Министерство образования Пензенской области

ГАПОУ ПО "Пензенский многопрофильный колледж"

Отделение «Архитектура»

Дипломная работа на тему:

**Разработка мобильного приложения**

**для ОС Android «ВДругТакси»**

|  |
| --- |
| **Выполнила:**  Безобразова Ирина Петровна,  студентка группы 11ОА17  **Научный руководитель:**  Феоктистова Татьяна Анатольевна,  преподаватель дисциплин информационного цикла |

Пенза, 2015

Оглавление

[Введение 3](#_Toc421450316)

[Раздел 1. Информационный раздел 5](#_Toc421450317)

[1.1 Городской транспорт и его социальные функции 5](#_Toc421450318)

[1.2 Соединение информационных технологий и транспортной среды 6](#_Toc421450319)

[1.3 Карпулинг 7](#_Toc421450320)

[1.4 Использование инструментов бизнес-моделирования для создания программного обеспечения в качестве бизнес-проекта 9](#_Toc421450321)

[1.6 Этапы разработки программного обеспечения 10](#_Toc421450322)

[Раздел 2. Технологический раздел 13](#_Toc421450323)

[2.1 Анализ предметной области 13](#_Toc421450324)

[2.1 Анализ конкурентов 14](#_Toc421450325)

[2.3 Анализ функциональных требований 16](#_Toc421450326)

[2.4 Методы и средства решения задачи 19](#_Toc421450327)

[2.5 Проектирование пользовательского интерфейса 22](#_Toc421450328)

[2.6 Верстка дизайна 33](#_Toc421450329)

[1.4.1 Кодирование программных средств 33](#_Toc421450330)

[2.7 Тестирование программных средств 34](#_Toc421450331)

[Заключение 43](#_Toc421450332)

[Список используемой литературы 46](#_Toc421450333)

[Список сокращений 47](#_Toc421450334)

# Введение

Автомобильный транспорт является одним из важных факторов обеспечения жизнедеятельности города, базовой инфраструктурой экономического роста города и значимым фактором повышения уровня жизни. Его устойчивое, сбалансированное и эффективное развитие должно обеспечить растущий спрос на транспортные услуги.

Согласно Транспортной стратегии Российской Федерации на период до 2030 года, утвержденной распоряжением Правительства Российской Федерации от 22.11.2008 №1734-р, общественный транспорт должен оказывать качественные и доступные для населения услуги пассажироперевозок в соответствии с социальными стандартами.

Достижение данной цели означает удовлетворение в полном объеме растущих потребностей населения в перевозках, обеспечение устойчивой связи микрорайонов города с социально значимой магистральной транспортной сетью.

Одной из важнейших проблем современных городов является обеспечение транспортного обслуживания и организация городского движения. Города стремительно развиваются, и поэтому все более высокие требования предъявляются к мощности транспортных сетей, скорости и регулярности движения.

Мировой опыт и отечественная практика свидетельствуют, что важнейшей проблемой любых реформ является формирование эффективных организационных и экономических механизмов, позволяющих повысить защищенность человека. Для решения проблемы отсутствия достаточного количества транспорта и загруженности автодорог можно использовать систему *карпулинга –* совместное использование частного автомобиля с помощью онлайн-сервисов для поиска попутчиков. Такие системы мало затратны, просты в разработке, внедрении и применении.

Сложившаяся транспортная обстановка в Пензе не всегда позволяет в назначенный срок горожанам добираться до места назначения, что приводит к стрессу и снижению работоспособности. Повышение безопасности транспортного обслуживания населения - **актуальная проблема** в настоящее время для жителей нашего города, решение которой мы видим в необходимости разработки мобильного приложения «ВдругТакси».

**Цель работы:** разработка мобильного приложения для совершенствования организационно-методического обеспечения процессов управления транспортными средствами в городе Пенза, которое дает возможность комфортно и быстро добраться в пункт назначения.

Чтобы достичь поставленной цели, были поставлены следующие **задачи**:

- анализ необходимости разработки приложения;

- изучение и применение современных инструментов бизнес-моделирования для создания программного обеспечения;

- разработка мобильного приложения.

**Объект исследования:** процесс перевозки пассажиров перевозчиками, основанного на технологии карпулинга.

**Предмет исследования:** реализация способа доставки пассажиров согласно им потребностям с помощью мобильного приложения.

**Научная новизна** состоит в использовании нового метода обеспечения процессов управления транспортными средствами в Пензенском регионе.

**Практическая значимость.** Данный проект будет способствовать:

- снижению нагрузок на общественный транспорт;

- экономии средств на поездки для водителя и пассажира;

- развитию социальной активности населения.

# Раздел 1. Информационный раздел

## 1.1 Городской транспорт и его социальные функции

Транспортная стратегия должна определять активную позицию государства по созданию условий для социально-экономического развития, прежде всего в целях повышения качества транспортных услуг, снижения совокупных издержек общества, зависящих от транспорта, повышения конкурентоспособности отечественной транспортной системы, усиления инновационной, социальной и экологической направленности развития транспортной отрасли.

Выбор направлений развития транспортной системы базируется на проекте концепции долгосрочного социально-экономического развития Российской Федерации на период до 2020 года, бюджетных посланиях Президента Российской Федерации Федеральному Собранию Российской Федерации, а также на широком спектре документов, определяющих перспективные направления развития общества и экономики России, ее регионов, транспортной системы страны в целом и отдельных видов транспорта (включая трубопроводный), международной транспортной интеграции, прежде всего в рамках СНГ и ЕврАзЭС, на законодательных и иных нормативных правовых актах в области обороны и национальной безопасности Российской Федерации.

Городской общественный пассажирский автомобильный транспорт занимает особое место в общественной жизни. Наряду со здравоохранением, образованием, розничной торговлей, жилищно-коммунальным хозяйством, сферой организации досуга, он входит в комплекс отраслей социальной инфраструктуры, связанных с воспроизводством рабочей силы и жизнедеятельностью населения.

Автомобильный транспорт выполняет ряд важнейших социальных функций и оказывает заметное влияние на экономическое и социальное развитие административно-территориальных единиц, способен выступать как средство ускорения или замедления развития городов. Общественный транспорт предназначен для перевозок групп населения с малым доходом, обеспечения территориальной целостности городов, доступности всех элементов городского хозяйства.

Практика функционирования городского общественного пассажирского автомобильного транспорта в Пензенском регионе показывает, что общественного транспорта не достаточно, чтобы обеспечить оптимальное удовлетворение интересов горожан, при этом слабо формируются условия для возникновения эффективной конкурентной среды.

## 1.2 Соединение информационных технологий и транспортной среды

Современный уровень развития информационного общества характеризуется изменениями представлений о получении и использовании информации человеком во всех сферах его жизнедеятельности. Интенсивное развитие информационных и коммуникационных технологий привело к их широкому распространению и применению в различных организациях, общественных местах, на транспорте, в космосе. Получение, передача и обработка информации происходит постоянно, ежесекундно, во всех направлениях. Использование информационных и коммуникационных технологий распространяется на все возрастные категории людей без исключения. Нынешнее молодое поколение нашей страны – это первое поколение, которое выросло в условиях высокого уровня развития компьютерной и мобильной техники. Информационные технологии, включающие технические средства, предназначенные для получения, обработки, накопления и передачи информации, а также программные средства, используемые в данных технических устройствах, постоянно развиваются, заставляя осваивать все новые и новые направления переработки информации. Современные люди живут в новой среде, именуемой информационным пространством и важно, чтобы это существования было комфортным.

Мы живем в век информационных технологий, которые также проникли и в мобильные устройства. Количество задач, решаемые с помощью смартфонов, с каждым годом растет. Они автоматизируют как бизнес-процессы, так и решают социальные проблемы.

Число пользователей мобильных устройств также устойчиво растет. И, как следствие, повышается необходимость в  разработке мобильных приложений различных категорий. Сегодня на рынке мобильного программного обеспечения преобладают нативные приложения, т. е. приложения, разработанные для конкретной платформы. Самыми распространенной мобильной платформой является Android. Смартфоны на Android занимают 70% от всего доли рынка смартфонов.

На смену современной парадигме идет качественный высокотехнологичный общественный транспорт, комфортная информационная среда, когда человек с помощью смартфона управляет своими поездками. Он вписывает свой режим в режим общественного транспорта, у него удобное билетно-тарифное меню, позволяющее ему делать пересадки без потери денег.

## 1.3 Карпулинг

Ввиду отсутствия достаточного количества муниципальных маршрутов движения, решением вопроса по перевозкам людей может являться система **карпулинга** (от [англ.](https://ru.wikipedia.org/wiki/%D0%90%D0%BD%D0%B3%D0%BB.) *car*— машина + *pool* — объединение) — это совместное использование частного автомобиля с помощью онлайн-сервисов поиска попутчиков.

Движение карпулинга достаточно популярное в Европе и в России, набирает оно силу и в Пензе: существуют объединения граждан в социальных сетях, с помощью которых они реализуют систему карпулинг (<http://vk.com/companion58>). Это движение не только дешевая альтернатива такси, но и один из методов борьбы с пробками.

Осуществление перевозок пассажиров автотранспортом удовлетворяет как индивидуальную потребность человека по его перемещению, так и выполняет социально значимую функцию. В результате перемещения автотранспортом людей достигается социально полезный эффект, который заключается в нормальном функционировании экономики, обеспечении работы государственных органов, жизнедеятельности человека.

При совместном использовании автомобиля можно сэкономить ощутимое количество денег, т.к. между участниками совместных поездок стоимость проезда делится в равных долях. Чем больше попутчиков, тем меньше водитель заплатит за топливо и дешевле поездка.

Поездки планируется заблаговременно или спонтанно, могут быть регулярными или разовыми**.** Карпулинг дает ощутимые преимущества:

- он позволит сократить количество автомобилей в городе, а значит и объем вредных выбросов в окружающую среду;

- решается злободневная проблема городских пробок и нехватки парковочных мест***;***

- совместные поездки помогают заводить новые знакомства, они гораздо дешевле и веселее. Карпулинг формирует позитивные качества гражданина.

Чередуясь по дням, участники carpool могут по очереди брать на себя роль водителя и пассажирам во избежание стресс и нагрузок на вождение особенно при поездках на далекие расстояния.

С карпулом можно забыть о всех проблемах с общественного транспорта (отсутствие, опоздание, давка) , что положительно скажется на эффективности работы и настроения!

При частых поездках, например, по городу карпул позволит сэкономить время в пути, и его можно с пользой тратить на общение с близкими.

Такой метод передвижения может подойти всем жителям г. Пензы, но сейчас он больше востребован молодыми людьми различных социальных групп и ему свойственна своя особая культура - она наполнен романтикой, приключенческим духом, легкостью восприятия дорог и стремлением получить удовольствие не от благ цивилизации, а от человеческого общения и созерцания новых мест.

## 1.4 Использование инструментов проектирования для создания программного обеспечения

В современной индустрии IT-разработки, чтобы повысить качество конечного продукта, используют инновационные инструменты для создания программ. Более того их применение определяет то, насколько эффективна будет дальнейшая работа и итоговый результат, становится возможным уменьшить количество времени и денег, уходящих на разработку программного обеспечения.

Одним из таких инструментов является методика разработки ПО «LeanStartUp». Она предполагает продуманное проектирование программы, чтобы она продуктивно решала заложенные в нее задачи.

Ключевые принципы Lean Startup:

* обобщение гипотез создателей в виде [канвы бизнес-модели](https://ru.wikipedia.org/wiki/%D0%9A%D0%B0%D0%BD%D0%B2%D0%B0_%D0%B1%D0%B8%D0%B7%D0%BD%D0%B5%D1%81-%D0%BC%D0%BE%D0%B4%D0%B5%D0%BB%D0%B8) - это схематическое изображение того, как компания создает ценность для себя и своих клиентов;
* использование методики [гибкой разработки](https://ru.wikipedia.org/w/index.php?title=%D0%93%D0%B8%D0%B1%D0%BA%D0%BE%D0%B9_%D1%80%D0%B0%D0%B7%D1%80%D0%B0%D0%B1%D0%BE%D1%82%D0%BA%D0%B8&action=edit&redlink=1) продукта - это процесс, при котором стартапы создают минимальные жизнеспособные продукты, которые они затем тестируют.

Работы по проектированию осуществляется в несколько этапов.

## 1.5 Этапы разработки программного обеспечения

Этапы разработки любого ПО называются его жизненным циклом – это период времени, который начинается с момента принятия решения о необходимости создания программного продукта и заканчивается в момент его полного изъятия из эксплуатации. Этот цикл — процесс построения и развития ПО.

В разработке мобильного приложения была применена гибкая методология разработки ПО «Scrum», основанной на спиральной модели ЖЦ. При использовании этой модели ПО создается в несколько [итераций](https://ru.wikipedia.org/wiki/%D0%98%D1%82%D0%B5%D1%80%D0%B0%D1%86%D0%B8%D1%8F) (витков спирали) [методом созданием](https://ru.wikipedia.org/wiki/%D0%9F%D1%80%D0%BE%D1%82%D0%BE%D1%82%D0%B8%D0%BF%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D0%B5_%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%BD%D0%BE%D0%B3%D0%BE_%D0%BE%D0%B1%D0%B5%D1%81%D0%BF%D0%B5%D1%87%D0%B5%D0%BD%D0%B8%D1%8F) прототипов.

Каждая итерация соответствует созданию фрагмента или версии ПО, на ней уточняются цели и характеристики проекта, оценивается качество полученных результатов и планируются работы следующей итерации. Таким образом, было создано 3 прототипа приложения, каждое из которых протестировано, проанализировано и намечены планы на улучшение прототипа. Весь процесс разработки включал в себя нижеприведенные этапы.

* 1. Первоначальный анализ необходимости разработки приложения: определение цели и задач разработки ПО (приложение № «Бизнес-моделирование по канвас»); анализ целевой аудитории (приложение №); анализ конкурентов (приложение №).

2) Протопирование: словесное описание работы приложения; создание визуального прототипа. На прототипе показаны все функциональные элементы приложения, где можно проследить механизм работы программы. Версия последнего прототипа:

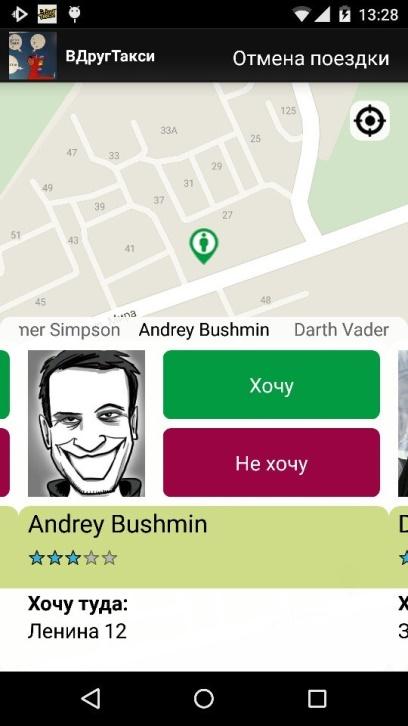
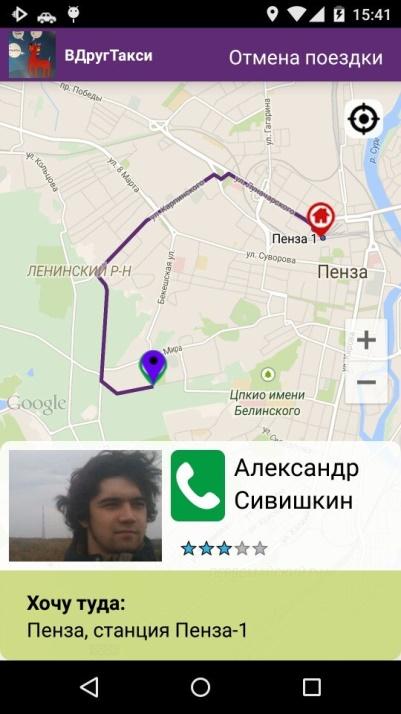
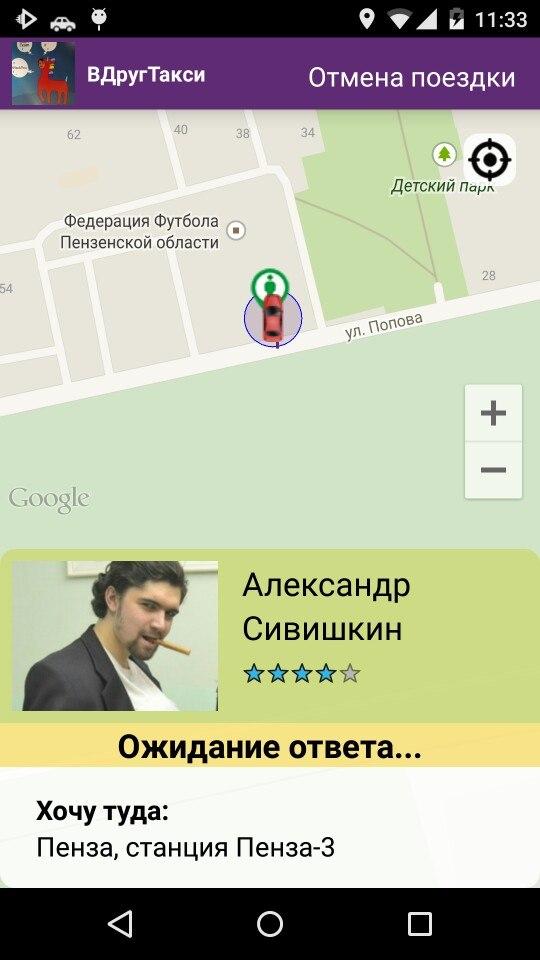
  

Рисунок – Версия последнего прототипа

Пользоваться приложением очень просто, достаточно: 1) установить приложение; 2) авторизоваться через социальную сеть (ВК/Facebook); 3) выбрать роль пассажира или водителя; 4) указать адрес пункта назначения; 5) выбрать походящего попутчика из предложенных; 6) связаться и встретится с попутчиком.

3) Создание имиджа проекта: дизайн и название приложения. Протестированный и измененный прототип дальше переходит к дизайнеру, и разработчики вместе с ним разрабатывают дизайн.

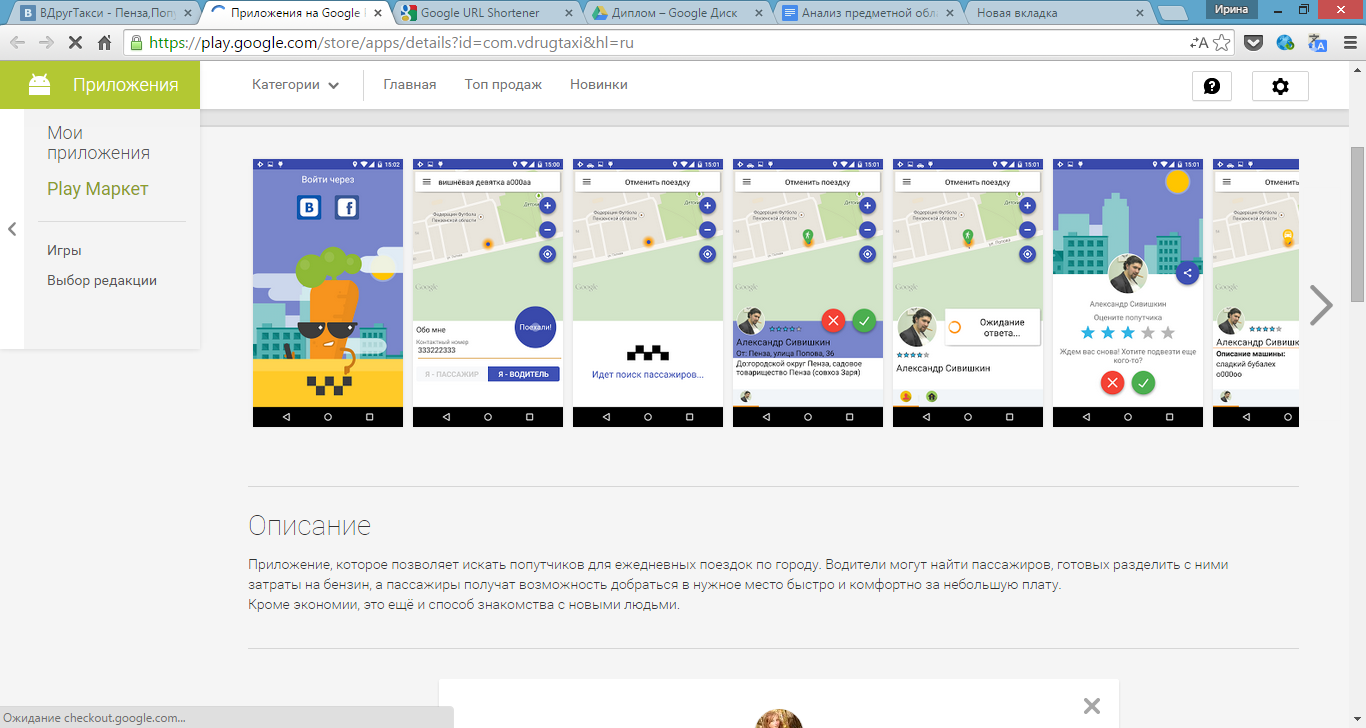


Рисунок – версия последнего дизайна

Название «ВДругТакси» состоит из 3-х слов:

* Вдруг – что означает механизм работы приложения «здесь и сейчас»;
* Друг – попутчики становятся интересными собеседниками и друзьями;
* Такси – можно подвезти попутчика и заработать немного денег.

4) Программирование. Полученный дизайн переходит к разработчику, который превращает его в код. Разработка ведется с использованием языка программирования Java под мобильную платформу Android. Процесс программирования делится на две части: клиентскую и серверную. Клиентская часть отвечает за работу интерфейса, за сам интрефейс и выполнение функций приложения. Серверная часть отвечает за связь с сервером, обрабатывающим запросы клиентов и обеспечивающим работу с базой данных.

В свою часть процесс программирования тоже делится на несколько этапов, которые описаны в следующем разделе.

5) Тестирование приложения. Тестирование происходит во время каждой итерации, неразрывно с процессом программирования. Каждая ново разработанная функция тестируется на ошибки и удобство использования. Законченная программа в свою очередь тестируется всей командой, но уже в реальных условиях. По итогам теста идет обсуждение, как приложение можно изменить и улучшить, сделать его использование более удобными.

6) Сдача проекта. Приложение протестировано и отлажено. Выполняет все заявленные в нем функции. Программа опубликована в Play Store (магазин Андроид-приложений) по ссылке: <http://goo.gl/GqFJZL> .

7) Продвижение. Для рекламы использовались следующие каналы продвижения: социальные сети, телевидение, радио, новостные порталы, блог-встречи. Основная работа ведется через онлайн-маркетинг, который включал в себя рекламу в социальных сетях, работу с популярными сообществами и тематическими партнерами.

## 1.5 Анализ конкурентов

В настоящее время существует множество приложений по поиску попутчиков. Они были заимствованы от западных стран, где пользуются популярностью.

Самым распространенным в России считается приложение “BlaВlaСar”  (http://www.blablacar.ru/).  Оно рассчитано на междугородние поездки, которые планируются заранее.  Другие приложения не настолько популярны, и их можно найти в Google Play.  К таким программам можно отнести: “Автостоп 2.0”, “Попутчик”, “HopIn”, “ВПути”, “GiveMeALift”. Выбранные приложения будут рассматриваться как конкуренты. Был проведен анализ сильных и слабых сторон программ, чтобы создать конкурентное преимущество разрабатываемого продукта.  Результат исследования приведены в таблице 1.

Таблица 1 – Сравнение разрабатываемого приложения с конкурентами

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  | ВДругТакси | BlaBlaCar | Автоcтоп 2.0 | Попутчик | HopIn | ВПути | GiveMeALift |
| Перемещение внутри города | + | - | + | + | + | + | + |
| Предзаказ | - | + | - | + | - | - | - |
| Кроссплатформенность | + (предполагается) | + | - | - | + | - | - |
| Сложность регистрации | - | - | - | - | - | + | - |
| Сложность использования | - | - | - | + | - | - | + |
| Соц. профили | + | + | + | + | + | - | + |
| Рейтинговая система | + | + | + | + | + | - | - |
| Удобство  интерфейса | + | + | + | - | + | - | - |
| Русский язык | + | + |  |  |  | + | - |
| Популярность в Пензе | (предполагается) | + | - | - | - | - | - |

Приложение “ВДругТакси” прост в использовании, так как имеет удобный и понятный интерфейс. Легко зарегистрироваться, потому что использует авторизацию через социальные сети. Рейтинговая система позволяет отслеживать надежность попутчиков.

Все приложения-конкуренты реализуют поездку от точки до точки, т.е. и пассажир и водитель должны вводить полный маршрут поездки, а программа подбирает подходящие попутчикам маршруты. В разрабатываемом приложении было решено отказаться от такого варианта, так как маршрут водителя может измениться, и он поедет другой дорогой. Поэтому в приложении водителю предлагаются ближайшие пассажиры по пути его следования.

Это и является главным отличием разрабатываемого приложения от других - система моментальных поездок, здесь и сейчас. Водитель и пассажир получают попутчика сразу как вошли в программу. Поэтому в данном приложении нет необходимости в планировании поездок и готовых маршрутов.

А в сочетании с концепцией простого интерфейса “в один клик”, который позволяет создать поездку в наименьшее количество этапов, приложение составляет мощную конкуренцию остальным продуктам.

В данном исследовании видно, что  приложение “ВДругТакси” имеет сильное преимущество перед конкурентами: сочетание концепции “здесь и сейчас” и “в один клик”. Данное объединение позволяет легко и быстро создать поездку и встретится с попутчиком сразу после запуска приложения. А в тенденциях требований современного пользователя: все упрощать. Более того, конкурентные приложения малоизвестны в Пензе, а «ВДругТакси» предполагается массово продвигать.

# Раздел 2. Технологический раздел

Разработка любого программного обеспечения является очень сложным и многоэтапным процессом. Каждый из ее этапов является очень важным и помогает сделать программу более продуманной и эффективной.

Создание сложной и хорошей программы — это процесс, объединяющий специалистов в различных областях. Проектирование определяет то, насколько эффективна будет дальнейшая работа и итоговый результат.

При помощи стандартизации становится возможным обеспечить качество программного обеспечения и уменьшить количество времени и денег, уходящих на его разработку.

Он занимает продолжительное время, поэтому разработку программы делят на определенные этапы.

Необходимо с самого начала обозначить цели проекта, разузнать, что нужно целевой аудитории, регламентировать структуру и логику взаимодействия программы c пользователями.

При соблюдении этого, Ваши разработки предоставят Вам качественную и организованную программу, с помощью которой можно будет продуктивно решать любые задачи.

деление на этапы является грамотным потому – что

принцип построения взаимодействия приложения с пользователем

## 2.1 Построение архитектуры приложения

Промоделируем ситуацию осуществления системы карпулинга с помощью мобильного приложения. Основными действующими лицами в carpool являются владелец личного автомобиля (водитель) и человек без автомобиля, которому нужно добраться из одной части города в другую (пассажир). Оба являются владельцами смартфонов с установленным приложением.

С помощью приложения водитель должен узнать, кого он может подвезти, а пассажир, кто может подвезти его. Эту задачу можно возложить на социальные сети, в которых люди оставляют достаточно информации о себе.

Для организации работы приложений требуется сервер. Приложение должно связываться с сервером и передавать ему свое состояние (координаты) используя технологию Rest API. Для взаимодействия “на ходу” клиентов и сервера, сервер должен отсылать push-уведомления, на которые клиенты приложения и должны реагировать.

Исходя из особенностей предметной области, был составлен словарь терминов:

**Водитель** – автомобилист, желающий общения и подвезти кого–либо.

**Пассажир** – автостопщик, ищущий попутный автомобиль.

**Клиент** **– водитель, Клиент – пассажир** – мобильные устройства, с установленным приложением “ВДругТакси”.

**Социальные сети** – онлайн-ресурсы для социальных взаимодействий.

**Push - уведомления** – распространяемая информация от сервера клиенту, используется для взаимодействия с клиентами (Водителями и Пассажирами).

**Сервер** - удаленный компьютер с программным обеспечением, обрабатывающим запросы клиентов, и обеспечивающим работу с базой данных.

**Rest API** - API, отвечающее принципам REST (Representational State Transfer — репрезентативная передача состояния) – средство взаимодействия клиентов (водителей и пассажиров) с сервером, путем передачи своих состояний.

На основе терминов предметной области, представленных в словаре, была разработана инфологическая модель предметной области в нотации UML 2.0, которая включает в себя 8 сущностей. Полученная модель предметной области представлена на рисунке 1.

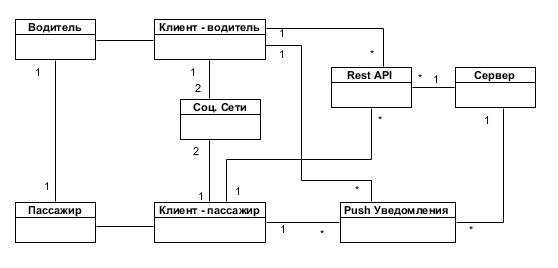


Рисунок 2.1.1 – Модель предметной области

Каждый пользователь выбирает роль водителя или пассажира. Водители и пассажиры используют приложения клиенты. Приложения взаимодействуют с социальными сетями (Вконтакте, Facebook) а также с сервером через REST API, передавая свои состояния. Сервер, исходя из состояний пассажиров и водителей подбирает их друг-другу и взаимодействует с ними, путем отправки push-уведомлений.

## 2.2 Анализ функциональных требований

Основными функциональными к приложению требованиями являются:

1)    Авторизация, через социальные сети.

2)    Заполнить минимальное количество данных для создания поездки.

3)    Начать поездку

4)    Ожидать попутчиков

5)    Выбрать попутчика, когда он появляется

6)    Ехать с попутчиком

7)    Оценить попутчика после поездки для дальнейших пользователей

8)    Отменить поездку в любой момент после ее начала

Создание программных средств позволит автоматизировать поиск попутчиков для водителя и пассажира.

С точки зрения системы можно выделить пользователя приложения – обычного пользователя, у которого есть мобильное устройство (телефон или планшет) с операционной системой Android 4.0 и выше, установлено приложение и есть выход в интернет.

Варианты использования, кратко рассматриваются в таблице 3.

Таблица 3 – Основные варианты использования системы

|  |  |  |
| --- | --- | --- |
| **Действующее лицо** | **Цель** | **Краткое описание** |
| Пользователь | Авторизация | Пользователь должен пройти авторизацию через соц. сети (ВКонтакте, Facebook) |
| Заполнить данные | Перед созданием поездки пользователь выбирает роль (Пассажир/Водитель), заполняет номер телефона (По желанию) и заполняет требуемый для роли минимум данных |
| Начать поездку | Пользователь создает поездку |
| Ожидать попутчика | Пользователь ожидает попутчика |
| Выбрать попутчика | Пользователь на основе данных соц сетей, рейтинга, и положения других пользователей выбирает попутчика |
| Ехать с попутчиком | Для встречи с попутчиком пользователь может позвонить ему.  От пассажира требуется обозначить начало и конец поездки с водителем. |
| Оценить попутчика | Пользователь выставляет оценку попутчику для дальнейших пользователей |
| Отменить поездку | Если в какой то момент пользователь решает не ехать, он отменяет поездку. |

На основе вариантов использования можно выделить общую схему взаимодействия водителя и пассажира для одной поездки, изображенной на рисунке ниже.

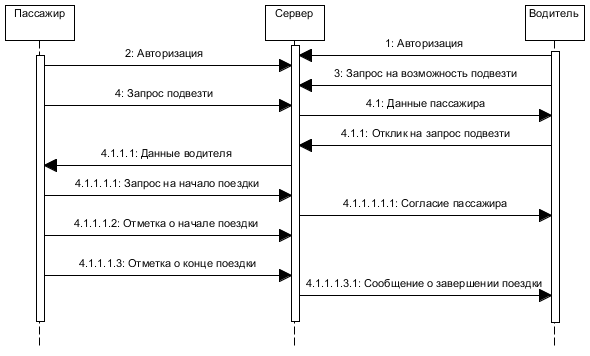


Рисунок 1.3 – Диаграмма последовательности “Общая схема взаимодействий водителя и пассажира для одной поездки”

Как видно из представленной диаграммы, можно выделить поездку для одного водителя и пассажира. Взаимодействие пассажира и водителя проходит, через сервер. Каждый водитель и пассажир должны быть авторизированы на сервере. У водителя поездка начинается с запроса, на возможность подвезти. У пассажира поездка начинается с запроса подвезти его. Далее водитель должен получить с сервера данные о пассажире и, выбрав данного пассажира, должен отправить отклик на запрос подвезти. После того, как водитель откликнулся, пассажир должен получить его данные и, выбрав водителя, должен отправить запрос на начало поездки.

После того, как пассажир согласится на поездку, водитель должен получить согласие от пассажира. Теперь, водитель должен забрать пассажира, а пассажир отметить начало поездки. После поездки, пассажир отмечает ее окончание, которое отсылается водителю.

Таким образом, сервер организует безопасное взаимодействие пассажира и водителя во время поездки.

## 2.3 Методы и средства решения задачи

Было решено разрабатывать приложение для операционной системы Android, поскольку она является наиболее распространённой ОС для смартфонов.

В 86 % смартфонов, проданных во втором квартале 2014 года, была установлена операционная система Android. При этом за весь 2014 год было продано более 1 миллиарда Android-устройств.

Прозрачность архитектуры приложения для пользователя означает, что пользователю не нужно ничего знать о том, как оно устроено. В случае если будет необходимо добавить дополнительный сервер баз данных или заменить компьютеры на более мощные, это будет очень легко сделать. При этом пользователь ничего не узнает о подобных изменениях.

Определившись с видом разрабатываемой системы, необходимо выбрать средства для ее реализации.

В серверной части требуется реализовать сервер, который будет получать данные от клиентов, содержать базу данных пользователей, подбирать попутчиков и рассылать push-уведомления. Для создания и функционирования БД была выбрана СУБД H2, открытая кроссплатформенная СУБД, полностью написанная на языке Java. Возможности этой СУБД полностью удовлетворяют поставленные задачи. Для реализации серверного приложения был выбран Spark framework – простой и легкий framework для быстрой разработки. Для доставки push-уведомлений был выбран сервис Google Cloud Messaging.

Для взаимодействия клиентов и сервера должна быть разработаны API, которые должны удовлетворять принципам REST. Использование принципов REST должно унифицировать интерфейс, и упростить создание, отправку и обработку запросов, как на сервере, так и на клиенте.

Для отправки push-уведомлений от сервера клиентам выбрана технология Google Cloud Messaging. Данная технология разработана компанией Google, и позволяет отсылать на мобильные устройства push-уведомления, в условиях изменяющееся ip адреса на данных устройствах.

Язык Java активно используется для создания мобильных приложений под операционную систему Android. При этом программы компилируются в нестандартный байт-код, для использования их виртуальной машиной Dalvik. Для такой компиляции используется дополнительный инструмент, а именно Software Development Kit, разработанный компанией Google.

Для того чтобы пользователям было удобнее находить друг друга, в приложении должна использоваться карта. В качестве карты были выбраны карты Google Maps, использующие Google Maps Android API v2. Это обусловлено тем, что на большинстве Android устройств находятся данные карты, что упрощает их использование для пользователей.

Конечный интерфейс приложения должен имитировать Material Design, разработанный компанией Google. Так как на момент разработки устройств, поддерживающий работу данного дизайна еще мало, то должны использоваться библиотеки поддержки, имитирующие его интерфейс.

В качестве социальных сетей были выбраны Вконтакте и Facebook, поэтому соответственно должны использоваться VK Android SDK и Facebook Android SDK.

В качестве среды разработки была выбрана Android Studio, кроссплатформенная среда разработки, открыто распространяемая компанией Google.

Для отладки приложения должны быть использованы эмуляторы ОС Android, входящие в состав Android SDK, а также реальные устройства.

Средства для разработки выбраны, теперь можно разбить создание приложения на основные подзадачи:

* спроектировать api взаимодействия клиента и сервера;
* спроектировать пользовательский интерфейс;
* реализовать приложение с помощью выбранных средств;
* отладить код и протестировать разработанное приложение.

Определившись со средствами для реализации нужно определиться с технологией разработки приложения. Для разработки приложения была выбрана методология SCRUM (описывалась в первом разделе), где основой является - спиральная модель жизненного цикла.

Scrum — набор принципов, на которых строится процесс разработки, позволяющий в жёстко фиксированные и небольшие по времени итерации, называемые спринтами (sprints), предоставлять конечному пользователю работающее ПО с новыми возможностями, для которых определён наибольший приоритет.

Таким образом, разработка была поделена на итераций. Перед каждой итерацией оценивалось качество полученных результатов, и ставились определенные задачи по улучшению приложения. Каждое новое требование для приложения заносилась в бэклог проекта, и было доступно всем участникам. Каждый день оценивался объем выполненных работ, а также объем работы, который нужно проделать для завершения спринта. В конце каждой итерации получалась либо новая версия, либо новый фрагмент ПО.

Разработка происходила в несколько итерации. На первом этапе был произведен анализ конкурентов и маркетинговое исследование, был описан общий алгоритм работы приложения, а также был создан прототип прототипа интерфейса клиентской части в виде набора изображений.

В каждую последующую итерацию входило:

- разработка и дополнение API для взаимодействия сервера и клиентов;

- разработка и дополнение клиентской части приложения;

- разработка и дополнение серверной части приложения (идет параллельно с клиентской);

- тестирование приложения;

- оценка полученных результатов, постановка задач, по улучшению проекта.

Таким образом, было создано более 3 прототипов приложения, каждое их которых протестировано, проанализировано и намечены планы на улучшение прототипа.

## 2.4 Проектирование пользовательского интерфейса

Исходя из функциональных требований к приложению, было разработано несколько версий интерфейсов приложения.

В результате, был разработан интерфейс, состоящий из 3 экранов:

1. Экран загрузки;
2. Экран авторизации;
3. Основной экран приложения.

Подробное описание каждого из экранов представлено ниже.

**Экран загрузки**



Рисунок 1.8 – Экран загрузки приложения.

На экране загрузки происходит проверка наличия интернета, источников определения координат (GPS и Wi-Fi сети), получение ключа GCM, обновление и восстановление данных пользователя, определение координат.

**Экран авторизации**



Рисунок 1.9 – Экран авторизации.

Экран авторизации содержит кнопки авторизации через соц. сети и включает в себя авторизацию через ВКонтакте (кнопка В) и через Facebook (кнопка f).

**Основной экран приложения.**

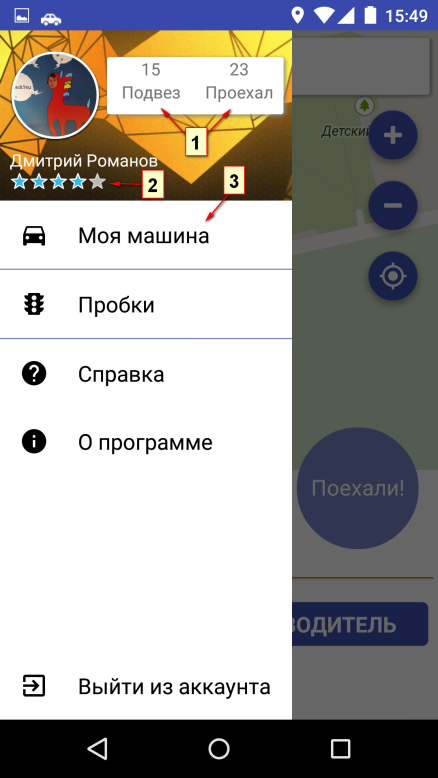


Рисунок 1.10 – Меню приложения

При загрузке основного экрана приложения появляется боковое меню, открывающееся оттягиванием экрана слева направо. В меню отображаются:

1. аватар и имя пользователя (из соц. сетей);
2. информация о количестве поездок (цифра 1)
3. рейтинг пользователя (цифра 2)
4. соответственно меню (цифра 3), включающее в себя пункты: “Моя машина” (отображает экран с информацией о машине), “Пробки” (включает/отключает отображение пробок на карте), “Cправка” ( отображает справку для текущего состояния поездки), “О программе” ( отображает информацию о программе).

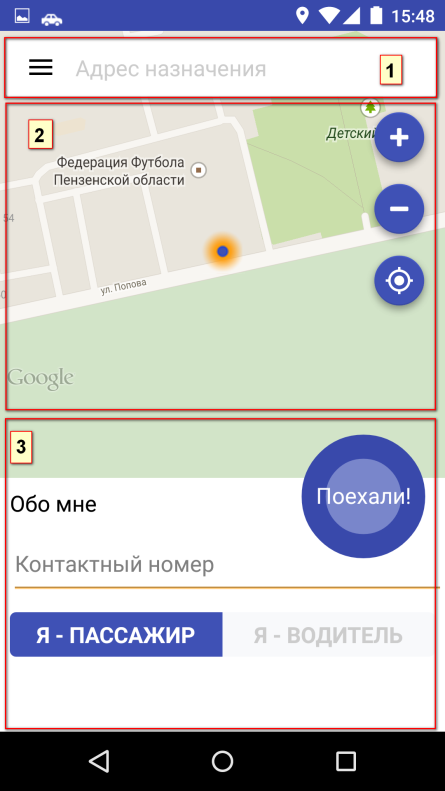


Рисунок 1.11 – Разделение основного экрана.

Основной экран делится на 3 области:

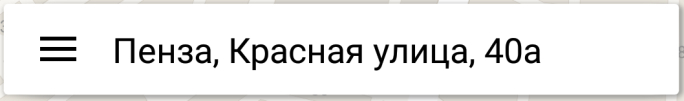
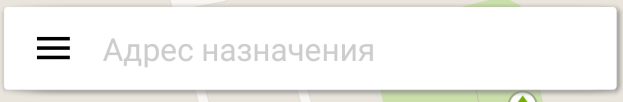
1) Верхняя панель действий;

2) Карта;

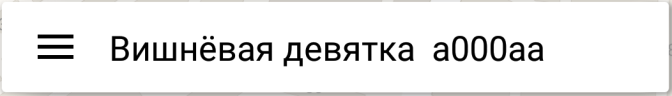
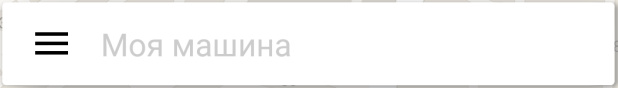
3) Нижний контейнер;

4) Нижняя выдвижная панель, закрывающая весь экран.

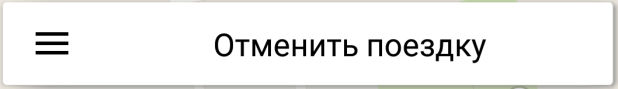
**Верхняя панель действий**



а) б)



в) г)



д)

Рисунок 1.12 – Варианты верхней панели действий

Верхняя панель действий постоянно находится на основном экране и не меняет свое положение. Содержит кнопку меню (левый край), открывающую меню, а также центральную кнопку, зависящую от состояния приложения.

Во время создания поездки часть данных отображается в центральной кнопке и зависит от выбранной роли и наполненности данных.

Над буквами а и б отображены состояния незаполненной и заполненной информации, при выборе роли пассажира. При нажатии на центральную кнопку выдвигается нижняя шторка для заполнения адреса назначения, описываемая ниже.

Над буквами в и г отображены состояния незаполненной и заполненной информации при выборе роли водителя. При нажатии на центральную кнопку выдвигается нижняя шторка для заполнения информации о машине, описываемая ниже.

Над буквой д отображено состояние, отображаемое во время поездки. При нажатии на кнопку происходит отмена поездки и возврат к ее созданию.

**Карта**

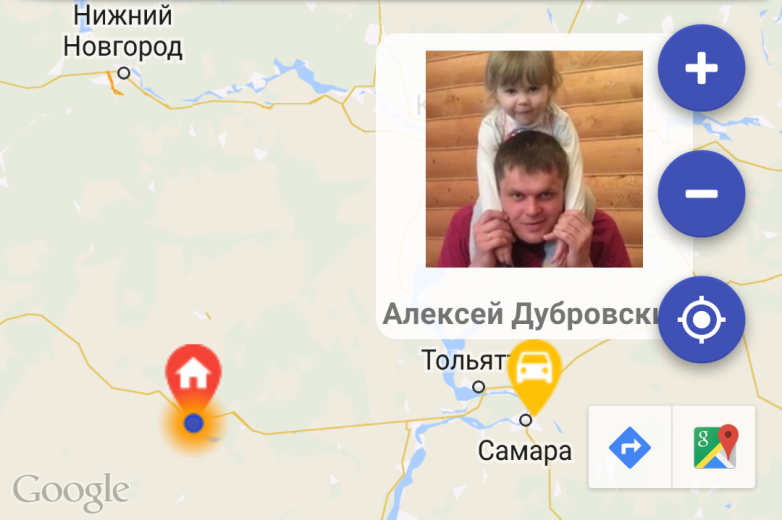


Рисунок 1.13 – Область карты

Область карты постоянно находится на основном экране. Содержит кнопки приближения, удаления карты, а также кнопку центрирования на положении пользователя. На карте отображаются маркеры, обозначающие: положение пользователя, адрес назначения пассажира, положения водителей и пассажиров. При нажатии на маркеры отображается информационное окно, зависящее от маркера.

**Нижний контейнер**

Нижний контейнер постоянно находится на основном экране, содержит фрагменты заменяющие друг друга в зависимости от состояния поездки.

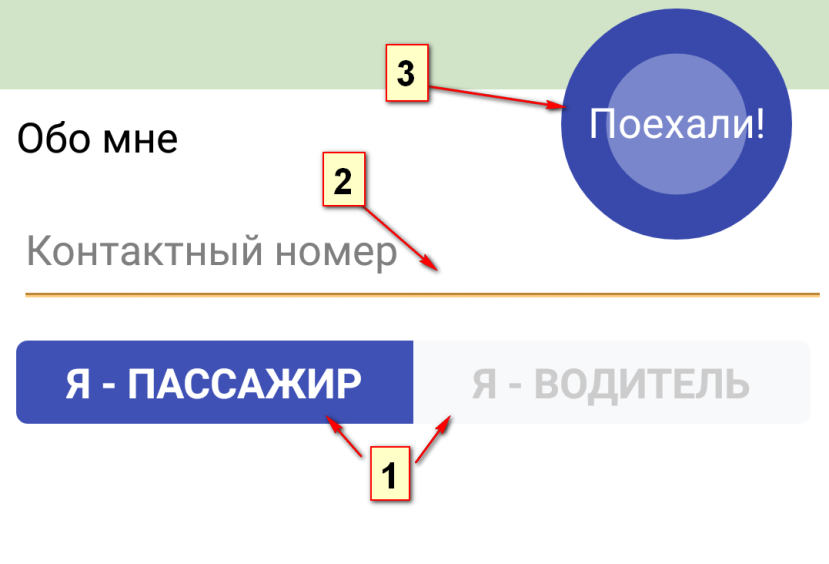


Рисунок 1.14 – Фрагмент создания поездки

Фрагмент создания поездки содержит в себе:

1. переключатель роли, для выбора роли пользователя;
2. поле ввода номера телефона;
3. кнопку “Поехали!” для создания поездки.



Рисунок 1.15– Фрагмент ожидания попутчиков

Фрагмент ожидания попутчиков содержит сообщение “Идет поиск водителей…” и “Идет поиск пассажиров…” в зависимости от роли пользователя.

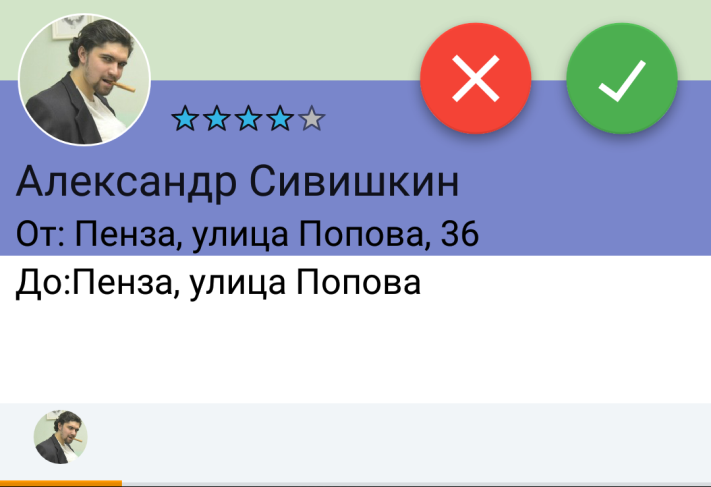
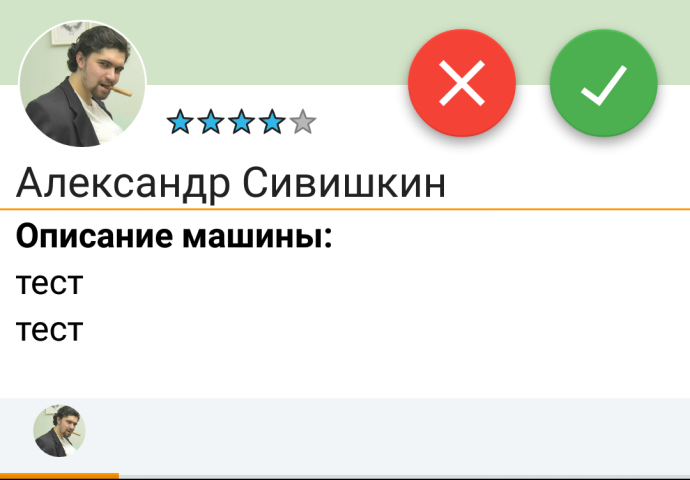


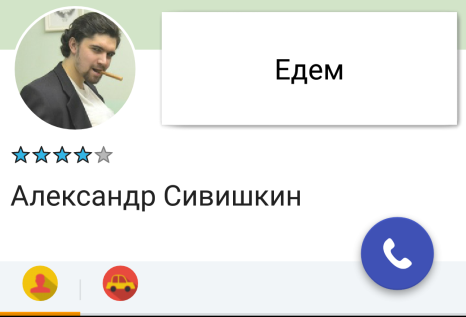
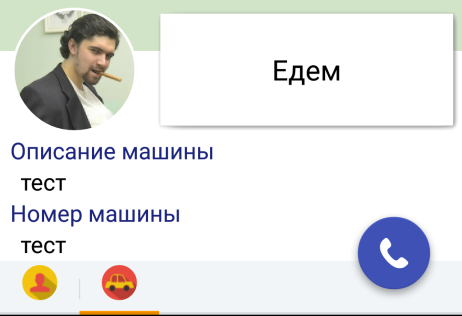
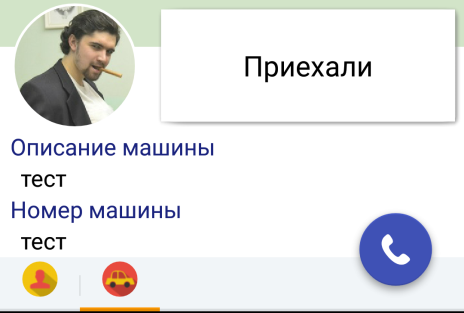
Рисунок 1.16 – Фрагмент выбора попутчиков для пассажира и для водителя

Для каждой роли фрагмент содержит:

1. аватар попутчика (левый верхний угол)
2. рейтинг попутчика (рядом с аватаром)
3. кнопку удаления попутчика из текущей поездки (крестик)
4. кнопку выбора попутчика (галочка)
5. список попутчиков в виде их аватаров (отображает одного из выбираемых) (снизу)
6. “листалку” попутчиков, отображающую текущего из выбираемых (центр)

Для пассажира в “листалке” отображаются имя водителя и описание машины. При нажатии на них водитель отображается на карте.

Для водителя в “листалке” отображаются имя пассажира и адреса (где находится пассажир и конечный адрес). При нажатии на один из адресов карта отображает его положение.

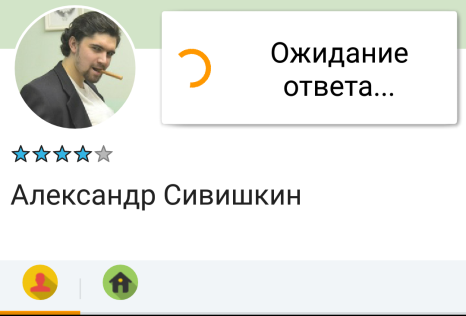
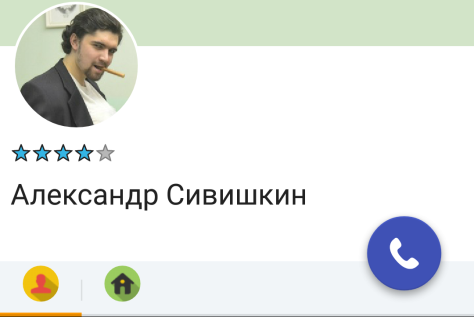
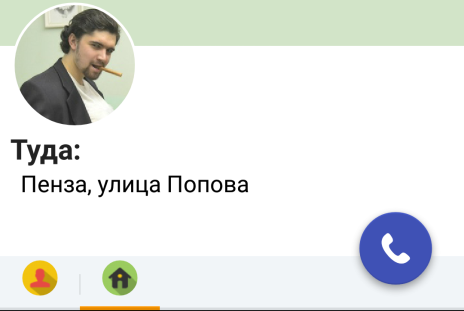
  

Рисунок 1.17 – Переключение фрагмента активной поездки для пассажира (сверху) и водителя (снизу).

Для пассажира фрагмент содержит:

1. аватар водителя
2. кнопку – “позвонить попутчику”
3. переключаемый фрагмент с информацией о водителе (верхний левый рисунок)
4. переключаемый фрагмент с информацией о машине (верхний центральный рисунок)
5. кнопку состояния поездки

Для водителя фрагмент содержит:

1. аватар попутчика
2. кнопку – “позвонить попутчику”
3. переключаемый фрагмент с информацией о пассажире (нижний левый рисунок)
4. переключаемый фрагмент с адресом назначения пассажира (нижний правый рисунок)
5. элемент, отмечающий состояние ожидания ответа

**Выдвижная панель**

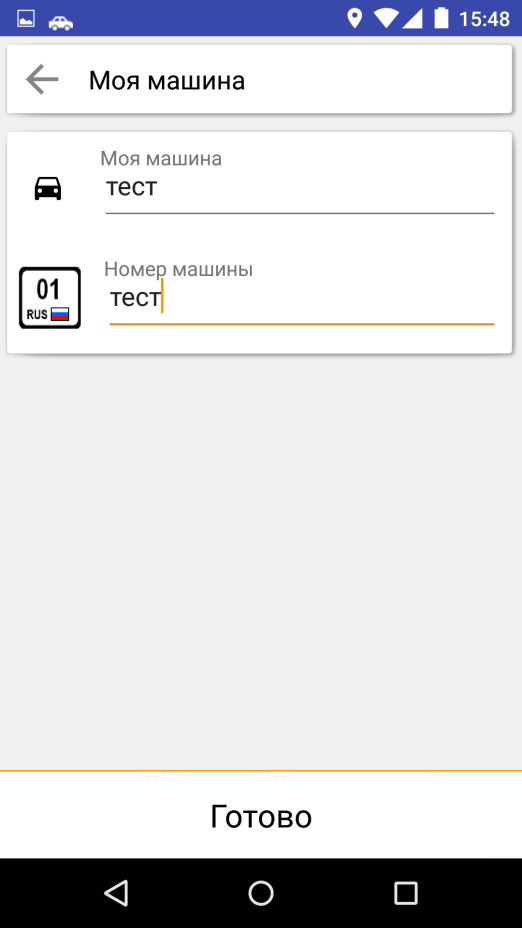
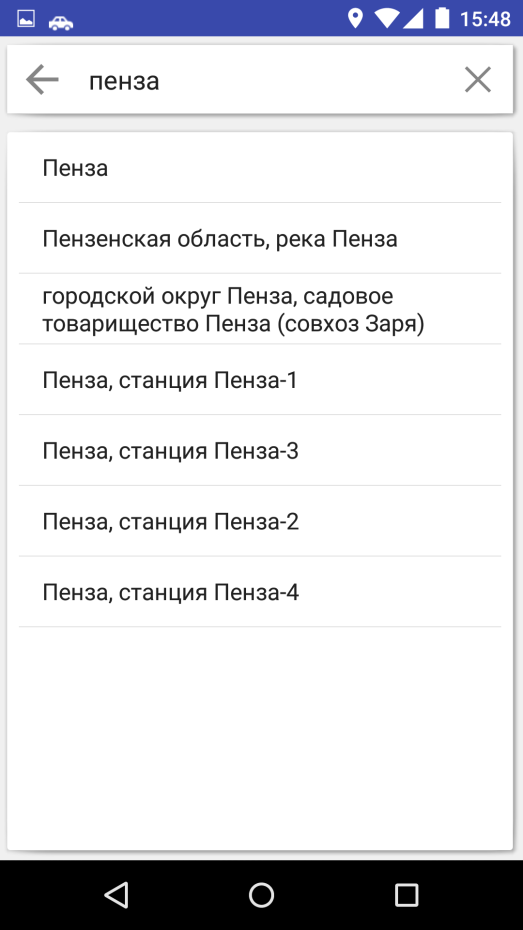


Рисунок 1.18 – Панели поиска адреса и информации о машине

При нажатии на адрес назначения в верхней панели действий выдвигается панель с поиском адреса. Она содержит вводимое поле, для адреса и список подходящих адресов. При нажатии на адрес происходит его выбор.

При нажатии на описание машины в верхней панели действий выдвигается панель с информацией о машине. Она содержит поля ввода описания и номера машины. При нажатии на кнопку готово происходит подтверждение введенной информации.

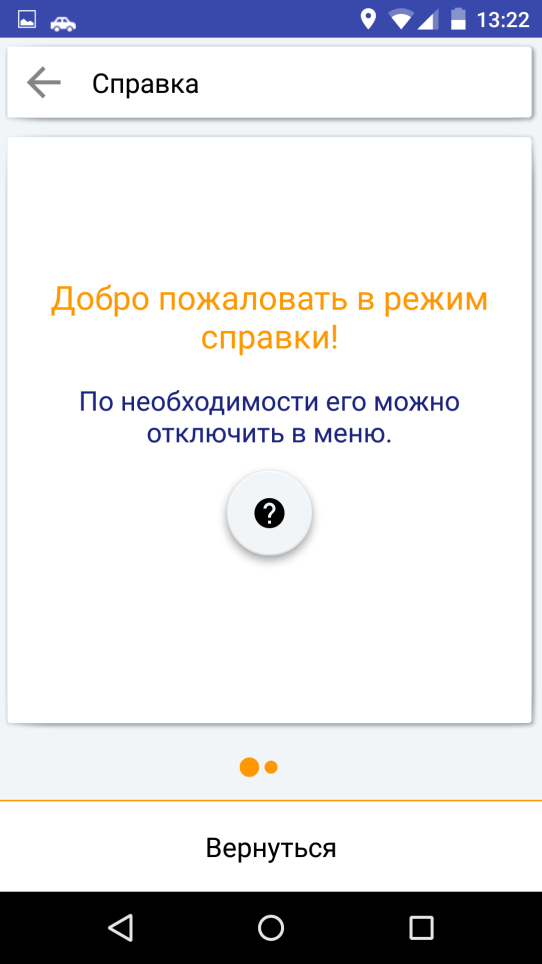


Рисунок 1.19 – Панели “О программе” и “Справка”

При нажатии на кнопки “О программе” и “Справка” открываются соответствующие панели. “О программе” содержит кнопки со ссылками на ВКонтакте, Facebook, PlayMarket. Справка содержит справочную информацию о текущем состоянии поездки. Обе панели закрываются на кнопку вернуться.

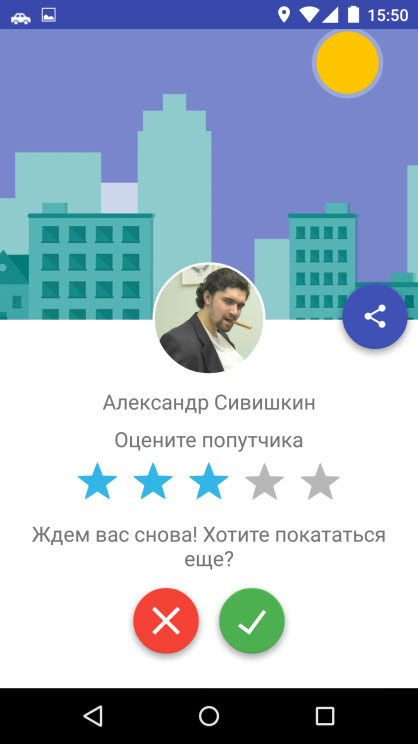


Рисунок 1.20 – Панель “О программе” и “Справка”

По завершении поездки, выдвигается панель, предлагающая оценить попутчика, или поделиться сообщением о поездке в соц. сетях. Кнопка поделиться находится справа от аватара попутчика. Оценка попутчику ставится в виде звездочек, при нажатии на них. Ниже находится сообщение, предлагающее повторить поездку и кнопки “не повторять” (крестик) и “повторить” (галочка). При нажатии на данные кнопки приложение либо закрывается (крестик), либо переходит на экран создания поездки (галочка).

**Сообщения в панели уведомлений**

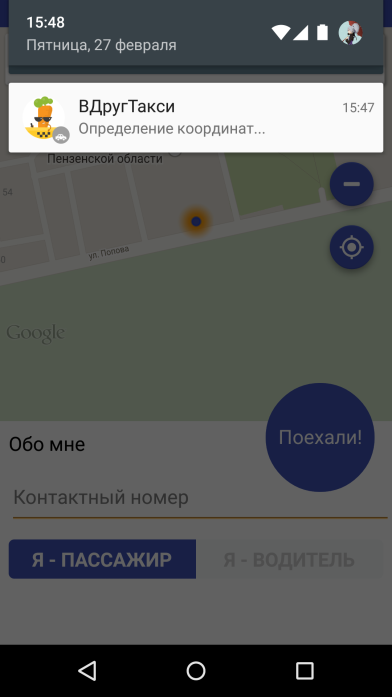


Рисунок 1.21– Сообщения в области уведомлений телефона.

Во время поиска попутчиков приложение позволяет свернуть себя, но отправка координат будет продолжаться. Для этого на панель уведомлений добавляется сообщение, которое невозможно убрать, пока пользователь не отменит поездку. Нажав на уведомление можно снова запустить приложение.

Также в область уведомлений приходят другие сообщения, если приложение находится в свернутом состоянии. Эти сообщения позволяют себя убрать и убираются автоматически при нажатии на них, и запуске приложения.

Список возможных сообщений:

1. Вы можете подвезти n пассажиров.
2. Вас хотят подвезти n водителей.
3. Пассажир согласился на поездку!
4. Пассажир отменил поездку
5. Водитель отменил поездку
6. Спасибо, что подвезли
7. За рулем? Найдите попутчика!
8. Определение координат…

## 2.5 Верстка дизайна

Верстка – превращение картинки в программный код. При разработке под андроид верстка осуществляется с помощью XML файлов.

### 1.4.1 Кодирование программных средств

Кодирование программы было произведено на языке Java 7 с использованием Android Framework в среде объектно-ориентированного визуального программирования Android Studio.

Интерфейс программы был создан с помощью языка XML, и png рисунков с использованием Model View Controller.

## 2.7 Тестирование программных средств

Тестирование позволяет убедиться в правильности работы приложения, а также определить ошибки и неточности в логике его работы, в способах отображения информации, реакции системы на действия пользователей. Выявленные на этапе тестирования ошибки и неточности устраняются и повторно тестируются.

Для данного приложения был разработан набор тестов, покрывающий функциональное тестирование и некоторые особенности системы. Все тесты разбиты на группы, связанные с какой либо часть программы. Подробное описание и результат выполнения тестов приведены ниже. Были проведены полевые тесты, в которых было установлено, что с помощью приложения действительно можно найти попутчиков. Проведены альфа и бета тесты, для проверки каждого прототипа приложения, перед его непосредственным внедрением, в том числе в приложение была добавлена Google Аналитика для отлова и исправления ошибок на всех этапах тестирования.

Таблица 1.3 – Тесты

|  |  |  |  |
| --- | --- | --- | --- |
| Описание теста | Ожидаемый результат | Полученный результат | Ссылка |
| Запуск приложения | | | |
| Запуск приложения с отключенным интернетом | Сообщение об отсутствии интернета. | Запускается приложение. На стартовом экране возникает диалог, предлагающий включить интернет. | Рис В.1 |
| Запуск приложения с включенным GPS,Wi-Fi и интернетом и пользователем | Переход на основной экран | Приложение обновляет данные из соц. сетей и переходит на основной экран | - |
| Авторизация через социальные сети | | | |
| Авторизация с использованием Вконтакте или Facebook | Авторизация | Пользователь авторизуется, и приложение переходит на основной экран. | Рис В.3 |
| Авторизация, при отключенном интернете | Сообщение об отсутствии интернета | Пользователю отображается сообщение, об отсутствии интернета либо из социальной сети, либо из приложения. | Рис В.4 |
| Заполнение данных | | | |
| Выбор роли пассажира или водителя | Смена заполняемых данных | Приложение отображает на верхней панели действий адрес или информацию о машине, в зависимости от роли | Рис В.7 |
| Выбор роли пассажира, заполнение адреса, нажатие на кнопку “Поехали!” | Создание поездки и отображение ее на экране | Приложение отображает экран ожидания водителей, отправляет на сервер запрос о создании поездки, на карте отображаются маркеры водителей | Рис В.9 |
| Выбор роли водителя, нажатие на кнопку “Поехали!” | Не создавать поездку, пока не будут определены данные о машине | Приложение отображает сообщение, что нужно заполнить данные о машине и открывает панель ввода данных о машине | Рис В.10 |
| Просмотр информации о попутчике в социальных сетях | | | |
| Нажать на аватар попутчика | Открыть профиль в соц. сети | Приложение вызывает системное меню, предлагающее открыть профиль попутчика в соц. сети, через имеющиеся приложения, откуда его можно открыть в соц. сетях | Рис В.13 |
| Получение уведомления о завершении поездки | | | |
| Пришло уведомление о том, что поездка завершилась, при запущенном приложении | Уведомить водителя о завершении поездки и открыть экран с рейтингом | Приложение, издает звуковой сигнал и открывает экран с рейтингом | Рис В.19 |
| Пришло уведомление о том, что поездка завершилась, при свернутом приложении | Уведомить водителя о том, что поездка завершилась | На панели уведомлений отобразилось сообщение: “Спасибо, что подвезли!”  Сообщение сопровождается звуковым сигналом. Приложение перестает отправлять координаты на сервер | - |
| Выставление рейтинга | | | |
| Выставить рейтинг и нажать на кнопку “Крестик” (Не согласен на повторную поездку) | Завершение работы приложения | Приложение полностью завершает свою работу | - |
| Выставить рейтинг и нажать на кнопку “Галочка” (Согласен на повторную поездку) | Открыть экран создания поездки | Приложение открывает экран создания поездки | - |

# Заключение

Анализ феномена мобильного маркетинга показал, что данное направление бизнеса очень перспективно и мгновенно окупается, так как спрос на мобильные устройства неуклонно растёт и увеличивается спрос на интернет-услуги. Одна из тенденций развития современного рынка рекламных услуг проявляется в постепенном смещении интернет-рекламы в сторону удовлетворения потребностей целевой аудитории, выходящей в интернет с мобильных устройств.

Задачи, поставленные в работе

…

были решены. Приложение “ВДругТакси” просто в использовании, так как сочетает концепцию простого интерфейса “в один клик” и авторизации через социальные сети, доступно всем категориям граждан. Рейтинговая система позволяет отслеживать надежность попутчиков.

Программный продукт подбирает ближайшие к пассажирам, подходящие попутчикам по пути их следования маршруты, в отличие от известных приложений, где реализуются поездки от точки до точки, и пассажир и водитель должны вводить полный маршрут движения. Мобильное приложение составляет мощную конкуренцию продуктам-аналогам ввиду решения следующих проблем, волнующие горожан: скорость, цена и качество, комфорт, безопасность.

Погружение транспортной среды в информационную привело к положительным результатам. С помощью приложения и сервисов мобильного телефона стало возможным решение не только транспортных, но и социальных проблем города Пензы, улучшая транспортную обстановку и приводя к сплочению общества за счет коллективного использования личного транспорта, знакомства попутчиков в процессе карпулинга, взаимоуважения

развитие сервисов на основе информационных и телекоммуникационных технологий в сферах культуры, образования и здравоохранения;

возможность осуществления трудовой деятельности дистанционно и содействие самозанятости;

обеспечении нормального функционирования сложного хозяйственного комплекса страны

# Список используемой литературы

1. Брюс Эккель. Философия Java = Thinking in Java. — 3-е изд. — СПб.:Питер 2003. — 976 с.
2. Герберт Шилдт. Java 8. Полное руководство, 9-е издание = Java 8. The Complete Reference, 9th Edition. — М.: [«Вильямс»](https://ru.wikipedia.org/w/index.php?title=%D0%92%D0%B8%D0%BB%D1%8C%D1%8F%D0%BC%D1%81_(%D0%B8%D0%B7%D0%B4%D0%B0%D1%82%D0%B5%D0%BB%D1%8C%D1%81%D1%82%D0%B2%D0%BE)&action=edit&redlink=1), 2015. — 1376 с. .
3. Джошуа Блох. Java. Эффективное программирование = Effective Java. — М.: [Лори](https://ru.wikipedia.org/w/index.php?title=%D0%9B%D0%BE%D1%80%D0%B8_(%D0%B8%D0%B7%D0%B4%D0%B0%D1%82%D0%B5%D0%BB%D1%8C%D1%81%D1%82%D0%B2%D0%BE)&action=edit&redlink=1), 2002. — 224 с.
4. Кей С. Хорстманн, Гари Корнелл. Java. Библиотека профессионала, том 1. Основы. 9-е издание = Core Java, Volume I: Fundamentals (9th Edition). —М.: [«Вильямс»](https://ru.wikipedia.org/w/index.php?title=%D0%92%D0%B8%D0%BB%D1%8C%D1%8F%D0%BC%D1%81_(%D0%B8%D0%B7%D0%B4%D0%B0%D1%82%D0%B5%D0%BB%D1%8C%D1%81%D1%82%D0%B2%D0%BE)&action=edit&redlink=1), 2013. — 864 с.
5. Кей С. Хорстманн. Java SE 8. Вводный курс = Java SE 8 for the Really Impatient. — М.: [«Вильямс»](https://ru.wikipedia.org/w/index.php?title=%D0%92%D0%B8%D0%BB%D1%8C%D1%8F%D0%BC%D1%81_(%D0%B8%D0%B7%D0%B4%D0%B0%D1%82%D0%B5%D0%BB%D1%8C%D1%81%D1%82%D0%B2%D0%BE)&action=edit&redlink=1), 2014. — 208 с..
6. Монахов Вадим. Язык программирования Java и среда NetBeans. — 3-е изд. — СПб.: [БХВ-Петербург](https://ru.wikipedia.org/w/index.php?title=%D0%91%D0%A5%D0%92-%D0%9F%D0%B5%D1%82%D0%B5%D1%80%D0%B1%D1%83%D1%80%D0%B3&action=edit&redlink=1), 2011. — 704 с.
7. Роберт Седжвик, Кевин Уэйн. Алгоритмы на Java, 4-е издание = Algorithms, 4th Edition. — М.: [«Вильямс»](https://ru.wikipedia.org/w/index.php?title=%D0%92%D0%B8%D0%BB%D1%8C%D1%8F%D0%BC%D1%81_(%D0%B8%D0%B7%D0%B4%D0%B0%D1%82%D0%B5%D0%BB%D1%8C%D1%81%D1%82%D0%B2%D0%BE)&action=edit&redlink=1), 2013. — 848 с.
8. Фрэд Лонг, Дхрув Мохиндра, Роберт С. Сикорд, Дин Ф. Сазерленд, Дэвид Свобода. Руководство для программиста на Java: 75 рекомендаций по написанию надежных и защищенных программ = Java Coding Guidelines: 75 Recommendations for Reliable and Secure Programs. — М.: [«Вильямс»](https://ru.wikipedia.org/w/index.php?title=%D0%92%D0%B8%D0%BB%D1%8C%D1%8F%D0%BC%D1%81_(%D0%B8%D0%B7%D0%B4%D0%B0%D1%82%D0%B5%D0%BB%D1%8C%D1%81%D1%82%D0%B2%D0%BE)&action=edit&redlink=1), 2014. — 256 с.
9. Ссылки: 1. Министерство транспорта Российской Федерации. URL: http://www.mintrans.ru/ (дата обращения: 10.03.2014). 2. Транспортная стратегия Российской Федерации на период до 2030 г. URL: http://www.mintrans.ru/documents/detail.php?ELEMENT\_ID=19188 (дата обращения: 10.03.2014). 3. Там же. 4. Там же.

# Список сокращений

ЖЦ - жизненный цикл

ОС - операционная система

ПО - программного обеспечения

SDK - Software Development Kit

БД - база данных

СУБД - система управления базами данных

ИКТ информационные компьютерные технологии