

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

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

Кафедра «Прикладная математика и информатика»
(наименование)

02.03.03 Математическое обеспечение и администрирование информационных систем
(код и наименование направления подготовки / специальности)

Мобильные и сетевые технологии
(направленность (профиль) / специализация)

ВЫПУСКНАЯ КВАЛИФИКАЦИОННАЯ РАБОТА (БАКАЛАВРСКАЯ РАБОТА)

на тему «Разработка мобильного приложения для мониторинга транспортных средств через систему ГЛОНАСС»

Обучающийся

Н.А.Зеленов

(Инициалы Фамилия)

(личная подпись)

Руководитель

канд.эконом.наук., Н.А.Раченко

(ученая степень (при наличии), ученое звание (при наличии), Инициалы Фамилия)

Тольятти 2023

Содержание

Введение.....	3
1. Анализ работы системы ГЛОНАСС и существующих мобильных приложений.....	5
1.1 Анализ предметной области	5
1.2 Алгоритм работы системы ГЛОНАСС	6
1.3 Анализ существующих решений мобильных приложений.	8
1.4 Техническое задание на разработку мобильного приложения	9
Выводы по первой главе.....	10
2. Проектирование и разработка мобильного приложения	11
2.1 Выбор среды реализации.....	11
2.2 Разработка интерфейса мобильного приложения.....	12
2.3 Авторизация.....	20
2.4 Учет данных.....	21
2.5 Описание кода мобильного приложения	22
3. Тестирование мобильного приложения.....	28
3.1.Метод тестирования.....	28
3.2.Результаты тестирования	31
Заключение	35
Список используемой литературы	36

Введение

Актуальность исследования связана с тем, что все больше людей предпочитают получать быстрый доступ к информации с помощью мобильных устройств, а доля пользователей мобильных версий сайтов растет. Кроме того, популярность мобильных устройств приводит к затруднению отслеживания всех потоков информации на всех устройствах. В такой ситуации разработка мобильного приложения для мониторинга транспортных средств, которое облегчит процесс мониторинга и не потребует лишнего времени и усилий, является весьма актуальной.

Многие программные продукты цифровой эпохи были разработаны для мобильных телефонов. Многие новейшие электронные устройства не уступают по функциональности и мощности громоздким мобильным телефонам и персональным компьютерам[5].

Сегодня в мире практически нет человека, который бы не пользовался мобильным телефоном.

Причинами их популярности являются функциональность, доступ ко всем формам связи и, конечно же, небольшие размеры устройств. Мобильные устройства практически заменили настольные компьютеры в домах. Однако это имеет свои недостатки: совершенствование мобильных технологий затрудняет отслеживание всех потоков информации на всех устройствах.[13]

Благодаря тому, что большое количество работников пользуются телефонами на операционной системе Android, это способствует делать разрабатываемое приложение более доступным.[32]

В настоящее время использование веб-сайта с мобильного устройства может вызвать ряд неудобств. Вам приходится запоминать адрес сайта или использовать поисковую систему, чтобы найти его. Затем приходится искать нужный раздел сайта среди большого количества текста и огромного числа ссылок.[16]

В связи с этим было решено разработать мобильное приложение “Автоград-Люкс”, которое будет давать возможность производить мониторинг транспортных средств.

Практической значимостью является то, что написанное приложение облегчит мониторинг транспорта не тратя на это лишнее количество времени.

Сотрудникам компаний оно позволит в любое время отслеживать работу своих подчиненных и видеть все данные по машине.

Объект исследования: процесс отслеживания данных с транспортных средств в режиме реального времени.

Предмет исследования: мониторинг транспортного средства через систему ГЛОНАСС с использованием мобильного приложения "Автоград-Люкс".

Целью выпускной квалификационной работы является разработать мобильное приложение для учета у системы ГЛОНАСС на транспортное средство.

Выпускная квалификационная работа состоит из 3 основных глав:

1. Анализ работы системы ГЛОНАСС и существующие мобильные приложения.
2. Проектирование и разработка мобильного приложения.
3. Тестирование мобильного приложения.

1 Анализ работы системы ГЛОНАСС и существующих мобильных приложений

1.1 Анализ предметной области

ООО «Автоград-люкс» - официальный партнер и агент АО «ГЛОНАСС».

ГЛОНАСС (Глобальная навигационно-спутниковая система) - это глобальная система спутниковой радионавигации, созданная для обеспечения навигационной информацией на территории России, СНГ и других стран.

С 2010 года компания «Автоград-Люкс» занимается продажей, установкой и обслуживанием систем мониторинга ГЛОНАСС, устройств тахографического контроля и видеонаблюдения.

Автоград-Люкс на рынке работают уже более 10 лет. За это время они оборудовали более 10 тысяч машин система ГЛОНАСС/GPS. На данный момент они ведут мониторинг более 3 тысячам транспортным средствам в Российской Федерации.

Оснащение транспорта системами ГЛОНАСС/GPS стало набирать популярность после выхода Приказа Министерства транспорта Российской Федерации. От 26 января 2012 г. "Об утверждении Порядка оснащения транспортных средств, находящихся в эксплуатации, включая специальные транспортные средства, категории М, используемых для коммерческих перевозок пассажиров, и категории N, используемых для перевозки опасных грузов, аппаратурой спутниковой навигации ГЛОНАСС или ГЛОНАСС/GPS".

Являются официальными дилерами ведущих компаний-производителей оборудования. Среди клиентов – транспортные, промышленно-металлургические, машиностроительные и др. компании.

Предоставляют услуги по идентификации аппаратуры спутниковой навигации на транспортных средствах категорий М2, М3 и транспортных средств категории N, используемых для перевозки опасных грузов.

Одни из самых крупных партнеров компании являются : Газпром, РОСНЕФТЬ, АКРОН, ЭКОВОЗ, Самарская скорая помощь, Администрация города Сызрань.

1.2 Алгоритм работы системы ГЛОНАСС

ГЛОНАСС - это глобальная спутниковая навигационная система, аналогичная GPS. Она предназначена для определения местоположения объектов на поверхности Земли с высокой точностью, используя сигналы, передаваемые спутниками. Система ГЛОНАСС имеет ряд преимуществ перед другими системами навигации, так как она обеспечивает более точное определение координат в условиях густого городского застройки, многоканальность, а также доступность в любой точке земного шара.

Установка системы ГЛОНАСС необходима во многих сферах, таких как автомобильный транспорт, морская и воздушная навигация, геодезия и другие. Например, в автомобильной отрасли, система ГЛОНАСС используется для контроля скорости и маршрута движения транспортных средств, а также для предотвращения краж и угонов. В морской и воздушной навигации система ГЛОНАСС позволяет определять координаты судов и самолетов, что повышает безопасность плавания и полетов.

Таким образом, установка системы ГЛОНАСС позволяет повысить эффективность работы в различных отраслях, а также обеспечивает более точное и надежное определение координат объектов на поверхности Земли.

Система ГЛОНАСС состоит из двенадцати активных космических аппаратов, расставленных по орбите вокруг Земли, и земной инфраструктуры в виде станций управления, передающих сигналы на спутники и обрабатывающих полученные данные. Сигналы от спутников принимаются специальными приемниками, которые могут быть установлены на различных объектах.

1.Сбор сигналов от спутников

Система ГЛОНАСС использует сигналы, которые передаются с помощью спутников, расположенных на космической орбите. Каждый спутник отправляет свой уникальный сигнал, который может быть собран приемником на земле. Чем больше сигналов собрано, тем более точным становится определение местоположения.

2.Обработка сигналов

После сбора сигналов они подвергаются обработке на специальном сервере. Программное обеспечение системы ГЛОНАСС анализирует различные параметры каждого сигнала, такие как время отправки и приема, частота и др. Эти параметры помогают определить точные координаты местоположения.

3. Передача данных на приемник GPS

После обработки сигналов данные о местоположении передаются на приемник GPS. Это устройство, которое может использоваться в автомобилях, самолетах или других транспортных средствах. Приемник GPS принимает данные от системы ГЛОНАСС и определяет точное местоположение.

4.Вывод информации на дисплей

Информация о местоположении может быть выведена на дисплей приемника GPS. Это могут быть карты с подробными данными о маршруте, а также информация о скорости и времени прибытия в пункт назначения. Пользователь может использовать эту информацию для навигации и планирования маршрутов.

5.Обновление данных

Система ГЛОНАСС постоянно обновляет данные о местоположении спутников на орбите Земли. Это позволяет улучшать точность определения местоположения и обеспечивать более надежную навигацию. Кроме того, система может автоматически корректировать координаты приемника GPS при изменении положения спутников.

6. Ошибки и проблемы

Несмотря на высокую точность системы ГЛОНАСС, могут возникать ошибки и проблемы при определении местоположения. Например, плохая погода или многоэтажные здания могут повлиять на качество сигнала. В таких случаях можно использовать дополнительное оборудование, например, усилители сигнала или антенны с более высокой чувствительностью. Кроме того, система ГЛОНАСС может использовать данные других навигационных систем, таких как GPS или Galileo, для повышения точности определения местоположения.

Применение ГЛОНАСС используется в разных сферах жизни, таких как: геодезия, картография, мониторинг, авиация, космос, водный транспорт и так далее.

1.3 Анализ существующих решений мобильных приложений.

Существует небольшое количество продуктов, которые уже ведут свою деятельность в данном направлении. Поэтому принято решение проанализировать 3 приложения.

В ходе сравнения, были выявлены одни из самых частых и наглядных недостатков:

1. Отсутствие карты навигации
2. Отсутствием информации о заправках
3. Отсутствие способа объединения транспортных средств по папкам Автоскан+

Данная организация занимается системой мониторинг и имеет мобильное приложение. В нем были замечены такие недостатки как: отсутствует карта для отслеживания транспортного средства и отсутствует информация о заправках.

КГК Мониторинг

Аналогичная компания с мобильным приложением. В ходе анализа, было зафиксировано отсутствие информации о заправках, общем расходе топлива и расходе топлива на 100км.

SKYriver

В последнем проанализированном приложении так же был один недостаток, а именно: недоступность объединения транспортных средств по папкам. Эта функция очень полезна для поиска автомобиля и распределения машин по каким-либо группам.

1.4 Техническое задание на разработку мобильного приложения

Было принято решение о создании мобильного приложения на платформе Android для повышения имиджа, расширения рынка сбыта, для рекламных и маркетинговых целей, также для простоты доступа к информации[1]-[3].

Разработано техническое задание для написания мобильного приложения, состоящее из следующих пунктов:

1. Задачи мобильного приложения. Использовать удобный и простой способ доступа к информации, включать в себя информацию о мониторинге транспортных средств (в рабочем/ нерабочем состоянии, координаты, уровень топливного бака и т.д).

2. Язык реализации. Приложение будет реализовано на русском языке.

3. Экран. При использовании мобильного телефона, планшета ориентация экрана будет портретной.

4. Операционная система. Приложение должно работать на операционной системе Android.

5. Язык программирования. При написании мобильного приложения, используется язык программирования: Java.

6. Дизайн. Эскизы, цвет и стиль оформления должны соответствовать корпоративному стилю.

Выводы по первой главе

Была выявлена актуальность разработки мобильного приложения, сформулирован объект, предмет и цель выпускной квалификационной работы.

Так же был разобран алгоритм работы системы ГЛОНАСС и выведены основные пункты его действий.

В этой же главе были проанализированы существующие мобильные разработки для мониторинга систем ГЛОНАСС. Показаны определенные недостатки, которые будут учтены при разработке собственного мобильного приложения.

Было изучено техническое задание для разработки мобильного приложения.

2 Проектирование и разработка мобильного приложения

2.1 Выбор среды реализации

При выборе среды реализации для создания мобильного приложения необходимо учитывать множество факторов, таких как целевая платформа, требования к приложению, наличие опыта работы с конкретными инструментами, а также доступные бюджет и сроки разработки[26]-[29].

Существует множество сред разработки мобильных приложений, которые предоставляют широкий спектр инструментов и возможностей для создания приложений для разных платформ. Например, для создания мобильных приложений под iOS можно использовать среду Xcode, а для Android - Android Studio. Также существуют кроссплатформенные среды, такие как Xamarin, React Native, Flutter, которые позволяют создавать приложения для нескольких платформ с помощью общего кода.[20]

Важно также учитывать наличие готовых библиотек и фреймворков в выбранной среде разработки, которые могут значительно ускорить процесс разработки и улучшить качество приложения.

Наконец, важно учитывать не только функциональность, но и стоимость выбранной среды разработки, а также доступность технической поддержки и сообщества разработчиков при возникновении проблем.

В итоге, выбор среды разработки мобильного приложения зависит от множества факторов и требует внимательного анализа каждого конкретного случая.

Было проанализировано несколько сред разработок, такие как React Native, jQuery Mobile и Android Studio.

Выбор был сделан в пользу Android Studio, потому что :

1. Интуитивный и удобный интерфейс

Она имеет интуитивный и удобный интерфейс, который становится все более доступным с каждым обновлением. Кроме того, Android Studio

предоставляет шаблоны проектов, которые помогают быстро создавать типичные функции приложений.

2. Высокая скорость разработки

Она позволяет быстро создавать новые проекты и быстро обрабатывать изменения в коде. Это особенно полезно для разработчиков, которые стремятся быстро достигать целей на проекте и быстро получать обратную связь от пользователей[15].

3. Бесплатный и открытый исходный код

Она является бесплатной средой разработки, доступной для загрузки с сайта разработчиков Google. Также Android Studio является открытым исходным кодом, что означает, что вы можете использовать его для своих проектов, изучать код и вносить изменения, если это необходимо[6]-[8].

4. Тестирование приложений

Данная среда предоставляет инструменты для тестирования приложений, что позволяет разработчикам проверять, насколько хорошо работает их приложение на разных устройствах и операционных системах. Это помогает убедиться, что приложение будет работать правильно на всех устройствах и не вызовет ошибок[9]-[10].

2.2 Разработка интерфейса мобильного приложения

Разработка интерфейса играет очень важную роль в создании программного продукта, особенно при разработке мобильных приложений и веб-сайтов. Она влияет на конечное восприятие и впечатление пользователя от продукта. Вот несколько причин, почему важно уделить достаточно времени и внимания разработке интерфейса[20]-[23]:

1. Пользовательская дружелюбность: пользователь должен чувствовать себя комфортно и уверенно при работе с программным продуктом. Хороший интерфейс помогает пользователю легко понимать функционал продукта и использовать его без лишних усилий[11].

2. Визуальный дизайн: правильное оформление и расположение элементов на экране компьютера или на мобильном устройстве помогает пользователю быстро находить нужную информацию. Это повышает удобство использования продукта и уменьшает время на обучение.

3. Брендинг продукта: корпоративный стиль продукта начинается с правильного и узнаваемого дизайна интерфейса. Цвета, шрифты, логотипы и другие элементы могут быть использованы для укрепления бренда продукта[30]-[31].

4. Увеличение продаж: выглядеть привлекательно и организованно может помочь в увеличении продаж, общей оценке продукта и повторности использования.

5. Повышение конкурентоспособности: хороший интерфейс помогает продукту выглядеть лучше по сравнению с конкурирующими продуктами на рынке[33]-[34].

Таким образом, разработка интерфейса является неотъемлемой частью процесса создания программного продукта. Хороший интерфейс может помочь увеличить удовлетворенность пользователя продуктом, повысить продажи и укрепить бренд продукта[9]-[13].

В ходе анализа было решено, что в мобильное приложение будут внедрены в качестве основных разделов следующие:

1. Стартовый экран – вход/ регистрация.
2. Раздел «Объекты мониторинга/ меню» - лента, содержащая папки о машинах клиентов.
3. Раздел «Период просмотра данных», для настройки даты мониторинга авто.
4. Раздел «Информация по ТС», который содержит информацию транспортном средстве.
5. Раздел «Поисковая строка », который быстро помогает найти машину в любой организации.

6. Раздел «Карта», на котором показывается местонахождение автомобиля клиента.

Стартовый экран

На рисунке 1 представлен стартовый экран мобильного приложения.

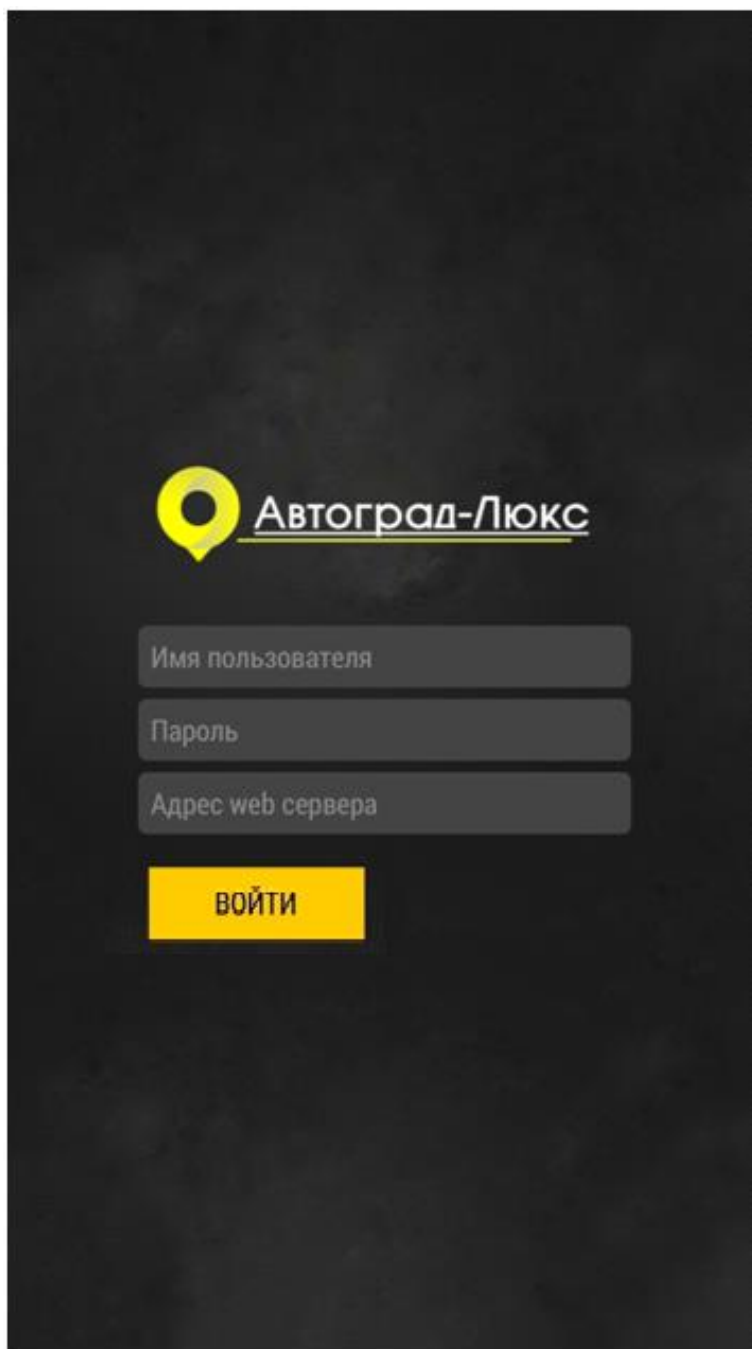


Рисунок 1 – Стартовый экран

На данной странице пользователь совершает вход в личный кабинет своей организации на правах администратора.

Если у клиента нет личного кабинета, ему нужно обратиться в нашу компанию для предоставления доступа в учетную запись.

Объекты мониторинга/ меню

На рисунке 2 показано меню мобильного приложения.

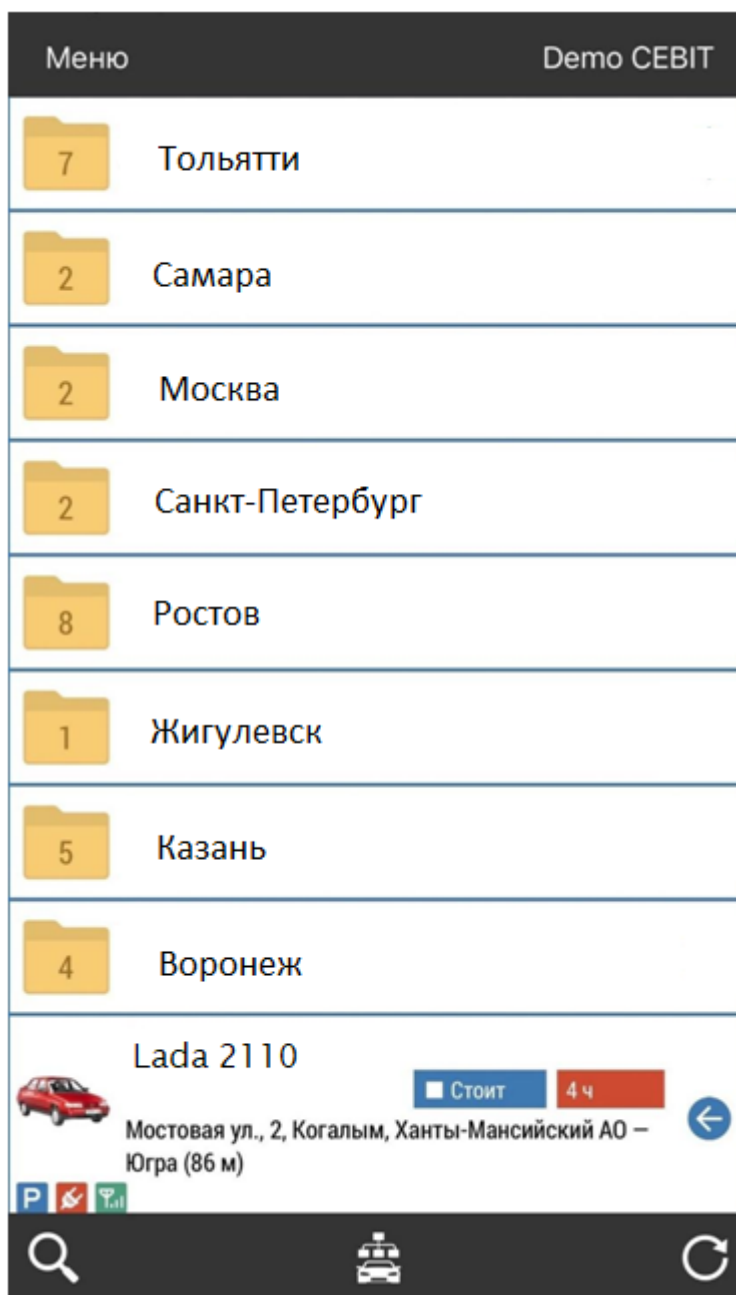


Рисунок 2 – Меню

После входа в систему клиент попадает во вкладку “Меню”. В ней расположены данные о всех транспортных средствах компании, разбитые по папкам. Кроме того, автомобили (по желанию) пользователей могут быть расположены вне папок.

Период просмотра данных

На рисунке 3 представлен период просмотра данных.



Рисунок 3 – Период просмотра данных

Это окно предоставляет возможность фильтрации раздела “Меню” по выбранному времени.

Бегунок в разделе “Период просмотра данных” отвечает за выбранный промежуток времени для просмотра информации по транспортному средству.

Ниже расположены кликабельные кнопки для перехода в другие вкладки (техподдержка, настройки, о программе, выход).

В нижнем левом углу экрана отображается имя пользователя, под которым был выполнен вход в учетную запись.

Информация по ТС

На рисунке 4 изображена информация по транспортному средству.

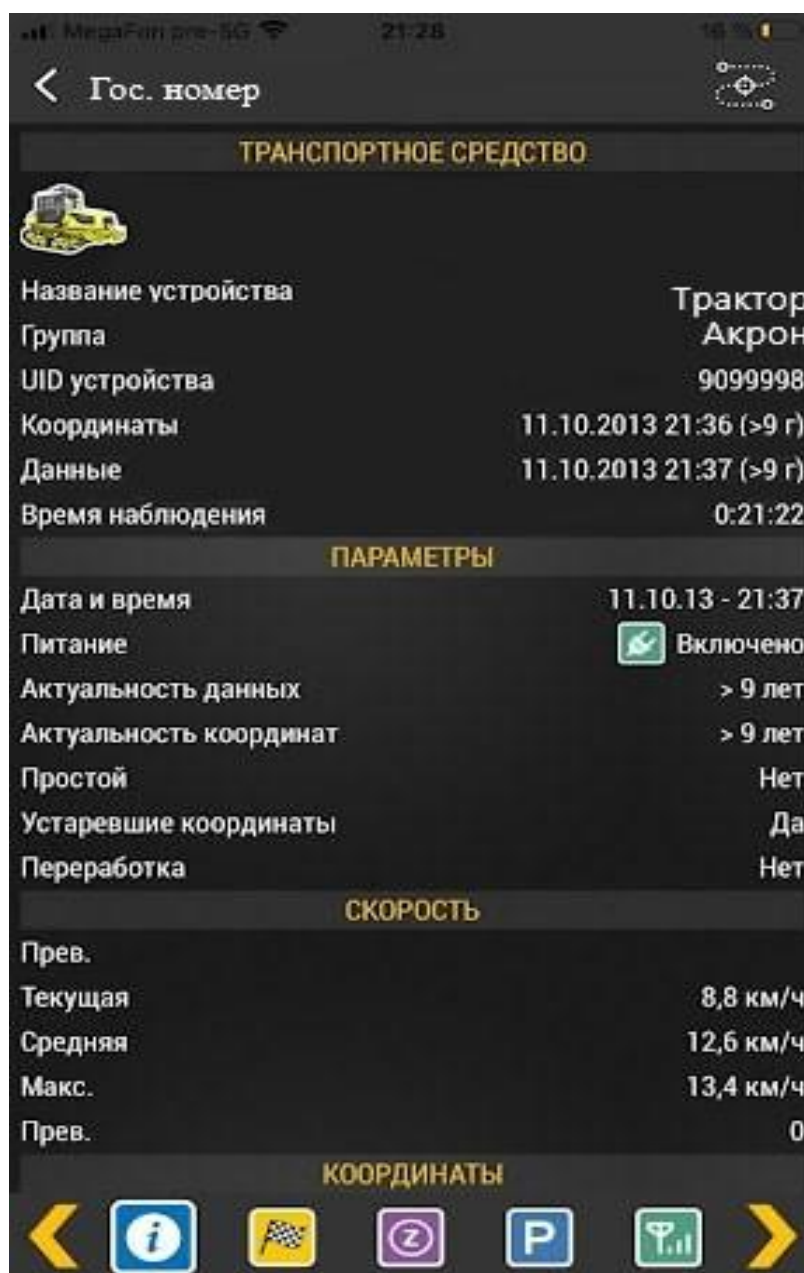


Рисунок 4 – Информация по транспортному средству

На этой странице пользователь может просмотреть информацию по отдельно выбранному ТС. Значки в нижней части экрана отображают состояние транспортного средства (общая информация по ТС, по двигателю, по остановкам, наличие сигнала и т.д.). Данная вкладка отображает основные данные по автомобилю.

Поисковая строка

На рисунке 5 изображена поисковая строка мобильного приложения.

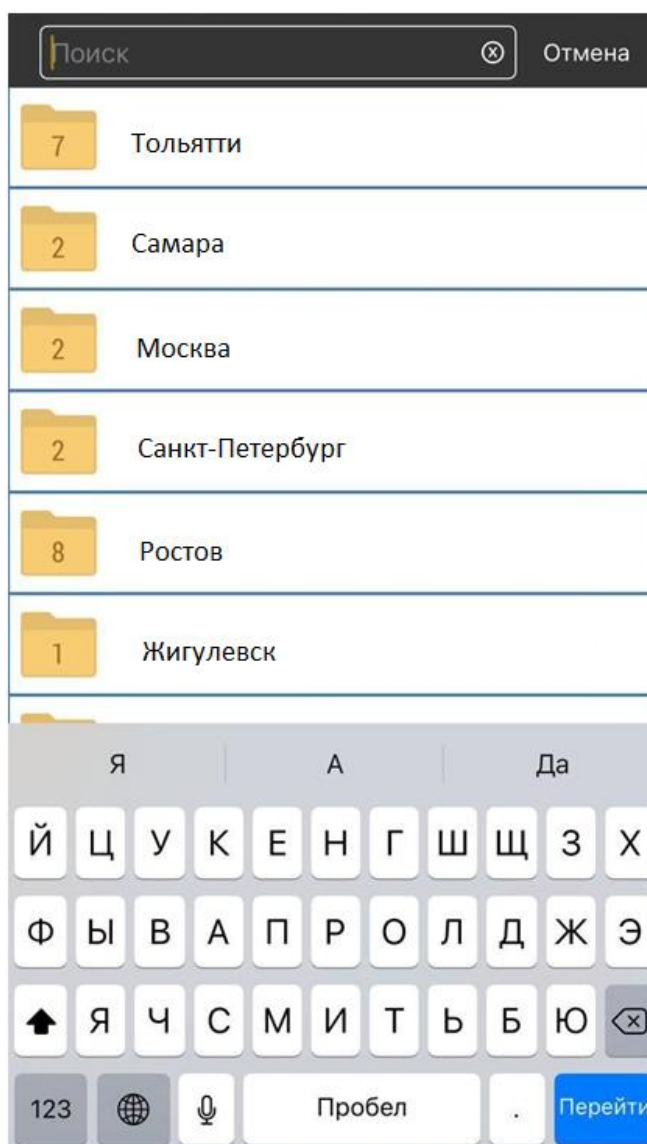


Рисунок 5 – Поисковая строка

Поисковая строка служит для удобства клиента в целях простоты поиска транспортного средства и экономии времени.

Так же оно позволяет быстро найти нужное транспортное средство и отображает его государственные номера.

Карта

На рисунке 6 отображена карта мобильного приложения.

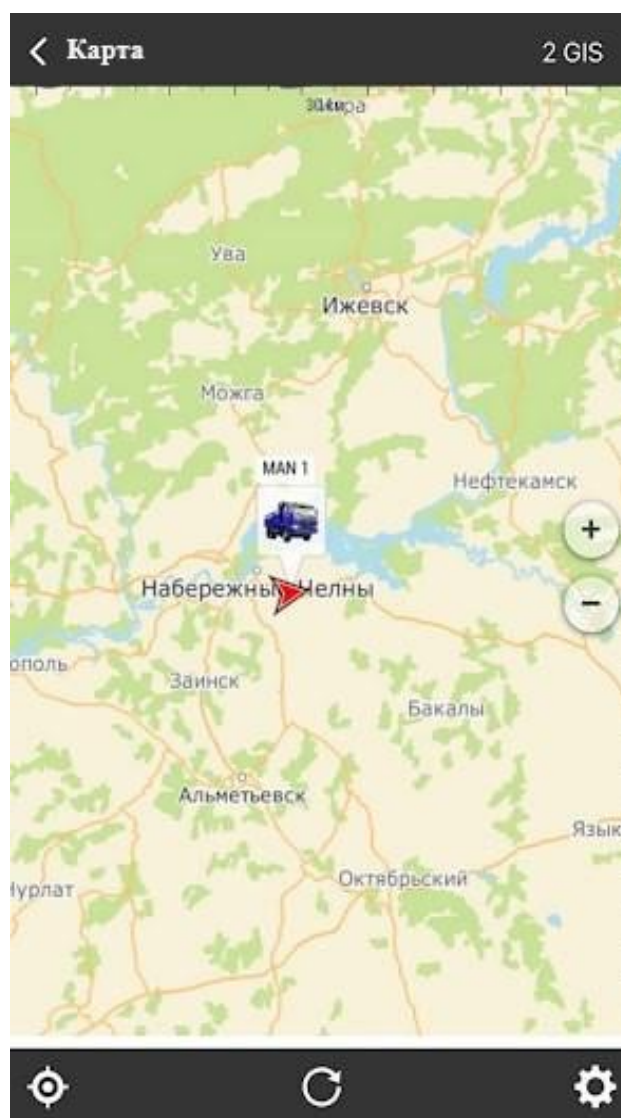


Рисунок 6 – Карта

Карта отображает местонахождение автомобиля и его координаты. На ней отображается символ ТС, в какую сторону он движется. Кнопка в нижнем левом углу экрана, отвечает за местоположение клиента (с

его разрешения).

Кнопка в нижней части экрана посередине, обновляет(перезагружает) карту.

Кнопка в нижнем правом углу экрана(настройки), предоставляет выбор карт (Яндекс, Google, 2ГИС и другие).

2.3 Авторизация

Авторизация пользователя - это процесс проверки подлинности и прав доступа пользователя к определенному ресурсу, например, к сайту или приложению[4]. Обычно авторизация осуществляется путем ввода логина и пароля, но могут использоваться и другие методы, такие как смс-коды или отпечаток пальца. После успешной авторизации пользователь может получить доступ к своему аккаунту, персонализированным настройкам и функциям, которые ограничены для неавторизованных пользователей.

Процесс авторизации не так прост, как в повседневных мобильных приложениях.

Что бы стать клиентом данного мобильного приложения, первоочередным действием должно быть заключение договора между физическим или юридическим лицом с “Автоград-Люкс”.

После чего, клиенту предстоит оборудовать свои транспортные средства системой ГЛОНАСС, для отображения информации в приложении.

Далее, клиент запрашивает логин и пароль от своей учетной записи. Почему же клиент сам не может создать свой личный кабинет? Потому что, в-первую очередь, учетная запись создается в web-приложении, где указываются определенные права для каждого клиента. Далее, этот же логин и пароль сообщаются человеку, заключившему договор, после чего он может совершить вход.

На рисунке 7 представлена авторизация пользователя в мобильном приложении.

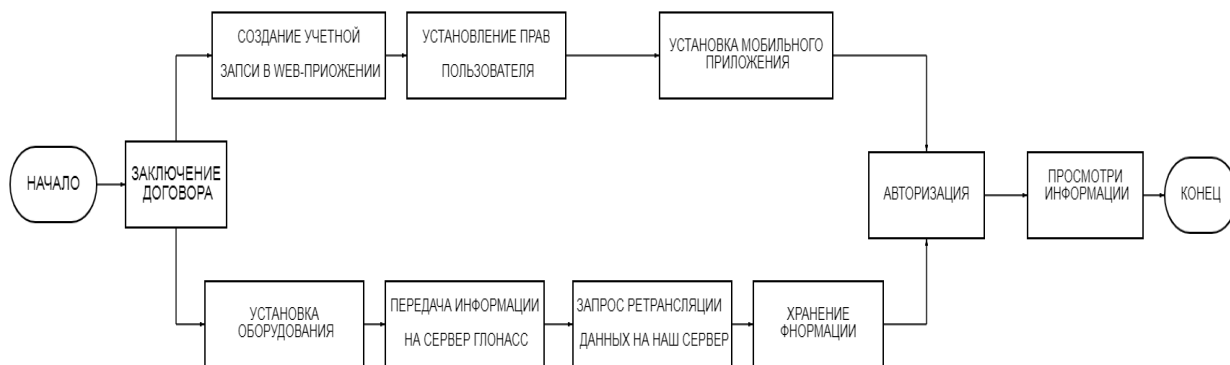


Рисунок 7 – Авторизация пользователя

2.4 Учет данных

На рисунке 8 изображена передача данных от блока в мобильное приложение.



Рисунок 8 – Передача данных от блока в мобильное приложение

2.5 Описание кода мобильного приложения

Авторизация пользователя

Данный код реализует авторизацию пользователей с помощью проверки логина и пароля. В начале программы создается список пользователей с их логинами и паролями. Затем запрашивается у пользователя логин и пароль, которые сохраняются в соответствующих переменных. Далее происходит поиск пользователя в списке пользователей и если он найден и его логин и пароль совпадают с теми, что были введены пользователем, то устанавливается флаг `isAuthenticated` в значение `true`[25]. Если пользователь не найден или его логин и/или пароль неверны, то флаг остается в значении `false`. В конце программы выводится сообщение о результате авторизации.

Рассмотрим на примере кода:

Для реализации авторизации пользователей с разными логинами и паролями в Java, можно использовать классы и методы из пакета `java.util`[17]:

```
import java.util.Scanner;

public class UserAuthenticator {

    public static void main(String[] args) {

        // Создаем список пользователей с логинами и паролями

        String[] users = {

            {"user1", "password1"},

            {"user2", "password2"},

            {"user3", "password3"}

        };

        // Запрашиваем у пользователя логин и пароль

        Scanner scanner = new Scanner();

        System.out.print("Введите логин: ");
```

```

String login = scanner.nextLine();
System.out.print("Введите пароль: ");
String password = scanner.nextLine();

// Ищем пользователя в списке
boolean isAuthenticated = false;
for (String[] user : users) {
    if (user[0].equals(login) && user[1].equals(password)) {
        isAuthenticated = true;
        break;
    }
}

// Выводим результат авторизации
if (isAuthenticated) {
    System.out.println("Добро пожаловать, " + login + "!");
} else {
    System.out.println("Неверный логин или пароль.");
}
}
}

```

Интерфейс карты в приложении

На примере подробнее разберем код для отображения карты Google. Разберем каждый пункт подробнее и приведем соответствующий код.

1. Получение API-ключа:

Для получения API-ключа нужно зарегистрироваться в Google Cloud Console, создать проект и включить API для сервиса Google Maps. Затем необходимо запросить API-ключ, который будет использоваться в вашем приложении.

```
// Запрашиваем API-ключ для карт Google
```

```
String apiKey = "your_api_key";
```

2. Добавление библиотеки Google Maps:

Библиотеку Google Maps можно добавить в проект, добавив следующую зависимость в файл build.gradle:

```
dependencies {  
    implementation 'com.google.android.gms:play-services-maps:17.0.0'  
}
```

3. Добавление элемента MapView в макет:

Добавим элемент MapView в макет activity_main.xml:

```
<com.google.android.gms.maps.MapView  
    android:id="@+id/map_view"  
    android:layout_width="match_parent"  
    android:layout_height="match_parent" />
```

4. Инициализация MapView в коде Activity:

Инициализируем MapView в методе onCreate() Activity:

```
public class MainActivity extends AppCompatActivity implements  
OnMapReadyCallback {
```

```
    private MapView mapView;
```

```
    protected void onCreate(Bundle savedInstanceState) {
```

```
        super.onCreate(savedInstanceState);
```

```
        setContentView(R.layout.activity_main);
```

```
        mapView = findViewById(R.id.map_view);
```

```
        mapView.onCreate(savedInstanceState);
```

```
        mapView.getMapAsync(this);
```



```
}  
}
```

5. Установка обратного вызова для получения объекта GoogleMap[18]-[19]:

Добавим обратный вызов метода `getMapAsync()`:

```
public class MainActivity extends AppCompatActivity implements  
OnMapReadyCallback {  
    private GoogleMap googleMap;  
    public void onMapReady(GoogleMap map) {  
        googleMap = map;  
    }  
    protected void onCreate(Bundle savedInstanceState) {  
        mapView.getMapAsync(this);  
    }  
}
```

6. Настройка отображения картых[5]:

Можно настроить отображение карты, например, задать тип карты, добавить маркер на карту и переместить камеру на этот маркер.

```
public class MainActivity extends AppCompatActivity implements  
OnMapReadyCallback {  
    private GoogleMap googleMap;  
    public void onMapReady(GoogleMap map) {  
        googleMap = map;  
  
        // Задаем тип карты  
        googleMap.setMapType(GoogleMap.MAP_TYPE_NORMAL);  
  
        // Добавляем маркер на карту
```

```

LatLng sydney = new LatLng(-33.852, 151.211);
MarkerOptions markerOptions = new MarkerOptions()
    .position(sydney)
    .title("Marker in Sydney");
googleMap.addMarker(markerOptions);

// Перемещаем камеру на маркер
googleMap.moveCamera(CameraUpdateFactory.newLatLng(sydney));
}
}

```

7. Управление состоянием MapView:

MapView содержит свое собственное состояние, которое нужно управлять в соответствии с жизненным циклом Activity. Для этого нужно переопределить методы жизненного цикла Activity и вызывать соответствующие методы у MapView.

```

public class MainActivity extends AppCompatActivity implements
OnMapReadyCallback {
    private MapView mapView;
    private GoogleMap googleMap;
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        mapView = findViewById(R.id.map_view);
        mapView.onCreate(savedInstanceState);
        mapView.getMapAsync(this);
        MapsInitializer.initialize(this);
    }
    protected void onResume() {

```

```
    super.onResume();
    mapView.onResume();
}
protected void onPause() {
    super.onPause();
    mapView.onPause();
}
protected void onDestroy() {
    super.onDestroy();
    mapView.onDestroy();
}
public void onLowMemory() {
    super.onLowMemory();
    mapView.onLowMemory();
}
public void onMapReady(GoogleMap map) {
    googleMap = map;
}}

```

3 Тестирование мобильного приложения

3.1.Метод тестирования

На рисунке 9 изображено окно с тестированием в Android Studio.

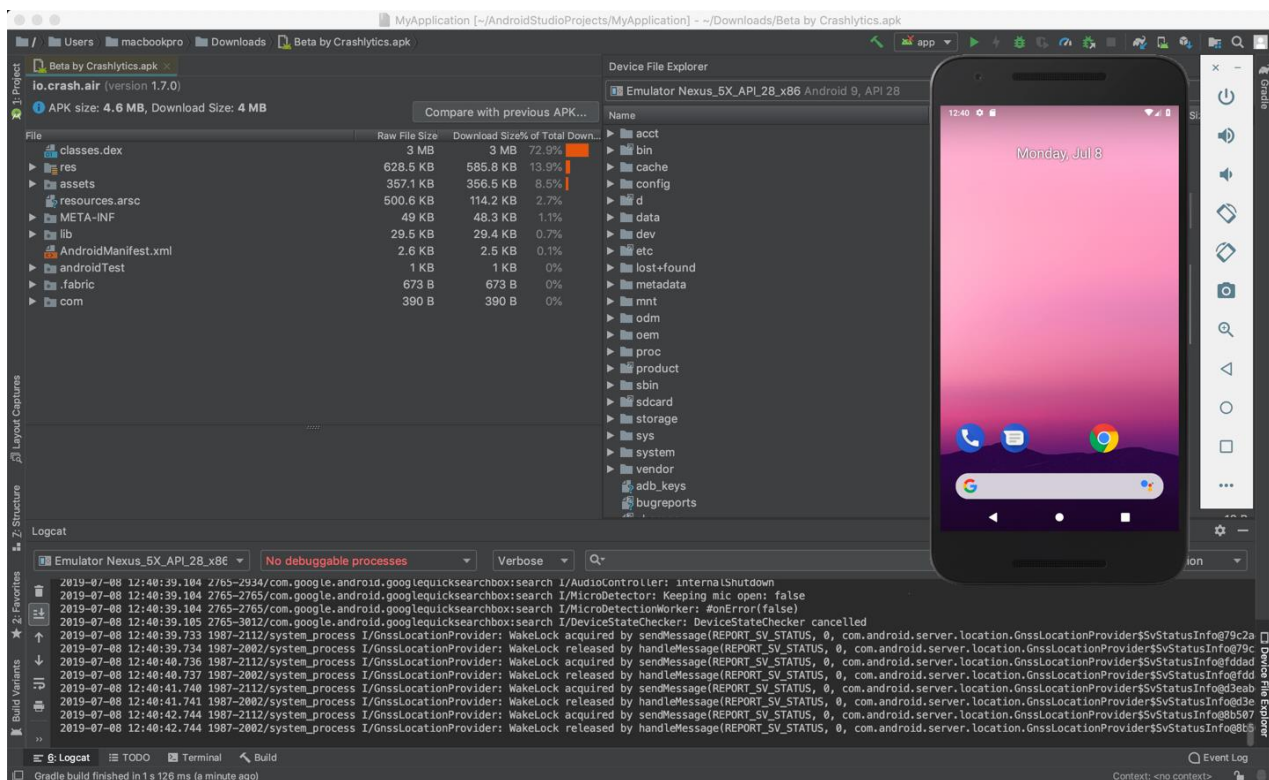


Рисунок 9 – Тестирование в Android Studio

Тестирование помогает выявить и устранить ошибки и недоработки в мобильном приложении до его релиза на рынок. Если тестирование было проведено успешно, то это означает, что все основные функции приложения работают без ошибок и не вызывают замечаний у пользователей[24]. Это может быть подтверждено результатами функционального, интеграционного, нагрузочного и других видов тестирования.

Тестирование мобильного приложения в Android Studio может происходить с помощью инструментов, предоставляемых самой средой разработки. Для этого можно использовать Android Instrumented Tests или Unit Tests.

Instrumented Tests - это тесты, которые выполняются на устройстве Android и используют AndroidJUnitRunner для выполнения тестовых сценариев. Эти тесты могут проверять функциональность приложения, включая взаимодействие с пользовательским интерфейсом, базой данных, сетью и другими компонентами[30]-[31].

Процесс тестирования с помощью Instrumented Tests включает в себя следующие шаги:

1. Создание тестового проекта, который будет содержать тестовые классы и зависимости.
2. Написание тестовых сценариев, которые будут проверять функциональность приложения.
3. Настройка AndroidJUnitRunner для запуска тестов на устройстве Android.
4. Запуск тестов на устройстве Android и анализ результатов.
5. Исправление выявленных ошибок и повторное тестирование до тех пор, пока все тесты не будут пройдены успешно.

Instrumented Tests - это важный инструмент для обеспечения качества приложения на платформе Android. Они позволяют проверить функциональность приложения и выявить ошибки, которые могут быть выявлены при тестировании на эмуляторе.

Unit Tests - это процесс тестирования программного кода на уровне отдельных модулей или компонентов (юнитов). Каждый модуль тестируется независимо от других, что позволяет выявлять и исправлять ошибки в ранней стадии разработки.

Процесс тестирования включает в себя написание тестовых сценариев, которые проверяют корректность работы каждой функции или метода в отдельности. Тесты могут быть написаны как вручную, так и автоматически с использованием специальных фреймворков для тестирования.

При написании тестовых сценариев важно учитывать все возможные варианты входных данных и выходных результатов, а также проверять

граничные условия. Кроме того, необходимо убедиться, что тестируемый код не взаимодействует с другими компонентами системы, чтобы исключить возможность ошибок, вызванных взаимодействием различных модулей.

После написания тестовых сценариев происходит их запуск и анализ результатов. Если тесты проходят успешно, то можно считать, что тестируемый код работает корректно. Если же тесты выявляют ошибки, то необходимо исправить код и повторить процесс тестирования до тех пор, пока все тесты не будут пройдены успешно.

В целом, Unit Tests - это важный инструмент для обеспечения качества программного кода и улучшения его надежности и стабильности.

В итоге принято решение воспользоваться методом Instrumented Tests , так как он имеет ряд преимуществ , которые важны при тестировании мобильного приложения:

1. Более реалистичное тестирование: Instrumented Tests позволяют тестировать приложение на реальном устройстве, что дает более точные результаты, чем тестирование на эмуляторе.

2. Большая покрытие кода: позволяют тестировать не только функциональность приложения, но и взаимодействие с системой, такое как доступ к базе данных, работа с сетью и т.д.

3. Более простое отладка: позволяют отслеживать и исправлять ошибки в приложении непосредственно на устройстве, что упрощает процесс отладки.

4. Улучшение качества приложения: помогают выявлять ошибки и уязвимости в приложении, что позволяет улучшить его качество и уменьшить количество возможных проблем для пользователей.

5. Ускорение процесса разработки: позволяют автоматизировать тестирование, что ускоряет процесс разработки и позволяет сосредоточиться на других задачах.

3.2. Результаты тестирования

Далее следует проверка соответствия приложения требованиям. Это может включать в себя проверку функциональности, например, проверку работы всех кнопок и ссылок, проверку правильности отображения данных и проверку соответствия результатов работы приложения ожидаемым результатам.

В результате тестирования в теоретической части должен быть составлен отчет, содержащий информацию о найденных проблемах и рекомендации по их устранению. Это помогает разработчикам улучшить приложение и обеспечить его более качественную работу.

В результате тестирования, было выявлено, что приложение имело мелкие недоработки, такие как:

- Не работающий бегунок в разделе “Период просмотра данных”
- Не обновляющиеся данные по транспортным средствам
- Кнопка перехода на карту

Вышеперечисленные пункты были исправлены с помощью тестирования приложения на эмуляторе.

Выявлены ошибки в коде и неправильно построенные пути, вследствие чего, неполадки были устранены.

Очередность тестирования происходит в следующем порядке :

- Результаты тестирования
- Ошибки
- Степень критичности
- Рекомендации по устранению ошибок
- Исправление ошибок
- Повторное тестирование
- Подтверждение исправлений

Исправленный бегунок :

```
import java.awt.*;
```

```

import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
import javax.swing.*;

public class DateSwitcher extends JPanel implements ActionListener {
    private JButton prevButton, nextButton;
    private JLabel dateLabel;
    private DateCounter dateCounter;
    public DateSwitcher(DateCounter dateCounter) {
        super();

        this.dateCounter = dateCounter;

        prevButton = new JButton("<");
        prevButton.setMargin(new Insets(0, 5, 0, 5));
        // Добавляем небольшие отступы между текстом и границей кнопки
        prevButton.addActionListener(this);
        // Добавляем обработчика события к кнопке
        nextButton = new JButton(">");
        nextButton.setMargin(new Insets(0, 5, 0, 5));
        nextButton.addActionListener(this);
        dateLabel = new JLabel(dateCounter.getDate().toString());
        // Создаем метку, которая будет отображать текущую дату
        add(prevButton);
        add(dateLabel);
        add(nextButton);
    }
    public void actionPerformed(ActionEvent e) {
        if (e.getSource() == prevButton) {
            dateCounter.decrement();
        } else if (e.getSource() == nextButton) {

```



```

        dateCounter.increment();
    }
    dateLabel.setText(dateCounter.getDate().toString()); // Обновляем
метку с новой датой
}

```

```

public static void main(String[] args) {
    JFrame frame = new JFrame("Date Switcher");
    DateCounter dateCounter = new DateCounter();
    // Создаем объект-счетчик даты
    DateSwitcher dateSwitcher = new DateSwitcher(dateCounter);
    // Создаем объект-переключатель даты
    frame.getContentPane().add(dateSwitcher);
    frame.pack();
    frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
    frame.setVisible(true);
}
}

```

```

class DateCounter {
    private long currentDate;
    public DateCounter() {
        // Используем текущую дату и время в качестве начального значения
        this.currentDate = System.currentTimeMillis(); }
    public void increment() {
        // Увеличиваем значение на один день (в миллисекундах)
        currentDate += 86400000; }
    public void decrement() {
        currentDate -= 86400000;
        // Уменьшаем значение на один день (в миллисекундах)
    }
}

```

```
}  
public java.sql.Date getDate() {  
    return new java.sql.Date(currentDate);  
}  
}
```

Заключение

В целом, разработка мобильного приложения является сложным и многопроцессным процессом, который требует множества времени и компетенций в области программирования, дизайна и тестирования. Однако, благодаря преимуществам современных технологий и инструментов, разработка мобильного приложения становится более доступной и эффективной.

В рамках выпускной квалификационной работы было разработано мобильное приложение, позволяющее производить мониторинг транспортных средств и получать полную информацию по ним.

Мною были проанализированы такие мобильные приложения, как: Автоскан+, КГК Мониторинг, SKYriver. Ссылаясь на техническое задание, была выбрана операционная система мобильного устройства для работы приложения и среда для ее разработки.

В ходе выполнения выпускной квалификационной работы были решены следующие задачи:

- Проанализированы аналогичные мобильные приложения на рынке.
- Создан дизайн и добавлен функционал приложению.
- Разработано мобильное приложение на операционной системе Android.
- Выбрана среда реализации.
- Произведен учет данных.

Результатом работы является создание мобильного приложения «Автоград-люкс».

Разработанный проект соответствует всем требованиям технического задания. Таким образом, задачи выпускной квалификационной работы полностью решены и цель исследования достигнута.

Список используемой литературы

1. Аарон Хиллегасс. Objective-C. Программирование для Android.
https://vk.com/wall-54530371_679
2. 1. Гриффитс Р. Д. Head First. Программирование для Android [Текст] / Р. Д. Гриффитс – Санкт-Петербург: Питер, – 704 с.
<https://dokumen.pub/head-first-android-2nbsped-978-5-4461-0708-7.html>
3. Дэрси Л. Разработка приложений для Android-устройств. Базовые принципы [Текст] /Л. Дэрси, Ш. Кондер – Том 1. – Москва: Эксмо, 2019. – 598 с. <https://b2b2.su/book-2045.html>
4. Джордж Шеферд. Программирование на Microsoft Visual C++, 2018, 423 с. <https://koha.lib.tsu.ru/cgi-bin/koha/opac-detail.pl?biblionumber=211800>
5. Использование мобильных устройств [Электронный ресурс]. – Режим доступа: <http://www.wi-life.ru/stati/wi-fi/marketingovye-stati2/mobiledevices-use-aruba-research>
6. Якоб Нильсен, Ралука Будиу. Как создавать идеально удобные приложения для мобильных устройств. 2019г.
<https://ux.pub/editorial/kompliksnioie-rukovodstvo-po-dizainu-mobilnykh-prilozhenii-5962>
7. Основные этапы разработки мобильных приложений [Электронный ресурс] – Режим доступа: <https://blog.sbssoft.ru/post/etapy-razrabotki-mobilnogo-prilozheniya>
https://www.aisol.ru/articles/sozдание_mobilnyh_prilozheniy_etapy_razrabotki
8. Основные этапы разработки мобильных приложений
<https://spark.ru/startup/componentix/blog/4499/osnovnie-etapi-razrabotkimobilnih-prilozhenij>
9. Майер Р. Программирование приложений для планшетных компьютеров и смартфонов [Текст] / Р. Майер – Москва: Эксмо, 2018. – 816с.
<https://monster-book.com/android-2-programmirovanie-prilozheniy>

10. Амелин К. С., Граничин О. Н., Кияев В. И., Корявко А. В.. Введение в разработку приложений для мобильных платформ. [Текст] Издательство ВВМ, <https://math.spbu.ru/user/gran/mobile/AtomBookM.pdf>
11. Глушаков, С.В. Все секреты, трюки и эффекты Photoshop, <https://www.chitalkino.ru/glushakov-s-v/vse-sekrety/>
12. Illustrator, Corel [Текст] / С.В. Глушаков. - Москва: РГГУ, <http://product.corel.com/help/CorelDRAW/540223850/Main/RU/Documentation/wwhelp/wwhimpl/common/html/wwhelp.htm#href=CorelDRAW-Finding-editing-and-converting-text.html&single=true>
13. Head AUT. Программирование для Андроид. <https://metanit.com/java/android/>
14. М.Федотенко. Разработка мобильных приложений. Первые шаги. <https://dokumen.pub/9785001011927.html>
15. Эд Бурнет. Привет, Android! Разработка мобильных приложений. <https://knigogid.ru/books/277036-privet-android-razrabotka-mobilnyh-prilozheniy>
16. Денис Колисниченко. Программирование для Android https://bookz.ru/authors/denis-kolisni4enko/pdf_programm_5582097.html
17. Герберт Шилдт. Lava. Полное руководство. https://vk.com/doc26879026_492639796?hash=BOzfSwfyQMNgDfa1egEGevjkzFZ1lp9KXErp7AZ4Acs&dl=VkZth6QhxQ9StlTu3shJeK29xGl7zFBMhQ08DSJFVt0
18. Гохберг, Г.С. Информационные технологии: Учебник / Г.С. Гохберг. - М.: Academia, https://academia-moscow.ru/ftp_share/books/fragments/fragment_21985.pdf
19. Android. Сборник рецептов. Задачи и решения для разработчиков приложений. 2-е издание, https://vk.com/doc26879026_470723087?hash=GmpbWUfzw5lehXQA0urxVssFLqyaWzrtlg80dlM38LP&dl=1Nw0fZxbeEhV8rCERIZjjBrVTQDcK2YR8wuE9G3AtU0

20. Java 8. Руководство для начинающих,
https://vk.com/doc26879026_493091194?hash=jdfMtGro7jCnHQiMzz0YzloKfMhyRBguOkgHBQIAzzD&dl=MSSfj8uquCXfl947rktScYFT6vv8T0cX3SiSBsHt60z

21. Пол Дейтел, Харви Дейтел, Александер Уолт. Android для разработчиков

[.file:///C:/Users/nikit/AppData/Local/Temp/Rar\\$DIa4092.37292/Android%20%D0%B4%D0%BB%D1%8F%20%D1%80%D0%B0%D0%B7%D1%80%D0%B0%D0%B1%D0%BE%D1%82%D1%87%D0%B8%D0%BA%D0%BE%D0%B2,%20%D0%B5%20%D0%B8%D0%B7%D0%B4%D0%B0%D0%BD%D0%B8%D0%B5.pdf](file:///C:/Users/nikit/AppData/Local/Temp/Rar$DIa4092.37292/Android%20%D0%B4%D0%BB%D1%8F%20%D1%80%D0%B0%D0%B7%D1%80%D0%B0%D0%B1%D0%BE%D1%82%D1%87%D0%B8%D0%BA%D0%BE%D0%B2,%20%D0%B5%20%D0%B8%D0%B7%D0%B4%D0%B0%D0%BD%D0%B8%D0%B5.pdf)

22. Android App Development for Dummies,
https://vk.com/doc255577237_618008864?hash=1BzjgEG4GISQiy14CE7JICGZ3PDmFQ4Xz1XjkFcIHw4&dl=doX1W2Zp38l0fCP3NZnmJsU5wPri6Tspz73IbcvTOyD

23. Android NDK Game Development Cookbook,
<https://pdfkeys.com/download/2639381-The-Android-Developers-Cookbook-Building-Applications-With-The-Android-Sdk-Building-Applications-With-The-Android-Sdk-Developers-Library.pdf>

24. Проектирование пользовательского интерфейса в Android
<https://www.fandroid.info/postroenie-prostogo-polzovatelskogo-interfejsa/>

25. Head First. Программирование для Android. 2-е издание
https://vk.com/doc26879026_470791860?hash=sqrrPQSu3O2yZLzJuVALIJBzPH5KtxSUKsYTORLS2ic&dl=Hed03n9QIokp6UHSzoR3OFUIScC06ZKSIov1ysBjdLL

26. Ян Ф. Дарвин «Android. Сборник рецептов. Задачи и решения для разработчиков приложений»
https://vk.com/doc26879026_470723087?hash=GmpbWUfzw5lehXQA0urxVssFL

qyaWzrtIg80dlM38LP&dl=1Nw0fZxbeEhV8rCERIZjjBrVTQDcK2YR8wuE9G3AtU0

27. John Horton «Learning Java by Building Android Games — Explore Java Through Mobile Game Development»
<https://www.packtpub.com/product/learning-java-by-building-android-games-second-edition/9781788839150>

28. James S. Cho «The Beginner's Guide to Android Game Development»
https://i-share-kis.primo.exlibrisgroup.com/discovery/fulldisplay?context=L&vid=01CARLI_KIS:CARLI_KIS&search_scope=MyInstitution&tab=LibraryCatalog&docid=alma991198277505853

29. Руководство по разработке мобильных приложений
<https://appmaster.io/ru/blog/rukovodstvo-po-razrabotke-mobil-nykh-prilozhenii>

30. Саид, Хашими Android 3 для профессионалов. Создание приложений для планшетных компьютеров и смартфонов / Хашими Саид. - М.: Диалектика / Вильямс
https://books.google.ru/books?id=e1jrDwAAQBAJ&pg=PT527&lpg=PT527&dq=Саид,+Хашими+Android+3+для+профессионалов.+Создание+приложений+для+планшетных+компьютеров+и+смартфонов+/-+Хашими+Саид.+/-+М.:+Диалектика+/-+Вильямс&source=bl&ots=T9K8at7VZz&sig=ACfU3U2AxVRQ12d-AIN2abWMzru-g8C6wA&hl=ru&sa=X&ved=2ahUKEwiCqxqydsNb_AhURm4sKHWM2ASA4ChDoAXoECCMQAw#v=onepage&q&f=false

31. Машнин, Тимур Eclipse: разработка RCP-, Web-, Ajax- и Android-приложений на Java / Тимур Машнин. - М.: БХВ-Петербург,
https://books.4nmv.ru/books/eclipse_razrabotka_rcp-web-ajax-i-android-prilozhenii_na_java_3643559.pdf

32. Коматинени, Сатия Android 3 для профессионалов. Создание приложений для планшетных компьютеров и смартфонов / Сатия Коматинени, Дэйв Маклин, Саид Хашими

https://books.google.ru/books?id=e1jrDwAAQBAJ&pg=PT527&lpg=PT527&dq=Коматинени,+Сатия+Android+3+для+профессионалов.+Создание+приложений+для+планшетных+компьютеров+и+смартфонов+/+Сатия+Коматинени+,+Дэйв+Маклин+,+Саид+Хашими&source=bl&ots=T9K8at7YTy&sig=ACfU3UIGs1tI8mEV6Vl6RBCH6-dPwo9zcg&hl=ru&sa=X&ved=2ahUKEwiw6vW0sdb_AhVVm4sKHQgDCIE4FBD0AXoECBAQAw#v=onepage&q=Коматинени, Сатия Android 3 для профессионалов. Создание приложений для планшетных компьютеров и смартфонов / Сатия Коматинени , Дэйв Маклин , Саид Хашими&f=false

33. Голощاپов, Алексей Google Android. Создание приложений для смартфонов и планшетных ПК (2-е издание) / Алексей Голощاپов
<https://lib.fbtuit.uz/assets/files/Google-Android.PDFDrive.pdf>

34. Голощاپов, А.Л. Google Android. Программирование для мобильных устройств (+ CD-ROM) / А.Л. Голощاپов
https://books.google.ru/books?id=eSQ4BAAAQBAJ&pg=PA25&hl=ru&source=gb_s_toc_r&cad=4#v=onepage&q&f=false