|  |
| --- |
|  |
| МИНОБРНАУКИ РОССИИ |
| Федеральное государственное бюджетное образовательное учреждение высшего образования  **«МИРЭА – Российский технологический университет»**  **РТУ МИРЭА** |

**Институт информационных технологий (ИТ)**

**Кафедра инструментального и прикладного программного обеспечения (ИиППО)**

**КУРСОВАЯ РАБОТА**

по дисциплине: Тестирование и отладка программного обеспечения

по профилю: Разработка программных продуктов и проектирование информационных си стем

направления профессиональной подготовки: 09.03.04 «Программная инженерия»

Тема: Тестирование программы “Бот-помощник для мессенджера Telegram”

Студент:Петров Анатолий Валерьевич Группа: ИКБО-03-18

Работа представлена к защите 17.12.2021 (дата)

/Петров А.В./ (подпись и ф.и.о. студента)

Руководитель: к.т.н., доцент Алпатов Алексей Николаевич

Работа допущена к защите (дата)

/ Алпатов А.Н./ (подпись и ф.и.о. рук-ля)

Оценка по итогам защиты:

/ , к.т.н., доцент Алпатов А.Н./

/

(подписи, дата, ф.и.о., должность, звание, уч. степень двух преподавателей, принявших защиту)

2021г.



МИНОБРНАУКИ РОССИИ

Федеральное государственное бюджетное образовательное учреждение высшего образования

**«МИРЭА - Российский технологический университет» РТУ МИРЭА**

**Институт информационных технологий (ИТ)**

**Кафедра инструментального и прикладного программного обеспечения (ИиППО)**

## ЗАДАНИЕ

**на выполнение курсовой работы**

по дисциплине: Тестирование и отладка программного обеспечения

по профилю: Разработка программных продуктов и проектирование информационных систем

Студент: Петров Анатолий Валерьевич

Группа: ИКБО-03-18

Срок представления к защите: 17.12.2021 Руководитель: доцент Алпатов Алексей Николаевич

**Тема:** Тестирование программы “Бот-помощник для выбора обуви для мессенджера Telegram”

**Исходные данные**: Индивидуальное задание на курсовую работу. ГОСТ Р 56920- 2016/ISO/IEC/IEEE 29119-3:2013, Test Plan Template RUP, STD-EASS008 (по выбору). Программное обеспечение для поддержки тестирования, методика тестирования с учетом особенностей реализуемого ПО.

**Перечень вопросов, подлежащих разработке, и обязательного графического материала**: 1. Реализовать тестируемый модуль. 2. Выбрать методологию тестирования; 3. Разработать тестовый план, с учетом особенностей разрабатываемого ПО; 4. Разработать тест-кейс с соответствующими атрибутами; 5. Провести тестирование и анализ полученных результатов.

Руководителем произведён инструктаж по технике безопасности, противопожарной технике и правилам внутреннего распорядка.

Зав. кафедрой ИиППО: / Болбаков Р.Г. /, « » 2021 г. Задание на КР выдал: / Алпатов А.Н. /, « » 2021 г. Задание на КР получил: / Петров А.В. /, « » 2021 г.

УДК 004.942

Петров А.В. Тестирование программы “Бот-помощник для мессенджера Telegram” **/ Курсовая работа** по дисциплине «Тестирование и отладка программного обеспечения» профиля «Разработка программных продуктов и проектирование информационных систем» направления профессиональной подготовки бакалавриата 09.03.04 «Программная инженерия» (7ой семестр) / руководитель к.т.н., доцент А.Н. Алпатов / кафедра ИиППО Института ИТ РТУ МИРЭА – с. 42, илл. 29, ист. 7 (в т.ч. 1 на англ. яз.).

Целью работы является закрепление практических навыков применения на практике в проектировании и разработке программного обеспечения подходов и методов тестирования программного обеспечения.

В рамках работы проводится тестирование чат-бота, которое включает в себя формирование спецификации требований к приложению, тестового плана и тестовых кейсов.

Petrov Anatoly. Testing of the program "Bot-assistant for messenger Telegram". / Term work on Software testing and debugging profiled in Software development and planning of information systems. Baccalaureate major in software development — 09.03.04 (7th semester). Advisor — A. Alpatov, PhD in technical sciences. Department of Instrumental and Applied Software Engineer- ing, Institute of information technologies, RTU — MIREA.

The purpose of the work is to consolidate practical skills of applying soft- ware testing approaches and methods in practice in the design and development of software.

As part of the work, the chat bot is being tested, which includes the formation of a specification of application requirements, a test plan and test cases.

М. МИРЭА. Ин-т ИТ. Каф. ИиППО. 2021 г. @ А. Петров

# СОДЕРЖАНИЕ

[СПИСОК ИСПОЛЬЗОВАННЫХ СОКРАЩЕНИЙ 6](#_bookmark0)

[ВВЕДЕНИЕ 7](#_bookmark1)

1. [РЕАЛИЗАЦИЯ ТЕСТИРУЕМОГО МОДУЛЯ 9](#_bookmark2)
2. [АНАЛИЗ ТЕСТИРУЕМОГО ПРОГРАММНОГО ПРОДУКТА 10](#_bookmark3)
   1. [Описание предметной области 10](#_bookmark4)
   2. [Анализ существующих чат-ботов 10](#_bookmark5)
      1. [Чат-бот TrackBot 11](#_bookmark6)
      2. Чат-бот [Kinomanbot 12](#_bookmark7)
      3. [Чат-бот InMind 13](#_bookmark8)
   3. [Вывод 14](#_bookmark9)
3. [ТЕСТИРОВАНИЕ ТРЕБОВАНИЙ ЗАКАЗЧИКА ПРОГРАММНОГО](#_bookmark10) [ПРОДУКТА 15](#_bookmark10)
   1. [Тестируемые требования 15](#_bookmark11)
   2. [Анализ техник тестирования требований 17](#_bookmark12)
      1. [Метод просмотра 17](#_bookmark13)
      2. [Формулировка вопросов 18](#_bookmark14)
      3. [Формирование чек-листов или тест-кейсов проверяемости 18](#_bookmark15)
      4. [Графическое представление 18](#_bookmark16)
      5. [Исследование поведения 19](#_bookmark17)
      6. [Прототипирование 19](#_bookmark18)
      7. [Применение методологий 19](#_bookmark19)
   3. [Подготовка списка вопросов 19](#_bookmark20)
   4. [Вывод 20](#_bookmark21)
4. [ФОРМИРОВАНИЕ ТЕСТОВОГО ПЛАНА 21](#_bookmark22)
   1. [Анализ существующих подходов и международных стандартов,](#_bookmark23) [применяемых для написания тестового плана 21](#_bookmark23)
      1. [Мастер-план тестирования 21](#_bookmark24)
      2. [Детальный тестовый план 22](#_bookmark25)
      3. [Планы тестирования для конкретных типов тестирования 22](#_bookmark26)
      4. [Вывод 22](#_bookmark27)
   2. [Создание тестового плана 22](#_bookmark28)
      1. [Определение цели тестирования 22](#_bookmark29)
      2. [Области, подвергаемые тестированию 23](#_bookmark30)
      3. [Области, не подвергаемые тестированию 23](#_bookmark31)
      4. [Разработка тестовой стратегии 25](#_bookmark32)
         1. [Методы тестирования 24](#_bookmark33)
         2. [Определения серьезности ошибок 25](#_bookmark34)
      5. [Определение критериев тестирования 26](#_bookmark35)
      6. [Планирование ресурсов 26](#_bookmark36)
      7. [Расписание времени выполнения тестирования 27](#_bookmark37)
      8. [Оценка рисков 28](#_bookmark38)
      9. [Тестовая документация 28](#_bookmark39)
      10. [Метрики тестирования 29](#_bookmark40)
      11. [Покрытие требований 29](#_bookmark41)
   3. [Вывод 29](#_bookmark42)
5. [ФОРМИРОВАНИЕ ТЕСТОВЫХ КЕЙСОВ 30](#_bookmark43)
   1. [Разработка чек-листа 30](#_bookmark44)
   2. [Тест-кейсы 30](#_bookmark45)
   3. [Выводы 33](#_bookmark46)
6. [ТЕСТИРОВАНИЕ ПРОГРАММНОГО МОДУЛЯ 34](#_bookmark47)
   1. [Выполнение тест-кейсов 34](#_bookmark48)

[ЗАКЛЮЧЕНИЕ 41](#_bookmark49)

[СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ 42](#_bookmark50)

# СПИСОК ИСПОЛЬЗОВАННЫХ СОКРАЩЕНИЙ

* ОС — операционная система;
* ПО — программное обеспечение;
* ЯП — язык программирования;
* IDE — (англ. Integrated development environment) Интегрированная среда разработки, ИСP.

# ВВЕДЕНИЕ

Тестирование программного обеспечения — процесс исследования, испытания программного продукта, имеющий своей целью проверку соответствия между реальным поведением программы и её ожидаемым поведением на конечном наборе тестов, выбранных определённым образом [1].

Тема курсовой работы — Тестирование программы “Бот-помощник для мессенджера Telegram”.

Актуальность темы связана с повышенным интересом к относительно новой технологии. Действительно, очень удобно иметь автоматизированную машину не просто отдельным приложением, а лишь одним чатом в мессенджере. А еще, зачастую, чат-бот не требует авторизации в отличии почти от любой услуги в интернете.

Тестирование приложений, разработанных для мессенджера, позволяет определить некорректно работающие модули программ, оценить удобство пользования пользовательским интерфейсом, разработать требования для формирования списков недочетов, связанных с программным продуктом.

Целью курсовой работы является тестирование чат-бота помощника Telegram, который показывает случайные картинки природы и окружения из коллекции.

Поставленная цель определяет следующие задачи:

* + реализация тестируемого модуля;
  + анализ тестируемого программного продукта;
  + тестирование требований заказчика программного продукта;
  + формирование тестового плана;
  + формирование тестовых кейсов;
  + тестирование программного модуля.

Предметом курсовой работы являются подходы к тестированию программного обеспечения.

Объектом курсовой работы являются разработанные тестируемые модули, требования заказчика программного продукта, тестовый план, тестовые кейсы.

# РЕАЛИЗАЦИЯ ТЕСТИРУЕМОГО МОДУЛЯ

Подавляющую часть рынка мобильных операционных систем в России занимают два семейства ОС — Android и iOS. Реализация тестируемого продукта, доступного большинству потребителей в программе Telegram, уже заключает в себя требования к доступности программы.

Система представляет собой чат-бота, работающего в реальном времени и зависящее от управления посредством касаний по экрану пользователя.

Для пользователя чат-бот будет выглядеть как несколько кнопок из развернутого готового окошка и строка ввода. После начала работы на главном окне пользователя встречает приветствие и приглашение выбрать кнопку для дальнейшего взаимодействия с ботом.

Скриншот стартового окна тестируемого модуля представлен на рисунке 1.1.



Рисунок 1.1 — Стартовое окно программы

# АНАЛИЗ ТЕСТИРУЕМОГО ПРОГРАММНОГО ПРОДУКТА

## Описание предметной области

Предметной областью курсовой работы является тестирование программного модуля. Разрабатываемый программный продукт относится к классу программного обеспечения, которое должно быть предназначено для работы с приложением Telegram.

Рассмотрим уже существующие чат-боты.

## Анализ существующих чат-ботов

Боты – это программы, которые выполняют различные задачи для пользователя, находящегося в мессенджере. Бот выглядит как обычный чат, однако общение происходит не с человеком, а с программой, которая может принять заказ на вызов машины, если это – бот такси, или прислать свежие статьи, если это – новостной бот, или заказать доставку еды на дом, если это – ресторанный бот. Не так давно чат-боты обрели большую популярность, перевоплотившись из развлечения в более серьезную вещь, так как они, в основном, стали использоваться для решения серьезных бизнес-задач. В эпоху информационных технологий – это нормально явление, а тем более – сети Интернет, ведь общество давно перешло на «новое» и деловое, и неделовое общение. Во-первых, чат-боты – это «платформы» для решения бизнес-задач. Во-вторых, чат-бот – это приложение, которое поддерживает диалог с пользователем, выбирая ответы из базы данных: вы спрашиваете, где пообедать и тут же получаете мгновенный ответ. Кроме того, чат-боты выполняют множество полезных функций по исполнению рутинных операций, поиску информации, объединению данных, работе с клиентурой.

Чат-бот как виртуальный собеседник имеет базу знаний, которая представляет собой наборы возможных вопросов пользователя и соответствующих им ответов. Наиболее распространенными вариантами для получения нужного ответа являются ключевые слова, совпадение фразы, совпадение контекста. Всегда существуют какие-то простые и легкие по выполнению дела, на которые не хочется тратить время. Тут на помощь всегда могут прийти чат-боты. В наше время для сбора информации можно использовать чат-ботов. Конечно, это делается при помощи общения с людьми, но отнюдь не всегда. Например, в рамках какого-либо мероприятия чат-боты могут сообщать всем участникам новости и предоставлять справочную информацию.

Для более подробного анализа выберем три чат-бота для примера: TrackBot, Kinomanbot и Excel Bot.

## Чат-бот TrackBot

Track Bot — бот для отслеживания почтовых отправлений. Этот бот относится к виду Информатор. Нужно всего лишь сообщить ему трек-код или загрузить штрих-код посылки, и бот будет автоматически оповещать вас о каждом изменении статуса отправления и его местонахождении, рисунок 2.1. Бот следит за всеми мировыми почтовыми и курьерскими службами [2].

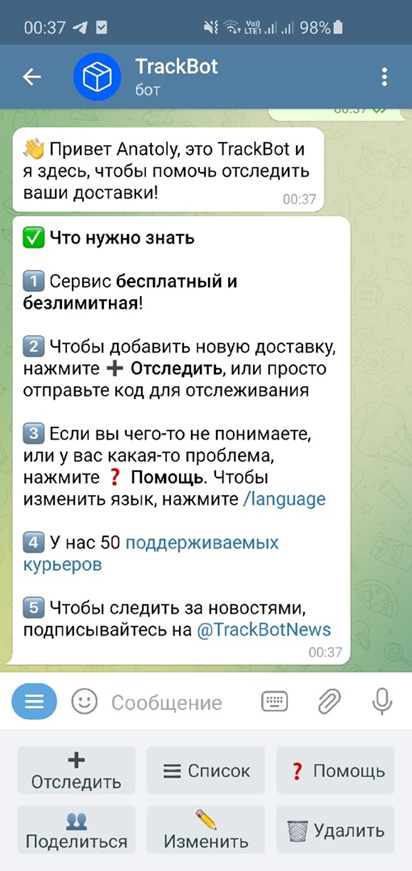


Рисунок 2.1 – Пример работы бота TrackBot

## Чат-бот Kinomanbot

А с помощью бота Kinomanbot удобно выбирать фильм или сериал для онлайн-просмотра. Также бот предлагает афишу кинотеатров с возможностью сразу купить билеты, новости кино и многое другое, рисунок 2.2. Этот бот относится одновременно и к разновидности Информатора, и Продавцу.

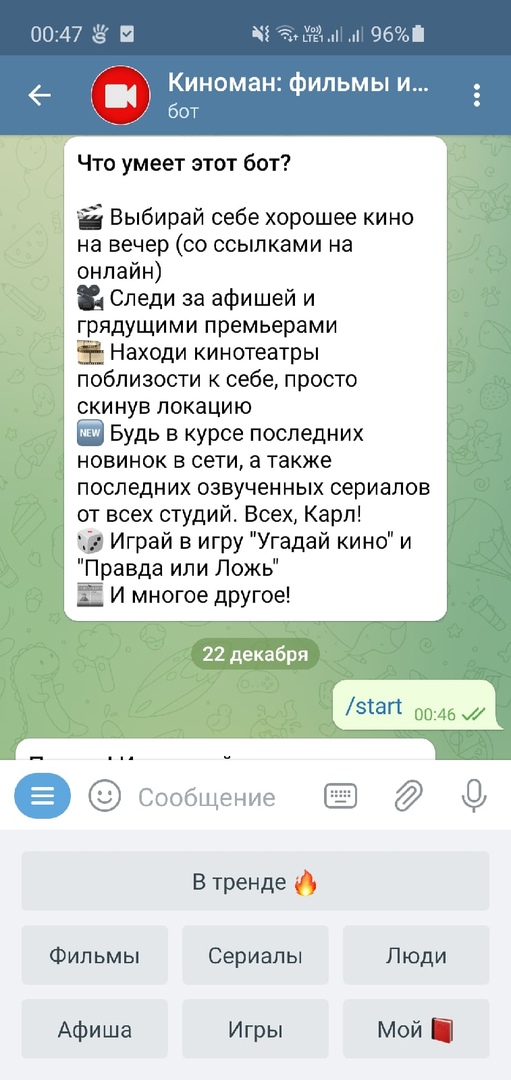


Рисунок 2.2 – Пример работы бота Kinomanbot

## Чат-бот InMind

Чат-бот в Telegram, который помогает учить новые слова на английском, можно настроить уровень сложности, рисунок 2.3. Бот относится к виду Информатора и Интерфейса.

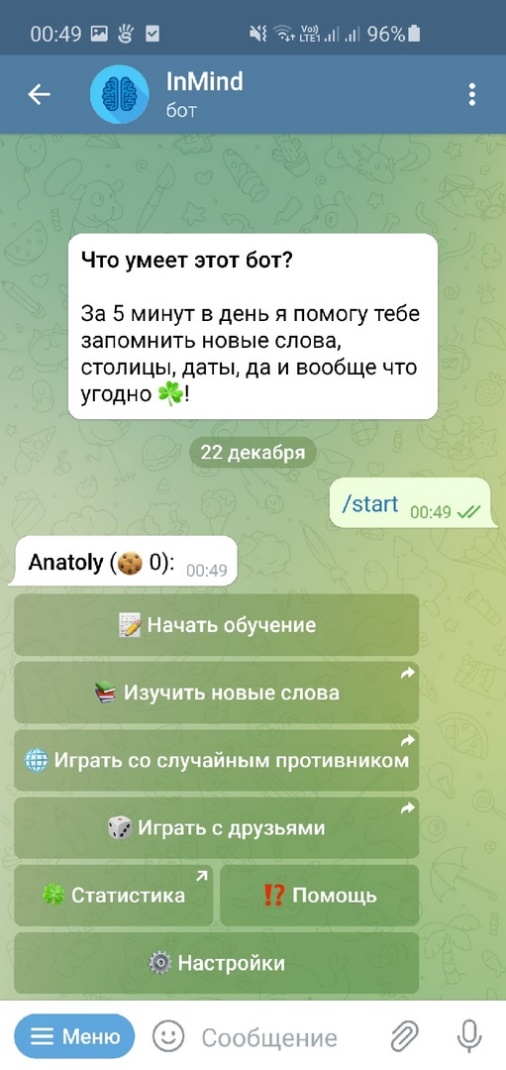


Рисунок 2.3 – Пример работы бота InMind

## Вывод

Из проведенного анализа программ, можно сделать следующие выводы: TrackBot — довольно простой, но полезный бот, Kinomanbot — многофункциональный бот, информирующий о фильмах, сериалах и актерах, но и с возможностью посмотреть афишу кинотеатров и заказать билеты в кино, InMind — бот, помогающий учить слова на английском. Среди представленных чат-ботов нет такого, который подошел бы всем, каждому необходим свой функционал и услуги, которые предоставляет конкретное программное обеспечение. Можно сделать вывод, что все эти боты создан для разных, не пересекающихся задач и благодаря этому востребован каждый из них.

# ТЕСТИРОВАНИЕ ТРЕБОВАНИЙ ЗАКАЗЧИКА ПРОГРАММНОГО ПРОДУКТА

## Тестируемые требования

Охарактеризуем основные недостатки программ, рассмотренных в разделе 2, и составим план тестируемых требований к разрабатываемому приложению.

В некоторых приложениях отсутствует возможность вызвать помощь, а в некоторых нет настроек.

У некоторых систем нет описания функционала в начале работы бота-помощника.

Невозможность добавить запись в избранное, следовательно, возникновение затруднений по их различению между собой.

В некоторых программах отсутствует возможность поделится результатом.

В таблице 3.1 представлены требования к функционалу чат-бота для Telegram.

*Таблица 3.1 — Матрица прослеживания требований*

|  |  |  |  |
| --- | --- | --- | --- |
| **№** | **Задача на разработку** | **Требование** | **Тест-кейс** |
| **1** | Разработка базы данных | Пользователь должен иметь возможность самостоятельно добавлять запись в базу данных | С1 |
| **2** | Разработка базы данных | Пользователь должен иметь возможность изменить запись из базы данных | С2 |

|  |  |  |  |
| --- | --- | --- | --- |
| **3** | Разработка базы данных | Пользователь должен иметь возможность просматривать записи, содержащиеся в базе данных | С3 |
| **4** | Разработка базы данных | Пользователь должен иметь возможность удалять запись из базы данных | С4 |
| **5** | Разработка модуля поддержки | Пользователь должен иметь возможность вызвать помощь и получать нужную ему информацию | С5 |
| **6** | Разработка модуля настроек | Пользователь должен иметь возможность настроить своего бота для комфортного использования | С6 |
| **7** | Разработка модуля настроек | Пользователь должен иметь возможность поделится записью | С7 |
| **8** | Обеспечение производительности | Чат-бот корректно запускается в приложении Telegram на системе Android | С8 |
| **9** | Обеспечение производительности | Чат-бот корректно запускается в приложении Telegram на системе IOS | С9 |
| **10** | Интерфейс | Весь текст вводимый пользователем и окно с кнопками должны быть хорошо видны и различимы  пользователем | C10 |
| **11** | Интерфейс | Пользователь должен иметь возможность просматривать описания функционала бота | C11 |
| **12** | Разработка функционала бота | Пользователю должен быть доступен весь представленный функционал бота | C12 |

*Продолжение таблицы 3.1 — Матрица прослеживания требований*

## Анализ техник тестирования требований

С целью соблюдения требований заказчика необходимо провести тестирование разработанной системы. Среди наиболее часто используемых методик тестирования выделяют: метод просмотра, формулировка вопросов, формирование чек-листов или тест-кейсов проверяемости, графическое представление, исследование поведения, прототипирование [3].

## Метод просмотра

Среди методов просмотра можно выделить беглый просмотр, технический просмотр и формальную инспекцию. Краткая характеристика этих методов представлена в таблице 3.2.

*Таблица 3.2 — Характеристика методов просмотра*

|  |  |  |
| --- | --- | --- |
| **Наименование метода просмотра** | **Цель просмотра** | **Особенности** |
| Беглый | Получение обратной связи, обмен результатами между двумя и более авторами | Самый дешевый, быстрый и популярный способ |
| Технический | Получение обратной связи от группы специалистов | Тестируемый продукт не может считаться достаточно качественным, пока хотя бы у одного просматривающего остаются замечания |
| Формальная инспекция | Документирование и структурирование анализа документации | Большое количество специалистов, требует большого количества времени, используется редко |

## Формулировка вопросов

Подразумевает подготовку списка вопросов, на которые клиент должен будет ответить. Ответы на вопросы должны быть полными и обоснованными.

К достоинствам данного метода можно отнести, что задача решается качественнее и более ориентированно на конкретный проект. Основными недостатками является возможно долгое ожидание ответа от заказчика; возникновение недопонимания при неоднозначной формулировке вопроса.

## Формирование чек-листов или тест-кейсов проверяемости

Чек-лист — список факторов, свойств, параметров, аспектов, компонентов, критериев или задач, структурированных особым образом с целью достижения поставленных задач.

Тест-кейс — это профессиональная документация тестировщика, последовательность действий, направленная на проверку какого-либо функционала, описывающая как прийти к фактическому результату.

Набор тест-кейсов называют тест-комплектом.

В процессе составления тест-кейса тестировщику становится понятно, насколько корректно тестируемое требование. Однако тест-кейсы требуют постоянного обновления и поддержки; ресурсы квалифицированных работников тратятся на выполнение последовательностей, для которых не нужно квалификации [4].

## Графическое представление

В целях повышения наглядности представления системы используются графические изображения: рисунки, иллюстрации, схемы, 3D-модели. Графические изображения воспринимаются заказчиками более доступно, чем в текстовой форме. Однако разработка графических представлений не всегда возможна, трудозатратна и не всегда оправдана.

## Исследование поведения

Исследование поведения предполагает, что тестировщик должен мысленно промоделировать процесс работы пользователя с системой. В процессе моделирования осуществляет поиск неоднозначностей и неописанных вариантов поведения системы. Этот подход сложен, требует достаточной квалификации тестировщика, но способен выявить недоработки, которые невозможно заметить, тестируя требования по отдельности.

## Прототипирование

Прототипирование часто является следствием создания графического представления и анализа поведения системы. С использованием специальных инструментов можно быстро сделать наброски пользовательских интерфейсов, оценить применимость тех или иных решений и даже создать заготовку для дальнейшей разработки, возможно, что реализованное в прототипе с небольшими доработками устраивает заказчика.

## Применение методологий

С целью улучшения работы системы клиенту был предложен опросник, представленный в таблице 3.3.

## Подготовка списка вопросов

*Таблица 3.3 — Список вопросов*

|  |  |
| --- | --- |
| **Требование** | **Вопрос** |
| **1** | Есть ли возможность у пользователя сохранять результат запроса? |
| **2** | Какие есть ограничения по вводу в текстовое поле? |
| **3** | Каким функционалом должен обладать бот-помощник? |
| **4** | Как именно пользователь будет получать запрошенную информацию? |
| **5** | Как пользователю стоит оказывать помощь? |

## Вывод

В процессе анализа существующих техник тестирования требований были выявлены особенности перечисленных методов, их преимущества и недостатки. Исходя из этого, лучшим вариантом проведения тестирования для заданного проекта была выбрана и использована техника тестирования требований методом вопросов, так как это очень бюджетный способ, а разработка подобного приложения не занимает слишком много времени, также программа не имеет много функций, что делает этот способ самым эффективным.

# ФОРМИРОВАНИЕ ТЕСТОВОГО ПЛАНА

## Анализ существующих подходов и международных стандартов, применяемых для написания тестового плана

В настоящее время, в зависимости от типа решаемых задач, можно выделить следующие виды тестовых планов:

* Мастер-план тестирования.
* Детальный тестовый план.
* Планы тестирования для конкретных типов тестирования.

Охарактеризуем каждый из перечисленных тестовых планов и определим оптимальный для решения поставленных задач.

## Мастер-план тестирования

Мастер-план — документ высокого уровня, описывающий не только различные тестовые планы, которые в него войдут (каждый отдельным документом), но и дополнительную критическую информацию, затрагивающую весь процесс тестирования во время работы [5].

В мастер тест-плане содержится следующая информация:

* общая информация о продукте, ссылки на документацию, баг-трекер и прочие проектные ресурсы;
* общие правила тестирования: требования к заводимым дефектам, условия принятия сборки на тестирование;
* критерии готовности продукта к выпуску, метрики качества;
* используемые инструменты и техники.

Мастер-план создается в следующих случаях: если продукт имеет множество релизов или итераций, между которыми сохраняется общая информация, которую нет смысла повторять и, если различные тестовые команды работают над одним продуктом, выполняя различные задачи, которые необходимо объединить в рамках одного документа.

## Детальный тестовый план

Детальный тест-план составляется на каждый релиз/итерацию или для каждой команды в рамках проекта. Его основная цель - кратко и доходчиво показать задачи тестирования [6].

В детальном тест-плане содержится следующая информация: Перечень областей тестирования с приоритетами, стратегия тестирования, проектные риски, ресурсы, необходимые для выполнения задач, проектный план (сроки готовности ключевых задач).

## Планы тестирования для конкретных типов тестирования

Для планов тестирования производительности и планов тестирования безопасности и других задач организации могут создаваться отдельные планы тестирования.

## Вывод

В зависимости от целей и задач тестирования объема работ используются следующие тестовые планы: мастер-план тестирования, детальный тестовый план и планы тестирования для конкретных типов. В пункте 4.2 представлен тестовый план в соответствии со стандартом IEEE 829-2008.

## Создание тестового плана

* + 1. **Определение цели тестирования**

Цель тестирования — проверка соответствия между реальным поведением программы и её ожидаемым поведением на конечном наборе тестов, выбранных определённым образом.

## Области, подвергаемые тестированию

В таблице 4.1 приводятся области системы, которые буду протестированы и их приоритет, где 1 – критически важно, 5 – незначительно.

*Таблица 4.1 — Области, подвергаемые тестированию*

|  |  |  |  |
| --- | --- | --- | --- |
| **№** | **Область** | **Описание** | **Приоритет** |
| **1** | Разработка модуля поддержки | Пользователь должен иметь возможность вызвать помощь и получать нужную ему информацию | 3 |
| **2** | Обеспечение производительности | Чат-бот корректно запускается в приложении Telegram на системе Android | 1 |
| **3** | Интерфейс | Весь текст, вводимый пользователем, и окно с кнопками должны быть хорошо видны и различимы  пользователем | 2 |
| **4** | Интерфейс | Пользователь должен иметь возможность просматривать описания функционала бота | 4 |
| **5** | Разработка функционала бота | Пользователю должен быть доступен весь представленный функционал бота | 1 |

## Области, не подвергаемые тестированию

В проекте будут протестированы не все функциональные возможности.

Области, не подвергаемые тестированию, приведены в таблице 4.2.

*Таблица 4.2 — Области, не подвергаемые тестированию*

|  |  |  |  |
| --- | --- | --- | --- |
| № | **Область** | **Описание** | **Причина отказа** |
| 1 | Разработка базы данных | Пользователь должен иметь возможность самостоятельно добавлять запись в базу данных | Система не имеет слишком сложную структуру, и она не хранит несколько объектов одновременно, следовательно нет необходимости доступа пользователя к базе данных |
| 2 | Разработка базы данных | Пользователь должен иметь возможность изменить запись из базы данных | Система не имеет слишком сложную структуру, и она не хранит несколько объектов одновременно, следовательно нет необходимости доступа пользователя к базе данных |
| 3 | Разработка базы данных | Пользователь должен иметь возможность просматривать записи, содержащиеся в базе данных | Система не имеет слишком сложную структуру, и она не хранит несколько объектов одновременно, следовательно нет необходимости доступа пользователя к базе данных |
| 4 | Разработка базы данных | Пользователь должен иметь возможность удалять запись из базы данных | Система не имеет слишком сложную структуру, и она не хранит несколько объектов одновременно, следовательно нет необходимости доступа пользователя к базе данных |
| 5 | Разработка модуля настроек | Пользователь должен иметь возможность настроить своего бота для комфортного использования | Приложение имеет простую структуру и настройки не требуются. |
| 6 | Разработка модуля настроек | Пользователь должен иметь возможность поделится записью | Приложение имеет простую структуру и возможность поделится записью не требуется. |
| 7 | Обеспечение производительности | Приложение запускается на системе iOS | Большинство русскоязычных пользователей пользуются Android, отсутствует необходимое оборудование |

## Разработка тестовой стратегии

Стратегия тестирования будет основана на следующих методах: тестирование белого ящика, функциональное тестирование.

## Методы тестирования

Основным методом тестирования будет являться тестирование метода белого ящика, поскольку тестировщик принимал участие в разработке подобного продукта и знаком с процессом создания программ с таким функционалом [7].

Функциональное тестирование – тестирование соответствия поведения системы ожиданиям. Проверяется правильность, надёжность, тестируемость и точность вывода системы.

## Определения серьезности ошибок

В таблице 4.3 показано распределение уровней серьезностей ошибок.

*Таблица 4.3 — Распределение уровней серьезностей ошибок*

|  |  |  |
| --- | --- | --- |
| **Уровень серьёзности** | **Расшифровка уровня** | **Описание** |
| 1 | Блокирующая | Блокирует разработку и/или тестирование, производство не может быть запущено |
| 2 | Критическая | Системные сбои, ошибки или повреждения файла, потенциальная потеря данных, программные зависания, требующие перезагрузки |
| 3 | Серьезная | Вызывает недостаточную функциональность или значительно замедляет выполнение функций. Имеет сложный способ обхода. |
| 4 | Незначительная | Незначительная потеря функциональности или другая проблема, при которой есть простой обходной путь |
| 5 | Тривиальная | Косметическая проблема, такая как неправильно написанные слова или неровный текст. |

## Определение критериев тестирования

Критерии начала тестирования: готовность тестового стенда, включая все аппаратные и программные составляющие; все программные инструменты должны быть успешно установлены и функционировать должным образом; тестовые данные должны быть подготовлены.

Критерии окончания тестирования: был достигнут определенный уровень покрытия кода и требований; ошибок высокого приоритета или серьезных ошибок не осталось; тестирование удалось.

Критерии приостановки тестирования: выявлены нарушения более, чем 20% требований к функциональности приложения; программные / аппаратные проблемы; возникла иная серьезная проблема, которая не позволяет продолжить тестирование.

## Планирование ресурсов

Для успешного применения стратегии тестирования необходимы следующие ресурсы: программные, аппаратные, человеческие и временные. В таблице 4.4 показано распределение ресурсов.

*Таблица 4.4 — Распределение ресурсов*

|  |  |
| --- | --- |
| **Ресурсы** | **Состав ресурса** |
| Программные | ОС Android не ниже Android 4.4 KitKat. |
| Аппаратные | Рабочий смартфон со свободным местом в размере 175 МБ. |
| Человеческие | Ведущий тестировщик, менеджер продукта с опытом управления проектами участвует на протяжении всего процесса тестирования. Тест-аналитик с опытом тестирования участвует в начале процесса тестирования.  Руководитель разработки, по контролю качества, проекта. |
| Временные | На проведение процесса тестирования необходимо 2 месяца. |

## Расписание времени выполнения тестирования

* 01.09.2021 – 07.09.2021 — разработка мастер-плана тестирования.
* 07.09.2021 – 12.09.2021 — проработка требований заказчика.
* 13.09.2021 – 20.09.2021 — разработка тестовых случаев.
* 21.09.2021 – 01.10.2021 — выполнение тестовых сценариев.
* 02.10.2021 – 17.10.2021 — тестирование удобства использования.
* 18.10.2021 – 30.10.2021 — завершения тестирования и подведение итогов.

## Оценка рисков

Руководитель по контролю качества, разработки и руководитель проекта примут участие в совещаниях по рассмотрению ошибок, определят приоритет всех текущих активных ошибок. Ответственный за обеспечение качества отвечает за организацию этих совещаний на регулярной основе для устранения текущего набора новых и существующих, но нерешенных проблем.

## Тестовая документация

*Таблица 4.5 –* *Тестовая документация*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **№** | **Название** | **Ответственные** | **Сроки предоставления** | **Метод предоставления** |
| 1 | План тестирования | Руководитель проекта | Единоразово, прежде чем приступить к тестированию | Microsoft Word |
| 2 | Чек-листы | Руководитель проекта | Прежде чем приступить к тестированию | Microsoft Word |
| 3 | Тест-кейсы | Руководитель проекта | Прежде чем приступить к тестиро ванию | Microsoft Word |
| 4 | Отчёт об ошибках | Тестировщик | При выявлении ошибки | Microsoft Word |

## Метрики тестирования

* процентное отношение выполненных тест-кейсов к ранее выполненным не менее 60 %;
* процент успешного завершения тестовых случаем не менее 94 %;
* процент блокировки тест-кейсов — не более 0,03 %;
* эффективность устранения дефектов стремится к 100 %.

## Покрытие требований

Покрытие обеспечивается показателями следующих метрик:

* метрика покрытия требований;
* метрика плотности покрытия требований;
* метрика покрытия классов эквивалентности;
* метрика покрытия граничных условий.

## Вывод

Исходя из назначения и особенностей рассмотренных тест-планов для рассматриваемого проекта был выбран детальный подход к написанию тест- плана. С помощью данного подхода был построен ориентированный на проект тест-план в соответствии со стандартом IEEE 829-2008.

# ФОРМИРОВАНИЕ ТЕСТОВЫХ КЕЙСОВ

## Разработка чек-листа

*Таблица 5.1 — Чек-лист*

|  |  |  |
| --- | --- | --- |
| **№** | **Название тестируемой функции** | **Статус** |
| **Разработка модуля поддержки** | | |
| **1** | Пользователь должен иметь возможность вызвать помощь и получать нужную ему информацию. | Пройден |
| **Обеспечение производительности** | | |
| **2** | Приложение запускается на системе Android. | Пройден |
| **Интерфейс** | | |
| **3** | Весь вводимый текст пользователем и окно с кнопками должны быть хорошо видны и различимы  пользователем. | Пройден |
| **4** | Пользователь должен иметь возможность просматривать описания функционала бота. | Пройден |
| **Разработка функционала бота** | | |
| **5** | Пользователю должен быть доступен весь представленный функционал бота. | Пройден |

## Тест-кейсы

Ориентируясь на составленный чек-лист, разделим тест кейсы на 4 тест- комплекта, соответствующих задачам на разработку. Необходимо обозначить диапазон приоритетов для тест-кейсов. В рамках данной работы было решено использовать приоритеты от 1 до 4, где 1 означает наивысшую важность.

В таблице 5.2 описан тест-кейс для тестирования разработки модуля поддержки. В таблицах 5.3 описан тест-кейс для тестирования обеспечения производительности. В таблицах 5.4-5.5 описаны тест-кейсы для тестирования интерфейса. В таблице 5.6 тест-кейс для тестирования разработки функционала.

*Таблица 5.2 – Тест-кейс проверки возможности вызвать помощь*

|  |  |  |  |
| --- | --- | --- | --- |
| **TC ID / Priority** | C5 | | 3 |
| **IDEA:** Пользователь должен иметь возможность вызвать помощь и получать нужную ему информацию.  **SETUP and ADDITIONAL INFO:** | | | |
| **Revision history** | | | |
| **Created on:** 10.12.2021 by Петров А. В. | | Новый тест-кейс | |
| **Execution part** | | | |
| **PROCEDURE** | | **EXPECTED RESULT** | |
| 1. Нажать на кнопку со значком из 3 палочек. 2. Нажать на кнопку «/help». | | Появятся информация о всех командах | |

*Таблица 5.3 – Тест-кейс проверки запуска приложения на Android*

|  |  |  |  |
| --- | --- | --- | --- |
| **TC ID / Priority** | C8 | | 1 |
| **IDEA:** Приложение запускается на системе Android.  **SETUP and ADDITIONAL INFO:** | | | |
| **Revision history** | | | |
| **Created on:** 10.12.2021 by Петров А. В. | | Новый тест-кейс | |
| **Execution part** | | | |
| **PROCEDURE** | | **EXPECTED RESULT** | |
| 1. Включить смартфон. 2. Запустить приложение Telegram. 3. Зайти в канал к боту. | | Чат-бот запустился. | |

*Таблица 5.4 – Тест-кейс вводимого текста и кнопок*

|  |  |  |  |
| --- | --- | --- | --- |
| **TC ID / Priority** | C10 | | 2 |
| **IDEA:** Весь вводимый текст пользователем и окно с кнопками должны быть хорошо видны и различимы пользователем.  **SETUP and ADDITIONAL INFO:** | | | |
| **Revision history** | | | |
| **Created on:** 10.12.2021 by Петров А. В. | | Новый тест-кейс | |
| **Execution part** | | | |
| **PROCEDURE** | | **EXPECTED RESULT** | |
| 1. Ввести в строку текст и проверить, что текст различим. 2. Нажать на любую кнопку из меню бота и проверить, что кнопка видна. | | Пользователь убедится, что текст и кнопки различимы. | |

*Таблица 5.5 – Тест-кейс проверки описания функционала бота*

|  |  |  |  |
| --- | --- | --- | --- |
| **TC ID / Priority** | C11 | | 4 |
| **IDEA:** Пользователь должен иметь возможность просматривать описания функционала бота.  **SETUP and ADDITIONAL INFO:** | | | |
| **Revision history** | | | |
| **Created on:** 10.12.2021 by Петров А. В. | | Новый тест-кейс | |
| **Execution part** | | | |
| **PROCEDURE** | | **EXPECTED RESULT** | |
| 1. При первом подключении к боту, посмотреть на его сообщение. 2. В сообщение указано описание функционала. | | Описание функционала представлено пользователю. | |

*Таблица 5.6 – Тест-кейс проверки функционала бота*

|  |  |  |  |
| --- | --- | --- | --- |
| **TC ID / Priority** | C12 | | 1 |
| **IDEA:** Пользователю должен быть доступен весь представленный функционал бота.  **SETUP and ADDITIONAL INFO:** | | | |
| **Revision history** | | | |
| **Created on:** 10.12.2021 by Петров А. В. | | Новый тест-кейс | |
| **Execution part** | | | |
| **PROCEDURE** | | **EXPECTED RESULT** | |
| 1. Нажать на меню бота. 2. Протестировать каждую кнопку. 3. Дождаться после каждого нажатия на кнопку появления соответствующей картинке. | | Функционал бота-помощника будет протестирован. | |

## Выводы

В данном разделе были разработаны и сформированы тест-кейсы и чек-листы для описанных в области для тестирования тестового плана тестируемых требований. Эти тест-кейсы и чек-листы будут использоваться при тестировании программного продукта.

# ТЕСТИРОВАНИЕ ПРОГРАММНОГО МОДУЛЯ

## Выполнение тест-кейсов

Результаты выполнения тест-кейса C5 представлен на рисунке 6.1.

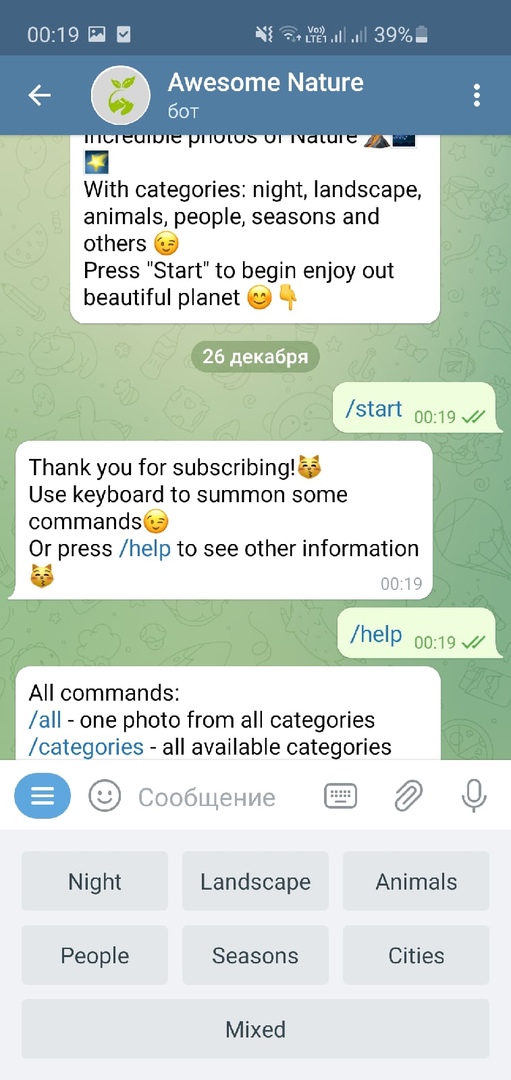


Рисунок 6.1 — Экран «Помощь»

Результаты выполнения тест-кейса C8 представлены на рисунках 6.2. и 6.3.



Рисунок 6.2 — иконка приложения



Рисунок 6.3 — Запущенное приложение

Результаты выполнения тест-кейса C10 представлен на рисунке 6.4

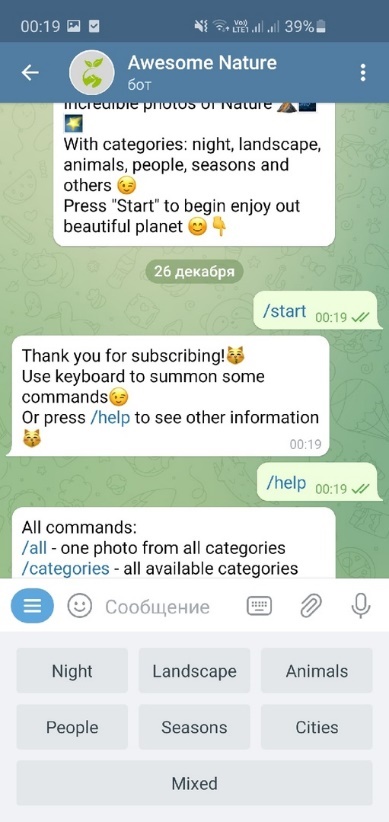


Рисунок 6.4 — Отправленный текст и интерфейс кнопок

Результаты выполнения тест-кейса C11 представлены на рисунках 6.5.



Рисунок 6.5 — Описание функционала бота

Результаты выполнения тест-кейса C12 представлены на рисунках 6.6-6.12.

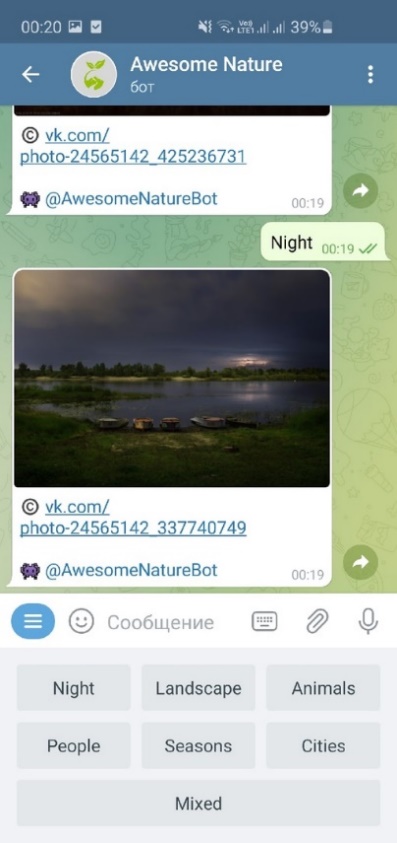


Рисунок 6.6 — Результат нажатие на кнопку “Night”

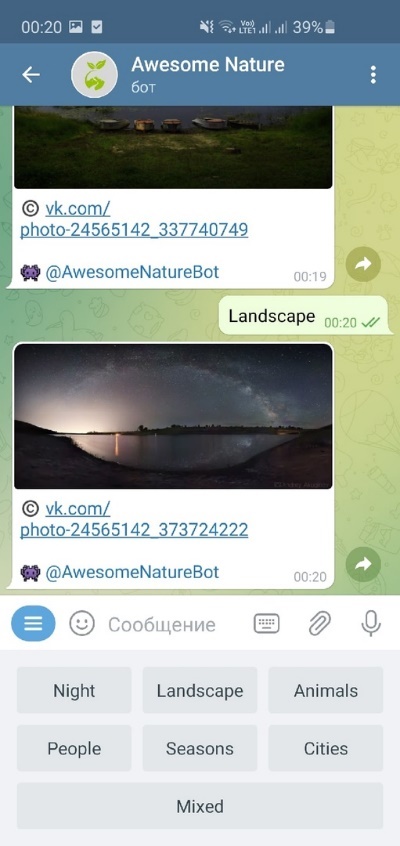


Рисунок 6.7 — Результат нажатие на кнопку “Landscape”

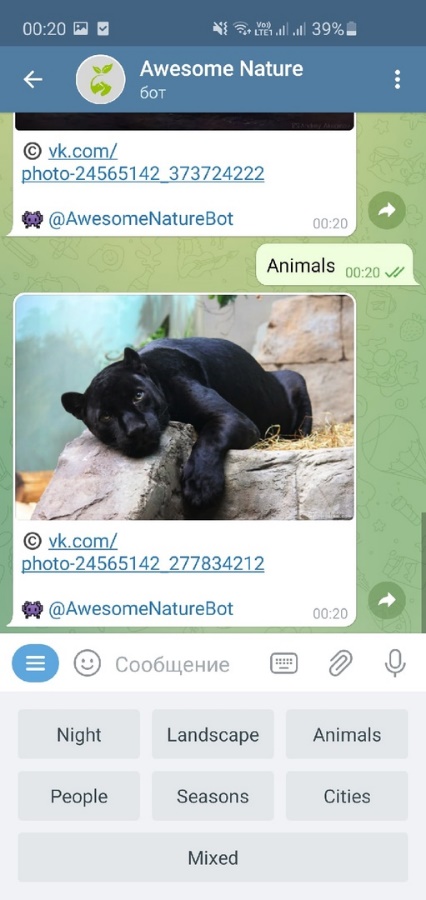


Рисунок 6.8 — Результат нажатие на кнопку “Animals”



Рисунок 6.9 — Результат нажатие на кнопку “People”

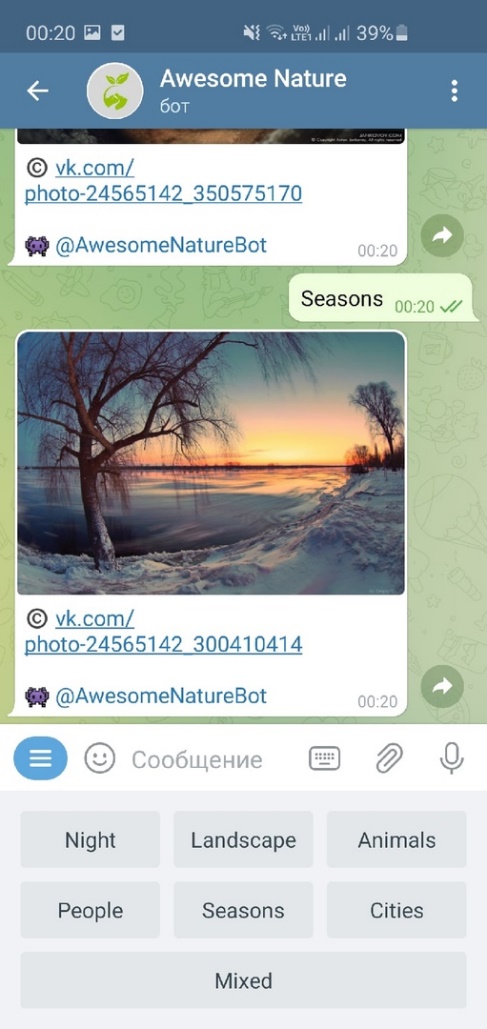


Рисунок 6.10 — Результат нажатие на кнопку “Seosons”



Рисунок 6.11 — Результат нажатие на кнопку “Cities”

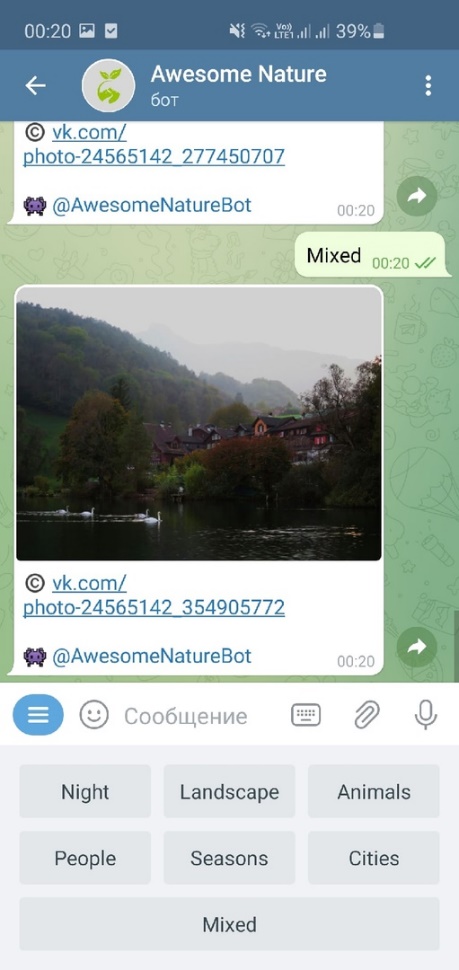


Рисунок 6.12 — Результат нажатие на кнопку “Mixed”

# ЗАКЛЮЧЕНИЕ

В процессе выполнения курсовой работы была выполнена поставленная цель — провести тестирование чат-бота для мессенджера Telegram. На протяжении процесса тестирования приложения были изучены основы тестирования с помощью различных методик.

Для тестирования программного продукта были изучены современные технологии планирования хода проведения тестирования программного обеспечения различного назначения, проведения тестовых испытаний, документирования, подготовки чек-листов и тест-кейсов и проведения тестирования согласно этим тест-кейсам.

По итогам курсовой работы были достигнуты поставленные цели: были получены практические навыки в проектировании и разработке, планировании хода проведения тестирования, проведении тестовых испытаний программного обеспечения; подготовки документации по проведенным действиям и составлении отчетов о результатах тестирования.

# СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. ISO/IEC TR 19759:2005 — ISO/IEC TR 19759:2015 Software Engineering — Guide to the software engineering body of knowledge (SWEBOK).
2. [Электронный ресурс]: Идеи для чат-ботов. URL: https://tgrm.su/blog/faq/boty-telegram/#4 (дата обращения: 01.12.2021)
3. Кент Б. Экстремальное программирование: разработка через тестирование. — «Издательский дом Питер», 2017.
4. Блэк Р. Ключевые процессы тестирования. Планирование, подготовка, проведение, совершенствование // М.: Лори. — 2006.
5. Криспин Л., Грегори Д. Гибкое тестирование. Практическое руководство для тестировщиков ПО и гибких команд //М.: Вильямс. — 2010.
6. Савин Р. Тестирование «Dot Com» //Изд. дом «Дело». — 2007.

— 310 с. – 2007.

1. Бейзер,Б. Тестирование черного ящика. Технологии функционального тестирования программного обеспечения и систем. / Борис Бейзер. — Питер, 2004. – 187 с.