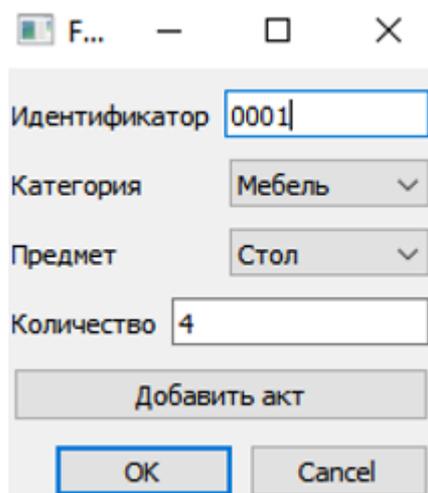


Рисунок 42 – Интерфейс добавления данных

После нажатия на кнопку «Ок» формируется запрос на сервер и в БД записываются данные о новом поступлении. В случае необходимости списания предметов по какой-либо причине необходимо выбрать идентификатор поступления и указать ссылку на распоряжение по списанию предмета хозяйственной деятельности.

## 2.3 Разработка серверной компоненты

### 2.3.1 Структура базы данных

Целью построения физической модели является получение графического представления логической структуры исследуемой предметной области. На рисунке 43 показана физическая модель базы данных для разрабатываемого приложения.

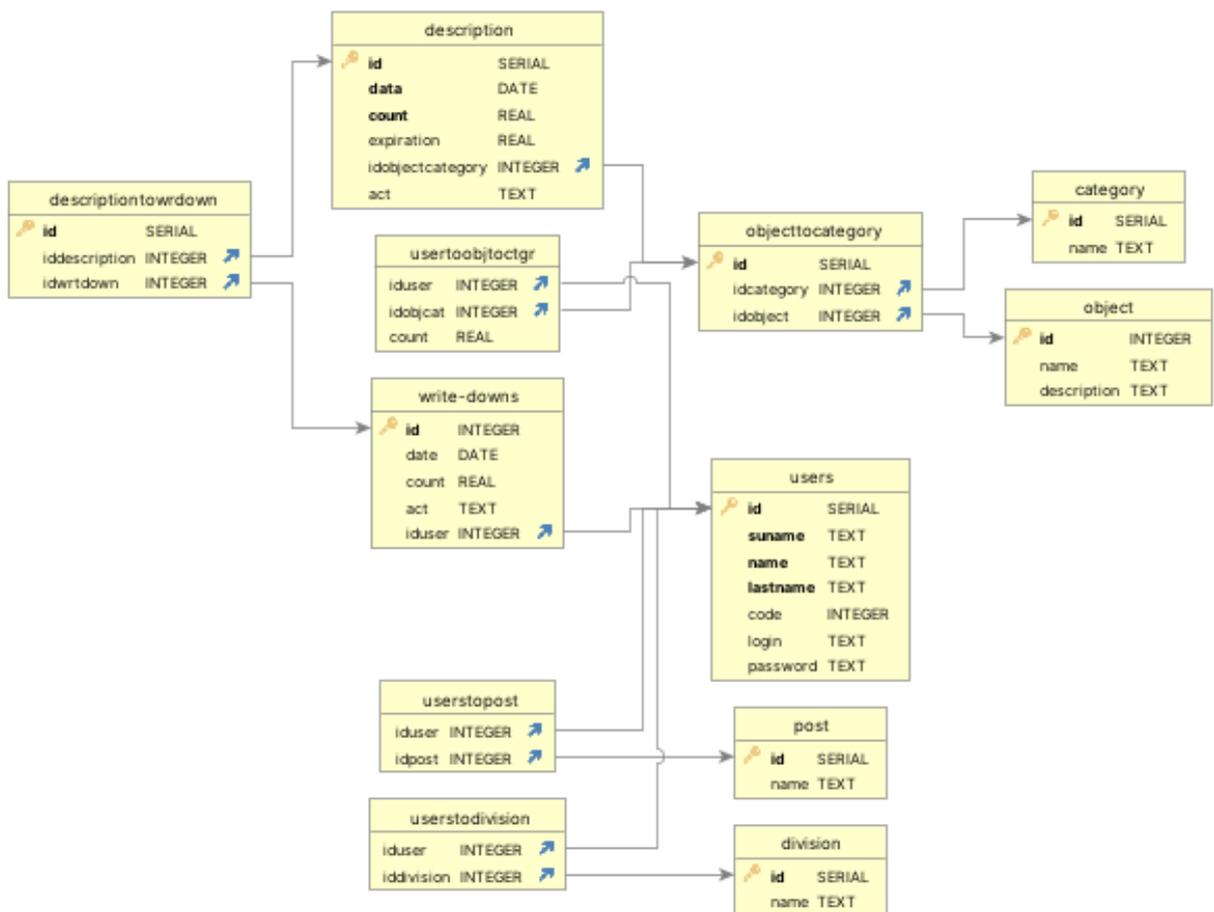


Рисунок 43 – Физическая модель базы данных

Сущность «users» описывает информацию о пользователе.

Атрибуты:

- id (serial) (PK) – уникальный идентификатор пользователя;
- login (text) – хранит уникальный логин пользователя, применимо

для авторизации;

- password (text) – пароль от учетной записи для авторизации;
- name (text) – имя пользователя.
- surname (text) – фамилия пользователя.
- lastname (text) – отчество пользователя.
- code (integer) – уникальный личный код пользователя.

Сущность «post» описывает информацию о должности.

Атрибуты:

- id (serial) (PK) – уникальный идентификатор записи;
- name (text) – хранит наименование должности.

Сущность «division» описывает информацию о должности.

Атрибуты:

- id (serial) (PK) – уникальный идентификатор записи;
- name (text) – хранит наименование отдела.

Таблицы «usertopost» и «usertodivision» связывают данные пользователей с должностью и отделом. При этом следует отметить, что пользователь системы может работать по совместительству в разных структурных подразделениях и на различных должностях.

Сущность «object» описывает информацию о предметах хозяйственной деятельности.

Атрибуты:

- id (serial) (PK) – уникальный идентификатор записи;
- name (text) – хранит наименование предмета.

Сущность «category» описывает информацию о предметах хозяйственной деятельности.

Атрибуты:

- id (serial) (PK) – уникальный идентификатор записи;
- name (text) – хранит наименование категории.

Для связи предметов с категориями используется таблица «objecttcategory».

Для связи пользователей и предметов хозяйственной деятельности используется таблица «usertoobjtoctgr».

Сущность «description» описывает информацию о поступлении предметов хозяйственной деятельности.

Атрибуты:

- id (serial) (PK) – уникальный идентификатор пользователя;
- data(date) – дата поступления предмета;
- count( real) – количество предметов;
- idobjectcategory(integer) – внешний ключ для связи с имеющимися связками предметов и категорий;
- act(text) – хранит ссылку на локальный ресурс с файлом акта.
  
- login (text) – хранит уникальный логин пользователя, применимо для авторизации;
- password (text) – пароль от учетной записи для авторизации;
- name (text) – имя пользователя.
- surname (text) – фамилия пользователя.
- lastname (text) – отчество пользователя.
- code (integer) – уникальный личный код пользователя.

Целью построения физической модели является получение графического представления логической структуры исследуемой предметной области. На рисунке 33 показана физическая модель базы данных для разрабатываемого приложения.

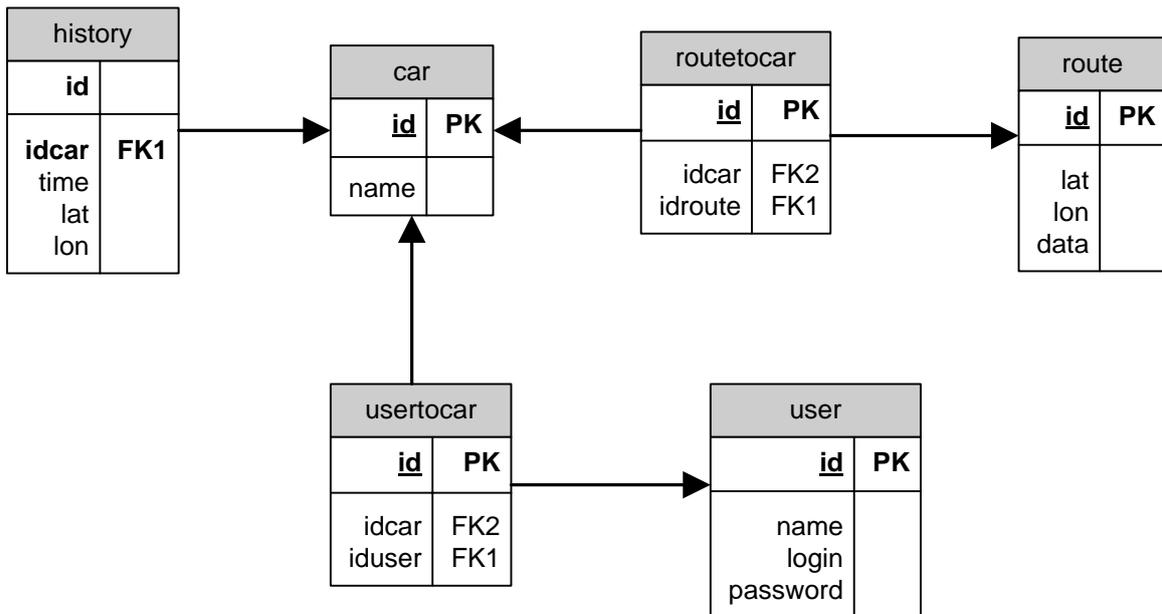


Рисунок 44 – Физическая модель базы данных

Сущность «user» описывает информацию о пользователе.

Атрибуты:

- id (bigint) (PK) – уникальный идентификатор пользователя;
- login (varchar (255)) – хранит уникальный логин пользователя, применимо для авторизации;
- password (varchar (255)) – пароль от учетной записи для авторизации;
- name (varchar (255)) – полное имя пользователя.

Сущность «car» описывает информацию о пользователе.

Атрибуты:

- id (bigint) (PK) – уникальный идентификатор автотранспортного средства;
- name (varchar (255)) – хранит наименование автотранспортного средства.

Сущность «usertocar» описывает информацию о пользователе.

Атрибуты:

- id (bigint) (PK) – уникальный идентификатор автотранспортного средства;

- idcar (bigint) – вторичный ключ, указывающий на связь с идентификатором автотранспортного средства;

- iduser (bigint) – вторичный ключ, указывающий на связь с идентификатором пользователя.

Сущность «history» описывает информацию о пользователе.

Атрибуты:

- id (bigint) (PK) – уникальный идентификатор;

- idcar (bigint) – вторичный ключ, указывающий на связь с идентификатором автотранспортного средства;

- lat (varchar (255)) – текущие показатели широты;

- lon (varchar (255)) – текущие показатели долготы.

- time (date and time) – время и дата записи данных.

Сущность «route» описывает информацию о пользователе.

Атрибуты:

- id (bigint) (PK) – уникальный идентификатор точки маршрута;

- lat (varchar (255)) – показатели широты;

- lon (varchar (255)) – показатели долготы.

- time (date and time) – время и дата записи данных.

Сущность «routetocar» описывает информацию о связи маршрута с автотранспортным средством.

Атрибуты:

- id (bigint) (PK) – уникальный идентификатор автотранспортного средства;

- idcar (bigint) – вторичный ключ, указывающий на связь с идентификатором автотранспортного средства;

- idroute (bigint) – вторичный ключ, указывающий на связь с маршрутом.

### 2.3.1 Разработка алгоритма взаимодействия с клиентом

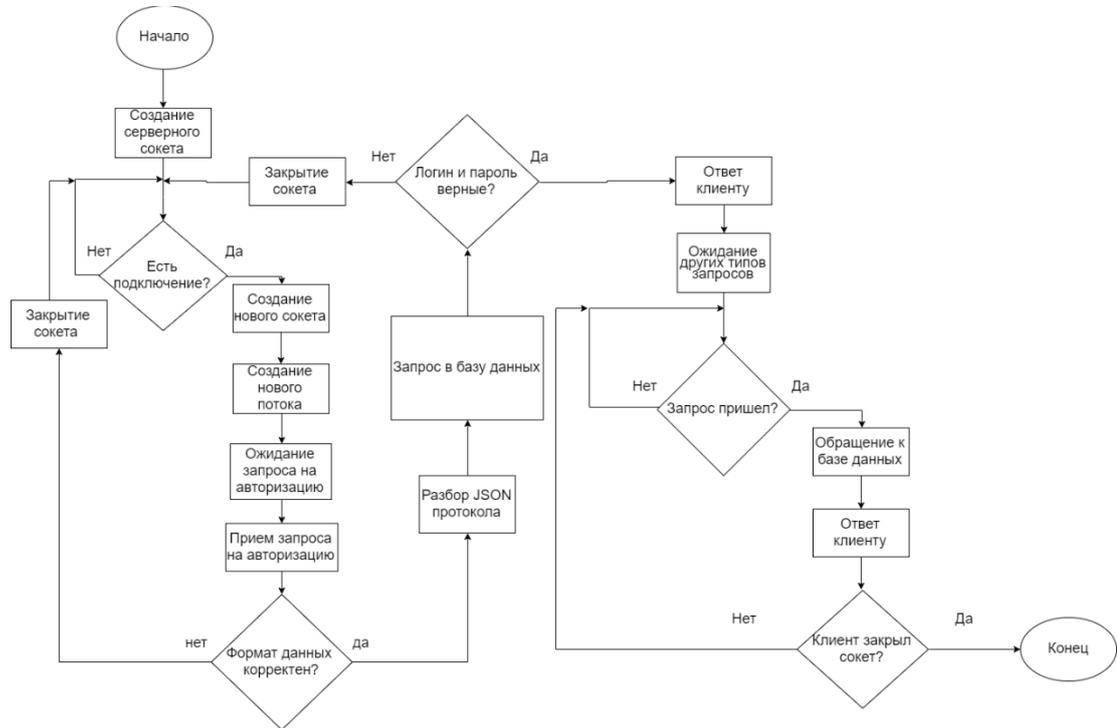


Рисунок 45 – Алгоритм взаимодействия с клиентом

Следует отметить, что серверная компонента при запуске занимает порт 333 и осуществляет ожидание подключения клиентов. После подключения клиента для взаимодействия с ним создается сокет. Созданный сокет передается в класс `dataserverrequesthandler`, который используется в качестве обработчика запросов. В нем создается отдельный поток, в котором осуществляется взаимодействие с клиентами. `Dataserverrequesthandler` является фактически менеджером, который отвечает за распределение информационных потоков между клиентами. После получения данных от клиента осуществляется проверка на корректность протокола. Путем выполнения процедуры десериализации. Если данные корректны JSON пакет будет использоваться в дальнейшей работе в ином случае этот пакет отбрасывается. Следующим этапом является сопоставление логина и пароля пришедших от пользователя с данными, хранящимися в БД. В случае удачной проверки, генерируется ответ в формате JSON. Далее сгенерированный пакет данных отправляется клиенту.

Далее сервер ожидает запросы обработку от клиента. Если запрос поступил, то выполняется процедура, описанная выше. Далее выполняются необходимые действия в зависимости от запроса и формируется ответ пользователю и опять сервер переходит в режим ожидания запроса. В случае если клиент закрыл приложение, то алгоритм работы с сервером прекращает свою работу.

### 2.3.3 Разработка алгоритма взаимодействия с аппаратным блоком

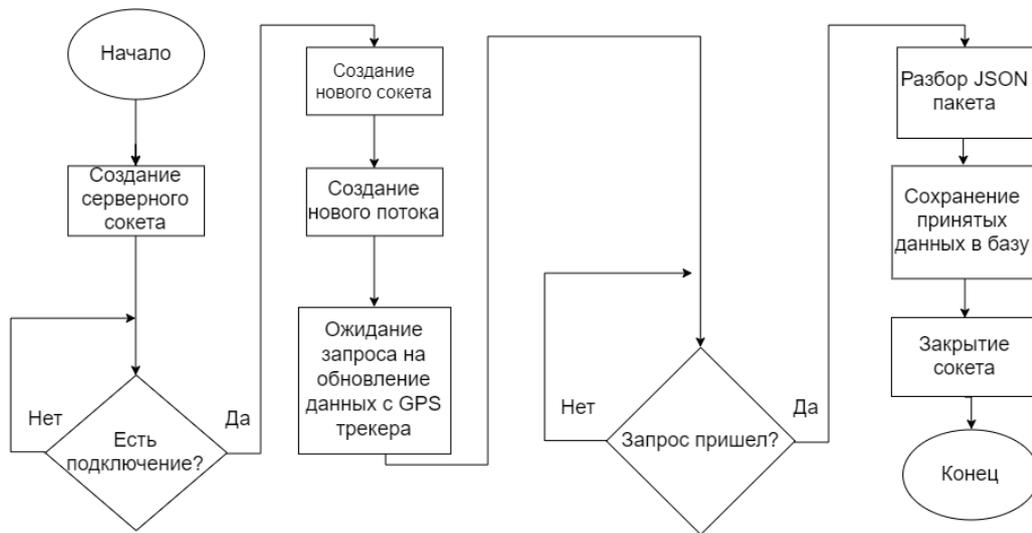


Рисунок 46 – Алгоритм взаимодействия с аппаратным блоком

Алгоритм работы сервера с аппаратным блоком похож на алгоритм работы клиентского ПО. Отличие заключается что сервер занимает порт 4444. А данные от аппаратного блока отправляются по принципу отправил и забыл, т.е. сервер не сохраняет соединение с аппаратным блоком до отключения.

## ЗАКЛЮЧЕНИЕ

В ходе работы была разработана система разработана АИС для по планированию снабжения сырьем и материалами транспортного предприятия. Которая состоит из аппаратной части, серверной компоненты, базы данных и клиентского программного обеспечения. Структура аппаратного блока состоит из модуля питания, телекоммуникационного блока и модуля вычислителя. В качестве реализации модуля телекоммуникации был выбран модуль SIMCOM SIM868, для него была разработана схема электрическая принципиальная, разработан алгоритм обработки AT команд, обеспечивающие запрос диагностических данных, установку GPRS соединения и передачу сообщения. Модуль вычислителя построен на базе микроконтроллера Atmega 328p, для этого модуля была разработано схемотехническое решение и разработан алгоритм работы этого модуля и реализован в исходных текстах программного обеспечения. Для модуля питания была разработана схема электрическая принципиальная.

Для клиентского программного обеспечения был выбран фреймворк Qt. Данный фреймворк позволят писать кроссплатформенные приложения. Реализованная программа позволяет подключаться к серверу, запрашивать и получать данные о месторасположении автотранспорта, маршрутах движения, а также отображать их на карте.

Для разработки серверной компоненты был выбран язык программирования C++. Сервер является многопоточным. В качестве протокола выбран TCP/IP. В качестве формата представления протокола взаимодействия был выбран JSON. В качестве СУБД использовался PostgreSQL.

## СПИСОК ИСПОЛЬЗУЕМОЙ ЛИТЕРАТУРЫ

1. Российская федерация. Законы. Об информации, информационных технологиях и о защите информации [Текст] : Федеральный закон от 27.07.2006 № 149-ФЗ.
2. Российская федерация. Законы. О государственной тайне [Текст] : Федеральный закон от 21 июля 1993 года № 5485-1.
3. Российская Федерация. Министерство юстиции. Об утверждении состава и содержания организационных и технических мер по обеспечению безопасности персональных данных при их обработке в информационных системах персональных данных с использованием средств криптографической защиты информации, необходимых для выполнения установленных Правительством Российской Федерации требований к защите персональных данных для каждого из уровней защищенности [Текст] : Приказ ФСБ Российской федерации от 10.07.2014 № 378.
4. Моргунов Е. П. PostgreSQL. Основы языка SQL / Е. П. Моргунов; под ред. Е. В. Рогова, П. В. Лузанова. — СПб.: БХВ-Петербург, 2018. — 336 с.
5. Джонсон, Говард В., Грэхем, Мартин. «Конструирование высокоскоростных цифровых устройств: начальный курс черной магии.» : Пер. с англ. — М. : Издательский дом “Вильямс”, 2006. — 624 с.
6. Михеев, В.П. Датчики и детекторы / В.П. Михеев, А.В. Просандеев // Учебное пособие. – М.: МИФИ, 2007. – 172 с.
7. Бройдо, В.Л. Вычислительные системы, сети и телекоммуникации / В.Л. Бройдо, О.Н. Ильина – СП-б. : Питер, 2008. – 704 с.
8. Кукин, П.П. Безопасность жизнедеятельности. Безопасность технологических процессов и производств (Охрана труда): Учеб. пособие для вузов / П.П. Кукин, В.Л. Лапин, Н. Л. Пономарев, Н.И. Сердюк М. : Высш. шк., 2004. – 319 с.
9. Кенин, А.М. Самоучитель системного администратора / А.М. Кенин –

2-е изд. перераб. и доп. – Спб. : БХВ-Петербург, 2008. – 560с.

10. Джонсон, Г. В. Конструирование высокоскоростных цифровых устройств: начальный курс черной магии / Г.В. Джонсон // Пер. с англ. — М. : Издательский дом “Вильямс”, 2006. — 624 с.

11. Базилевский М.П. Способ определения параметра  $m$  в задаче частично-булевого линейного программирования для отбора регрессоров в линейной регрессии // Вестник Технологического университета. 2022. Т. 25. № 2. С. 62-66.

12. Антипов О.В., Дмитриева Т.А., Москвитина О.А., Парфилова Н.И. Алгоритмические языки и программирование. Сер. Информатика Том 2 Алгоритмы и структуры данных. Москва, 2022.

13. Вехова Л.Д. Сканирование UDP\TCP портов средствами языка программирования PYTHON 3 // Системный администратор. 2022. № 6 (235). С. 64-67.

14. Мартыненко А.С. Прогнозирование, планирование и программирование как элемент управления социально-экономическими процессами в государстве // Студенческий форум. 2022. № 23-2 (202). С. 52-53.

15. Николаенко В.С. Сравнительный анализ обратной разработки проприетарных программ в зависимости от алгоритмического языка программирования // Вестник студенческого научного общества ГОУ ВПО "Донецкий национальный университет". 2022. Т. 1. № 14. С. 189-192.

16. Рихтер Т.В., Забелин Ю.К. Автоматизация процесса кадрового учета сотрудников организации средствами языка программирования с# // International Journal of Advanced Studies in Computer Engineering. 2022. № 1. С. 55-59.

17. Ткаченко К.С. Программирование аналитического моделирования системы массового обслуживания на языках программирования Fantom и Pike // Системный администратор. 2022. № 10 (239). С. 90-94.

18. Чагуб С., Бентерки Д. Эффективный метод логарифмических

барьеров без линейного поиска для выпуклого квадратичного программирования // Сибирский журнал вычислительной математики. 2022. Т. 25. № 2. С. 193-207.

19. Янарсанов О.А.К. Интерфейс прикладного программирования // Студенческий вестник. 2022. № 4-7 (196). С. 6-8.

20. Яковис Л.М., Стронгин П.Я. Применение линейного программирования для робастной стабилизации многосвязных объектов управления // Математические методы в технологиях и технике. 2022. № 2. С. 7-10.

## ПРИЛОЖЕНИЕ

Листинг файлов тестовой программы

main.cpp

```
#include "widget.h"
#include <QApplication>
#include <QScreen>
#include <QRect>
int main(int argc, char *argv[])
{
    QApplication a(argc, argv);
    Widget w;
    w.showMaximized();
    return a.exec();
}
```

widget.h

```
#ifndef WIDGET_H
#define WIDGET_H
#include <QWidget>
#include <QDebug>
#include <QMessageBox>
#include "addnewprojectwindow.h"
#include "predictionsmodel.h"
#include "infoaboutprojectwnd.h"
#include "productionprojectmodel.h"
#include "projectproductioninfownd.h"
#include "shipmentmodel.h"
#include "startshipmentwindow.h"
#include "addmaterialstoproductwindow.h"
#include "shipmenthistorymodel.h"
#include "historyofshipmentwindow.h"
QT_BEGIN_NAMESPACE
namespace Ui { class Widget; }
QT_END_NAMESPACE
```

```

class Widget : public QWidget
{
    Q_OBJECT
public:
    Widget(QWidget *parent = nullptr);
    ~Widget();
private slots:
    void on_createProject_but_clicked();
    void slotClickedOnPredListObj(const QModelIndex &indx);
    void on_addMaterials_but_clicked();
    void on_getInfoProjectPredictState_but_clicked();
    void on_nextToProductonState_but_clicked();
    void slotClickedOnProductionList(const QModelIndex& indx);
    void on_getInfoProjectProductionState_but_clicked();
    void on_nextToShipmentState_but_clicked();
    void slotClickedOnShipmentList(const QModelIndex& indx);
    void on_configureShipment_but_clicked();
private:
    Ui::Widget *ui;
    PredictionsModel *predModel;
    ProductionProjectModel *productionModel;
    QString tmpPredData;
    QModelIndex tmpPredIndex;
    QString tmpProductionData;
    QModelIndex tmpProductionIndex;
    ShipmentModel *shipmentModel;
    QString tmpShipmentData;
    QModelIndex tmpShipmentIndex;
    ShipmentHistoryModel *shipmentHistoryModel;
    QString dateOfPass;
};
#endif // WIDGET_H
widget.cpp
#include "widget.h"

```

```

#include "ui_widget.h"
Widget::Widget(QWidget *parent)
    : QWidget(parent)
    , ui(new Ui::Widget)
{
    ui->setupUi(this);
    ui->deleteProject_but->setEnabled(false);
    ui->getInfoProjectPredictState_but->setEnabled(false);
    ui->nextToProductonState_but->setEnabled(false);
    ui->nextToShipmentState_but->setEnabled(false);
    ui->getInfoProjectProductionState_but->setEnabled(false);
    ui->configureShipment_but->setEnabled(false);
    ui->addMaterials_but->setEnabled(false);
    predModel = new PredictionsModel(this);
    ui->predictionList->setModel(predModel);
    ui->predictionList->setSpacing(5);
    productionModel = new ProductionProjectModel(this);
    ui->productionList->setModel(productionModel);
    ui->productionList->setSpacing(5);
    shipmentModel = new ShipmentModel(this);
    ui->shipmentList->setModel(shipmentModel);
    ui->shipmentList->setSpacing(5);
    shipmentHistoryModel = new ShipmentHistoryModel(this);
    connect(ui->predictionList, &QListView::clicked, this, &Widget::slotClickedOnPredListObj);
    connect(ui->productionList, &QListView::clicked, this,
    &Widget::slotClickedOnProductionList);
    connect(ui->shipmentList, &QListView::clicked, this, &Widget::slotClickedOnShipmentList);
}
Widget::~Widget()
{
    delete ui;
}
void Widget::on_createProject_but_clicked()
{

```

```

AddNewProjectWindow * addNewProjectWnd = new AddNewProjectWindow;
if(addNewProjectWnd->exec())
{
    predModel->addNewProjectInModel(addNewProjectWnd->getNewProjectName(),
                                    addNewProjectWnd->getNewProjectConstructNumber(),
                                    addNewProjectWnd->getNewProjectCost(),
                                    addNewProjectWnd->getNewProjectDate(),
                                    addNewProjectWnd->getNewProjectCustomer(),
                                    addNewProjectWnd->getNewProjectDescription(),
                                    addNewProjectWnd->getPassDate());
}
delete addNewProjectWnd;
}
void Widget::slotClickedOnPredListObj(const QModelIndex &indx)
{
    qDebug() << indx;
    tmpPredData = predModel->data(indx).toString();
    tmpPredIndex = indx;
    ui->deleteProject_but->setEnabled(true);
    ui->getInfoProjectPredictState_but->setEnabled(true);
    ui->nextToProductonState_but->setEnabled(true);
    ui->addMaterials_but->setEnabled(true);

    connect(ui->deleteProject_but, &QPushButton::clicked, this, [&]() { predModel-
>removeRow(indx.row()); });
}
void Widget::on_addMaterials_but_clicked()
{
    AddMaterialsToProductWindow *addMaterialsWnd = new
AddMaterialsToProductWindow(this);
    if(addMaterialsWnd->exec())
    {

    }
}

```

```

    delete addMaterialsWnd;
}
void Widget::on_getInfoProjectPredictState_but_clicked()
{
    InfoAboutProjectWnd *infoAboutProjectWnd = new InfoAboutProjectWnd(this);
    QString number = predModel->data(tmpPredIndex, predModel->ContractNumber).toString();
    QString title = predModel->data(tmpPredIndex).toString();
    QString date = predModel->data(tmpPredIndex, predModel->DateOfContract).toString();
    QString description = predModel->data(tmpPredIndex, predModel->Description).toString();
    QString cost = predModel->data(tmpPredIndex, predModel->ProductCost).toString() + " руб.";
    QString customer = predModel->data(tmpPredIndex, predModel->Customer).toString();
    QString dateOfpass = predModel->data(tmpPredIndex, predModel->DateOfPass).toString();
    infoAboutProjectWnd->setDataAboutProject(title,number,date,cost,customer,description,
dateOfpass);
    if(infoAboutProjectWnd->exec())
    {
    }
    delete infoAboutProjectWnd;
}
void Widget::on_nextToProductonState_but_clicked()
{
    QString title = predModel->data(tmpPredIndex).toString();
    QString number = predModel->data(tmpPredIndex, predModel->ContractNumber).toString();
    QString date = predModel->data(tmpPredIndex, predModel->DateOfContract).toString();
    QString dateOfPass = predModel->data(tmpPredIndex, predModel->DateOfPass).toString();
    productionModel->addNewObjInModel(title, number,date, dateOfPass);
    predModel->removeRow(tmpPredIndex.row());
    ui->deleteProject_but->setEnabled(false);
    ui->getInfoProjectPredictState_but->setEnabled(false);
    ui->nextToProductonState_but->setEnabled(false);
    ui->addMaterials_but->setEnabled(false);
}
void Widget::slotClickedOnProductionList(const QModelIndex& indx)
{

```

```

qDebug() << "indx: " << indx;
ui->getInfoProjectProductionState_but->setEnabled(true);
ui->nextToShipmentState_but->setEnabled(true);
tmpProductionIndex = indx;
tmpProductionData = productionModel->data(indx).toString();
}
void Widget::on_getInfoProjectProductionState_but_clicked()
{
    ProjectProductionInfoWnd *projectProductionInfoWnd = new ProjectProductionInfoWnd(this);
    QString title = productionModel->data(tmpProductionIndex).toString();
    QString date = productionModel->data(tmpProductionIndex, productionModel->ContractDateOfReception).toString();
    QString number = productionModel->data(tmpProductionIndex, productionModel->ContractNumber).toString();
    QString dateOfPass = productionModel->data(tmpProductionIndex, productionModel->ContractDateOfPass).toString();
    projectProductionInfoWnd->setInformationAboutProduction(title,number,date,dateOfPass);
    if(projectProductionInfoWnd->exec())
    {

    }
    delete projectProductionInfoWnd;
}
void Widget::on_nextToShipmentState_but_clicked()
{
    if(productionModel->data(tmpProductionIndex, productionModel->StateOfProduction) ==
productionModel->GreenState)
    {
        shipmentModel->addNewObjInShipmentModel(tmpProductionData);
        productionModel->removeRow(tmpProductionIndex.row());
        ui->nextToShipmentState_but->setEnabled(false);
        ui->getInfoProjectProductionState_but->setEnabled(false);
    }
    else

```

```

{
    QMessageBox::information(this, "Error!", "Изделие еще не готово!\n Попробуйте позже");
    ui->nextToShipmentState_but->setEnabled(false);
    ui->getInfoProjectProductionState_but->setEnabled(false);
}
}

```

```

void Widget::slotClickedOnShipmentList(const QModelIndex &indx)

```

```

{
    tmpShipmentData = shipmentModel->data(indx).toString();
    tmpShipmentIndex = indx;
    ui->configureShipment_but->setEnabled(true);
}

```

```

void Widget::on_configureShipment_but_clicked()

```

```

{
    StartShipmentWindow *startShipmentWnd = new StartShipmentWindow(this);
    startShipmentWnd->setDataToShipment(shipmentModel->data(shipmentModel-
>index(tmpShipmentIndex.row(),0)).toString());
    if(startShipmentWnd->exec())
    {
        QString name = startShipmentWnd->getProductName();
        QString date = startShipmentWnd->getDate();
        QString avtoNumber = startShipmentWnd->getAvtoNumber();
        QString employee = startShipmentWnd->getEmployee();
        shipmentHistoryModel->addDataInModel(name, date, avtoNumber, employee);
        ui->configureShipment_but->setEnabled(false);
        shipmentModel->removeRow(tmpPredIndex.row());
        connect(ui->shipmentHistory_but, &QPushButton::clicked, this, [&]() {
            HistoryOfShipmentWindow *historyShipmentWnd = new
HistoryOfShipmentWindow(this);
            historyShipmentWnd->setHistoryModel(shipmentHistoryModel);
            if(historyShipmentWnd->exec())

```

```

    {

    }
    delete historyShipmentWnd;
});
}
delete startShipmentWnd;
}
widget.ui

```

```

<?xml version="1.0" encoding="UTF-8"?>
<ui version="4.0">
<class>Widget</class>
<widget class="QWidget" name="Widget">
<property name="geometry">
<rect>
<x>0</x>
<y>0</y>
<width>1055</width>
<height>602</height>
</rect>
</property>
<property name="windowTitle">
<string>Widget</string>
</property>
<layout class="QHBoxLayout" name="horizontalLayout_2">
<item>
<layout class="QVBoxLayout" name="verticalLayout">
<item>
<widget class="QLabel" name="label">
<property name="font">
<font>
<pointsize>14</pointsize>
<weight>75</weight>

```

```

    <bold>true</bold>
    <underline>true</underline>
  </font>
</property>
<property name="text">
  <string>Предварительный этап</string>
</property>
<property name="alignment">
  <set>Qt::AlignCenter</set>
</property>
</widget>
</item>
<item>
  <layout class="QHBoxLayout" name="horizontalLayout">
    <item>
      <widget class="QPushButton" name="createProject_but">
        <property name="font">
          <font>
            <pointsize>10</pointsize>
            <weight>75</weight>
            <bold>true</bold>
          </font>
        </property>
        <property name="text">
          <string>Создать</string>
        </property>
      </widget>
    </item>
    <item>
      <widget class="QPushButton" name="deleteProject_but">
        <property name="font">
          <font>
            <pointsize>10</pointsize>
            <weight>75</weight>

```

```

    <bold>true</bold>
  </font>
</property>
<property name="text">
  <string>Удалить</string>
</property>
</widget>
</item>
<item>
  <widget class="QPushButton" name="getInfoProjectPredictState_but">
    <property name="sizePolicy">
      <sizepolicy hstretch="Expanding" vsizetype="Fixed">
        <horstretch>0</horstretch>
        <verstretch>0</verstretch>
      </sizepolicy>
    </property>
    <property name="font">
      <font>
        <pointsize>10</pointsize>
        <weight>75</weight>
        <bold>true</bold>
      </font>
    </property>
    <property name="text">
      <string>Информация о проекте</string>
    </property>
  </widget>
</item>
</layout>
</item>
<item>
  <widget class="QPushButton" name="addMaterials_but">
    <property name="font">
      <font>

```

```

    <pointsize>10</pointsize>
    <weight>75</weight>
    <bold>true</bold>
  </font>
</property>
<property name="text">
  <string>Добавить материалы</string>
</property>
</widget>
</item>
<item>
  <widget class="QListView" name="predictionList"/>
</item>
</layout>
</item>
<item>
  <widget class="QPushButton" name="nextToProductonState_but">
    <property name="font">
      <font>
        <pointsize>12</pointsize>
        <weight>75</weight>
        <bold>true</bold>
      </font>
    </property>
    <property name="text">
      <string>-&gt;</string>
    </property>
  </widget>
</item>
<item>
  <layout class="QVBoxLayout" name="verticalLayout_2">
    <item>
      <widget class="QLabel" name="label_2">
        <property name="font">

```

```

<font>
  <pointsize>14</pointsize>
  <weight>75</weight>
  <bold>true</bold>
  <underline>true</underline>
</font>
</property>
<property name="text">
  <string>Этап изготовления</string>
</property>
<property name="alignment">
  <set>Qt::AlignCenter</set>
</property>
</widget>
</item>
<item>
  <widget class="QPushButton" name="getInfoProjectProductionState_but">
    <property name="font">
      <font>
        <pointsize>10</pointsize>
        <weight>75</weight>
        <bold>true</bold>
      </font>
    </property>
    <property name="text">
      <string>Информация об изготовлении</string>
    </property>
  </widget>
</item>
<item>
  <widget class="QListView" name="productionList"/>
</item>
</layout>
</item>

```

```
<item>
  <widget class="QPushButton" name="nextToShipmentState_but">
    <property name="font">
      <font>
        <pointsize>12</pointsize>
        <weight>75</weight>
        <bold>true</bold>
      </font>
    </property>
    <property name="text">
      <string>-&gt;</string>
    </property>
  </widget>
</item>
<item>
  <layout class="QVBoxLayout" name="verticalLayout_3">
    <item>
      <widget class="QLabel" name="label_3">
        <property name="font">
          <font>
            <pointsize>14</pointsize>
            <weight>75</weight>
            <bold>true</bold>
            <underline>true</underline>
          </font>
        </property>
        <property name="text">
          <string>Этап отгрузки</string>
        </property>
        <property name="alignment">
          <set>Qt::AlignCenter</set>
        </property>
      </widget>
    </item>
```

```

<item>
<layout class="QHBoxLayout" name="horizontalLayout_3">
  <item>
    <widget class="QPushButton" name="configureShipment_but">
      <property name="font">
        <font>
          <pointsize>10</pointsize>
          <weight>75</weight>
          <bold>true</bold>
        </font>
      </property>
      <property name="text">
        <string>Начать отгрузку</string>
      </property>
    </widget>
  </item>
  <item>
    <widget class="QPushButton" name="shipmentHistory_but">
      <property name="font">
        <font>
          <pointsize>10</pointsize>
          <weight>75</weight>
          <bold>true</bold>
        </font>
      </property>
      <property name="text">
        <string>История отгрузок</string>
      </property>
    </widget>
  </item>
</layout>
</item>
<item>
  <widget class="QListView" name="shipmentList"/>

```

```

    </item>
</layout>
</item>
</layout>
</widget>
<resources/>
<connections/>
</ui>
whatsmeancolorswindow.h
#ifndef WHATSMEANCOLORSWINDOW_H
#define WHATSMEANCOLORSWINDOW_H

#include <QDialog>

namespace Ui {
class WhatsMeanColorsWindow;
}

class WhatsMeanColorsWindow : public QDialog
{
    Q_OBJECT

public:
    explicit WhatsMeanColorsWindow(QWidget *parent = nullptr);
    ~WhatsMeanColorsWindow();

private:
    Ui::WhatsMeanColorsWindow *ui;
};

#endif // WHATSMEANCOLORSWINDOW_H

whatsmeancolorswindow.cpp

```

```
#include "whatsmeancolorswindow.h"
#include "ui_whatsmeancolorswindow.h"

WhatsMeanColorsWindow::WhatsMeanColorsWindow(QWidget *parent) :
    QDialog(parent),
    ui(new Ui::WhatsMeanColorsWindow)
{
    ui->setupUi(this);

    connect(ui->close_but, &QPushButton::clicked,this, &QDialog::reject);
}
```

```
WhatsMeanColorsWindow::~WhatsMeanColorsWindow()
{
    delete ui;
}
```

WhatsMeanColorsWindow.ui

```
<?xml version="1.0" encoding="UTF-8"?>
<ui version="4.0">
<class>WhatsMeanColorsWindow</class>
<widget class="QDialog" name="WhatsMeanColorsWindow">
<property name="geometry">
<rect>
<x>0</x>
<y>0</y>
<width>321</width>
<height>118</height>
</rect>
</property>
<property name="windowTitle">
<string>Dialog</string>
</property>
<layout class="QVBoxLayout" name="verticalLayout">
<item>
```

```

<layout class="QHBoxLayout" name="horizontalLayout">
  <item>
    <widget class="QLabel" name="label">
      <property name="styleSheet">
        <string notr="true">background-color: rgb(255, 0, 0);</string>
      </property>
      <property name="text">
        <string/>
      </property>
    </widget>
  </item>
  <item>
    <widget class="QLabel" name="label_2">
      <property name="font">
        <font>
          <pointsize>10</pointsize>
          <weight>75</weight>
          <bold>true</bold>
        </font>
      </property>
      <property name="text">
        <string>Не принят ни одним из рабочих</string>
      </property>
    </widget>
  </item>
</layout>
</item>
<item>
  <layout class="QHBoxLayout" name="horizontalLayout_2">
    <item>
      <widget class="QLabel" name="label_3">
        <property name="styleSheet">
          <string notr="true">background-color: rgb(4, 255, 0);</string>
        </property>

```

```
<property name="text">
  <string/>
</property>
</widget>
</item>
<item>
  <widget class="QLabel" name="label_4">
    <property name="font">
      <font>
        <pointsize>10</pointsize>
        <weight>75</weight>
        <bold>true</bold>
      </font>
    </property>
    <property name="text">
      <string>Принят рабочими на изготовку</string>
    </property>
  </widget>
</item>
</layout>
</item>
<item>
  <widget class="QPushButton" name="close_but">
    <property name="font">
      <font>
        <pointsize>10</pointsize>
        <weight>75</weight>
        <bold>true</bold>
      </font>
    </property>
    <property name="text">
      <string>Закрывать </string>
    </property>
  </widget>
```

```
</item>
</layout>
</widget>
<resources/>
<connections/>
</ui>
```

```
startshipmentwindwos.h
```

```
#ifndef STARTSHIPMENTWINDOW_H
#define STARTSHIPMENTWINDOW_H
```

```
#include <QDialog>
```

```
namespace Ui {
class StartShipmentWindow;
}
```

```
class StartShipmentWindow : public QDialog
{
    Q_OBJECT
```

```
public:
```

```
    explicit StartShipmentWindow(QWidget *parent = nullptr);
    ~StartShipmentWindow();
```

```
    void setDataToShipment(const QString& name);
```

```
    QString getProductName();
```

```
    QString getEmployee();
```

```
    QString getDate();
```

```
    QString getAvtoNumber();
```

```
private:
```

```
    Ui::StartShipmentWindow *ui;
```

```

};

#endif // STARTSHIPMENTWINDOW_H

startshipmentwindows.cpp

#include "startshipmentwindow.h"
#include "ui_startshipmentwindow.h"

StartShipmentWindow::StartShipmentWindow(QWidget *parent) :
    QDialog(parent),
    ui(new Ui::StartShipmentWindow)
{
    ui->setupUi(this);

    QStringList lstAuto;
    lstAuto << "A777AAA136" << "E666KH777" << "X000XX007";

    ui->comboBox->addItem(lstAuto);

    connect(ui->back_but, &QPushButton::clicked, this, &QDialog::reject);
    connect(ui->shipmentProduct_but, &QPushButton::clicked, this, &QDialog::accept);
}

StartShipmentWindow::~StartShipmentWindow()
{
    delete ui;
}

void StartShipmentWindow::setDataToShipment(const QString &name)
{
    ui->productName_lbl->setText(name);
}

```

```
QString StartShipmentWindow::getProductName()
{
    return ui->productName_lbl->text();
}
```

```
QString StartShipmentWindow::getEmployee()
{
    return ui->workerCode_input->text();
}
```

```
QString StartShipmentWindow::getDate()
{
    return ui->dateShipment_input->text();
}
```

```
QString StartShipmentWindow::getAvtoNumber()
{
    return ui->comboBox->currentText();
}
```

startshipmentwindows.ui

```
<?xml version="1.0" encoding="UTF-8"?>
<ui version="4.0">
<class>StartShipmentWindow</class>
<widget class="QDialog" name="StartShipmentWindow">
    <property name="geometry">
        <rect>
            <x>0</x>
            <y>0</y>
            <width>535</width>
            <height>301</height>
        </rect>
    </property>
```

```

<property name="windowTitle">
  <string>Dialog</string>
</property>
<layout class="QVBoxLayout" name="verticalLayout">
  <item>
    <widget class="QLabel" name="label">
      <property name="sizePolicy">
        <sizepolicy hstretch="Preferred" vsizetype="Maximum">
          <horstretch>0</horstretch>
          <verstretch>0</verstretch>
        </sizepolicy>
      </property>
      <property name="font">
        <font>
          <pointsize>14</pointsize>
          <weight>75</weight>
          <bold>true</bold>
          <underline>true</underline>
        </font>
      </property>
      <property name="text">
        <string>Отгрузка проекта</string>
      </property>
      <property name="alignment">
        <set>Qt::AlignCenter</set>
      </property>
    </widget>
  </item>
  <item>
    <layout class="QHBoxLayout" name="horizontalLayout_2">
      <item>
        <widget class="QLabel" name="label_3">
          <property name="sizePolicy">
            <sizepolicy hstretch="Fixed" vsizetype="Maximum">

```

```

    <horstretch>0</horstretch>
    <verstretch>0</verstretch>
</sizepolicy>
</property>
<property name="font">
    <font>
        <pointsize>10</pointsize>
        <weight>75</weight>
        <bold>true</bold>
    </font>
</property>
<property name="text">
    <string>Название:</string>
</property>
</widget>
</item>
<item>
    <widget class="QLabel" name="productName_lbl">
        <property name="sizePolicy">
            <sizepolicy hstypе="Preferred" vsizetype="Maximum">
                <horstretch>0</horstretch>
                <verstretch>0</verstretch>
            </sizepolicy>
        </property>
        <property name="font">
            <font>
                <pointsize>10</pointsize>
                <weight>75</weight>
                <bold>true</bold>
            </font>
        </property>
        <property name="text">
            <string>Название продукта</string>
        </property>

```

```

</widget>
</item>
</layout>
</item>
<item>
<layout class="QHBoxLayout" name="horizontalLayout">
<item>
<widget class="QLabel" name="label_2">
<property name="sizePolicy">
<sizepolicy hstretch="Fixed" vstretch="Preferred">
<horstretch>0</horstretch>
<verstretch>0</verstretch>
</sizepolicy>
</property>
<property name="font">
<font>
<pointsize>10</pointsize>
<weight>75</weight>
<bold>true</bold>
</font>
</property>
<property name="text">
<string>Введите дату отгрузки:</string>
</property>
</widget>
</item>
<item>
<widget class="QLineEdit" name="dateShipment_input"/>
</item>
</layout>
</item>
<item>
<layout class="QHBoxLayout" name="horizontalLayout_3">
<item>

```

```

<widget class="QLabel" name="label_4">
  <property name="sizePolicy">
    <sizepolicy hstypе="Fixed" vstypе="Preferred">
      <horstretch>0</horstretch>
      <verstretch>0</verstretch>
    </sizepolicy>
  </property>
  <property name="font">
    <font>
      <pointsize>10</pointsize>
      <weight>75</weight>
      <bold>true</bold>
    </font>
  </property>
  <property name="text">
    <string>Номер транспортного средства</string>
  </property>
</widget>
</item>
<item>
  <widget class="QComboBox" name="comboBox">
    <property name="font">
      <font>
        <pointsize>10</pointsize>
        <weight>75</weight>
        <bold>true</bold>
      </font>
    </property>
  </widget>
</item>
</layout>
</item>
<item>
  <layout class="QHBoxLayout" name="horizontalLayout_4">

```

```

<item>
  <widget class="QLabel" name="label_5">
    <property name="sizePolicy">
      <sizepolicy hsize="Fixed" vsize="Preferred">
        <horstretch>0</horstretch>
        <verstretch>0</verstretch>
      </sizepolicy>
    </property>
    <property name="font">
      <font>
        <pointsize>10</pointsize>
        <weight>75</weight>
        <bold>true</bold>
      </font>
    </property>
    <property name="text">
      <string>Введите код сотрудника, отгрузившего товар:</string>
    </property>
  </widget>
</item>
<item>
  <widget class="QLineEdit" name="workerCode_input"/>
</item>
</layout>
</item>
<item>
  <layout class="QHBoxLayout" name="horizontalLayout_5">
    <item>
      <widget class="QPushButton" name="back_but">
        <property name="font">
          <font>
            <pointsize>10</pointsize>
            <weight>75</weight>
            <bold>true</bold>
          </font>
        </property>
      </widget>
    </item>
  </layout>
</item>

```

```

</font>
</property>
<property name="text">
  <string>Вернуться назад</string>
</property>
</widget>
</item>
<item>
  <widget class="QPushButton" name="shipmentProduct_but">
    <property name="font">
      <font>
        <pointsize>10</pointsize>
        <weight>75</weight>
        <bold>true</bold>
      </font>
    </property>
    <property name="text">
      <string>Отгрузить товар</string>
    </property>
  </widget>
</item>
</layout>
</item>
</layout>
</widget>
<resources/>
<connections/>
</ui>

```

shipmentmodel.h

```

#ifndef SHIPMENTMODEL_H
#define SHIPMENTMODEL_H

```

```

#include <QObject>
#include <QStandardItemModel>

class ShipmentModel : public QStandardItemModel
{
    Q_OBJECT
public:
    explicit ShipmentModel(QObject *parent = nullptr);

    void addNewObjInShipmentModel(const QString& value);

signals:

};

#endif // SHIPMENTMODEL_H

```

shipmentmodel.cpp

```
#include "shipmentmodel.h"
```

```

ShipmentModel::ShipmentModel(QObject *parent) : QStandardItemModel(parent)
{
    insertColumn(0);
}

```

```

void ShipmentModel::addNewObjInShipmentModel(const QString &value)
{
    insertRow(rowCount());
    setData(index(rowCount()-1,0),value);
}

```

shipmentthistorymodel.h

```

#ifndef SHIPMENTHISTORYMODEL_H
#define SHIPMENTHISTORYMODEL_H

#include <QObject>
#include <QStandardItemModel>

class ShipmentHistoryModel : public QStandardItemModel
{
    Q_OBJECT
public:
    explicit ShipmentHistoryModel(QObject *parent = nullptr);

    void addDataInModel(const QString& name, const QString& date, const QString& avtoNumber,
const QString& employee);

signals:

};

#endif // SHIPMENTHISTORYMODEL_H

```

shipmentthisstorymodel.cpp

```
#include "shipmenthistorymodel.h"
```

```

ShipmentHistoryModel::ShipmentHistoryModel(QObject *parent) : QStandardItemModel(parent)
{
    insertColumns(0, 4);

    setHeaderData(0, Qt::Horizontal, "Название товара");
    setHeaderData(1, Qt::Horizontal, "Дата отгрузки");
    setHeaderData(2, Qt::Horizontal, "Транспортное средство");
    setHeaderData(3, Qt::Horizontal, "Код сотрудника");
}

```

```

void ShipmentHistoryModel::addDataInModel(const QString &name, const QString &date, const
QString &avtoNumber, const QString &employee)
{
    insertRow(rowCount());
    setData(index(rowCount()-1, 0), name);
    setData(index(rowCount()-1, 1), date);
    setData(index(rowCount()-1, 2), avtoNumber);
    setData(index(rowCount()-1, 3), employee);
}

```

setdataofinishproductionwindwos.h

```

#ifndef SETDATATOFINISHPRODUCTIONWND_H
#define SETDATATOFINISHPRODUCTIONWND_H

#include <QDialog>

namespace Ui {
class SetDataToFinishProductionWnd;
}

class SetDataToFinishProductionWnd : public QDialog
{
    Q_OBJECT

public:
    explicit SetDataToFinishProductionWnd(QWidget *parent = nullptr);
    ~SetDataToFinishProductionWnd();

    QString getDateOfFinishProduction();

private:
    Ui::SetDataToFinishProductionWnd *ui;

```

```

};

#endif // SETDATATOFINISHPRODUCTIONWND_H

setdatatofinishproductionwind.cpp

#include "setdatatofinishproductionwnd.h"
#include "ui_setdatatofinishproductionwnd.h"

SetDataToFinishProductionWnd::SetDataToFinishProductionWnd(QWidget *parent) :
    QDialog(parent),
    ui(new Ui::SetDataToFinishProductionWnd)
{
    ui->setupUi(this);

    connect(ui->back_but, &QPushButton::clicked, this, &SetDataToFinishProductionWnd::reject);
    connect(ui->save_but, &QPushButton::clicked, this,
    &SetDataToFinishProductionWnd::accept);
}

SetDataToFinishProductionWnd::~SetDataToFinishProductionWnd()
{
    delete ui;
}

QString SetDataToFinishProductionWnd::getDateOfFinishProduction()
{
    return ui->date_input->text();
}

projectproduction.h
#ifndef PROJECTPRODUCTIONINFOWND_H
#define PROJECTPRODUCTIONINFOWND_H

```

```

#include <QDialog>

#include "whatsmeancolorswindow.h"
#include "setdatatofinishproductionwnd.h"

namespace Ui {
class ProjectProductionInfoWnd;
}

class ProjectProductionInfoWnd : public QDialog
{
    Q_OBJECT

public:
    explicit ProjectProductionInfoWnd(QWidget *parent = nullptr);
    ~ProjectProductionInfoWnd();

    void setInformationAboutProduction(const QString& title, const QString& number, const
        QString& dateOfReception,
        const QString& dateOfpass);

private slots:
    void on_whatIsColor_but_clicked();

    void on_addSrokSdachi_but_clicked();

private:
    Ui::ProjectProductionInfoWnd *ui;
};

#endif // PROJECTPRODUCTIONINFOWND_H

projectproduction.cpp

```

```

#include "projectproductioninfownd.h"
#include "ui_projectproductioninfownd.h"

#include <QtDebug>

ProjectProductionInfoWnd::ProjectProductionInfoWnd(QWidget *parent) :
    QDialog(parent),
    ui(new Ui::ProjectProductionInfoWnd)
{
    ui->setupUi(this);

    connect(ui->close_but, &QPushButton::clicked, this, &QDialog::reject);
}

ProjectProductionInfoWnd::~ProjectProductionInfoWnd()
{
    delete ui;
}

void ProjectProductionInfoWnd::setInformationAboutProduction(const QString &title, const
    QString &number,
    const QString &dateOfReception,
    const QString &dateOfPass)
{
    ui->date_of_lbl->setText(dateOfReception);
    ui->productName_lbl->setText(title);
    ui->number_lbl->setText(number);
    ui->dateOfPass_lbl->setText(dateOfPass);
}

void ProjectProductionInfoWnd::on_whatIsColor_but_clicked()
{
    WhatsMeanColorsWindow * whatsMeanColorWnd = new WhatsMeanColorsWindow(this);
}

```

```

if(whatsMeanColorWnd->exec())
{

}
delete whatsMeanColorWnd;
}

void ProjectProductionInfoWnd::on_addSrokSdachi_but_clicked()
{
    SetDataToFinishProductionWnd * dateToFinishProductionWnd = new
SetDataToFinishProductionWnd(this);
    if(dateToFinishProductionWnd->exec())
    {
        int raznica = 0;
        ui->dateOfPass_lbl->setText(dateToFinishProductionWnd->getDateOfFinishProduction());
        QStringList lstPass = dateToFinishProductionWnd->getDateOfFinishProduction().split('.');
        QStringList lstReception = ui->date_of_lbl->text().split('.');
        qDebug () << lstPass << lstReception;
        if(lstPass[2].toInt() > lstReception[2].toInt())
        {
            raznica += (lstPass[2].toInt() - lstReception[2].toInt()) * 365;
        }
        if(lstPass[1].toInt() > lstReception[1].toInt())
        {
            raznica += (lstPass[1].toInt() > lstReception[1].toInt()) * 30;
        }
        if(lstPass[0].toInt() > lstReception[0].toInt())
        {
            raznica += (lstPass[0].toInt() > lstReception[0].toInt());
        }
        qDebug() << raznica;
    }
    delete dateToFinishProductionWnd;
}

```

```
}
```

```
productionproject.h
```

```
#ifndef PRODUCTIONPROJECTMODEL_H  
#define PRODUCTIONPROJECTMODEL_H
```

```
#include <QObject>  
#include <QStandardItemModel>
```

```
class ProductionProjectModel : public QStandardItemModel  
{  
    Q_OBJECT  
public:  
    explicit ProductionProjectModel(QObject *parent = nullptr);
```

```
enum stateProduct {  
    RedState = 1,  
    YellowState,  
    GreenState,  
};
```

```
enum modelRoles {  
    ContractNumber = Qt::UserRole + 1,  
    ContractDateOfReception,  
    ContractDateOfPass,  
    StateOfProduction,  
    Employee,  
};
```

```
void addNewObjInModel(const QString& title, const QString& contractNumber,  
    const QString& dateOfReception,  
    const QString& dateOfPass = "Не задана",  
    const QString& employee = "Не задан",
```

```
const QString& state = QString::number(GreenState));
```

```
signals:
```

```
};
```

```
#endif // PRODUCTIONPROJECTMODEL_H
```

```
productionproject.cpp
```

```
#include "productionprojectmodel.h"
```

```
#include "projectproductioninfownd.h"
```

```
ProductionProjectModel::ProductionProjectModel(QObject *parent) :
```

```
QStandardItemModel(parent)
```

```
{
```

```
    insertColumn(0);
```

```
}
```

```
void ProductionProjectModel::addNewObjInModel(const QString &title, const QString  
&contractNumber, const QString &dateOfReception,
```

```
const QString &dateOfPass,const QString &employee, const QString
```

```
&state)
```

```
{
```

```
    insertRow(rowCount());
```

```
    setData(index(rowCount()-1,0), title);
```

```
    setData(index(rowCount()-1,0), contractNumber, ContractNumber);
```

```
    setData(index(rowCount()-1,0), dateOfReception, ContractDateOfReception);
```

```
    setData(index(rowCount()-1,0), dateOfPass, ContractDateOfPass);
```

```
    setData(index(rowCount()-1,0), employee, Employee);
```

```
    setData(index(rowCount()-1,0), state, StateOfProduction);
```

```
    switch (state.toInt()) {
```

```
    case GreenState:
```

```

{
    // setData(index(rowCount()-1,0), QColor(Qt::green), Qt::BackgroundColorRole);
    break;
}
case RedState:
{
    // setData(index(rowCount()-1,0), QColor(Qt::red), Qt::BackgroundColorRole);
    break;
}
case YellowState:
{
    // setData(index(rowCount()-1,0), QColor(Qt::yellow), Qt::BackgroundColorRole);
    break;
}
}

```

predictionmodel.h

```

#ifndef PREDICTIONSMODEL_H
#define PREDICTIONSMODEL_H

#include <QObject>
#include <QStandardItemModel>

class PredictionsModel : public QStandardItemModel
{
    Q_OBJECT
public:
    explicit PredictionsModel(QObject *parent = nullptr);

    enum modelRoles {
        ContractNumber = Qt::UserRole + 1,
        ProductCost,

```

```

        DateOfContract,
        Customer,
        Description,
        DateOfPass
    };

    void addNewProjectInModel(const QString& title, const QString& contractNumber, const
    QString& cost,
                                const QString& dateContract, const QString& customer, const QString&
    description,
                                const QString& dateOfPass);

signals:

};

#endif // PREDICTIONSMODEL_H

```

predictionmodel.cpp

```
#include "predictionsmodel.h"
```

```

PredictionsModel::PredictionsModel(QObject *parent) : QStandardItemModel(parent)
{
    insertColumn(0);
    insertRows(0,4);

    setData(index(0,0), "Стул");
    setData(index(0,0), "123123213", ContractNumber);
    setData(index(0,0), "8000", ProductCost);
    setData(index(0,0), "12.02.2013", DateOfContract);
    setData(index(0,0), "АО \"Концерн Созвездие\"", Customer);
    setData(index(0,0), "Отличная стремянка", Description);
    setData(index(0,0), "12.03.2014", DateOfPass);
}

```

```
setData(index(1,0), "Диван");
setData(index(1,0), "123123213", ConstructNumber);
setData(index(1,0), "8000", ProductCost);
setData(index(1,0), "12.02.2013", DateOfContract);
setData(index(1,0), "АО \"Концерн Созвездие\"", Customer);
setData(index(1,0), "Отличная стремянка", Description);
setData(index(1,0), "12.03.2014", DateOfPass);
```

```
setData(index(2,0), "Стол");
setData(index(2,0), "123123213", ConstructNumber);
setData(index(2,0), "8000", ProductCost);
setData(index(2,0), "12.02.2013", DateOfContract);
setData(index(2,0), "АО \"Концерн Созвездие\"", Customer);
setData(index(2,0), "Отличная стремянка", Description);
setData(index(2,0), "12.03.2014", DateOfPass);
```

```
setData(index(3,0), "Шкаф");
setData(index(3,0), "123123213", ConstructNumber);
setData(index(3,0), "8000", ProductCost);
setData(index(3,0), "12.02.2013", DateOfContract);
setData(index(3,0), "АО \"Концерн Созвездие\"", Customer);
setData(index(3,0), "Отличная стремянка", Description);
setData(index(3,0), "12.03.2014", DateOfPass);
```

```
}
```

```
void PredictionsModel::addNewProjectInModel(const QString &title, const QString
&contractNumber, const QString &cost,
const QString &dateContract, const QString &customer, const QString
&description,
const QString& dateOfPass)
```

```
{
```

```

insertRow(rowCount());
setData(index(rowCount()-1,0), title);
setData(index(rowCount()-1,0), contractNumber, ContractNumber);
setData(index(rowCount()-1,0), cost, ProductCost);
setData(index(rowCount()-1,0), dateContract, DateOfContract);
setData(index(rowCount()-1,0), customer, Customer);
setData(index(rowCount()-1,0), description, Description);
setData(index(rowCount()-1,0), dateOfPass, DateOfPass);
}

```

materialsmodel.h

```

#ifndef MATERIALSSKLADMODEL_H
#define MATERIALSSKLADMODEL_H

#include <QObject>
#include <QStandardItemModel>

class MaterialsSkladModel : public QStandardItemModel
{
    Q_OBJECT
public:
    explicit MaterialsSkladModel(QObject *parent = nullptr);

    void addMaterialDataInModel(QString newCount, QString material);

    QModelIndex checkObjModel(const QString& material);

signals:

};

#endif // MATERIALSSKLADMODEL_H

```

materialskladmodel.cpp

```
#include "materialsskladmodel.h"
```

```
MaterialsSkladModel::MaterialsSkladModel(QObject *parent) : QStandardItemModel(parent)
```

```
{  
    insertColumns(0,2);  
    setHeaderData(0, Qt::Horizontal, "Название материала");  
    setHeaderData(1, Qt::Horizontal, "Количество на складе");  
  
    insertRow(rowCount());  
    setData(index(rowCount()-1,0), "Стремянка");  
    setData(index(rowCount()-1, 1), "250");  
  
    insertRow(rowCount());  
    setData(index(rowCount()-1,0), "Стремянка2");  
    setData(index(rowCount()-1, 1), "250");  
}
```

```
void MaterialsSkladModel::addMaterialDataInModel(QString newCount, QString material)
```

```
{  
    QModelIndex curIndx = checkObjModel(material);  
    if(curIndx.isValid())  
    {  
        int curRow = curIndx.row();  
        int now = data(index(curRow, 1)).toInt();  
        int after = now + newCount.toInt();  
        setData(index(curRow,1), QString::number(after));  
    }  
}
```

```
QModelIndex MaterialsSkladModel::checkObjModel(const QString &material)
```

```
{  
    for(int i = 0; i < rowCount(); ++i)
```

```

{
    if(data(index(i,0)).toString() == material)
    {
        return index(i,0);
        break;
    }
}
return QModelIndex();
}

```

materialproductionmodel.cpp

```
#include "materialsproductmodel.h"
```

```
MaterialsProductModel::MaterialsProductModel(QObject *parent) : QStandardItemModel(parent)
```

```

{
    insertColumns(0,2);
    setHeaderData(0, Qt::Horizontal, "Название материала");
    setHeaderData(1, Qt::Horizontal, "Количество");
}

```

```
void MaterialsProductModel::addObjInProducsMaterialModel(QString material, QString count)
```

```

{
    QModelIndex curIndx = ckeckObjModel(material);
    if(!curIndx.isValid())
    {
        insertRow(rowCount());
        setData(index(rowCount()-1,0), material);
        setData(index(rowCount()-1,1), count);
    }
    else
    {
        int now = data(index(curIndx.row(), 1)).toInt();
        int current = now + count.toInt();
        setData(index(curIndx.row(),1),current);
    }
}

```

```
}  
}
```

```
QModelIndex MaterialsProductModel::ckeckObjModel(const QString &material)
```

```
{  
    for(int i=0; i<rowCount(); ++i)  
    {  
        if(data(index(i,0)).toString() == material)  
        {  
            return index(i,0);  
            break;  
        }  
    }  
    return QModelIndex();  
}
```

Файл main.cpp

```
#include <QGuiApplication>  
#include <QQtApplicationEngine>  
#include <mainwindow.h>  
#include <QApplication>  
#include <QGuiApplication>  
  
int main(int argc, char *argv[])  
{  
    QCoreApplication::setAttribute(Qt::AA_EnableHighDpiScaling);  
  
    QApplication app(argc, argv);  
  
    MainWindow *mw = new MainWindow;  
  
    return app.exec();  
}
```

Файл camerahandler.h

```
#ifndef CAMERAHANDLER_H
#define CAMERAHANDLER_H

#include <QObject>
#include <opencv2/dnn/dnn.hpp>
#include <opencv2/imgproc/imgproc.hpp>
#include <opencv2/imgproc.hpp>
#include <opencv2/objdetect/face.hpp>
#include <opencv2/objdetect/objdetect.hpp>
#include <opencv2/imgcodecs/imgcodecs.hpp>
#include <opencv2/highgui.hpp>
#include <opencv2/imgproc.hpp>
#include <opencv2/features2d.hpp>
#include <QImage>
#include <QPixmap>

class CameraHandler : public QObject
{
    Q_OBJECT
public:
    explicit CameraHandler(QObject *parent = nullptr);
    ~CameraHandler();
    cv::Mat captureFrame();
    cv::Mat getFrame();
    void releaseCap();
    bool isDone;

public slots:
    void started();
    void finished();
};
```

```
private:
    cv::VideoCapture cap;
    cv::Mat frame;

signals:
    void sendPixmap(QPixmap);
};

#endif // CAMERAHANDLER_H
```

Файл camerahandler.cpp

```
#include "camerahandler.h"
```

```
CameraHandler::CameraHandler(QObject *parent) : QObject(parent)
```

```
{
```

```
}
```

```
CameraHandler::~CameraHandler()
```

```
{
```

```
}
```

```
void CameraHandler::releaseCap()
```

```
{
```

```
    isDone = true;
```

```
    cap.release();
```

```
}
```

```
void CameraHandler::started()
```

```
{
```

```
    isDone = false;
```

```
    cap.open(0, cv::CAP_DSHOW);
```

```

cap.set(cv::CAP_PROP_FRAME_WIDTH, 1280);
cap.set(cv::CAP_PROP_FRAME_HEIGHT, 720);
while (true)
{
    if (isDone) return;
    cap >> frame;
    cv::imwrite("1.jpg", frame);
    cv::cvtColor(frame, frame, cv::COLOR_BGR2RGB);
    cv::line(frame, cv::Point(1280/2-1,0), cv::Point(1280/2-1,719), cv::Scalar(0,255,0),4);
    cv::resize(frame, frame, cv::Size(1280/2,720/2));
    QImage img= QImage((uchar*) frame.data, frame.cols, frame.rows, frame.step,
QImage::Format_RGB888);
    emit sendPixmap(QPixmap::fromImage(img));
}
}

void CameraHandler::finished()
{
    cap.release();
}

```

Файл mainwindow.h

```

#ifndef MAINWINDOW_H
#define MAINWINDOW_H

#include <QMainWindow>
#include <QDebug>
#include <QLabel>
#include <QPushButton>
#include <QApplication>
#include <QTimer>
#include <QObject>
#include <QVBoxLayout>

```

```

#include <QQmlEngine>
#include <QQmlComponent>
#include <QHBoxLayout>
#include <QPainter>
#include <QKeyEvent>
#include <QThread>
#include <QProcess>
#include <QDateTime>
#include "camerahandler.h"
#include "messagewidget.h"

class MainWindow : public QWidget
{
    Q_OBJECT
public:
    explicit MainWindow(QWidget *parent = nullptr);
    ~MainWindow();

private:
    QThread* thread;
    QProcess *proc;
    QLabel *label;
    bool isStarted = false;
    CameraHandler *handler;
    cv::Mat frame;
    int currentSegment = -3;
    int currentX = -3;
    int currentY = -3;
    int currentTime = -3;
    double gazeVelocity = -3;
    MessageWidget *msg;

public slots:
    void setCalibrationImage(QPixmap);

```

```
void calibrationDone();
void predictGaze();
void forceClose();
```

protected:

```
void paintEvent(QPaintEvent* event);
void keyPressEvent(QKeyEvent* event);
};
```

```
#endif // MAINWINDOW_H
```

Файл mainwindow.cpp

```
#include "mainwindow.h"
```

```
MainWindow::MainWindow(QWidget *parent) : QWidget(parent)
```

```
{
    thread = new QThread;
    handler = new CameraHandler;
    handler->moveToThread(thread);
    connect(thread, SIGNAL(started()), handler, SLOT(started()));
    connect(thread, SIGNAL(finished()), handler, SLOT(finished()));
    connect(handler, &CameraHandler::sendPixmap, this, &MainWindow::setCalibrationImage);
    setWindowTitle("Калибровка положения");
    setFixedSize(1280/2,410);
    label = new QLabel(this);
    label->setAlignment(Qt::AlignCenter);
    QVBoxLayout *layout = new QVBoxLayout;
    QPushButton *button = new QPushButton(this);
    button->setText("OK");
    button->setFixedSize(100,30);
    QLabel *txt = new QLabel("Калибровка веб-камеры. Лицо должно располагаться симметрично линии.");
    txt->setAlignment(Qt::AlignCenter);
```

```

    QHBoxLayout *hl = new QHBoxLayout;
    hl->addStretch();
    hl->addWidget(button);
    hl->addStretch();
    layout->addWidget(txt);
    layout->addWidget(label);
    layout->addLayout(hl);
    setLayout(layout);
    thread->start();
    show();
    connect(button, &QPushButton::clicked, this, &MainWindow::calibrationDone);
}

void MainWindow::setCalibrationImage(QPixmap pixmap)
{
    label->setPixmap(pixmap);
}

void MainWindow::predictGaze()
{
    if (msg->isVisible()) msg->close();
    QString reply = QString(proc->readAll());
    qDebug() << reply;
    QString segment = reply.split(",").at(0);
    QString x = reply.split(",").at(1);
    QString y = reply.split(",").at(2).split(" ").at(0);
    qDebug() << "SXY" << segment << x << y;
    if (segment.at(0) != "(") return;
    if (currentSegment > 0)
        gazeVelocity = sqrt(pow(round(x.toDouble()*1600)-currentX,2) +
pow(round(y.toDouble()*900)-currentY,2))*0.26/10;
    else
        gazeVelocity = 0;
    currentSegment = segment.remove(0,1).toInt();
}

```

```

currentX = round(x.toDouble()*1600);
currentY = round(y.toDouble()*900);//y.remove("").toInt();
update();
}

void MainWindow::calibrationDone()
{
    msg = new MessageWidget;
    msg->setFixedSize(300,50);
    msg->setWindowFlags(Qt::CustomizeWindowHint);
    connect(msg, &MessageWidget::timeToClose, this, &MainWindow::forceClose);
    disconnect(handler, &CameraHandler::sendPixmap, this, &MainWindow::setCalibrationImage);
    qDebug() << "NOOOOOOOOOOOOOOOOOOOOOOO";
    isStarted = true;
    qDeleteAll(findChildren<QWidget *>(QString(), Qt::FindDirectChildrenOnly));
    proc = new QProcess;
    proc->start("python predict.py");
    proc->waitForStarted();
    connect(proc, &QProcess::readyReadStandardOutput, this, &MainWindow::predictGaze);
    setWindowState(Qt::WindowFullScreen);
    showFullScreen();
    msg->move(width()/2-msg->width()/2, height()/2);
    msg->show();
    handler->releaseCap();
    thread->quit();
    thread->wait();
}

MainWindow::~MainWindow()
{
    thread->quit();
    thread->wait();
    if (proc)
    {

```

```

        QProcess::execute("taskkill /IM python.exe /F");
        qDebug() << "kill";
        proc->kill();
        delete proc;
    }
    if (handler) handler->~CameraHandler();
}

void MainWindow::paintEvent(QPaintEvent *event)
{
    if (isStarted)
    {
        QPainter painter(this);
        QPen pen;
        pen.setColor("black");
        pen.setWidth(3);
        painter.setPen(pen);
        QPolygon triangle;
        triangle << QPoint(width()/2-100,height()/2+100) << QPoint(width()/2+100,height()/2+100)
<< QPoint(width()/2,height()/2-100);
        QPainterPath path;
        path.addPolygon(triangle);
        painter.fillPath(path, Qt::blue);
        painter.drawPolygon(triangle);
        QRect square1;
        square1.setCoords(triangle.at(0).x()-300,          height()/2-100,          triangle.at(0).x()-100,
height()/2+100);
        painter.fillRect(square1, Qt::green);
        painter.drawRect(square1);
        QRect square2;
        square2.setCoords(triangle.at(1).x()+100,          height()/2-100,          triangle.at(1).x()+300,
height()/2+100);
        painter.fillRect(square2, Qt::red);
        painter.drawRect(square2);
    }
}

```

```

if (currentSegment >= 0)
{
    int x = currentSegment%4;
    int y = currentSegment/4;
    QRect rect;
    rect.setCoords(x*400, y*300, x*400+400, y*300+300);
    painter.setPen(QPen(QBrush(),0));
    //painter.setBrush(QBrush(QColor(128, 128, 255, 128)));
    //painter.drawEllipse(QPoint(currentX, currentY), 146, 146);
    painter.fillRect(rect, QBrush(QColor(128, 128, 255, 128)));
    painter.drawRect(rect);
    if (gazeVelocity > 1.7)
    {
        QPolygon activity;
        activity << QPoint(0,0) << QPoint(0,height()) << QPoint(width(),height()) << QPoint
(width(),0);
        pen.setColor("green");
        pen.setWidth(40);
        painter.setPen(pen);
        painter.setBrush(QBrush());
        painter.drawPolygon(activity);
    }
}
else if (currentSegment == -1 || currentSegment == -2)
{
    QPolygon screen_frame;
    screen_frame << QPoint(0,0) << QPoint(0,height()) << QPoint(width(),height()) << QPoint
(width(),0);
    pen.setColor("red");
    pen.setWidth(40);
    painter.setPen(pen);
    painter.drawPolygon(screen_frame);
}
}

```

```
}
```

```
void MainWindow::keyPressEvent(QKeyEvent* event)
{
    if (isStarted && event->key() == Qt::Key::Key_Escape)
    {
        if (proc)
        {
            qDebug() << "kill";
            proc->kill();
            proc->startDetached("taskkill /IM python.exe /F");
            proc->waitForFinished();
            proc->kill();
            delete proc;
        }
        if (handler) handler->~CameraHandler();
        close();
    }
}
```

```
void MainWindow::forceClose()
{
    msg->close();
    if (proc)
    {
        qDebug() << "kill";
        proc->kill();
        proc->startDetached("taskkill /IM python.exe /F");
        proc->waitForFinished();
        proc->kill();
        delete proc;
    }
    if (handler) handler->~CameraHandler();
    close();
}
```

```
}
```

Файл messagewidget.h

```
#ifndef MESSAGEWIDGET_H  
#define MESSAGEWIDGET_H
```

```
#include <QWidget>  
#include <QLabel>  
#include <QHBoxLayout>  
#include <QKeyEvent>
```

```
class MessageWidget : public QWidget  
{  
    Q_OBJECT  
public:  
    explicit MessageWidget(QWidget *parent = nullptr);  
  
protected:  
    void keyPressEvent(QKeyEvent *event);  
  
signals:  
    void timeToClose();  
};  
  
#endif // MESSAGEWIDGET_H
```

Файл messagewidget.cpp

```
#include "messagewidget.h"
```

```
MessageWidget::MessageWidget(QWidget *parent) : QWidget(parent)  
{  
    QLabel *label = new QLabel;
```

```

QHBoxLayout *hl = new QHBoxLayout;
label->setText("Подгрузка модулей. Ожидайте...");
hl->addStretch();
hl->addWidget(label);
hl->addStretch();
setLayout(hl);
}

void MessageWidget::keyPressEvent(QKeyEvent *event)
{
    if (isVisible() && event->key() == Qt::Key::Key_Escape)
        emit timeToClose();
}

```

Файл predict.py

```

import torch
import numpy as np
import argparse
import cv2
import mediapipe
import torchvision

```

```

class HybridNet(torch.nn.Module):
    def __init__(self):
        super().__init__()
        self.first_eye = torch.nn.Sequential(
            torch.nn.Conv2d(in_channels=3, out_channels=96, kernel_size=11),
            torch.nn.BatchNorm2d(96),
            torch.nn.ReLU(),
            torch.nn.Conv2d(in_channels=96, out_channels=256, kernel_size=5),
            torch.nn.BatchNorm2d(256),
            torch.nn.ReLU(),
            torch.nn.Conv2d(in_channels=256, out_channels=384, kernel_size=3),

```

```

torch.nn.BatchNorm2d(384),
torch.nn.ReLU(),
torch.nn.Conv2d(in_channels=384, out_channels=64, kernel_size=1),
torch.nn.BatchNorm2d(64),
torch.nn.ReLU())
self.first_eye_fc = torch.nn.Sequential(
    torch.nn.Linear(in_features=64*9*9, out_features=128),
    torch.nn.BatchNorm1d(128),
    torch.nn.ReLU())
self.second_eye = torch.nn.Sequential(
    torch.nn.Conv2d(in_channels=3, out_channels=96, kernel_size=11),
    torch.nn.BatchNorm2d(96),
    torch.nn.ReLU(),
    torch.nn.Conv2d(in_channels=96, out_channels=256, kernel_size=5),
    torch.nn.BatchNorm2d(256),
    torch.nn.ReLU(),
    torch.nn.Conv2d(in_channels=256, out_channels=384, kernel_size=3),
    torch.nn.BatchNorm2d(384),
    torch.nn.ReLU(),
    torch.nn.Conv2d(in_channels=384, out_channels=64, kernel_size=1),
    torch.nn.BatchNorm2d(64),
    torch.nn.ReLU())
self.second_eye_fc = torch.nn.Sequential(
    torch.nn.Linear(in_features=64*9*9, out_features=128),
    torch.nn.BatchNorm1d(128),
    torch.nn.ReLU())
self.coords = torch.nn.Sequential(
    torch.nn.Linear(in_features=4*2, out_features=128),
    torch.nn.BatchNorm1d(128),
    torch.nn.ReLU(),
    torch.nn.Linear(in_features=128, out_features=128),
    torch.nn.BatchNorm1d(128),
    torch.nn.ReLU())
self.union = torch.nn.Sequential(

```

```

torch.nn.Linear(in_features=128*3, out_features=128),
torch.nn.BatchNorm1d(128),
torch.nn.ReLU(),
torch.nn.Linear(in_features=128, out_features=2),
torch.nn.ReLU())

```

```

def forward(self, imgs, coords):

```

```

    x_1 = self.first_eye(imgs[0].unsqueeze(0))
    x_2 = self.second_eye(imgs[1].unsqueeze(0))
    x_1 = self.first_eye_fc(x_1.reshape(x_1.shape[0], -1))
    x_2 = self.second_eye_fc(x_2.reshape(x_2.shape[0], -1))
    if coords.size(0) > 1:
        coords = torch.cat(tuple(coords), dim=-1)
    x_3 = self.coords(coords)
    return self.union(torch.cat((x_1, x_2, x_3), dim=-1))

```

```

class BaselineNet(torch.nn.Module):

```

```

    def __init__(self):
        super().__init__()
        self.first_eye = torch.nn.Sequential(
            torch.nn.Conv2d(in_channels=3, out_channels=32, kernel_size=7, stride=2),
            torch.nn.BatchNorm2d(32),
            torch.nn.ReLU(),
            torch.nn.AvgPool2d(kernel_size=2),
            torch.nn.Conv2d(in_channels=32, out_channels=64, kernel_size=5, stride=2),
            torch.nn.ReLU(),
            torch.nn.BatchNorm2d(64),
            torch.nn.AvgPool2d(kernel_size=2),
            torch.nn.Conv2d(in_channels=64, out_channels=128, kernel_size=3, stride=1),
            torch.nn.BatchNorm2d(128),
            torch.nn.ReLU())
        self.second_eye = torch.nn.Sequential(
            torch.nn.Conv2d(in_channels=3, out_channels=32, kernel_size=7, stride=2),

```

```

torch.nn.BatchNorm2d(32),
torch.nn.ReLU(),
torch.nn.AvgPool2d(kernel_size=2),
torch.nn.Conv2d(in_channels=32, out_channels=64, kernel_size=5, stride=2),
torch.nn.ReLU(),
torch.nn.BatchNorm2d(64),
torch.nn.AvgPool2d(kernel_size=2),
torch.nn.Conv2d(in_channels=64, out_channels=128, kernel_size=3, stride=1),
torch.nn.BatchNorm2d(128),
torch.nn.ReLU())
self.coords = torch.nn.Sequential(
    torch.nn.Linear(in_features=4*2, out_features=128),
    torch.nn.BatchNorm1d(128),
    torch.nn.ReLU(),
    torch.nn.Linear(in_features=128, out_features=16),
    torch.nn.BatchNorm1d(16),
    torch.nn.ReLU(),
    torch.nn.Linear(in_features=16, out_features=16),
    torch.nn.BatchNorm1d(16),
    torch.nn.ReLU())
self.union = torch.nn.Sequential(
    torch.nn.Linear(in_features=16+4096, out_features=8),
    torch.nn.BatchNorm1d(8),
    torch.nn.ReLU(),
    torch.nn.Linear(in_features=8, out_features=4),
    torch.nn.BatchNorm1d(4),
    torch.nn.ReLU(),
    torch.nn.Linear(in_features=4, out_features=2),
    torch.nn.ReLU())

def forward(self, imgs, coords):
    x_1 = self.first_eye(imgs[0].unsqueeze(0))
    x_2 = self.second_eye(imgs[1].unsqueeze(0))
    x_1 = x_1.reshape(x_1.shape[0], -1)

```

```

x_2 = x_2.reshape(x_2.shape[0], -1)
if coords.size(0) > 1:
    coords = torch.cat(tuple(coords),dim=-1)
x_3 = self.coords(coords)
return self.union(torch.cat((x_1, x_2, x_3), dim=-1)).squeeze()

```

```

class OneEyeNetAsRegression(torch.nn.Module):

```

```

    def __init__(self):

```

```

        super().__init__()

```

```

        self.first_eye = torch.nn.Sequential(

```

```

            torch.nn.Conv2d(in_channels=3, out_channels=32, kernel_size=7, stride=2),

```

```

            torch.nn.BatchNorm2d(32),

```

```

            torch.nn.ReLU(),

```

```

            torch.nn.AvgPool2d(kernel_size=2),

```

```

            torch.nn.Conv2d(in_channels=32, out_channels=64, kernel_size=5, stride=2),

```

```

            torch.nn.ReLU(),

```

```

            torch.nn.BatchNorm2d(64),

```

```

            torch.nn.AvgPool2d(kernel_size=2),

```

```

            torch.nn.Conv2d(in_channels=64, out_channels=128, kernel_size=3, stride=1),

```

```

            torch.nn.BatchNorm2d(128),

```

```

            torch.nn.ReLU())

```

```

        self.coords = torch.nn.Sequential(

```

```

            torch.nn.Linear(in_features=6, out_features=128),

```

```

            torch.nn.BatchNorm1d(128),

```

```

            torch.nn.ReLU(),

```

```

            torch.nn.Linear(in_features=128, out_features=16),

```

```

            torch.nn.BatchNorm1d(16),

```

```

            torch.nn.ReLU(),

```

```

            torch.nn.Linear(in_features=16, out_features=16),

```

```

            torch.nn.BatchNorm1d(16),

```

```

            torch.nn.ReLU())

```

```

        self.union = torch.nn.Sequential(

```

```

            torch.nn.Linear(in_features=16+2048, out_features=8),

```

```

torch.nn.BatchNorm1d(8),
torch.nn.ReLU(),
torch.nn.Linear(in_features=8, out_features=4),
torch.nn.BatchNorm1d(4),
torch.nn.ReLU(),
torch.nn.Linear(in_features=4, out_features=2),
torch.nn.ReLU()

```

```
def forward(self, imgs, coords):
```

```

    x_1 = self.first_eye(imgs[1].unsqueeze(0))
    x_1 = x_1.reshape(x_1.shape[0], -1)
    if coords.size(0) > 1:
        coords = torch.cat(tuple([coords[3], coords[5], coords[7]]), dim=-1)
    x_2 = self.coords(coords)
    return self.union(torch.cat((x_1, x_2), dim=-1))

```

```
RIGHT_IRIS = [474, 475, 476, 477]
```

```
LEFT_IRIS = [469, 470, 471, 472]
```

```
R_H_LEFT = 33
```

```
R_H_RIGHT = 133
```

```
R_H_PUPIL = 468
```

```
L_H_LEFT = 362
```

```
L_H_RIGHT = 263
```

```
L_H_PUPIL = 473
```

```
LEFT_EYE_3_POINTS = [473, 263, 362]
```

```
RIGHT_EYE_3_POINTS = [468, 133, 33]
```

```
LEFT_EYE_6_POINTS = [385, 380, 387, 373, 362, 263]
```

```
RIGHT_EYE_6_POINTS = [160, 144, 158, 153, 33, 133]
```

```

transform = torchvision.transforms.Compose([
    torchvision.transforms.ColorJitter(),

```

```

        torchvision.transforms.ToTensor(),
        torchvision.transforms.Normalize([0.485, 0.456, 0.406], [0.229, 0.224, 0.225])
    ])

```

```

def distance(point1, point2):
    return ((point1[0] - point2[0])**2 + (point1[1] - point2[1])**2)**0.5

```

```

def get_ear(mesh, idxs):
    return (distance(mesh[idxs[0]], mesh[idxs[1]]) + distance(mesh[idxs[2]], mesh[idxs[3]]))/ 2 /
distance(mesh[idxs[4]], mesh[idxs[5]])

```

```

def get_is_darker(gray_frame, mesh, idxs):
    pupils_darker = []
    pupils_darker.append(gray_frame[mesh[idxs[0]][1],mesh[idxs[0]][0]]<=
gray_frame[mesh[idxs[1]][1], mesh[idxs[1]][0]])
    pupils_darker.append(gray_frame[mesh[idxs[0]][1],mesh[idxs[0]][0]]<=
gray_frame[mesh[idxs[2]][1], mesh[idxs[2]][0]])
    return sum(pupils_darker)

```

```

def setupModels():
    cap = cv2.VideoCapture(0, cv2.CAP_DSHOW)
    cap.set(cv2.CAP_PROP_FRAME_HEIGHT, 720)
    cap.set(cv2.CAP_PROP_FRAME_WIDTH, 1280)
    mp_face_mesh = mediapipe.solutions.face_mesh
    device = torch.device('cuda') if torch.cuda.is_available() else torch.device('cpu')
    model = HybridNet().to(device)
    model.load_state_dict(torch.load('./best_hybrid.pt', map_location=device))
    model.eval()
    return model, mp_face_mesh, cap

```

```

def predictGazeInLoop(model, mesh, cap):
    ret, frame = cap.read()
    mp_face_mesh = mediapipe.solutions.face_mesh
    frame = cv2.cvtColor(frame, cv2.COLOR_BGR2RGB)
    with mesh.FaceMesh(
        max_num_faces=1,
        refine_landmarks=True,
        min_detection_confidence=0.65,
        min_tracking_confidence=0.65) as face_mesh:
        res = face_mesh.process(frame)
        if not res.multi_face_landmarks:
            return -1,-1,-1
        else:
            gray_frame = cv2.cvtColor(frame, cv2.COLOR_RGB2GRAY)
            h, w = frame.shape[:2]
            norm_meshpoints = np.array([np.array([p.x, p.y]) for p in
res.multi_face_landmarks[0].landmark])
            meshpoints = np.array([np.array([p.x*w, p.y*h]).astype(int) for p in res.multi_face_
landmarks[0].landmark])
            is_darker = get_is_darker(gray_frame, meshpoints, LEFT_EYE_3_POINTS) +
get_is_darker(gray_frame, meshpoints, RIGHT_EYE_3_POINTS)
            ear = get_ear(meshpoints, LEFT_EYE_6_POINTS) + get_ear(meshpoints, RIGHT_EYE
_6_POINTS)
            if is_darker < 3 or ear / 2 < 0.2:
                return -2,-2,-2
            else:
                left_eye = frame[int(meshpoints[L_H_PUPIL][1]-abs(meshpoints[L_H_LEFT][0]-
meshpoints[L_H_RIGHT][0])/2):
int(meshpoints[L_H_PUPIL][1]+abs(meshpoints[L_H_LEFT][0]-
meshpoints[L_H_RIGHT][0])/2),
                    meshpoints[L_H_LEFT][0]:meshpoints[L_H_RIGHT][0]]
                right_eye = frame[int(meshpoints[R_H_PUPIL][1]-abs(meshpoints[R_H_RIGHT][0]-
meshpoints[R_H_LEFT][0])/2):

```

```

int(meshpoints[R_H_PUPIL][1]+abs(meshpoints[R_H_LEFT][0]-
meshpoints[R_H_RIGHT][0])/2),
        meshpoints[R_H_LEFT][0]:meshpoints[R_H_RIGHT][0]]
cv2.imwrite('re.jpg',right_eye)
cv2.imwrite('le.jpg', left_eye)
left_eye = transform(cv2.resize(cv2.flip(np.array(cv2.cvtColor(left_eye, cv2.COLOR_
BGR2RGB) / 255, dtype='float32'),1), (25,25), interpolation=cv2.INTER_AREA))
right_eye = transform(cv2.resize(np.array(cv2.cvtColor(right_eye,
cv2.COLOR_BGR2RGB) / 255, dtype='float32'), (25,25), interpolation=cv2.INTER_AREA))
coords =
torch.cat([torch.FloatTensor([norm_meshpoints[L_H_RIGHT][0],norm_meshpoints[L_H_RIGHT
][1]]),

torch.FloatTensor([norm_meshpoints[R_H_LEFT][0],norm_meshpoints[R_H_LEFT][1]]),

torch.FloatTensor([norm_meshpoints[L_H_LEFT][0],norm_meshpoints[L_H_LEFT][1]]),

torch.FloatTensor([norm_meshpoints[R_H_RIGHT][0],norm_meshpoints[R_H_RIGHT][1]]),
dim=-1)
device = torch.device('cuda') if torch.cuda.is_available() else torch.device('cpu')
left_eye = left_eye.to(device)
right_eye = right_eye.to(device)
coords = coords.to(device).unsqueeze(0)
with torch.no_grad():
    outputs = model([left_eye, right_eye], coords).squeeze()
    segment = int(torch.nn.functional.relu(outputs[1])*900)//300*4 + int(torch.nn.functional.
relu(outputs[0])*1600)//400
    return min(11, segment), float(outputs[0]), float(outputs[1])

if __name__ == "__main__":
    model, mesh, cap = setupModels()
    k = 0
    while True:

```

```
print(predictGazeInLoop(model,mesh,cap), k, flush=True)
k+=1
```

Файл test\_model.pro

```
QT += quick \
    widgets
```

```
CONFIG += c++11
```

```
# You can make your code fail to compile if it uses deprecated APIs.
# In order to do so, uncomment the following line.
#DEFINES += QT_DISABLE_DEPRECATED_BEFORE=0x060000
```

```
SOURCES += \
    camerahandler.cpp \
    main.cpp \
    mainwindow.cpp \
    messagewidget.cpp
```

```
RESOURCES += \
    qrc.qrc
```

```
# Additional import path used to resolve QML modules in Qt Creator's code model
QML_IMPORT_PATH =
```

```
# Additional import path used to resolve QML modules just for Qt Quick Designer
QML_DESIGNER_IMPORT_PATH =
```

```
# Default rules for deployment.
```

```
qnx: target.path = /tmp/${TARGET}/bin
else: unix:!android: target.path = /opt/${TARGET}/bin
!isEmpty(target.path): INSTALLS += target
```

```
HEADERS += \  
    camerahandler.h \  
    mainwindow.h \  
    messagewidget.h
```

```
INCLUDEPATH += D:\opencv\opencv\build\include
```

```
LIBS += D:\opencv\opencv-build\bin\libopencv_core454.dll  
LIBS += D:\opencv\opencv-build\bin\libopencv_highgui454.dll  
LIBS += D:\opencv\opencv-build\bin\libopencv_imgcodecs454.dll  
LIBS += D:\opencv\opencv-build\bin\libopencv_imgproc454.dll  
LIBS += D:\opencv\opencv-build\bin\libopencv_features2d454.dll  
LIBS += D:\opencv\opencv-build\bin\libopencv_calib3d454.dll  
LIBS += D:\opencv\opencv-build\bin\libopencv_objdetect454.dll  
LIBS += D:\opencv\opencv-build\bin\libopencv_videoio454.dll  
LIBS += D:\opencv\opencv-build\bin\libopencv_dnn454.dll
```