**Примерная форма и структура отчета по элементу практической подготовки[[1]](#footnote-1), заполняемого в электронной информационно-образовательной системе НИУ ВШЭ**

Федеральное государственное автономное образовательное учреждение

высшего образования

«Национальный исследовательский университет «Высшая школа экономики»

Факультет Магистерская Школа

Информационная Аналитика в Управлении Предприятием

*(название ОП)*

Магистратура

*(уровень образования)*

**38.04.05** «Бизнес-информатика»

*(Профиль/ специализация (если есть)*

**О Т Ч Е Т**

**по** *профессиональной / проектной / исследовательской* **практике**

*(указать вид практики)*

Проектная

*(тип практики (наименование ЭПП[[2]](#footnote-2))*

*Модель прогнозирования сердечных заболеваний*

*(если проект, название проекта)*

Выполнил студент гр. ИАУП-22

Кашицкий Кирилл Валерьевич

*(ФИО)*

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

*(подпись)*

**Проверил:**

*\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_*

*(должность, ФИО руководителя ЭПП)*

*\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_*

*(подпись)*

*\_\_\_\_\_\_\_\_\_\_\_\_\_\_*

*(дата)*

**Содержание**

[Общее описание проекта 3](#_Toc137589792)

[Описание работ над проектом 3](#_Toc137589793)

[1.1. Описание хода выполнения проектного задания 3](#_Toc137589794)

[1.2 Описание результатов проекта (продукта) 5](#_Toc137589795)

[1.3 Описание использованных в проекте способов и технологий 6](#_Toc137589796)

[1.4 Описание своей роли в проектной команде 8](#_Toc137589797)

[1.5 Описание отклонений и трудностей 15](#_Toc137589798)

[Заключение 17](#_Toc137589799)

[Оценка индивидуальных результатов выполнения проекта 17](#_Toc137589800)

[Результат проекта 19](#_Toc137589801)

# **Общее описание проекта**

Инициатор: Кафедра информационных технологий в бизнесе (Пермь)

Тип проекта: прикладной

Место работы по проекту: Высшая Школа Экономики, Кафедра информационных технологий в бизнесе, Пермь.

# **Описание работ над проектом**

## **Описание хода выполнения проектного задания**

Для выполнения проекта мы с коллегами объединились в группу, состоящую из 5 человек. Первым этапом было определение вектора нашего проекта. Коллегиально было принято решение сосредоточиться на построении модели машинного обучения. Исследовав возможные темы для прогнозирования, выбор пал на вычисление вероятности возникновения сердечно-сосудистых заболеваний. Тема актуальна на сегодняшний день, поскольку сердечно-сосудистые заболевания, такие как ишемическая болезнь сердца, инфаркт миокарда и гипертония, являются одной из основных причин смертности по всему миру. Имея точную модель прогнозирования, медицинские специалисты могут идентифицировать рисковые группы и принимать меры по предотвращению и лечению заболеваний. С помощью интернет-ресурсаKaggle была найдена база данных случаев сердечно-сосудистых заболеваний, в которой содержится информация о состоянии здоровья человека и результат наличия или отсутствия заболевания. На данной базе будет основано обучение будущей модели. Чтобы проект являлся продуктом реальным для пользования было принято решение разработать чат-бота для Telegram и интегрировать в него модель машинного обучения.

Вторым организационным шагом являлось распределение ролей проекта по участникам группы. Мы выделили основные роли продуктовой команды и распределили их по навыкам и сильным сторонам каждого участника группы. Наш ролевой состав:

* Backenddeveloper- Михаил Кукаркин
* ML Engineer - Кашицкий Кирилл
* Project Manager - Завгородних Екатерина
* Аналитик - Стрекаловская Елена
* QA Engineer - Сырвачева Екатерина

Следующим этапом было планирование задач по реализации проекта и определение зон ответственности. Мы выделили следующие задачи и ответственные роли:

1. Подготовить среду для коммуникаций, составить график статусов по процессу работы, подготовить план график сроков реализации проекта - Ответственный Project Manager.
2. Подготовить базу данных возникновения сердечно-сосудистых заболеваний для модели, разработать и обучить модель машинного обучения на входных данных на языке python - Ответственный ML engineer.
3. Протестировать обученную модель на тестовых данных - Ответственный QA Engineer.
4. Спроектировать схему логики чат-бота, интегрируемого с моделью машинного обучения с помощью Figma - Ответственный Аналитик.
5. Разработать чат-бота для Telegram на основе схемы логики бота - Ответственный Backenddeveloper.
6. Провести тестирование чат-бота на наличие багов - Ответственный QA Engineer.
7. Опубликовать чат-бота в Telegram - Ответственный Backenddeveloper.
8. Подготовить проектную документацию: отчет о проекте, проектное предложение, задание на разработку, оценочный лист - Ответственный Project Manager, но в процессе участвовал каждый член команды.

После распределения ролей, зон ответственности и формулировки задач мы приступили к реализации самого проекта. В ходе работы проходили статусы для обсуждения прогресса и постановки следующих шагов, а также сроков. Разработка началась с реализации модели машинного обучения ML Engineer. После того, как был определен набор полей для модели бизнес-аналитик приступил к проработке схемы бизнес-логики чат-бота в Figma. Для оценки трудозатрат была собрана общая встреча, где каждый участник команды оценил будущую работу в часах, после чего Project manager составил план график всего проекта. Формализованные и установленные сроки помогли придать команде рабочий настрой и чувство коллективной ответственности. По завершению разработки модели QA Engineer занимался тестированием модели на уязвимости, качество прогноза и корректности поведения. Здесь процесс коммуникации с ML Engineer и QA Engineer был выстроен с помощью группового чата, куда тестировщик отправлял найденные баги по шаблону, а разработчик модели МА отписывался по готовности исправленных багов. Параллельно с этим Project Manager приступил к оформлению документации по проекту. Далее была собрана общая встреча с Project Manager, аналитиком, Backend Developer для обсуждения схемы бизнес-логики чат-бота, после чего разработчик приступил к реализации чат-бота для Telegram, интегрируемого с моделью машинного обучения. По завершению разработки к процессу подключился QA Engineer, который тестировал бота на соответствие логики схемы, корректность поведения и ошибки. Коммуникация с разработчиком также осуществлялась через групповой чат. После исправления всех багов аналитик приступил к закрывающему документу - Инструкции для пользователя. Инструкция также оформлялась в Figma. Когда вся разработка была завершена, Project Manager совместно с другими участниками команды оформляли документы для сдачи проекта.

## **Описание результатов проекта (продукта)**

Результатом проекта является разработанный и реализованный чат-бот для платформы Telegram. Чат-бот представляет собой программное приложение, способное взаимодействовать с пользователями через мессенджер Telegram. Чат-бот предназначен для использования в медицинских целях и помогает оценить риск возникновения сердечно-сосудистых заболеваний на основе предоставленных данных.

Основные характеристики и функциональность чат-бота включают:

1. Интеграция с Telegram: Чат-бот взаимодействует с пользователями через Telegram, используя API Telegram для отправки и получения сообщений.
2. Автоматическое общение: Чат-бот обладает возможностью автоматического общения с пользователями на основе заранее определенных правил и алгоритмов. Он способен отвечать на вопросы, предоставлять информацию на основе предопределенных сценариев.
3. Пользовательский интерфейс: Чат-бот предоставляет удобный и интуитивно понятный пользовательский интерфейс внутри Telegram, позволяя пользователям взаимодействовать с ботом.
4. Ввод и обработка данных: Чат-бот позволяет пользователям вводить необходимую информацию, такую как возраст, пол, наличие факторов риска и другие медицинские показатели.
5. Анализ и предсказание: Чат-бот использует передовые методы машинного обучения и аналитики данных для осуществления анализа и предсказания риска возникновения сердечных заболеваний у пользователя. Он обрабатывает введенные данные и применяет сложные математические модели и алгоритмы, обученные на основе исследований и медицинских рекомендаций, чтобы предсказать вероятность развития заболевания.
6. Вывод результатов: Чат-бот отображает результаты анализа в понятной форме для пользователя. Он может предоставить вероятность сердечных заболеваний в виде числа или процента, а также предоставить пояснение.

## **Описание использованных в проекте способов и технологий**

Для проекта использовались различные методы и способы разработки модели МА и чат-бота для прогнозирования сердечно-сосудистых заболеваний.

1. Сбор данных: первоначально нам потребовалось собрать данные, необходимые для обучения модели. Данные искались на сайте Kaggle. Найденная база содержала информацию о состоянии здоровья людей и заключение о наличии или отсутствии сердечно-сосудистых заболеваний.
2. Предварительная обработка данных: перед обучением модели, данные требовали предварительную обработку. Это включает в себя удаление выбросов, заполнение пропущенных значений, нормализацию или стандартизацию данных.
3. Выбор модели: выбор подходящей модели машинного обучения является важным шагом. В нашем случае, с учетом прогнозирования сердечно-сосудистых заболеваний, мы выбирали из моделей, такие как логистическая регрессия, случайный лес, градиентный бустинг или нейронные сети.
4. Обучение модели: после выбора модели необходимо было обучить ее на доступных данных. Обучение включает в себя разделение данных на обучающую и тестовую выборки, настройку гиперпараметров модели, выполнение процесса обучения и проверку качества модели на тестовых данных.
5. Интеграция модели с чат-ботом: после разработки и обучения модели машинного обучения, мы интегрировали ее с чат-ботом. Чат-бот получает входные данные от пользователей и передает их в модель для прогнозирования вероятности развития сердечно-сосудистых заболеваний. Модель возвращает прогнозы, которые чат-бот отображает пользователям.
6. Тестирование и оптимизация: важно было провести тестирование и оценку производительности нашей модели и чат-бота. Это позволило определить точность прогнозов модели, а также эффективность и удобство использования чат-бота.

Если рассматривать методы, которые использовались Project Manager в ходе проекта, то можно выделить следующий перечень:

1. Планирование проекта: Руководитель проекта (Project Manager) разрабатывал план проекта, определял задачи, ресурсы, сроки, а также последовательность действий и зависимости между задачами.
2. Управление командой: Project Manager координировал работу команды, назначал задачи, отслеживал прогресс и обеспечивал соблюдение сроков. Также Руководитель проекта организовывал регулярные статусы и общался с членами команды для обмена информацией и решения проблем.
3. Управление коммуникациями: Руководитель проекта обеспечивал эффективное взаимодействие с заинтересованными сторонами проекта, включая команду разработчиков, бизнес-аналитика, тестировщика. Руководитель проекта организовывал и контролировал взаимодействие участников команды.
4. Контроль качества: Project Manager следил за качеством работы. Все выполненные задачи проверялись Руководителем проекта для оценки полноты и корректности реализации. Также руководитель организовывал тестирование и проверку результатов работы команды перед внедрением модели и чат-бота.

Аналитик в свою очередь использовал такие методы как: анализ требований. Это первоначальный этап для подбора данных и проработки логики чат-бота. Также аналитик моделировал бизнес-процесс получения пользователем прогноза вероятности риска сердечно-сосудистых заболеваний по входным данным с помощью схемы логики чат-бота.

Для реализации проекта использовались различные инструменты и технологии каждым членом команды, перечень которых представлен ниже:

1. Project Manager оформлялпланграфикв MS Project. Для распределения задач и сроков MS Project является удобным сервисом, который широко применяется в IT компаниях.
2. Аналитик использовал сервис Figma для отрисовки интерфейсов чат-бота.
3. ML Engineer при разработке модели машинного обучения использовал язык программирования Python, фреймворк Tensorflow, а также библиотеки Pandas, Keras.
4. Backend Developer при разработки бота для чата “Telegram” использовал открытую библиотеку “python-telegram-bot” на языке программирования Python.
5. QA Engineer для проверки функциональности и поведения телеграм бота в реальном времени использовал чат-тестирование.

## **Описание своей роли в проектной команде**

Описание задач каждого члена команды будет представлено по блокам в зависимости от роли.

**Project Manager:**

Первоначальным шагом для начала работы было определение места для коммуникаций по проекту. Мною был создан групповой чат в Telegram, где на протяжении всей реализации проекта шло взаимодействие между участниками команды. Для коммуникации по тестированию был создан отдельный чат, куда отправлялись баги. Первым этапом была постановка и формулирование задач для каждого члена команды, после чего мною собрана со всеми участниками команды для обсуждения зоны ответственности по каждой задаче проекта. Помимо организаций встреч моей ответственностью было оформление мемо с итогами встречи и отправка его в чат. Мемо — это инструмент для фиксирования выводов встречи и предстоящих задач с указанием ответственных. После получения данных от QA Engineer на набор полей для модели машинного обучения была собрана встреча с участием аналитика и QA Engineer. На встрече я поставила задачу на проработку схемы логики чат-бота в Figma. Схема была мною согласована перед тем, как начать по ней работы. Следующим этапом была оценка трудозатрат на реализацию продукта, для которой также была проведена встреча со всеми участниками команды. Каждый участник давал оценку задач, оценка длительности разработки чат-бота происходила на основе схемы бизнес-логики чат-бота. Мною был составлен план график работ в MS Project. Далее было решено организовать еженедельные статусы по понедельникам в 19:00 для предоставления результатов работ. На статусе обсуждались выполненные задачи, возникшие проблемы, а также строился план работ на будущую неделю. Совместно с аналитиком и QA Engineer мы определили сценарии для тестирования модели и чат-бота. Также в мои обязанности входило консультирование каждого члена команды при появлении вопросов. Параллельно с аналитикой, тестированием и разработкой моей задачей стояло написание проектной документации. При описании своей роли в проекте привлекались все участники команды, также как и при техническом описании разработки.

**Аналитик**:

В роли аналитика передо мной стояли следующие задачи: описать алгоритм работы бота, который занимается прогнозированием вероятности сердечно-сосудистых заболеваний. Для начала работы были определены цели бота, функциональные и нефункциональные требования, которые должны описывать назначение программы. Следующий шаг включал в себя представление наглядного алгоритма: используя Figma спроектировать схему логики чат-бота, интегрируемого с моделью машинного обучения.

Работа над проектом с командой началась с участия во встречах для обсуждения модели. Это помогло лучше понять, какие данные будут использоваться и какие вопросы может задавать чат-бот пользователям.

Одной из первых задач являлось составление полного чек-листа функций бота. Был сформирован список функциональных возможностей бота:

* Распознавание и понимание сообщений: Бот должен быть способен распознавать и понимать сообщения, отправленные пользователями. Это включает в себя способность обрабатывать текстовые вводы и извлекать смысловую информацию из них. Также бот должен иметь возможность запрашивать дополнительные данные у пользователя для более точного прогнозирования вероятности заболевания.
* Предоставление информации: Бот должен быть способен предоставлять пользователю информацию, связанную с прогнозированием вероятности сердечно-сосудистых заболеваний.
* Интеграция с моделью машинного обучения: Бот должен быть интегрирован с моделью машинного обучения, чтобы осуществлять прогнозирование на основе полученных данных. Он должен отправлять запросы на модель и получать от нее результаты.
* Управление ошибками и исключениями: Бот должен быть способен обрабатывать ошибки и исключения, которые могут возникнуть в процессе работы. Например, если пользователь предоставляет некорректные данные или если модель не может выполнить прогноз.

Нефункциональные требования:

* Производительность: Бот должен обеспечивать быструю и отзывчивую работу, чтобы пользователи получали ответы на свои запросы в разумное время.
* Надежность: Бот должен быть стабильным и надежным, чтобы минимизировать возможность сбоев и ошибок.
* Безопасность: Бот должен обеспечивать безопасность данных пользователей, сохраняя их конфиденциальность.
* Масштабируемость: Бот должен быть способен масштабироваться и обрабатывать большое количество одновременных запросов без потери производительности.

Схема логики чат-бота представляет собой визуальное представление последовательности диалога между ботом и пользователем, а также определяет логику переходов между состояниями бота. На основе полученной со встреч информации я разработала общую структуру диалога. Определилапоследовательность вопросов и ответов между ботом и пользователем.

Разделила диалог на логические блоки, такие как:

* приветствие: начальное состояние бота, где он приветствует пользователя и предлагает помощь. Пользователь может выбрать или ввести команду: прогноз (получение оценки риска возникновения заболевания) или справка (для дополнительной информации);
* получение информации: бот запрашивает у пользователя необходимую информацию, связанную с рисками для прогнозирования вероятности сердечно-сосудистых заболеваний: возраст, пол, тип болей, давление и т.д;
* обработка запросов: бот анализирует полученные данные пользователя и передает их в модель машинного обучения для прогнозирования вероятности сердечно-сосудистых заболеваний. Модель возвращает результат, который используется ботом для принятия решения.
* завершение диалога: Бот формирует ответ на основе результатов модели и отправляет его пользователю.

С использованием инструмента Figma я создала визуальную схему, отображающую структуру диалога и логику переходов. Разместила блоки, представляющие состояния чат-бота, и соединила их стрелками для обозначения переходов. Обратила внимание на понятность и логичность схемы для удобства понимания.

В процессе разработки схемы логики чат-бота я консультировала QA Engineer, помогая понять логику взаимодействия бота с пользователями и ответы на различные сценарии. Объясняла, какие ответы и действия ожидаются от бота на основе схемы.

Также необходимо было дать пошаговую инструкцию для бота, чтобы быстро и эффективно рассказать клиенту, какие вопросы может задавать пользователь и какие ответы может получить. Я учла возможные варианты ответов и предусмотрела реакцию бота на них.

Для разработчиков чат-бота необходимо было обозначить точки взаимодействия с моделью машинного обучения. Это включало передачу введенных данных, текстовых запросов модели, получение ответов и дальнейшую обработку результатов.

**Backenddeveloper:**

Для реализации чат бота для Telegram в первую очередь необходимо зарегистрировать отдельный чат для бота, используя BotFather. Данный бот был разработан командой Telegram в качестве конструктора для ботов и сервиса для их модерации. После успешной регистрации бота на платформе необходимо задать индивидуальное имя боту. Для нашего проекта было выбрано имя hypertension\_classifier\_bot, а для отображения используется имя Health Check Bot. После завершения регистрации был получен уникальный токен, с помощью которого можно взаимодействовать с ботом через программный код.

Следующим шагом было изучение открытой библиотеки python-telegram-bot, которая предназначена для взаимодействия сервера с чатом Telegram. На этом этапе был разработан основной скрипт взаимодействия пользователя с сервером через отдельный чат в Telegram. Также для отслеживания текущего состояния программы были использованы различные значения. Каждое значение соответствует конкретному вопросу, ответ на который сейчас вводит или выбирает пользователь. Таким образом, каждое новое сообщения пользователя имеет индивидуальный контекст, согласно которому обрабатывается введенные значения. Помимо этого, для улучшения пользовательского опыта, было реализовано меню с описанием доступных команд.

На заключающемся этапе разработки бота было необходимо собрать все ответы пользователя и добавить их в качестве входных данных модель для классификации. Для хранения пользовательских данных используется локальное хранилище. Данное решение позволяет не терять контекст при долгом ответе между вопросами.

**ML engineer:**

Первым шагом идёт сбор данных о сердечно-сосудистых заболеваниях и объединение их в единый датасет, рис 1. Далее с полученными данными необходимо провести предобработку, исследовательский и статистический анализ. На обработанном датасете будет в дальнейшем обучаться модель. Следующим шагом было принято решение за итоговую модель взять нейронную сеть с прямой связью или многослойный перцептрон, как инструмент для бинарной классификации. Модель должна иметь 11 входных нейронов и принимать 11 числовых значений, на основании которого будет сделан прогноз, 2 скрытых слоя по 128 нейронов с функциями активации «ReLU» и одним выходным нейроном с функцией активации «Sigmoid», рис 2. Нейронная сеть должна обучаться в течение 100 поколений. Ключевой метрикой оценки качества модели необходимо считать Recall – полноту правильных ответов. По окончании обучения нейросети, обученная модель сохраняется в файл в формате .h5, который будет открыт и имплементирован в проект в Backend части. В перечень технологий для создания модели входят язык программирования Python, фреймворк Tensorflow, а также библиотеки Pandas, Keras.

**QA Engineer:**

На первом шаге выполнения работы были изучены требования к разрабатываемому телеграм-боту. Для этого был изучен список функциональных и нефункциональных требований, а также интерфейс, составленный аналитиком. Также были проведены совместные встречи с командой для понимания важных аспектов функционирования системы и обсуждения спорных моментов.

Основным документом для проверки бота была схема логики, которую использовали для сверки воркфлоу бота и соответствия формулировок. Данный документ содержал следующие основные пункты:

1. Проверка начального приветствия:
   * Проверка, что бот приветствует пользователя и предлагает ввести необходимые данные для прогноза, рис 3.
2. Проверка ввода данных:
   * Проверка правильности ввода каждого параметра, таких как возраст, пол, артериальное давление и другие факторы риска, экран 4, экран5, экран6.
   * Проверка обработки ошибочных вводов и вывода соответствующих сообщений пользователю для исправления, экран7.
3. Проверка логики прогноза:
   * Проверка соответствия прогнозируемой вероятности сердечных заболеваний введенным данным и ожидаемому результату, экран8, экран9.
4. Проверка навигации и продолжения работы:
   * Проверка продолжения работы бота после завершения прогноза, предоставляя возможность пользователю возможность повторного использования или других функций бота.

Тестирование проводилось путем последовательного ввода различных комбинаций данных и проверки соответствия результатов ожидаемым значениям и логике работы бота. В случае обнаружения несоответствий, backend разработчиком осуществлялась корректировка логики, и после этого проводилось повторное тестирование для проверки исправлений. Ошибки передавались разработчику по следующему шаблону:

* Шаги повторения: 1)… 2)… 3)… (Пользовательские действия, повторение которых приводит к ошибке)
* Ожидаемый результат:…
* Фактический результат:…

Далее были составлены тест кейсы, то есть сценарии для тестирования, которые включают в себя шаги выполнения пользовательских действий, а также ожидаемую реакцию системы. Тест кейсы играют важную роль в процессе разработки и помогают обнаружить ошибки и дефекты, улучшить качество продукта и удостовериться в его соответствии требованиям.

**Пример тест кейса для тестирования модели:**

*Описание: Проверяем, что бот корректно обрабатывает запрос пользователя и возвращает правильный прогноз вероятности сердечных заболеваний при нормальных входных данных.*

*Шаги:*

1. *Открыть чат с телеграм ботом.*
2. *Ввести команду или сообщение для получения прогноза вероятности сердечных заболеваний.*
3. *Ввести следующие входные данные:*
   1. *Возраст: 20 лет*
   2. *Пол: Мужской*
   3. *Тип боли в груди: асимптоматическая*
   4. *Давление: Нормальное (120/80 мм рт. ст.)*
   5. *Холестерин: Нормальный (180 мг/дл)*
   6. *Уровень сахара в крови натощак больше 120мг/дл: нет*
   7. *Введите, какую максимальную достигнутую частоту сердечных сокращений Вы зафиксировали: 68*
   8. *Зафиксирована ли при физической нагрузке стенокардия?: нет*
   9. *Введите значение депрессии сегмента ST, вызванной физической нагрузкой по сравнению с покоем:0,3*
   10. *Укажите результат исследования таламической гиперактивности: Отклонений не выявлено*
4. *Дождаться ответа от бота.*
5. *Проверить, что бот возвращает прогноз, что вероятность заболеваний низкая.*

*Ожидаемый результат: Бот корректно обрабатывает входные данные и возвращает правильный прогноз вероятности сердечных заболеваний, указывая низкую вероятность при нормальных показателях.*

*Дополнительные шаги:*

*6. Повторить шаги 1-5 с различными комбинациями входных данных, чтобы проверить корректность работы бота в других сценариях.*

**Пример тест кейса для проверки введенных данных:**

1. *Открыть чат с ботом*
2. *Выбрать команду для прогноза вероятности заболевания*
3. *Ввести данные о возрасте: Двадцать лет*

*Ожидаемый результат: Бот отображает ошибку в формате: Неверный формат данных, введите число.*

После запуска тестовой версии бота было проведено тестирование по составленным тестовым сценариям. Было проверено, что бот правильно реагирует на ввод пользователя, обрабатывает данные, генерирует прогнозы и предоставляет пользователю корректную информацию. Дополнительно было проведено тестирование на необработанные ошибки - ввод неверного формата данных, нажатие неверных кнопок. После данного тестирования было обнаружено несколько ошибок, а именно бот не учитывал неверный формат данных. Информация по ошибкам была передана Backenddeveloper, после чего было успешно проведено повторное тестирование.

В ходе работы также было поддержано активное сотрудничество с другими членами команды, включая разработчиков, Project manager и аналитика. Был осуществлен обмен информацией, обсуждение найденных дефектов и предложения по улучшению качества и функциональности бота.

## **Описание отклонений и трудностей**

**Project Manager:**

Единственным отклонением по запланированному процессу было то, что заявленная тема содержала в себе прогнозирование 2 аспектов: диабета и сердечно-сосудистых заболеваний. В ходе коллективного обсуждения было принято решение сосредоточиться только на прогнозировании сердечно-сосудистых заболеваний для более качественного результата в сжатые сроки. Из трудностей могу выделить организацию коммуникаций с участниками команды, поскольку у всех разный рабочий график. Несмотря на это коллективно удалось выбрать подходящее время для всех звонков, чтобы работа при этом не задерживалась.

**Backend Developer:**

Во время разработки чат бота возникли общие трудности, связанные особенностью взаимодействия с открытым API Telegram. Для работы с данным интерфейсом потребовалось изучение официальной документации.

Также основной проблемой при реализации бота является значительное ограничение архитектуры чата. Программный интерфейс Telegram устроен таким образом, что каждое сообщение от пользователя передает только текстовое значение. Для решения данной проблемы были реализованы переменные для отслеживания состояния, которые отражают конкретный шаг в диалоге между пользователем и сервером. Подробное описание решения представлено выше.

**QA Engineer:**

Возникшая трудность была связана с тестированием бота на разных операционных системах. Операционные системы, такие как Android, iOS и Windows, могут иметь разные ограничения, возможности и специфические настройки, которые могут повлиять на работу бота. Поэтому необходимо было убедиться, что бот функционирует правильно и предоставляет ожидаемый функционал на каждой поддерживаемой операционной системе. Необходимо было наличие устройств с каждой из перечисленных ОС для полного тестирования бота.

**Аналитик:**

Мне в роли аналитика потребовалось провести тщательное исследование и понимание данной предметной области, чтобы точно определить, какие цели должны быть достигнуты и какие функциональные и нефункциональные требования должны быть учтены. Выполнение поставленных задач протекало достаточно гладко, особых трудностей не возникло.

**ML Engineer:**

В ходе выполнения проекта в роли ML инженера возникли определённые трудности, которые были успешно нивелированы. Главной трудностью можно посчитать подбор оптимальной модели от базовой логистической регрессии к полносвязной нейронной сети.

Ещё одной сложностью было написание нейронной сети при помощи парадигмы ООП, потому что изначально алгоритм рассматривался, как класс, импортируемый в чат-бота. Далее было принято решение добавить в класс метод, сохраняющий обученную модель в файл с расширением .h5 и его дальнейшее использование в боте.

Следующей проблемой был подбор гипер параметров для нейронной сети: функции активации (изначально планировалось взять пороговые функции, но это ухудшало метрики оценки), количество скрытых слоёв и количество нейронов в слоях.

# **Заключение**

## **Оценка индивидуальных результатов выполнения проекта**

**Project Manager:**

Моими основными задачами были организация и контроль процесса реализации проекта. Эти задачи включали в себя полное участие на каждом этапе проекта, консультирование при возникновении вопросов и помощь в решение проблем. На мой взгляд главный показатель работы Project manager- это проект, сданный в срок, который соответствует всем заявленным требованиям. Проект можно считать успешным, поскольку все поставленные задачи были выполнены. В процессе работы над проектом удалось придать общекомандный дух, чувство ответственности каждому члену команды, а также вызвать доверие.

Ввиду опыта работы по ведению реальных IT проектов на работе данный проект стал средой для усовершенствования своих навыков. Было крайне полезно переложить рабочий опыт на учебный проект. Данный проект можно охарактеризовать как тренировку своих уже существующих компетенций, а именно постановка задач разработчикам, аналитикам и тестировщикам, организация тестирования, подготовка план графика, организация и проведение статусов, решение возникающих проблем и нивелирование рисков. Совершенно точно можно сказать, что навык коммуникации и организации комфортной рабочей среды усилился, поскольку на работе есть давно сплоченная команда, которая занимается подобными проектами регулярно, а для нашей команды в таком составе это был первый опыт.

Подводя итог, проект был крайне полезен мне как специалисту, потому что он ставит в непривычную среду: новая команда, отсутствие привычных инструментов для работы (Jira), разный опыт каждого члена команды. Тем не менее проект получился приближенным к реальной работе в IT компаниях и был полезен всем участникам команды.

**Аналитик**:

В результате выполнения проекта бота для прогнозирования вероятности сердечно-сосудистых заболеваний, я успешно справилась с поставленными задачами. Были разработаны функциональные и нефункциональные требования, соответствующие назначению программы, также была спроектирована схема логики чат-бота с использованием Figma, для представления наглядного алгоритма работы бота.

В ходе выполнения были продемонстрированы следующие компетенции:

* аналитические навыки: на основе проведенных с командой встреч была успешно выполнена задача анализа требований;
* технические навыки: для достижения поставленной цели были использованы современные инструменты проектирования, в результате была создана наглядная схема работы программы;
* коммуникационные навыки: в процессе работы над проектом все участники взаимодействовали и общались, проявляли способность слушать и выражать свои мысли и идеи.

Успешно выполненные поставленные задачи в этом проекте позволяют оценить возможности и перспективы дальнейшего развития в данной области и возможность эффективного выполнение подобных проектов в будущем.

**QA Engineer:**

В ходе выполнения проекта по реализации чат-бота для Telegram, предсказывающего риск сердечных заболеваний, я успешно справилась с поставленными задачами, применив свой опыт работы в IT проектах. Работая в роли аналитика ранее, мне было интересно попробовать себя в другой роли и взять на себя ответственность за качество разрабатываемого продукта.

Одной из ключевых компетенций, которую я применила в данном проекте, была способность анализировать и интерпретировать требования проекта. Активное взаимодействие с командой разработчиков и бизнес-аналитиком позволило мне полно и точно понять требования и функциональность бота. Это в свою очередь позволило мне создать релевантные тестовые сценарии, которые обеспечили полное покрытие функциональности приложения.

Кроме того, в данном проекте я смогла применить свои компетенции в области тестирования программного обеспечения. Благодаря предварительному исследованию основных принципов тестирования, методов и техник, я повысила уровень своих знаний, а также смогла провести анализ и успешно составить и выполнить тестовые сценарии.

В целом, мое участие в проекте позволило мне расширить свой кругозор и развить новые навыки в области тестирования программного обеспечения, что важно для моего профессионального роста и повышения эффективности работы в команде проекта.

**Backend Developer:**

Моей задачей в качестве основного разработчика была непосредственная реализация чат бота. В процессе работы я использовал свои навыки разработки программных продуктов. Во время работы я изучил архитектуру программного интерфейса Telegram и ознакомился с основными библиотеками на языке программирования Python.

В целом, участие в этом проекте позволило мне применить и улучшить свои навыки программирования и работы в команде. Я получил ценный опыт, который поможет мне в будущих проектах.

**ML Engineer:**

Моей задачей в качестве инженера по машинному обучению было создание модели, которая предсказывает вероятность сердечно-сосудистых заболеваний. В процессе работы я познакомился с библиотеками для анализа данных, а также для машинного и глубокого обучения.

В целом, участие в данном проекте позволило мне изучить основы базовых библиотек глубокого обучения, а также на практике познакомиться с полносвязными нейронными сетями.

## **Результат проекта**

Перед началом разработки и аналитики были составлен план график проекта, где были сформулированы основные этапы, зоны ответственности и сроки. План график представлен на Рисунок 1.

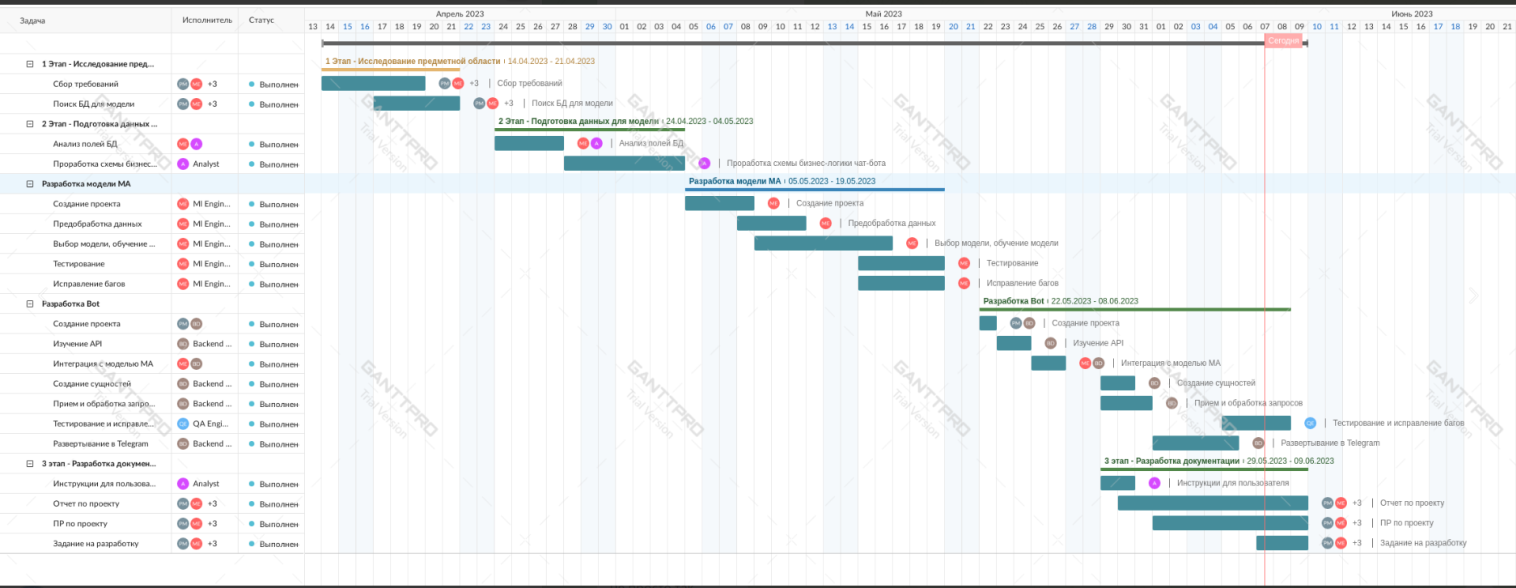


Рисунок 1 - План график проекта

В результате работы над проектом был составлен датасет, на котором проходило обучение модели. Часть данных представлена на Рисунок 2.

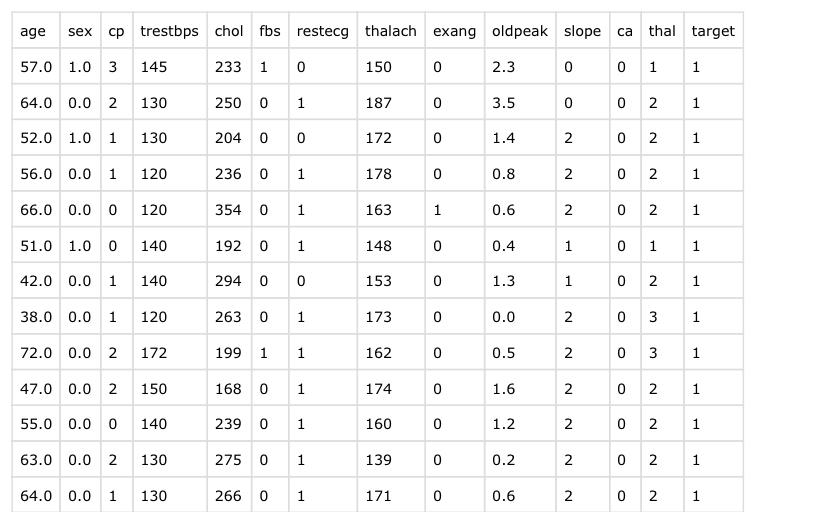


Рисунок 2 - Данные для обучения модели

Дополнительно была сформирована архитектура нейросети, которая определяет, как данные будут обрабатываться и преобразовываться внутри нейросети. Она включает в себя различные слои и их взаимосвязи, определяющие поток данных через нейросеть. Архитектура представлена на Рисунок 3и более подробно описана в разделе Описание своей роли в проектной команде.

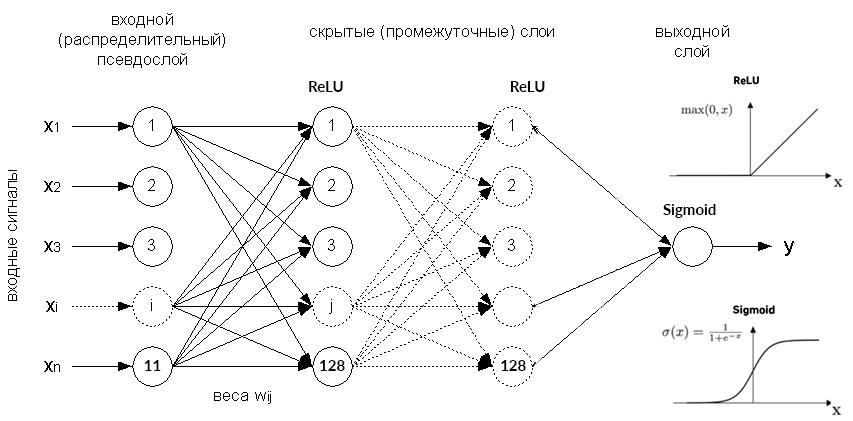


Рисунок 3 - Архитектура нейросети

Фрагмент кода для обучения и тестирования модели представлен на Рисунок 4.

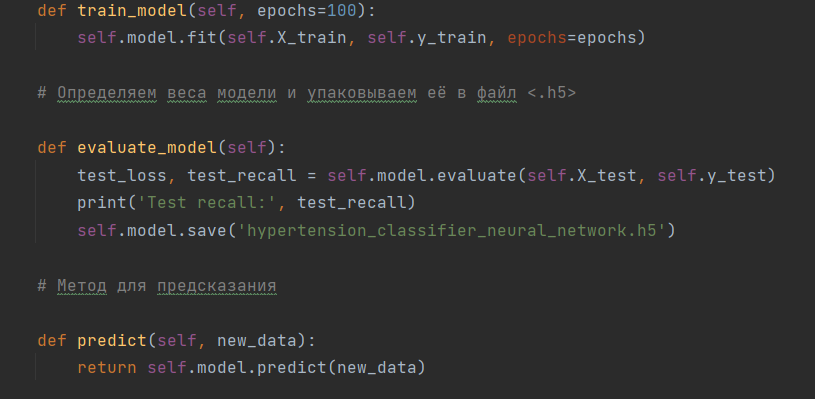


Рисунок 4 - Фрагмент кода модели машинного обучения

График RecalltoGenerations, представленный наРисунок 5, отражает изменение значения метрики recall (полноты) в зависимости от количества итераций в алгоритме обучения модели по 100 поколениям.

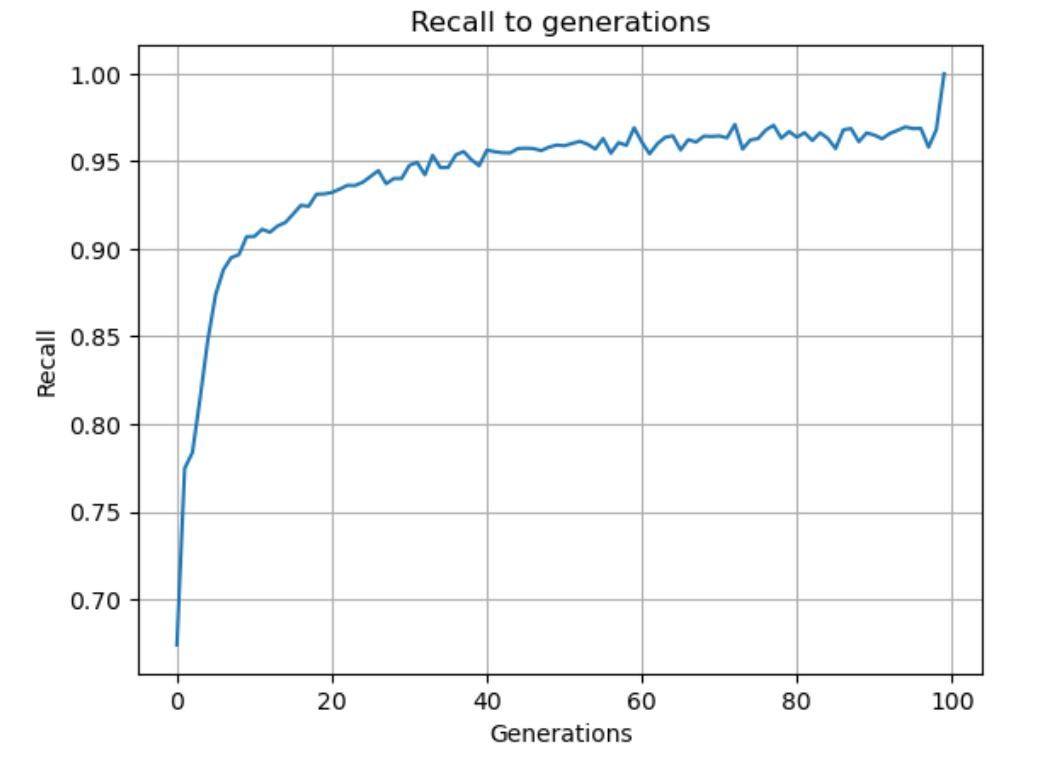


Рисунок 5 - График изменения метрики оценки производительности модели классификации

После разработки и обучения модели была спроектирована схема-логики чат-бота, которая показывала флоу взаимодействия бота с пользователем и формулировки сообщений. Часть экранов схемы представлена на рисунках 6 и 7.

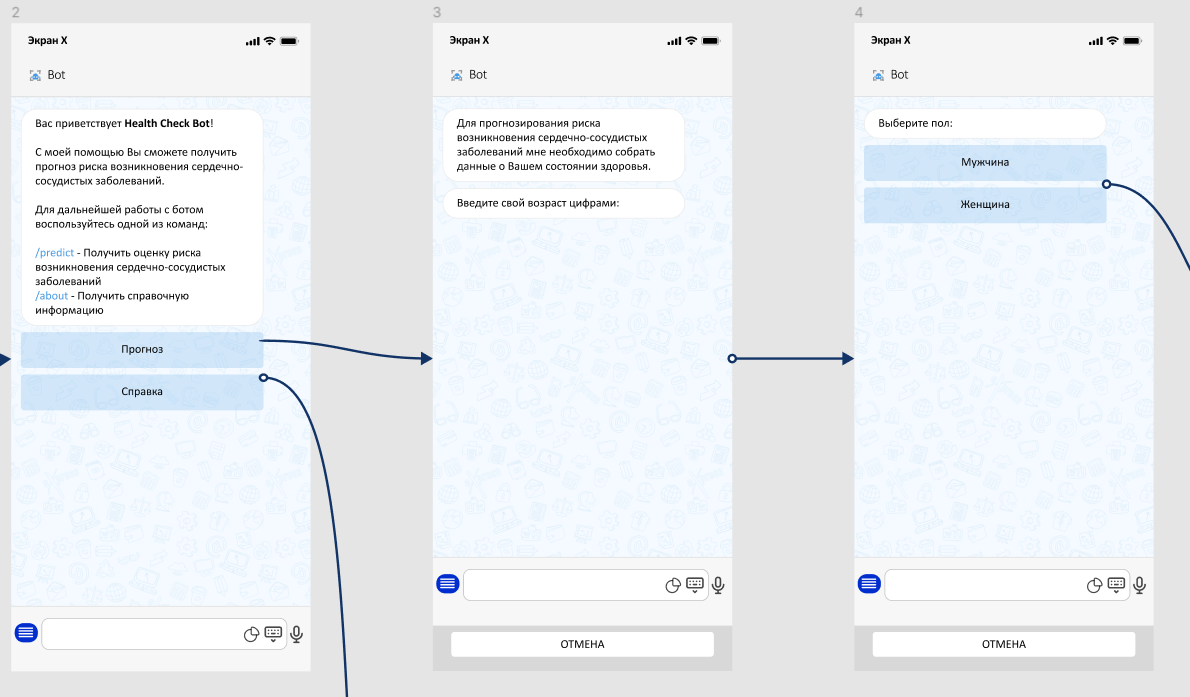


Рисунок 6 - Схема бизнес-логики чат-бота

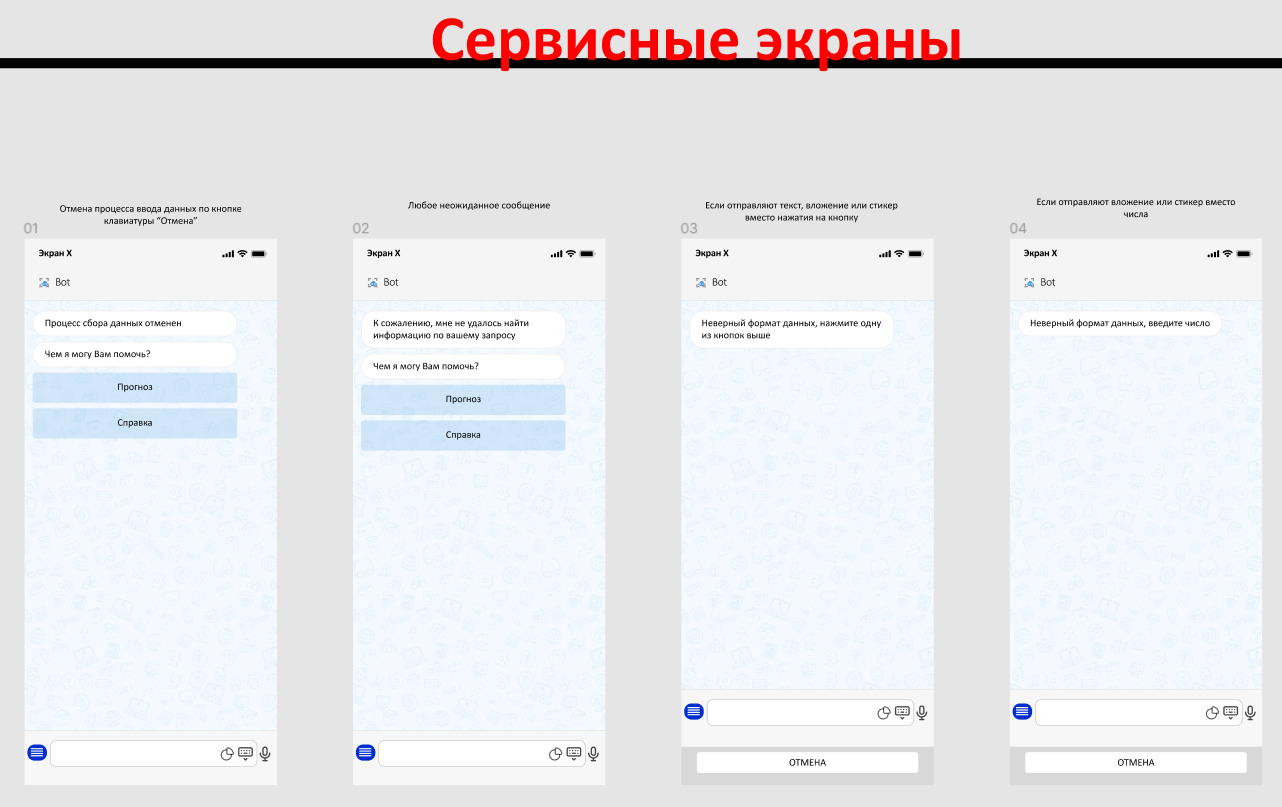


Рисунок 7 - Сервисные экраны схемы логики

После проработки схемы бизнес-логики Backend Developer приступил к разработке чат-бота, интегрируемого с моделью машинного обучения. Результаты работы чат-бота представлены ниже. Чат-бот выдает справку о своей функциональности и командах - Рисунок 8.

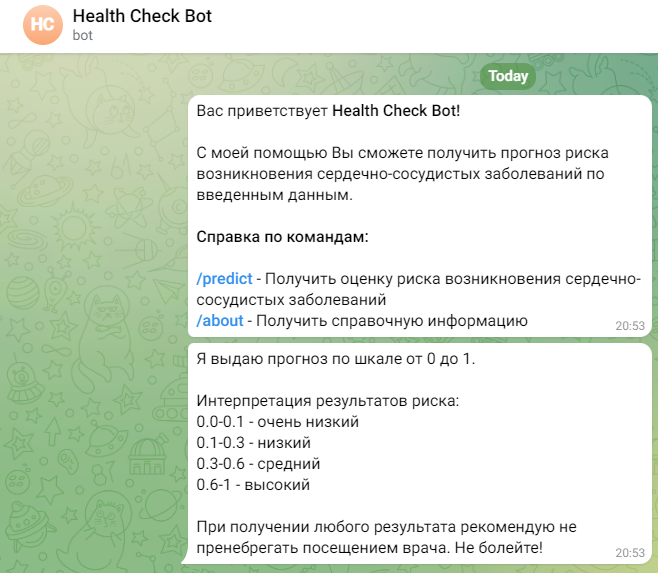


Рисунок 8 - Запуск бота в Telegram

По команде /predict или кнопке прогноз запускается процесс заполнения данных о здоровье пациента.

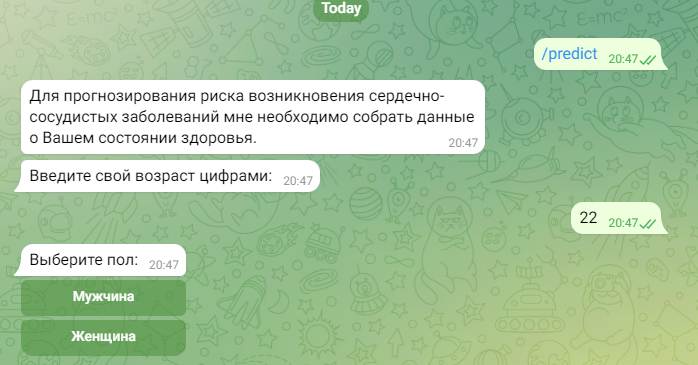


Рисунок 9 - Использование команды для получения прогноза

Каждое поле для модели заполняется через бот отдельным сообщением. Некоторые значения заполняются пользователем вручную, а также через кнопки действия.

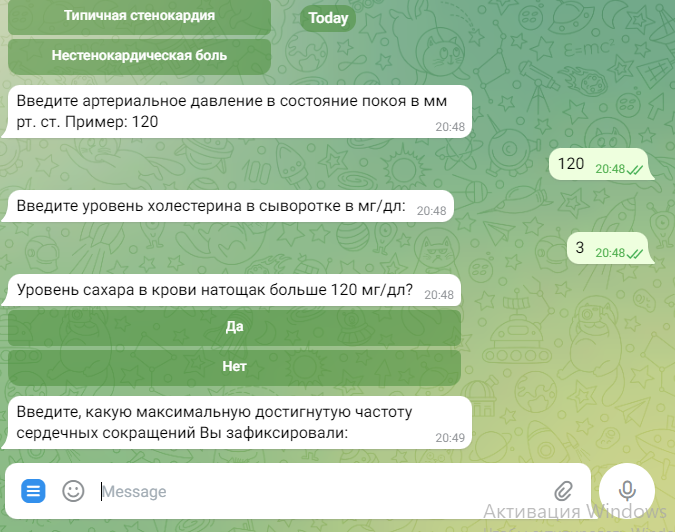


Рисунок 10 - Ввод данных для получения прогноза

Чат-бот обрабатывает неверные форматы ввода значений для сопровождения корректности заполнения данных о состоянии здоровья, требуемых для прогноза модели.



Рисунок 11 - Ввод неверного формата данных

По завершению заполнения данных о состоянии здоровья чат-бот отправляет запрос к модели на получение прогноза от 0 до 1.

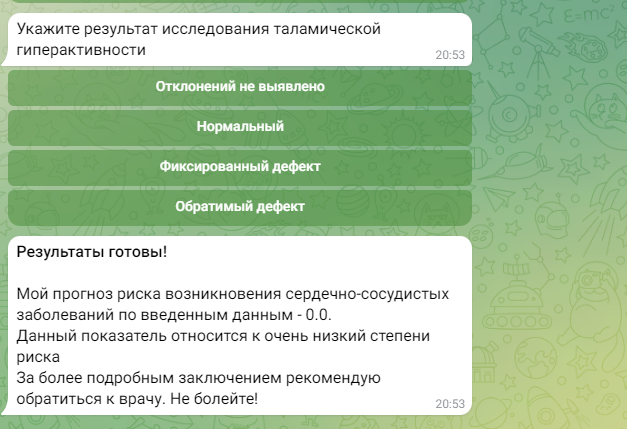


Рисунок 12 - Вывод результатов предсказания риска заболеваний

По шкале, описанной в справочной информации к боту, чат-бот определяет уровень риска возникновения сердечно-сосудистых заболеваний: очень низкий, низкий, средний, высокий.

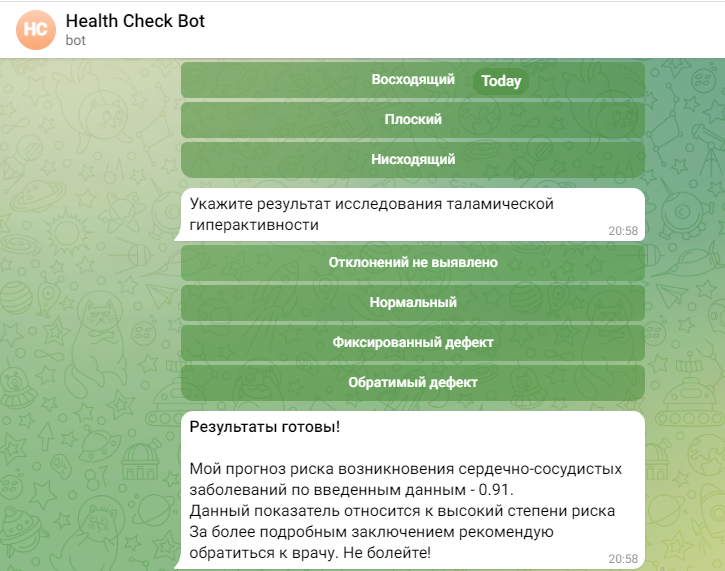


Рисунок 13 - Вывод результатов предсказания риска заболеваний

После запуска чат-бота было проведено тестирование по составленным тест-кейсам. Пример чек-листа с кейсами представлен на Рисунке 14.

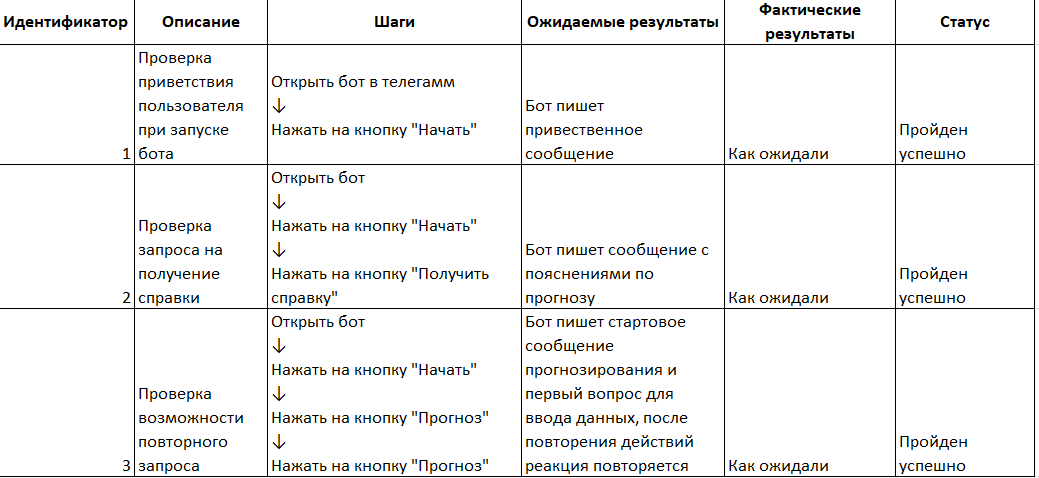


Рисунок 14 – Чек-лист тестирования

После тестирования чат-бота и исправления всех багов была подготовлена инструкция для пользователя по взаимодействию с чат-ботом. Инструкция представлена ниже на Рисунок 15.



Рисунок 15 - Инструкция для пользователя чат-бота

1. Для элементов практической подготовки в форме проектов или профессиональных практик. Оформление текста КР / ВКР описывается в приложении \_ Порядок оформления курсовых и выпускных квалификационных работ НИУ ВШЭ к настоящему Положению. [↑](#footnote-ref-1)
2. ЭПП – элемент практической подготовки по учебному плану образовательной программы. [↑](#footnote-ref-2)