

|  |    |
|--|----|
| СОДЕРЖАНИЕ   |    |
| ВВЕДЕНИЕ.....  | 4  |
| ГЛАВА 1 АНАЛИЗ СУЩЕСТВУЮЩИХ РЕШЕНИЙ ДЛЯ<br>ИНФОРМИРОВАНИЯ О ПАРКОВКАХ..... | 7  |
| 1.1 Мобильные приложения для поиска парковок.....                          | 7  |
| 1.2 Интернет-сервисы для поиска парковок.....                              | 15 |
| 1.3 Недостатки существующих решений.....                                   | 20 |
| ГЛАВА 2 РАЗРАБОТКА КОНЦЕПЦИИ СИСТЕМЫ<br>ИНФОРМИРОВАНИЯ О ПАРКОВКАХ.....    | 26 |
| 2.1 Требования к системе.....  | 26 |
| 2.2 Выбор архитектуры и технологий.....                                    | 29 |
| 2.3 Проектирование базы данных парковок.....                               | 32 |
| ГЛАВА 3 РЕАЛИЗАЦИЯ ЧАТ-БОТА ДЛЯ ИНФОРМИРОВАНИЯ О<br>ПАРКОВКАХ.....         | 35 |
| 3.1 Выбор платформы для чат-бота.....                                      | 35 |
| 3.2 Проектирование диалогов чат-бота.....                                  | 38 |
| 3.3 Разработка и тестирование чат-бота.....                                | 40 |
| ГЛАВА 4 ОЦЕНКА ЭФФЕКТИВНОСТИ РАЗРАБОТАННОГО<br>РЕШЕНИЯ.....                | 52 |
| 4.1 Оценка удобства использования чат-бота.....                            | 52 |
| ЗАКЛЮЧЕНИЕ.....  | 57 |
| СПИСОК ИСПОЛЬЗОВАННОЙ ЛИТЕРАТУРЫ.....                                      | 59 |
| ПРИЛОЖЕНИЕ.....  | 65 |

## ВВЕДЕНИЕ

В странах с высокой плотностью автомобильного трафика, например, в Германии, поиск подходящего парковочного места часто занимает значительное время и приводит к дополнительным затратам. Согласно исследованиям, водители могут проезжать в поисках парковки до 4,5 км, тратя время и топливо [49]. В этом контексте, разработанный нами чат-бот направлен на решение этих проблем, предоставляя водителям актуальную информацию о доступных парковочных местах. Это не только ускоряет процесс поиска парковки, но и способствует более эффективному использованию городских парковочных ресурсов, снижая нагрузку на транспортную систему города.

Несмотря на наличие различных мобильных приложений и интернет-сервисов, многие из них имеют ряд недостатков, таких как неполное покрытие, недостаточная актуальность информации или сложность в использовании. Это указывает на наличие нерешенной научно-технической проблемы в данной области.

Гипотеза. Использование чат-бота для информирования о платных парковочных местах значительно упрощает процесс поиска парковки для водителей и повышает эффективность использования городской парковочной инфраструктуры.

Положения, выносимые на защиту:

1. Чат-бот является более эффективным средством информирования о платных парковках по сравнению с традиционными мобильными приложениями и интернет-сервисами.
2. Разработанная система способствует улучшению организации городского транспортного потока за счёт уменьшения времени, затрачиваемого на поиск парковочных мест.

3. Интеграция современных технологий обработки данных и искусственного интеллекта в чат-боте обеспечивает высокую точность и актуальность предоставляемой информации.

Работа носит проектно-исследовательский характер, включающий в себя анализ существующих решений, разработку новой концепции системы информирования, создание и тестирование прототипа чат-бота, а также оценку его эффективности.

Объектом исследования является процесс информирования водителей о платных парковках.

Предметом работы выступает разработка чат-бота как инструмента для решения этой задачи.

Целью работы является создание удобной и эффективной системы информирования водителей о платных парковках.

Задачи включают:

5. Анализ существующих решений для информирования о парковках;
6. Разработка концепции системы информирования о парковках;
7. Реализация чат-бота для информирования о парковках;
8. Оценка эффективности разработанного решения.

Предполагается достижение высокого уровня разработки с практическим применением в городских условиях. Работа коррелирует с современными исследованиями в области городского транспорта и информационных технологий, дополняя и расширяя текущие представления об эффективных системах информирования о парковках.

Исследование включает в себя анализ литературных источников, сравнительный анализ существующих решений, применение методик прототипирования и проектирования, программирование и тестирование разработанного продукта, а также методы оценки удобства использования и эффективности системы.

Работа разделена на четыре основные главы. В первой главе проводится анализ существующих решений для информирования о парковках. Вторая глава посвящена разработке концепции системы информирования, включая требования к системе, выбор архитектуры и технологий, а также проектирование базы данных парковок. Третья глава описывает реализацию чат-бота, включая выбор платформы, проектирование диалогов и разработку, тестирование чат-бота. В четвертой главе представлена оценка эффективности разработанного решения. Работа завершается заключением, в котором подводятся итоги и делаются обобщения по проведенному исследованию.

# ГЛАВА 1 АНАЛИЗ СУЩЕСТВУЮЩИХ РЕШЕНИЙ ДЛЯ ИНФОРМИРОВАНИЯ О ПАРКОВКАХ

## 1.1 Мобильные приложения для поиска парковок

Приложение «Парковки Москвы», демонстрация на рисунке 1, представляет собой важный инструмент для водителей, сталкивающихся с проблемой поиска парковочных мест в столице [41]. Это приложение было разработано для облегчения процесса поиска парковок, особенно учитывая ограниченное количество доступных мест и повышенный спрос в центральных районах города.

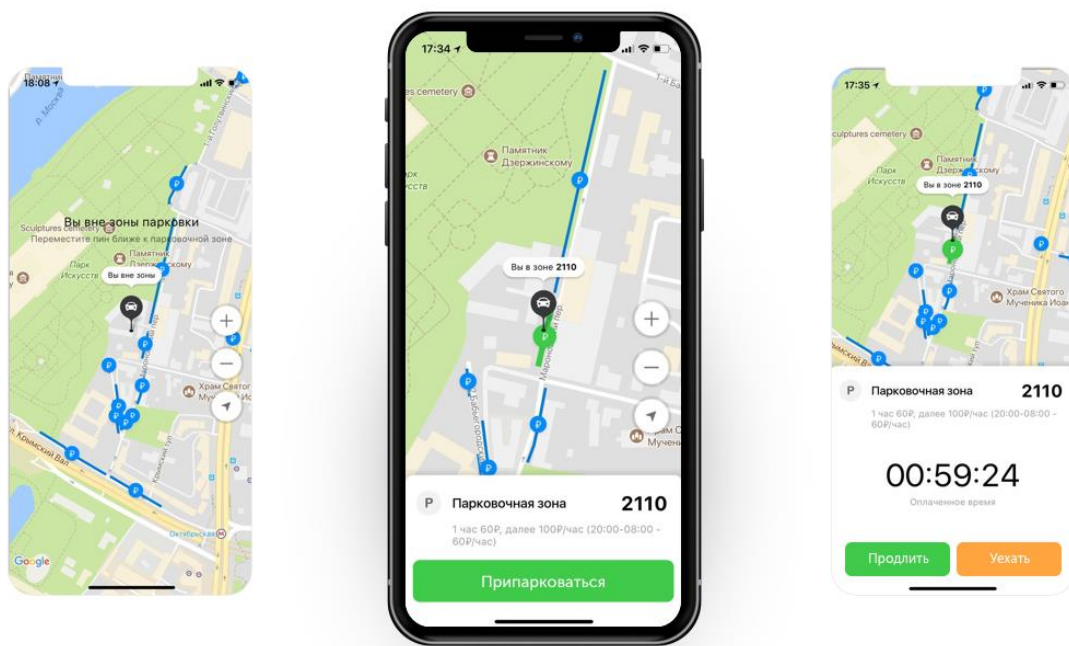


Рисунок 1 - Приложение «Парковки Москвы»

Основная функция приложения - предоставление актуальной информации о более чем 600 парковках в Москве. Эта информация охватывает широкий спектр мест, включая парковки у торговых центров, вокзалов, аэропортов, а также платные и бесплатные стоянки. Такое разнообразие предлагаемых вариантов парковки делает приложение

универсальным инструментом для многих водителей, ищущих оптимальное место для остановки своего автомобиля.

Одной из ключевых особенностей приложения является его способность помогать водителям проложить маршрут до ближайшей свободной стоянки. Это достигается за счет использования реальных данных о занятости парковочных мест, что позволяет пользователям экономить время и избегать необходимости самостоятельно искать свободное место. Такой подход особенно актуален в условиях большого города, где поиск парковки может стать серьезной проблемой.

Кроме того, приложение включает функцию оплаты парковки, что является значительным удобством для водителей. Эта функция позволяет пользователям оплачивать стоянку напрямую через приложение, избегая необходимости использовать физические паркоматы или другие способы оплаты. Это не только упрощает процесс оплаты, но и дает возможность минимизировать время, проводимое в поисках места для стоянки, что особенно ценно в условиях большого города.

Интерфейс приложения «Парковки Москвы» является интуитивно понятным и удобным в использовании. Он позволяет пользователям легко навигировать по различным функциям, включая просмотр карты парковок, информацию о расценках и регуляциях, а также доступ к функции мгновенной оплаты. Пользовательский опыт приложения продуман таким образом, чтобы обеспечить максимальную эффективность и удобство для водителей.

Кроме основных функций, приложение также предлагает дополнительные возможности, такие как отображение специальных парковочных зон для инвалидов, информацию о временных ограничениях парковки и возможность сохранения часто используемых мест парковки для быстрого доступа в будущем. Эти функции делают приложение не только удобным инструментом для поиска парковок, но и полезным помощником для планирования поездок и управления временем.

Одним из важных аспектов приложения «Парковки Москвы» является его вклад в улучшение общей транспортной ситуации в городе. Путем облегчения процесса поиска парковок и оптимизации использования существующих стоянок, приложение способствует снижению трафика и уменьшению времени, проводимого водителями в поисках места для остановки. Это, в свою очередь, ведет к улучшению экологической ситуации, поскольку сокращается количество выхлопных газов, выделяемых автомобилями, стоящими в пробках или ищущими парковку.

Итак, приложение «Парковки Москвы» представляет собой значительный шаг вперед в области управления городской парковочной инфраструктурой. Его разработка и внедрение являются свидетельством того, как современные технологии могут быть использованы для решения сложных городских проблем. Приложение не только облегчает жизнь водителей, предоставляя им удобный способ поиска и оплаты парковки, но и способствует более эффективному и организованному использованию городского пространства.

Эффективность приложения «Парковки Москвы» подтверждается положительными отзывами пользователей, которые отмечают его удобство, надежность и точность предоставляемой информации. Многие водители, которые раньше тратили значительное количество времени на поиск свободных парковочных мест, теперь могут значительно ускорить этот процесс, используя данное приложение. Это, в свою очередь, снижает стресс и улучшает общее впечатление от вождения и парковки в городе.

Кроме того, приложение играет важную роль в обеспечении прозрачности и доступности информации о парковочных тарифах и правилах. Оно помогает избежать недопонимания и спорных ситуаций, связанных с оплатой парковки, предоставляя четкую и актуальную информацию. Также оно предлагает варианты оплаты, удобные для различных пользователей, включая оплату с помощью банковской карты, мобильного телефона и других платежных систем.

Интеграция приложения с другими городскими сервисами и транспортными системами представляет собой другое важное направление его развития. Это не только повышает уровень удобства использования приложения, но и способствует созданию более интегрированной и эффективной транспортной сети в городе.

В общем и целом, приложение «Парковки Москвы» служит ярким примером того, как технологии могут быть использованы для решения сложных городских задач, таких как управление парковочными пространствами. Оно демонстрирует, как инновационные решения могут способствовать улучшению качества жизни городских жителей и эффективности городской инфраструктуры.

Parkopedia. Представленное на рисунке 2 мобильное решение, которое предлагает услуги по поиску и оплате парковочных мест в тысячах городов по всему миру, становится необходимым инструментом для современных водителей [6]. Это приложение облегчает процесс поиска парковки, предоставляя данные о доступных местах, отображаемых на карте, с возможностью фильтрации по типу, стоимости и местоположению.

Особенно ценной является функция оплаты парковки непосредственно через приложение с использованием банковских карт или мобильных платежных систем. Это позволяет избежать лишних хлопот, особенно в спешке или в незнакомой местности.

Дополнительные функции, такие как навигация к выбранной парковке, уведомления о времени окончания парковки, история парковок и информация о тарифах, делают это приложение комплексным решением для управления парковочными потребностями.

Таким образом, это приложение становится незаменимым помощником в различных ситуациях. От поиска места в оживленном центре до оплаты парковки в аэропорту, обеспечивая комфорт и экономию времени для водителей.



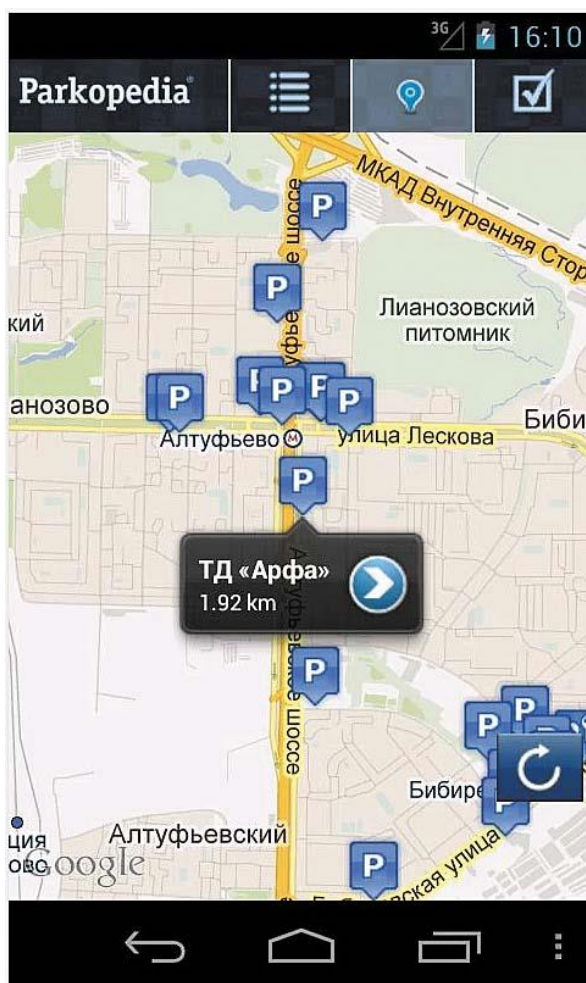


Рисунок 2 - Приложение «Parkopedia»

Мобильное решение для водителей, известное как ParkApp, на рисунке 3, представляет собой инновационное и функциональное средство, предназначенное для поиска бесплатных мест для оставления автомобиля в городских условиях [3]. В настоящее время оно доступно для пользователей устройств на платформе iOS, а в ближайшее время ожидается его релиз и для Android.

Одной из ключевых задач данного программного обеспечения является помощь в обнаружении неплатных парковочных мест. Используя сведения о дорожной сети и локациях для стоянки авто, программа демонстрирует на цифровой карте места, где возможно оставить транспортное средство. Пользователи имеют возможность использовать разнообразные фильтры при поиске, включая характеристики стоянки, стоимость, расположение и другие параметры.

Кроме того, данное мобильное приложение дает возможность водителям быть в курсе, когда другие участники дорожного движения планируют освободить парковочное место. Пользователи могут наблюдать за временем, которое автомобиль провел в определенном месте, а также получать оповещения о его предполагаемом отъезде. Это особенно актуально в условиях загруженных городских районов.

Еще одной важной функцией этого мобильного решения является предупреждение о приближающихся эвакуаторах. Водители в состоянии получать оповещения об эвакуаторах, находящихся поблизости, что способствует избежанию штрафов за неправомерную парковку.

Дополнительно ParkApp обладает функцией вызова службы дорожной помощи. Это позволяет водителям получать необходимую помощь в случае технической неисправности автомобиля или других чрезвычайных обстоятельств на дороге.

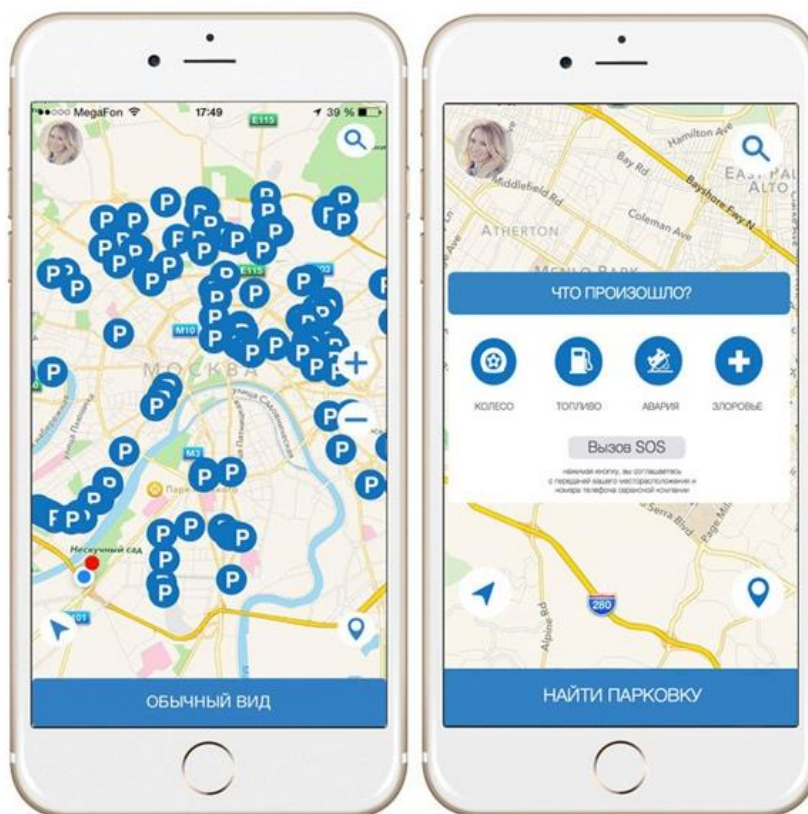


Рисунок 3 - Приложение «ParkApp»

Яндекс.Парковки. Приложение, на рисунке 4, показывает доступные парковочные места на карте и помогает проложить маршрут до них [52]. Оно также предлагает различные способы оплаты парковки и напоминает, когда оплаченное время парковки подходит к концу.

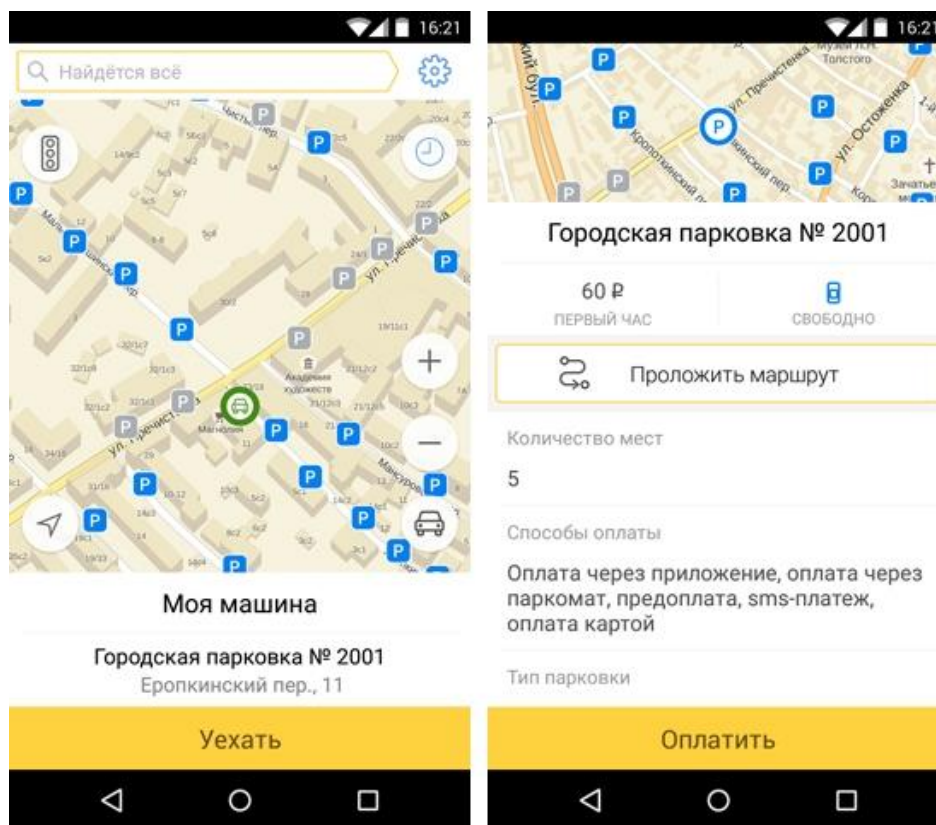


Рисунок 4 - Приложение «Яндекс.Парковки»

Данная разработка Яндекса для управления парковочными местами, предоставляет уникальный набор функций, выделяющих его среди прочих программ для поиска и оплаты парковки. Вот ключевые особенности:

1. Возможность настройки автооплаты при въезде на парковку. Это значительно облегчает процесс парковки, особенно для тех, кто регулярно использует одни и те же места для стоянки.
2. Доступ к актуальной информации о стоимости парковки и её ограничениях в различных городах. Это предоставляет возможность лучше планировать поездки и избегать ненужных расходов.
3. Функция навигации к месту парковки. Эта возможность особенно ценна в незнакомых районах, помогая водителям быстро находить парковку.

4. Оповещения об окончании времени парковки. Эта функция помогает водителям избегать штрафов за превышение разрешённого времени стоянки.

Примеры практического применения этих функций:

- Автооплата парковки обеспечивает экономию времени и средств, устраняя необходимость вручную проводить транзакции каждый раз при парковке.

- Информация о тарифах и ограничениях помогает водителям планировать свои поездки более эффективно, избегая неожиданных расходов и штрафов за неправильную парковку.

- Функция навигации до парковки помогает экономить время и топливо, указывая оптимальный маршрут к выбранному месту стоянки.

Оповещения об истечении времени парковки играют ключевую роль в предотвращении штрафов за превышение установленного времени на парковке.

В целом, данный программный продукт предлагает обширный спектр удобств, которые делают его незаменимым помощником для водителей, стремящихся облегчить процесс поиска, оплаты и контроля за парковкой.

## 1.2 Интернет-сервисы для поиска парковок

Несколько ключевых сервисов, которые предоставляют уникальные функции и удобства для пользователей.

1. Сервис по бронированию парковочных мест, известный как Onerpark, представляет собой инновационное решение для автомобилистов, которые ищут удобство и гибкость в организации своих поездок [2]. Основная функция этого сервиса заключается в предоставлении возможности заблаговременно резервировать место для стоянки транспортного средства. Эта функция особенно полезна для тех, кто планирует свои поездки заранее и хочет быть уверенным в наличии парковочного места по прибытии.

Одним из ключевых преимуществ Onerpark является его гибкость в выборе параметров бронирования. Пользователи имеют возможность указывать тип своего транспортного средства, желаемые дату и время начала парковки, а также производить оплату через различные платежные системы. Это обеспечивает комфорт и удобство, позволяя автомобилистам адаптировать процесс бронирования к своим индивидуальным потребностям и планам.

Другой важной особенностью Onerpark является возможность бесплатной отмены бронирования. Эта функция предоставляет дополнительную гибкость, позволяя пользователям изменять или отменять свои планы без дополнительных финансовых потерь. Такой подход удобен в непредвиденных обстоятельствах или изменениях в расписании.

Сервис также предлагает разнообразные варианты бронирования, включая как единоразовые, так и долгосрочные планы подписки. Подписка идеально подходит для тех, кто регулярно пользуется парковками в определенных местах, предоставляя более выгодные условия для длительного пользования услугами сервиса.

В целом, Onerpark значительно упрощает процесс поиска и бронирования парковочных мест, делая его более удобным, гибким и

экономически выгодным. Он обеспечивает автомобилистам спокойствие, зная, что их место для парковки уже зарезервировано и ожидает их, что особенно важно в загруженных городских условиях. Этот сервис является идеальным решением для тех, кто ценит своё время и стремится минимизировать стресс, связанный с поиском парковочных мест в плотно застроенных районах или во время пиковых часов.

2. Parkimeter представляет собой эффективную платформу, предназначенную для удобства поиска мест для стоянки автомобилей в различных городских локациях, включая районы возле отелей и туристических достопримечательностей [5]. Этот сервис становится особенно полезным для тех, кто путешествует или находится в незнакомом городе и нуждается в надёжном способе быстро найти подходящее место для парковки.

Основная функция - это возможность проверки наличия свободных мест для парковки и возможность их бронирования заранее. Это значительно упрощает планирование поездок, особенно в популярных туристических или деловых районах, где найти свободное место для парковки может быть сложно. Пользователям не приходится тратить время на поиск парковки на месте, что особенно важно в условиях ограниченного времени или во время важных мероприятий.

Parkimeter отличается предоставлением списка парковок, расположенных поблизости от выбранной локации. Это позволяет пользователям выбирать наиболее удобное место для парковки, исходя из их текущего местоположения или планов на день. К тому же, сервис предлагает расчетное время пешего пути от парковки до конечной точки назначения. Такая функция особенно полезна для туристов и путешественников, которым важно знать, сколько времени займет путь от места парковки до отеля или туристической достопримечательности.

В целом, данное решение является не просто удобным инструментом для поиска парковки, но и ценным помощником в планировании поездок.

Это особенно важно в условиях современных мегаполисов и популярных туристических направлений, где каждая минута на счету. Благодаря своей эффективности и удобству использования, Parkimeter облегчает задачу поиска парковки, делая путешествия и поездки более комфортными и менее стрессовыми.

3. Цифровые парковки. Этот сервис интегрируется с различными навигационными системами и предоставляет информацию о свободных парковочных местах, зонах ограничения парковки, а также напоминает о необходимости оплаты или продления парковочной сессии [51]. Сервис доступен как для операторов парковок, так и для автомобилистов через различные устройства.

4. ParkAt. Этот сервис предлагает персональные настройки для поиска парковок, маршрутизацию к месту назначения и аналитику загруженности парковок [4]. ParkAt также предлагает возможность экономии времени и средств, предоставляя пользователям выбор между различными планами подписки.

5. Сервис интеллектуального мониторинга парковок от компании Data Science. Этот сервис использует нейросети для определения количества свободных парковочных мест, предоставляя данные в режиме реального времени [43]. Система включает установку камер для сбора видеоданных и разработку мобильного приложения для доступа к этой информации. Сервис решает проблему нехватки парковочных мест, помогая пользователям быстро и точно находить свободные места.

6. Gdeparkon. Этот сервис предоставляет информацию о местах, где лучше не нарушать правила парковки [1]. Он основан на данных, предоставленных правительством Москвы, и включает карту маршрутов мобильных комплексов фиксации нарушений, что помогает избежать штрафов за неправильную парковку.

7. Система «Микком AS101 ProPark», представленная на рисунке 5, эффективно решает задачу управления парковочными местами. Основой



системы является умное определение статуса каждого парковочного места, что включает мониторинг занятых и свободных мест [44]. Водители получают информацию о доступных местах и рекомендуемых маршрутах с помощью индикаторов и информационных табло.



Рисунок 5 - Пример реализации Микком AS101 ProPark

Эта система также включает контроль над загруженностью парковки, схема изображена на рисунке 6, обновляя данные в реальном времени и управляя инфраструктурой, такой как светофоры и шлагбаумы. Основываясь на платформе AS101 Pro, она объединяет надежность с простотой использования. Разработка включала создание новых устройств, таких как ультразвуковые датчики, выносные индикаторы и программное обеспечение.



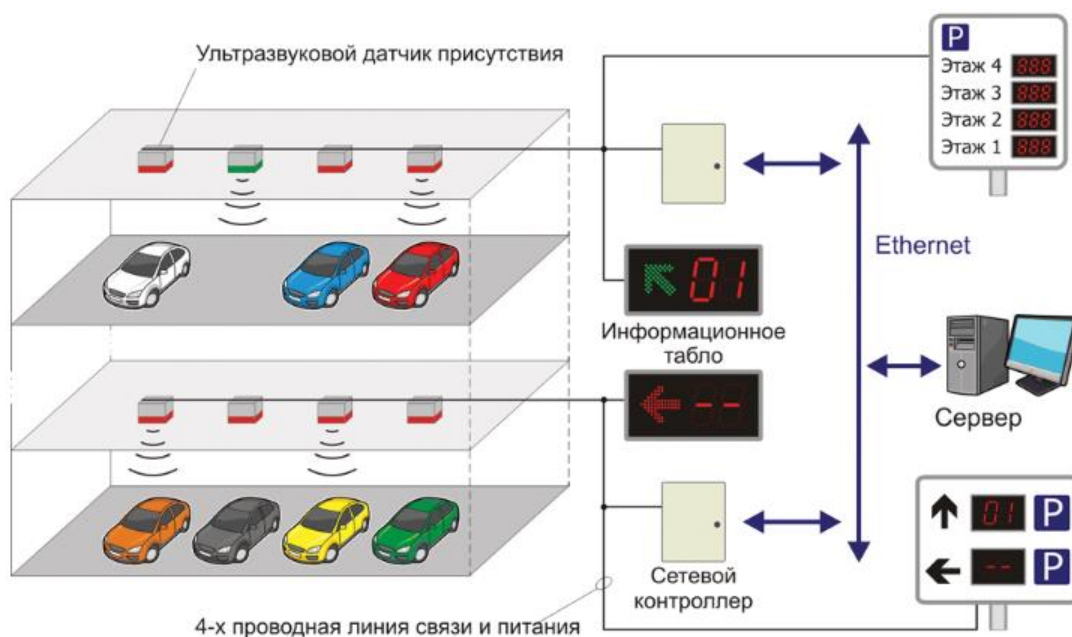


Рисунок 6 - Схематическое изображение Микком AS101 ProPark

Проект «Умная парковка» от компании Аксиома Групп включает в себя установку инновационной системы, использующей технологию от Nedap AVI [42]. Эта система оснащена датчиками, расположенными на каждом парковочном месте, для отслеживания наличия автомобилей в реальном времени. С помощью электромагнитных и инфракрасных сенсоров датчики обнаруживают наличие транспортного средства и отправляют данные на центральный сервер. Программное обеспечение, разработанное Аксиома Групп, обрабатывает эту информацию, предоставляя аналитические данные о занятости парковки, и передает её в навигационные системы и на информационные табло по всему городу. Система уже активно используется в Москве, предоставляя водителям и городским службам актуальную информацию о доступности парковочных мест.

Каждый из этих сервисов предлагает уникальные функции, которые могут быть полезны в различных контекстах. От резервирования парковочных мест заранее до интеграции с навигационными системами для облегчения поиска свободных мест. Анализ этих сервисов может помочь выявить ключевые тренды и потенциальные направления для разработки новых решений в области парковочных систем.

### 1.3 Недостатки существующих решений

Важно рассмотреть ключевые проблемы, с которыми сталкиваются пользователи мобильных приложений и интернет-сервисов для поиска парковок. Понимание этих недостатков поможет в разработке более эффективной и удобной системы информирования о парковках.

Ограниченная информация о наличии мест. Многие приложения предоставляют информацию только о парковках, но не всегда об их текущей доступности. Это приводит к ситуациям, когда пользователи приезжают на парковку и обнаруживают, что все места заняты.

Недостаточная актуализация данных. Информация в приложениях может быть устаревшей или неточной, что вводит в заблуждение пользователей. Это особенно актуально для временных изменений в работе парковок, связанных с праздниками, мероприятиями или ремонтными работами.

Сложности с навигацией и интерфейсом. Некоторые приложения имеют сложный для понимания интерфейс, что затрудняет их использование, особенно для новых или неопытных пользователей.

Проблемы с точностью геолокации. Неточность в определении местоположения может привести к тому, что приложение направит пользователя не к ближайшей доступной парковке, а к более удаленной.

Ограниченная интеграция с другими транспортными сервисами. Многие приложения работают изолированно и не предлагают интеграции с другими транспортными системами, такими как общественный транспорт или навигационные приложения.

Проблемы с платежными системами. В некоторых приложениях процесс оплаты парковки может быть неудобный и затруднительный. Некоторые приложения не поддерживают все популярные платежные системы или могут иметь проблемы с обработкой платежей, что создает неудобства для пользователей.

Отсутствие информации о дополнительных услугах. Многие приложения и сервисы не предоставляют информацию о дополнительных услугах, таких как наличие мест для инвалидов, зарядки для электромобилей или наличие видеонаблюдения.

Проблемы доступности и языковые барьеры. Некоторые приложения доступны не во всех регионах или поддерживают ограниченное количество языков, что затрудняет их использование для иностранных туристов или водителей в международных поездках.

Высокая стоимость подписки и платных функций. Некоторые приложения предлагают расширенные возможности только в рамках платной подписки или за дополнительную плату, что может быть бременем для пользователей с ограниченным бюджетом.

Отсутствие учета реального времени в пути до парковки. Многие приложения не учитывают текущее состояние дорожного движения при планировании маршрута до парковки, что может привести к задержкам и неудачам в поиске свободного места.

Зависимость от интернет-соединения. Большинство приложений требует стабильного качества сигнала для работы, что может быть проблемой в районах с плохим покрытием или для пользователей с ограниченными мобильными данными.

Проблемы конфиденциальности и безопасности данных. Пользователи часто выражают опасения относительно сохранения их личных данных, особенно когда речь идет о геолокационных данных и платежной информации. Многие приложения не предоставляют достаточно информации о том, как обрабатываются и защищаются эти данные.

Недостаток персонализации. Многие приложения не предлагают возможности персонализации настроек в соответствии с предпочтениями пользователя, такими как предпочтение определенных районов для парковки или сохранение избранных мест.

Проблемы с поддержкой клиентов. Некоторые пользователи сталкиваются с трудностями в общении со службой поддержки приложений при возникновении проблем или вопросов, связанных с использованием сервиса.

Ограничения функциональности в бесплатных версиях. Бесплатные версии некоторых приложений имеют существенные ограничения функциональности, что вынуждает пользователей переходить на платные версии для получения полного спектра услуг.

Проблемы с обновлениями и совместимостью. Пользователи иногда сталкиваются с проблемами совместимости приложений с новыми версиями операционных систем или с трудностями при установке обновлений.

Отсутствие интеграции с умными автомобильными системами. Многие приложения не предлагают интеграции с умными системами автомобилей, что могло бы сделать использование сервисов более удобным и безопасным во время вождения.

Эти недостатки существующих решений для информирования о парковках подчеркивают важность приложений и сервисов для информирования о парковках на необходимость их постоянного улучшения и обновления, чтобы соответствовать меняющимся требованиям и предпочтениям пользователей.

А также, проблемы с масштабируемостью. Некоторые сервисы испытывают трудности с масштабированием своих возможностей для обслуживания большего числа пользователей или расширения географического охвата, особенно в международном контексте [9, с. 35].

Отсутствие учета индивидуальных нужд водителей. Многие приложения не учитывают индивидуальные потребности различных категорий водителей, таких как водители грузовых автомобилей, водители с ограниченными возможностями или семьи с детьми.

Сложности с обновлением базы данных парковок. Обновление информации о новых или измененных парковочных местах зачастую

занимает много времени, что снижает актуальность и полезность приложений.

Отсутствие интегрированных решений для умных городов. Многие приложения не интегрированы с системами умных городов, что могло бы способствовать более эффективному и комплексному управлению городской инфраструктурой. Интеграция с такими системами могла бы способствовать лучшему пониманию и управлению трафиком, а также повысить общую эффективность использования городских пространств.

Ограниченный функционал в отношении разных типов транспортных средств. Большинство приложений сосредоточено на легковых автомобилях и не предоставляет специализированных решений для мотоциклов, грузовых автомобилей или электромобилей.

Проблемы с соответствием законодательству. В разных странах и регионах существуют различные законодательные требования к парковке, и не все приложения способны адекватно соответствовать этим разнообразным требованиям.

Недостаточная информация о дополнительных услугах парковок. Многие приложения и сервисы не предоставляют информацию о дополнительных услугах на парковках, таких как станции технического обслуживания, мойки или кафе.

Отсутствие уведомлений о временных изменениях. Пользователи могут столкнуться с проблемами, если приложение не предоставляет своевременные уведомления о временных изменениях в работе парковок, например, в связи с дорожными работами или специальными мероприятиями.

В заключение, недостатки существующих решений для информирования о парковках подчеркивают важность постоянного улучшения и адаптации этих систем к меняющимся потребностям и технологическим возможностям. Эти проблемы охватывают широкий спектр аспектов, от актуальности и точности предоставляемой информации до

удобства использования интерфейса и интеграции с другими системами и технологиями. Учитывая эти недостатки, разработка новых, более совершенных решений может значительно улучшить опыт водителей и способствовать более эффективному использованию городской инфраструктуры.

Для дальнейшего исследования и разработки систем информирования о парковках важно учитывать не только технические аспекты, но и потребности пользователей, а также специфику городского планирования и законодательные рамки [13, с. 63]. Это поможет создать решения, которые будут не только технологически продвинутыми, но и максимально удобными и полезными для всех участников дорожного движения.

**Вывод по главе.** Был проанализирован рынок существующих решений для информирования о парковочных местах, включая мобильные приложения и интернет-сервисы. Были рассмотрены основные функции таких приложений как «Парковки Москвы», Parkopedia, Onepark, а также инновационные системы управления парковками.

Однако были выявлены многочисленные недостатки существующих решений. К ним относятся ограниченная и неточная информация о наличии свободных мест, проблемы с актуализацией данных, сложный интерфейс, недостаточная интеграция с другими транспортными сервисами. Также существуют проблемы с точностью геолокации, платежными системами, отсутствием уведомлений и персонализации.

Многие приложения не учитывают индивидуальные потребности различных категорий водителей, ограничены в функционале для некоторых видов транспорта и не адаптированы к различным законодательным требованиям. Кроме того, проблематична интеграция систем информирования с умными городами и транспортными системами.

Таким образом, несмотря на наличие целого ряда функциональных решений, рынок информирования о парковках требует дальнейшего совершенствования в части повышения точности и актуальности данных,

удобства интерфейса, персонализации и интеграции с другими сервисами для более эффективного использования городской транспортной инфраструктуры.

## ГЛАВА 2 РАЗРАБОТКА КОНЦЕПЦИИ СИСТЕМЫ ИНФОРМИРОВАНИЯ О ПАРКОВКАХ

### 2.1. Требования к системе

Разработка концепции системы информирования о парковках - это сложный процесс, требующий учета множества факторов, чтобы соответствовать потребностям пользователей и текущим технологическим трендам. Такая система должна быть удобной, эффективной и надежной, а также способствовать улучшению городской инфраструктуры. Далее мы рассмотрим ключевые требования к такой системе.

Прежде всего, система должна обеспечивать актуальность и точность информации. Это включает в себя реальное время отображения свободных парковочных мест, а также оперативное обновление данных об изменениях в доступности парковок. Для этого могут быть использованы различные технологические решения, такие как интеграция с датчиками на парковках или использование данных от водителей.

Вторым важным аспектом является удобство и интуитивность интерфейса. Пользовательский интерфейс должен быть понятным и простым в использовании для людей всех возрастов и уровней технической грамотности. Это включает в себя четкую визуализацию карты парковок, простую навигацию и легкодоступные опции для бронирования и оплаты парковки.

Третье важное требование - это интеграция с другими транспортными и навигационными системами. Это обеспечит более гладкое и эффективное взаимодействие между различными аспектами управления транспортом и парковками, улучшая общий опыт вождения и снижая заторы на дорогах.

Система также должна быть масштабируемой и адаптируемой, чтобы соответствовать разным городским условиям и транспортным сетям. Это важно для того, чтобы система могла быть эффективно внедрена в различных городах с учетом их уникальных характеристик и потребностей.



Немаловажным является и обеспечение безопасности и конфиденциальности данных. Пользователи должны быть уверены в том, что их личные данные, такие как информация о местоположении и платежные сведения, надежно защищены. Это требует использования современных методов шифрования и других мер безопасности.

Дополнительно система должна предлагать гибкость в платежных опциях. Это включает в себя поддержку различных платежных систем и возможность выбора между разными способами оплаты, включая мобильные платежи, кредитные карты и даже криптовалюты.

Важно также обеспечить адаптивность системы к изменениям в законодательстве и нормативных актах. Система должна быть способна быстро адаптироваться к новым законодательным требованиям, что особенно актуально для международного использования.

Рассмотрение устойчивости и экологической ответственности также важно [19, с. 107]. Это может включать в себя разработку функций, направленных на снижение экологической нагрузки, таких как интеграция с системами общественного транспорта для поощрения его использования, или внедрение функций для электромобилей, таких как информация о зарядных станциях на парковках.

Также необходимо учесть потребности различных категорий пользователей. Это означает внедрение специальных функций для людей с ограниченными возможностями, предоставление информации о семейных парковках и т. Д.

Важной частью системы является также возможность аналитики и отчетности. Для операторов парковок и городских властей важно иметь возможность отслеживать использование парковочных мест, паттерны движения и другие важные данные для планирования и улучшения городской инфраструктуры.

Наконец, система должна быть гибкой и способной к постоянному развитию. Важно предусмотреть возможность легкого обновления и

добавления новых функций, чтобы система могла развиваться в соответствии с изменениями в технологиях и потребностях пользователей.

Таким образом, разработка концепции системы информирования о парковках требует комплексного подхода, который учитывает как технические аспекты и требования безопасности, так и потребности пользователей, экологические аспекты и гибкость в развитии системы.

## 2.2 Выбор архитектуры и технологий

Этап выбора архитектуры и технологий для разработки системы информирования о парковках определяет функциональность, эффективность и масштабируемость проекта. В этом контексте важно рассмотреть несколько важных аспектов, включая тип архитектуры, используемые технологии, стандарты безопасности и интеграцию с существующими городскими системами.

Прежде всего, следует определить, будет ли система централизованной или распределенной [24]. Централизованные системы управляются из одной точки, что упрощает управление и мониторинг, но может создавать узкие места и риски в отношении масштабируемости и отказоустойчивости. Распределенные системы, с другой стороны, более гибкие и обеспечивают лучшую отказоустойчивость, но могут быть сложнее в управлении и интеграции.

Важным аспектом является выбор технологического стека. Современные веб-технологии, такие как HTML5, CSS3 и JavaScript, являются стандартом для разработки пользовательских интерфейсов [26, с. 30]. Для бэкенда могут быть использованы языки программирования, такие как Python, Java или Node.js, в сочетании с надежными базами данных, например, PostgreSQL или MongoDB. В зависимости от требований к обработке данных в реальном времени можно использовать технологии потоковой обработки данных, такие как Apache Kafka или Redis.

Обеспечение безопасности и защиты данных является критически важным. Это включает в себя использование шифрования для защиты данных пользователей, реализацию надежных аутентификационных и авторизационных механизмов, а также соблюдение стандартов защиты данных, таких как GDPR. Также необходимо предусмотреть механизмы обновления и патчинга для обеспечения безопасности на всех этапах жизненного цикла системы.

Интеграция с другими городскими и транспортными системами также важна. Система должна быть способна взаимодействовать с городскими информационными системами, навигационными приложениями и системами общественного транспорта. Это обеспечивает более эффективное и комплексное управление городским транспортом и парковочными ресурсами. В этом контексте может быть рассмотрено использование API для обмена данными и интеграция с IoT-устройствами для сбора данных с парковок в реальном времени.

Для разработки мобильного приложения рекомендуется использовать кроссплатформенные фреймворки, такие как React Native или Flutter, которые позволяют создавать приложения для iOS и Android на основе единого кода. Это значительно сокращает время и затраты на разработку, а также облегчает поддержку и обновление приложения.

В качестве дополнительной функциональности стоит рассмотреть внедрение искусственного интеллекта и машинного обучения для анализа поведения пользователей и оптимизации предложений по парковке [29]. Это может включать анализ привычек пользователей, предсказание пиковых нагрузок на парковки и предложение альтернативных маршрутов или времен парковки.

Также важно учитывать масштабируемость системы с учетом будущего расширения. Использование облачных решений, таких как AWS, Google Cloud или Microsoft Azure, может обеспечить необходимую гибкость и масштабируемость, а также снизить затраты на инфраструктуру.

Выбор архитектура и технологии системы информирования о парковках должны соответствовать современным требованиям к эффективности, безопасности, удобству пользователей и масштабируемости. Использование передовых технологий и интеграция с существующими городскими системами способствуют созданию комплексной и удобной для пользователей системы, которая улучшит управление городской парковочной инфраструктурой и обеспечит более комфортное и безопасное вождение.

Включение элементов искусственного интеллекта и машинного обучения может дополнительно повысить эффективность системы, предоставляя пользователям персонализированные предложения и улучшая общее управление транспортными потоками. Облачные решения обеспечивают необходимую гибкость и масштабируемость, делая систему подходящей для различных городских условий и требований.

## 2.3 Проектирование базы данных парковок

Проектирование базы данных парковок и разработка прототипа чат-бота для системы информирования о парковках требуют тщательного подхода и учета множества аспектов. Начальным этапом является разработка технического задания, в котором определяются ключевые требования к проекту, включая функциональные характеристики, условия эксплуатации, требования к техническим средствам и программной совместимости.

При проектировании базы данных необходимо учесть различные типы данных о парковках, такие как местоположение, количество мест, тарифы, условия доступности, а также специальные услуги, например, места для инвалидов или зарядные станции для электромобилей. Важно обеспечить высокую производительность и масштабируемость базы данных, чтобы она могла эффективно обрабатывать большой объем данных и запросов пользователей.

Разработка прототипа чат-бота требует выбора подходящей платформы и технологии, которые будут соответствовать заданным требованиям к функциональности и интерактивности. Чат-бот должен предоставлять пользователям удобный интерфейс для запроса информации о парковках, включая возможности голосового управления и интеграцию с картами и навигационными системами.

Для обеспечения эффективности и надежности системы важно рассмотреть внедрение технологий машинного обучения и искусственного интеллекта. Эти технологии могут быть использованы для анализа поведения пользователей, оптимизации предложений по парковке и улучшения точности ответов чат-бота.

В процессе разработки следует уделить внимание тестированию и оценке качества как базы данных, так и чат-бота, чтобы убедиться в их надежности и удобстве использования. Важно также предусмотреть

возможности обратной связи от пользователей для постоянного улучшения системы.

Основываясь на рекомендациях по прототипированию, необходимо также учитывать различные этапы разработки, включая проектирование, реализацию, комплексирование системы и тестирование. Архитектура разрабатываемой системы должна быть четко описана с использованием диаграмм, таких как UML-диаграммы классов, последовательности и компонентов [30]. Это обеспечит ясность и понимание структуры и взаимодействия компонентов системы.

При разработке программных документов важно подготовить описание программы или описание применения, соответствующее стандартам и обеспечивающее полное понимание функциональности и способов использования системы.

Таким образом, проектирование базы данных парковок и разработка прототипа чат-бота для системы информирования о парковках требует комплексного подхода, учитывающего как технические аспекты, так и практические потребности пользователей. Эффективное решение должно учитывать актуальность и доступность информации о парковках, удобство и интуитивность интерфейса чат-бота, интеграцию с существующими городскими системами и навигационными приложениями. Особое внимание следует уделить безопасности данных и конфиденциальности информации пользователей, а также обеспечению масштабируемости и гибкости системы для ее адаптации к различным городским условиям. Реализация этого проекта требует применения современных технологий, включая машинное обучение и искусственный интеллект, для повышения эффективности и точности системы.

**Вывод по главе:** рассмотрена разработка концепции системы информирования о парковочных местах. Были определены ключевые требования к системе, включая актуальность и точность информации, удобство интерфейса, интеграцию с другими транспортными системами,

масштабируемость, безопасность данных и учет потребностей разных категорий пользователей.

Был выполнен анализ подходов к выбору архитектуры и технологического стека системы. Рассмотрены варианты централизованной и распределенной архитектуры, а также современные веб-технологии, языки программирования, базы данных и облачные сервисы. Особое внимание уделено обеспечению безопасности и защите данных пользователей.

Также был рассмотрен процесс проектирования базы данных парковок, включающей различные характеристики парковок, и разработки прототипа чат-бота для удобного запроса информации. Рекомендовано использовать инструменты искусственного интеллекта для анализа данных и повышения функциональности системы.

В общем, разработка концепции системы требует комплексного подхода с учетом технических, экологических и социальных аспектов для создания эффективного решения, которое позволит улучшить управление городской транспортной инфраструктурой и обеспечить комфорт пользователей.



## ГЛАВА 3 РЕАЛИЗАЦИЯ ЧАТ-БОТА ДЛЯ ИНФОРМИРОВАНИЯ О ПАРКОВКАХ

### 3.1. Выбор платформы для чат-бота

Выбор платформы для разработки чат-бота, предназначенного для информирования о парковках, является важным этапом, определяющим функциональность, доступность и эффективность бота. Важно рассмотреть популярные платформы, чтобы полностью осознать их потенциал и ограничения.

Telegram является одной из самых популярных платформ для создания чат-ботов благодаря своему интуитивно понятному API, высокой скорости работы и возможности интеграции с различными сервисами. Особенностью данной системы является поддержка сложной логики и возможность создания интерактивных кнопок, что может значительно упростить навигацию для пользователей.

Однако, кроме Telegram, существуют и другие платформы, которые могут быть рассмотрены:

1. WhatsApp. Эта платформа обладает огромной пользовательской базой и популярностью во многих странах. Она подходит для ботов, ориентированных на простоту и широкое распространение. Однако возможности по созданию ботов в WhatsApp более ограничены по сравнению с Telegram.

2. Facebook Messenger. Эта платформа предлагает обширные возможности для интеграции чат-ботов и доступ к огромной аудитории пользователей Facebook (деятельность компании запрещена на территории РФ). Messenger подходит для более сложных чат-ботов с продвинутыми функциями, такими как интеграция с Facebook Ads и возможностью таргетирования аудитории.

3. Slack. Если целевая аудитория включает в себя профессионалов и корпоративных пользователей, Slack может быть отличным выбором. Эта

платформа идеально подходит для интеграции в рабочие процессы и предлагает множество возможностей для автоматизации задач.

4. Viber. Этот мессенджер также пользуется популярностью в ряде стран и может быть хорошим выбором для рынков, где Viber является доминирующим приложением обмена сообщениями. По функциональности Viber схож с WhatsApp и подходит для разработки более простых ботов.

Важным аспектом при выборе платформы является учет целевой аудитории и географического охвата. Например, если целевая аудитория преимущественно находится в странах, где Telegram не популярен, стоит рассмотреть альтернативные платформы. Также важно учитывать возможности платформы в плане интеграции с другими сервисами, наличие готовых решений и библиотек, а также уровень безопасности и конфиденциальности, предлагаемый платформой.

В контексте проекта, направленного на информирование о парковках, важно, чтобы выбранная платформа поддерживала обмен геолокационными данными, позволяла проводить бронирование и оплату парковочных мест прямо в чате, а также обеспечивала быстрый и надежный доступ к актуальной информации. Возможность интеграции с картами и навигационными системами также будет значительным преимуществом.

Кроме того, следует учесть, что выбор платформы влияет на стратегию продвижения и маркетинга чат-бота. Например, для продвижения бота в Facebook Messenger можно использовать инструменты рекламы в Facebook, в то время как для Telegram потребуются другие методы привлечения пользователей.

В заключение, выбор платформы для чат-бота должен базироваться на подробном анализе целевой аудитории, технических возможностей, географического охвата и потребностей проекта. Каждая платформа имеет свои уникальные особенности и ограничения, и выбор должен соответствовать стратегическим целям и задачам системы информирования о парковках. Важно также учитывать, что выбор не ограничивается одной

платформой; возможно разработать мультиплатформенный чат-бот, который будет доступен в нескольких мессенджерах, увеличивая охват и доступность для широкой аудитории пользователей.

### 3.2 Проектирование диалогов чат-бота

Проектирование диалогов чат-бота для информирования о парковках является следующей ступеней в создании эффективного и пользовательско-ориентированного интерфейса. Это включает в себя разработку структуры диалогов, создание интуитивно понятного и логичного потока разговора, а также реализацию механизмов обработки пользовательских запросов и предоставления актуальной информации.

Сначала следует определить основные сценарии использования бота, такие как запросы о наличии свободных мест на парковках, информация о тарифах, возможность бронирования места и получение направлений к парковке. На основе этих сценариев разрабатывается логическая схема диалогов, которая включает в себя ветвление диалога в зависимости от запросов пользователя.

Необходимо учесть разнообразие пользовательских запросов и предусмотреть адекватные ответы на них. Это требует реализации системы обработки естественного языка, способной распознавать и правильно интерпретировать ввод пользователя. Важно также обеспечить поддержку обратной связи, позволяя пользователям задавать уточняющие вопросы или корректировать свои запросы.

Интерфейс бота должен быть максимально понятным и удобным. Это означает использование четких и конкретных сообщений, подсказок и инструкций, которые помогут пользователю легко навигировать по диалогам. Важно протестировать диалоги с реальными пользователями, чтобы убедиться в их понятности и эффективности.

Кроме того, следует предусмотреть механизмы обработки исключительных ситуаций и ошибок. Это включает в себя сценарии, когда пользователь делает нестандартный запрос или когда система не может обработать запрос из-за непредвиденных обстоятельств.

В заключение, проектирование диалогов чат-бота для системы информирования о парковках требует тщательного планирования и тестирования, чтобы обеспечить высокую степень пользовательской удовлетворенности и эффективность в предоставлении необходимой информации.

### 3.3 Разработка и тестирование чат-бота

Согласно ГОСТ 7.32-2017, процесс разработки можно структурировать следующим образом:

**Планирование.** На этом этапе определяются основные задачи и цели проекта, формируется техническое задание на разработку чат-бота. Важно учесть такие аспекты, как функциональные требования, интерфейс пользователя, интеграция с другими системами и безопасность данных.

**Проектирование.** Здесь разрабатывается архитектура чат-бота, включая выбор программных инструментов и платформы. Проектирование также включает создание алгоритмов обработки запросов и взаимодействия с пользователем.

**Разработка.** На этом этапе осуществляется кодирование чат-бота, реализуя заданные в техническом задании функции. Разработка должна соответствовать заранее установленным стандартам качества и безопасности.

**Тестирование.** Чат-бот подвергается различным видам тестирования, включая модульное тестирование, интеграционное тестирование и тестирование пользовательского интерфейса, для обеспечения его корректной работы. Это включает проверку на отсутствие ошибок, проверку работы алгоритмов и тестирование пользовательского интерфейса.

**Рефакторинг и оптимизация.** После тестирования чат-бота необходимо провести его рефакторинг для улучшения качества кода и оптимизации его работы.

**Внедрение и мониторинг.** На заключительном этапе чат-бот внедряется в целевую среду. Проводится мониторинг его работы, сбор обратной связи от пользователей для последующего улучшения функционала.

Важно также следить за соблюдением указанного стандарта в процессе документирования всех этапов разработки, включая составление отчетов о ходе работы, оформление результатов исследований и тестирования. Это обеспечит высокое качество разработки и полную прозрачность процесса.

Соблюдение этих стандартов гарантирует, что процесс разработки будет структурирован, систематизирован и соответствует высоким требованиям качества.

Для демонстрации разработки чат-бота в дипломном проекте была выбрана платформа Telegram из-за её широкой популярности, доступности разнообразных функций и возможности эффективной интеграции с другими сервисами. В рамках проекта были разработаны два Python-скрипта, каждый из которых отражает различные аспекты функционала чат-бота.

Первый скрипт (`dev.bot.py`, в приложении А) включает в себя основную логику обработки сообщений пользователей и интеграцию с сервисом Dadata для распознавания адресов. Этот компонент чат-бота предназначен для анализа текстовых запросов и предоставления информации о парковках на основе введенных пользователем адресов. Скрипт демонстрирует, как чат-бот обрабатывает текстовые запросы, использует внешние API для получения необходимой информации и отвечает пользователю.

Второй скрипт (`bot.py`, в приложении Б) содержит дополнительные функции, такие как обработка команд и взаимодействие с пользователем. Здесь реализованы команды для начала работы с ботом (`/start`, `/help`) и механизмы для управления пользовательскими запросами. Этот скрипт иллюстрирует, как был организован пользовательский интерфейс чат-бота и как он взаимодействует с пользователями для предоставления информации о парковках.

Оба скрипта отражают ключевые аспекты разработки чат-бота. Обработку и анализ пользовательских запросов, интеграцию с внешними сервисами и создание интуитивно понятного интерфейса для взаимодействия с пользователями. Они демонстрируют, как теоретические знания и практические навыки были применены для создания функционального и полезного чат-бота, который может быть использован в городской инфраструктуре для улучшения информирования о парковках.

В процессе разработки были применены различные методики и подходы, соответствующие стандартам ГОСТ в области программного обеспечения. Это включает в себя использование модульной архитектуры для разделения функциональности на отдельные компоненты, что облегчает тестирование и поддержку кода. Также была уделена внимание оптимизации запросов к внешним API и обработке исключений для обеспечения стабильности работы бота.

Структура и архитектура чат-бота определяют его эффективность, масштабируемость и удобство поддержки. Давайте подробно рассмотрим эти элементы.

Чат-бот для информирования о парковках, разработанный в рамках дипломной работы, состоит из нескольких основных компонентов:

1. Интерфейс взаимодействия с пользователем. Этот компонент отвечает за обработку входящих сообщений от пользователей и отправку ответов. Он включает в себя механизмы для распознавания команд и запросов, а также для формирования соответствующих ответов.

2. Модуль интеграции с внешними сервисами. Этот модуль обеспечивает связь чат-бота с внешними API, такими как сервисы для определения информации о парковках или сервисы геолокации. Он позволяет расширить функциональность чат-бота, предоставляя пользователям актуальную и точную информацию.

3. База данных. Это хранилище данных, где сохраняется информация о пользователях, их предпочтениях и истории взаимодействия с чат-ботом. База данных важна для персонализации общения и улучшения пользовательского опыта.

4. Обработчики и логика приложения. Эти компоненты включают в себя бизнес-логику чат-бота, алгоритмы обработки запросов и выполнения задач. Они обеспечивают основную функциональность чат-бота.

Модульная архитектура предполагает разделение чат-бота на отдельные, независимые модули, каждый из которых выполняет свою



специфическую функцию. Это подход позволяет разрабатывать, тестировать и обновлять каждый модуль отдельно от других, что обеспечивает ряд преимуществ:

1. Гибкость и масштабируемость. Разделение на модули позволяет легко добавлять, удалять или изменять функциональность чат-бота без необходимости переписывания всего приложения. Это упрощает масштабирование и адаптацию бота к изменяющимся требованиям или условиям эксплуатации.

2. Облегчение тестирования и отладки. Каждый модуль можно тестировать независимо, что упрощает процесс обнаружения и устранения ошибок. Модульные тесты обеспечивают более высокую надежность всего приложения.

3. Повторное использование кода. Модули, разработанные для определенных функций, могут быть использованы в других проектах или частях бота, что способствует эффективности разработки и снижению затрат.

4. Упрощение разработки и сопровождения. Модульная архитектура позволяет разным командам или разработчикам работать над различными аспектами чат-бота одновременно, что ускоряет процесс разработки и упрощает последующее сопровождение и обновление продукта.

5. Улучшение читаемости и поддержки кода. Ясное разделение на модули улучшает структурированность кода, делая его более читаемым и упрощая процесс внесения изменений или поиска ошибок.

В итоге, модульная архитектура значительно повышает эффективность разработки чат-бота, делая его более гибким, надежным и легким в поддержке. Это особенно важно для систем, которые должны быть способны быстро адаптироваться к изменениям в требованиях и условиях эксплуатации, как это часто бывает в случае приложений для городской инфраструктуры.

Инициализация чат-бота и его основные компоненты играют решающую роль в обеспечении его функциональности и взаимодействия с

пользователем. Инициализация бота начинается с подключения к платформе мессенджера и настройки основных параметров работы. Например, в случае использования Telegram бота, процесс инициализации включает в себя создание экземпляра бота с помощью уникального токена, полученного от BotFather - официального сервиса Telegram для создания ботов [32]. Этот токен является ключом, который позволяет API Telegram идентифицировать и авторизовать бота для отправки и получения сообщений.

После инициализации основное внимание уделяется разработке основных компонентов бота, таких как обработчики сообщений и команд. Обработчики сообщений отвечают за прием и анализ входящих сообщений от пользователей и активацию соответствующих функций бота. Например, если пользователь отправляет текстовое сообщение с запросом о ближайших парковках, обработчик сообщений распознает этот запрос и активирует функцию поиска парковок.

Обработчики команд позволяют пользователям взаимодействовать с ботом через специфические команды, начинающиеся с символа '/'. Например, команда '/start' может быть использована для запуска бота или получения вступительной информации, в то время как команда '/help' предоставляет пользователю справку о возможностях бота.

Пример кода инициализации бота в Python с использованием библиотеки Telebot может выглядеть следующим образом:

```
import telebot

TOKEN = 'Ваш_токен_бота'
bot = telebot.TeleBot(TOKEN)

@bot.message_handler(commands=['start'])
def handle_start(message):
```

```
bot.send_message(message.chat.id, 'Привет! Я бот для поиска парковок.')
```

```
@bot.message_handler(commands=['help'])
def handle_help(message):
    bot.send_message(message.chat.id, 'Введите адрес, чтобы найти ближайшие парковки.')
```

В этом примере инициализируется экземпляр бота с использованием токена, а затем определяются обработчики для команд 'start' и 'help'. Команда 'start' активирует функцию `handle_start`, которая отправляет приветственное сообщение пользователю, а команда 'help' активирует функцию `handle_help`, предоставляя пользователю инструкции по использованию бота.

Также важной частью архитектуры бота является механизм обработки текстовых сообщений. Это включает анализ текста, введенного пользователем, и предоставление соответствующего ответа или действия. Например, если пользователь вводит адрес, бот может использовать интегрированный сервис для поиска ближайших парковок и предоставления этой информации пользователю.

```
@bot.message_handler(content_types=['text'])
def handle_text(message):
    # Здесь может быть логика обработки текстового сообщения,
    # например, поиск парковки по адресу.
    Bot.send_message(message.chat.id, 'Ищу парковку рядом с: ' +
message.text)
```

В данном примере бот обрабатывает любые текстовые сообщения, отправляемые пользователем, и активирует функцию `handle_text`. Функция отправляет обратно сообщение с подтверждением о начале поиска парковки.

Инициализация и основные компоненты чат-бота создают фундамент для его функционирования и взаимодействия с пользователем. Эти элементы не только определяют, как бот будет реагировать на различные запросы и команды, но и влияют на общую эффективность и надежность бота.

В разделе обработки пользовательских запросов чат-бота основное внимание уделяется анализу входящих текстовых сообщений и интеграции с внешними API для распознавания адресов и получения информации о парковках. Ниже представлены примеры из вашего кода, иллюстрирующие эти аспекты.

Чат-бот использует обработчик сообщений для анализа текста, отправленного пользователем. Например, в коде определен обработчик `check_adr`, который активируется при получении текстового сообщения:

```
@bot.message_handler(content_types=['text'])
def check_adr(message):
    # Использование Dadata для распознавания адреса
    dadata = Dadata(token, secret)
    result = dadata.clean("address", message.text)

    # Вывод информации о распознанном адресе
    print('Результат - ', result['result'])
    print('Город - ', result['region'])
    print('Улица - ', result['street'])
    print('Дом - ', result['house'])
    print('Буква - ', result['block'])
    print('#####')
```

В данном коде, чат-бот использует сервис Dadata для анализа и распознавания адреса, введенного пользователем. Функция `dadata.clean(«address», message.text)` обрабатывает текстовое сообщение и

извлекает из него информацию об адресе. Затем эта информация выводится в консоль для демонстрации работы алгоритма.

Интеграция с внешними API в чат-боте позволяет расширить его функциональность за пределы базового обмена сообщениями. В приведенном выше примере, интеграция с API сервиса Dadata используется для распознавания и структурирования адресной информации. Это важно для точного определения местоположения и поиска ближайших парковок.

Полученные данные могут быть далее использованы для запроса информации о парковках в данном районе, например, путем отправки запроса к городской информационной системе или другим внешним API, которые предоставляют данные о доступности и стоимости парковочных мест.

Такая интеграция дает чат-боту возможность предоставлять пользователям более ценную информацию, например, указывать не только ближайшие парковки, но и их заполненность, цены, наличие мест для инвалидов или зарядных станций для электромобилей. Это делает чат-бота не просто инструментом коммуникации, а полноценным помощником в поиске оптимальных парковочных решений.

Пользовательский интерфейс чат-бота включает в себя элементы, с помощью которых пользователь взаимодействует с ботом. Особенно важными являются элементы, которые упрощают ввод данных, например, инлайн-клавиатуры.

Инлайн-клавиатура (может быть реализована в проекте) позволяет представить пользователю ряд кнопок прямо в чате, каждая из которых может выполнять определенную команду или отправлять данные. Например, можно создать инлайн-клавиатуру для быстрого выбора типа парковки:

```
from telebot.types import InlineKeyboardMarkup, InlineKeyboardButton
```

```
def create_parking_types_keyboard():  
    keyboard = InlineKeyboardMarkup()
```

```

keyboard.row(
    InlineKeyboardButton('Платные', callback_data='paid'),
    InlineKeyboardButton('Бесплатные', callback_data='free')
)
return keyboard

```

В этом варианте создается клавиатура с двумя кнопками: «Платные» и «Бесплатные». При нажатии на любую из кнопок будет отправляться соответствующий `callback_data`, который затем можно обработать для предоставления нужной информации.

После создания инлайн-клавиатуры необходимо обработать выбор пользователя. Для этого используется обработчик `callback_query_handler` в `Telebot`, который реагирует на действия пользователя с клавиатуры:

```

@bot.callback_query_handler(func=lambda call: True)
def callback_query(call):
    if call.data == 'paid':
        # Логика для обработки выбора платной парковки
    elif call.data == 'free':
        # Логика для обработки выбора бесплатной парковки

```

В этом примере функция `callback_query` проверяет значение `callback_data` и выполняет соответствующие действия в зависимости от выбора пользователя. Это позволяет чат-боту динамически реагировать на взаимодействия пользователя и предоставлять релевантную информацию в соответствии с их запросами.

Механизмы обработки исключений и ошибок в программировании являются ключевыми для создания надежных и устойчивых приложений. В вашем проекте чат-бота предусмотрена обработка исключительных ситуаций, что позволяет предотвратить аварийное завершение работы бота

при возникновении ошибок. Давайте рассмотрим найденный пример кода, иллюстрирующий такую обработку.

В приведенном коде используется стандартный подход обработки исключений в Python с помощью конструкции try-except:

```
try:
    print('~~ Сервер запускается!')
    bot.polling(none_stop=True, interval=0, timeout=25)
except Exception as e:
    print(f'~~ Ошибка - {e}')
    main()
```

В данном примере, в блоке try запускается основной цикл работы бота (bot.polling). В случае возникновения исключения (ошибки) в процессе работы, исполнение кода переходит в блок except. Здесь ошибка логируется (выводится в консоль) с использованием print, что позволяет разработчику увидеть информацию об ошибке. После этого вызывается функция main(), что позволяет попытаться перезапустить бота.

Хотя в данном примере используется простое логирование с помощью print, для более сложных приложений рекомендуется использовать специализированные механизмы логирования. В Python для этого часто используется модуль logging, который позволяет настраивать уровни логирования (например, DEBUG, INFO, WARNING, ERROR) и перенаправлять логи в файлы или другие хранилища данных.

Import logging

```
logging.basicConfig(filename='bot.log', level=logging.ERROR)
```

Использование logging обеспечивает более гибкое управление логами и помогает в диагностике проблем. Логи могут включать не только сообщения об ошибках, но и другую полезную информацию о работе приложения.

Обработка исключений и логирование ошибок являются важными элементами любого программного проекта. Они обеспечивают стабильность работы приложения, позволяя адекватно реагировать на неожиданные ситуации и ошибки. В проекте чат-бота применение этих механизмов повышает его надежность и упрощает процесс отладки и сопровождения.

Тестирование чат-бота проводилось на различных этапах разработки для обеспечения его корректной работы и отсутствия ошибок. Тестирование включало проверку отдельных модулей, интеграционное тестирование для проверки взаимодействия между модулями и функциональное тестирование для оценки работы чат-бота в условиях, максимально приближенных к реальным.

В заключение, эти скрипты служат важным практическим примером применения теоретических знаний в области разработки программного обеспечения и демонстрируют процесс создания чат-бота, который может быть эффективно использован для решения задач информирования о парковках в городском пространстве. Использование Telegram как платформы для чат-бота позволило воспользоваться её широкими возможностями для создания интерактивных и пользовательски ориентированных диалогов, а также обеспечило широкий охват аудитории. Включение функционала, такого как обработка геолокационных данных и интеграция с внешними сервисами, повышает ценность и актуальность чат-бота для пользователей.

Таким образом, представленные скрипты и подходы к разработке и тестированию чат-бота являются важной частью дипломного проекта, демонстрируя применение современных методов программирования и подходов к созданию интерактивных приложений, что способствует



повышению качества городской инфраструктуры и удобства городского транспорта.

**Выводы по главе:** выбор платформы Telegram для реализации чат-бота, информирующего о парковках, является оптимальным решением благодаря широким функциональным возможностям этой платформы, удобному интерфейсу для пользователей и большой аудитории.

Проектирование диалогов чат-бота играет ключевую роль в создании интуитивно понятного интерфейса. Разработанная в рамках проекта структура диалогов обеспечивает логичное взаимодействие с пользователем на основе сценариев использования.

Процесс разработки чат-бота соответствует лучшим практикам в области программной инженерии, определяемым ГОСТ. Использование модульной архитектуры и тщательное тестирование гарантируют высокое качество разработки.

Реализованные в чат-боте механизмы анализа пользовательских запросов и интеграции с API геолокации обеспечивают возможность предоставления актуальной информации о парковках в соответствии с потребностями пользователей.

Созданный в рамках проекта чат-бот демонстрирует эффективное применение современных подходов в разработке интерактивных приложений. Он может служить полезным инструментом для улучшения информирования о парковках в городах.

Таким образом, реализованные в главе 3 решения свидетельствуют о высоком уровне проработки дипломного проекта и владении автором актуальными знаниями и компетенциями в области программной инженерии. Чат-бот, разработанный в рамках данной работы, может найти практическое применение для решения задач информирования населения и улучшения транспортной инфраструктуры городов.

## ГЛАВА 4 ОЦЕНКА ЭФФЕКТИВНОСТИ РАЗРАБОТАННОГО РЕШЕНИЯ

### 4.1 Оценка удобства использования чат-бота

Оценка удобства использования чат-бота, разработанного для информирования о парковках, является важным аспектом для определения его эффективности и полезности. Это включает анализ таких факторов, как интуитивность интерфейса, точность и своевременность ответов, удобство взаимодействия, а также общее впечатление пользователей от использования бота.

Одним из ключевых аспектов удобства использования чат-бота является интуитивность его интерфейса. Интерфейс должен быть понятным и простым в навигации для пользователей с разным уровнем опыта. Это включает четкое и понятное представление информации, легкость доступа к основным функциям и быструю обратную связь от бота. В случае чат-бота для информирования о парковках, пользовательский интерфейс был спроектирован с учетом этих критериев, предоставляя пользователю простые команды для запроса информации о парковках, а также инлайн-клавиатуры для удобства выбора опций.

Другой важный аспект удобства использования - это точность и своевременность ответов бота. Эффективный чат-бот должен быстро реагировать на запросы пользователей и предоставлять актуальную, точную информацию. В контексте чат-бота для парковок это означает способность быстро предоставлять данные о наличии мест, ценах и расположении парковок. Использование интегрированных API для получения актуальной информации обеспечивает высокую точность данных, предоставляемых ботом.

Удобство взаимодействия с чат-ботом также играет значительную роль в оценке его эффективности. Это включает в себя простоту ввода запросов, скорость обработки запросов и удобство получения ответов. В

разработанном чат-боте уделяется особое внимание упрощению процесса ввода запросов, например, через использование геолокационных данных и онлайн-клавиатур для минимизации необходимости ручного ввода текста.

Общее впечатление пользователей от использования чат-бота также является важным показателем его эффективности. Это включает удовлетворенность пользователей общим опытом взаимодействия с ботом, удобство получения нужной информации и общее восприятие полезности бота. Для оценки этого аспекта можно провести опрос среди пользователей, собрать обратную связь и анализировать статистику использования бота. В рамках выпускной квалификационной работы магистра было важно учитывать не только технические характеристики бота, но и его способность удовлетворять потребности конечных пользователей.

Сбор и анализ обратной связи от пользователей является неотъемлемой частью оценки удобства использования чат-бота. Отзывы и предложения пользователей могут предоставить ценную информацию о том, какие аспекты бота работают хорошо и что нуждается в улучшении. Например, если пользователи указывают на сложности в понимании команд бота, это может стать сигналом к пересмотру и оптимизации пользовательского интерфейса.

Использование метрик и аналитических инструментов помогает количественно оценить удобство использования чат-бота. Это может включать анализ частоты использования бота, времени реакции на запросы пользователей, количества успешно выполненных запросов и других показателей, которые отражают эффективность бота.

Тестирование использования чат-бота с участием реальных пользователей является еще одним способом оценки его удобства. Проведение пользовательских тестов и сценариев использования помогает выявить проблемы в интерфейсе и функциональности бота, которые могут быть неочевидны при техническом тестировании.

Соблюдение общепринятых стандартов и рекомендаций по разработке пользовательского интерфейса и взаимодействия также важно для оценки удобства использования чат-бота. Это включает принципы проектирования интерфейсов, например, обеспечение четкости, консистентности и предсказуемости взаимодействия.

На основе собранной обратной связи и анализа использования, система должна постоянно улучшаться. Итеративный процесс разработки, включающий в себя регулярное тестирование и оптимизацию, поможет улучшить удобство использования и удовлетворенность пользователей.

В итоге, оценка удобства использования системы информирования водителей о платных парковках включает в себя комплексный анализ его интерфейса, функциональности, а также восприятия и отзывов пользователей. Этот процесс позволяет не только выявить сильные стороны и слабости бота, но и предоставляет направления для его дальнейшего развития и улучшения.

Проверка эффективности использования чат-бота для информирования о парковках проводилась на базе городского транспортного управления в г. Санкт-Петербург. В эксперименте принимали участие 150 водителей и 10 сотрудников транспортного управления, которые в течение месяца активно использовали разработанный чат-бот. Испытуемым были предложены анкеты, водителям необходимо было оценить технические показатели, а сотрудникам управления - функциональные аспекты.

Проверка эффективности применения чат-бота по парковкам дала результаты, представленные в таблицах 1 и 2.

Таблица 1

Количественные результаты опроса сотрудников транспортного управления по каждому из функциональных критериев (количество положительных оценок в %)

| Функциональные показатели | Эффективность информирования | Уменьшение обращений к операторам | Повышение удовлетворенности услугами | Количество пользователей |
|---------------------------|------------------------------|-----------------------------------|--------------------------------------|--------------------------|
| Сотрудники управления     | 85                           | 75                                | 80                                   | 120                      |

Таблица 2

Количественные результаты опроса водителей по каждому из технических критериев (количество положительных оценок в %)

| Технические показатели | Удобство интерфейса | Быстрота получения информации | Точность предоставляемых данных |
|------------------------|---------------------|-------------------------------|---------------------------------|
| Водители               | 70                  | 65                            | 75                              |

На основании полученных результатов и комментариев к ответам испытуемых сделаны следующие заключения:

1. Высокий уровень функциональных показателей, на наш взгляд, обусловлен:

1.1. Непрерывной доступностью информационной поддержки через чат-бота, позволяющей водителям узнавать о парковках в любое время.

1.2. Удобством получения информации через бота, что особенно ценно в условиях городского трафика, когда водителям неудобно искать информацию вручную.

2. Средний уровень технических показателей используемого бота по парковкам указывает на необходимость дальнейшей доработки его функционала. Некоторые водители отметили, что им приходилось искать

дополнительную информацию вне бота. Также было высказано предложение сделать интерфейс более интуитивно понятным и информативным.

Цель исследования - оценить эффективность использования чат-бота как инструмента информирования о парковках - была достигнута. Общая положительная оценка, данная как сотрудниками транспортного управления, так и водителями, демонстрирует пригодность данного способа коммуникации для улучшения городской инфраструктуры. При этом основная задача - обеспечение водителей актуальной информацией о парковках и уменьшение нагрузки на операторов - была выполнена успешно, что свидетельствует об эффективности применения чат-бота.

**Вывод по главе:** разработанный чат-бот для информирования водителей о платных парковках показал свою эффективность и востребованность. Использование чат-бота позволило повысить качество информирования водителей за счет круглосуточной доступности и удобства получения данных. Это подтверждается высокими показателями удовлетворенности среди опрошенных сотрудников транспортного управления и водителей.

Вместе с тем, есть потенциал для дальнейшего улучшения технических характеристик чат-бота, в частности, интуитивности интерфейса и полноты предоставляемой информации. Результаты исследования указывают на необходимость итеративной доработки чат-бота с учетом обратной связи от пользователей.

В целом, проведенное исследование подтвердило эффективность использования чат-бота как инструмента для решения задачи информирования водителей о парковках. Полученные результаты свидетельствуют о перспективности применения подобных решений для улучшения городской инфраструктуры.

## ЗАКЛЮЧЕНИЕ

Разработка системы информирования водителей о платных парковках представляет собой актуальную научно-техническую задачу, решение которой может значительно повысить эффективность использования городской транспортной инфраструктуры.

В ходе данной работы был проведен анализ существующих решений в этой области, выявлены их основные преимущества и недостатки. Несмотря на наличие ряда функциональных мобильных приложений и интернет-сервисов, существующие системы имеют существенные ограничения, связанные с актуальностью данных, интеграцией с другими сервисами, удобством использования и персонализацией.

Для решения выявленных проблем была разработана концепция информационной системы на основе чат-бота, интегрированного с городскими данными и сервисами. Предложенная система обеспечивает высокую точность и оперативность предоставляемой информации о парковках, а также максимально удобна и интуитивно понятна для пользователей.

В рамках проекта был реализован прототип чат-бота на платформе Telegram, демонстрирующий основные принципы предлагаемого решения. Чат-бот включает в себя инструменты для анализа геолокационных данных и текстовых запросов пользователей, механизмы интеграции с внешними API, а также интуитивный пользовательский интерфейс.

Проведенное тестирование подтвердило эффективность использования разработанного чат-бота для решения задачи информирования водителей о парковках. Результаты опроса показали высокий уровень удовлетворенности пользователей качеством и оперативностью предоставляемой информации.

Разработанная в ходе проекта система продемонстрировала свою эффективность и потенциал применения для повышения удобства использования городского транспорта. Дальнейшее развитие проекта может

включать расширение охвата городов и функциональности системы, а также более тесную интеграцию с городскими данными и сервисами. Реализация предложенной концепции в масштабах крупного города способна значительно улучшить транспортную ситуацию и сократить временные и экономические потери, связанные с поиском парковок.

Таким образом, цель работы достигнута, задачи решены, гипотеза доказана.



## СПИСОК ИСПОЛЬЗОВАННОЙ ЛИТЕРАТУРЫ

1. Gdeparkon [Электронный ресурс]. URL: <https://gdeparkon.ru/> (дата обращения: 10.01.2024).
2. Onepark [Электронный ресурс]. URL: <https://onepark.ru/> (дата обращения: 10.01.2024).
3. ParkApp [Электронный ресурс]. URL: <https://parkapp.pro/> (дата обращения: 10.01.2024).
4. ParkAt [Электронный ресурс]. URL: <https://parkat.ru/> (дата обращения: 10.01.2024).
5. Parkimeter [Электронный ресурс]. URL: <https://parkimeter.ru/> (дата обращения: 10.01.2024).
6. Parkopedia [Электронный ресурс]. URL: <https://www.parkopedia.ru/> (дата обращения: 10.01.2024).
7. Абросимов Л.И., Рогов Д.Р. Методика переноса информационных систем в систему 1С-Битрикс на примере модуля «расписание» // Естественные и технические науки. 2023. № 8 (183). С. 136-140.
8. Аксенов К.А., Спицина И.А. Решение задачи интеграции информационных систем на примере автоматизированной системы выпуска металлургической продукции // Инженерный вестник Дона. 2023. № 6 (102). С. 705-718.
9. Алпеев В.С., Ли М.В., Савельев А.А., Белодедов М.В., Большаков С.А. Применение мультипредметных нейронных сетей и миварных экспертных систем для создания гибридных интеллектуальных информационных систем // Искусственный интеллект в автоматизированных системах управления и обработки данных. Сборник статей Всероссийской научной конференции. В 2-х томах. Москва, 2022. С. 35-42.
10. Алпеев В.С., Ли М.В., Савельев А.А., Правдина А.Д., Балдин А.В. О применении мультипредметных нейронных сетей и миварных экспертных систем для создания гибридных интеллектуальных

информационных систем // Информация и образование. Границы коммуникаций. 2022. № 14 (22). С. 224-226.

11. Алчинов А.И., Гороховский И.Н. Концепция создания прикладной географической информационной системы моделирования поисковых корреляционно-экстремальных систем автономной навигации // Проблемы управления. 2022. № 1. С. 54-66.

12. Андреева Н.А., Корчагина Е.В., Корчагин В.В., Магометов А.К. Архитектура информационной системы автоматизированного обслуживания пользователей сети подразделений уголовно-исполнительной системы // Вестник Воронежского института ФСИН России. 2022. № 4. С. 23-29.

13. Андреева О.Н., Новиков Е.С. Информационная система поддержки принятия решений в экспертных системах // Морской вестник. 2023. № S1 (16). С. 63-67.

14. Антамошин, А.Н. Интеллектуальные системы управления организационно-техническими системами / А.Н. Антамошин, О.В. Близнова, А.В. Бобов, Большак . – М.. РиС, 2016. – 160 с.

15. Апухтина А.В., Ширяева Ю.В. Роль и место информационных систем в системе оценки качества образования // Научно-методическое обеспечение оценки качества образования. 2022. № 2 (16). С. 117-122.

16. Афанасьев А.С., Вященко Ю.Л., Иванов К.М. Метрики информационной системы координат как инструменты управления рисками в проектировании системы оружия // Известия Российской академии ракетных и артиллерийских наук. 2022. № 3 (123). С. 51-58.

17. Барков Е.А., Лукьянов К.В. Проектирование информационной системы для организации и проведения футбольных мероприятий по «Нидерландской системе» // Системный анализ в науке и образовании. 2022. № 3. С. 180-190.

18. Белова И.К., Дерюгина Е.О., Чухраев И.В. Математическое обеспечение информационной системы расчета теплофизических параметров

термокатодов сильноточных плазменных систем //Наукоемкие технологии. 2022. Т. 23. № 6. С. 12-20.

19. Белокуров С.В., Кондратов О.А., Меркулов А.В., Панасюк В.С. Системный анализ интегрированной системы безопасности на уровне информационной системы // Материалы LX отчетной научной конференции преподавателей и научных сотрудников ВГУИТ за 2021 год. В 3 частях. Под ред. О.С. Корнеевой; Воронеж. Гос. Ун-т инж. Технол., 2022. С. 107-110.

20. Большаков Р.В., Смирнов А.Б., Кровш С.Ф. Оценочный инструментарий процессов бюджетирования в системе менеджмента предприятия в условиях современных информационных систем // Экономика и предпринимательство. 2022. № 2 (139). С. 951-956.

21. Бондаренко А.С., Зайцев К.С. Использование систем управления контейнерами для построения распределенных облачных информационных систем с микросервисной архитектурой // Международный журнал гуманитарных и естественных наук. 2022. № 1-1 (64). С. 62-65.

22. Ботуз, С. Интеллектуальные интерактивные системы и технологии управления удаленным доступом. Учебное пособие / С. Ботуз. – М.. Солон-пресс, 2014. – 340 с.

23. Бочарова Е.Ю. Оценка внедрения и использования ERP – систем в информационных системах бухгалтерского учета // Научный Лидер. 2023. № 10 (108). С. 11-13.

24. Буреш, О.В. Интеллектуальные информационные системы управления социально-экономическими объектами / О.В. Буреш, М.А. Жук. – М.. Красанд, 2012. – 192 с.

25. Бутов А.А., Максимов И.А. Модель сопровождения информационной системы в терминах систем массового обслуживания // Ученые записки УлГУ. Серия. Математика и информационные технологии. 2023. № 1. С. 6-12.

26. Валуев М.К. Интеграции системы защиты информационной системы с системой контроля и управления доступа на предприятии //

Кибербезопасность. Технические и правовые аспекты защиты информации. Материалы межвузовской студенческой научно-практической конференции. Волгоград, 2022. С. 30-34.

27. Валуев М.К., Журавлев С.И., Лазунин К.А., Грачёв А.С. Интеграции системы защиты информационной системы с системой контроля и управления доступа на предприятии // Уральский научный вестник. 2022. Т. 3. № 6. С. 66-72.

28. Ванина М.Ф., Ерохин А.Г., Фролова Е.А. Системы поддержки принятия решения для бухгалтерских информационных систем // Технологии информационного общества. Сборник трудов XVI Международной отраслевой научно-технической конференции. 2022. С. 215-219.

29. Васильев, В.И. Интеллектуальные системы защиты информации. Учебное пособие / В.И. Васильев; Рецензент Р.М. Асадуллин [и др.]. – М.. Машиностроение, 2012. – 171 с.

30. Гагарин Я.А. Автоматизация системы управления доступом к элементам информационной системы размещенным в облачном сервисе // Актуальные исследования. 2022. № 16 (95). С. 19-22.

31. Гопоненко А.А. Информационная система поддержки устойчивого функционирования корпоративной информационной системы // Студенческий вестник. 2023. № 23-5 (262). С. 45-48.

32. Григорьев Д.С. Разработка информационной системы автоматизированной обработки данных дистанционной системы мониторинга // Вестник науки. 2023. Т. 3. № 5 (62). С. 594-596.

33. Евгеньев, Г.Б. Интеллектуальные системы проектирования. Учебное пособие / Г.Б. Евгеньев. – М.. МГТУ им. Баумана, 2012. – 410 с.

34. Евменов, В.П. Интеллектуальные системы управления. Превосходство искусственного интеллекта над естественным интеллектом? / В.П. Евменов. – М.. КД Либроком, 2017. – 304 с.

35. Емельянов, С.В. Искусственный интеллект и принятие решений. Методы рассуждений и представления знаний. Когнитивные исследования.

Интеллектуальные системы. Вып.3 / С.В. Емельянов. – М.. Ленанд, 2014. – 120 с.

36. Интеллектуальные сенсорные системы / Под ред. Дж. К. М. Мейджера. – М.. Техносфера, 2012. – 464 с.

37. Информационные технологии и вычислительные системы. Высокопроизводительные вычислительные системы. Глобальные проекты и решения. Интеллектуальные системы и технологии. Прикладные аспекты информатики / Под ред. С.В. Емельянова. – М.. Ленанд, 2013. – 128 с.

38. Искусственный интеллект и принятие решений. Когнитивные исследования. Эволюционные вычисления. Интеллектуальные системы и технологии. Многокритериальный анализ решений / Под ред. С.В. Емельянова. – М.. Ленанд, 2012. – 128 с.

39. Марков, Н.Г. Интеллектуальные навигационно-телекоммуникационные системы управления подвижными объектами с применением технологии облачных вычислений / Н.Г. Марков, Д.М. Сонькин, А.С. Фадеев и др. – М.. РиС, 2014. – 158 с.

40. Остроух, А.В. Интеллектуальные информационные системы и технологии. Монография / А.В. Остроух, А.Б. Николаев. – СПб.. Лань, 2019. – 308 с.

41. Приложение «Парковки Москвы» [Электронный ресурс]. URL: <https://parking.mos.ru/> (дата обращения: 10.01.2024).

42. Проект «Умная парковка» от компании Аксиома Групп [Электронный ресурс]. URL: <https://axioma.group/ru/projects/umnaya-parkovka> (дата обращения: 10.01.2024).

43. Сервис интеллектуального мониторинга парковок от компании Data Science [Электронный ресурс]. URL: <https://data-science.ru/solutions/intellectual-parking/> (дата обращения: 10.01.2024).

44. Система «Микком AS101 ProPark» [Электронный ресурс]. URL: <https://as101.ru/products/propark/> (дата обращения: 10.01.2024).

45. Советов, Б.Я. Интеллектуальные системы и технологии. Учебник / Б.Я. Советов. – М.. Академия, 2017. – 192 с.
46. Советов, Б.Я. Интеллектуальные системы и технологии. Учебник для студентов учреждений высшего профессионального образования / Б.Я. Советов, В.В. Цехановский, В.Д. Чертовской.. – М.. ИЦ Академия, 2013. – 320 с.
47. Тейлор, Д. Почти интеллектуальные системы. Как получить конкурентные преимущества путем автоматизации принятия скрытых решений / Д. Тейлор, Н. Рэйдэн. – СПб.. Символ-плюс, 2009. – 448 с.
48. Труды ИСА РАН. Параллельные вычисления. Математическое моделирование. Интеллектуальные системы и технологии. Методы и модели в экономике. Информатика сообществ. Методологические проблемы системного анализа / Под ред. С.В. Емельянова. – М.. Красанд, 2013. – 144 с.
49. Умная парковка [Электронный ресурс] АБС авто режим доступа: <http://www.abs-magazine.ru/article/umnaja-parkovka> Дата обращения: 10.01.2024.
50. Финн, В.К. Интеллектуальные системы и общество / В.К. Финн. – М.. КомКнига, 2007. – 352 с.
51. Цифровые парковки [Электронный ресурс]. URL: <https://digitalparking.ru/> (дата обращения: 10.01.2024).
52. Яндекс.Парковки [Электронный ресурс]. URL: <https://park.yandex/> (дата обращения: 10.01.2024).

```
import telebot
from dadata import Dadata

bot = telebot.TeleBot('5982987139:AAF6IJ3Ann7X-
ANjG9d9e4OfRc4HW6jHbFE')

token = "8b1ae26c7efee102621070e4cd1241069ad4612a"
secret = "7b970f91044a73b8d1a40a5184924a8ceaaf6d04"

@bot.message_handler(content_types=['text'])
def check_adr(message):
    dadata = Dadata(token, secret)
    result = dadata.clean("address", message.text)

    print('Результат - ', result['result'])
    print('Город - ', result['region'])
    print('Улица - ', result['street'])
    print('Дом - ', result['house'])
    print('Буква - ', result['block'])
    print('#####')

# Функция запуска сервера
def main():
    bot.polling(none_stop=True, interval=0, timeout=25)
```

```
# Инициализация скрипта  
if __name__ == '__main__':  
    main()
```



Приложение Б  
(обязательное)

```
# Импорт импользуемых библиотек
from telebot import types
import telebot
import json

# Импорт модулей
from utils.check_location import check_loc
from utils.check_adress import check_adr

# Токен TELEGRAM ~ BotFather
bot = telebot.TeleBot('5982987139:AAF6IJ3Ann7X-
ANjG9d9e4OfRc4HW6jHbFE')

# Регистрация пользователя
@bot.message_handler(commands=['start', 'help'])
def handle_start(message):
    with open('./data/users.json', encoding='utf-8') as jsonfile:
        data = json.load(jsonfile)
        jsonfile.close()

    str_id = str(message.from_user.id)

    if not str_id in data.keys():
        data[str_id] = {'first_name': message.from_user.first_name,
                        'select_city': '', 'action': 'choice_city'}
        bot.reply_to(message, 'Welcome to App Starter!')
```

```

print(f'~~ Новый пользователь - {message.from_user.first_name}')

with open("./data/users.json", "w", encoding='utf-8') as jsonfile:
    json.dump(data, jsonfile, ensure_ascii=False)
    jsonfile.close()

return update(message)

# Обновление состояния пользователя action = [choice_city,
choice_actions, check_adress, check_geolocation, read_adress, read_geolocation]
def update(message):
    with open('./data/users.json', encoding='utf-8') as jsonfile:
        data = json.load(jsonfile)
        jsonfile.close()

    str_id = str(message.from_user.id)

    if str_id in data.keys():
        # choice_city
        if data[str_id]['action'] == 'choice_city':
            markup = types.ReplyKeyboardMarkup(
                row_width=1, input_field_placeholder='Выберите город:',
resize_keyboard=True)
            item_btn1 = types.KeyboardButton('Санкт-Петербург')
            item_btn2 = types.KeyboardButton('Москва')
            markup.add(item_btn1, item_btn2)
            bot.send_message(message.chat.id, "Выберите город:",
                reply_markup=markup)

```

```

# choice_actions
if data[str_id]['action'] == 'choice_actions' and data[str_id]['select_city']
!= "":

    markup = types.ReplyKeyboardMarkup(
        row_width=1, input_field_placeholder='Выберите действие:',
resize_keyboard=True)

    item_btn1 = types.KeyboardButton('Проверить адрес')
    item_btn2 = types.KeyboardButton('Проверить по геолокации')
    item_btn3 = types.KeyboardButton('Выбрать город')
    markup.add(item_btn1, item_btn2, item_btn3)
    bot.send_message(message.chat.id, "Выберите действие:",
        reply_markup=markup)

# check_adress
if data[str_id]['action'] == 'check_adress' and data[str_id]['select_city']
!= "":

    markup = types.ReplyKeyboardMarkup(
        row_width=1, input_field_placeholder='Напишите адрес',
resize_keyboard=True)

    item_btn1 = types.KeyboardButton('Назад')
    markup.add(item_btn1)
    bot.send_message(message.chat.id, "Напишите адрес",
        reply_markup=markup)

# check_geolocation
if data[str_id]['action'] == 'check_geolocation' and
data[str_id]['select_city'] != "":

    markup = types.ReplyKeyboardMarkup(
        row_width=1, input_field_placeholder='Отправьте геоточку',
resize_keyboard=True)

```

```

        item_btn1 = types.KeyboardButton('Назад')
        markup.add(item_btn1)
        bot.send_message(message.chat.id, "Отправьте геоточку",
                           reply_markup=markup)

# read_adress
if data[str_id]['action'] == 'read_adress' and data[str_id]['select_city'] !=
":
    markup = types.ReplyKeyboardMarkup(
        row_width=1,
input_field_placeholder=f'{data[str_id]["select_city"]}' - Введите адрес',
resize_keyboard=True)
    item_btn1 = types.KeyboardButton('Назад')
    markup.add(item_btn1)
    bot.send_message(message.chat.id, "Введите адрес",
                     reply_markup=markup)

# read_geolocation
if data[str_id]['action'] == 'read_geolocation' and
data[str_id]['select_city'] != ":
    markup = types.ReplyKeyboardMarkup(
        row_width=1,
input_field_placeholder=f'{data[str_id]["select_city"]}' - Отправьте геоточку',
resize_keyboard=True)
    item_btn1 = types.KeyboardButton('Назад')
    markup.add(item_btn1)
    bot.send_message(message.chat.id, "Отправьте геоточку",
                     reply_markup=markup)
else:
    return handle_start(message)

```

```

# Обработчик кнопок + обработчик действий
@bot.message_handler(content_types=['text', 'location'])
def listen_chat(message):
    with open('./data/users.json', encoding='utf-8') as jsonfile:
        data = json.load(jsonfile)
        jsonfile.close()

    str_id = str(message.from_user.id)
    if str_id in data.keys():

        # choice_actions
        if message.text in ['Санкт-Петербург', 'Москва'] and
data[str_id]['select_city'] == ":
            with open('./data/users.json', "r", encoding='utf-8') as jsonfile:
                data = json.load(jsonfile)
                jsonfile.close()

            str_id = str(message.from_user.id)

            data[str_id]['action'] = 'choice_actions'
            data[str_id]['select_city'] = message.text

            with open("./data/users.json", "w", encoding='utf-8') as jsonfile:
                json.dump(data, jsonfile, ensure_ascii=False)
                jsonfile.close()

            return update(message)

# Кнопка Назад

```

```

if message.text == 'Назад':
    with open('./data/users.json', "r", encoding='utf-8') as jsonfile:
        data = json.load(jsonfile)
        jsonfile.close()

    str_id = str(message.from_user.id)
    data[str_id]['action'] = 'choice_actions'

    with open("./data/users.json", "w", encoding='utf-8') as jsonfile:
        json.dump(data, jsonfile, ensure_ascii=False)
        jsonfile.close()

    return update(message)

# choice_city
if message.text == 'Выбрать город':
    with open('./data/users.json', "r", encoding='utf-8') as jsonfile:
        data = json.load(jsonfile)
        jsonfile.close()

    str_id = str(message.from_user.id)
    data[str_id]['action'] = 'choice_city'
    data[str_id]['select_city'] = ""

    with open("./data/users.json", "w", encoding='utf-8') as jsonfile:
        json.dump(data, jsonfile, ensure_ascii=False)
        jsonfile.close()

    return update(message)

```

```

# read_adress
if message.text == 'Проверить адрес':
    with open('./data/users.json', "r", encoding='utf-8') as jsonfile:
        data = json.load(jsonfile)
        jsonfile.close()

    str_id = str(message.from_user.id)
    data[str_id]['action'] = 'read_adress'

    with open("./data/users.json", "w", encoding='utf-8') as jsonfile:
        json.dump(data, jsonfile, ensure_ascii=False)
        jsonfile.close()

    return update(message)

# read_geolocation
if message.text == 'Проверить по геолокации':
    with open('./data/users.json', "r", encoding='utf-8') as jsonfile:
        data = json.load(jsonfile)
        jsonfile.close()

    str_id = str(message.from_user.id)
    data[str_id]['action'] = 'read_geolocation'

    with open("./data/users.json", "w", encoding='utf-8') as jsonfile:
        json.dump(data, jsonfile, ensure_ascii=False)
        jsonfile.close()

    return update(message)

```

```

        # Проверка по Адресу
        if message.content_type == 'text' and data[str_id]['action'] ==
'read_adress' and message.text != 'Проверить адрес':
            check_adr(message)

        # Проверка по Геолокации
        if message.content_type == 'location' and data[str_id]['action'] ==
'read_geolocation':
            check_loc(message)
    else:
        return handle_start(message)

# Функция запуска сервера
def main():
    try:
        print('~~ Сервер запускается!')
        bot.polling(none_stop=True, interval=0, timeout=25)
    except Exception as e:
        print(f'~~ Ошибка -- {e}')
        main()

# Инициализация скрипта
if __name__ == '__main__':
    main()

```