МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РЕСПУБЛИКИ КАЗАХСТАН
Некоммерческое акционерное общество
«АЛМАТИНСКИЙ УНИВЕРСИТЕТ ЭНЕРГЕТИКИ И СВЯЗИ ИМЕНИ ГУМАРБЕКА ДАУКЕЕВА»
Кафедра IT-инжиниринг

«ДОПУЩЕН К ЗАЩИТЕ»
Зав. Кафедрой PhD, доцент Досжанова Алия Амантаевна
(ученая степень, звание, Ф.И.О.)
_____ «____» _____202___г.
(подпись)

ДИПЛОМНЫЙ ПРОЕКТ

На тему: Разработка Telegram бота для поиска авиабилетов

Специальность 5В060200 Информатика
Выполнила Бухметова Алина Амангельдиевна Группа Инф-16-2
(Ф.И.О.)

Научный руководитель к.т.н., доцент Балгабаева Ляззат Шайхановна
(ученая степень, звание, Ф.И.О.)
Консультанты:
по экономической части: к.э.н, профессор Габелашвили Кахабер Ревазович
(ученая степень, звание, Ф.И.О.)
_____ «____» _____202___г.
(подпись)
по безопасности жизнедеятельности: ассистент Тыщенко Елена Михайловна
(ученая степень, звание, Ф.И.О.)
_____ «____» _____202___г.
(подпись)
по применению вычислительной техники:
ст. преп. Майкотов Мухит Нурдаулетович
(ученая степень, звание, Ф.И.О.)
_____ «____» _____202___г.
(подпись)
Нормоконтролер: ст. преп. Абсатарова Бибигуль Рыскуловна
(ученая степень, звание, Ф.И.О.)
_____ «____» _____202___г.
(подпись)
Рецензент: _____
(ученая степень, звание, Ф.И.О.)
_____ «____» _____202___г.
(подпись)

Алматы 2020

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РЕСПУБЛИКИ КАЗАХСТАН
Некоммерческое акционерное общество
«АЛМАТИНСКИЙ УНИВЕРСИТЕТ ЭНЕРГЕТИКИ И СВЯЗИ ИМЕНИ ГУМАРБЕКА ДАУКЕЕВА»

Институт Систем Управления и Информационных Технологий

Кафедра IT-инжиниринг

Специальность 5В060200 – Информатика

ЗАДАНИЕ
на выполнение дипломного проекта

Студенту Бухметовой Алине Амангельдиевне
(фамилия, имя, отчество)
Тема проекта: Разработка Telegram бота для поиска авиабилетов

Утверждена приказом по университету №____ от «__» _____ 202___ г.

Срок сдачи законченного проекта «_____»_____202___г.

Исходные данные к проекту (требуемые параметры результатов исследования (проектирования) и исходные данные объекта):
Разработка Telegram бота, обеспечивающего поиск авиабилетов для клиентов компании ТОО “Аэлита-2014”.

Перечень вопросов, подлежащих разработке в дипломном проекте, или краткое содержание дипломного проекта:
Обзор служб обмена мгновенными сообщениями, сравнение и анализ функций служб обмена мгновенными сообщениями, Telegram, проектирование и создание бота, расчёт трудоёмкости и себестоимости, определение экономической эффективности, расчёт интегральной бальной оценки тяжести труда, оптимизация условий труда.

Перечень графического материала (с точным указанием обязательных чертежей): UML диаграмма, диаграмма потоков данных (flow diagram).

Основная рекомендуемая литература: Telegram Bot API; JavaScript. Сильные стороны. Д. Крокфорд, 2013. - 486 с.; Node.js Разработка серверных веб-приложений на JavaScript. Д. Хэррон, 2014. - 144 с.

Консультации по проекту с указанием относящихся к ним разделов проекта

Раздел	Консультант	Сроки	Подпись
Экономическая часть	Габелашвили К. Р.	24.04.2020	
Безопасность жизнедеятельности	Тыщенко Е. М.	24.04.2020	
Программная часть	Майкотов М. Н.	13.05.2020	
Нормоконтроль	Абсатарова Б. Р.	13.05.2020	

График
подготовки дипломного проекта

Наименование разделов, перечень разрабатываемых вопросов	Сроки представления научному руководителю	Примечания
Аналитическая часть	10.02.2020	
Проектная часть	24.02.2020	
Экспериментальная часть	16.03.2020	
Экономическая часть	24.04.2020	
Безопасность жизнедеятельности	24.04.2020	

Дата выдачи задания «___» _____ 2020 г.

Заведующий кафедрой _____ Досжанова А. А.
(подпись) (Ф.И.О.)

Руководитель работы
(проекта) _____ Балгабаева Л. Ш.
(подпись) (Ф.И.О.)

Задание принял к исполнению
студент _____ Бухметова А. А.
(подпись)
(Ф.И.О.)

АҢДАТПА

Бұл дипломдық жобада Телеграм бот арқылы авиабилеттерді іздеуіне арналған жоба. Ойлап шығарылған бот пайдаланушыға қажетті авиабилеттерді табуға мүмкіндік береді. Пайдаланушы керекті қаланы, жөнелтілген күнін және қажет болған жағдайда кері қайту билеттің күнін таңдай алады. Таңдалған сүзгілер бойынша, бот барлық рейстер бойынша ақпарат беріп және ыңғайлы интерфейстің арқасында, пайдаланушы өзі үшін ыңғайлы рейсті таңдап, сілтеме бойынша өтіп, билетті сатып ала алады. Бот Node.js бағдарламалық платформасында JavaScript тілінде әзірленген және Telegram мессенджері пайдаланушыларына бағытталған.

Дипломдық жобада еңбек ауырлығын интегралды балдық бағалау, еңбек шарттарын оңтайландыру есебі жүргізілді және дипломдық жобаның техникалық-экономикалық негіздемесі сипатталды.

АННОТАЦИЯ

Данный дипломный проект посвящён разработке Telegram бота для поиска авиабилетов. Разрабатываемый бот несёт в себе информационный характер и позволяет, не выходя из популярного мессенджера, найти авиабилеты, необходимые пользователю. Пользователь может выбрать город отправления, дату отправления и при необходимости также дату обратного билета. Бот предоставит информацию по всем рейсам выбранной даты и благодаря удобному интерфейсу, пользователь сможет получить информацию, выбрать для себя оптимальный рейс и купить билет, перейдя по ссылке. Бот был разработан на языке JavaScript в программной среде Node.js и ориентирован на пользователей мессенджера Telegram.

В дипломном проекте был произведён расчёт интегральной бальной оценки тяжести труда, оптимизации условий труда и описано технико-экономическое обоснование дипломного проекта.

ANNOTATION

This thesis is devoted to the development of a Telegram bot for searching for air tickets. The bot being developed is informational in nature and allows you to find the tickets that the user needs without leaving the popular messenger. The user can choose the departure city, departure date and, if necessary, the date of the return ticket. The bot will provide information on all flights of the selected date and thanks to a convenient interface, the user will be able to get information, choose the optimal flight and buy a ticket by clicking on the link. The bot was developed in JavaScript in the Node.js cross-platform and is aimed at users of the Telegram messenger.

In the thesis, the calculation of an integral point assessment of the severity of work, optimization of working conditions and described the feasibility study of the diploma project.

Содержание

Введение

В нынешнее время мессенджеры играют исключительно активную роль в жизни каждого пользователя сети интернет. Весь мир обязывает нас пользоваться службами мгновенного обмена сообщениями. Темпы современного мира набирают обороты, тем самым лишая доступа в Интернет любого активного пользователя, мы замечаем его беспомощным и беззащитным.

Интернет кроме источника полезной и разнообразной информации для пользователя, также является главной платформой для виртуального общения посредством социальных сетей и мессенджеров. Поддерживание связи с родными, близкими, деловыми партнёрами, коллегами по работе, новые знакомства – всё это является важным компонентом повседневной жизни современного человека, причем выбор наиболее удобного способа онлайн общения стоит за пользователем и достаточно велик.

Актуальность выпускного дипломного проекта обусловлена высокой популярностью мессенджера Telegram и таких средств автоматизации, как боты, среди пользователей сети Интернет. Боты позволяют упростить ежедневные рутинные задачи, такие как получение информации о погоде, пробках, последних новостях и многое другое. Главным достоинством, относительно классических приложений, является возможность многофункциональности на платформе одного мессенджера.

1 Аналитическая часть

В данной главе производится обзор, анализ и сравнение функций популярных бесплатных служб обмена мгновенными сообщениями.

Каждый день миллионы людей обмениваются сообщениями с помощью служб мгновенного обмена сообщениями или, скорее, мессенджеров (чат-приложений). Однако пользователи не знают, что происходит с сообщениями после их отправки. Первоначально считалось, что шифрование используется только параноидальными пользователями или людьми с повышенной потребностью в секретности. После разоблачений Эдварда Сноудена потребители стали более осведомленными о конфиденциальности в интернете и опасностях цифрового захвата данных и кражи личных данных.

Среди людей во всем мире были подняты значительные проблемы, в то время как законы о хранении данных выполняются. Мы живем в цифровую эпоху, когда наблюдение и регистрация данных происходят почти на всех наших коммуникациях. Компании хотят собрать как можно больше личной информации о потребителях. Правительства некоторых государств взламывают мобильные устройства своих граждан, чтобы получить несанкционированный доступ для наблюдения и других неизвестных причин. Недавно российское правительство несколько раз обращалось в мессенджер Telegram с просьбой выдать им ключи шифрования граждан, зарегистрированных в мессенджере. Несмотря на то, что Telegram не подчинился, он теперь находится под угрозой быть запрещенными в России.

Хотя приложения для обмена сообщениями существуют уже несколько лет, развитие безопасных мобильных приложений растёт, сосредотачиваясь на обеспечении конфиденциальности пользователей и удовлетворении их потребностей. Последние исследования показывают, что пользователи начинают беспокоиться о защите конфиденциальности на своих смартфонах и противостоят приложениям, которые собирают их контакты.

1.1 Обзор приложений для обмена сообщениями

Две из основных причин, по которым мессенджеры стали настолько популярными за короткий период. Во-первых, быстрый рост доступа к мобильным телефонам (дешевизна и доступность) и Интернету. Во-вторых, спад SMS-сообщений, так как мессенджеры позволяют использовать другие методы удаленной связи.

Приложение для чата Facebook называется Messenger. Это приложение используется более чем двумя миллиардами пользователей, зарегистрированных на Facebook. Мессенджер доступен, через Facebook и используется для передачи сообщений, голосовых звонков и видеозвонков. Сквозное шифрование не включено по умолчанию. Для того чтобы использовать сквозное шифрование, необходимо включить параметр секретный чат при обмене сообщениями с контактом.

WhatsApp имеет простую установку и настройку путем автоматической синхронизации контактов на вашем телефоне. Он позволяет создавать текстовые и мультимедийные сообщения со сквозным шифрованием по умолчанию. Он также периодически запрашивает пароль для доступа к приложению.

Telegram был запущен после разоблачений Сноудена для пользователей, которые осознают необходимость безопасной цифровой связи. Он предлагает шифрование клиент-сервер для сообщений чата и секретных чатов, где конфиденциальность не может быть нарушена. Эти чаты самоуничтожаются через определенное время на устройствах с обоих концов (как для индивидуальных, так и для групповых чатов), а также на сервере.

Signal был разработан компанией Open Whisper Systems. Эдвард Сноуден заявил, что этой компании можно доверять: “используйте всё, что угодно, с помощью открытых систем Whisper”. Приложение использует сквозное шифрование военного уровня. Сигнал усиливается с помощью платформы с открытым исходным кодом, которая тщательно контролируется и рассматривается для повышения безопасности. Это предпочтительное приложение для хактивистов и ведущих экспертов по безопасности.

WeChat имеет более чем 1,151 миллиард пользователей и доминирует в китайской сети. Приложение предлагает текстовые сообщения, голосовые и видеозвонки, групповые чаты и богатый мультимедийный опыт. Он также предлагает такие функции, как “Friend Radar”, ”People Nearby” и “Shake”, чтобы найти новых людей в интернете. Это было одно из первых приложений, которое было доступно на Android Wear и Apple Watch. Он не предлагает сквозного шифрования, но обеспечивает защиту от клиента к серверу и от сервера к клиенту. WeChat также соответствует ISO 270001-2013, очень строгому международному стандарту, поэтому хакерам будет очень трудно взломать приложение.

Line это японское приложение с более чем 600 миллионами пользователей по всему миру. Line предлагает дополнительные функции,

такие как групповые чаты и звонки до 200 участников, а также позволяет звонить на мобильные и стационарные номера с помощью купленных кредитов. Он также позволяет следить за определенными каналами, новостными лентами и событиями. Skype был недавно обновлен в попытке сделать его более привлекательным для пользователей. Он по-прежнему известен своими большими аудио и видео возможностями и используется больше корпоративными пользователями. Цифровая связь защищена TLS (Transport Layer Security) и шифрованием Advanced Encryption Standard (AES), однако шифрование отсутствует при вызове стационарных или мобильных номеров.

Viber имеет аналогичные функции с WhatsApp, используя номер мобильного телефона для входа и синхронизации контактов в телефонной книге устройства.

1.2 Сравнение функций безопасности и конфиденциальности

Поскольку потребители постоянно требуют повышения безопасности и конфиденциальности в приложениях для обмена сообщениями, компании по разработке программного обеспечения пытаются решить эти проблемы. Одна из функций заключалась в запуске сквозного шифрования (смотреть Рисунок 1). Сквозное шифрование относится к тому, когда сообщения шифруются во время передачи, и никакая копия не хранится незашифрованной на серверах поставщиков услуг. Никто, кроме людей, ведущих диалог, не может просматривать эти сообщения; нет третьей стороны, даже правительство или разработчики этих приложений не имеют к ним доступ. Связь передается с использованием секретного кода, а не обычного текста [5].

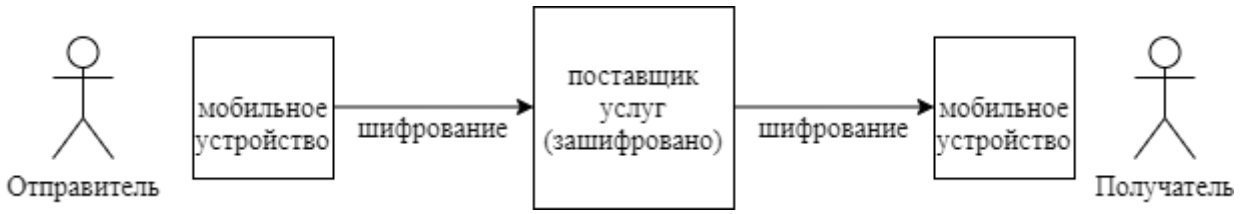

Рисунок 1.1 – Сквозное шифрование

Другим типом шифрования, который используется, является шифрование при передаче (смотреть Рисунок 2). Это означает, что сообщение шифруется между Пользователем и поставщиком услуг, но хранится на сервере в виде открытого текста. Это создает риск, так как сохраненные сообщения могут быть прочитаны поставщиком услуг или другими третьими сторонами, которые получают доступ к серверу.

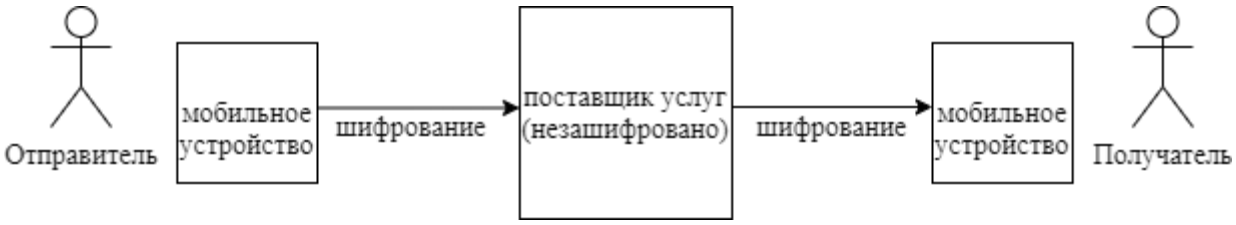

Рисунок 1.2 – Шифрование при передаче

В Таблице 1.1 представлено сравнение приложений обмена сообщениями в отношении функций безопасности и конфиденциальности.

Таблица 1.1 – Безопасность и конфиденциальность мессенджеров

Мессенджер	Hangouts	iMessage	Line	Messenger	Signal	Skype	Snapchat	Telegram	Viber	WeChat	WhatsApp
Сквозное шифрование		✓	✓	✓*	✓	✓*	✓	✓*	✓		✓
Шифрование в пути	✓			✓		✓		✓			
Закрытый ключ недоступен провайдеру		✓						✓	✓		✓
Удалено с сервера		✓							✓		
Сообщения самоуничтожения		✓		✓	✓			✓	✓		
Открытый исходный код					✓			✓			
Блокировка чатов паролем					✓			✓	✓		✓
СМС подтверждение/ электронная почта			✓								✓
Обнаружение скриншота											
Двухэтапная проверка								✓			
Удаленный выход								✓			
Удаленно уничтожать сообщения											✓
Самоуничтожение аккаунта								✓			
Бесплатно	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
П р и м е ч а н и е * - опционный выбор											

Можно утверждать, что адекватный уровень сквозного шифрования должен быть золотым стандартом и функцией безопасности по умолчанию, включаемой в мессенджеры. Большинство приложений, перечисленных в Таблице 1.1, обеспечивают сквозное шифрование. Signal, WhatsApp и

Facebook Messenger используют протокол шифрования сигнала от конца до конца. Хотя Telegram, Skype и Messenger предлагают сквозное шифрование, оно не включено по умолчанию. Telegram предлагает эту функцию в качестве опции "секретный чат". Если это не включено, используется шифрование при передаче. Skype недавно добавил сквозное шифрование, но оно не включено по умолчанию. Нужно начать «частный разговор», чтобы включить сквозное шифрование.

Секретные разговоры в Messenger в настоящее время доступны только в мобильном приложении, таким образом, они не будут отображаться в чате Facebook или messenger.com и, кроме того, он виден только на устройстве, на котором вы создаете разговор, и на устройстве, которое получатель использует для открытия разговора. Поэтому Messenger, Skype и Telegram имеют проверки как для сквозного, так и для транзитного шифрования, как показано в Таблице 1.1. (Сквозное шифрование по умолчанию шифрует также при передаче).

Hangouts не обеспечивает сквозное шифрование, но вместо этого использует шифрование в пути. Это сразу же делает его менее надёжным. Google Hangouts - это приложение, которого следует избегать. Приложение якобы имеет многочисленные проблемы безопасности и конфиденциальности. Оно использует шифрование при транзите, сообщения хранятся на сервере в виде открытого текста. Google может получить доступ к личным сообщениям любого пользователя в любое время и передавать информацию государственным учреждениям и другим третьим лицам. Согласно докладу Amnesty International за 2016/17 год, WeChat имеет серьезные проблемы с конфиденциальностью. Они заняли последнее место со счетом 0 из 100 баллов, когда речь заходит о конфиденциальности. Facebook Messenger и WhatsApp набрали 73 балла, а Apple iMessage 67 из 100 баллов [5]. WeChat не обеспечивает сквозного шифрования и не публикует отчеты о прозрачности по запросу правительства Китая на информацию. Исходя из этого, WeChat подвергался как цензуре, так и надзору. Из-за отсутствия конфиденциальности и безопасности в WeChat, самое верное решение это удалить приложение с вашего устройства.

Поскольку большинство приложений используют сквозное шифрование, следующая возникающая проблема заключается в том, чтобы знать, доступен ли закрытый ключ поставщиком услуг. Apple, Telegram и WhatsApp утверждают, что они не могут получить закрытый ключ. В Таблице 1 не было найдено никакой информации об остальных приложениях. Хотя Apple утверждает, что они не могут получить доступ к закрытому ключу и не могут читать сообщения пользователя, исследование, проведенное хакерами, показало обратное. Они доказали, что технически Apple может читать ваши сообщения iMessage, когда они хотят. Одна из проблем заключается в том, что они не уведомляют пользователей, когда создается новый ключ, в то время как приложения, такие как iMessage и WhatsApp, предупреждают пользователей об этом.

iMessage и Viber - это единственные мессенджеры, которые утверждают, что они удаляют ваши сообщения с сервера. iMessage удаляет сообщения автоматически через семь дней с сервера. Telegram, iMessage, Viber, Messenger, и Signal имеют функцию, которая позволяет сообщениям самоуничтожаться или исчезать через определенное время как для отправителя, так и для устройств получателей. Facebook выпускает таймер самоуничтожения для сообщений, который позволяет пользователям устанавливать таймер, который будет автоматически удалять сообщения.

Как Signal, так и Telegram имеют политику с открытым исходным кодом. Любой желающий может проверить исходный код, протокол и API. Signal, Telegram, Viber и WhatsApp имеют блокировку кода доступа к мессенджеру, которое нужно ввести, прежде чем приложение может быть использовано. При регистрации нового пользователя, как WhatsApp, так и Line отправляют проверочный код через SMS, который требуется до завершения установки. Telegram предлагает функцию двухэтапной проверки, где приложение требует, чтобы использовать как SMS-код и код-пароль для входа в приложение. Приложение также позволяет настроить адрес электронной почты для восстановления на случай, если пользователь забудет пароль. Удаленный выход из системы - это функция, предлагаемая только Telegram. Большинство приложений позволяют войти в мессенджер с нескольких устройств. С помощью этой функции можно выйти из всех клиентов из устройства, которое используется на данный момент. Еще одна функция - самоуничтожение аккаунта. Только Telegram предлагает эту функциональность. Если учетная запись была неактивна в течение определенного периода, где шесть месяцев является значением по умолчанию, учетная запись автоматически самоуничтожится, и все сообщения и носители, связанные с учетной записью, будут удалены. Результаты в этом параграфе показывают что WhatsApp и Viber - это самые безопасные бесплатные приложения. Наименее защищенными приложениями являются WeChat и Google Hangouts, в первую очередь из-за того, что они не используют сквозное шифрование. В следующем параграфе сравниваются приложения с точки зрения доступности хранимых данных.

1.3 Сравнение доступности хранимых данных

Следующий вопрос при сравнении приложений обмена сообщениями с точки зрения безопасности и конфиденциальности заключается в том, где хранятся данные и насколько легко их можно восстановить – это будет включать историю чатов, сообщения, фотографии и видео, отправленные через эти приложения. В этом параграфе сравниваются приложения для обмена сообщениями, основанные на этом факторе. Приложения (бесплатные версии), которые были признаны наиболее безопасными из Таблицы 1.1,

сравниваются в Таблице 1.2. Line и WeChat также были добавлены к этому сравнению из-за их популярности в Японии и Китае.

Таблица 1.2 – Расположение резервных копий чатов в мессенджере

Мессенджер	Резервное копирование устройства: история чата	Резервное копирование устройства: изображения и видео	Облачное резервное копирование	Резервное копирование на ПК	Передача истории чата между мобильными устройствами	Копии отправляются на электронную почту
Line	✓	✓	✓*	✓*	✓	✓
Signal	✓	✓		✓*		
Telegram	✓	✓	✓*	✓*	✓	✓
Viber	✓	✓	✓*	✓*	✓	✓
WeChat	✓	✓	✓*	✓**	✓	✓
WhatsApp	✓	✓	✓*	✓**	✓	✓
П р и м е ч а н и я * - опционный выбор ** - опционный выбор, по умолчанию в веб-версии						

Line позволяет пользователям создавать резервные копии истории чата в различных местах: Line Keep или Memo (на устройстве), share to email, Google Drive, OneDrive и на ПК. Приложение имеет дополнительные функции для резервного копирования в облаке и на ПК.

Signal хранит метаданные только на устройстве, которое требуется для работы приложения. Signal позволяет создавать резервную копию на устройстве, но не на ПК, сервере или облаке резервного копирования.

Telegram сохраняет все изображения в папке image на внутренней памяти устройства или SD-карте. Удаленные чаты, изображения и видео также хранятся в памяти мобильного устройства. Telegram хранит все чаты и медиа на своем облачном сервисе. Секретные чаты не хранятся на сервере, но секретные носители сохраняются (шифруются).

Viber позволяет пользователям сохранять копии своих чатов, отправляя их на свою электронную почту через пользовательские настройки. Чаты будут помещены в архив zip как .csv - файлы с именами контактов, с которыми вы общались. Резервная копия чата может быть создана и прочитана как текст, но она не может быть восстановлена в самом приложении; копии отправленных файлов (фотографий, видео и т. д.) Не сохраняются в таких текстовых файлах.

Viber также хранит данные в отдельной папке, расположенной во внутренней системной памяти устройства пользователя. Такие резервные данные могут быть доступны только пользователю с правами Root или с помощью своего рода программного обеспечения Root explorer.

Данные WeChat могут быть скопированы на ПК с помощью клиентского программного обеспечения WeChat, загруженного из интернета. WeChat, а также другие приложения (например, WhatsApp, Line, Kik, Viber и т.д.), данные также могут быть сохранены на ПК с помощью USB-соединения и программного обеспечения без подключения к интернету. Резервное копирование также может быть сделано путем передачи данных на другой смартфон. Функция резервного копирования в облаке больше не доступна в WeChat 6.2.5. История чата постоянно хранится в приложении на устройстве до тех пор, пока приложение не будет удалено, и телефон имеет достаточное пространство для хранения. В интересах конфиденциальности WeChat не хранит историю чатов на своем сервере, если только пользователь не выбирает функцию резервного копирования. История чата не может быть восстановлена после удаления.

Данные WhatsApp хранятся на устройстве отправителя и получателя. Резервные копии создаются на устройстве пользователя по умолчанию ежедневно в пределах внутренней памяти, и резервные копии также могут храниться на Google диске пользователя. Данные мессенджера хранятся в аккаунте пользователя Facebook. История всех созданных сообщений, будь то на веб-сайте Facebook, Messenger для ПК или для Android, всегда передается через учетную запись Facebook и сохраняется там. Facebook предоставляет пользователям возможность сохранять копию всей информации о пользователе, включая загруженные изображения и видео, контактную информацию, друзей, и, самое главное, полную историю сообщений.

Данные Messenger также хранятся на устройствах Android таким же образом, как и аналогичные приложения во внутренней памяти или папке данных SD-карты Android.

Все приложения в Таблице 1.2 создают резервные копии данных на устройстве. Signal - это единственное приложение, которое не позволяет создавать резервные копии на ПК или облаке, все другие приложения имеют эту дополнительную функцию. Все мессенджеры позволяют передавать сообщения с одного мобильного устройства на другое. Все мессенджеры позволяют отправлять чаты на электронную почту. Это создает риск для того, что если электронная почта пользователя будет скомпрометирована, все резервные копии чата будут доступны злоумышленнику. Основываясь на результатах поиска в этом параграфе, Signal будет наиболее безопасным вариантом из-за того, что приложение не позволяет создавать резервные копии в облаке или на ПК. Все мессенджеры используют эту функцию как необязательную, поэтому, если эта функция не используется, они будут наравне с Signal в этом сравнении.

1.4 Рекомендации по безопасности и конфиденциальности

Данные, генерируемые и передаваемые через приложения для обмена сообщениями, могут быть уязвимы с точки зрения безопасности и конфиденциальности. Выше сравниваются приложения для обмена сообщениями, основанные на встроенных функциях безопасности этих приложений, а также на расположении и последующем доступе к хранимым данным. На основе этих двух аспектов пользователям предлагаются рекомендации по сохранению безопасности и конфиденциальности [5].

Основываясь на выводах в параграфах 1.2 и 1.3, лучшими и наиболее безопасными приложениями для мгновенного обмена сообщениями являются Signal, Telegram, WhatsApp, Viber и Line. WeChat исключается из-за проблем конфиденциальности, указанных в параграфе 1.2.

Пользователи часто имеют необходимость восстанавливать свою историю чата или данные и, следовательно, могут использовать резервные функции различных приложений, чтобы активизировать восстановление данных. Такие резервные копии обычно хранятся на самом устройстве и/или где-то в облаке. Пользователи имеют дополнительные параметры в различных приложениях, чтобы сделать резервные копии по электронной почте или на своем компьютере. Если устройства содержат секретную информацию (личную или иную), необходимо применить необходимые меры безопасности, чтобы обеспечить конфиденциальность данных в случае, если устройство или компьютер будут скомпрометированы.

Выбор того, какое приложение для обмена сообщениями является лучшим, зависит от критериев, которые имеют различную важность для разных пользователей, а также влияет на необходимый уровень безопасности.

К таким критериям относятся:

- использование различными странами разных приложений;
- доступность (достаточно иметь интернет, чтобы пользоваться мессенджером);
- отсутствие доступных минут и дорогая сотовая связь также могут быть важными факторами;
- поставщик услуг сотовой связи не владеет данными – поэтому приложения обмена сообщениями уже более безопасны, чем SMS;
- для обычного человека нет смысла переключаться на какое-то очень безопасное платное приложение, если люди, с которыми ему нужно взаимодействовать, не используют это же приложение.

Поэтому пользователям рекомендуется применять правильные настройки при использовании их предпочтительного приложения обмена сообщениями, в зависимости от того, есть ли у них потребность в восстановлении истории чата и других носителей, созданных в этих приложениях.

Вывод: самый безопасный вариант – не создавать резервные копии чатов и носителей на ПК или облачном сервисе. Если устройство, содержащее

резервные копии, будет потеряно, все резервные копии будут потеряны вместе с ним.

Пользователи выбирают приложения для обмена сообщениями на основе различных критериев и, следовательно, имеют различные требования с точки зрения уровней безопасности (конфиденциальность своих данных). Пользователям рекомендуется применять правильные настройки при использовании их предпочтительного приложения обмена сообщениями, в зависимости от того, есть ли у них потребность в восстановлении истории чата и других носителей, созданных на этих приложениях.

Сравнение с точки зрения доступности для сохраненных данных приложения позволяют предположить, что Signal является наиболее безопасным. В остальных мессенджерах в Таблице 2, при условии применения правильных пользовательских настроек для резервного копирования данных, они будут наравне с Signal в отношении хранения и резервного копирования данных. Наименее защищенными приложениями в Таблице 2 являются WhatsApp и WeChat, в первую очередь из-за того, что при использовании веб-версии, чаты и медиа по умолчанию сохраняются на ПК.

1.5 WhatsApp, Viber и Telegram

В нынешнее время наиболее высокую популярность имеют WhatsApp, Viber и Telegram. Ниже приводится краткий обзор лидирующих служб обмена мгновенными сообщениями.

1.5.1 WhatsApp

В последнее время WhatsApp стал самым первым и популярным мессенджером с более чем, 1 миллиардом пользователей в качестве своей пользовательской базы. В 2014 году компания Facebook приобрела WhatsApp из-за его огромного количества пользователей. В первые годы своего существования WhatsApp взимал с пользователя целых 0,99 доллара США в год (первые 12 месяцев были бесплатным пробным периодом). Позднее абонентская плата была полностью отменена с 18 января 2016 года. WhatsApp предоставляет услуги для обмена текстовыми и звуковыми сообщениями, бесплатными голосовыми вызовами и обменом фотографиями или видео, а также для обмена документами. Но главное преимущество WhatsApp - это его простота для расширения пользовательской базы. При его установке можно использовать все контакты в своей адресной книге на телефоне, которые установили одни и те же приложения. WhatsApp явно доминирует на мировом рынке мгновенных сообщений в настоящее время.

1.5.2 Viber

Viber, возможно, является мессенджером с наибольшим количеством функций в нём. Он имеет множество опций, которые просто превосходят другие; например, он имеет высокое качество видеозвонка. Viber предоставляет услугу голосовой связи, которая может набирать любой мобильный номер, даже если набранный номер не использует Viber (с дополнительной платой, в этом случае).

Viber предлагает групповые чаты так же, как и другие мессенджеры, и в дополнение к этой услуге он также предлагает так называемый публичный чат, который позволяет пользователям общаться открыто, что может подходить для сообщества, общения в клубе поклонников или даже больших объединений. Ряд крупных групп в социальных сетях часто создают свой канал в Viber. Он предлагает множество настраиваемых функций, таких как фон для различных чатов. Вместо того чтобы отправлять только фотографии, он также использует граффити, нарисованных от руки. Viber также предоставляет официальную веб-версию, которую можно использовать на ноутбуке или ПК. Однако, несмотря на свои возможности, он не может привлечь людей. Считается, что его менее интуитивный интерфейс (по сравнению с WhatsApp, например) и его спорное решение для доминирующего фиолетового цвета имеют несколько причин, по которым он отстаёт от аналогов.

Viber, предлагал услуги голосового общения и видеозвонков в течение нескольких лет прежде чем WhatsApp и Telegram начали предоставлять аналогичные голосовые и видео услуги. Viber может предложить лучшее качество голоса с меньшим шумом, несмотря на всю полосу пропускания, по сравнению с WhatsApp и Telegram. Viber также предлагает два режима качества голоса: нормальный и HD-режимы, причем последний очень выгоден для сетей с высокой пропускной способностью. Пожалуй, самой уникальной особенностью является Viber Out, которые позволяют пользователям Viber быть в контакте с людьми, которые не находятся на службе.

1.5.3 Telegram

Telegram - это популярный сервис обмена сообщениями, который основан на платформе с открытым исходным кодом. В дополнение к своему полностью бесплатному сервису без каких-либо платежей, он также предлагает без рекламы среду с чистым и быстрым интерфейсом. Telegram был создан в августе 2013 года российским предпринимателем Павлом Дуровым и его братом Николаем Дуровым. Telegram приобретает статус популярности и становится наиболее загружаемым приложением для обмена сообщениями в магазине Google Play [1].

Telegram прост в регистрации и использовании, он имеет много общего с WhatsApp с точки зрения работы с идентификатором пользователя и

контакта. Номер телефона используется для первичной идентификации пользователя. После того, как пользователь установит приложение, он может связаться с любым номером в списке контактов, у которого тоже установлено это приложение.

Однако Telegram может предложить больше. Любой пользователь может создать имя пользователя в качестве уникального идентификатора в Telegram, что позволяет другим пользователям, связаться напрямую, без необходимости знать номер телефона. Кроме того, добавление имени пользователя в список контактов Telegram, не будет автоматически раскрывать номер телефона. Учитывая, что номер телефона является конфиденциальной информацией, эта функция является отличным способом повышения защиты безопасности. Кроме того, архив чатов хранится в облаке, поэтому пользователям не нужно беспокоиться о том, что они потеряли свои чаты после смены телефона.

Telegram - это приложение, которое может работать на платформах Android, iOS, Windows Phone, Mac и Windows OS. Кроме того, к аккаунту Telegram можно получить доступ с нескольких устройств, даже в заданное время, и сообщения появляются одновременно на всех устройствах.

WhatsApp, Viber и Telegram что лучше всего подходит для обмена мгновенными сообщениями?

1.6 Сравнение функций WhatsApp, Viber и Telegram

Таблица 1.3 – Сравнение функций для WhatsApp, Viber и Telegram

	WhatsApp	Viber	Telegram
Резервные копии	✓(Google Drive integration)	✓(backup file)	✓(cloud-based)
Блокировка пользователя	✓	✓	✓
Трансляции	✓	-	-
Доставка сообщения	✓	✓	✓
Веб-клиент	✓	✓	✓
Форматирование текста	✓	-	✓
Групповой чат	✓ 256 участников	✓ 99 участников	✓ 200000 участников
Голосовые сообщения	✓	✓	✓
Мультиустройство	✓(только одно мобильное	✓	✓(только одно мобильное

	устройство, а также использование веб-клиента)		устройство, а также использование веб-клиента)
Статус онлайн	✓	✓	✓

Продолжение таблицы 1.3

Телефонные звонки	-	✓(комиссии согласно тарифу провайдера)	-
Платформы	iOS, Android, Windows Phone, BlackBerry, Symbian, Nokia Series 40, Nokia Asha	Windows, OS X, Android, BlackBerry, iOS, Series 40, Symbian, Bada, Windows Phone, Linux	Android, iOS, Windows Phone, Windows, OS X, Linux
Фоторедактор	-	-	✓
Цена	Бесплатно	Бесплатно	Бесплатно
Уведомление о прочтении сообщения	✓	✓	✓
Тип регистрации	Телефонный номер	Телефонный номер, email	Телефонный номер
Тип шифрования	Сквозное шифрование	Сквозное шифрование	Сквозное шифрование
Отправка изображений	✓	✓	✓
Отправка видео	✓	✓	✓
Отправка файлов (doc, zip, mp3 и др.)	✓(до 160 MB каждый)	✓(до 200 MB каждый)	✓(до 1.5 GB каждый)
Поделиться контактом	✓	-	✓
Поделиться локацией	✓	✓	✓
Рисование	-	✓	-
Стикеры	✓	✓	✓
Использование имени пользователя	-	-	✓
Видеозвонки	✓	✓	✓

Аудиозвонки	✓	✓	✓

1.7 Сквозное шифрование

Сквозное шифрование (E2EE - End-to-end encryption) - это система связи, в которой только пользователи диалога могут читать сообщения. Данная система не позволяет потенциальным перехватчикам - включая провайдеров телекоммуникационных услуг, интернет-провайдеров и даже провайдеров услуг связи - получить доступ к криптографическим ключам, необходимым для расшифровки разговора [5].

Сквозное шифрование гарантирует, что доступ к исходному тексту сообщения имеется только у отправителя и получателя. Это означает, что пользовательская информация становится недоступной даже серверам, передающим данные.

Во многих системах обмена сообщениями, включая электронную почту, социальные сети и мессенджеры, сообщения проходят через посредников и хранятся третьей стороной, из которой они извлекаются получателем. Даже если сообщения зашифрованы, они, как правило, шифруются только «в пути» и хранятся в расшифрованном виде третьей стороной. Это позволяет третьей стороне обеспечивать поиск и другие функции или сканировать нелегальный и неприемлемый контент, но также означает, что они могут быть прочитаны и использованы ненадлежащим образом, любым, кто имеет доступ к сохраненным сообщениям в сторонней системе, независимо от того, является ли это преднамеренным или нет. Это можно рассматривать как проблему во многих случаях, когда конфиденциальность очень важна.

Системы, такие как Telegram и Google Allo , были подвергнуты критике за то, что они не имеют сквозного шифрования, которое предполагается выключенным по умолчанию. К 2020 году, Telegram по-прежнему не имеет сквозного шифрования по умолчанию, сквозного шифрования для групповых чатов и сквозного шифрования для своих настольных клиентов. Сквозное шифрование в Telegram используется при включении функции “секретного чата”.

Сквозное шифрование предназначено для предотвращения чтения или тайного изменения данных, кроме как истинным отправителем и получателем. Сообщения зашифрованы отправителем, но третья сторона не имеет средств, для их расшифровки и хранит их в зашифрованном виде. Получатели извлекают зашифрованные данные и дешифруют их сами.

Поскольку никакие третьи стороны не могут расшифровать передаваемые или хранимые данные, например, компании, которые используют сквозное шифрование, не могут передавать тексты сообщений своих клиентов властям.

Недавние изменения в безопасности приложений породили жесткую конкуренцию в предоставлении пользователям безопасных сервисов обмена мгновенными сообщениями, как показано в таблице 4, и эта тенденция наблюдается уже много лет.

Таблица 1.4 – WhatsApp, Viber и Telegram в оценке безопасности

Оценка безопасности	WhatsApp	Viber	Telegram	Telegram (Секретные чаты)
Зашифрована ли связь при передаче?	✓	✓	✓	✓

Продолжение таблицы 1.4

Зашифрована ли связь ключом от провайдера?	✓	-	✓	✓
Может ли быть проверена личность?	✓	-	-	✓
Защищены ли коммуникации в случае кражи ключей?	✓	-	-	✓
Код открыт для независимой проверки?	-	-	✓	✓
Хорошо ли документирован криптографический дизайн?	✓	-	✓	✓
Был ли независимый аудит безопасности?	✓	✓	✓	✓

Вывод: Telegram предлагает возможность синхронизации, супер быстрый сервис, надежное резервное копирование и лучшую функцию безопасности. Хотя WhatsApp доминирует в пространстве социальных медиа, из-за своей простоты, и поддерживается гигантом-компанией Facebook, Telegram по существу обеспечивает лучшую и более безопасную платформу, чем другие.

1.8 Постановка задачи

Целью выпускного дипломного проекта является разработка бота для поиска авиабилетов, на основе мессенджера Telegram.

Исходя из цели дипломного проекта, были поставлены следующие задачи:

- сравнить и выполнить анализ функций мессенджеров;
- выявить лучший мессенджер, посредством сравнения и анализа;
- выбор языка и среды разработки;
- разработка бота на платформе Telegram.

2 Проектная часть

2.1 О Telegram

Службы обмена мгновенными сообщениями (IM – Instant Messaging) кардинально изменили способ общения людей друг с другом в последние годы. Хотя это и не новые средства коммуникации, но появление смартфонов и мобильных широкополосных технологий стимулировало эволюцию моделей коммуникации между людьми и организациями [3].

Telegram - это облачная служба обмена мгновенными сообщениями и передачи голоса по IP. Клиентские приложения Telegram доступны для Android, iOS, Windows Phone, Windows, macOS и Linux. Пользователи могут отправлять сообщения и обмениваться фотографиями, видео, наклейками, аудио и файлами любого типа, а также создавать группы на 200 000 человек или каналы для вещания с неограниченной аудиторией. Поскольку в группах Telegram может быть до 200 000 человек, Telegram также обеспечивает ответы, упоминания и хэштеги, которые помогают поддерживать порядок и поддерживать связь в больших сообществах.

Клиентский код Telegram является программным обеспечением с открытым исходным кодом, но исходный код для последних версий не всегда публикуется сразу, в то время как его код на стороне сервера является закрытым исходным кодом. Сервис также предоставляет API для независимых разработчиков.

По состоянию на апрель 2020 года Telegram насчитывает 400 миллионов активных пользователей в месяц, при этом ежедневно регистрируется не менее 1,5 миллиона новых пользователей. Папки, облачное хранилище, поддержка работы с компьютеров — все это делает Telegram незаменимым инструментом для работы и учебы во время карантина. Неудивительно, что сегодня Telegram занимает первое место в AppStore в списке самых загружаемых мессенджеров и социальных сетей в более чем 20 странах. Всё больше и больше людей по всему миру переносят общение в Telegram [18].

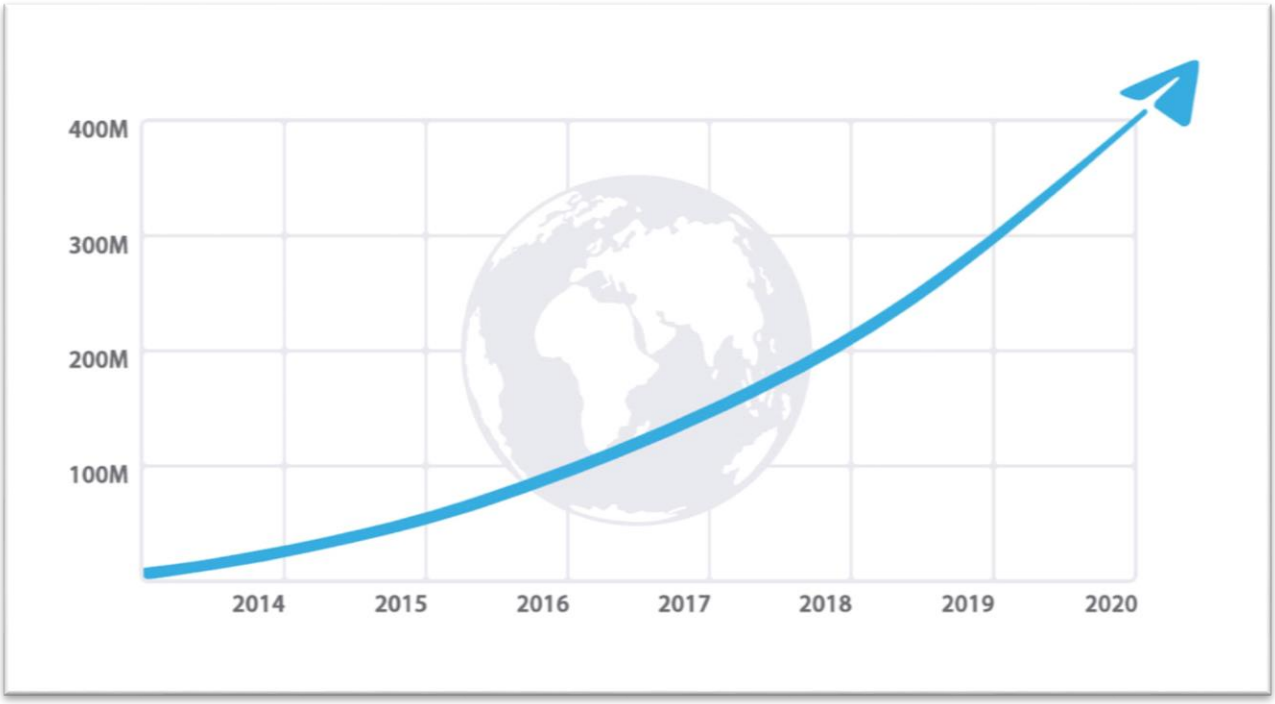

Рисунок 1.3 – Статистика добавления новых пользователей по годам

На данный момент Telegram используется на смартфонах, планшетах и ПК для iOS (9.0 и выше), Android (4.1 и выше) и Windows Phone. А также возможно использование веб-версии Telegram или установка приложения для Windows, macOS и Linux.

Существует три типа коммуникации в Telegram, один – одному (пользователь пользователю), один - многим (каналы и боты в Telegram) и многие - многим (группы и чаты в Telegram).

Кроме того, Telegram предоставляет функции получения услуг и информации с использованием платформы bot (что является предметом исследования дипломной работы). Эта функция уже используется такими официальными сервисами как @music, @youtube, @foursquare и другими верифицированными ботами [18].

Популярные каналы и боты в Telegram стали мощной платформой для мобильной рекламы, позволяющей внедрять в свою аудиторию новые продукты, услуги и т.д. В результате, владельцы каналов и разработчики ботов стараются увеличить свою аудиторию, предоставляя полезные сервисы и услуги. Несмотря на Twitter, Facebook и многие другие социальные сети, основанные на интернете, публичный контент в Telegram не доступен для поиска через поисковые системы, такие как Google и Bing, следовательно, владельцы каналов, которые находятся на ранних стадиях, как правило, рекламируют название и адрес своих каналов в других популярных каналах [6]. В результате, огромная часть сообщений в Telegram посвящена рекламе канала [3]. Таким образом, сообщения в Telegram могут быть

классифицированы на два класса, которые являются рекламой канала (т. е. спам-сообщения) и не рекламой (т. е. сообщения ham).

Telegram - это служба обмена мгновенными сообщениями, в которой пользователи могут отправлять текстовые сообщения, фотографии, видео, стикеры и файлы любого типа. Отправителем или получателем сообщения в Telegram может быть пользователь, группа, бот или канал. Помимо обмена сообщениями между пользователями, каналы и группы могут использоваться для трансляции сообщений в Telegram следующим образом:

Группы: Группа в Telegram состоит из набора пользователей, которые интересуются одной и той же темой. Все члены группы могут отправлять и получать сообщения в группе. Пользователи могут быть приглашены другим участником группы или присоединиться к ней по уникальной ссылке, предоставленной администраторами группы.

Каналы: каналы - это функция трансляции публичных сообщений большому количеству пользователей. В канале обычно есть один или несколько администраторов, которые являются единственными, кто может публиковать сообщения в канале. Неограниченное количество пользователей может подписаться на каждый канал по имени пользователя канала или его ссылке на присоединение.

Несмотря на Facebook, Twitter и многие другие социальные сети, в Telegram нет дружеских отношений между пользователями. Однако у пользователя в Telegram есть личный список контактов для управления сообщениями с близкими друзьями. Пользователи, группы или каналы с несколькими типами отношений между ними следующим образом:

Пересылка: пользователь или канал может переслать исходное сообщение (опубликованное другим пользователем или каналом) другому пользователю, группе или каналу. Пересылка сообщений в Telegram аналогична пересылке электронной почты и действию ретвита в Twitter. Содержание пересылаемого сообщения и исходного сообщения являются одинаковыми, но переадресованное сообщение содержит имя и ссылку профиля исходного поставщика содержимого.

Упоминание: пользователи и каналы в Telegram могут иметь уникальное имя пользователя, которое может быть использовано в качестве ссылки на человека канала или пользователя в сообщении. Каждое сообщение может содержать ноль, одно или несколько упоминаний имен пользователей, которые начинаются с символа@.

2.2 Боты в Telegram

Боты - специальные аккаунты в Telegram, созданные для того, чтобы автоматически обрабатывать и отправлять сообщения. Пользователи могут взаимодействовать с ботами при помощи сообщений, отправляемых через обычные или групповые чаты. Логика бота контролируется при помощи

HTTPS запросов к API для ботов. Для настройки бота не требуется дополнительный номер телефона [7].

Взаимодействие пользователей с ботами происходит двумя способами:

- отправление сообщений и команд ботам, открывая с ними чат или добавляя их в группы, к примеру, новостной бот в групповом чате;

- отправление запросов и команд непосредственно из поля ввода, набрав @username бота и сам запрос. Эта функция позволяет отправлять контент из ботов прямо в любой чат, группу или канал при условии, что бот является встроенным.

Сообщения, команды и запросы, отправленные пользователями, передаются программному обеспечению, работающему на серверах. Сервер - посредник обрабатывает всё шифрование и связь с Telegram API для бота. Общение с этим сервером происходит через простой HTTPS-интерфейс, который предлагает упрощенную версию Telegram API. Данный интерфейс называется Bot API [14].

2.2.1 Создание бота

BotFather - единственный, который управляет всеми ботами Telegram. Он поможет вам создать новых ботов и изменить настройки существующих. Для того чтобы создать бот, необходимо найти @BotFather и начать с ним беседу. Ввести команду /newbot и @BotFather запросит имя бота, а затем сгенерирует токен авторизации для вашего нового бота [14].

Имя пользователя это короткое имя, которое будет использоваться при поиске бота, а также в ссылке. Имена пользователей имеют длину от 5 до 32 символов и не чувствительны к регистру, но могут включать только латинские символы, цифры и символы подчеркивания. Имя пользователя вашего бота должно заканчиваться на “bot”, в данном дипломном проекте бот назван @findticket_bot.

Токен авторизации – это набор символов вида 110201543:AAHdqTcvCH1vGWJxfSeofSAs0K5PALDsaw, который нужен, чтобы получать и отправлять сообщения с помощью Bot API. Telegram рекомендует держать свой токен авторизации в безопасности, он может использоваться любым человеком для управления вашим ботом. Если ваш существующий токен скомпрометирован или вы по какой-то причине его потеряли, используйте команду /token в @BotFather, чтобы сгенерировать новый.

BotFather команды:

1) /mybots - возвращает список ваших ботов с удобными элементами управления для редактирования их настроек;

2) /mygames - возвращает список ваших игр с удобными элементами управления для редактирования их настроек.

Редактирование ботов:

1) /setname - изменить имя вашего бота;

2) /setdescription - изменить описание бота, краткий текст, длиной до 512 символов, описывающий вашего бота. Пользователи увидят этот текст в начале разговора с ботом под названием «Что может делать этот бот?»;

3) /setabouttext - изменение информации о боте, еще короче текст до 120 символов. Пользователи увидят этот текст на странице профиля бота. Когда они делятся вашим ботом с кем-то, этот текст отправляется вместе со ссылкой;

4) /setuserpic - изменить картинку профиля бота;

5) /setcommands - изменить список команд, поддерживаемых вашим ботом.

Пользователи увидят эти команды в качестве подсказок при вводе / в чат с вашим ботом. У каждой команды есть имя (должно начинаться с косой черты '/', буквенно-цифровых символов плюс подчеркивания, не более 32 символов, без учета регистра), параметров и текстового описания. Пользователи будут видеть список команд всякий раз, когда они будут вводить '/' в разговоре с вашим ботом. К примеру, «/deletebot» - удалить вашего бота и освободить его имя пользователя.

Изменение настроек:

1) /setinline - включить встроенный режим для вашего бота;

2) /setinlinegeo - запрашивать данные о местоположении, чтобы предоставить основанные на местоположении результаты;

3) /setjoingroups – позволить боту быть добавленным в группы; любой бот должен иметь возможность обрабатывать личные сообщения, но если ваш бот не предназначен для работы в группах, вы можете отключить это;

4) /setprivacy - установить, какие сообщения будет получать ваш бот при добавлении в группу. Если режим конфиденциальности отключен, бот будет получать все сообщения. Мы рекомендуем оставить режим конфиденциальности включенным. Чтобы изменения вступили в силу, вам нужно будет повторно добавить бота в существующие группы.

Управление играми:

1) /newgame - создать новую игру;

2) /listgames - получить список ваших игр;

3) /editgame - редактировать игру;

4) /deletegame - удалить существующую игру.

2.2.1 Отличие ботов от обычных пользователей

1) У ботов нет онлайн-статуса и последних посещенных меток времени, вместо этого на интерфейсе отображается метка «бот»;

2) У ботов ограниченное облачное хранилище - старые сообщения могут быть удалены сервером вскоре после их обработки;

3) Боты не могут инициировать разговоры с пользователями. Пользователь должен либо добавить их в группу, либо сначала отправить им сообщение. Люди могут использовать ссылки t.me/ <bot_username> или поиск по имени пользователя, чтобы найти вашего бота;

4) Имена пользователей ботов всегда заканчиваются на «бот» (например, @TriviaBot, @GitHub_bot);

5) При добавлении в группу, боты не получают все сообщения по умолчанию (режим конфиденциальности).

Встроенный режим:

Пользователи могут взаимодействовать с вашим ботом с помощью встроенных запросов прямо из поля ввода текста в любом чате. Все, что им нужно сделать, это начать сообщение с именем пользователя вашего бота, а затем ввести запрос. Получив запрос, ваш бот может вернуть некоторые результаты. Как только пользователь нажимает на один из них, он отправляется в открытый в данный момент чат пользователя. Таким образом, люди могут запрашивать контент у вашего бота в любом из своих чатов, групп или каналов.

Платежная платформа:

Пользователи могут использовать ботов для приема платежей от пользователей Telegram по всему миру. С точки зрения пользователей система является полностью цельной. Ваш бот отправляет сообщение в специально отформатированном счете. Такие сообщения содержат фотографию и описание продукта, а также заметную кнопку оплаты. Нажатие на эту кнопку открывает специальный интерфейс оплаты прямо в приложении Telegram. Пользователь вводит необходимую информацию, выбирает одну из своих сохраненных карт или вводит новую (Telegram также поддерживает Apple Pay и Android Pay) - и оплачивает продукт.

Игровая платформа:

Боты могут предлагать своим пользователям HTML5-игры, чтобы они играли в одиночку или соревновались друг с другом в группах и чатах один на один. Платформа позволяет вашему боту отслеживать высокие баллы за каждую игру, сыгранную в каждом чате. Всякий раз, когда в игре появляется новый лидер, другие участники чата уведомляются о том, что они должны побить его рекорд. Поскольку основной технологией является HTML5, игры могут быть чем угодно: от простых аркад и головоломок до многопользовательских 3D-шутеров и стратегических игр в реальном времени.

Клавиатуры:

Чат-ботов можно научить понимать человеческий язык. Но иногда боту необходимо получить более формальный ввод от пользователя - и именно здесь пользовательские клавиатуры могут стать чрезвычайно полезными. Всякий раз, когда бот отправляет сообщение, он может передавать специальную клавиатуру с предопределенными параметрами ответа (ReplyKeyboardMarkup). Приложения Telegram, получающие сообщение,

будут отображать вашу клавиатуру пользователю. Нажатие любой из кнопок немедленно отправит соответствующую команду. Таким образом, вы можете существенно упростить взаимодействие пользователя с вашим ботом. В настоящее время поддерживается текст и эмодзи для ваших кнопок.

Встроенные клавиатуры и оперативное обновление:

Есть моменты, когда вы предпочитаете делать что-то, не отправляя сообщения в чат. Например, когда ваш пользователь меняет настройки или просматривает результаты поиска. В таких случаях вы можете использовать встроенные клавиатуры, которые интегрированы непосредственно в сообщения, к которым они принадлежат. В отличие от пользовательских клавиатур для ответов, нажатие кнопок на встроенных клавиатурах не приводит к отправке сообщений в чат. Вместо этого встроенные клавиатуры поддерживают кнопки, которые работают за сценой: кнопки обратного вызова, кнопки URL и переключаются на встроенные кнопки. Когда используются кнопки обратного вызова, ваш бот может обновлять свои существующие сообщения (или только их клавиатуры), чтобы чат оставался чистым [4].

Глобальные команды:

Чтобы облегчить пользователям навигацию по боту, Telegram просит всех разработчиков поддержать несколько основных команд. Приложения Telegram будут иметь ярлыки интерфейса для этих команд.

1) /start - начинает взаимодействие с пользователем, например, отправляя приветственное сообщение. Эту команду также можно использовать для передачи дополнительных параметров боту (Глубокая связь);

2) /help - возвращает справочное сообщение. Это может быть краткий текст о том, что может делать бот, и список команд;

3) /settings - возвращает настройки бота для этого пользователя и предлагает команды для редактирования этих настроек.

Пользователи увидят кнопку «START», когда впервые откроют разговор с вашим ботом.

Форматирование: полужирный, курсив, текст с фиксированной шириной и встроенные ссылки. При создании бота имеется возможность использовать жирный шрифт, курсив или текст фиксированной ширины, а также встроенные ссылки в сообщениях ваших ботов. Клиенты Telegram будут отображать их соответственно.

Режим конфиденциальности:

Боты часто добавляются в группы, чтобы улучшить общение между людьми, например, предоставляя новости, уведомления от внешних служб или дополнительные функции поиска. Это особенно актуально для чатов с коллегами. Бот, работающий в режиме конфиденциальности, не будет получать все сообщения, которые люди отправляют в группу. Вместо этого он получит только:

- сообщения, начинающиеся с косой черты «/» (см. Команды выше);

- ответы на собственные сообщения бота;

- служебные сообщения (люди добавлены или удалены из группы и т.д.);

- сообщения с каналов, на которых он зарегистрирован (это помогает разработчику бота экономить много ресурсов, так как им не нужно обрабатывать десятки тысяч ненужных сообщений каждый день).

Режим конфиденциальности включен по умолчанию для всех ботов, кроме ботов, которые были добавлены в группу как администраторы (боты - администраторы всегда получают все сообщения). Его можно отключить, чтобы бот получал все сообщения, как обычный пользователь (бот должен будет быть повторно добавлен в группу, чтобы это изменение вступило в силу). Telegram рекомендует делать это только в тех случаях, когда это необходимо для работы вашего бота. Пользователи всегда могут увидеть текущие настройки конфиденциальности бота в списке участников группы. В большинстве случаев использование опции принудительного ответа для сообщений бота должно быть более чем достаточно.

Местоположение и номер:

Некоторые боты нуждаются в дополнительных данных от пользователя для правильной работы. Например, знание местоположения пользователя помогает обеспечить более релевантные геоспецифичные результаты. Телефонный номер пользователя может быть очень полезен для интеграции с другими службами, такими как банки и т. д. Боты могут запрашивать у пользователя их местоположение и номер телефона с помощью специальных кнопок. Необходимо обратить внимание, что кнопки телефона и запроса местоположения будут работать только в личных чатах. Когда эти кнопки будут нажаты, клиенты Telegram отобразят предупреждение о том, что должно произойти.

Оповещения о состоянии:

Миллионы пользователей выбирают Telegram за его скорость. Чтобы оставаться конкурентоспособным в этой среде, бот также должен быть отзывчивым. Чтобы помочь разработчикам поддерживать ботов в форме, Botfather будет отправлять оповещения о состоянии, если обнаружит, что что-то не так. Telegram будет проверять количество ответов и коэффициент конверсии запросов/ответов для популярных ботов. Если показатели будут ненормально низкие, вы получите уведомление от BotFather [14].

2.3 Выбор языка написания бота

2.3.1 Различия между Python и JavaScript

Python – это интерпретируемый язык программирования высокого уровня с динамической семантикой и объектно-ориентированным программированием, разработанный для удобства чтения и реализации. Это язык сценариев, такой как Perl/Ruby, который также используется для

создания веб-приложений. поддерживает различные парадигмы программирования, такие как объектно-ориентированное программирование, функциональное программирование, императивное программирование и процедурное программирование. Он придумал огромные встроенные модули и пакеты. Это позволяет программистам использовать разные стили программ для простых и сложных программ.

Java-Script – это объектно-ориентированный язык программирования, который позволяет создавать динамические веб-страницы и стандартизирован в спецификации языка ECMAScript (ECMAScript была создана для стандартизации JavaScript, чтобы способствовать развитию реализаций). Java-Script поддерживает различные парадигмы программирования, такие как объектно-ориентированное, функциональное и императивное программирование, но не процедурное программирование. Он широко используется в браузерах для обеспечения динамической функциональности, которую мы не можем достичь с помощью обычного HTML и CSS. Он поддерживает стандартные приложения с текстом, регулярными выражениями и датами.

Наряду с HTML и CSS, JavaScript является одной из основных технологий Всемирной паутины. JavaScript обеспечивает интерактивные веб-страницы и является важной частью веб-приложений. Подавляющее большинство веб-сайтов используют его для поведения страниц на стороне клиента, и все основные веб-браузеры имеют специальный механизм JavaScript для его выполнения.

JavaScript поддерживает управляемые события, функциональные и императивные стили программирования. Он имеет интерфейсы прикладного программирования (API) для работы с текстом, датами, регулярными выражениями, стандартными структурами данных и объектной моделью документа (DOM). Однако сам язык не включает в себя никакого ввода/вывода (I/O), такого как сетевое оборудование, хранилище или графические средства, поскольку среда хоста предоставляет эти API [16].

Двигатели JavaScript первоначально использовались только в веб - браузерах, но теперь они встроены в некоторых серверах, как правило, с помощью Node.js.

Для написания Telegram бота для поиска авиабилетов останавливаемся на языке Java-Script из-за удобства создания серверных приложений и переходим к выбору среды.

2.4 Выбор среды выполнения кода

Node.js представляет среду выполнения кода на JavaScript, которая построена на основе движка JavaScript Chrome V8, который позволяет транслировать вызовы на языке JavaScript в машинный код. Node.js прежде всего предназначен для создания серверных приложений на языке JavaScript.

Хотя также существуют проекты по написанию десктопных приложений и даже по созданию кода для микроконтроллеров. Но прежде всего мы говорим о Node.js, как о платформе для создания веб-приложений.

Для разработки под Node JS, достаточно простейшего текстового редактора, в частности, Notepad++. Также можно использовать более изощренные редакторы типа Atom, Sublime, Visual Studio Code, либо среды разработки, которые поддерживают работу с Node.JS, например, Visual Studio или WebStorm. В данной дипломной работе используется Sublime Text.

Node.js позволяет создавать веб-серверы и сетевые инструменты, используя JavaScript и набор «модулей», которые выполняют различные основные функции. Модули предназначены для ввода/вывода файловой системы, работы в сети, двоичных данных (буферы), функций криптографии, потоков данных и другие основные функции. Модули Node.js используют API для уменьшения сложности написания серверных приложений.

JavaScript - единственный язык, который Node.js поддерживает изначально, но доступно много языков компиляции в JS. В результате приложения Node.js могут быть написаны на CoffeeScript, Dart, TypeScript, ClojureScript и других.

Node.js, в основном используется для создания сетевых программ, таких как веб-серверы. Наиболее существенным отличием между Node.js и PHP является то, что большинство функций в PHP блокируются до завершения (команды выполняются только после завершения предыдущих команд), тогда как функции Node.js не блокируются (команды выполняются одновременно или даже параллельно, и использовать обратные вызовы для оповещения о завершении или сбое) [17].

3 Экспериментальная часть

3.1 Проектирование

Программа состоит из нескольких модулей:

- bot.js – содержит код, который отвечает за интеграцию с Telegram;
- searchScene.js – класс и функции ответственные за контекст чата;
- api.js – класс, получающий серверные данные;
- parser.js - класс, отвечающий за преобразование данных, полученных сервера, в надлежащий формат;
- formatter.js – класс, отвечающий за преобразование данных в читаемый для пользователя формат.

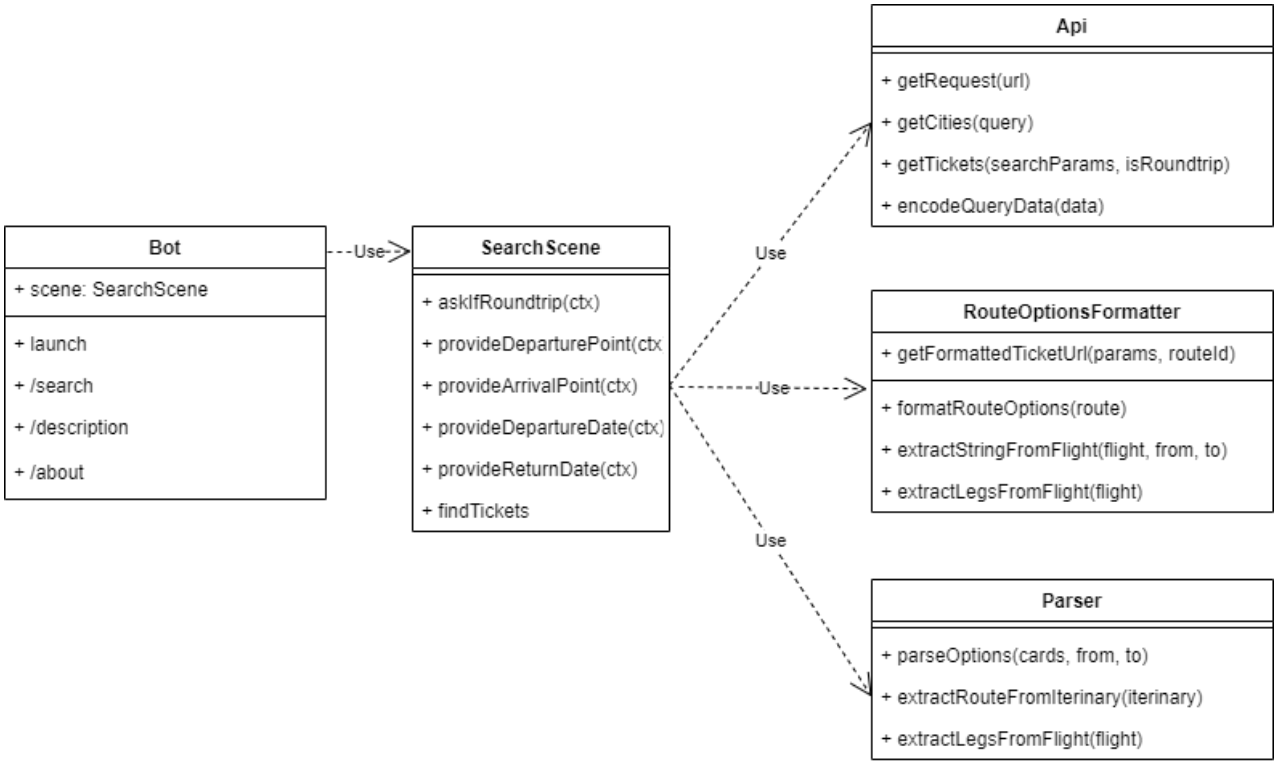

Рисунок 3.1 – Описание классов

Основная логика Telegram бота происходит в классе SearchScene:

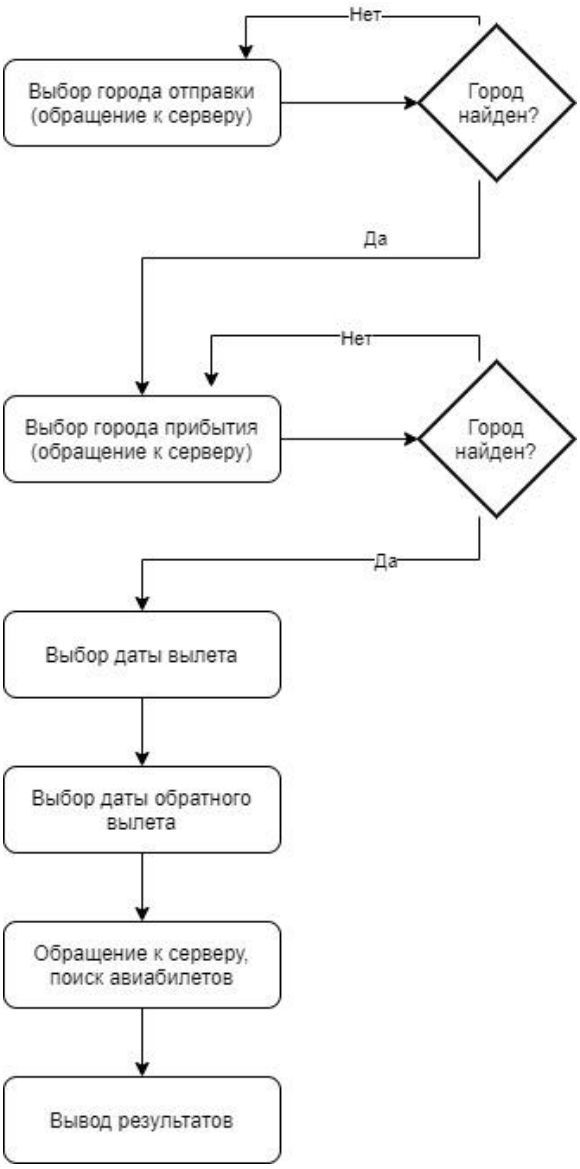

Рисунок 3.2 – Описание классов

3.2 Программная реализация

Для начала работы с ботом нам потребуется Командная строка. Для того чтобы запустить командную строку необходимо открыть приложение «Выполнить», ввести cmd и нажать «ОК»:

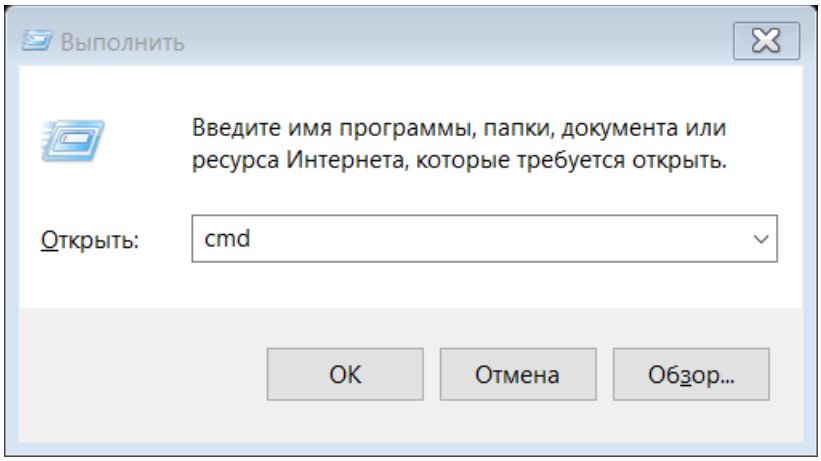

Рисунок 3.3 – Запуск приложения «Выполнить»

По умолчанию путь каталога установлен C:\Users\Alina Bukhmetova.

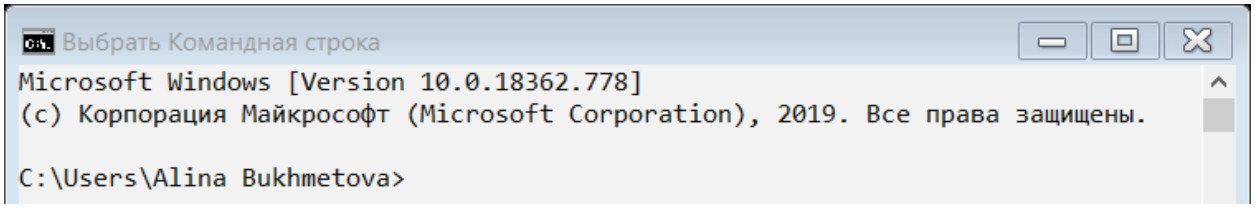

Рисунок 3.4 – Командная строка

Нам необходимо сменить наш текущий каталог на каталог в котором находится разработанный бот. Используем команду cd для смены текущего каталога на нам необходимый:

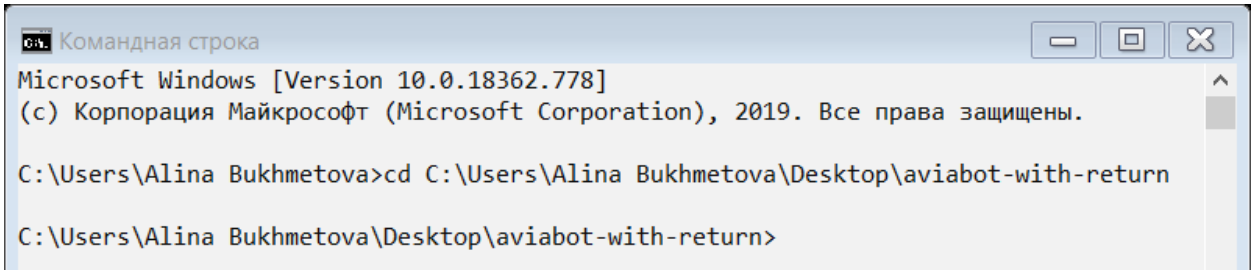

Рисунок 3.5 – Смена текущего каталога

Запуск бота осуществляется посредством ввода “node bot.js”, после смены текущего каталога, с помощью программной платформы Node.js, заранее установленной на компьютере:

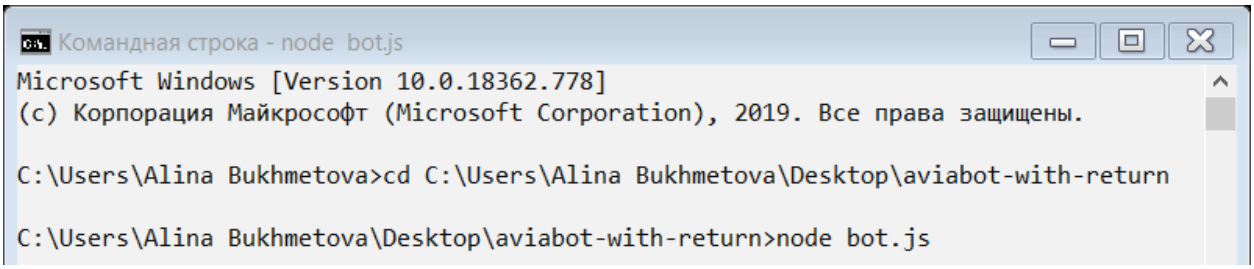

Рисунок 3.6 – Запуск файла Bot.js, отвечающего за интеграцию с Telegram

Вбиваем в поиск наименование разработанного бота @findticket_bot, на экране появляется краткое описание того, что умеет делать бот, нажимаем на кнопку “Начать”:

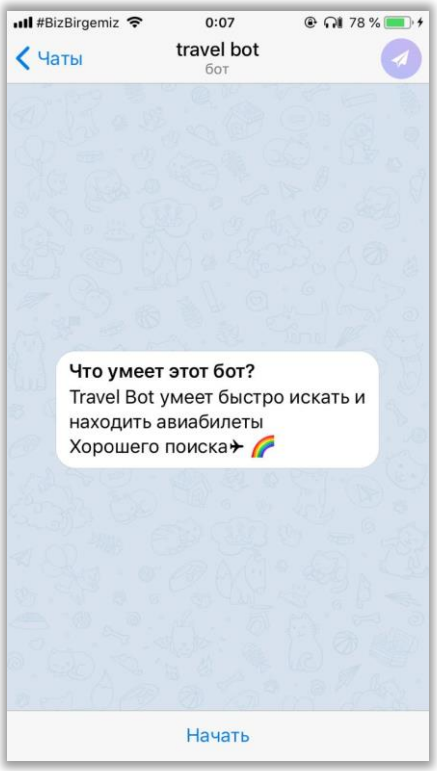

Рисунок 3.7 – Страница приветствия

Происходит автоматический ввод глобальной команды /start, что позволяет начать работу с ботом:

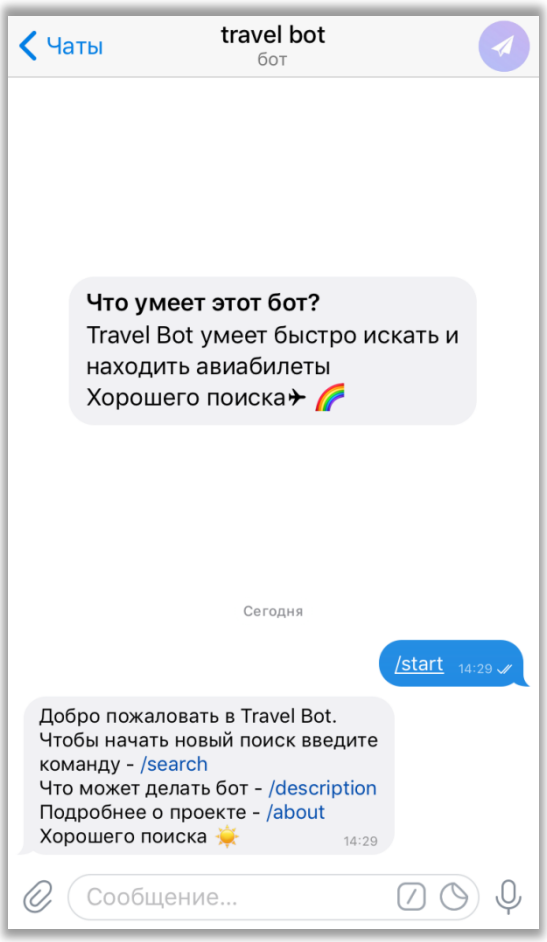

Рисунок 3.8 – Реализация глобальной команды /start

При вводе «/» автоматически появляются все доступные глобальные команды разработанного бота. Внести изменения в данный список можно отправив @BotFather команду /setcommands и ввести необходимые команды и их описание в формате:

“command1 – описание
command2 – другое описание”

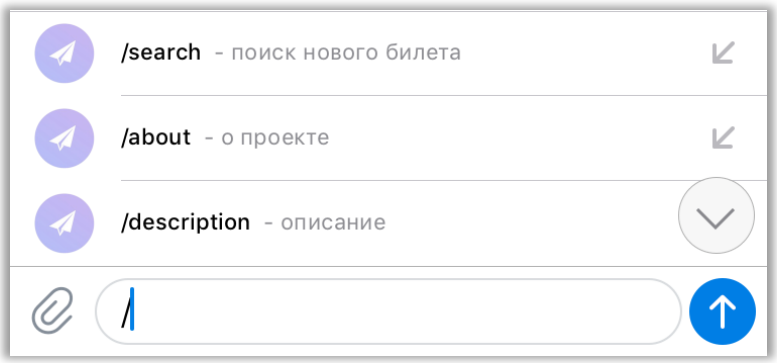

Рисунок 3.9 – Доступные глобальные команды

При вводе команды /description появляется описание того, что может делать бот и пожелание “Хорошего поиска”:

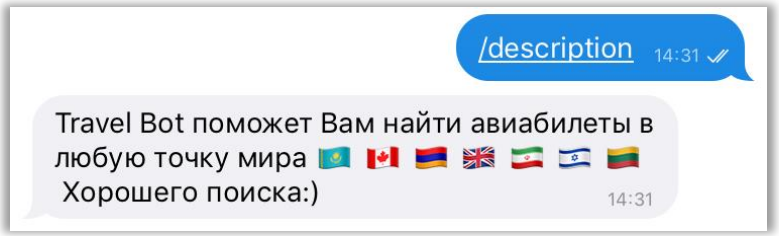

Рисунок 3.10 – Реализация глобальной команды /description

При вводе команды /about, появляется информация о проекте:

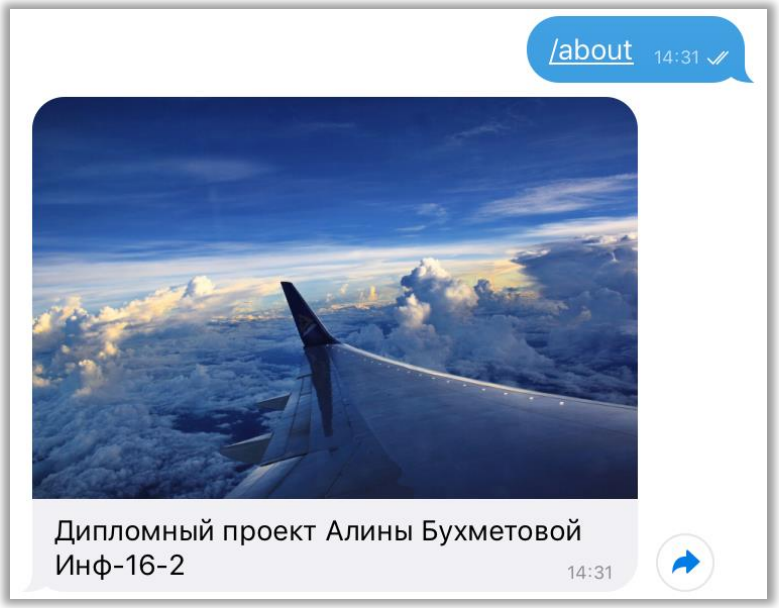

Рисунок 3.11 – Реализация глобальной команды /about

При вводе команды /search мы начинаем поиск билетов:

Пример 3.1, в случае необходимости обратного билета на вопрос «Вам нужен обратный билет?» отвечаем да, в таком случае, бот запросит город вылета, город прибытия, желаемую дату вылета и дату обратного билета.

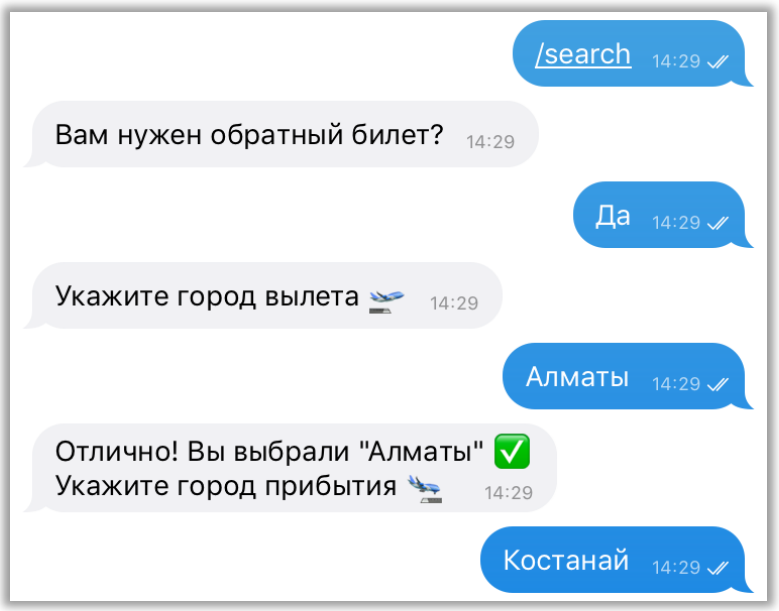

Рисунок 3.12 – Реализация Примера 3.1

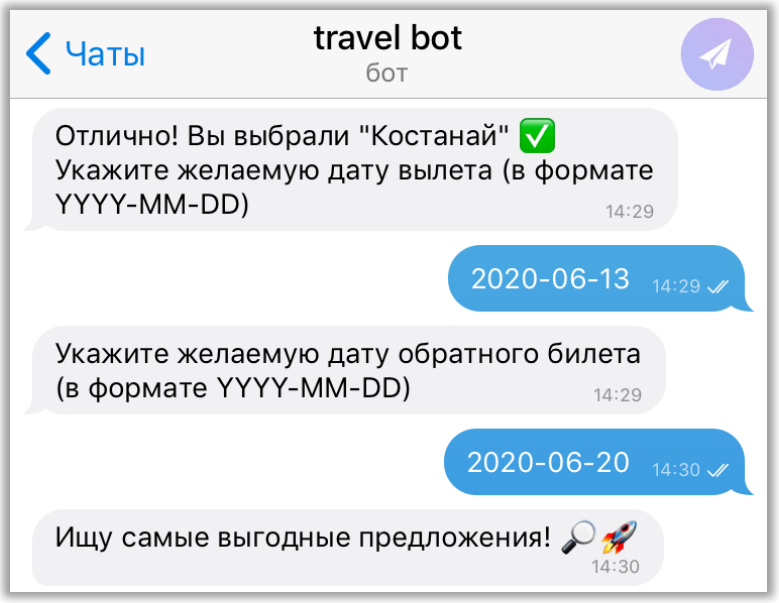

Рисунок 3.13 – Реализация Примера 3.1

Бот предлагает выгодные предложения с сайта https://santufei.com/, также кликнув на “Купить на Santufei.kz”, вы переходите на официальный сайт, для покупки билета именно на выбранный вами рейс.

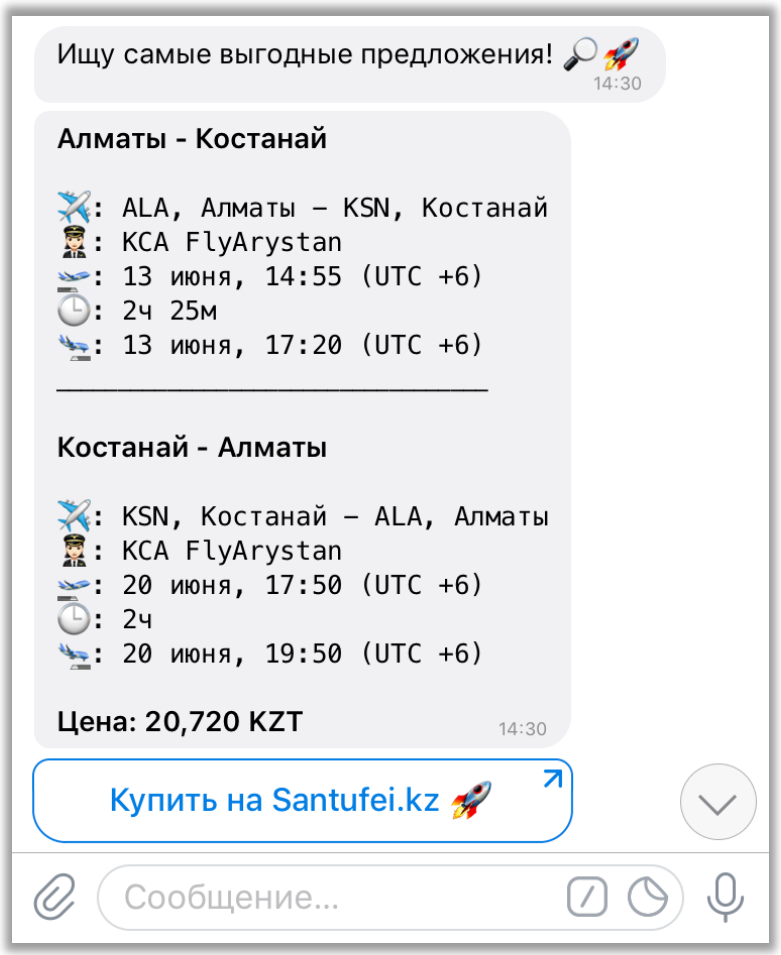

Рисунок 3.14 – Реализация Примера 3.1

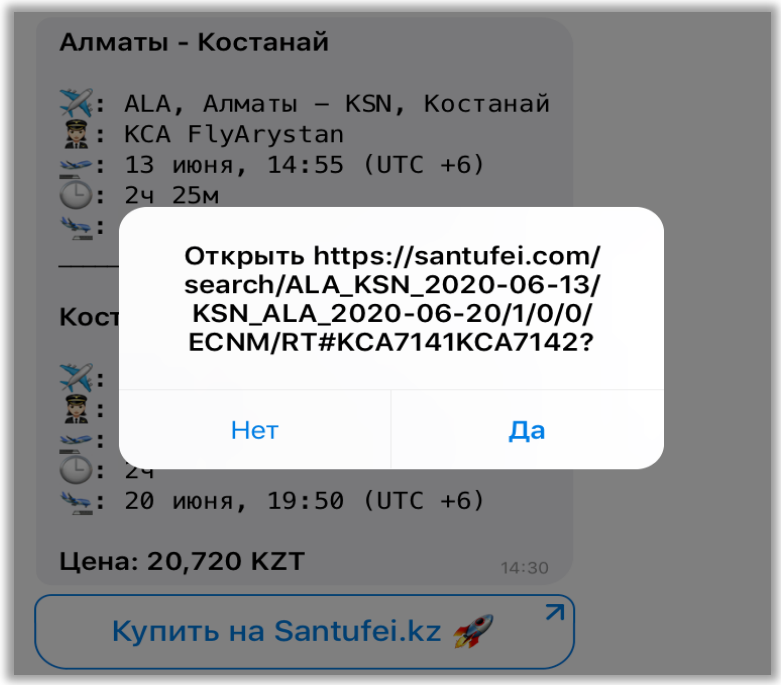

Рисунок 3.15 – Реализация Примера 3.1

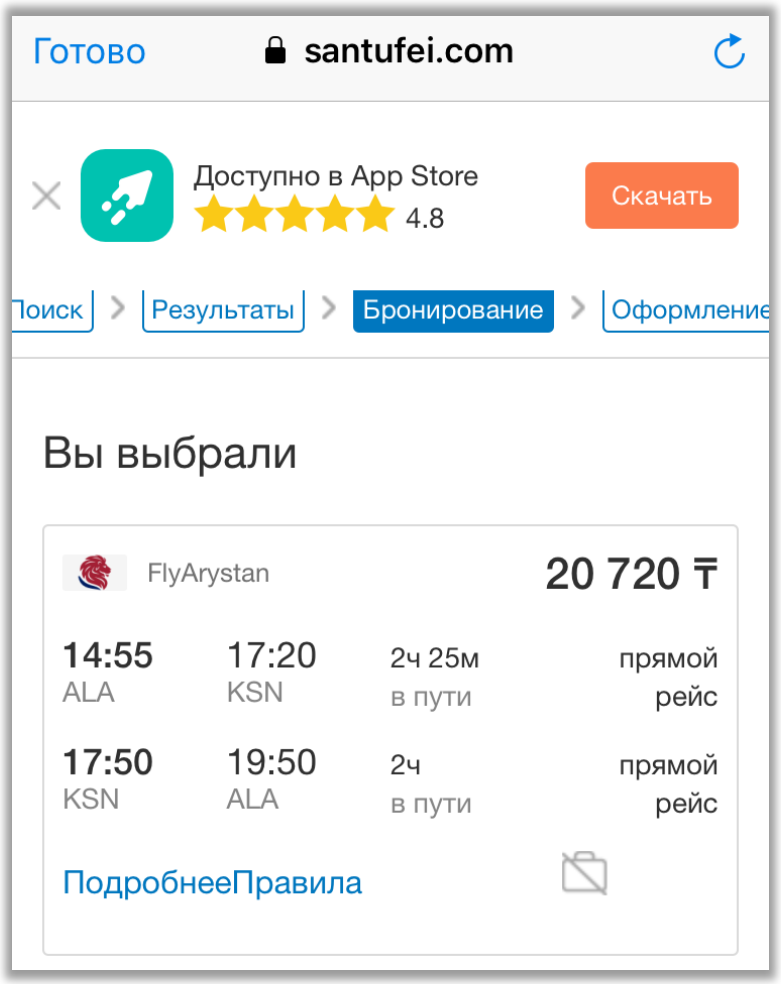

Рисунок 3.16 – Реализация Примера 3.1

Пример 3.2, если нет необходимости в обратном билете, на вопрос «Вам нужен обратный билет?» отвечаем «нет», в таком случае, бот запросит город вылета, город прибытия, желаемую дату вылета без даты обратного билета.

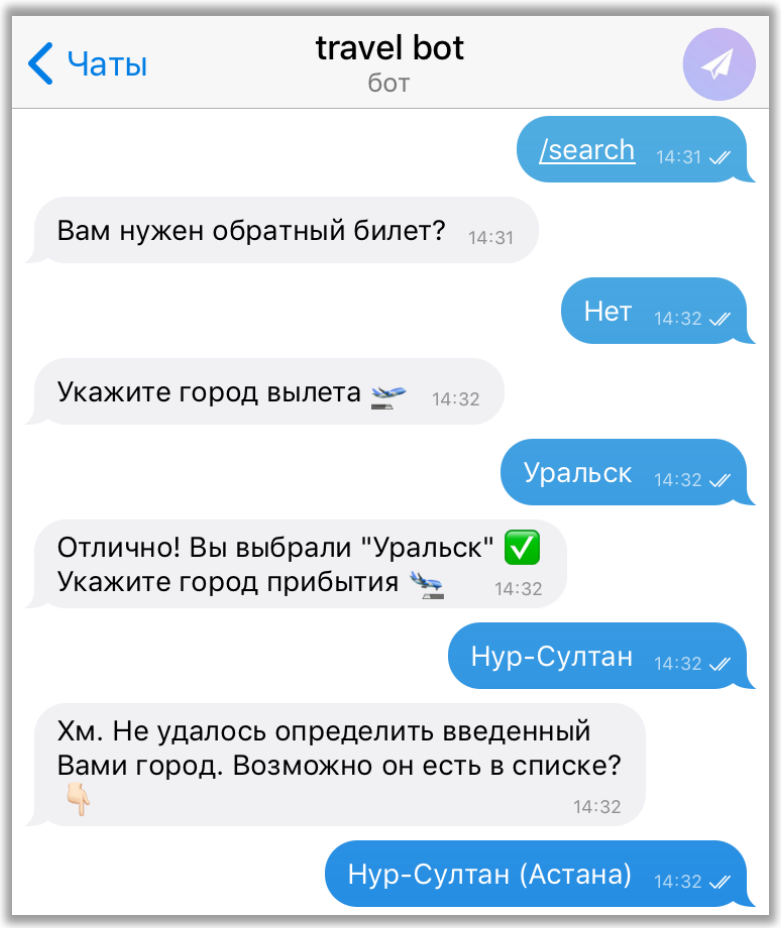

Рисунок 3.17 – Реализация Примера 3.2

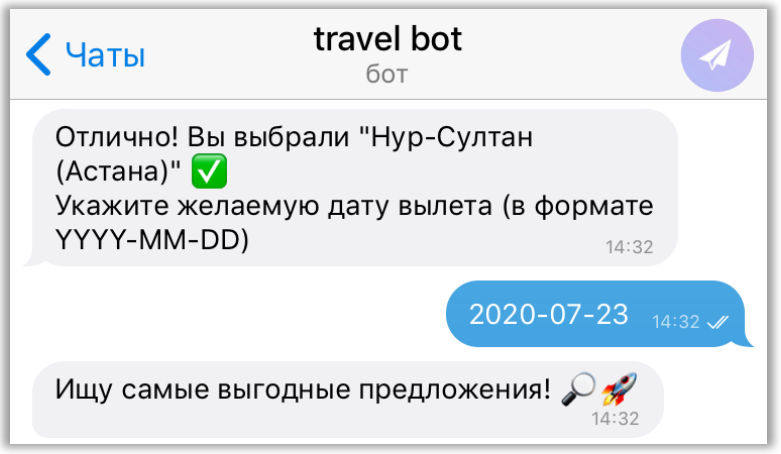

Рисунок 3.18 – Реализация Примера 3.2

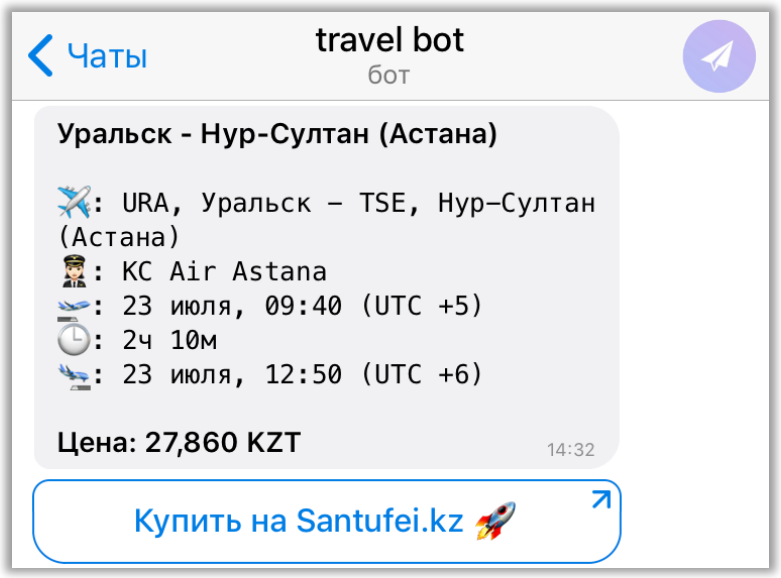

Рисунок 3.19 – Реализация Примера 3.2

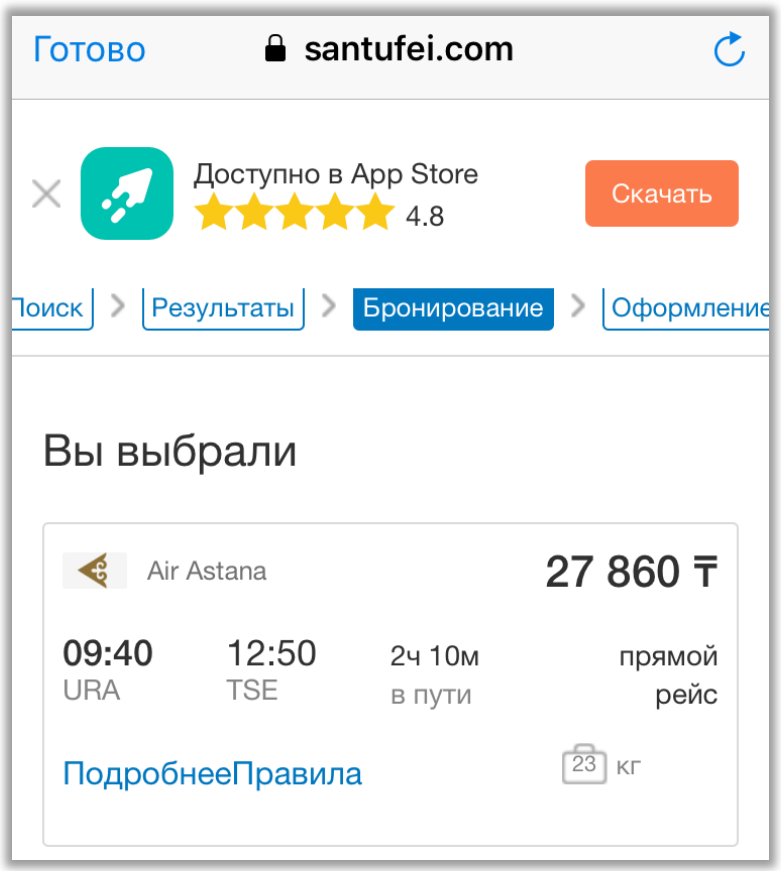

Рисунок 3.20 – Реализация Примера 3.2

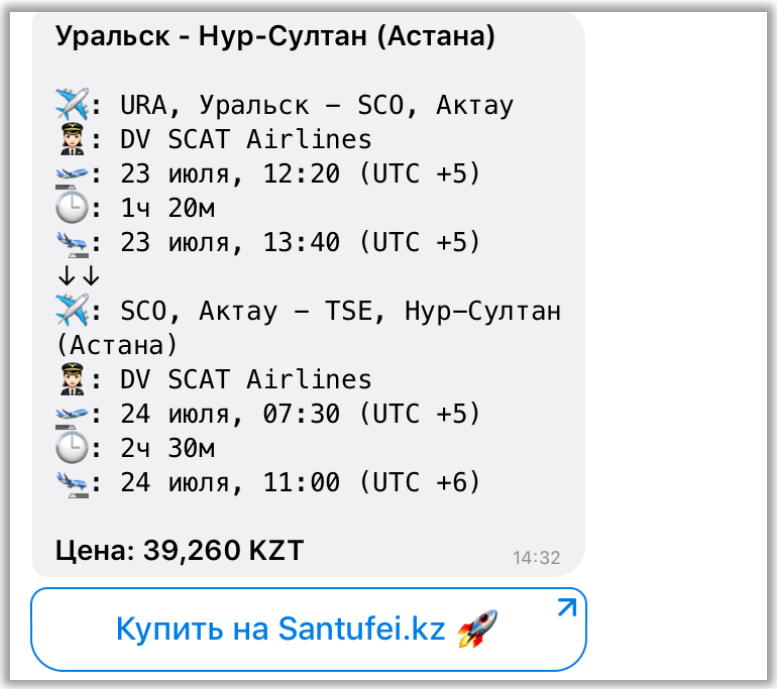

Рисунок 3.21 – Реализация Примера 3.2

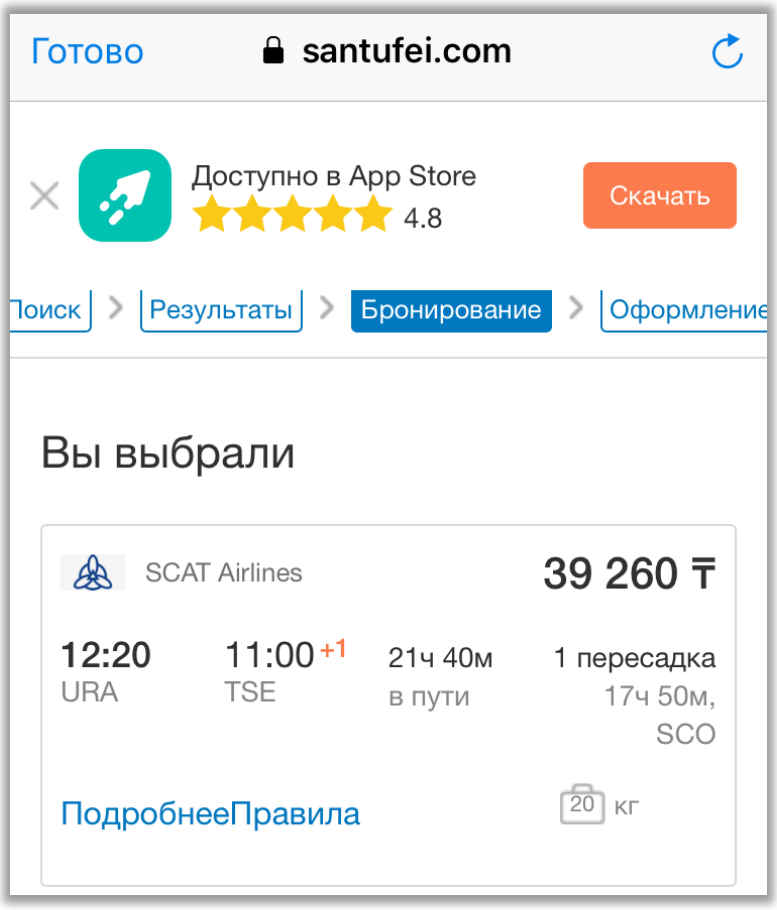

Рисунок 3.22 – Реализация Примера 3.2

4 Экономическая часть

4.1 Введение

В данной дипломной работе разрабатывается Telegram бот для поиска авиабилетов. Разрабатываемый бот несёт в себе информационный характер и позволяет, не выходя из популярного мессенджера, найти авиабилеты, необходимые пользователю. Пользователь может выбрать город отправления, дату отправления и при необходимости также запросить информацию об обратном билете. Бот предоставит информацию по всем рейсам выбранной даты и благодаря удобному интерфейсу, пользователь сможет получить информацию, выбрать для себя оптимальный рейс и купить билет, перейдя по ссылке. Основная эффективность Telegram-бота состоит в экономии времени и быстром доступе к информации, а также автоматизации устаревающих процессов. Основной задачей экономической части является расчет трудоемкости, себестоимости и определение экономической эффективности.

4.2 Трудоемкость разработки ПП

Таблица 4.1 – Распределение работ по этапам и видам и оценка их трудоемкости

Этапы разработки ПП	Вид работы на данном этапе	Трудоемкость разработки ПП	
		Чел. х час	Час х день
Анализ требований	Анализ предметной области, выявление целей и задач	1 х 16	8 х 2
Анализ рынка	Анализ конкурентов, подобных продуктов, потребность в данном продукте, нахождение преимуществ перед альтернативными товарами	1 х 16	8 х 2
Проектирование	Формирование требований к проекту, к интерфейсу, разработка и получение технического задания, индивидуальные требования руководства к продукту	1 х 32	8 х 4
Реализация	Построение диаграмм, реализация интерфейса, разработка ПО	1 х 40	8 х 5

Продолжение таблицы 4.1

Тестирование продукта	Тестирование, исправление ошибок и неполадок продукта	1 х 48	8 х 6
Подготовка инструкции	Подготовка подробной и понятной инструкции и руководства по работе с программой	1 х 8	8 х 1
Внедрение и поддержка	Установка программного обеспечения, исправление выявленных ошибок, сопровождение программного продукта.	1 х 32	8 х 4
Итого трудоемкость выполнения проекта		1 х 192	8 х 24

4.3 Расчет затрат на разработку ПП

Нужно рассчитать следующие затраты для составления:
- материальные затраты;
- затраты на оплату труда;
- социальный налог;
- амортизация основных фондов.

4.4 Материальные затраты

Затраты по материалам, необходимым для программного продукта, рассчитываются с помощью таблицы

Таблица 4.2 – Стоимость оборудования и ПО

№	Наименование	Описание	Цена за единицу, тг	Сумма, тг
1	Ноутбук	HP 15-rb028ur	98 000	98 000
2	Операционная система	Microsoft Windows 10 Профессиональная	85 000	85 000
4	Антивирус	Dr. Web	2 700	2 700
5	Принтер	HP Laser 107a	40 000	40 000
Итого:				225 700

Также необходимо учесть затраты машинного времени на разработку информационной системы. Затраты машинного времени за время разработки ИС определяются по формуле:

$$З_м = К \times q \quad (4.1)$$

где К – количество часов использования ПК за время разработки ИС;

q – стоимость часа машинного времени (146 тенге/час) [9].

На разработку, внедрение и поддержку суммарно затрачено 160 часов. Исходя из этого, получится:

$$З_м = 160 \times 146 = 23\ 360 \text{ тг.}$$

4.5 Затраты на электроэнергию

Общая сумма затрат рассчитывается по формуле (4.2):

$$З_э = \sum_{i=1}^{n} М_i * K_i * T_i * Ц \quad (4.2)$$

С 1 января 2020 года цена на электроэнергию по тарифу ТОО «АлматыЭнергоСбыт» составляет 19,17 тенге за 1 кВтч с НДС.

Таблица 4.4 –Затраты на технологические нужды

Наименование оборудования	Паспортная мощность, кВт	Коэффициент использования мощности	Время работы оборудования для разработки ПП, ч	Цена элекроэнергии, тг/кВт*ч	Сумма, тг
Ноутбук	0,4	0,7	192	19,17	1030,579
Принтер	0,35	0,7	16	19,17	75,1464
Итого затраты на электроэнергию					1105,726

4.6 Затраты на оплату труда

В этой статье идет расчет оплаты труда всех работников, которые были задействованы в разработке программного продукта. Общая сумма затрат на оплату труда рассчитывается по формуле (4.3):

$$З_{тр} = \sum_{i=1}^{n} ЧС_i * T_i \quad (4.3)$$

Часовая ставка работника равняется – 600 тг/час.

Таблица 4.5 – Затраты на оплату труда

Категория работника	Трудоемкость разработки ПП, чел. х ч	Часовая ставка, тг/ч	Сумма, тг
Разработчик - программист	1 х 192	600	115 200
Дополнительная зарплата	192	600×10%=60	11 520
ИТОГО затрат на оплату труда:			126 720

4.7 Социальный налог

Отчисления на социальные нужды 10,46 % от затрат которых баннеры ограждающих на оплату труда всех работников, но пенсионные отчисления (10% от затрат) не облагаются данным видом налога.

Таблица 4.6 – Налоговые отчисления

Уплаченные налоги юридическим лицом	10,46	ФОТ	126 720
СО (Социальные отчисления)	3,5	(ЗП - ОПВ)×3,5%	3 628,8
ВОСМСЮ (Отчисления на ВОСМСЮ)	2,0	ЗП×2%	2 304
СН (Социальный налог)	9,5	(ЗП - ОПВ - ВОСМСФ) ×9,5%-СО	6 111,36
Всего уплаченные налоги			12 044,16

Амортизационные отчисления определяются согласно Таблице 4.7. Сумма амортизационных отчислений вычисляется по формуле (4.4):

$$З_{ам} = \frac{С_{обор} * Н_а * N}{100 * 12 * t} \quad (4.4)$$

где $Н_а$ – норма амортизации (%);

$С_{обор}$ – первоначальная стоимость оборудования;

N – время использования оборудования;

t – количество рабочих дней в месяце.

Норма амортизации для линейного способа вычисляется по формуле (4.4).

Использование ОФ варьируется от 3 до 10 лет. Все используется в течение 7 лет. Программное обеспечение – 3 года. Используя формулу (4.4), заполним Таблицу 4.6 для отображения амортизации основных фондов.

$$Н_{А1} = 100/7 = 14{,}29\%.$$

$$Н_{А3} = 100/3 = 33{,}33\%.$$

Расчеты фактической амортизации:

$$З_{ам} = (98000 \times 0{,}1429 \times 24)/(1 \times 12 \times 21) = 1333{,}73 \text{ тг};$$

$$З_{ам} = (85000 \times 0{,}3333 \times 24)/(1 \times 12 \times 21) = 2698{,}14 \text{ тг};$$

$$З_{ам} = (2700 \times 0{,}3333 \times 24)/(1 \times 12 \times 21) = 85{,}71 \text{ тг};$$

$$З_{ам} = (40000 \times 0{,}1429 \times 2)/(1 \times 12 \times 21) = 45{,}37 \text{ тг}.$$

Таблица 4.7 – Амортизационные отчисления

Наименование оборудования и ПО	Стоимость оборудования и ПО, тг	Годовая норма амортизации, %	Время работы оборудования и ПО для разработки ПП, д	Сумма, тг
Ноутбук HP 15-rb028ur	98 000	14,29	24	1333,73
Microsoft Windows 10 Профессиональная	85 000	33,33	24	2698,14
Dr. Web	2 700	33,33	24	85,71
HP Laser 107a	40 000	14,29	2	45,37
Итого:				4162,95

Величина затрат на материалы на основании исходных данных определяется по формуле (4.5):

$$Mi = \frac{З_{осн} \cdot Н_{мз}}{100} \qquad (4.5)$$

где $Н_{мз}$ - норма расхода материалов от основной заработной платы (3-5%).

Величина затрат на материалы при норме расхода – 3%:

$$M_i = \frac{126\,720 \cdot 3}{100} = 3\,801{,}6 \text{ тенге}$$

Научные командировки не предусмотрены.

Расходы по статье «Прочие затраты» ($П_{зi}$) на конкретное ПО, включают затраты на приобретение и подготовку специальной научно-технической

информации и специальной литературы. Определяются по нормативу, разрабатываемому в целом по организации, в процентах к основной заработной плате:

$$П_{зi} = З_{оi} \cdot \frac{Н_{пз}}{100} \quad (4.6)$$

где $Н_{пз}$ - норматив прочих затрат в целом по организации в (%), в дипломной работе нужно брать 20%.

$$П_{з} = 126\,720 \cdot 0{,}2 = 25\,344 \text{ тенге}$$

Затраты по статье «Накладные расходы» ($Р_{нi}$), связанные с необходимостью содержания аппарата управления, вспомогательных хозяйств и опытных (экспериментальных) производств, а также с расходами на общехозяйственные нужды ($Рнi$), относятся на конкретное ПО по нормативу ($Н_{нр}$) в процентном отношении к основной заработной плате исполнителей. Норматив устанавливается в целом по организации:

$$Р_{нi} = З_{оi} \cdot \frac{Н_{нр}}{100} \quad (4.7)$$

где $Р_{нi}$ - накладные расходы на конкретное ПО (тыс. тенге);

$Н_{нр}$ - норматив накладных расходов в целом по организации в (%), в дипломной работе нужно брать 70%.

$$Р_{нi} = 126\,720 \cdot 0{,}7 = 88\,704 \text{ тенге}$$

4.8 Смета затрат на разработку ПП

Таблица 4.8 – Смета затрат на разработку ПП

Статья затрат	Сумма, тг
Стоимость оборудования и ПО	225 700
Оплата труда	126 720
Социальный налог	12 044,16
Электроэнергия	1 105,73
Амортизация основных фондов	4 162,95
Затраты на машинное время	23 360
Накладные расходы	126 720
Прочие расходы	25 344
Материальные затраты	3 801,6

ИТОГО по смете	548 958

Рентабельность и прибыль по создаваемому ПО ($П_{сi}$) определяются исходя из результатов анализа рыночных условий, переговоров с заказчиком (потребителем) и согласования с ним отпускной цены, включающей дополнительно налог на добавленную стоимость. В случае разработки ПО, для использования внутри организации оценка программного продукта производится по действующим правилам и показателям внутреннего хозрасчета (по ценам, устанавливаемым для расчета за услуги между подразделениями). Прибыль рассчитывается по формуле:

$$П_{Oi} = C_{ni} \cdot \frac{У_{рni}}{100} \quad (4.8)$$

где $П_{Oi}$ - прибыль от реализации ПО заказчику (тыс. тенге);

$У_{рni}$ - уровень рентабельности ПО (%), в дипломной работе брать 40-60%;

C_{ni} - себестоимость ПО (тыс. тенге).

$$П_{Oi} = 548\,958 \cdot 0.5 = 274\,479{,}22 \text{ тенге}$$

Прогнозируемая цена ПО без налогов ($Ц_{пi}$):

$$Ц_{пi} = C_{ni} + П_{Oi} = 548\,958 + 274\,479{,}22 = 823\,438 \text{ тенге}$$

Прогнозируемая отпускная цена ($Ц_{о1}$):

$$Ц_{01} = Ц_{ni} + НДС \quad (4.9)$$

Ставка налога на добавленную стоимость НДС в РК на 2020 год составляет 12% от отпускной цены ПО:

$$Ц_{01} = 823\,438 + \frac{823\,438 \cdot 12}{100} = 922\,250{,}18 \text{ тенге}$$

Организация-разработчик участвует в освоении ПО и несет соответствующие затраты, на которые составляется смета, оплачиваемая заказчиком по договору. Затраты на освоение определяются по нормативу ($Н_О$=10%) от себестоимости ПО в расчете на 3 месяца и рассчитываются по формуле:

$$P_{Oi} = C_{ni} \cdot \frac{Н_О}{100} = 548\,958 \cdot 0{,}1 = 54\,895{,}8 \text{ тенге}$$

Затраты на сопровождение ПО ($Р_{Ci}$). Организация-разработчик осуществляет сопровождение ПО и несет соответствующие расходы, которые оплачиваются заказчиком в соответствии с договором и сметой на сопровождение. Затраты на сопровождение определяются по установленному нормативу ($Н_С$=20%) от себестоимости ПО (в расчете на год) и рассчитываются по формуле:

$$P_{Ci} = C_{ni} \cdot \frac{Н_С}{100} = 548\,958 \cdot 0{,}2 = 109\,791{,}69 \text{ тенге}$$

Капиталовложения программного обеспечения с учетом затрат на освоение и сопровождение составит:

$$K = 922\,250{,}18 + 54\,895{,}8 + 109\,791{,}69 = 1\,086\,937{,}71 \text{ тенге}$$

4.9 Оценка эффективности внедрения программных средств

Экономическая эффективность была рассчитана программистом-разработчиком. Затраты на решение задачи без использования программного средства рассчитываются по формуле (4.10):

$$З_{тр} = ФЗП_р + ОТ_{з/п} \quad (4.10)$$

где $ФЗП_р$ – фонд заработной платы работников, решающих данную задачу;

$ОТ_{з/п}$ – отчисления на социальные нужды (10,46%).

Фонд заработной платы работников рассчитывается по формуле (4.11):

$$ФЗП_р = ЗП_р * N * 12 \quad (4.11)$$

где $ЗП_р$ – оклад работника, тенге/месяц;

N – количество работников.

Оклад работника составляет 150 000 тенге в месяц.

Исходя из этого, фонд заработной платы сотрудников за год трафика составляет:

$$ФЗП_р = 150\,000 * 12 = 1\,800\,000 \text{ тг.}$$

$$ОТ_{\frac{з}{п}} = 1800000 * 10{,}46\% = 188480 \text{ тг.}$$

Далее рассчитываются затраты на решение задач без использования программного продукта:

$$З_{тр} = 1800000 + 188480 \ = 1988280 \text{ тг.}$$

Годовые затраты машинного времени определяются по формуле (4.12):

$$З_{м} = K * q * 12, \quad (4.12)$$

где К количество часов использования ПК в месяц;

q – стоимость часа аренды сервера (146 тенге/час).

С учетом 8 часового рабочего дня, а также 21 рабочего дня в месяц, получаем часы использования ПК в месяц - К=168 часов. Исходя из этого, получится:

$$З_{м} = 168 * 146 * 12 = 294\,336 \text{ тг}$$

Суммарные затраты после внедрения программного продукта ($З_{ом}$) составят 294 336 тг.

Экономия затрат от внедрения программного продукта определяется по формуле (4.13):

$$Э = З_{тр} - З_{ом} \quad (4.13)$$

где $З_{тр}$ – затраты до внедрения системы;

$З_{ом}$ – затраты после внедрения системы.

Подставив значения, получим следующее:

$$Э = 1988280 \ - 294\,336 = 1693944 \text{ тг.}$$

Так как разработанная информационная система несет экономический эффект, целесообразно оценивать его эффективность за счет экономии в сравнении с предыдущим периодом работы без использования ИС.

Величина ожидаемого годового экономического эффекта от внедрения ИС рассчитывается по формуле (4.14):

$$Э_{г} = Э_{уг} - К \ \cdot Е_{Н} \quad (4.14)$$

где $Э_{г}$ – ожидаемый годовой экономический эффект, тенге:

$Э_{уг}$ – ожидаемая условная годовая экономия, тенге;

К – капитальные вложения, тенге;

$Е_{Н}$ - нормативный коэффициент экономической эффективности капитальных вложений.

Нормативный коэффициент экономической эффективности капитальных вложений определяется по формуле (4.15):

$$E_Н = \frac{1}{T_Н} \quad (4.15)$$

где $T_н$ — нормативный срок окупаемости капитальных вложений, лет.

Нормативный срок окупаемости капитальных вложений принимается исходя из срока морального старения - технических средств и проектных решений ИС ($T_н$=1,2,3...n), для программных продуктов срок окупаемости принимаем равным 4 года.

$$E_Н = \frac{1}{4} = 0{,}25$$

$$Э_г = 1\,693\,944 - 1\,086\,937{,}71 \cdot 0{,}25 = 1\,422\,209{,}57 \text{ тг}$$

Расчетный коэффициент экономической эффективности капитальных вложений составляет:

$$E_р = \frac{Э_{уг}}{К} \quad (4.16)$$

где $E_р$ – расчетный коэффициент экономической эффективности капитальных вложений;

$Э_{уг}$ – ожидаемая условная годовая экономия, тенге;

К – капитальные вложения на создание системы, тенге.

$$E_р = \frac{1\,693\,944}{1\,086\,937{,}71} = 1{,}56$$

Расчетный срок окупаемости капитальных вложений составляет:

$$T_Р = \frac{1}{E_Р} \quad (4.17)$$

где $E_р$ - коэффициент экономической эффективности капитальных вложений.

$$T_Р = \frac{1}{1{,}56} = 0{,}64 \text{ года} \approx 8 \text{ месяцев}$$

Таблица 4.9 – Показатели сравнительной экономической эффективности от внедрения программного продукта

Наименование показателей	Значение
Условная годовая экономия затрат, тенге	1 693 944
Коэффициент экономической эффективности капитальных	1,56

вложений ($E_р$)	
Срок окупаемости капитальных вложении ($Т_р$)	0,64

4.10 Заключение по экономическому разделу

Таким образом, разработанная информационная система позволила не только упросить процесс управления работой проектной группы организации и процесс принятия решений руководством, но и значительно сэкономить на различных факторах. Во-первых, автоматизация процесса управления позволит больше не использовать услуги дополнительных кураторов проектов по филиалам, так как теперь отсутствует необходимость осуществлять контроль непосредственно на месте филиала, вся работа организации сосредоточена в ИС. Во-вторых, значительно снизится расход материалов, за счет того, что назначение на задания и все изменения по заданиям можно сделать прямиком в системе, не прибегая к формированию документов по назначению и их распечатке.

Ожидаемый годовой экономический эффект составляет 1 693 944 тенге. Приложение окупится за 8 месяцев после начала использования.

5 Безопасность жизнедеятельности

5.1 Введение

В данной дипломной работе разрабатывается Telegram-bot для поиска авиабилетов. Разрабатываемый бот несёт в себе информационный характер и позволяет, не выходя из популярного мессенджера, найти авиабилеты, необходимые пользователю. Пользователь может выбрать город отправления, дату отправления и при необходимости также запросить информацию об обратном билете. Бот предоставит информацию по всем рейсам выбранной даты и благодаря удобному интерфейсу, пользователь сможет получить информацию, выбрать для себя оптимальный рейс и купить билет, перейдя по ссылке. Основная эффективность Telegram-бота состоит в экономии времени и быстром доступе к информации, а также автоматизации устаревающих процессов.

Специфика использования ПЭВМ состоит в том, что в процессе диалога человека и машины пользователь воспринимает интеллектуальную машину как равноправного собеседника. Поэтому возникает много совершенно новых психологических и психофизиологических проблем, суть которых нужно учитывать при проектировании трудового процесса. Другой особенностью является значительная информационная нагрузка. Значительная нагрузка на центральную нервную и зрительную системы вызывает повышение нервно-эмоционального напряжения, и, как следствие, негативно влияет на сердечнососудистую систему. Важной стороной функционирования организма пользователя является влияние на него комплекса факторов трудовой среды, включающих действие электромагнитных волн разных частотных диапазонов, статического электричества, шума, микроклиматических факторов и др. Воздействие этого специфического комплекса может оказать на здоровье человека отрицательное влияние. При работах с использованием компьютеров возникает целый ряд эргономических проблем, решение которых может значительно снизить нагрузку.

В этом случае имеются в виду только вопросы конструирования рабочего места пользователя и не охватываются вопросы формирования рационально построенных символов на экране и других, изменение которых возможно только при конструировании новой техники. Работа пользователя ЭВМ чаще всего проходит при активном взаимодействии с другими людьми. Поэтому возникают вопросы межличностных взаимоотношений, включающие как психологические, так и социально-психологические аспекты. Таким образом, на пользователя ЭВМ воздействуют 4 группы факторов трудовой среды: физические, эргономические, информационные и социально-психологические.

5.2 Расчётная часть

Для определения тяжести труда пользователей информационной системы необходимо каждому фактору, который характеризует условия труда, соответствующий балл. Исходные данные представлены в таблице 5.1.

Таблица 5.1 – Оценка условий труда на рабочем месте

№	Факторы	Значение	Баллы
1	Температура воздуха на рабочем в теплый период года, t °C	22	2
2	Температура воздуха на рабочем месте в холодный период года, t °C	22	1
3	Рабочее место, поза и перемещение в пространстве	Рабочее место стационарное, поза свободная, масса перемещаемого груза до 5 кг	1
4	Относительная влажность воздуха, φ, %	53	1
5	Скорость движения воздуха, V, мс.	0,7	3
6	Освещенность, Е, лк.	250	2
7	Шум, L, дБ.	57	3
8	Число важных объектов наблюдения	2	1
9	Темп (число движений пальцев в час)	500	1
10	Длительность сосредоточенного наблюдения, % от рабочего времени	81	4
11	Нервно-эмоциональная нагрузка	Простые действия по индивидуальному плану	1

Расчет интегральной бальной оценки тяжести труда:

$$И_{т1} = 10 \times (X_{оп} + X_{ср} \times \frac{6 - X_{оп}}{6} \quad (5.1)$$

где $И_{т1}$ - интегральная бальная оценка тяжести труда;

$X_{оп}$ – элемент условий труда, который получил наибольшую оценку;

$X_{ср}$ – средний балл всех активных элементов условий труда кроме определяющего $X_{оп}$.

Средний балл всех активных элементов условий труда, кроме определяющего $X_{оп}$, рассчитывается по формуле (5.2):

$$X_{ср} = \frac{\sum_{i=1}^{n} X_i}{n - 1} \quad (5.2)$$

где $\sum_{i=1}^{n} X_i$ – сумма всех элементов кроме определяющего $X_{оп}$;

n – количество учтенных элементов условий труда.

В соответствии с имеющимися данными, $X_{оп} = 3$, а количество учетных элементов условий труда равняется 10.

Средний балл всех активных элементов условий труда:

Хср = (2 + 1 + 1 + 1 + 3 + 2 + 3 + 1 + 1 + 1) / (11 – 1) = 16 / 10 = 1,6 балла.

Далее производится интегрально-балльная оценка тяжести труда:

$И_т = 10 \times (4 + 1,6 \times ((6 – 4)/6) = 45,33$ балла.

Для определения категории тяжести труда необходимо воспользоваться таблицей 5.2.

Таблица 5.2 - Категории тяжести труда

Категория тяжести труда	I	II	III	IV	V	VI
Интегральная оценка элементов условий труда, $И_т$, баллы	до 18	18,1- 33	33,1- 45	45,1- 53	53,1- 59	59,1- 60

Интегральная оценка тяжести труда составляет 45,33 балла, что соответствует IV категории тяжести труда.

Повышение тяжести труда будет влиять на работоспособность человека. Снижение работоспособности непосредственно связано с состоянием утомления, которое количественно можно оценить при помощи показателя утомления, выраженного в условных единицах.

Интегральная бальная оценка тяжести труда позволяет определить влияние условий труда на работоспособность человека. Степень утомления определяется в условных единицах по формуле:

$$У = (И_т – 15,6)/0,64 \quad (5.3)$$

где У – степень утомления;

$И_т$ – интегральная бальная оценка тяжести труда;

15,6 и 0,64 – коэффициенты регрессии.

Таким образом, степень утомления составит:

$$У = (45,33 – 15,6) / 0,64 = 46,46$$

Зная степень утомления можно рассчитать уровень работоспособности. Уровень работоспособности является величиной противоположной утомлению и рассчитывается по формуле:

$$R = 100 – У \quad (5.4)$$

где R - работоспособность человека;

У - степень утомления.

Исходя из данной формулы, работоспособность составит:

$$R = 100 – 48,02 = 53,54$$

Тяжесть и напряженность труда оказывает влияние на рост производственного травматизма. Так как интегральная балльная оценка дает возможность определить категорию тяжести труда, то величину производственного травматизма можно рассчитать по формуле:

$$К = 1 / (1,3 – 0,0185 \times И_т) \quad (5.5)$$

Таким образом, производственный травматизм составит:

$$К = 1 / (1,3 – 0,0185 \times 45,33) = 2,17$$

5.3 Заключение по разделу безопасность жизнедеятельности

Исходя из полученных результатов, можно сделать следующие выводы:

- Коэффициент работоспособности (53,54) превышает коэффициент степени утомления (46,46);
- Коэффициент производственного травматизма составил 2,17.

В заключении можно отметить, что рабочее место удовлетворяет нормам условий труда и внесения каких-либо изменений не требует.

Заключение

В нынешнее время мессенджеры стали неотъемлемой частью повседневной жизни людей, просыпаясь и засыпая, люди проверяют уведомления в мессенджерах и социальных сетях. Боты в мессенджерах позволяют упрощать многие рутинные задачи, тем самым, облегчая жизнь людей. В мессенджере Telegram, боты активно используются всеми пользователями, ускоряя обычные процессы. Telegram-боты стали прорывом в своей области как удобная платформа для реализации различных программных средств под цели пользователя. Таким образом Telegram-боты являются наиболее выгодными и популярными средствами для продвижения бизнеса.

В данном дипломном проекте был разработан бот для поиска авиабилетов на основе мессенджера Telegram. Также были выполнены задачи, такие как: сравнение и анализ функций мессенджеров, выявление лучшего мессенджера, выбор языка и среды разработки бота.

В аналитическом разделе дипломного проекта были рассмотрены ключевые моменты обзора приложений, сравнение доступности резервных копий, безопасность, конфиденциальность, сравнение функций и сквозное шифрование.

В проектном разделе дипломного проекта были рассмотрены ключевые моменты создания и проектирования бота, наиболее подробно рассмотрена информация о мессенджере Telegram с точки зрения проектирования.

В экспериментальном разделе углубленно рассмотрено проектирование и также показана программная реализация нескольких примеров по поиску авиабилетов.

В экономической части дипломного проекта были рассмотрены вопросы автоматизации процесса, с точки зрения экономики. По итогам данной главы ожидаемый годовой экономический эффект составил 1 693 944 тенге. Telegram бот окупится за 8 месяцев использования.

В разделе безопасности жизнедеятельности данного дипломного проекта была выполнена интегральная бальная оценка тяжести труда для определения показателей утомляемости и работоспособности для конкретного рабочего места.

Список литературы

1 Телеграм-бот как простой и удобный способ получения информации. Козлов А.А., Батищев А.В. Территория науки. 2017.

2 Технологии создания и применения чат-ботов. Матвеева Н.Ю., Золотарюк А.В. Научные записки молодых исследователей. 2018.

3 Использование мессенджера Telegram в качестве медиа-платформы для распространения информации СМИ и блогерами. Максименко В.В. Москва, 2019. С. 187-198.

4 Особенности медиатекста в мессенджере Telegram. Сулаквелидзе Н.М. Самара, 2018. С. 241-247.

5 К вопросу выбора безопасного мессенджера. Гатиятуллин Т.Р. 2016.

6 Мессенджеры как цифровой бизнес-инструмент. Стефанова Н.А., Шматок К.О. 2018.

7 Описание концепции Telegram ботов и их разработка. Бийбосунов Б.И., Бийбосунова С.К., Жолочубеков Н.Ж. Colloquium-journal. 2020.

8 Технико-экономическое обоснование дипломных проектов. Брест, БГТу, 2014 – 15с.

9 Информационный менеджмент. Симионов Ю.Ф., Боромотов В.В. — Ростов н.Д: Феникс, 2013, 250с.

10 Методические указания по выполнению экономической части дипломных работ для студентов специальностей 5В070400 – «Вычислительная техника и программное обеспечение», 5В060200 – «Информатика», 5В070300 – «Информационные системы». Боканова Г.Ш. – Алматы, АУЭС, 2020. - 35 с.

11 Экономическая эффективность мероприятий в области безопасности жизнедеятельности. Машкин А.Л., Гоголина Е.С., Казицкая Н.В., Дрейцен М.А. Материалы Всероссийской научно-практической конференции «Наука и социум». 2019.

12 Оптимизация защитных мероприятий по безопасности жизнедеятельности людей. Фесечко А.И. Труды Международного симпозиума «Надежность и качество». 2011.

13 Безопасность жизнедеятельности. Учебное пособие для всех специальностей. Санатова Т.С. 2017. - 407 с.

14 Telegram Bot API https://core.telegram.org/bots/api

15 Программная платформа Node. js. Долгов А.Н., Нуруллин Р.Ю. Достижения науки и образования. 2016.

16 JavaScript. Подробное руководство, 6-ое издание. Флэнаган Д. 2012

17 Node.js Разработка серверных веб-приложений на JavaScript. Хэррон Д. 2012.

18 https://telegram.org/

Приложение А

Листинг программы

Код файла Bot.js:

```
process.env.BOT_TOKEN = "1140150254:AAEOQHxmKQ9aeEHHn4Wd-aoatmmPWK8upF4"

const Telegraf = require('telegraf')
const session = require('telegraf/session')
const Stage = require('telegraf/stage')
const Scene = require('telegraf/scenes/base')
const SearchScene = require('./searchScene')
const Extra = require('telegraf/extra')

const searchSceneId = 'searchScene'
const searchScene = new SearchScene(searchSceneId)

const helperSceneId = 'helper'
const helperScene = new Scene(helperSceneId)
helperScene.command('start', (ctx) => {
	const msg = `
Добро пожаловать в Travel Bot.
Чтобы начать новый поиск введите команду - /search
Что может делать бот - /description
Подробнее о проекте - /about
Хорошего поиска ☀
	`
	ctx.reply(msg)
})
helperScene.command('search', (ctx) => {
	ctx.scene.leave()
	ctx.scene.enter(searchSceneId)
})
helperScene.command('about', (ctx) => {
	const url = "https://cs8.pikabu.ru/post_img/big/2016/02/28/8/1456667944151428298.jpg"
	ctx.replyWithPhoto(url, Extra.caption('Дипломный проект Алины Бухметовой Инф-16-2').HTML())
})
helperScene.use((ctx) => {
	const msg = `
```

```
            Travel Bot поможет Вам найти авиабилеты в любую точку мира KZ
CA AM GB IR IL LT
            Хорошего поиска:)`
            ctx.reply(msg)
        })

        const stage = new Stage([helperScene, searchScene], { default: helperSceneId
})
```

```
const bot = new Telegraf(process.env.BOT_TOKEN)
bot.use(session())
bot.use(stage.middleware())

bot.launch()
```

Код файла SearchScene.js:

```
const Composer = require('telegraf/composer')
const ApiService = require('./api')
const Markup = require('telegraf/markup')
const { parseOptions } = require('./parser')
const { formatRouteOption, getFormmatedTicketUrl } = require('./formatter')
const { compose, unwrap } = Composer

const api = new ApiService()

const Step = {
  START: { message: "" },
  IS_ROUNDTRIP: { message: "Вам нужен обратный билет?" },
  DEPARTURE_POINT: { message: "Укажите город вылета 🛫" },
  ARRIVAL_POINT: { message: "Укажите город прибытия 🛬" },
  DEPARTURE_DATE: { message: "Укажите желаемую дату вылета (в
формате YYYY-MM-DD)" },
  RETURN_DATE: { message: "Укажите желаемую дату обратного
билета (в формате YYYY-MM-DD)" },
  RESULTS: { message: "Ищу самые выгодные предложения! 🔎🚀" },
}

class SearchScene extends Composer {

  constructor (id) {
    super()
    this.options = {}
    this.id = id
    this.currentStep = Step.START
  }

  set ttl (value) {
    this.options.ttl = value
  }
```

```
get ttl () {
  return this.options.ttl
}

middleware () {
  return compose([
```

```
      (ctx, next) => {
      if (ctx.scene.state.currentStep == undefined) {
         ctx.scene.state.currentStep = Step.START
      }
      console.log(ctx.message.text)
      return unwrap(this.stepHandler(ctx))(ctx, next)
    }
  ])
 }

 stepHandler (ctx) {
  const currentStep = ctx.scene.state.currentStep

  if (currentStep === Step.START) {
    return greet
  } else if (currentStep === Step.IS_ROUNDTRIP) {
    return askIfRoundtrip
  } else if (currentStep === Step.DEPARTURE_POINT) {
     return provideDeparturePoint
  } else if (currentStep === Step.ARRIVAL_POINT) {
     return provideArrivalPoint
  } else if (currentStep === Step.DEPARTURE_DATE) {
     return provideDepartureDate
  } else if (currentStep === Step.RETURN_DATE) {
     return provideReturnDate
  } else if (currentStep === Step.RESULTS) {
    return findTickets
  }
 }
}

 function nextStep (ctx) {
  const currentStep = ctx.scene.state.currentStep

  if (currentStep === Step.START) {
     ctx.scene.state.currentStep = Step.IS_ROUNDTRIP

  } else if (currentStep === Step.IS_ROUNDTRIP) {
    ctx.scene.state.currentStep = Step.DEPARTURE_POINT

  } else if (currentStep === Step.DEPARTURE_POINT) {
     ctx.scene.state.currentStep = Step.ARRIVAL_POINT
```

```
} else if (currentStep === Step.ARRIVAL_POINT) {
    ctx.scene.state.currentStep = Step.DEPARTURE_DATE

} else if (currentStep === Step.DEPARTURE_DATE) {

    if (ctx.scene.state.isRoundtrip) {
```

```
        ctx.scene.state.currentStep = Step.RETURN_DATE
        } else {
          ctx.scene.state.currentStep = Step.RESULTS
        }

    } else if (currentStep === Step.RETURN_DATE) {
       ctx.scene.state.currentStep = Step.RESULTS

    }

    if (ctx.scene.state.currentStep === Step.RESULTS) {
        findTickets(ctx)
    }

  }

function nextStepWithMsg(ctx) {
  nextStep(ctx)
  ctx.reply(ctx.scene.state.currentStep.message)
}

function greet (ctx) {
  nextStep(ctx)
  const msg = ctx.scene.state.currentStep.message
  const options = ["Да", "Нет"]
  replyWithKeyboard(ctx, msg, options)
}

function askIfRoundtrip (ctx) {
  const query = extractLowercasedQuery(ctx)

  if (query == "да") {
     ctx.scene.state.isRoundtrip = true
  } else if (query == "нет") {
     ctx.scene.state.isRoundtrip = false
  } else {
     const msg = ctx.scene.state.currentStep.message
     ctx.replyWithHTML(`${msg}\n(да / нет)`)
     return
  }

  nextStepWithMsg(ctx)
```

```
}

function provideDeparturePoint (ctx) {
  updatePoint(ctx, (value) => {
    ctx.scene.state.departurePoint = value.code
    ctx.scene.state.departurePointTitle = value.title
  })
```

```
}

function provideArrivalPoint (ctx) {
  updatePoint(ctx, (value) => {
    ctx.scene.state.arrivalPoint = value.code
    ctx.scene.state.arrivalPointTitle = value.title
  })
}

async function updatePoint (ctx, updateState) {
  const query = extractLowercasedQuery(ctx)

  const cities = await api.getCities(query)

  const mainPoint = cities.filter((city) => {
     return city.title.toLowerCase() == query
  }).sort((a,b) => {
     return (a.priority > b.priority) ? -1 : 1
  })[0]

  if (cities.length == 0) {
     ctx.reply("Такого города не найдено. Попробуйте поискать другой
🙃")
     return
  } else if (mainPoint !== undefined) {
     updateState(mainPoint)
     nextStep(ctx)
     const newMsg = ctx.scene.state.currentStep.message
     const confirmationMsg = `Отлично! Вы выбрали "${mainPoint.title}"
✅`
     const fullMsg = confirmationMsg + "\n" + newMsg
     ctx.reply(fullMsg, Markup.removeKeyboard().extra())
     return
  }

  const cityNames = cities.map( x => x.title )

  const cityListMsg = `Хм. Не удалось определить введенный Вами город.
Возможно он есть в списке? 👇`
  replyWithKeyboard(ctx, cityListMsg, cityNames)
}
```

```
function replyWithKeyboard(ctx, msg, options) {
    ctx.reply(msg, Markup
      .keyboard(options)
      .oneTime()
      .resize()
      .extra()
  )
}
```

```
function provideDepartureDate (ctx) {
  updateDate(ctx, "вылета", (query) => {
   ctx.scene.state.departureDate = query
  })
}

function provideReturnDate (ctx) {
  updateDate(ctx, "желаемого обратного билета", (query) => {
   ctx.scene.state.returnDate = query
  })
}

function updateDate (ctx, dateTypeStr, updateState) {
  const query = extractLowercasedQuery(ctx)

  if (isValidDate(query)) {
     updateState(query)
     nextStepWithMsg(ctx)
     return
  }

  const msg = `Не удалось распознать дату ${dateTypeStr}. Введите её в
формате YYYY-MM-DD`
  ctx.reply(msg)
}

async function findTickets (ctx) {
  const tickets = await api.getTickets(ctx.scene.state,
ctx.scene.state.isRoundtrip)

  const from = ctx.scene.state.departurePointTitle
  const to = ctx.scene.state.arrivalPointTitle

  if (tickets === undefined || tickets.cards.length == 0) {
     ctx.reply("К сожалению, по Вашему запросу предложений не
найдено ☹")
  } else {
     const options = parseOptions(tickets.cards, from, to)
     options.forEach(opt => {
       const ticketUrl = getFormmatedTicketUrl(ctx.scene.state, opt.uniqueId)
```

```
            const button =  Markup.inlineKeyboard([Markup.urlButton("Купить на
Santufei.kz 🚀", ticketUrl)]).extra()
            ctx.replyWithHTML(formatRouteOption(opt), button)
          })
        }

        ctx.scene.leave()
      }

      function extractLowercasedQuery (ctx) {
        return ctx.message.text.toLowerCase()
```

```
}

function isValidDate (str) {
  return /^([12]\d{3}-(0[1-9]|1[0-2])-(0[1-9]|[12]\d|3[01]))$/.test(str)
}

module.exports = SearchScene
```

Код файла Api.js:

```
const fetch = require("node-fetch");

class ApiService {

        async getRequest (url) {
                console.log(url)
           return fetch(url)
                        .then(function(response) {
                           if (!response.ok) {
                             return reject()
                           }
                           return response.json()
                        }).catch(function(error) {
                             console.log(error);
                        })
        }

        getCities (query) {
                const params = this.encodeQueryData({q: query})
                const url =
`https://santufei.com/api/v1/tickets/new_locations/?${params}`
                return this.getRequest(url)
        }

        getTickets (params, isRoundtrip) {
                const oneWay = {
                        segment:
`${params.departurePoint}_${params.arrivalPoint}_${params.departureDate}`
                }

                const oneQuery = this.encodeQueryData(oneWay)
```

```
                let url =
`https://santufei.com/api/v1/tickets/search/?${oneQuery}`

                if (isRoundtrip) {
                        const returnWay = {
                                segment:
`${params.arrivalPoint}_${params.departurePoint}_${params.returnDate}`
                        }
```

```
                const returnQuery = this.encodeQueryData(returnWay)
                url = url + `&${returnQuery}`
            }

            return this.getRequest(encodeURI(url))
        }

        encodeQueryData (data) {
          const ret = [];
          for (let d in data)
            ret.push(encodeURIComponent(d) + '=' +
encodeURIComponent(data[d]));
          return ret.join('&');
        }
    }

    module.exports = ApiService
```

Код файла Parser.js:

```
    class RouteParser {

        static parseOptions (cards, from, to) {
            return cards.map(card => {
                let route =
RouteParser.extractRouteFromIterinary(card.itineraries[0])
                route.from = from
                route.to = to
                return route
            })
        }

        static extractRouteFromIterinary (iterinary) {
            let option = iterinary.options[0]

            let flights = option.flights.map((flight) => {
                let flightLegs = RouteParser.extractLegsFromFlight(flight)
                return flightLegs
            })

            function numberWithSpaces(x) {
            return x.toString().replace(/\B(?=(\d{3})+(?!\d))/g, ",")
```

```
                }

                let route = {
                        price: `${numberWithSpaces(iterinary.price)}
${iterinary.currency}`,
                        departure: option.departure,
                        stopsCount: option.stops_count,
                        flights: flights,
```

```
                uniqueId: option.unique_id
            }

            return route
        }

        static extractLegsFromFlight (flight) {
            let legs = flight.legs.map(leg => {
                return {
                    airline: leg.operating_airline.str,
                    origin: `${leg.origin.code},
${leg.origin.city.name}`,
                    destination: `${leg.destination.code},
${leg.destination.city.name}`,
                    departureTime:
`${leg.departure.human_readable.date}, ${leg.departure.human_readable.time}
(UTC +${leg.origin.utc_offset_hours})`,
                    arrivalTime: `${leg.arrival.human_readable.date},
${leg.arrival.human_readable.time} (UTC +${leg.destination.utc_offset_hours})`,
                    duration: leg.time_readable
                }
            })

            return legs
        }
    }

    module.exports = RouteParser
```

Код файла Formatter.js:

```
    class RouteOptionFormatter {

        static getFormmatedTicketUrl(params, routeId) {
            let url = "https://santufei.com/search/"

            url +=
`${params.departurePoint}_${params.arrivalPoint}_${params.departureDate}`

            if (params.isRoundtrip) {
                url +=
`/${params.arrivalPoint}_${params.departurePoint}_${params.returnDate}`
```

```
        }

        const ticketType = params.isRoundtrip ? 'RT' : 'OW'

        url += `/1/0/0/ECNM/${ticketType}#${routeId}`

        return url
}
```

```
        static formatRouteOption (route) {
            let str =
RouteOptionFormatter.extractStringFromFlight(route.flights[0], route.from,
route.to)

            if (route.flights[1] !== undefined) {
                str += "\r_____\n"
                str +=
RouteOptionFormatter.extractStringFromFlight(route.flights[1], route.to,
route.from)
            }

            str += `\n<b>Цена: ${route.price}</b>`
            return str
        }

        static extractStringFromFlight (flight, from, to) {
            let legs = RouteOptionFormatter.extractLegsFromFlight(flight)
            let str =`
    <b>${from} - ${to}</b>

    ${legs}
    `
            return str
        }

        static extractLegsFromFlight (flight) {
            let legsStr = ""

            for (var i = 0; i < flight.length; i++) {
                const leg = flight[i]

                legsStr += `<code>`
                legsStr += `✈: <b>${leg.origin} -
${leg.destination}</b>\n`
                legsStr += `🛫✈: <b>${leg.airline}</b>\n`
                legsStr += `🛫: ${leg.departureTime}\n`
                legsStr += `🕒: ${leg.duration}\n`
                legsStr += `🛬: ${leg.arrivalTime}`
                legsStr += `</code>`
```

```
                if (flight.length > 1 && i < flight.length - 1) {
                    legsStr += "\n↓↓\n"
                }
            }

            return legsStr
        }
}

module.exports = RouteOptionFormatter
```

Приложение Б

Акт внедрения

Утверждаю
Директор ТОО «Аэлита-2014»
Сарсембаев Р. Р.
«4» мая 2020г.

АКТ ВНЕДРЕНИЯ

Настоящий акт составлен о том, что результаты выпускной работы студентки НАО АУЭС группы Инф-16-2 очной формы обучения Бухметовой Алины Амангельдиевны на тему «*Разработка Telegram бота для поиска авиабилетов*» внедрены в ТОО «Аэлита-2014» и используются в компании в качестве программного обеспечения для обслуживания клиентов. Результат выпускной работы Бухметовой А. А. обеспечивает быстрый доступ к получению информации о доступных авиабилетах.

Показатели эффективности программного обеспечения:
- Значительное сокращение времени на поиск;
- Оптимизация ввода данных.

Директор ТОО «Аэлита-2014»
Сарсембаев Р. Р. «подпись, печать»

Исполнитель
Бухметова А. А.