

УДК 004

## ***ИССЛЕДОВАНИЕ ОСОБЕННОСТЕЙ ПРОЕКТИРОВАНИЯ МОБИЛЬНЫХ ПРИЛОЖЕНИЙ***

***Цебренько К.Н.***

*к.т.н., доцент*

*Академии маркетинга и социально-информационных технологий-ИМСИТ (г. Краснодар)*

*г. Краснодар, Российская Федерация*

### **Аннотация**

Актуальность темы исследования обусловлена высокими темпами роста мобильного бизнеса и широким распространением мобильных устройств. Целью работы является исследование особенностей проектирования мобильного приложения для сервиса такси. Для достижения цели проведено исследование на примере проектирования мобильного приложения сервиса такси с использованием методов гибридного моделирования. Разработаны модели бизнес-процессов. На их основе реализована UML модель мобильного приложения. Разработана структура информационного обеспечения. На основе разработанных проектных решений показана эффективность описанных подходов к проектированию.

**Ключевые слова:** Мобильное приложение, такси, ИТ-проект, IDEF, UML, проектирование.

### ***RESEARCH FEATURES OF DESIGNING MOBILE APPLICATIONS***

***Tsebrenko K.N.***

*PhD, Associate professor*

*Academy of Marketing and Social information technologies – IMSIT*

*Krasnodar, Russian Federation*

**Abstract**

The relevance of the research topic is due to the high growth rates of mobile business and the widespread use of mobile devices. The aim of the work is to study the design features of a mobile application for a taxi service. To achieve the goal, a study was carried out on the example of designing a mobile application for a taxi service using hybrid modeling methods. Business process models have been developed. Based on them, the UML model of a mobile application is implemented. The structure of information support has been developed. Based on the developed design solutions, the effectiveness of the described design approaches is shown.

**Keywords:** Mobile application, taxi, IT project, IDEF, UML, design.

Согласно заявлению, аналитическая компания Pew Research Center, опубликовавшей статистику в 2019 году, относительно популярности смартфонов в мире: более 5 миллиардов человек имеют мобильные устройства, и более половины из этих подключений являются смартфонами 78% опрошенных взрослых в мире являются владельцами смартфонов, 22% пользуются простыми мобильными телефонами, а у 8% нет ни того, ни другого. Из этого стоит сделать выводы, что большое количество людей используют не просто кнопочные мобильные телефоны, а полноценные android или iOS девайсы [1]. Рассмотрим вопросы проектирования мобильных приложений на примере разработки приложения для сервиса «такси».

Количество занятых в отрасли такси на 2019 год составляет порядка 584 тыс. человек, при этом в течение года в этой сфере работали порядка 117 тыс. человек. С 2015 по 2017 год численность таксистов росла на 25% ежегодно. В 2018 г. по сравнению с 2017 годом значимого роста численности водителей не произошло. Однако при этом возросло число активно эксплуатируемых автомобилей такси (используемых в две смены). Численность водителей, работающих в одну из смен на таких автомобилях, увеличилась на 21% по Дневник науки | [www.dnevniknauki.ru](http://www.dnevniknauki.ru) | СМИ Эл № ФС 77-68405 ISSN 2541-8327

сравнению с 2018 годом. Рост рынка, таким образом, шел за счет интенсификации труда и повышения утилизации автомобилей такси. Следовательно, люди очень часто пользуются такси [2].

Мобильные приложения для такси распространены очень сильно. Начиная от Яндекс.Такси, одной из самостоятельных бизнес-единиц Яндекса, предлагающей сервисы агрегатора такси, и заканчивая Uber, которое предлагает мобильное приложение для поиска, вызова и оплаты такси или частных водителей [3].

Имея необходимые сведения – можно сделать вывод, что такси используют очень много людей, и почти у каждого есть мобильное устройство. Но не каждый хочет звонить в диспетчерскую пару минут объяснять куда надо приехать и ждать очень медленно пишущего оператора. Гораздо проще скачать приложение и выбрать все параметры самостоятельно. Можно буквально «одним щелчком пальца» заказать такси. Мобильное приложение решит проблему перегрузки операторов, за счёт автоматизации поступления и приема заказов. Нагрузка на операторов ниже, а прибыльность компании увеличится.

Мобильное приложение предназначено для работы на планшетах, смартфонах и остальных мобильных устройствах. Приложения могут быть загружены из сведущих онлайн магазинов: Google play, App Store официальный сайт компании [4].

Целью работы является исследование особенностей проектирования мобильного приложения для сервиса такси.

Объектом исследования является мобильное приложение такси.

Предметом исследования является проектирование мобильного приложения для работы с данными таксистов и клиентов.

На первом этапе проектирования проведем структурный анализ мобильного приложения такси [5] в нотации IDEF0. На рисунке 1 изображена концептуальная модель мобильного приложения такси. Вход: номер телефона, Ф.И.О Пользователя, запрос, исходная БД. Выход: ответ, измененная БД.  
Дневник науки | [www.dnevniknauki.ru](http://www.dnevniknauki.ru) | СМИ Эл № ФС 77-68405 ISSN 2541-8327

Управление: требования заказчика, пользовательское соглашение. Механизм: пользователь. Данный блок имеет входные и выходные данные, а также механизмы исполнения и управления [6]. Все рисунки содержат результаты моделирования, выполненные автором.

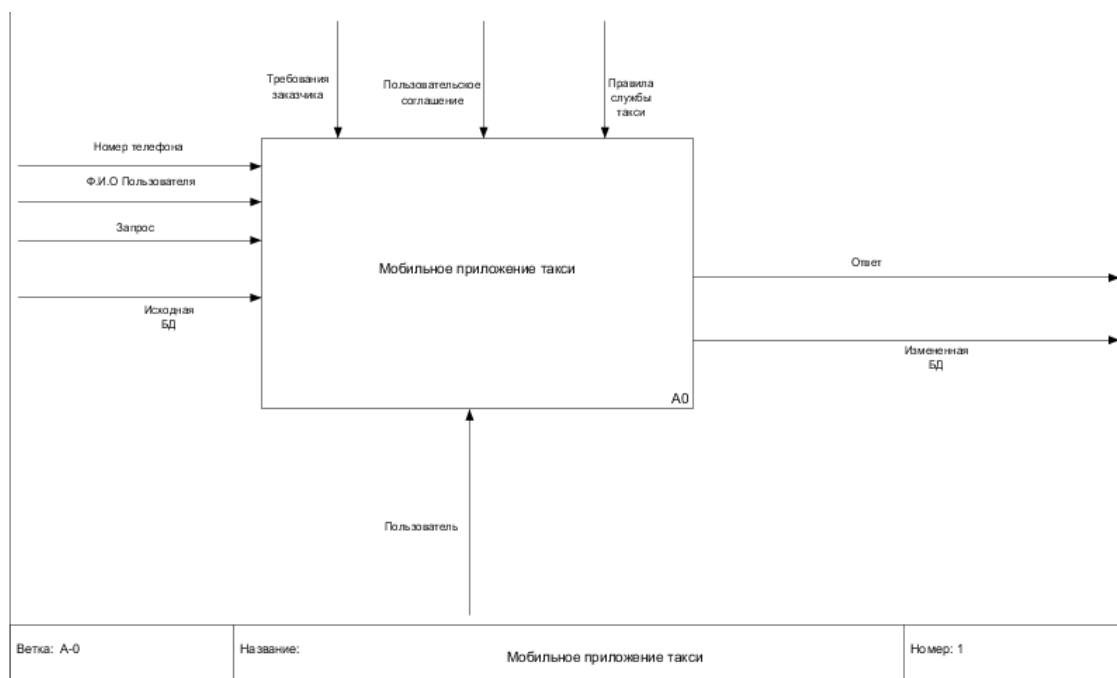


Рисунок 1 – Концептуальная модель

Проведя декомпозицию концептуальной модели первого уровня мы можем заметить новые блоки: авторизация пользователя, работа с данными такси, техническая поддержка, изменение БД и генерация ответа. Данная диаграмма представлена на рисунке 2.

На рисунке 3 предоставлена декомпозиция блока “Авторизация пользователя”. Сперва, перед входом в приложение идет проверка на существование пользователя. Если это первый вход в приложение – система отправит смс подтверждение номера с помощью блока Рассылка СМС, а затем вернётся в авторизацию для подтверждения входа. Дальше блок сам решит входить ему или регистрироваться. Работа функционального блока “Вход в систему” упрощена.

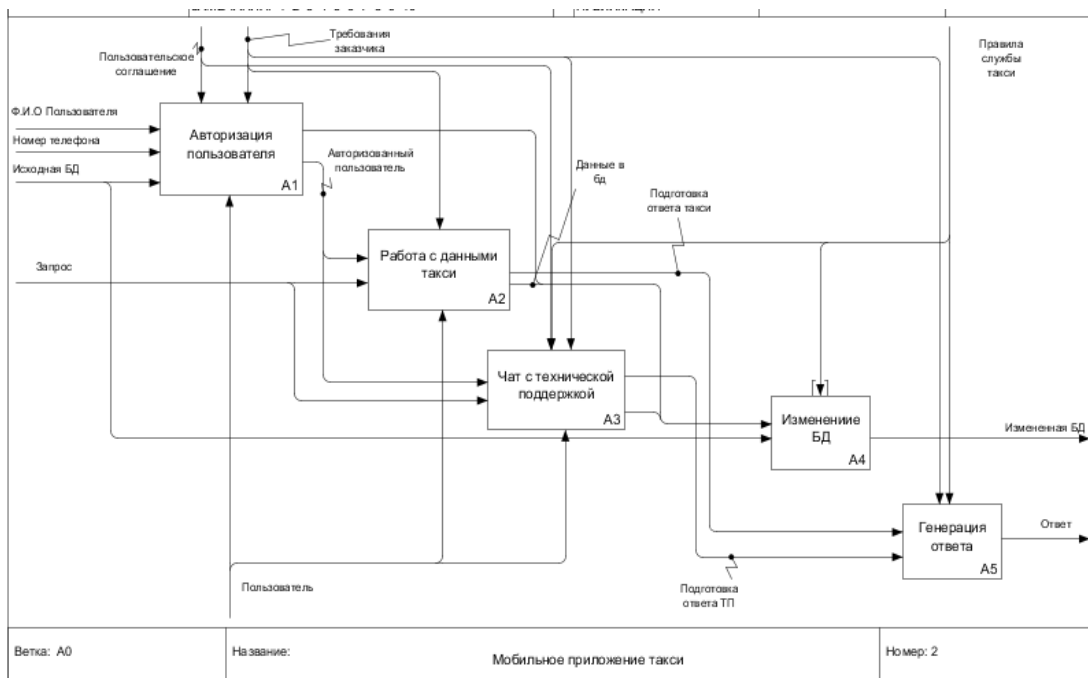


Рисунок 2 – Концептуальная модель второго уровня

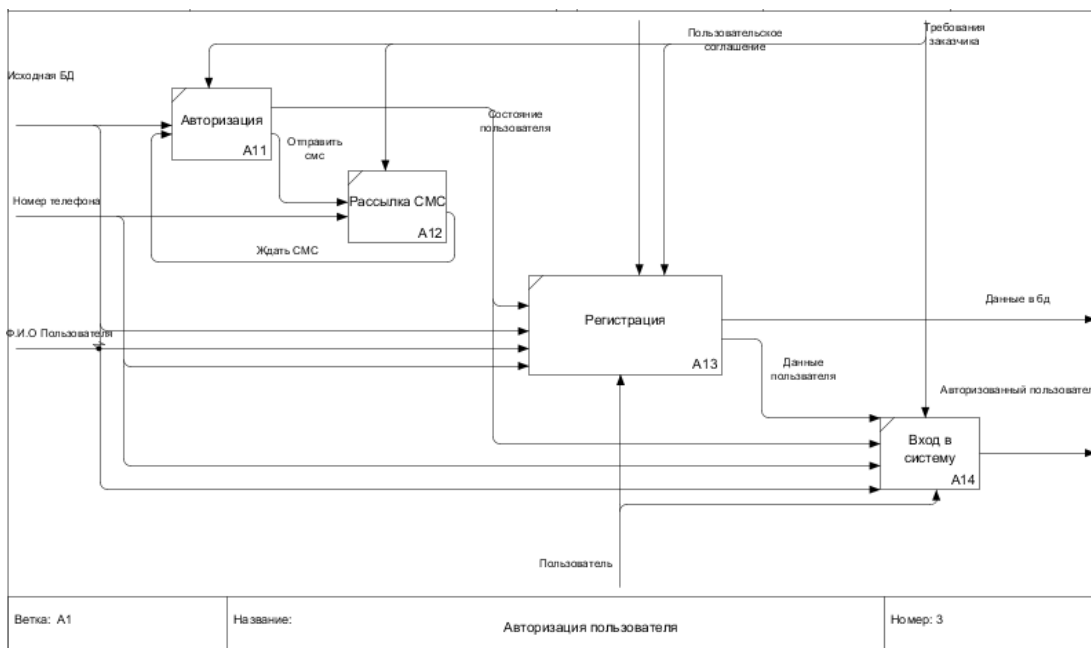


Рисунок 3 – Декомпозиция блока “Авторизация пользователя”

Для того что бы войти в приложение такси - достаточно лишь ввести номер телефона, получить сообщение, ввести указанный код. Данные манипуляции упрощают работу пользователя с приложением, нежели запоминание и хранение логина и пароля. В случае регистрации, для блока “Регистрация”, у пользователя

запросят его фамилию, имя и отчество. Эта процедура является одноразовой и хранится в базе данных.

Функциональный блок “Работа с данными такси”, который изображен на рисунке 4, является одним из важнейших блоков мобильного приложения [7]. Ведь именно он и является той самой основой, благодаря чему приложение может существовать. В него входят следующие блоки: заказ такси, информация о состоянии заказа, отказ от такси и завершение поездки. Последний пункт предназначен для того что бы закрыть поездку в базе данных и дать пассажиру доступ к оценке водителя и наоборот. После заказа такси пользователю сразу же покажется информация о состоянии заказа, а именно: поиск машины, машина найдена или заказ отменен.

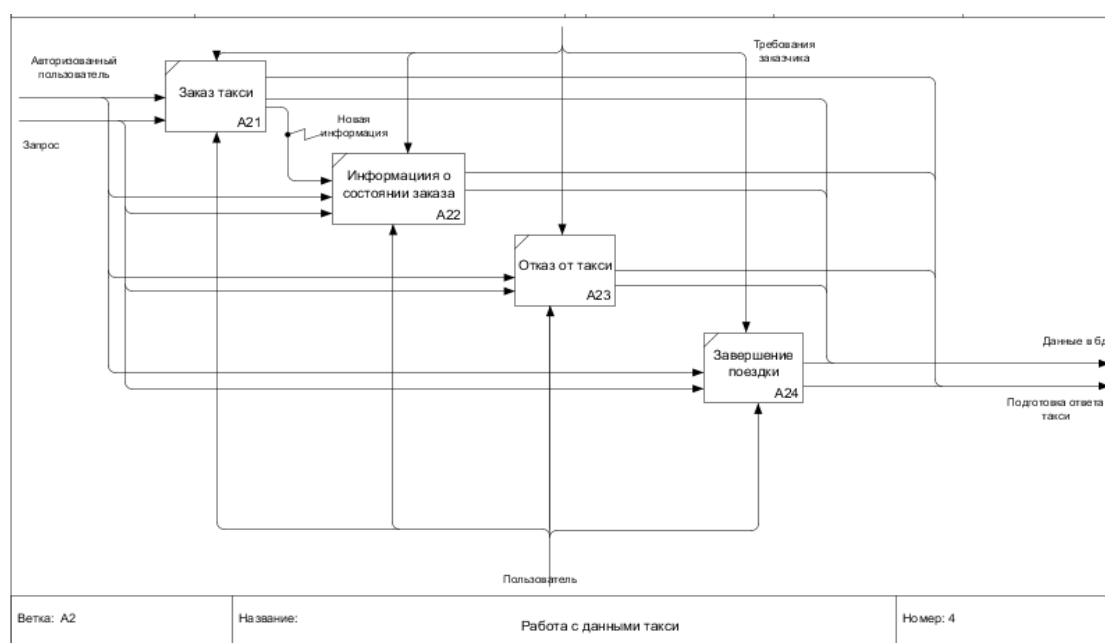


Рисунок 4 – Декомпозиция блока “Работа с данными такси”

В первую очередь, блок “Техническая поддержка” создан для тех групп лиц, которые нашли ошибки или недоработки в приложении, но не исключено, что люди смогут открывать тикет для того что бы пожаловаться на водителя или спросить: “как заказать такси”. К счастью для блока “Ответ на тикет” компания такси может собрать высококвалифицированных специалистов, работающих в

данных сфере и диспетчеров, которые могут помочь разобраться с заказом такси. Сразу же после ответа специалиста – тикет закрывается, до тех пор, пока человек, который задал вопрос не продолжит разговор. Для этого момента в базе данных выделено место под чат со специалистами, которых можно будет менять в процессе всего диалога. Благодаря данной системе работа и общение со специалистом может происходить в режиме реального времени с учетом пользователя и множеством операторов. Декомпозиция блока “Техническая поддержка” предоставлена на рисунке 5.

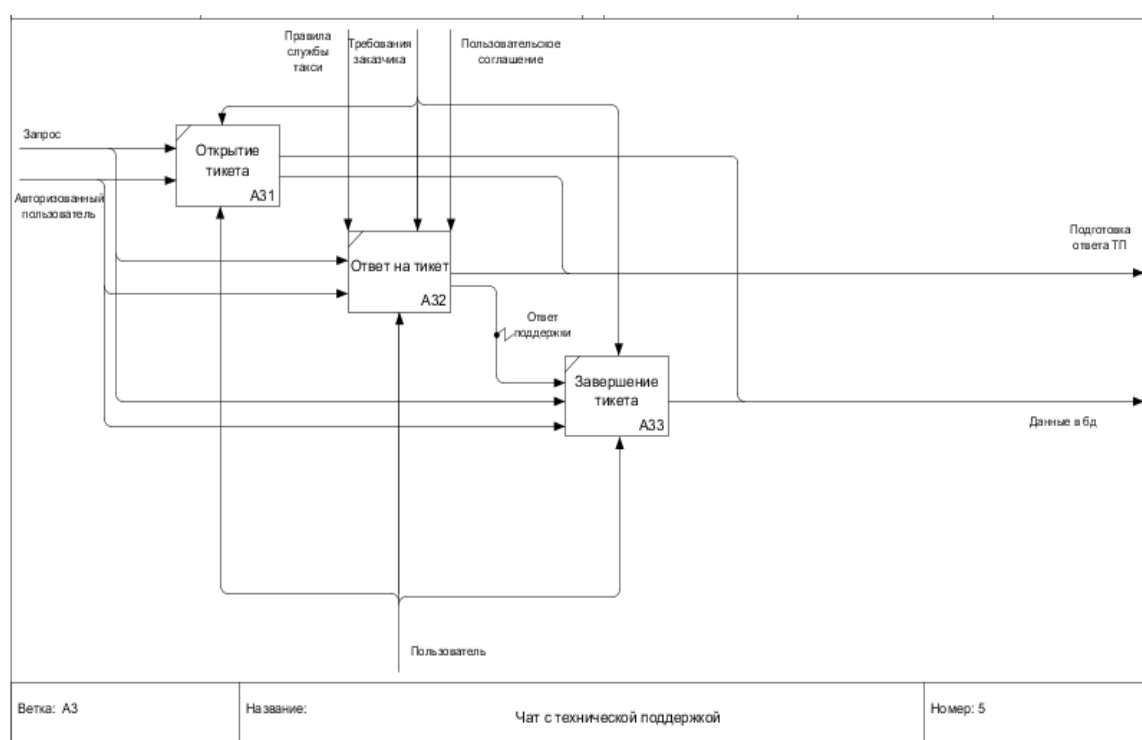


Рисунок 5 – Декомпозиция блока “Техническая поддержка”

Почти за каждым действием человека – следует изменение определенной таблицы в базе данных. Будь то регистрация, создание тикета в тех. Поддержке, заказ такси в приложении – пользователь обратится к функциональному блоку “Создание данных”. Если же такси будет перемещаться, каждый раз меняя свои GPS координаты на более новые или пользователь, который вдруг резко поменяет свою фамилию, то данные обновятся, а, следовательно, будет затрагиваться ранее неизвестный нам блок “обновление данных”. В случае если

пользователь передумал ставить оценку за поездку, отменил поиск машины или администратор, который решил заблокировать пользователя, то за вышеперечисленными действиями последует блок “Удаление данных”. Создание, обновление, удаление – все действия происходят исключительно в отдельной базе данных на купленном сервере.

Вся информация, которая обрабатывается в приложении имеет весьма непривычный вид для чтения. Пользователь будет весьма удивлен, если на выходе у приложения вместо привычных букв - последует текстовый формат обмена данными, основанный на JavaScript. За счёт своей лаконичности по сравнению с XML формат JSON может быть более подходящим для сериализации сложных структур. Обычно, он применяется в веб-приложениях как для обмена данными между браузером и сервером (AJAX), так и между серверами (программные HTTP-сопряжения). Для этого случая разработана декомпозиция блока “Генерация ответа”. Декодер JSON строк, а именно блок “Перевод JSON строк в читабельный вид”, позволит перевести текст из объектов в необходимую информацию, а блок “Форматирование текста” поможет в выборе начертания шрифта и использовании эффектов, позволяющих менять вид текста.

Проектирование работы программной части выполнено с помощью объектно-ориентированного подхода [8-9].

Диаграмма прецедентов (Use case diagram, диаграмма вариантов использования) — диаграмма, на которой отражены отношения, существующие между актерами и прецедентами [10].

Основная задача — представлять собой единое средство, дающее возможность заказчику, конечному пользователю и разработчику совместно обсуждать функциональность и поведение системы.

Опишем систему, изображенную на рисунке 6. Во время входа в мобильное приложение пользователь проходит через авторизацию и ему



присваивается определённая роль, а именно: клиент, водитель, администратор или тех. Поддержка. У каждой группы людей свой набор разрешенных задач.

Диаграмма последовательности — это диаграмма, на которой показано взаимодействие объектов (обмен между ними сигналами и сообщениями), упорядоченное по времени, с отражением продолжительности обработки и последовательности их проявления.

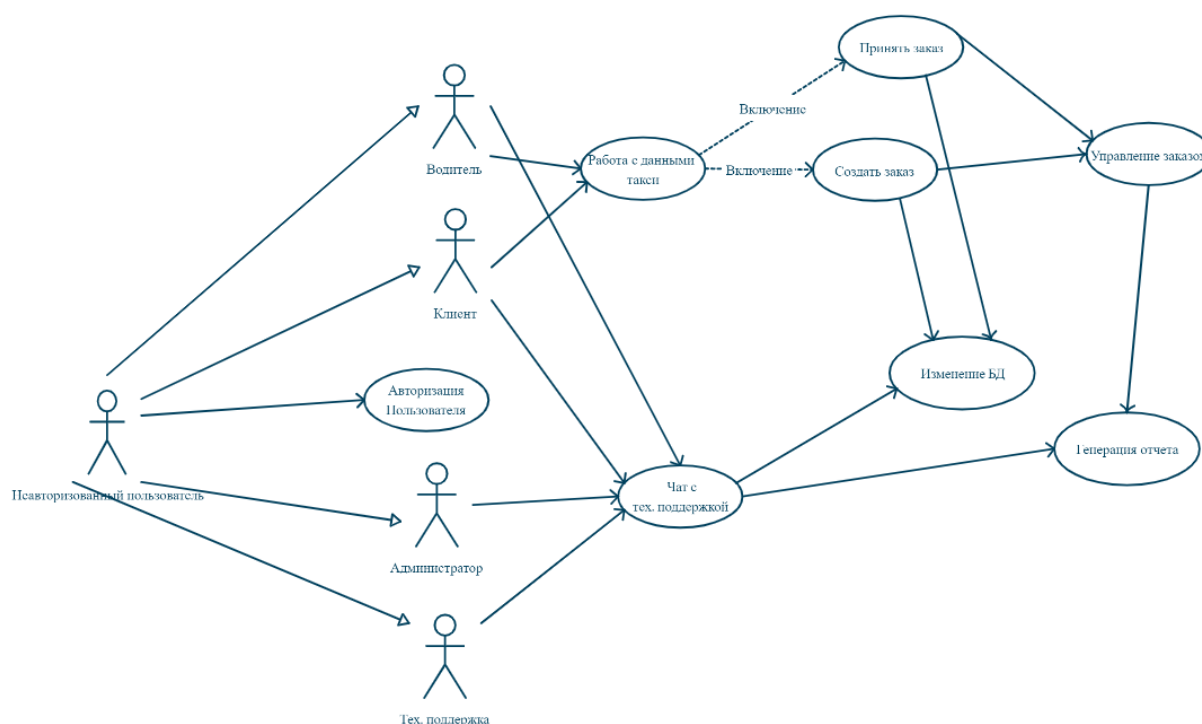


Рисунок 6 – Диаграмма вариантов использования

Диаграмма последовательности Авторизации изображена на рисунке 7. Действующим лицом является Пользователь. Сначала приложение проверяет свои авто сохранения входил ли человек ранее в приложение методом (1://ВходилЛиПользовательРанее?()), если имеется вся необходимая информация, то автоматически осуществляется переход в метод 7://ПользовательЕстьВСистеме(Номер\_пользователя, Ключ\_сессии), в противном случае отправляется СМС (4://ОтправитьСМС(Номер\_телефона)) с четырехзначным кодом подтверждения через специальные мобильные сервисы. В случае, если СМС сообщение не пришло или утерялось, то можно его

отправить заново (6://ПовторноОтправитьСМС(Номер\_телефона)). После подтверждения номера телефона на уровень его существования система Смотрит, входил ли пользователь ранее, если да, то осуществляется вход в приложение (7://ПользовательЕстьВСистеме(Номер\_пользователя, Ключ\_сессии)), если пользователя не было ранее в системе, то запрашивается Его Ф.И.О и методом 8: //ПользовательНеВСистеме(Номер\_телефона, ФИО), переходит в регистрацию. Сразу же, после успешной регистрации осуществляется переход на Вход (9: //ПользовательЗарегистрировался(Номер\_пользователя, Ключ\_сессии)). Во время входа мы должны определить кем является пользователь: клиентом, водителем, администратором и т.д (10://ОпределениеДоступа()) и переходит на стартовую форму (11://ПереходНаГлавную())

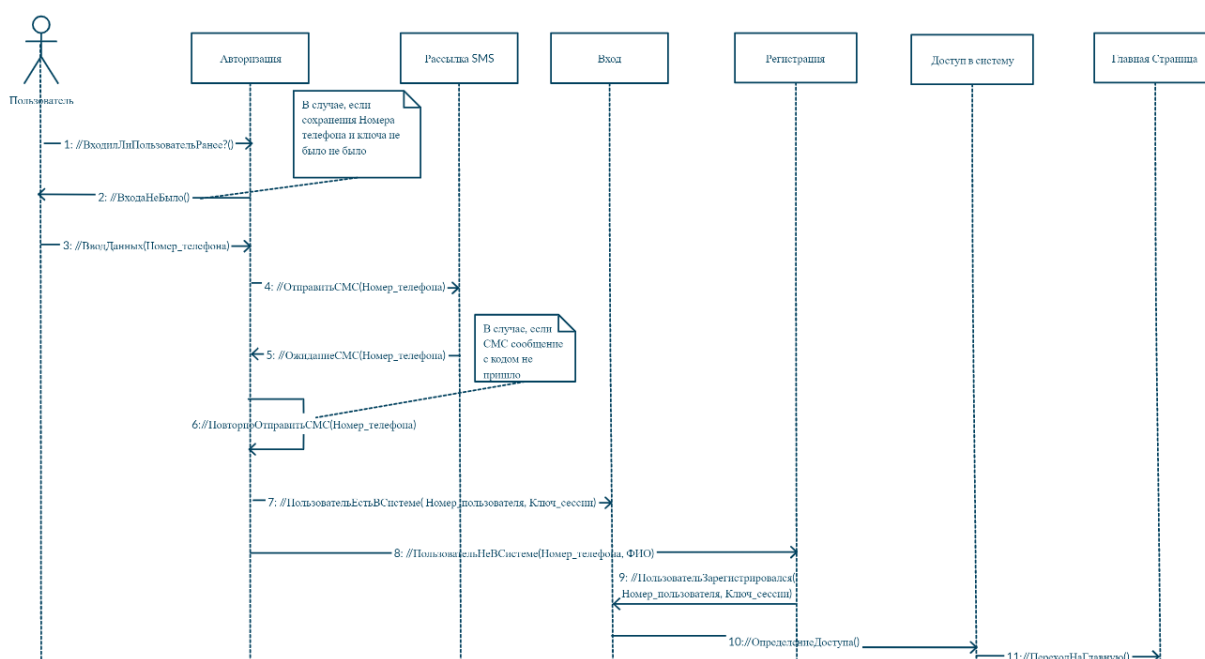


Рисунок 7 – Диаграмма использования Авторизации

Следующим шагом рассмотрим диаграмму влечения “Создать заказ” от “Работа с данными такси” на рисунке 8.

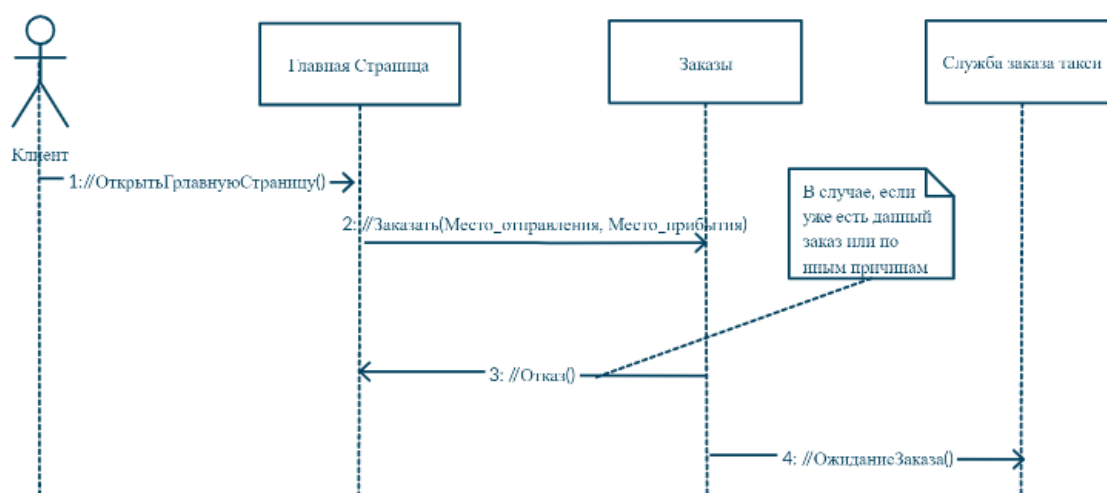


Рисунок 8 – Диаграмма включения “Создать заказ”

Первым делом пользователь выбирает на главной странице создание заказа. В ней авторизированный клиент может заказать такси заранее выбрав в приложении место отправления и место прибытия (2://Заказать (Место\_отправления, Место\_прибытия)). Если пользователя заблокировали, был выбран неверный маршрут или пользователь случайно решил заказать машину, то последует автоматическое отклонение запроса на вызов такси (3:// Отказ ()). Если же все будет хорошо, то пользователь переходит в режим ожидания заказа (4:// ОжиданиеЗаказа()) и ждет ближайшего водителя.

На рисунке 9 показана диаграмма включения “Принять заказ” от “Работа с данными такси”. Когда водитель начинает поиск ближайшего клиента срабатывает метод 1://ПоискЗаказа(). Списки клиентов отображаются в ряд, обновляются с определенным интервалом времени благодаря 2://ПовторныйПоиск() таксист в праве выбирать подходящего человека. Если водителю понравился клиент или же клиент очень близко, то достаточно кликнуть на аватар что бы сработал новый метод (3://ПринятиеЗаказа(НомерЗаказа)), который гласит, что если никто не опередил в выборе данного человека, то заказ принимается, в противном случае (4://Отказ()), поисковик вернется в режим поиска с предупреждением, что запрос отклонен.

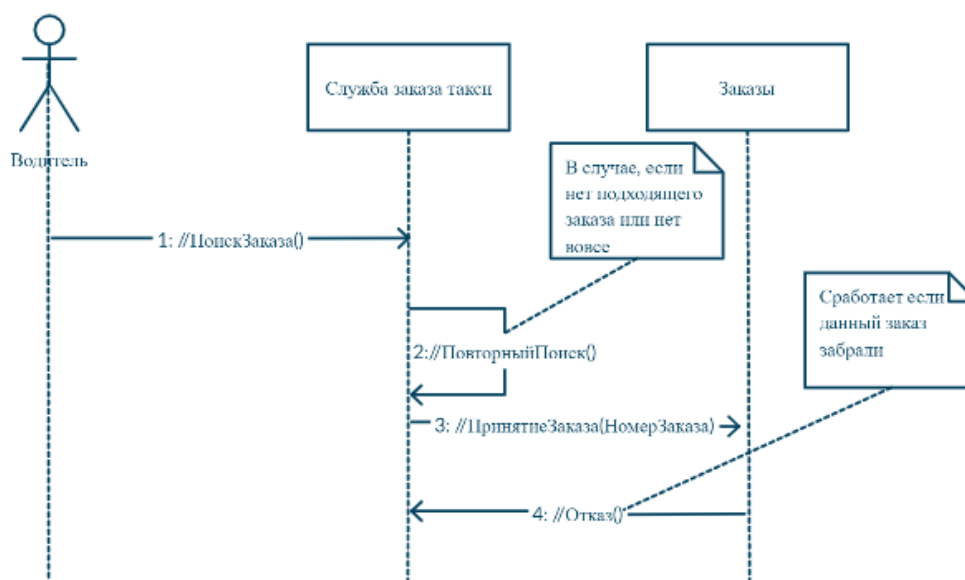


Рисунок 9 – Диаграмма использования “Принять заказ”

На рисунке 10 продемонстрирована диаграмма использования “Управление заказом”. Клиент имеет право узнать все от текущем заказе (1://ИнформацияОЗаказе()) и даже заменить информацию (2://ИзменитьСостояниеЗаказа()), к примеру, добавив комментарий к поездке, чтобы водитель знал, что рядом клиентом поедет собака или очень тяжелый багаж. К плохому готовится тоже можно, например, если пассажир случайно заказал такси не туда, то может в любой момент отменить (3://ОтменитьЗаказ()). Так же предусмотрена оплата онлайн (4://ОплатитьЗаказ()). И самое главное отличие мобильных приложений – отследить передвижение автомобиля (5://ОтследитьЗаказ()).

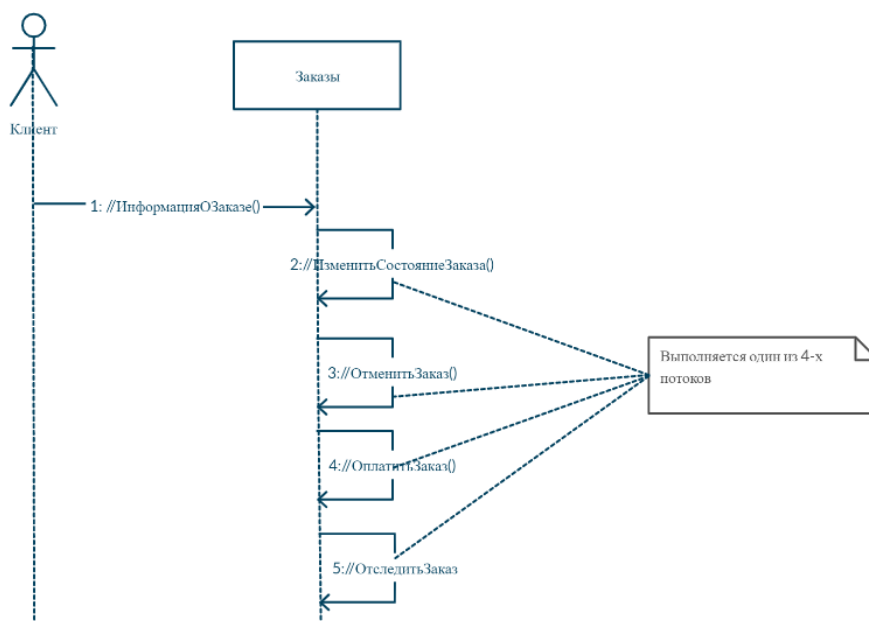


Рисунок 10 – Диаграмма использования “Управление заказом”

На рисунке 11 изображена диаграмма использования “Управление персоналом”. Администратор в праве следить за всеми пользователями приложения (1: //ВывестиВсехПользователей()) и выбирать конкретного человека (2: //ВывестиПользователя(Номер)). Проводя различные манипуляции изменения аккаунта (3://ИзменитьПользователя(Номер)), удаления (4://УдалитьПользователя(Номер)) и блокировки (5://ЗаблокироватьПользователя(Номер)).

На рисунке 12 изображена диаграмма использования “Вопросы в тех. поддержке”. В случае, когда пользователь не может найти ответ на свой вопрос, он выбирает на главной странице вкладку тех.поддержки. Если вопрос был задан (2://МоиТикеты()), то можно найти его в списке, а если его нет, то он может задать его тех. поддержке (3://ОткрытьТикет()) или После входа в тикет программа покажет его содержимое (4://ПросмотрТикета(Номер)) и покажет всю переписку с сотрудником тех. поддержки (5://ПорейтиКДиалогу(Номер)). Если на наш вопрос имеется ответ и он не удовлетворяет требованиям, то можно продолжить диалог (6://НаписатьСообщение()). И ожидать ответа, автоматически обновляя страницу (7://ОбновитьЧат()). После каждого ответа тех. поддержка закрывает тикет (8://ЗакретьТикет(Номер)).

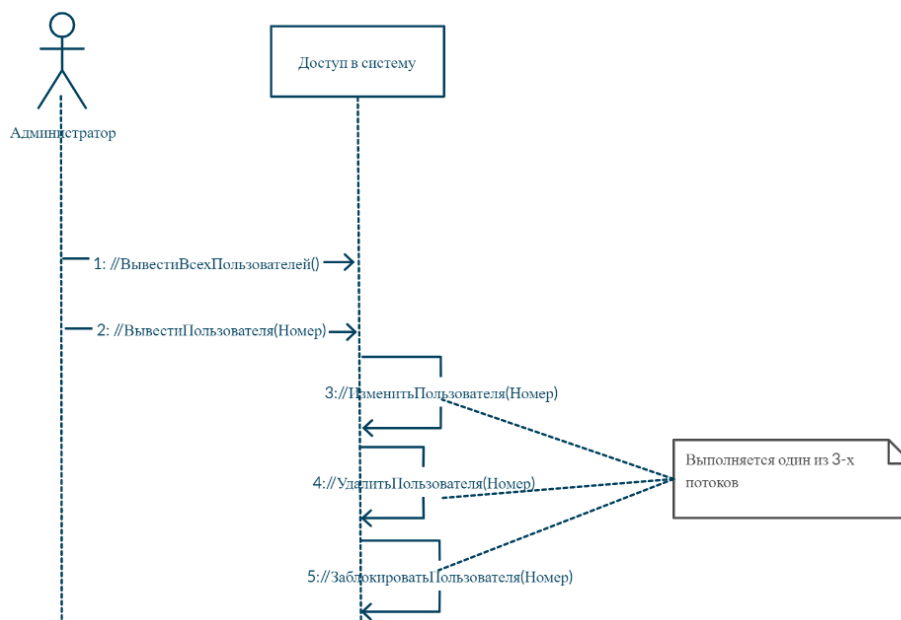


Рисунок 11 – Диаграмма использования “Управление персоналом”

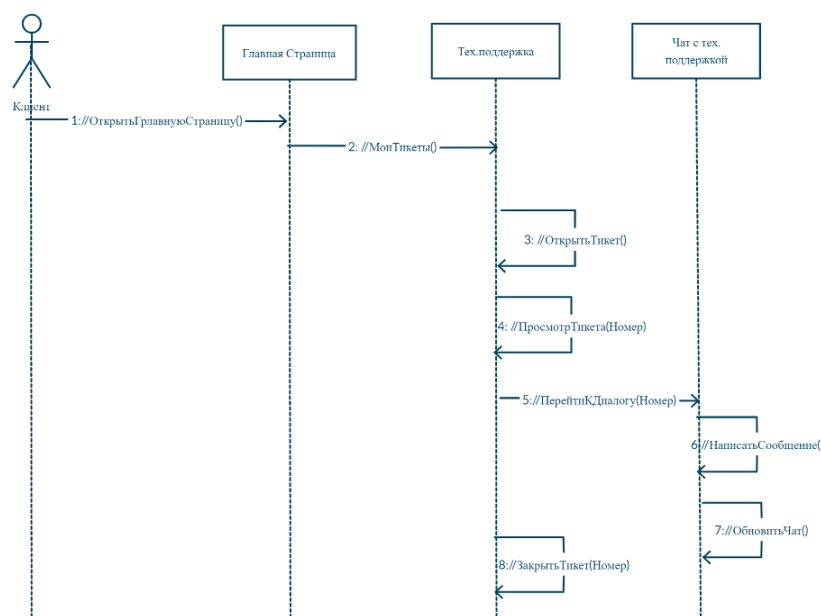


Рисунок 12 – Диаграмма использования “Вопросы в тех. поддержке”

Диаграмма – это графическое представление набора элементов, чаще всего изображенного в виде связного графа вершин (сущностей) и путей (связей). Язык UML включает 13 видов диаграмм, среди которых на первом месте в списке — диаграмма классов, о которой и пойдет речь.

Диаграммы классов (рисунок 13) показывают набор классов, интерфейсов, а также их связи. Диаграммы этого вида чаще всего используются для

Дневник науки | [www.dnevniknauki.ru](http://www.dnevniknauki.ru) | СМИ Эл № ФС 77-68405 ISSN 2541-8327

моделирования объектно-ориентированных систем. Они предназначены для статического представления системы. И чаще всего группируют представляющие интерес наборы сущностей.

Большинство элементов UML имеют уникальную и прямую графическую нотацию, которая дает визуальное представление наиболее важных аспектов элемента.

Класс (Авторизация) имеет следующие методы: ПользовательВСистеме(), ВходаНеБыло(), ОтправитьСМСНомер(), ПовторноОтправитьСМС(), ПользовательВСистеме(), ПользовательНеВСиситеме(). Класс (Регистрация) имеет следующие методы: ПользовательЗарегистрировался(). Класс (Вход) имеет метод: ОпределениеДоступа(). Класс (Рассылка SMS) имеет ОжиданиеСМС(). Класс (Доступ в систему) имеет методы: ИзменитьПользователя(), УдалитьПользователя(), ЗаблокироватьПользователя(), ПереходНаГлавную(). Класс (Главная страница) имеет методы: Заказать(), МоиТикеты(). Класс (Заказы) имеет методы: Отказ(), ОжиданиеЗаказа(), ИзменитьСостояниеЗаказа(), ОтменитьЗаказ(), ОплатитьЗаказ(), ОтследитьЗаказ(). Класс (Служба заказов такси) имеет методы : ПовторитьПоиск(), ПринятиеЗаказа(). Класс (Тех.Поддержка) имеет методы: ОткрытьТикет(), ПросмотрТикета(Номер), ПерейтиКДиалогу(Номер), ЗакретьТиикет(Номер). Класс (Чат с тех.поддержкой) имеет методы: НаписатьСообщение(), ОбновитьЧат().

На основе полученных моделей построена структура баз данных для системы и разработан эскиз интерфейса.

Таким образом мы на основе функционально-структурного анализа исследовали предметную область работы мобильного приложения. Получена IDEF модель системы. Благодаря IDEF модели была построена концептуальная модель мобильного приложения такси, декомпозиция блока “Авторизация пользователя”, “Работа с данными такси”, “Авторизация пользователя”, “Работа с данными такси”, “Техническая поддержка”, “Изменение БД” и “Генерация

Дневник науки | [www.dnevniknauki.ru](http://www.dnevniknauki.ru) | СМИ Эл № ФС 77-68405 ISSN 2541-8327

ответа”. В DFD отразилась “Изменение БД”. По полученной IDE модели построили диаграмму вариантов использования на языке UML.

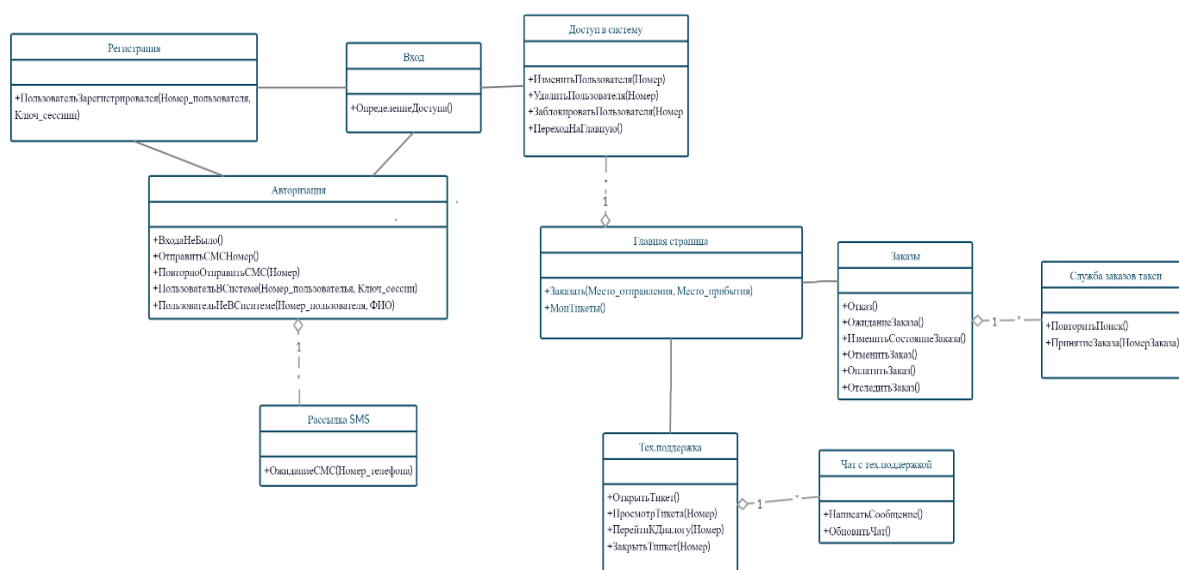


Рисунок 13 – Диаграмма классов

Для разработки модели выбран унифицированный язык моделирования UML в визуальной среде Creately. На ее основе функционально структурного анализа бизнес-процессов выполнено объектное моделирование UML, которое включает следующие методы: авторизация, чат с тех. поддержкой, изменение БД, управление заказами, генерация отчета и работа с данными такси с включениями: принять заказ и создать заказ. Разработка UML модели завершилась построением диаграммы классов описывающей набор классов, интерфейсов в приложении, а также их связи. После проектирования физических и логических моделей данных были построены ER-диаграмма. Так же, для базы данных, разработан эскиз интерфейса программного обеспечения, и описана архитектура логики взаимодействия компонентов программы.

В процессе проектирования показана эффективность использования гибридной модели [10] для разработки мобильного приложения. Разработанные модели мобильного приложения могут быть использованы при реализации



проектов мобильного приложения сервиса такси.

### **Библиографический список:**

1. Владение смартфонами быстро растет по всему миру, но не всегда одинаково [Электронный ресурс] – 2019 – URL: <https://www.pewresearch.org/global/2019/02/05/smartphone-ownership-is-growing-rapidly-around-the-world-but-not-always-equally/> (дата обращения 12.05.2020).
2. Исследование рынка такси [Электронный ресурс] – 2019 – URL: <https://ac.gov.ru/files/content/24166/rynok-taksi-2019-pdf.pdf> (дата обращения 11.05.2020).
3. Фролов Р.Н. Информатизация современного Российского общества: социально-экономические и правовые аспекты // Информационные ресурсы России. 2020. № 4. С. 26-29.
4. Мобильное приложение как инструмент бизнеса [Электронный ресурс] / В. Семенчук Альпина Паблишер 2017г. 240стр Режим доступа: <https://znanium.com/catalog/document?id=333412> (дата обращения 11.05.2020).
5. Современные информационные технологии в управлении сложными социально-экономическими системами: электронная монография / Отв. Ред. Г.Д. Нестеров/ Нестеров Г.Д., Нестерова Н.С., Саакян Р.Р., Бужан В.В., Камалян Р.З. и др. – Краснодар: Новация, 2018. 115 с.
6. Каттави Р.З., Зарипова Р.С. Разработка информационной системы агрохимической компании для управления взаимоотношениями с клиентами // Студенческий. 2019. № 42-1 (86). С. 66-68.
7. Саакян Р.Р., Шпехт И.А. Проектирование информационной системы мониторинга и распределения заказов курьерской службы на основе машинного обучения // Информационные ресурсы России. 2020. № 3. С. 34-38.
8. Цебренок К.Н. Моделирование электронной среды образовательной организации в соответствии с требованиями федеральных стандартов // Информационные ресурсы России. 2018. № 4. С. 38-43.

9. Кубатин Д.В. Моделирование автоматизированной системы агрегации скидок, акций и предложений // Вестник ИМСИТ, Академия ИМСИТ, Краснодар, 2017, № 4(72) - С.22-23.

10. Цебренок К.Н. Гибридное моделирование информационной системы агентства недвижимости // Естественно-гуманитарные исследования, Академия знаний, Краснодар, 2017, № 17(3) - С.28-34.

*Оригинальность 87%*