



Министерство науки и высшего образования Российской Федерации
Федеральное государственное бюджетное образовательное учреждение
высшего образования
«Московский государственный технический университет
имени Н.Э. Баумана
(национальный исследовательский университет)»
(МГТУ им. Н.Э. Баумана)

ФАКУЛЬТЕТ ИНФОРМАТИКА И СИСТЕМЫ УПРАВЛЕНИЯ

КАФЕДРА КОМПЬЮТЕРНЫЕ СИСТЕМЫ И СЕТИ (ИУ6)

НАПРАВЛЕНИЕ ПОДГОТОВКИ 09.03.01 Информатика и вычислительная техника

БАКАЛАВРСКАЯ ПРОГРАММА 09.03.01/03 Вычислительные машины, комплексы,
системы и сети

ОТЧЕТ ПО ПРОИЗВОДСТВЕННОЙ ПРАКТИКЕ

Тип практики Преддипломная практика

Название
предприятия «НУК ИУ МГТУ им. Н.Э.Баумана»

Студент ИУ6-82Б

03.03.2023 М.Э. Хабаров
(Подпись, дата) (И.О. Фамилия)

Руководитель практики

03.03.2023 В.В. Гуренко
(Подпись, дата) (И.О. Фамилия)

Оценка

Отлично

2023 г.

ЗАДАНИЕ на производственную практику

по теме Программная система отбора медицинских новостей

Студент группы ИУ6-82Б

Хабаров Марк Эдуардович

(Фамилия, имя, отчество)

Направление подготовки 09.03.01 Информатика и вычислительная техника

Бакалаврская программа 09.03.01/03 Вычислительные машины, комплексы, системы и сети

Тип практики Преддипломная практика

Название предприятия НУК ИУ МГТУ им. Н.Э. Баумана

Техническое задание Выполнить анализ предметной области программной системы отбора медицинских новостей. На основе проведенного анализа разработать прототип программной системы отбора медицинских новостей.

Оформление отчета по практике:

Отчет на 15-25 листах формата А4.

Графический и иллюстративный материал оформляется в виде рисунков и помещается в отчет.


Дата выдачи задания « 07 » февраля 2023 г.

Руководитель практики

 07.02.2023
(Подпись, дата)

М.Э. Хабаров
(И.О. Фамилия)

Студент

 04.02.2023
(Подпись, дата)

В.В. Гуренко
(И.О. Фамилия)

Примечание: Задание оформляется в двух экземплярах.

СОДЕРЖАНИЕ

Сокращения и определения	4
Введение	5
1 Анализ технических требований и уточнение спецификаций	6
1.1 Анализ и выбор технологии, языка и среды разработки	6
1.2 Разработка диаграммы вариантов использования	7
2 Проектирования структуры и компонентов программного продукта	12
2.1 Разработка структурной схемы программного продукта.....	12
2.2 Проектирование пользовательского интерфейса	14
2.2.1 Построение графа состояний интерфейса	14
2.2.2 Проектирование диалогов и форм интерфейса	15
2.3 Проектирование базы данных.....	20
2.3.1 Разработка инфологической модели базы данных	20
2.3.2 Разработка даталогической модели базы данных	22
2.4 Разработка диаграммы компоновки программной системы	23
Заключение.....	25
Список использованных источников.....	26

СОКРАЩЕНИЯ И ОПРЕДЕЛЕНИЯ

Фейк (от англ. fake – подделка, фальсификация, обман) – наименование изменённой, искажённой или заведомо ложной информации, представленной в виде текста, фото или видеофайлов.

DRY (от англ. don't repeat yourself – не повторяйся) – основополагающая концепция разработки, заключающаяся в том, что нужно избегать дублирование одного и того же кода.

KISS (от англ. keep it stupid simple – будь проще) – принцип разработки, гласящий, что простые решения задачи будут работать лучше и надёжнее, чем сложные.

API (от англ. application programming interface – программный интерфейс приложения) – это набор способов и правил, по которым различные программы взаимодействуют и обмениваются данными.

ВВЕДЕНИЕ

В процессе прохождения преддипломной практики была поставлена цель – выполнить анализ предметной области и разработать прототип программной системы отбора медицинских новостей.

Основной задачей преддипломной практики является проектирование классификатора для реализации функции рекомендации, можно ли доверять полученному от пользователя контенту, а также разработка на его основе прототипа программной системы отбора медицинских новостей.

Актуальность данной темы заключается в том, что проблема фейковых новостей привлекла большое внимание ещё в 2007 году, когда была опубликована разоблачительная информация о всемирно известных издательствах, которые не пренебрегали фабрикацией фактов. Однако угроза сфальсифицированной информации остаётся открытой в информационной повестке и в настоящее время.

1 Анализ технических требований и уточнение спецификаций

1.1 Анализ и выбор технологии, языка и среды разработки

Для упрощения манипуляций данными и соблюдения принципов DRY и KISS, приводящих к модульной архитектуре, был выбран смешанный подход программирования. Подобное решение было обусловлено тем, что структурный подход, позволяющий получать достаточно надёжные программы и подпрограммы, является основным подходом к проектированию прототипа программной системы отбора медицинских новостей, однако, при работе с базой данных проявляется объектный подход, позволяющий облегчить процесс работы с информацией [1].

При реализации проекта был выбран высокоуровневый язык программирования Python, главным достоинством которого является простота команд и синтаксиса, а также широкий спектр библиотек для решения большого разнообразия задач. Для удобства использования программную систему было принято решение реализовать на основе Telegram. Для взаимодействия с Telegram API был выбран Aiogram, который является современной библиотекой, работающей с асинхронным подходом, что позволяет не останавливать работу системы в ожидании ответа пользователя. Также Aiogram обладает подробной документацией и большим русскоязычным сообществом [2].

Для работы с медицинскими новостями было принято решение использовать библиотеку PyMorphu2, которая представляет собой морфологический анализатор русских текстов и работает с словарём, созданным на базе «Грамматического словаря русского языка» А.А. Зализняка. Для реализации возможности совершения покупок была выбрана российская платёжная система Qiwi, доступ к API которой осуществляется с помощью библиотеки ruqiwi2p.

В качестве СУБД была выбрана встраиваемая СУБД SQLite, которая характеризуется простотой использования и высокой скоростью операций, так

как все запросы и команды к базе данных идут напрямую из системы, а также обладает высокой стабильностью и надёжностью.

В качестве редактора исходного кода для разработки и отлаживания прототипа была выбрана среда Visual Studio Code от компании Microsoft. VS Code – это надёжный и нетребовательный к ресурсам инструмент обладающий функциями IDE, и вокруг которого собралось мощное сообщество разработчиков расширений.

1.2 Разработка диаграммы вариантов использования

Use-case диаграмма предназначена для описания взаимодействия системы с действующим лицом – пользователя с активной или неактивной подпиской. В результате анализа задания для программной системы отбора медицинских новостей были выделены следующие варианты использования:

- получение приветственного сообщения;
- перезагрузка системы;
- получение описания возможностей системы;
- вызов меню изменения языка;
- переключение текущего языка;
- обработка текстов новостей;
- отправка текстового сообщения;
- вызов меню для покупки подписки;
- просмотр списка товаров;
- покупка подписки;
- получение сообщения с результатом оплаты;
- получение сообщения о некорректном вводе.

Основные варианты использования рассмотрены более подробно в таблицах 1–3.

Таблица 1 – Вариант использования «обработка текстов новостей».

Действие исполнителя	Отклик системы
1. Пользователь, находясь на главной странице, нажимает на кнопку «Начать работу».	2. Демонстрация сообщения о готовности к обработке информации.
3. Пользователь отправляет текстовое сообщение.	4. Система осуществляет проверку на длину сообщения и на наличие активной подписки у пользователя. Если длина не превышает 1500 символов или есть активная подписка, то демонстрируется сообщение с результатами обработки информации, иначе демонстрируется сообщение о превышении допустимой длины необходимости покупки подписки.

Таблица 2 – Описание варианта использования «вызов меню изменения языка».

Название варианта	Вызов меню изменения языка
Цель	Переключение текущего языка
Действующие лица	Пользователь с активной или неактивной подпиской
Краткое описание	Нажатием кнопки «Сменить язык» вызывает сообщение с возможностью выбора языка. После выбора необходимого языка происходит локализация системы.
Тип	Основной

Таблица 3 – Вариант использования «вызов меню покупки подписки».

Действие исполнителя	Отклик системы
1. Пользователь нажимает на кнопку «Купить подписку».	2. Система осуществляет проверку на наличие активной подписки и, если находит её, то демонстрирует сообщение со сроком окончания подписки. Если пользователь не имеет активной подписки, то демонстрируется сообщение со списком товаров и открывается меню покупки.

Для наглядного представления возможностей пользователя на рисунке 1 представлена диаграмма вариантов использования.

На диаграмме вариантов использования показано, что у программной системы отбора медицинских новостей двадцать три варианта использования, два действующих лица (пользователь с активной подпиской и пользователь с неактивной подпиской) и связи между вариантами использования и действующими лицами.

2 Проектирование структуры и компонентов программного продукта

2.1 Разработка структурной схемы программного продукта

Процесс проектирования программного продукта начинается с уточнения его структуры, то есть определения структурных компонентов и связей между ними. Результатом подобного уточнения структуры может являться структурная схема, показывающая наличие подсистем и других структурных элементов. Разработку структурной схемы выполняют методом пошаговой детализации [1].

Исходя из задач и функций, которые выполняет система в целом и её подсистемы в частности, была получена структурная схема программной системы отбора медицинских новостей (см. рисунок 2).



Рисунок 2 – Схема структурная программного продукта

Структурная схема программной системы отбора медицинских новостей содержит пять подсистем, десять сопрограмм и два модуля.

2.2 Проектирование пользовательского интерфейса

2.2.1 Построение графа состояний интерфейса

В соответствии с вариантами использования можно выделить следующие состояния интерфейса:

- приветственная страница, содержащая описание программной системы;
- стартовое состояние;
- состояние смены языка;
- состояние обработки информации;
- состояние списка цен;
- состояние покупки;
- форма оплаты.

На основе выделенных выше состояний интерфейса был разработан граф состояний интерфейса, который представлен на рисунке 3.

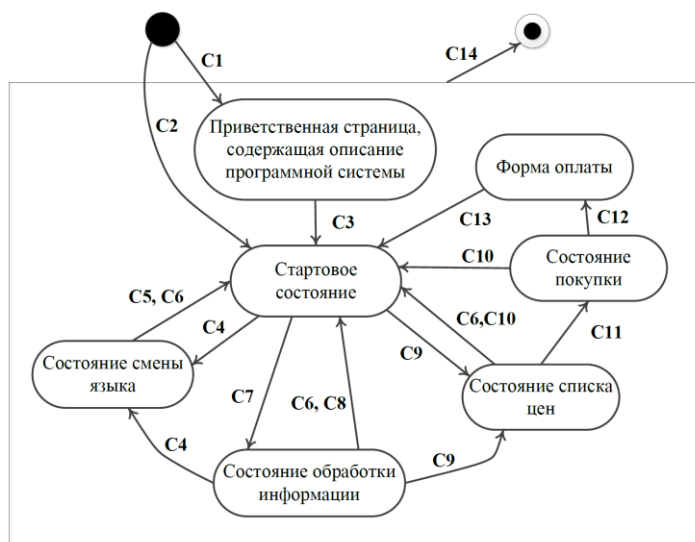


Рисунок 3 – Граф состояний интерфейса

События, которыми вызваны переходы между состояниями:

- C1 – первое открытие программной системы;
- C2 – открытие программной системы;

- C3 – нажатие на кнопку «Старт»;
- C4 – нажатие на кнопку «Сменить язык»;
- C5 – нажатие на кнопку выбора языка;
- C6 – нажатие на кнопку перезапуска системы;
- C7 – нажатие на кнопку «Начать работу»;
- C8 – нажатие на кнопку «Завершить работу»;
- C9 – нажатие на кнопку «Купить подписку»;
- C10 – нажатие на кнопку «Главное меню»;
- C11 – нажатие на одну из кнопок товара;
- C12 – нажатие на кнопку «Оплатить»;
- C13 – нажатие на кнопку «Оформить заказ» в форме оплаты Qiwi;
- C14 – закрытие программной системы.

Разработанный граф состояний интерфейса имеет семь состояний и четырнадцать событий.

2.2.2 Проектирование диалогов и форм интерфейса

При разработке пользовательского интерфейса, который представляет собой совокупность аппаратных и программных средств, обеспечивающих взаимодействие пользователя с системой, было принято решение создавать процедурно-ориентированный интерфейс со свободной навигацией. Такой подход подразумевает, что интерфейсы ориентированы на использование экрана в графическом режиме с высокой разрешающей способностью [1].

Существенной особенностью GUI является способность изменяться в процессе взаимодействия с пользователем, предлагая выбор только тех операций, которые имеют смысл в конкретной ситуации.

В процессе разработки интерфейса было установлено, что основой взаимодействия человека с программным продуктом является диалог комбинированного типа. Основываясь на этом утверждении, был разработан граф абстрактного диалога программного продукта (см. рисунок 4) [3].

Таким образом, в процессе создания программного продукта, были спроектированы четыре основные формы интерфейса:

- приветственная страница;
- главная страница;
- обработка информации от пользователя;
- покупка подписки.

Указанные выше формы интерфейса представлены на рисунках 5-8.

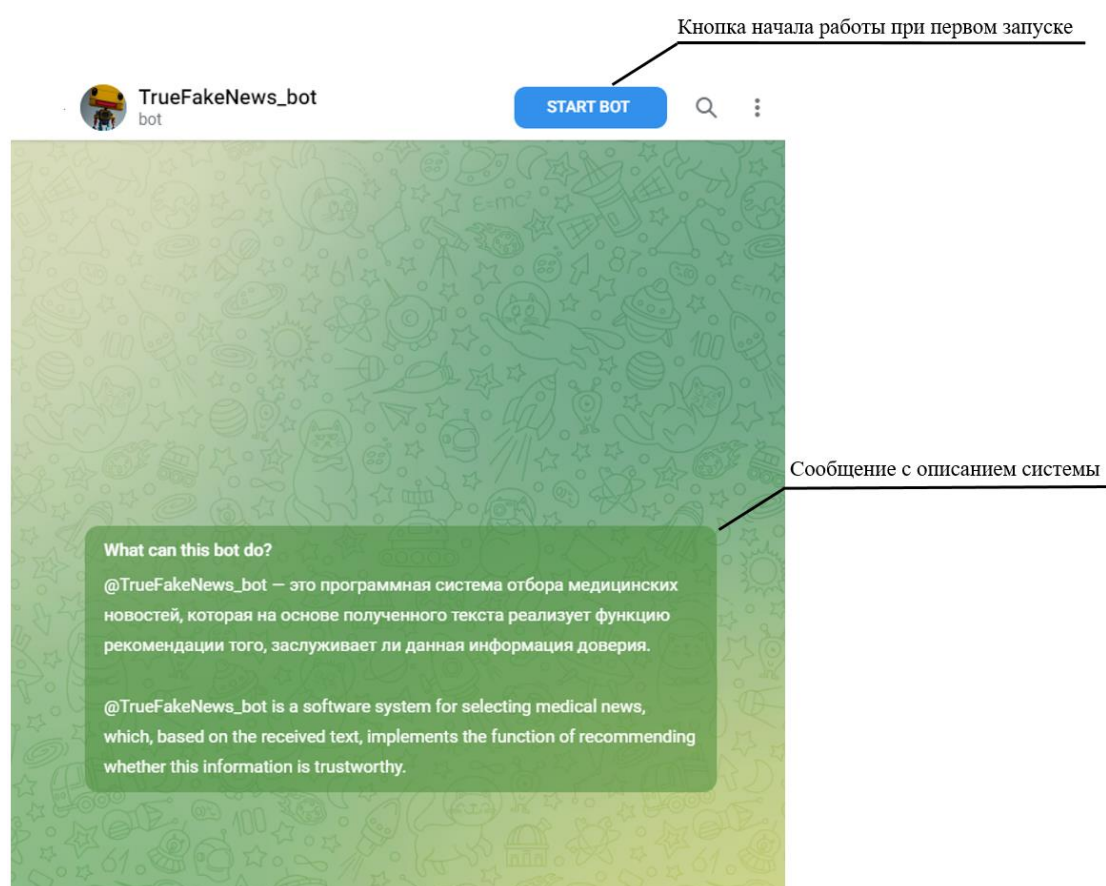


Рисунок 5 – Экранная форма «Приветственная страница»

На приветственной странице, представленной на рисунке 5, пользователь, впервые запустивший систему, может:

- ознакомиться с описанием системы;
- начать работу с системой при помощи кнопки «Start».

На главной странице, представленной на рисунке 6, пользователь получает стартовое (приветственное) сообщение и может:

- начать обработку текстов новостей при помощи кнопки «Начать работу»;

- запросить сообщение с возможностью выбора языка системы, нажав кнопку «Сменить язык»;
- запросить список товаров и открыть меню покупки при помощи кнопки «Купить подписку»;
- перезагрузить систему или запросить сообщение о возможностях системы, выбрав соответствующий пункт в блоке меню;
- получить сообщение о некорректных действиях при попытке отправить сообщение системе.

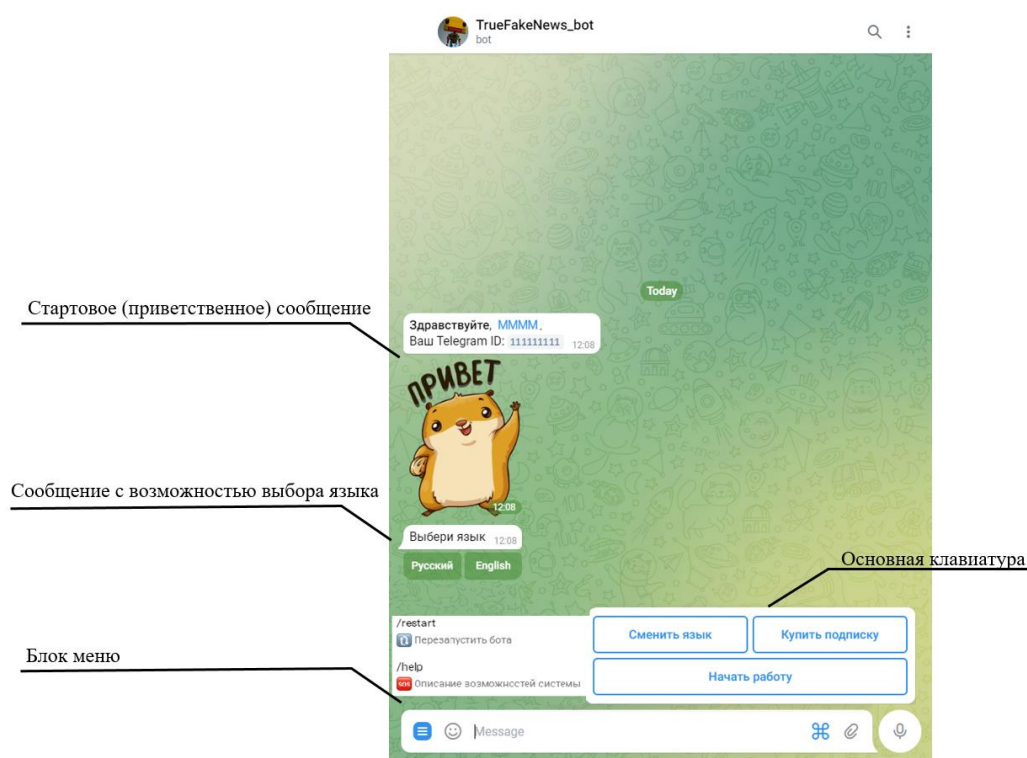


Рисунок 6 – Экранная форма «Главная страница»

На странице обработки информации, представленной на рисунке 7, пользователь получает сообщение о готовности к обработке и может:

- отправить текст новости для обработки;
- получить сообщение о некорректном вводе при отправке системе не текстового сообщения;
- получить сообщение о необходимости покупки подписки для обработки текстов длиной более 1500 символов при отсутствии подписки и отправке информации, превышающей максимальный размер;
- получить сообщение с результатами обработки информации;

- перезагрузить систему или запросить сообщение о возможностях системы, выбрав соответствующий пункт в блоке меню;
- завершить обработку информации при помощи кнопки «Завершить работу»;
- запросить сообщение с возможностью выбора языка системы, нажав кнопку «Сменить язык»;
- запросить список товаров и открыть меню покупки при помощи кнопки «Купить подписку».

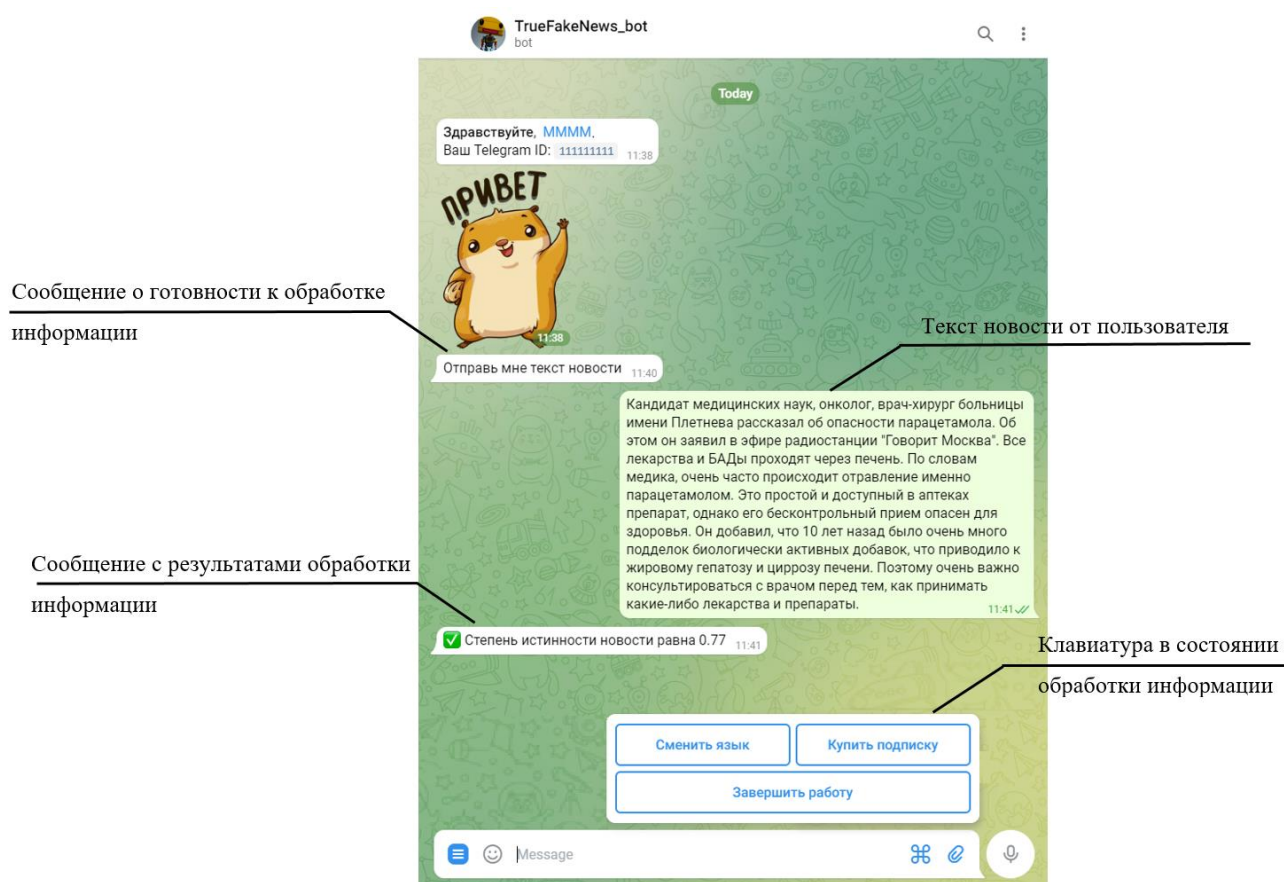


Рисунок 7 – Экранная форма «Страница обработки информации»

На странице покупки подписки, представленной на рисунке 8, пользователь получает сообщение со списком товаров и может:

- получить сообщение с ID транзакции и кнопкой «Оплатить» при помощи нажатия на клавиатуре одной из кнопок с наименованием товара;
- вернуться на главную страницу при помощи нажатия на кнопку «В главное меню»;
- перейти на форму оплаты Qiwi при нажатии на кнопку «Оплатить»;

- получить сообщение об успешности или не успешности оплаты;
- получить сообщение о некорректных действиях при попытке отправки сообщения системе.

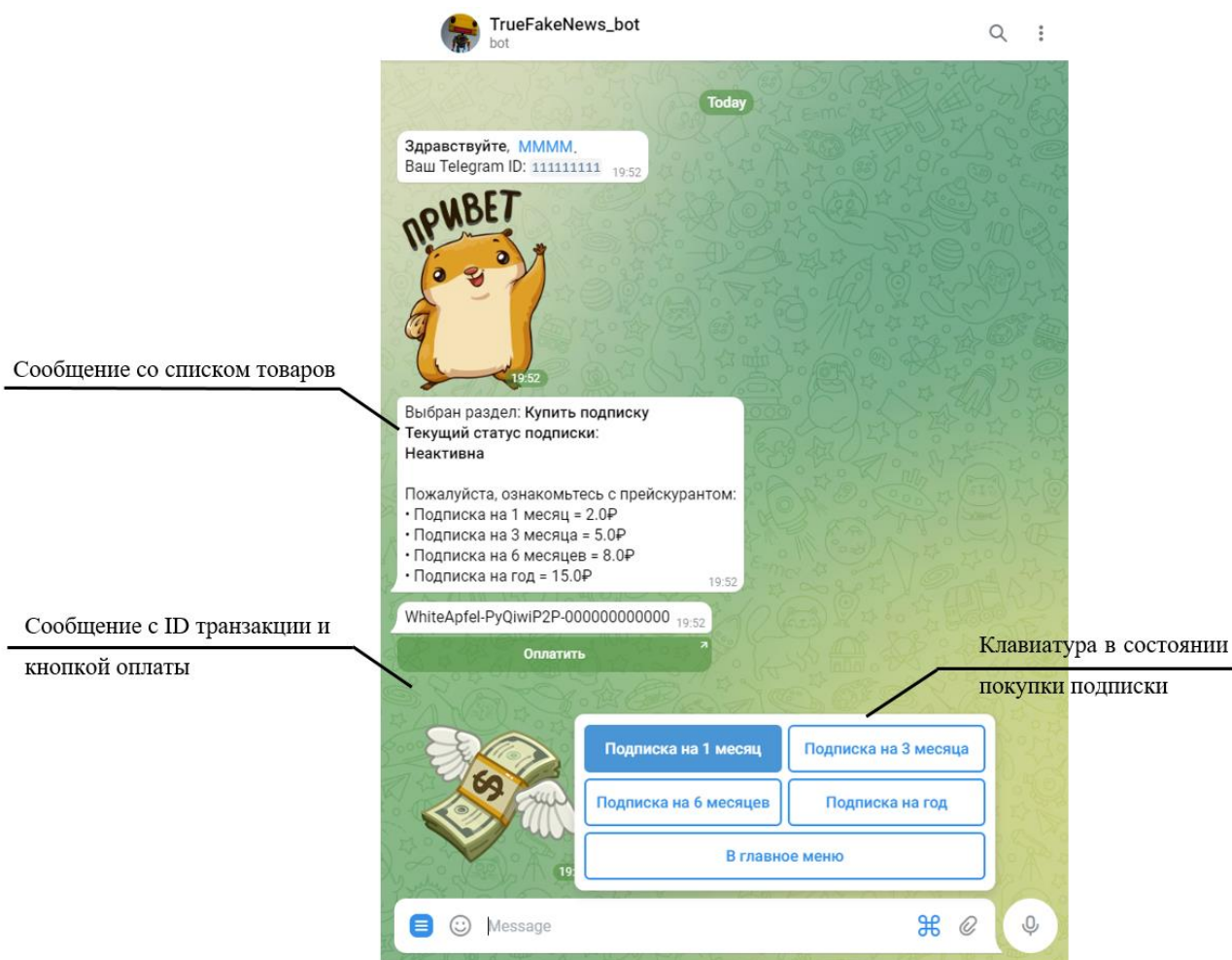


Рисунок 8 – Экранная форма «Страница покупки подписки»

Таким образом, на основе графа состояний интерфейса, графа абстрактного диалога и с учётом психофизических особенностей человека, связанных с восприятием, запоминанием и обработкой информации, был спроектирован пользовательский интерфейс программной системы отбора медицинских новостей.

2.3 Проектирование базы данных

2.3.1 Разработка инфологической модели базы данных

Инфологическая модель данных определяет совокупность объектов предметной области и демонстрирует связи между ними [1]. В результате

анализа предметной области были выявлены следующие объекты: пользователь, результат обработки информации, транзакция, товар, подписка. Информация, используемая системой, хранится в базе данных.

На основе полученных объектов была построена схема инфологической модели базы данных, представленная на рисунке 9.

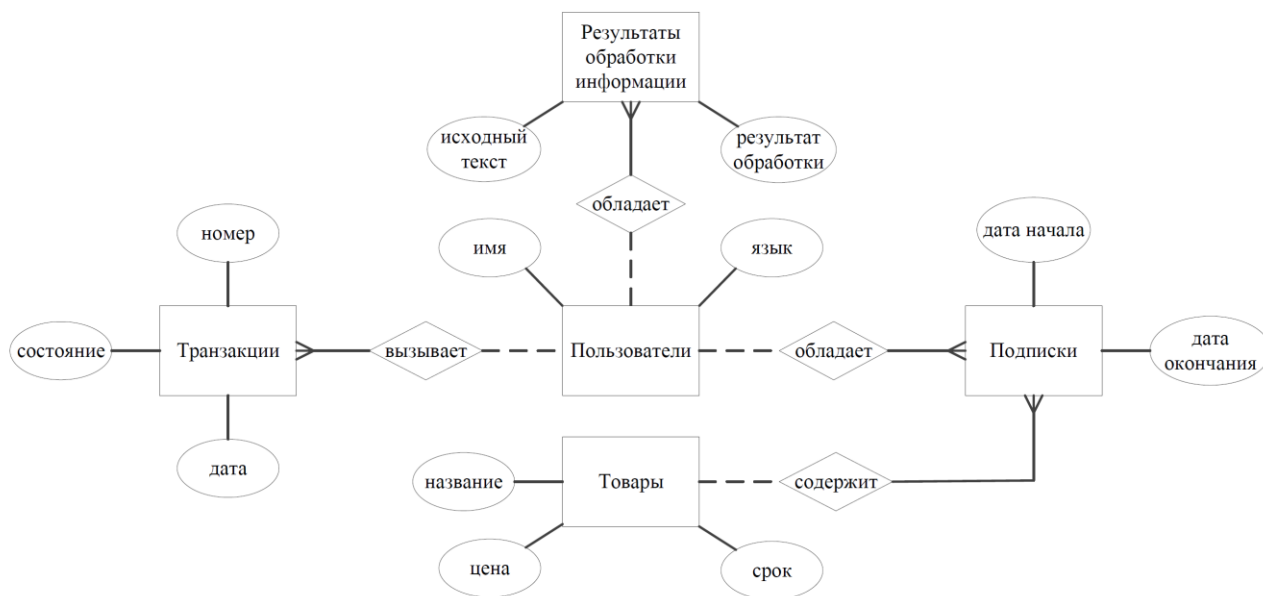


Рисунок 9 – Инфологическая модель базы данных

В базе данных хранится информация:

- о пользователях, которые когда-либо пользовались программной системой;
- о транзакциях на покупку товаров, которые завершились успешно или были отозваны;
- о результатах обработки информации, полученной от пользователя;
- о подписках, оформленных пользователями;
- о товарах, которые предлагаются к покупке.

При первом входе в систему информация о пользователе добавляется в соответствующую таблицу базы данных. Когда пользователь передаёт системе информацию для обработки, то эта информация и результат её обработки записывается в таблицу результатов обработки. Если пользователь решил купить подписку, то информация об открытой транзакции добавляется в таблицу транзакций. После успешной оплаты, информация о транзакции

обновляется, изменяя статус на «закрыта». Если же покупка была отменена, то статус транзакции изменяется на «отозвана». Когда пользователь оплатил покупку одной из подписок из таблицы товаров, то информация об этой записывается в таблицу подписок.

2.3.2 Разработка даталогической модели базы данных

Для хранения данных, с которыми работает разрабатываемая система, используется реляционная база и встраиваемая СУБД SQLite, которая обладает простотой подключения и настройки и быстрым выполнением запросов и команд [4].

На рисунке 10 приведена даталогическая модель базы данных.

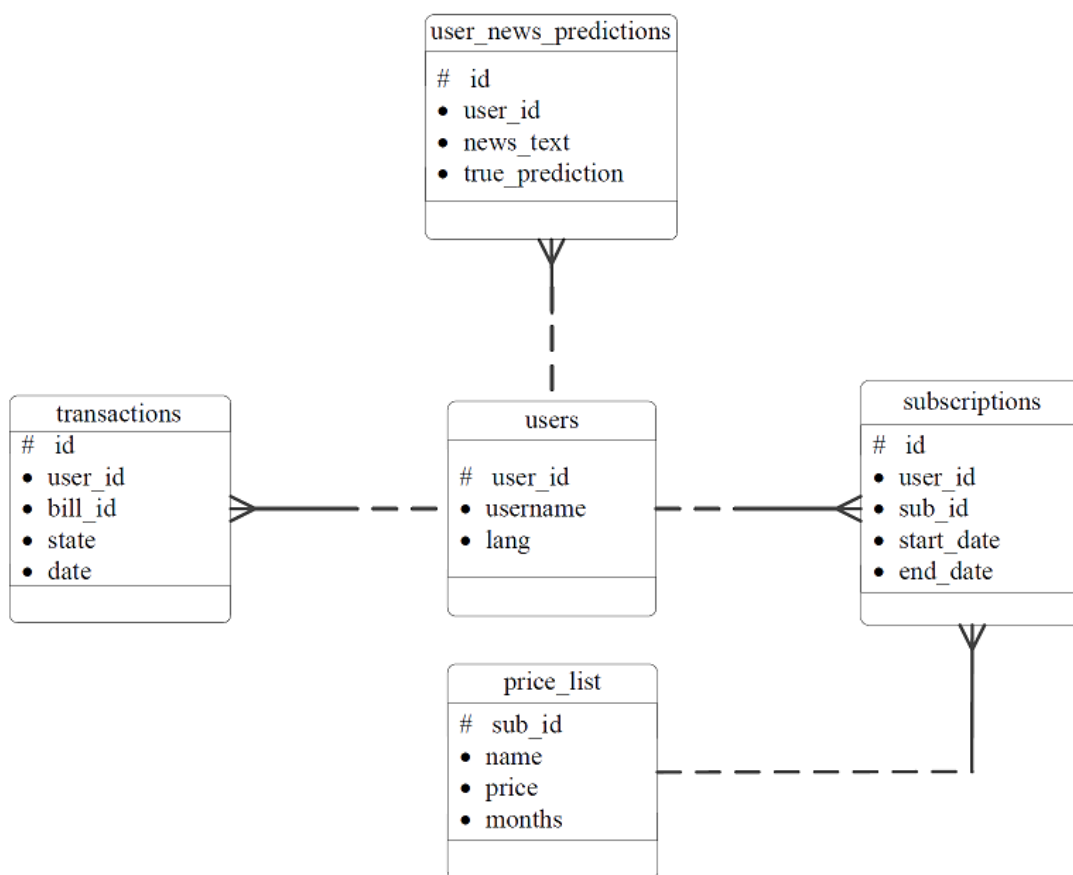


Рисунок 10 – Даталогическая модель базы данных

Таким образом, на основе указанных выше таблиц и схем была разработана база данных программной системы отбора медицинских новостей.

2.4 Разработка диаграммы компоновки программной системы

Процесс проектирования физической структуры программного продукта не может обойтись без диаграммы компоновки, которая показывает, из каких частей состоит система и как эти части взаимодействуют между собой. Эта диаграмма позволяет получить высокоуровневое представление о компонентах системы – физических заменяемых частях программного продукта, которые соответствуют некоторому набору интерфейсов и обеспечивают их реализацию.

Исходя из количества компонентов программного обеспечения, которые могут быть повторно использованы и представляют собой отдельную и самостоятельную часть, была получена диаграмма компоновки разрабатываемой системы (см. рисунок 11).

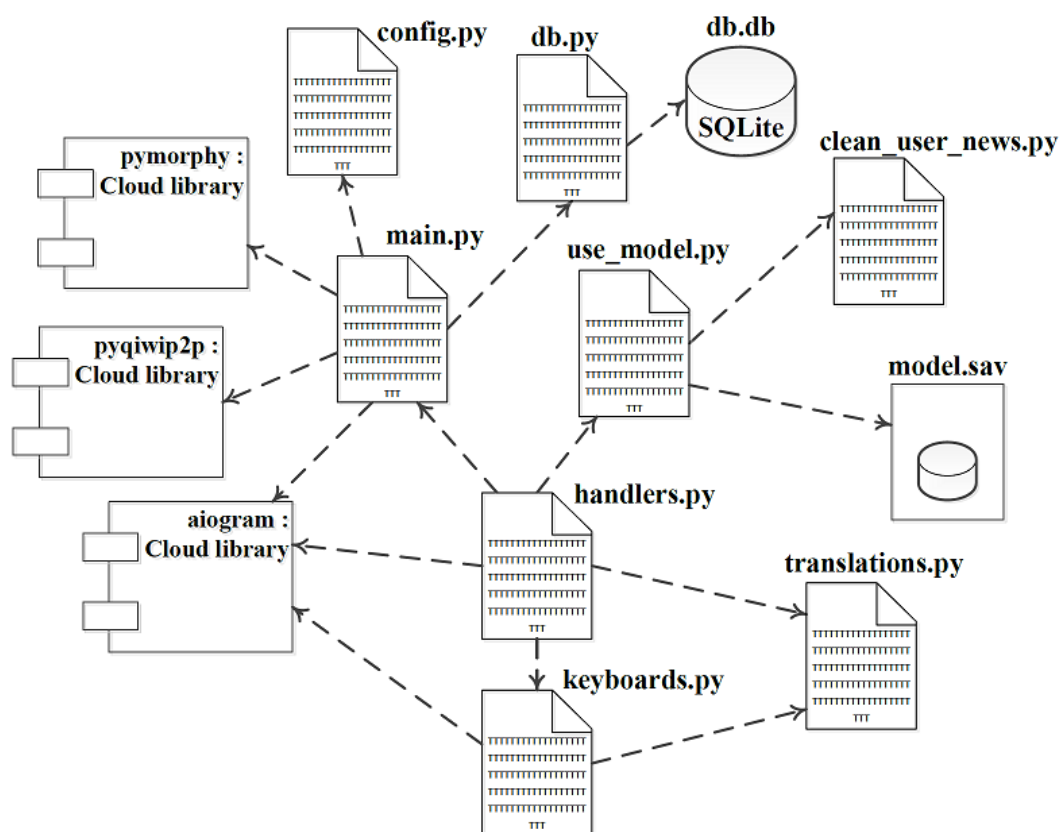


Рисунок 11 – Диаграмма компоновки программного продукта

Диаграмма компоновки программной системы отбора медицинских новостей содержит:

- девять файлов: **main.py** (файл Python, представляющий собой главный файл системы), **config.py** (файл Python, представляющий собой

конфигурационный файл системы), `db.py` (файл Python, представляющий собой файл с описанием методов для работы с базой данных), `handlers.py` (файл Python, представляющий собой файл с методами обработки действий пользователя), `keyboard.py` (файл Python, представляющий собой файл с реализацией различных вариантов клавиатур системы), `translations.py` (файл Python, предназначенный для реализации локализации системы), `use_model.py` (файл Python, предназначенный для взаимодействия с классификатором), `clean_user_news.py` (файл Python, предназначенный для предобработки текста, полученного от пользователя), `model.sav` (файл с расширением резервного сохранения, предназначенный для хранения настроенного классификатора);

- три библиотеки: `pymorphy` (библиотека морфологического анализатора русских текстов), `pyqiwip2p` (библиотека для доступа к API платёжной системы Qiwi), `aiogram` (библиотека для взаимодействия с Telegram API);

- одну базу данных `db.db` (база данных SQLite).

ЗАКЛЮЧЕНИЕ

В ходе прохождения преддипломной практики был произведён анализ предметной области разрабатываемого программного продукта, выбраны язык разработки, библиотеки и база данных, которые необходимы для создания системы, спроектирован и реализован прототип программной системы отбора медицинских новостей, а также разработаны:

- диаграмма вариантов использования;
- схема структурная программного продукта;
- граф состояний интерфейса;
- граф абстрактных диалогов;
- инфологическая и даталогическая модели базы данных;
- диаграмма компоновки программного продукта.

Разработанный прототип предоставляет возможность передачи текстовой информации новости на вход системы для её обработки и получения результата с рекомендацией, можно ли доверять этому контенту, возможность смены языка работы системы, возможность перезагрузки и получения описания возможностей системы, а также возможность покупки подписки с целью обработки текстов длиной более 1500 символов.

В ходе преддипломной практики был приобретён опыт в разработке программной системы отбора медицинских новостей, изучены принципы работы библиотек Aiogram, PyMorphy2 и pyqiwip2p, а также получены навыки работы со встраиваемой СУБД SQLite.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Иванова Г.С. Технология программирования: учебник / Г.С. Иванова. – 3-е изд., стер. – Москва: КНОРУС, 2018. 336 с. (Бакалавриат).
2. Aiogram documentation. Режим доступа: <https://docs.aiogram.dev/en/latest/index.html> (дата обращения 08.02.2023).
3. Буч Г., Рамбо Д., Якобсон И. Язык UML. Руководство пользователя – 2-е изд. Пер. с англ. Мухин Н. – ДМК Пресс, 2006. 496 с.
4. SQLite – самая простая база данных, которая работает везде. Режим доступа: <https://thecode.media/sqlite/> (дата обращения 10.02.2023).