

Министерство науки и высшего образования Российской Федерации
Муромский институт (филиал)
Федерального государственного бюджетного образовательного учреждения высшего
образования
«Владимирский государственный университет
имени Александра Григорьевича и Николая Григорьевича Столетовых»

Факультет _____ ИТР _____

Кафедра _____ ПИН _____

КУРСОВАЯ РАБОТА

По _____ Разработка приложений для мобильных операционных систем _____

Тема _____ Приложение «Детейлинг центр» _____

(оценка)

Руководитель

Колпаков А.А.
(фамилия, инициалы)

(подпись) (дата)

Члены комиссии

(подпись) (Ф.И.О.)

(подпись) (Ф.И.О.)

Студент ПИН-121
(группа)

Кокурин Я.Д.
(фамилия, инициалы)

(подпись) (дата)

В данной курсовой работе было необходимо разработать мобильное приложение для детейлинг центра с использованием операционной системы Android и языка программирования Kotlin. В качестве средств разработки приложения была использована среда Android Studio 2024.1.2. Для хранения необходимой информации применяется SQLite, управление версиями осуществлялось через систему контроля версий GitHub. По окончании разработки приложения было выполнено функциональное тестирование интерфейса с использованием виртуальных устройств.

In this course work, it was necessary to develop a mobile application for a car detailing center using the Android operating system and the Kotlin programming language. The Android Studio 2024.1.2 environment was used as the application development tools. SQLite is used to store the necessary information, version control was carried out through the GitHub version control system. Upon completion of the application development, functional testing of the interface using virtual devices was performed

Содержание

Введение.....	6
1 Анализ технического задания.....	8
1.1 Постановка задачи	8
1.2 Назначение разрабатываемого программного продукта	8
1.3 Анализ исходных данных к курсовой работе	9
2 Разработка алгоритмов	11
2.1 Концептуальная модель данных	11
2.2 Логическая модель данных.....	12
2.3 Физическая модель данных	12
2.4 Разработка диаграммы прецедентов.....	13
2.5 Разработка и реализация приложения	14
2.6 Тестирование приложения.....	15
3 Руководство программиста	17
4 Руководство пользователя	24
Заключение	27
Список литературы	28
Приложение 1	29
Приложение 2	31
Приложение 3	32

					МИВлГУ 09.03.04 - 0.012			
<i>Изм.</i>	<i>Лист</i>	<i>№ докум.</i>	<i>Подпись</i>	<i>Дата</i>				
Разраб.		Кокурин Я.Д.			«Детейлинг центр»	<i>Лит.</i>	<i>Лист</i>	<i>Листов</i>
Провер.		Колпаков А.А.					5	43
Реценз.						МИ ВлГУ ПИН-121		
Н. Контр.								
Утверд.								

Введение

В настоящее время для достижения успехов любому предприятию необходимо точно понимать свои затраты, прибыли, ресурсы, бизнес-процессы и многое другое. Наглядная информация о происходящем поможет глубже проанализировать процесс и сделать правильные выводы, что в конечном итоге приведет к повышению эффективности.

Лидерами становятся наиболее эффективные предприятия, имеющие минимальные издержки, высочайший уровень производительности и полностью контролируемые и четко отлаженные процессы. Мобильные приложения способствуют повышению производительности, улучшению связи с клиентами или оптимизации бизнес-процессов.

Мобильные приложения позволяют компаниям лучше взаимодействовать со своими клиентами, предоставляя им удобный способ запросить информацию, обращаться в службу поддержки или совершать покупки.

Приложения для бизнеса могут служить дополнительным каналом продаж, позволяя клиентам совершать покупки и ознакомиться с продуктами или услугами непосредственно через приложение.

Мобильные приложения могут помочь компаниям собирать данные о поведении пользователей, предпочтениях и трендах, что позволяет проводить аналитику и улучшать стратегии маркетинга и продаж.

В данной курсовой работе рассматривается создание мобильного приложения для детейлинг центра, базирующееся на операционной системе Android с использованием языка программирования Kotlin.

Детейлинг центр – предприятие, занимающееся оказанием специализированных процедур по уходу за автомобилем. Детейлинг подразумевает максимально тщательную и качественную проработку внешнего и внутреннего вида автомобиля до состояния близкого к идеальному и сохранение этого результата на длительный период времени.

					МИВлГУ 09.03.04 – 0.012	Лист
Изм.	Лист	№ докум.	Подпись	Дата		6

Одной из возможностей повышения рентабельности детейлинг центра является использование клиентского приложения для оформления клиентами заявок на оказание услуг для их автомобилей.

Целями данной работы являются разработка моделей информационной системы детейлинг центра с учётом требуемых обработки данных для последующего проектирования клиентского мобильного приложения.

Задачей проекта является программная реализация мобильного приложения, проверка его работоспособности.

					МИВлГУ 09.03.04 – 0.012	Лист
						7
Изм.	Лист	№ докум.	Подпись	Дата		

1 Анализ технического задания

1.1 Постановка задачи

Разрабатываемое приложение «Детейлинг центр» предназначено для удобного оформления клиентами заявок оказания услуг детейлинг центром для автомобилей. Программный продукт будет использоваться частными лицами, желающими восстановить или поддержать состояние своего транспортного средства. Приложение предоставляет пользователю следующие возможности:

- добавление автомобилей в список своих автомобилей;
- оформление заявки на оказание услуги для автомобиля в выбранную дату;
- просмотр подробной информации об услугах детейлинг центра;
- просмотр данных и хранение истории по оставленным заявкам;
- добавление других пользователей с их автомобилями.

1.2 Назначение разрабатываемого программного продукта

Разрабатываемое мобильное приложение «Детейлинг центр» создается с целью обеспечения удобства записи своего автомобиля на оказание услуги детейлинг центром. Назначение разрабатываемого программного продукта включает в себя:

1) Упрощение процедуры записи: приложение предоставляет легкий и удобный способ добавления пользователей, автомобилей пользователей, а также возможность оформления заявки на оказание услуги для автомобиля в выбранную дату;

2) Хранение оставленных заявок: в приложении присутствует возможность просмотра списка оставленных заявок. В данных заявках пользователь может наблюдать статус заявки. В случае если услуга была выполнена по заявке - увидеть исполнителя;

					МИВлГУ 09.03.04 – 0.012	Лист
Изм.	Лист	№ докум.	Подпись	Дата		8

3) Предоставление информации по запросу: пользователи могут получить доступ к информации об актуальных услугах, предоставляемых детейлинг центром. Это упрощает процесс принятия решений при выборе услуг.

Преимущества автоматизации включают в себя:

1) Ускорение обработки информации: система обеспечивает быстрый и легкий доступ к данным, что ускоряет процесс оформления заявки на оказание услуги и снижает временные траты;

2) Удобство и доступность: пользователи могут легко и быстро взаимодействовать с приложением, вне зависимости от их технических навыков;

3) Адаптация к современным тенденциям: учитывается рост цифровых технологий и предоставляется возможность пользователям взаимодействовать с системой в рамках современных цифровых тенденций.

1.3 Анализ исходных данных к курсовой работе

Выбор использования операционной системы Android и языка программирования Kotlin обусловлен следующими факторами:

- Поддержка Google: Kotlin является официальным языком для разработки под Android. Google активно поддерживает его и внедряет новые функции, что делает язык более стабильным и перспективным для мобильной разработки;

- Современные возможности: Kotlin предлагает множество современных возможностей, к примеру - расширенные функции работы с коллекциями и мощную систему типизации, что упрощает и ускоряет разработку;

- Интероперабельность с Java: Kotlin полностью совместим с Java, что позволяет использовать существующие Java-библиотеки и постепенно переходить на новый язык без необходимости переписывать весь код;

- Сообщество и ресурсы: сообщество разработчиков Kotlin растёт, появляются новые библиотеки, фреймворки и учебные материалы, что делает процесс изучения более доступным и поддерживаемым;

Приложения, написанные на Android, могут быть запущены на множестве устройств, включая виртуальные машины и домашние приставки.

Разрабатываемое приложение должно иметь возможности:

- 1) Создание пользователей - клиентов;
- 2) Добавление автомобилей пользователями;
- 3) Просмотр актуальных услуг детейлинг центра;
- 4) Оформление заявок пользователем на оказание услуги для автомобиля;
- 5) Просмотр истории оформленных заявок и их статус;
- 6) Отправка уведомлений пользователю приложения.

При разработке мобильного приложения на Android для хранения и организации информации было принято решение об использовании базы данных. В качестве базы данных использовалась встраиваемая реляционная база данных SQLite.

SQLite представляет собой встраиваемую реляционную базу данных, которая обеспечивает легкий доступ к базе данных без необходимости отдельного сервера. SQLite широко используется в мобильных приложениях, встроенных системах и других проектах, где требуется компактная и простая в использовании база данных

2 Разработка алгоритмов

Разработка алгоритмов является ключевым этапом в создании приложения. Этот процесс не только обеспечивает эффективное функционирование программы, но и является основой для ряда других важных шагов в разработке. В данном контексте осуществление следующих пунктов является неотъемлемой частью разработки алгоритмов.

Разработка концептуальной модели данных: данная модель необходима для представления в визуально удобном виде связей между объектами данных и их характеристиками, впоследствии необходима для разработки логической и физической моделей данных.

Разработка логической и физической модели данных: эти модели служат основой для оптимизированного проектирования алгоритмов работы с информацией.

Разработка диаграммы прецедентов: диаграмма прецедентов помогает выделить основные функциональные возможности приложения и отобразить возможное взаимодействие пользователя с системой. Разработка алгоритмов тесно связана с определением шагов, которые пользователь может предпринять, и соответствующих им процессов обработки данных.

2.1 Концептуальная модель данных

Концептуальная модель хранилища данных представляет собой описание главных (основных) сущностей и отношений между ними. Концептуальная модель является отражением предметных областей, в рамках которых планируется построение хранилища данных.

При проектировании концептуальной модели структурируют данные и выявляют взаимосвязи между ними, без рассмотрения особенностей реализации и вопросов эффективности обработки, поэтому концептуальная модель не является полностью подходящей для дальнейшей разработки, все таблицы должны быть нормализованы для реляционной базы данных.

Составленная концептуальная модель представлена на рисунке 1 приложения 1.

2.2 Логическая модель данных

Логическая модель расширяет концептуальную путем определения для сущностей их атрибутов, описаний и ограничений, уточняет состав сущностей и взаимосвязи между ними.

Концептуальная модель изменяется так, чтобы она могла быть обеспечена конкретной моделью данных.

В результате формируется логическая модель. Логическая модель отражает логические связи между элементами данных вне зависимости от их содержания и среды хранения.

Логическая модель может быть реляционной, иерархической или сетевой.

В качестве способа организации информационной базы выбрана реляционная база данных. Именно такой способ хранения всех данных является наиболее подходящим для проектируемой информационной системы по следующим причинам:

- наглядность модели для пользователя: все данные в реляционной модели представлены в табличной форме;
- независимость данных от программного продукта для их обработки;
- реляционные базы данных являются наиболее распространенными среди разработчиков ПО, следовательно, использование этих баз позволит сэкономить время и бюджет на внедрение нового типа БД.

Составленная логическая модель представлена на рисунке 2 приложения 1.

2.3 Физическая модель данных

Физические модели данных служат для отображения моделей данных. Основными понятиями модели данных являются поле, логическая запись,

логический файл. Слово «логический» введено, чтобы отличать понятия, относящиеся к логической модели данных, от понятий, относящихся к физической модели данных. Основными понятиями физической модели данных, используемыми для представления логической модели данных, являются поле, физическая запись, физический файл. В частности, логическая запись, состоящая из полей, может быть представлена в виде физической записи (из тех же полей), логический файл – в виде физического файла. Имена таблиц и колонок будут сгенерированы на основе сущностей и атрибутов логической модели, учитывая максимальную длину имени и другие синтаксические ограничения, накладываемые СУБД. Если в имени сущности или атрибута встречается пробел, он заменяется на символ «_».

Физическая модель описывает способ хранения данных в базе данных. В физической модели мы учитываем типы данных, индексы, ограничения целостности и другие технические детали.

Составленная физическая модель представлена на рисунке 3 приложения 1.

2.4 Разработка диаграммы прецедентов

Разработка диаграммы прецедентов является одним из ключевых этапов анализа и проектирования системы, особенно в методологии разработки программного обеспечения. Диаграмма прецедентов позволяет идентифицировать и описать функциональные возможности системы, включая взаимодействие между пользователями и системой.

Диаграмма прецедентов представляет собой графическое представление взаимодействия пользователей с системой через выполнение различных прецедентов. Она отражает функциональные требования системы и ее возможности.

Диаграмма прецедентов помогает разработчикам и заинтересованным сторонам понять функциональные требования системы, определить основные актеры и их роли, а также определить взаимодействие между

актерами и прецедентами. Она служит основой для дальнейшего проектирования и разработки системы, а также для проверки соответствия системы требованиям пользователей.

На рисунке 4 в приложении 1 представлена диаграмма прецедентов для пользователя системы.

На диаграмме представлены действия, которые могут выполняться пользователями. Эти действия описывают функциональность системы и включают в себя последовательность шагов, выполняемых пользователем. Функциональность пользователя направлена на оформление заявки оказания услуги для автомобиля детейлинг центром.

2.5 Разработка и реализация приложения

Приложение «Детейлинг центр», созданное в рамках данной курсовой работы, включает в себя несколько компонентов, обеспечивающих удобное оформление заявок оказания услуг для автомобилей.

Первым компонентом приложения является экран списка услуг детейлинг центра. Здесь пользователь может просматривать все доступные услуги и ознакомиться с подробностями о заинтересованной услуге с помощью перехода на страницу информации об услуге.

Следующим компонентом является экран списка пользователей, на нём пользователь может перейти на экран добавления пользователя или экран со списком автомобилей выбранного пользователя. После появляется возможность добавления нового автомобиля или просмотр оставленных заявок для выбранного из списка автомобиля на соответствующих экранах.

На экране оставленных заявок пользователю предоставляется возможность просмотра существующей заявки и добавления новой. При добавлении необходимо выбрать услугу и дату её оказания, после чего заявка перейдёт в статус «Ожидание», а на телефон придёт уведомление об оставленной заявке.

Таким образом, приложение «Детейлинг центр» представляет удобный интерфейс для ознакомления с услугами, предоставляемых детейлинг центром и последующего оформления заявок. Функционал приложения обеспечивает эффективное взаимодействие с пользователями, соответствующее их потребностям при заказе услуги детейлинг центра.

2.6 Тестирование приложения

Целью проведения тестирования является подтверждение реализации требуемой функциональности системы. В процессе тестирования, результат каждого теста фиксируется. Считается, что тест прошёл успешно в случае, если результат совпадает с ожидаемым результатом. Если результат отличается от ожидаемого, он фиксируется в протоколе тестирования. В таблице 1 приведена методика тестирования разработанного программного продукта.

Таблица 1 - Методика тестирования разработанной программы

Выполненное действие	Полученный результат
Запуск приложения	Отображение стартовой страницы
Нажатие на кнопку «Посмотреть список услуг»	Отображение списка актуальных услуг детейлинг центра
Нажатие на услугу из списка	Переход на экран с информацией об услуге
Свайп влево	Возврат к предыдущему экрану
Нажатие на кнопку «Пользователи»	Отображение экрана со списком пользователей
Нажатие на кнопку «Добавить пользователя»	Отображение экрана с полями для ввода данных о пользователе
Повторное нажатие «Добавить пользователя»	В случае корректности введенных данных происходит проверка существует ли пользователь с указанным номером телефона в системе. Если не существует - добавление пользователя в список, возврат к предыдущему экрану с обновленным списком. В другом случае показ пользователю уведомления о том, что введенный номер телефона используется другим пользователем. В случае ввода некорректных данных показ пользователю уведомления что введенные данные неверны
Нажатие на пользователя из списка	Отображение экрана со списком автомобилей данного пользователя
Нажатие на кнопку «Добавить автомобиль»	Отображение экрана с полями для ввода данных об автомобиле пользователя

Продолжение таблицы 1

Выполненное действие	Полученный результат
Повторное нажатие «Добавить автомобиль»	В случае корректности введенных данных происходит проверка существует ли автомобиль с указанным гос. номером в системе. Если не существует - добавление автомобиля в список автомобилей выбранного пользователя, возврат к предыдущему экрану с обновленным списком. В другом случае показ пользователю уведомления о том, что введенный гос. номер присвоен другому автомобилю. В случае ввода некорректных данных показ пользователю уведомления что введенные данные неверны
Нажатие на автомобиль из списка	Отображение экрана со списком заявок на оказание услуг для данного автомобиля
Нажатие на заявку из списка	Отображение экрана с информацией о заявке, в случае если заявка была выполнена – указан сотрудник, выполнявший заявку, самой заявке присвоен статус выполнена. Заявка, требующая подтверждения сотрудника-оператора детейлинг центра, находится в статусе ожидания. Подтвержденная заявка находится в статусе подтверждена. Заявка, которую отменил пользователь находится в статусе отменена.
Нажатие на кнопку «Запросить услугу»	Отображение экрана с полями для выбора услуги и даты оказания услуги
Нажатие на поле ввода «Даты оказания услуги»	Отображение элемента с возможностью выбора даты.
Нажатие на поле ввода «Услуги»	Отображение экрана со списком актуальных услуг детейлинг центра
Нажатие на услугу из списка	Возвращение на предыдущий экран и заполнение поля услуга выбранным элементом
Повторное нажатие «Запросить услугу»	В случае, когда поля заполнены и выбрана дата не раньше сегодняшнего дня – добавление заявки в список и отправка уведомления на телефон о том, что заявка была создана. В случае если услуга не выбрана или введена дата раньше сегодняшнего дня – показ уведомления с информацией о том, что нужно выбрать услугу/выбранная дата не является допустимой.

Результаты, полученные в ходе тестирования разработанного программного продукта, позволяют сделать заключение в том, что разработанная программа соответствует требованиям технического задания.

3 Руководство программиста

В контексте разработки приложения «Детейлинг центр», основным требованием являлось обеспечение эффективной работы с базой данных и визуального отображение этих данных. Для достижения этой цели были выполнены следующие пункты.

1) Создание базы данных: для обеспечения эффективной работы с данными была создана физическая модель базы данных. Разработана SQL-структура базы данных с использованием названия «detail.db». Создан класс для подключения и взаимодействия с базой данных, обеспечивающий необходимые операции чтения и записи.

2) Разработка интерфейса: имеющиеся таблицы базы данных были интегрированы в удобный пользовательский интерфейс. Параметры отображения элементов интерфейса были адаптированы для обеспечения максимального удобства пользователя.

3) Алгоритмы проверки входных данных: написаны алгоритмы для проверки введенных пользователем данных. Реализованы механизмы сопоставления входных данных с данными в базе для исключения возможных ошибок в работе приложения.

4) Основные подпрограммы: подключение к базе данных: разработаны функции для установления связи с базой данных, обеспечивающие безопасное подключение и отключение.

5) Валидация данных: созданы алгоритмы валидации входных данных, гарантируя корректность и безопасность обработки информации.

Программа является полностью функциональной, однако при желании может быть доработана. Например, можно добавить возможность создания напоминания при оформлении заявки.

Класс DatabaseHelper.kt

DatabaseHelper является классом-помощником для работы с базой данных «detail.db». Он обеспечивает создание, обновление и копирование

базы данных, а также предоставляет методы для открытия и закрытия соединения с базой данных.

Методы класса:

`init(context: Context)` - Конструктор класса, инициализирует параметры и проверяет наличие базы данных. При необходимости копирует базу данных из ресурсов приложения.

`updateDataBase()` - Проверяет необходимость обновления базы данных и, при необходимости, копирует новую версию базы данных.

`checkDataBase()` - Проверяет наличие базы данных по указанному пути. Возвращает: `true`, если база данных существует; `false` в противном случае.

`copyDataBase()` - Копирует базу данных из ресурсов приложения, если она отсутствует. Вызывает: `copyDBFile()` для фактического копирования.

`copyDBFile()` - Копирует файл базы данных из ресурсов приложения в рабочий каталог приложения.

`openDataBase()` - Открывает базу данных для чтения и записи.

Возвращает: `true`, если база данных успешно открыта; `false` в противном случае.

`close()` - Закрывает соединение с базой данных.

`onCreate(db: SQLiteDatabase)` - Вызывается при создании базы данных.

`onUpgrade(db: SQLiteDatabase, oldVersion: Int, newVersion: Int)` - Вызывается при обновлении базы данных (изменении версии). Устанавливает флаг `mNeedUpdate` в `true`, если новая версия базы данных выше текущей.

Класс `MainActivity.kt`

`MainActivity.kt` представляет основную активность приложения «Детейлинг центр». Здесь происходит инициализация интерфейса, установка обработчиков событий и запуск дополнительных функций.

Методы класса:

onCreate(savedInstanceState: Bundle?) - Метод, вызываемый при создании активности. Здесь устанавливается макет, инициализируются элементы интерфейса, и задаются обработчики для кнопок.

Класс ServicesActivity.kt

ServicesActivity.kt является активностью приложения «Детейлинг центр» и отвечает за отображение списка актуальных услуг, с возможностью перехода на активность с подробной информацией о выбранной из списка услуге.

Методы класса:

onCreate(savedInstanceState: Bundle?) - Метод, вызываемый при создании активности. Здесь устанавливается макет, инициализируются элементы интерфейса, и задаются обработчики. Вызывает метод updateDB()

updateDB() – Метод для установления соединения с БД, вызывает метод updateList(mDb : SQLiteDatabase)

updateList(mDb : SQLiteDatabase) – Метод для получения данных из БД

Класс SingleServiceActivity.kt

SingleServiceActivity.kt является активностью для отображения информации о выбранной услуге.

Методы класса:

onCreate(savedInstanceState: Bundle?) - Метод, вызываемый при создании активности. Здесь устанавливается макет, инициализируются элементы интерфейса.

Класс UsersActivity.kt

UsersActivity.kt отвечает за отображение списка пользователей и добавление новых пользователей.

Методы класса:

onCreate(savedInstanceState: Bundle?) - Метод, вызываемый при создании активности. Здесь устанавливается макет, инициализируются элементы интерфейса, и задаются обработчики. Вызывает метод updateDB()

updateDB() – Метод для установления соединения с БД, вызывает метод updateList(mDb : SQLiteDatabase)

updateList(mDb : SQLiteDatabase) – Метод для получения данных из БД

onActivityResult(requestCode: Int, resultCode: Int, data: Intent?) – Метод, вызываемый при добавлении пользователя из AddUserActivity.kt. Необходим для обновления списка пользователей после добавления.

Класс AddUserActivity.kt

AddUserActivity.kt нужен для ввода данных о новых пользователях.

Методы класса:

onCreate(savedInstanceState: Bundle?) - Метод, вызываемый при создании активности. Здесь устанавливается макет, инициализируются элементы интерфейса, и задаются обработчики.

validUserInsert(name:String,phone:String,mDb : SQLiteDatabase): Boolean – Метод, вызываемый при клике на кнопку «Добавить пользователя». В нём происходит проверка корректности введённых данных и вызов метода checkExists(phone:String,mDb : SQLiteDatabase)

checkExists(phone:String,mDb : SQLiteDatabase) : Boolean – Метод проверки существует ли пользователь с введённым номером в БД, в случае если пользователя нет в БД возвращает TRUE

Класс SingleUserActivity.kt

SingleUserActivity.kt отвечает за отображение списка автомобилей пользователя и добавление новых автомобилей.

Методы класса:

onCreate(savedInstanceState: Bundle?) - Метод, вызываемый при создании активности. Здесь устанавливается макет, инициализируются элементы интерфейса, и задаются обработчики. Вызывает метод updateDB()

updateDB() – Метод для установления соединения с БД, вызывает метод updateList(mDb : SQLiteDatabase)

updateList(mDb : SQLiteDatabase) – Метод для получения данных из БД

onActivityResult(requestCode: Int, resultCode: Int, data: Intent?) – Метод, вызываемый при добавлении автомобиля из AddAutoActivity.kt. Необходим для обновления списка автомобилей пользователя после добавления.

Класс AddAutoActivity.kt

AddAutoActivity.kt нужен для ввода данных о новых автомобилях.

Методы класса:

onCreate(savedInstanceState: Bundle?) - Метод, вызываемый при создании активности. Здесь устанавливается макет, инициализируются элементы интерфейса, и задаются обработчики.

validAutoInsert(mark:String,model:String,gosnumber:String,mDb : SQLiteDatabase): Boolean – Метод, вызываемый при клике на кнопку «Добавить автомобиль». В нём происходит проверка корректности введённых данных и вызов метода checkExists(gosnumber: String,mDb : SQLiteDatabase).

checkExists(gosnumber: String,mDb : SQLiteDatabase) : Boolean – Метод проверки существует ли автомобиль с введённым гос. номером в БД, в случае если пользователя нет в БД возвращает TRUE.

Класс SingleAutomobileActivity.kt

SingleAutomobileActivity.kt отвечает за отображение списка заявок на оказание услуг и добавление новых заявок.

Методы класса:

onCreate(savedInstanceState: Bundle?) - Метод, вызываемый при создании активности. Здесь устанавливается макет, инициализируются элементы интерфейса, и задаются обработчики. Вызывает метод updateDB()

updateDB() – Метод для установления соединения с БД, вызывает метод updateList(mDb : SQLiteDatabase)

updateList(mDb : SQLiteDatabase) – Метод для получения данных из БД

onActivityResult(requestCode: Int, resultCode: Int, data: Intent?) – Метод, вызываемый при добавлении заявки из AddProvidedServiceActivity.kt. Необходим для обновления списка заявок на оказание услуг после оформления заявки.

Класс AddProvidedServiceActivity.kt

AddProvidedServiceActivity.kt нужен для оформления новых заявок на оказание услуги.

Методы класса:

onCreate(savedInstanceState: Bundle?) - Метод, вызываемый при создании активности. Здесь устанавливается макет, инициализируются элементы интерфейса, и задаются обработчики. В данном методе задаётся обработчик для клика по элементу tiService: editText для вызова активности ChoiceServiceActivity.kt

onActivityResult(requestCode: Int, resultCode: Int, data: Intent?) – Метод, вызываемый при выборе услуги из ChoiceServiceActivity.kt. Необходим для установки в tiService названия выбранной услуги.

validProvidedServiceInsert(service_id: Int, dateOfProvide: Date): Boolean – Метод, вызываемый при клике на кнопку «Запросить услугу». В нём происходит проверка корректности введённых данных.

Класс ChoiceServiceActivity.kt

ChoiceServiceActivity.kt активность для выбора услуги из списка актуальных услуг детейлинг центра.

Методы класса:

onCreate(savedInstanceState: Bundle?) - Метод, вызываемый при создании активности. Здесь устанавливается макет, инициализируются элементы интерфейса, и задаются обработчики. Вызывает метод updateDB()

updateDB() – Метод для установления соединения с БД, вызывает метод updateList(mDb : SQLiteDatabase)

updateList(mDb : SQLiteDatabase) – Метод для получения данных из БД. В нём также устанавливается обработчик клика по услуге в списке.

Класс SingleProvidedServiceActivity.kt

SingleProvidedServiceActivity.kt является активностью для отображения информации о заявке на оказание услуги.

Методы класса:

onCreate(savedInstanceState: Bundle?) - Метод, вызываемый при создании активности. Здесь устанавливается макет, инициализируются элементы интерфейса.

findEmployerFromDB(employer_id:Int):String – Метод, вызываемый при наличии сотрудника, оказывавшего услугу.

4 Руководство пользователя

Рекомендованные системные требования к мобильным устройствам на платформе Android:

1. Операционная система: Android;
2. Процессор: Рекомендуется устройство с процессором не ниже среднего уровня;
3. Оперативная память: Рекомендуется устройство с оперативной памятью не менее 2 ГБ;
4. Место на устройстве: Наличие свободного места на устройстве не менее 100 МБ;
5. Экран: Рекомендуется устройство с экраном не менее 5 дюймов и разрешением не менее 720x1280 пикселей.

Использование основных функций:

- 1) Просмотр подробной информации об услуге:

Перейти в раздел «Посмотреть список услуг» и выбрать интересующую услугу из списка.

- 2) Оформление заявки на оказание услуги:

Перейти в раздел «Пользователи», выбрать пользователя, выбрать автомобиль пользователя, перейти в раздел «Запросить услугу», выбрать услугу для оказания, выбрать дату для оказания услуги, подтвердить запрос услуги.

- 3) Выход из приложения:

Для выхода из приложения используйте стандартные функции управления задачами Android.

Приложение располагается локально на устройстве пользователя, для доступа к приложению не требуется доступ в интернет.

Работа с приложением и основной функционал:

					МИВлГУ 09.03.04 – 0.012	Лист
Изм.	Лист	№ докум.	Подпись	Дата		24

Запуск приложения производится нажатием на иконку приложения. При запуске приложения перед пользователем отывается главная форма приложения, представленная на рисунке 1 в приложении 3. На ней расположены две кнопки: «Посмотреть список услуг», «Пользователи», нажатие на каждую из кнопок переводит пользователя на формы, соответствующие названиям кнопок.

Форма «Список услуг детейлинг центра», представленная на рисунке 2 в приложении 3, отображает список актуальных услуг детейлинг центра. При нажатии услугу произойдёт переход на страницу с информацией об услуге, представленной на рисунке 3 приложения 3.

Форма «Список пользователей приложения», представленная на рисунке 4 в приложении 3, отображает список пользователей приложения. При нажатии на кнопку «Добавить пользователя» произойдёт переход на страницу добавления пользователя, представленную на рисунке 5 в приложении 3. При нажатии на пользователя из списка произойдёт переход на страницу со списком автомобилей выбранного пользователя, представленную на рисунке 6 в приложении 3.

Форма «Список автомобилей пользователя», представленная на рисунке 6 в приложении 3, отображает список автомобилей выбранного пользователя. При нажатии на кнопку «Добавить автомобиль» появится форма добавления автомобиля, представленная на рисунке 7 в приложении 3. При нажатии на автомобиль из списка произойдёт переход на форму со списком заявок оказания услуг для выбранного автомобиля, представленную на рисунке 8 в приложении 3.

Форма «Список заявок оказания услуг», представленная на рисунке 8 в приложении 3, отображает список заявок оказания услуг для выбранного автомобиля. При нажатии на кнопку «Запросить услугу» появится форма для запроса услуги, представленная на рисунке 9 в приложении 3. При нажатии на заявку из списка произойдёт переход на форму с информацией о заявке, представленной на рисунке 12 в приложении 3.

Форма «Запросить услугу», представленная на рисунке 9 в приложении 3, позволяет пользователю оформить заявку на оказание услуги для автомобиля. При нажатии на поле для заполнения услуги произойдёт переход на форму выбора услуги, представленную на рисунке 10 в приложении 3. При нажатии на поле для заполнения даты появится календарь выбора даты, представленный на рисунке 11 в приложении 3.

					МИВлГУ 09.03.04 – 0.012	Лист
Изм.	Лист	№ докум.	Подпись	Дата		26

Заключение

Результатом выполнения курсовой работы по разработке приложений для мобильных операционных систем является приложение «Детейлинг центр», выполненное с использованием среды разработки Android Studio и языка программирования Kotlin. В ходе данной работы были составлены концептуальная, логическая, физическая модели, создано приложение, отвечающее тем требованиям, которые были заложены перед его созданием.

В ходе разработки приложения в Android Studio были успешно пройдены этапы создания пользовательских интерфейсов для взаимодействия с пользователем, реализации логики взаимодействия с базой данных, а также обработки событий в приложении.

Дальнейшее развитие приложения может включать в себя расширение функциональности, оптимизацию пользовательского интерфейса, адаптацию под новые стандарты и технологии в области мобильной разработки.

					МИВлГУ 09.03.04 – 0.012	Лист
Изм.	Лист	№ докум.	Подпись	Дата		27

Список литературы

- 1) Аллен Г. Android. Программирование приложений для смартфонов и планшетов — СПб.: Питер, 2019. — 480 с.
- 2) Бурд Б. Android. Разработка приложений для чайников — СПб.: Питер, 2017. — 416 с.
- 3) Гриффитс Дэвид, Гриффитс Дон Head First. Программирование для Android. 2-е изд. — СПб.: Питер, 2018. — 912 с.
- 4) Дейтел П., Дейтел Х., Уолд А. Android для разработчиков. 3-е изд. — СПб.: Питер, 2016.
- 5) Колисниченко Д.Н. Программирование для Android 5. Самоучитель. — СПб.: БХВ-Петербург, 2015. — 303 с.
- 6) Климов И. Android. Разработка приложений на Java — СПб.: БХВ-Петербург, 2018. — 464 с.
- 7) Соломатин А. Android. Программирование приложений — М.: Издательский дом Вильямс, 2015. — 416 с.
- 8) Чернышев В.Н. Android. Разработка профессиональных мобильных приложений — М.: Издательство Эксмо, 2017. — 352 с.

Приложение 1



Рисунок 1 – Концептуальная модель данных

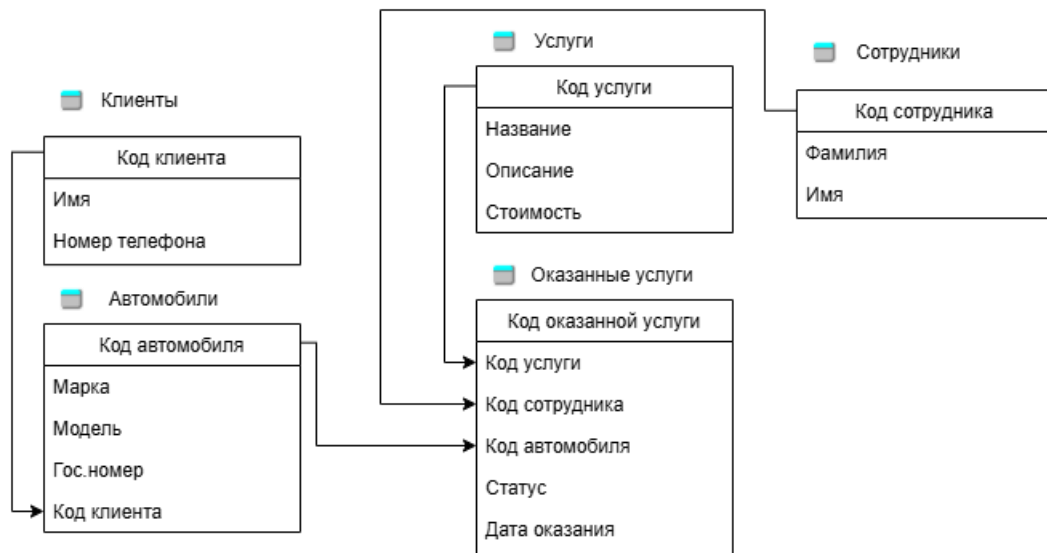


Рисунок 2 – Логическая модель данных

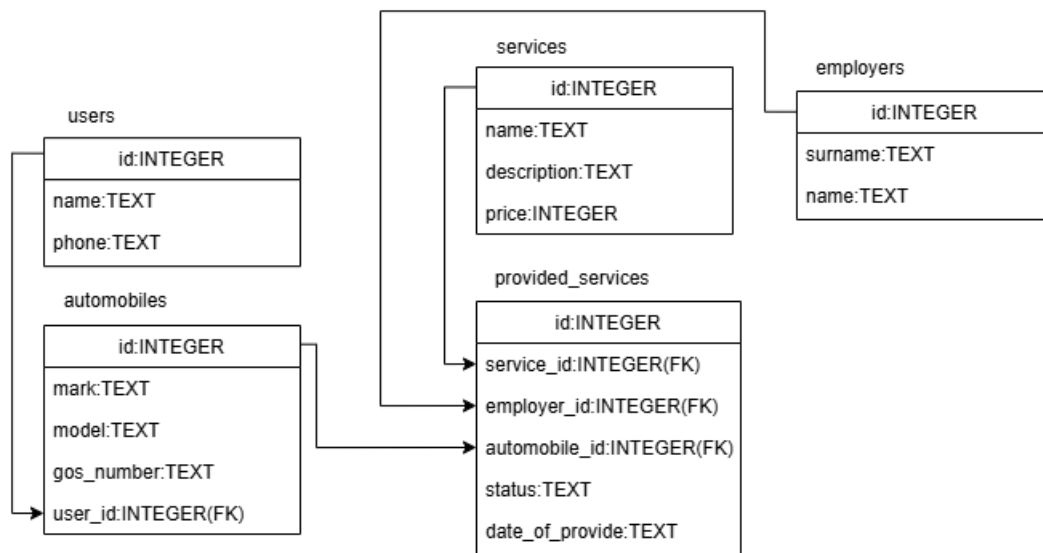


Рисунок 3 – Физическая модель данных

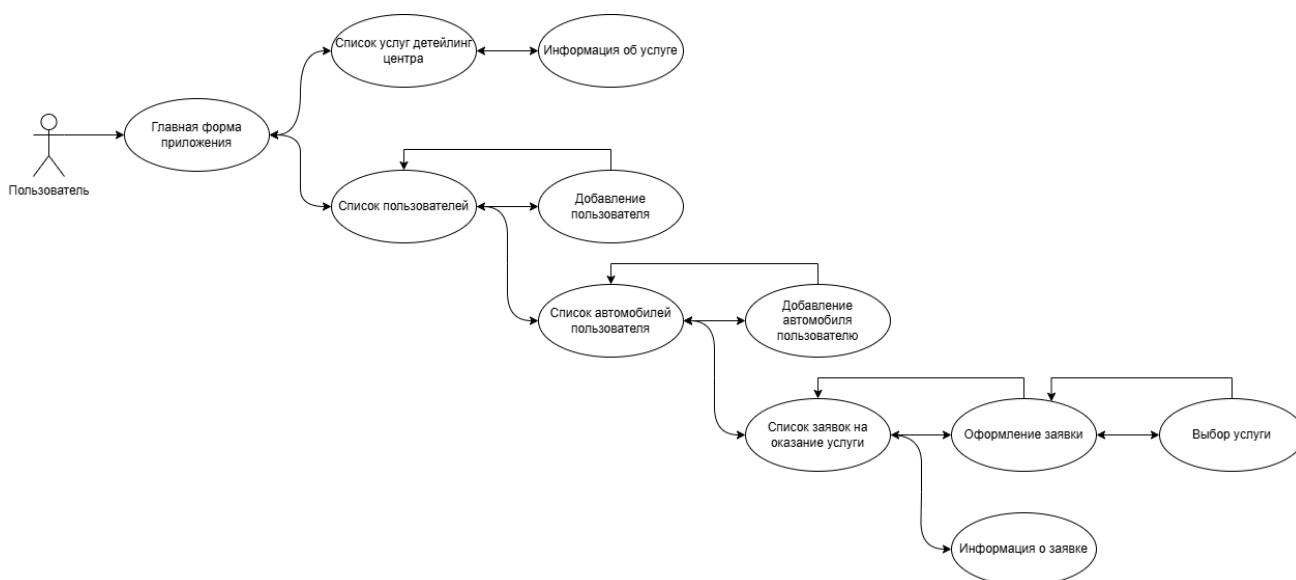


Рисунок 4 – Диаграмма прецедентов «Пользователь»

Приложение 2

В результате выполнения курсовой работы были разработаны классы «Activity» с необходимыми методами и обработчиками событий для корректной работы приложения, макеты «Layout» для отображения интерфейса приложения, макеты «Adapter» для корректного отображения списков в интерфейсе приложения.

При создании программного кода особое внимание акцентировалось на то, что данное приложение является мобильным. В результате чего интерфейс приложения удобен в использовании на любых видах мобильных устройств.

Приложение распространяется в виде apk - файла, с помощью которого оно может быть установлена на мобильное устройство под операционной системой Android.

Для подробного ознакомления с данным приложением можно использовать ссылку на репозиторий данного проекта или прикрепленный QR-код.

Ссылка на репозиторий:

https://github.com/Mejiory/KR_Android



QR-код для перехода на данный репозиторий

					МИВлГУ 09.03.04 – 0.012	Лист
Изм.	Лист	№ докум.	Подпись	Дата		31

Приложение 3



Рисунок 1 – Стартовая страница приложения

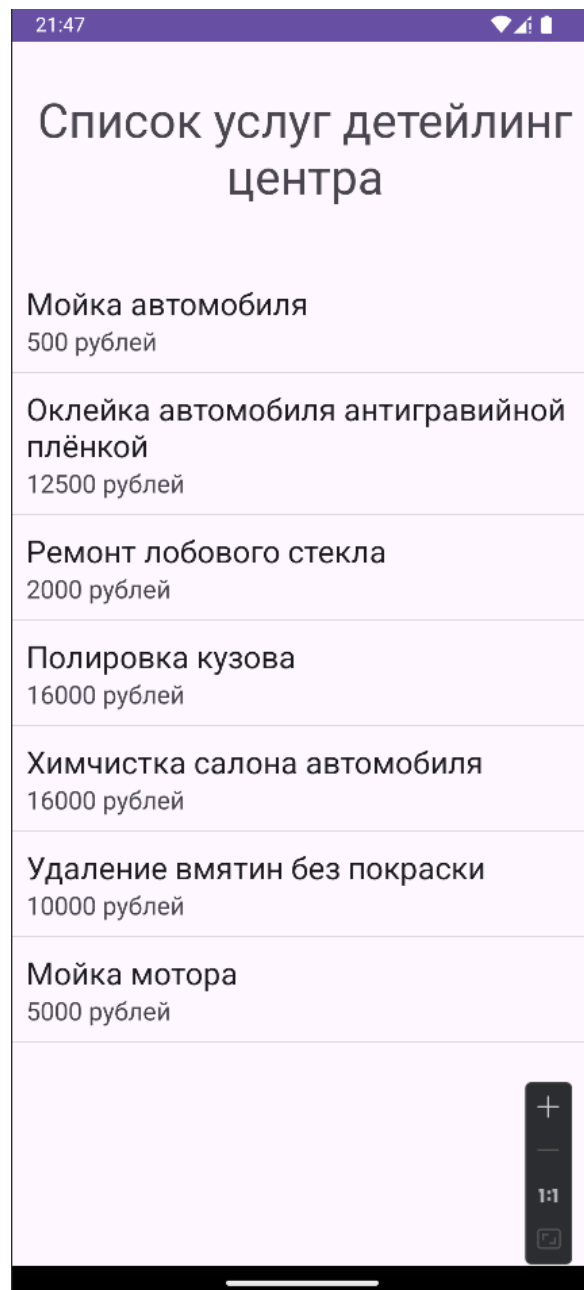


Рисунок 2 – Список услуг детейлинг центра



Рисунок 3 – Страница информации об услуге

					МИВлГУ 09.03.04 – 0.012	Лист
Изм.	Лист	№ докум.	Подпись	Дата		34

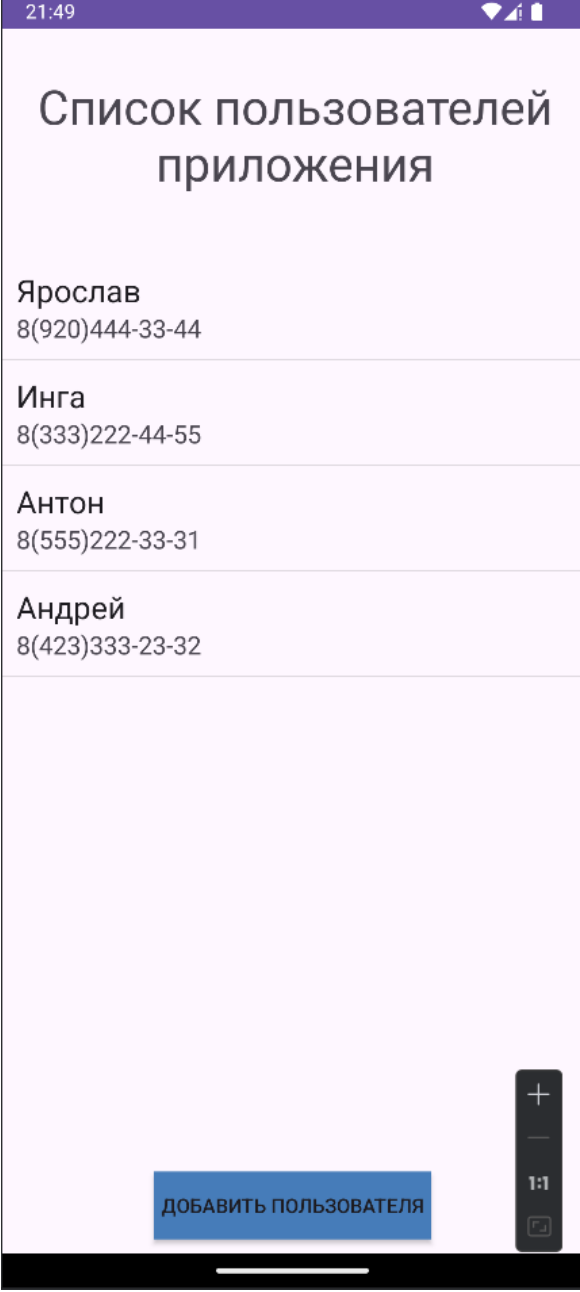


Рисунок 4 – Список пользователей приложения

21:51

Информация о пользователе

Имя пользователя _____

Номер телефона _____

ДОБАВИТЬ ПОЛЬЗОВАТЕЛЯ

+

—

1:1

📄

Рисунок 5 – Страница добавления нового пользователя

					МИВлГУ 09.03.04 – 0.012	Лист
						36
Изм.	Лист	№ докум.	Подпись	Дата		

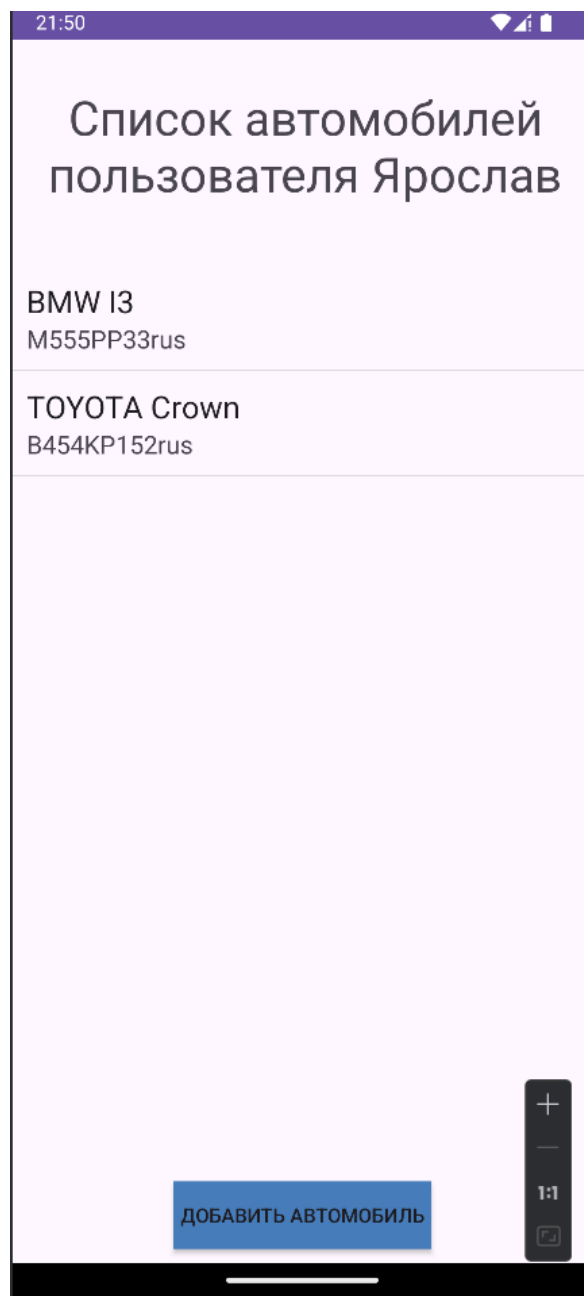


Рисунок 6 – Список автомобилей пользователя Ярослав

21:52

Информация об автомобиле

Марка

Модель

Гос. номер

ДОБАВИТЬ АВТОМОБИЛЬ

+

1:1

Рисунок 7 – Страница добавления автомобиля

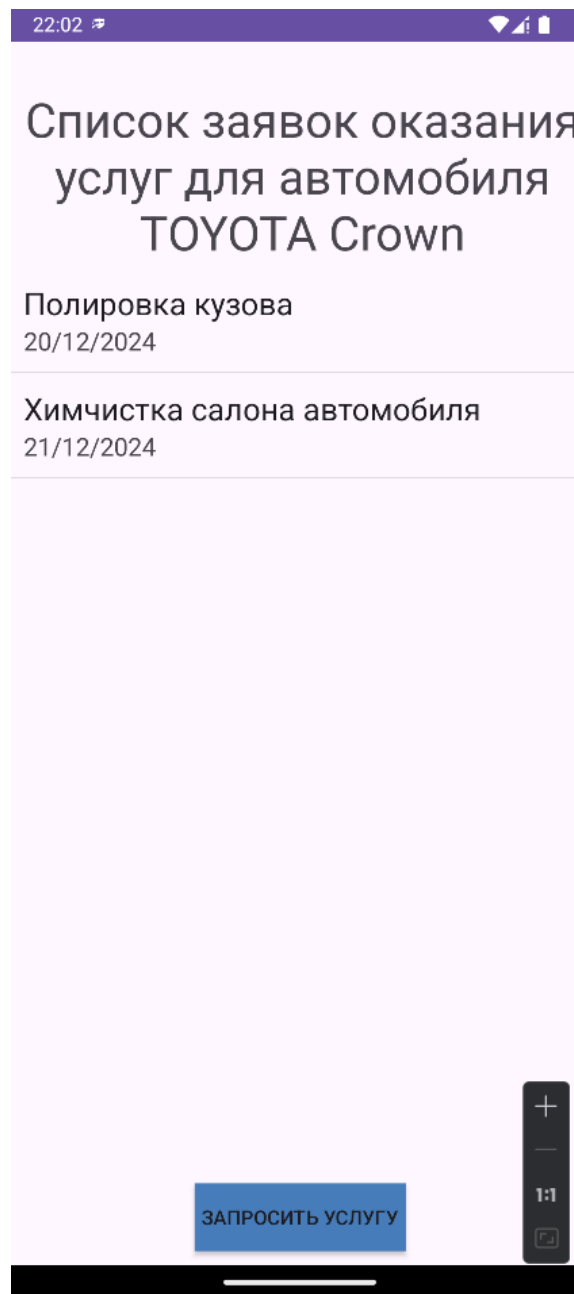


Рисунок 8 – Список заявок оказания услуг для автомобиля

					МИВлГУ 09.03.04 – 0.012	Лист
Изм.	Лист	№ докум.	Подпись	Дата		39

22:03

Запросить услугу

Услуга: _____

Дата оказания услуги: _____

ЗАПРОСИТЬ УСЛУГУ

Рисунок 9 – Страница оформления заявки на оказание услуги

					МИВлГУ 09.03.04 – 0.012	Лист
Изм.	Лист	№ докум.	Подпись	Дата		40

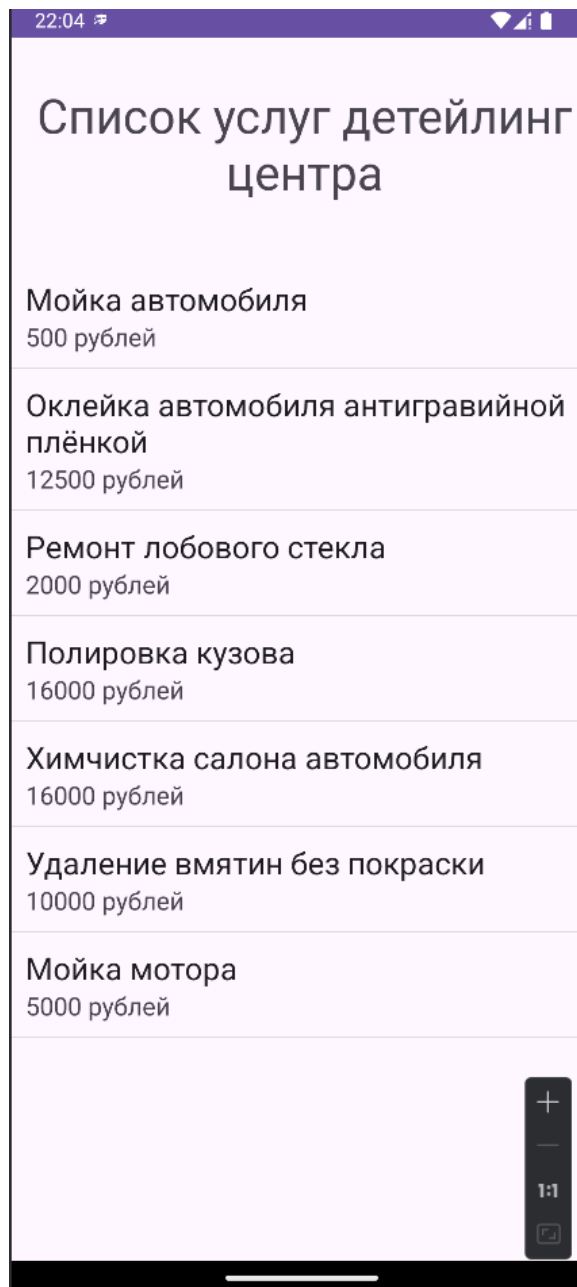


Рисунок 10 – Страница выбора услуги для оказания

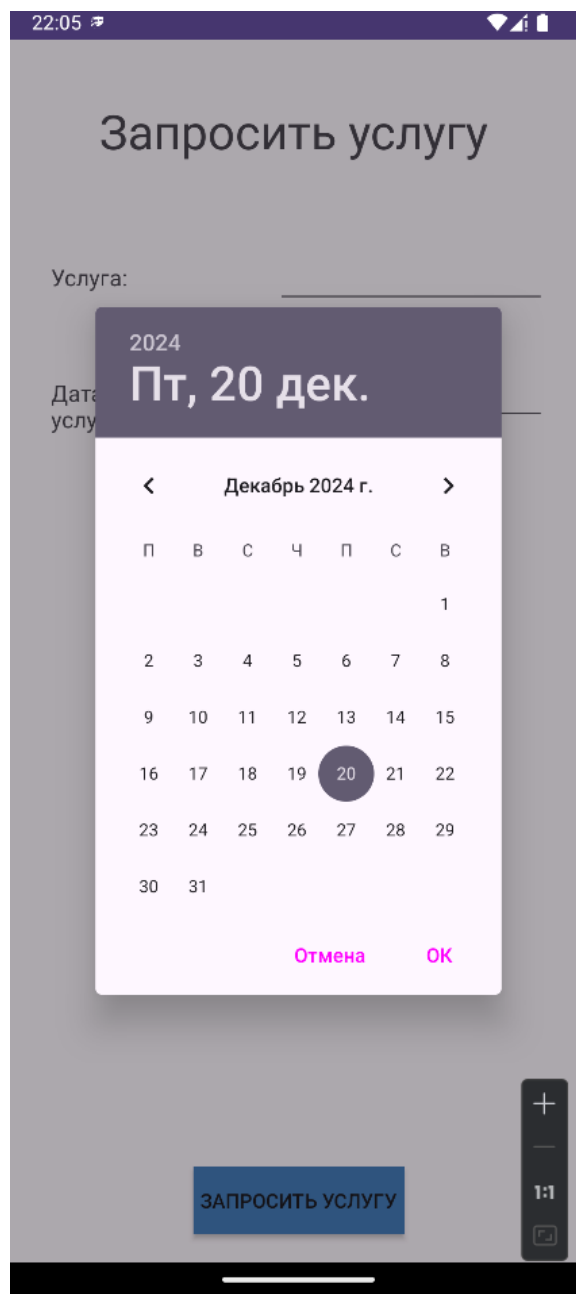


Рисунок 11 – Выбор даты для оказания услуги

					МИВлГУ 09.03.04 – 0.012	Лист
Изм.	Лист	№ докум.	Подпись	Дата		42

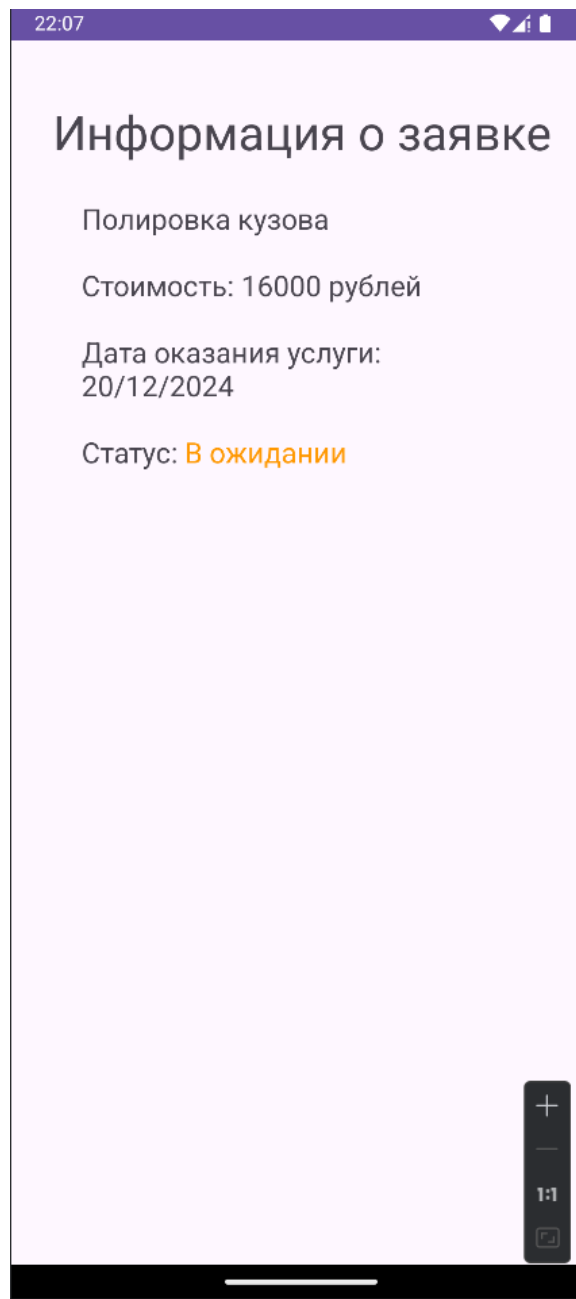


Рисунок 12 – Страница оформленной заявки

					МИВлГУ 09.03.04 – 0.012	Лист
						43
Изм.	Лист	№ докум.	Подпись	Дата		