**МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ**

**САРАТОВСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ**

КУРСОВАЯ РАБОТА

по дисциплине: «Современные технологии веб-программирования»

на тему: «Разработка прототипа мобильного приложения для оценки вероятности рецидивов заболеваний желудка»

Выполнил:

студент 3 курса, гр. Б1-ИВЧТ-31

Кузнецов А.А.

Руководитель:

к. т. н., доц.

Безруков А.И.

Саратов, 2024 г.

**Оглавление**

[Введение 3](#_Toc184035480)

[Концепция взаимодействия мобильного и десктопного приложения 4](#_Toc184035481)

[Система хранения данных 6](#_Toc184035482)

[Идентификация данных 10](#_Toc184035483)

[Передача уникального номера врачу 10](#_Toc184035484)

[Систематизация описания приложения 12](#_Toc184035485)

[Описание функций и структуры мобильного приложения 14](#_Toc184035486)

[1. User story 14](#_Toc184035487)

[2. Первый запуск приложения 14](#_Toc184035488)

[3. КНОПКИ И ОПИСАНИЕ КАЖДОГО ПРОЦЕССА 16](#_Toc184035489)

[3.1 Заполнение анкеты: 16](#_Toc184035490)

[3.2 Редактирование анкеты: 17](#_Toc184035491)

[4. Автоматические процессы: 17](#_Toc184035492)

[5. Диаграмма классов 17](#_Toc184035493)

[Заключение 20](#_Toc184035494)

[Список использованных источников 21](#_Toc184035495)

[Приложение 22](#_Toc184035496)

# **Введение**

(Нужно описать проблему (см. «Концепция проекта2»); описать концепцию двух приложений, обосновать необходимость разработки прототипа)

Концепцией проекта является обеспечение доступа молодым врачам к обширной базе клинических данных. Десктопное приложение будет ориентировано на подробный анализ случаев, позволяя пользователям изучать различные подходы к лечению и прогнозирование результатов на основе аналогичных клинических данных. Мобильное приложение даст возможность консультироваться и получать рекомендации в любое время и в любом месте, что особенно важно в условиях экстренной помощи.

Предоставление платформы для обмена знаниями между молодыми врачами и опытными хирургами станет важным аспектом проекта.

Целью курсовой работы является создание прототипа мобильного приложения, реализующего концепцию, позволяющего наглядно раскрыть суть идеи и проверить основные положения концепции на практике. По прототипу можно будет понять полезное действие приложения и его конек. Прототип — это важный этап перед стартом детального дизайна и программирования. Прототип интерфейса приложения помогает представить дизайн и логику приложения, а также протестировать идею. Это еще не настоящее полнофункциональное решение, он напичкан муляжами и заточен на работу только в определенных ситуациях и с определенном набором данных.

Разработка эффективных инструментов мобильного приложения для оценки вероятности рецидивов заболеваний желудка имеет решающее значение для оптимизации лечения и улучшения результатов лечения.

Задачи:

1. Разработать концепцию взаимодействия мобильного и десктопного приложения
2. Разработать децентрализованный подход к хранению данных о пациентах и всей сопутствующей информации, с возможностью сбора данных в облачном хранилище с различных устройств.
3. Разработать систему верификации данных, программа не должна принимать пациентов из непроверенных источников.
4. Сформировать справочную систему в прототипе системы
5. Интерфейс мобильного, показать его.(учесть поддержку российского аналога андроида) (https://www.comnews.ru/content/236295/2024-11-15/2024-w46/1007/rossiyskoe-oborudovanie-ne-ostanetsya-bez-zarubezhnykh-mikroskhem)

# **Концепция взаимодействия мобильного и десктопного приложения**

Десктопные и мобильные приложения — это два типа программного обеспечения, которые могут выполнять схожий функционал. Основное различие между ними заключается в том, что десктопные приложения устанавливаются на компьютер пользователя и работают на операционной системе Windows, тогда как мобильные на телефоны.

Десктопные приложения сейчас все больше становятся связаны со следующими тенденциями: облачными вычислениями, интеграцией с мобильными устройствами и веб-сервисами, а также развитием кроссплатформенных фреймворков. Это позволяет создавать десктопные приложения, которые взаимодействуют с облачными хранилищами данных, синхронизируются с мобильными приложениями, используют внешние API для получения информации.

Интеграция десктопного и мобильного приложений играет ключевую роль в достижении основной цели проекта – передаче опыта старшего поколения хирургов молодым специалистам. Это происходит следующим образом:

- Десктопное приложение может анализировать данные из базы знаний и выявлять закономерности в лечении различных типов пациентов. Эта аналитика может быть доступна через мобильное приложение. Это помогает молодым врачам получить более глубокое понимание болезни и эффективных методов лечения.

- Мобильное приложение может предоставлять доступ к клиническим случаям. Со временем это позволит создать ещё более точный и информативный инструмент для обучения и принятия решений.

Интеграция мобильного и десктопного приложений создает единую, постоянно обновляющуюся платформу для обучения и обмена опытом, ускоряя процесс передачи знаний между поколениями хирургов и повышая качество лечения пациентов с кровоточащими гастродуоденальными язвами.

## Описание процесса обмена данных с использованием внешнего хранилища данных

В разрабатываемой системе есть 2 типа приложений:

1. Десктопное.
2. Мобильное

Мобильное и десктопное приложения взаимодействуют друг с другом через облачное хранилище Яндекс Диск (рисунок 1), далее просто ЯД. Этот инструмент был выбран из-за ограниченного бюджета заказчика, поскольку ЯД предоставляет бесплатный и простой в использовании сервис для передачи данных.



Рисунок 1. Взаимодействие мобильного и десктопного приложений

## Описание задач, требующих обмен данных с другими устройствами

**Задачи, выполняемые в десктопном приложении:**

Загрузка на ЯД актуальной обучающей таблицы, необходимых данных для регистрации врача.

Скачивание обновленной обучающей таблицы, заполненных и загруженных на ЯД анкет пациентов.

**Задачи, выполняемые в мобильном приложении:**

Загрузка на ЯД данных о анкетированных пациентах с данного устройства на ЯД.

Скачивание обновленных данных о обучающей таблице с ЯД, данных необходимых для регистрации устройства в системе.

**Список основных взаимодействия приложений в системе с Яндекс Диском:**

1. После заполнения анкеты в мобильном приложении, постановки окончательного диагноза, согласия врача и пациента, прототип, при наличии интернета, будет собирать готовые анкеты в единый JSON файл и загружать его на ЯД.
2. Мобильное приложение просматривает дату последнего обновления на ЯД обучающей таблицы. При наличии новой версии обучающей таблицы, приложение загружает и использует новую версию.
3. При наличии новых файлов на ЯД с заполненными анкетами, десктопное приложение скачивает доступные файлы. Загружает данные из файлов в таблицу базы данных, далее БД, в таблицу Anket\_Val\_New. Когда данные из файла записаны в таблицу, файл из которого получены данные удаляется на ЯД.
4. Десктопное приложение отправляет измененную обучающую таблицу на ЯД.

Эта концепция взаимодействия обеспечивает синхронизацию данных между мобильным и десктопным приложениями, позволяет использование молодыми врачами опыта лечения пациентов с язвами, расширение обучающей таблицы для улучшения работы прогнозирования

# **Система хранения данных**

Ключевым аспектом успешного функционирования приложения является гибкая и масштабируемая система хранения данных, способная адаптироваться к изменениям в списке характеристик пациентов, используемых для оценки риска рецидива. Необходимость в такой гибкости обусловлена потенциальным расширением набора диагностических параметров и совершенствованием методик оценки вероятности рецидива. В связи с этим, архитектура системы хранения данных должна быть спроектирована таким образом, чтобы легко интегрировать новые характеристики без необходимости существенной перестройки всей базы данных.

## Логическая модель данных и ее описание

В системе хранения используются две таблицы с общими данными: пациентов и докторов.

В таблице Doctors указываются данные врача (ФИО, контактные данные), уникальный номер, который используется в передачи данных и т.д.

В таблице пациентов Anket содержатся необходимые данные о пациенте (ФИО, дата рождения, пол, дата заполнения анкеты и др.), сведения об окончательном диагнозе (при его наличии), id лечащего врача и т.д.

**Doctors**

**Anket**

**Char**

**Anket\_Val**

Doctor

Id\_Doctor

Expert

Id\_Char

**Char\_Val**

Code

Id\_Char

Id\_Anket

Id\_Anket

Id\_Char

Code

Value

Help

Help

Рисунок 2 Логическая модель базы данных

Чтобы была возможность добавления новых характеристик без изменения системы хранения используется следующая структура:

1. Общая таблица со всеми характеристиками, их минимальными и максимальными значениями и другими ограничениями. В этой же таблице будет возможно добавление новых характеристик по требованию врачей (без изменения структуры базы данных) (табл. Char)
2. Для категориальных и ранговых признаков добавлена таблица со списком значений для каждого признака. Набор: номер признака – номер значения – значение. (табл. Char\_Val)
3. Итоговая таблица, в которой собираются данные о пациентах. В нее записываются: номер анкеты, номер характеристики, значение характеристики ( поле Value – для числовых, поле Code – для категориальных и ранговых) Поле Code имеет внешний ключ в таблице Char\_Val.
4. Чтобы разделить пациентов на подходящих для обучающей таблицы или только переданных с телефонов существуют 2 таблицы одинаковой структуры: Anket\_Val\_Trust и Anket\_Val\_New. Когда эксперт проверит анкету пациента, то он в программе сможет перенести анкету пациента в обучающую таблицу с проверенными пациентами.

Value

Value

Code

Id\_Char

Code

Id\_Anket

Id\_Char

Id\_Anket

**Anket\_Val\_Trust**

**Anket\_Val\_New**

## Физическое описание структуры базы данных

Описание характеристик (таблица Char)

В данной таблице хранится полный список оцениваемых характеристик.

|  |  |  |
| --- | --- | --- |
| **Char** | | |
| Тип | Поле | Комментарий |
| Целое | Id\_Char |  |
| Строка | Characteristic | Название характеристики |
| Бинарное | Used | Используется в обучающей таблице |
| ENUM | Type\_Char | Тип характеристики |
| Число | L\_Level | Нижний уровень значений |
| Число | U\_Level | Верхний уровень значений |
| Число | Weight | Вес (значимость) характеристики |
| Ссылка | Help | Ссылка на топик с описанием характеристики |

В таблице Char\_Val приведены возможные значения категориальных и ранговых характеристик

|  |  |  |
| --- | --- | --- |
| Char\_Val | | |
| Тип | Поле | Комментарий |
| Целое | Id\_Char\_Val | Id записи |
| Целое | Id\_Char | Id характеристики |
| Целое | Code | Код значения |
| Строка | Value | Значение |

В таблице Anket приведена основная информация об анкете

|  |  |  |
| --- | --- | --- |
| Anket | | |
| Тип | Поле | Комментарий |
| Целое | ID\_Anket | Id анкеты |
| Строка | FIO | Фамилия, имя, отчество пациента |
| Дата/время | DateOfCompletion | Дата заполнения анкеты |
| Целое | Doctor | Id врача, заполнившего анкету |
| Строка | AdmittedFrom | Откуда поступил больной |
| ENUM | FinalDiagnosis | Окончательный диагноз (по умолчанию: «Пока не установлен») |
| Целое | Expert | Id врача, поставившего окончательный диагноз |

В таблице Аnket\_Val приведены значения характеристик конкретной анкеты

|  |  |  |
| --- | --- | --- |
| Аnket \_Val | | |
| Тип | Поле | Комментарий |
| Целое | Id\_ Аnket\_Val |  |
| Целое | Id\_ Аnket | Id анкеты |
| Целое | Id\_Char | Id характеристики |
| Число | Value | Значение характеристики (для числовых характеристик) |
| Целое | CODE | Код значения характеристики (для категориальных и ранговых характеристик) |

В таблице Doctors приведены характеристики врачей, участвующих в процессе диагностики

|  |  |  |
| --- | --- | --- |
| Doctors | | |
| Тип | Поле | Комментарий |
| Целое | Id\_ Doctor | Id доктора |
| Строка | FIO | Фамилия, имя, отчество врача |
| ENUM | Status | Имеет ли доктор право пополнять обучающую таблицу? |
| Строка | Contact | Контактная информация врача |
| Целое | COM | COM адрес мобильного устройства доктора |

Также в системе следующие данные хранятся в формате JSON:

1. Заполненные анкеты в мобильных приложениях.
2. Файл настроек мобильного приложения.

Для защиты данных от потери будет использована система бэкапов БД.

# **Идентификация данных**

Система приложений разрабатывается для использования множеством врачей. Алгоритм системы будет совершенствоваться с поступлением новых данных от пользователей. Для возможности уточнения, корректировки, исследований, данные, поступающие в систему должны быть закреплены за врачом, который их подготовил. Для этого к каждому врачу закрепляется уникальный код, который прикрепляется к передаваемым с мобильного устройства на ЯД данным, чтобы можно было однозначно идентифицировать врача, который их собрал и подготовил к отправке.

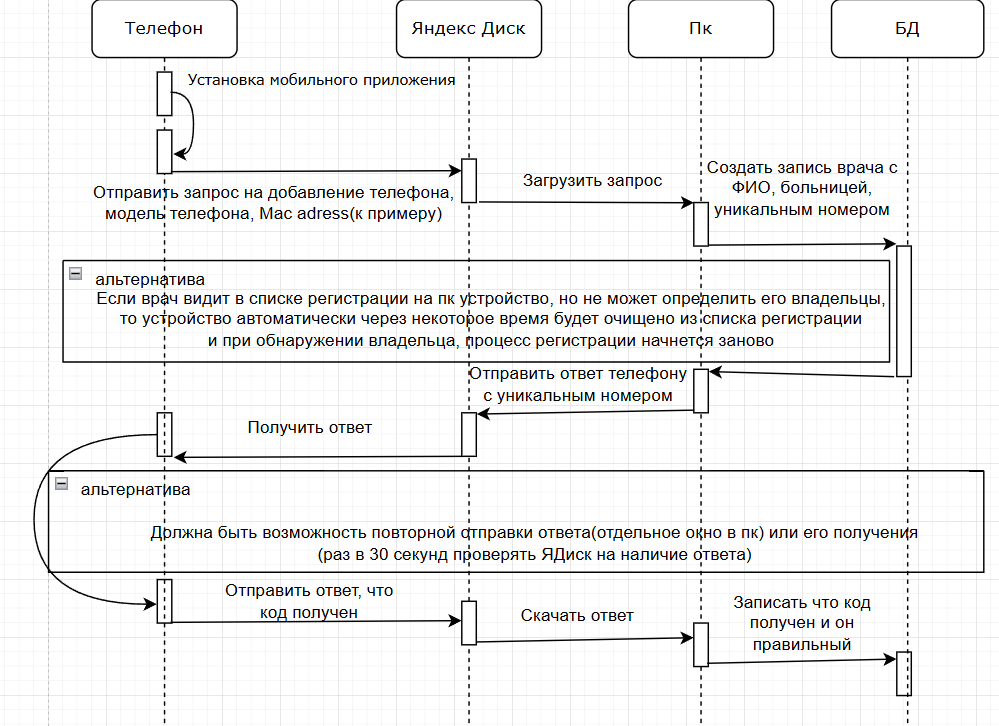
## Передача уникального номера врачу

Процесс передачи уникального номера (кода) происходит сразу после установки приложения врачом на свое мобильное устройство. Врач администратор вносит данные о новом враче-пользователе в таблицу с помощью десктопного приложения, новый врач только передает необходимые личные данные. Больше врачи в процессе не задействованы, остальное происходит автоматически.

Автоматизация процесса закрепления уникального кода за человеком важна по нескольким причинам:

Автоматизированный процесс исключает необходимость в ручном режиме передавать данные закрепленных устройств, что экономит время и усилия врачей и администраторов. Автоматизация снижает риск ошибок, которые могут возникнуть при ручном вводе уникального кода.

Для закрепления уникального идентификационного номера врача за мобильным устройством используется следующий подход:



Внедрение автоматизированного процесса закрепления устройств за врачами может значительно улучшить удобство и безопасность использования мобильного приложения для оценки вероятности рецидивов заболеваний желудка.

# **Систематизация описания приложения**

Система хелпов в приложении для оценки вероятности рецидивов заболеваний желудка должна быть интуитивно понятной и предоставлять исчерпывающую информацию пользователю – врачу или медицинскому работнику, реализована в операционной среде мобильного устройства и быть достаточно быстрой. Она должна быть функциональной, т.е. должна содержать исчерпывающую информацию о всех аспектах работы доктора с приложением, от ввода данных до интерпретации результатов. Быть безопасной и надежной.

## Существующие виды систем хелпов в приложениях:

Система хелпов может быть реализована несколькими способами:

Всплывающие подсказки (tooltips): Краткая информация о назначении каждого поля ввода данных, отображающаяся при наведении курсора или касании элемента.

Контекстная справка: Помощь, вызываемая нажатием кнопки "Справка" или значка "?" рядом с конкретным элементом интерфейса (например, рядом с названием показателя или термином). Она должна предоставлять подробное описание показателя, его значения и значимость в контексте оценки риска рецидива.

Раздел FAQ (часто задаваемые вопросы): Список ответов на наиболее распространенные вопросы пользователей, связанные с использованием приложения, интерпретацией результатов и методологией оценки риска.

Руководство пользователя: Подробное руководство в формате PDF или внутри приложения в виде контекстной справки, содержащее пошаговое описание работы с приложением, включая примеры заполнения форм и интерпретацию результатов.

Видеоуроки: Краткое видео, демонстрирующее процесс использования приложения.

Онлайн-чат или поддержка: Возможность связаться с разработчиками или специалистами для получения помощи.

## Система хелпов в прототипе мобильного приложения

Для приложения заказчики выбрали следующие типы хелпов: Руководство пользователя и контекстную справку. За наполнение хелпов отвечают заказчики. Нужно было лишь подготовить для них программное обеспечение и краткое руководство по его использованию (см Приложение).

## Выбор приложения для создания хелпов

Часть хелпов будут разрабатывать эксперты-врачи, поэтому нужно выбрать приложение для разработки хелпов понятное не техническим специалистам. Для этого были выделены следующие критерии, необходимые при использовании приложения для разработки хелпов:

1. Верстка хелпов в принципе «See What You Print» - это самый важный критерий, т.к. хелпы пишут врачи.
2. Поддержка конвектора в pdf, сайт (html), chm.
3. Цена
4. Наличие русской локализации

Ниже представлена сравнительная таблица разных программ:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Программа | “SWYP” | Конвертор | Цена | Русский язык |
| HTMLHelp Workshop | нет | Да | бесплатно | Да |
| KEL CHM Creator | да | нет | бесплатно | Да |
| Pandoc | нет | да | бесплатно | Нет |
| HelpNDoc | да | да | условно бесплатно | Разработка: Нет  Использование:Да |
| HelpCruiser | Да | Да | 2900 руб единоразово | Да |
| HelpScribble | Да | Да | бесплатно | Нет |
| CHM Editor | да | да | бесплатно | Нет |

Из всех рассмотренных решений наиболее подходящим является HelpNDoc т.к. она наиболее удовлетворяет выделенным критериям и обладает интуитивно понятным пользовательским интерфейсом.

# **Описание функций и структуры мобильного приложения**

Мой прототип программы позволит врачам вводить данные пациентов и получать результаты оценки риска рецидива в любом месте и в любое время, без привязки к стационарным компьютерам и интернету.

Он предоставит врачами быструю и точную оценку риска рецидива, не дожидаясь лабораторных результатов или записи на прием. Это позволит врачам принимать более обоснованные решения о лечении и последующем наблюдении пациентов.

## User story

Перед началом разработки прототипа мобильного приложения были определены основные функции, которые необходимы для корректной работы прототипа: работа с данными, заполнение анкеты, прогнозирование и др. Полный список приведен на рисунке 3.

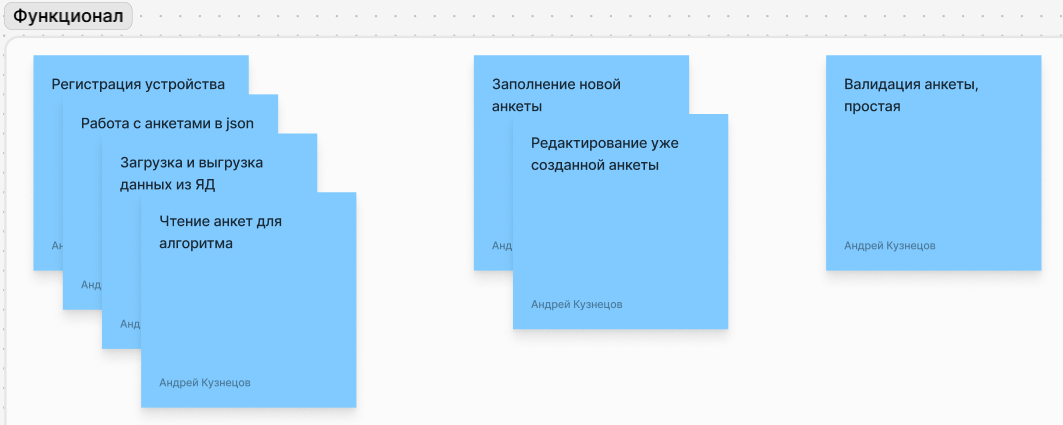


Рисунок 3 Список функций, необходимых в мобильном приложении

## **Первый запуск приложения**

При первом запуске приложения проверяется существование файла settings.json внутри мобильного приложения. Если файл был найден, то открывается доступ к главному меню и остальному функционалу прототипа мобильного приложения. Если файл не существует, то вызывается функция ProcessRegistration класса RegistrationPhone и сообщается пользователю, что для полноценной работы нужен доступ к интернету, только до момента завершения регистрации. При повторном запуске и успешной регистрации устройства прототип будет работать без интернета.

После регистрации устройства, приложение создает settings.json во внутренней файловой структуре, в котором будут храниться настройки приложения, дополнительная информация для работы приложения (например, уникальный код врача).

Так же создается файл ankets.json, хранящий внутри себя заполненные на мобильном устройстве анкеты пациентов, который имеет следующую структуру

[

{

“FIO” : “value”

“char” : “value”

…

“sending” = bool

“dateCreate” : “value”

},

{

“FIO” : “value”

“char” : “value”

…

“sending” = bool

“dateCreate” : “value”

},

…

]

(sending – говорит, отправлялась ли анкета или нет)

(dateCreate – заполнять автоматически)

## **КНОПКИ И ОПИСАНИЕ КАЖДОГО ПРОЦЕССА**

Перед началом разработки прототипа мобильного приложения нужно было определить сценарии использования приложения со стороны врачей. Для этого была разработана информационная структура прототипа мобильного приложения, см рисунок 4.

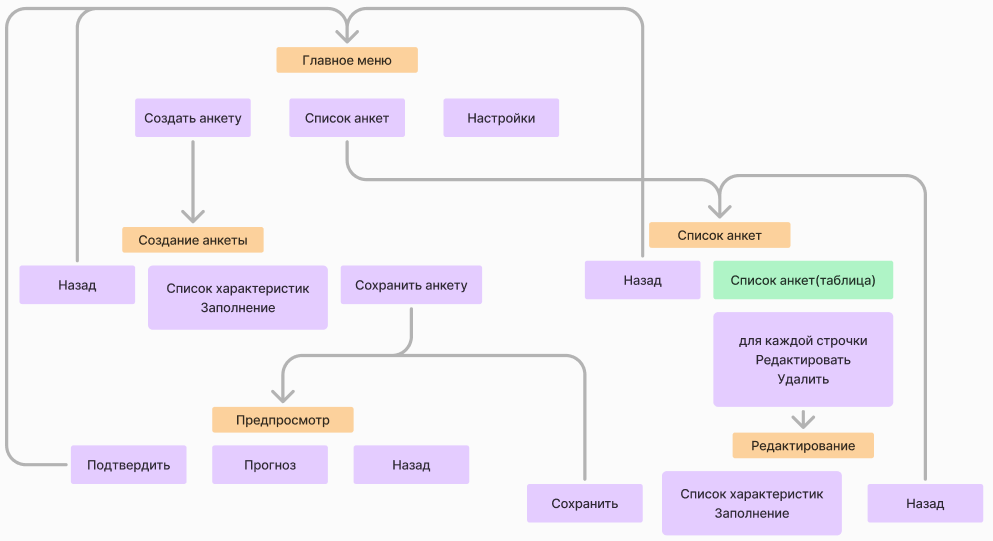


Рисунок Информационная структура прототипа мобильного приложения

По нажатию на кнопку прогноз, в анкету добавляются 2 новых поля, показывающих вероятность предрецидивного синдрома, достаточности консервативного лечения.

### **3.1 Заполнение анкеты:**

Заполнение анкеты происходит путем создания в программе экземпляра класса Patient. После заполнения анкеты она записывается в ankets.json В момент заполнения вызывается простая проверка.

Проверяется заполнение анкеты, был ли уже данный пациент (окончательно утверждать тот ли это человек или другой будет врач, приложение только сообщает, что такой набор ФИО уже существует и показывает его анкету)

## **3.2** Простая проверка анкеты:

Проверка анкеты осуществляется при заполнении каждого из полей анкеты, в момент, когда в поле анкеты вносятся данные. Проверяется соответствие значений характеристик установленным требованиям. Например: пульс человека не может быть меньше 30 или больше 300).

### **3.3 Редактирование анкеты:**

Для редактирования анкеты, необходимо выбрать анкету из списка, загрузить заполненные поля из json. После редактирования новые данные записываются назад в JSON.

Врач сам устанавливает готова ли анкета к отправке на ЯД, полностью ли заполнена анкета. У врача будет время, чтобы изменить свое решение, для этого нужно убрать галочку с поля: анкета готова к отправке. Тогда анкета остается для ее дальнейшего редактирования.

В момент работы с характеристиками, при изменении значения, новое значение сразу проверяется в Verification.CheckCharacteristic(name, value) с помощью простой проверки.

## **Автоматические процессы**:

Список процессов прототипа мобильного приложения, которые работают при запуске программы и наличии доступа к интернету:

1. На ЯД проверяется наличие обновленной частичной обучающей таблицы
2. Проверяется наличие новых анкет на отправку. При нахождении в мобильном приложении неотправленных, но готовых к отправке анкет, они собираются в data (num).json

Формата:

{

“Patients” : [

{

…

},

…

]

“Id\_doctor” : “value”

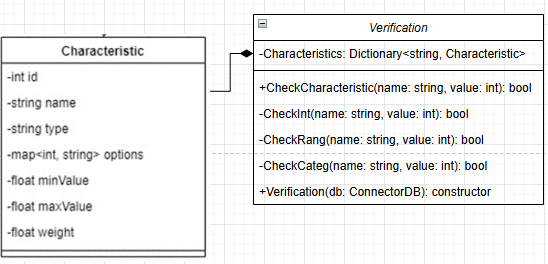
“dateSend” : “value”

}

1. Регистрация пользователя (см главу Идентификации данных)

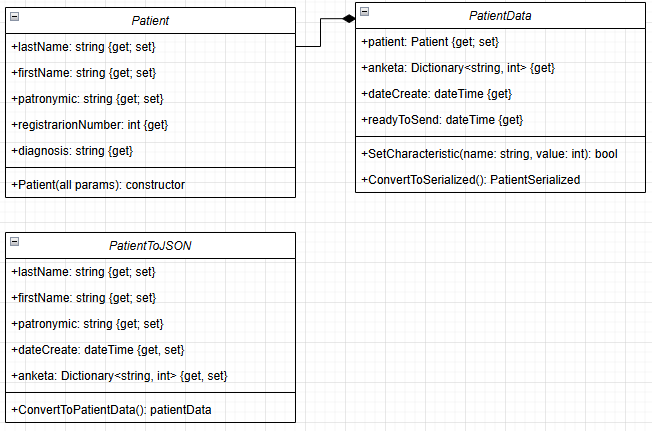
## **5. Диаграмма классов и их описание**

После того, как требования к прототипу были определены, была разработана диаграмма классов (см рисунок 6-8), которые нужны для реализации прототипа мобильного приложения.



Характеристика – данный класс хранит описание характеристик анкеты

Verification – класс, который проверяет правильность заполнения полей анкеты.



Пациент – класс, хранящий основные сведения о пациенте

PatientData – класс связывающий пациента и заполненную анкету

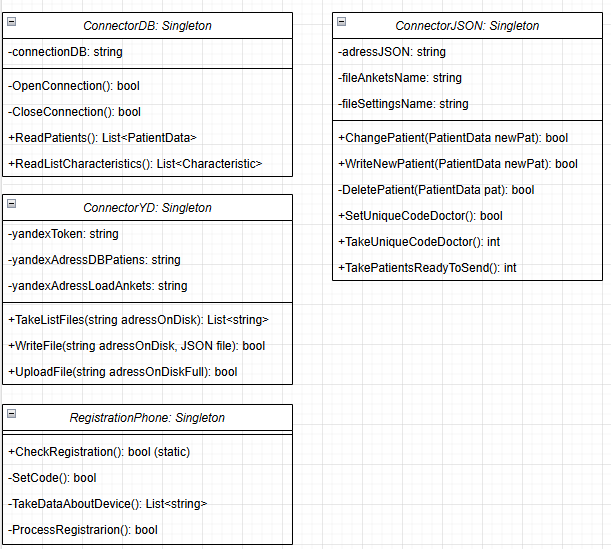
PatientToJSON – класс-прослойка для автоматической сериализации и десериализации JSON файлов

Класс RegistrationPhone реализует процесс регистрации устройства со стороны мобильного телефона.

Класс ConnectorDB используется для работы с бд, выгрузки данных.

Класс ConnectorYD используется для работы с ЯД, выгрузки, загрузки данных на облачное хранилище.

Класс ConnectorJSON используется для работы с JSON, записи, чтения данных этого формата.



# **Заключение**

Подводя итог, необходимо отметить, что в процессе создания прототипа мобильного приложения было выполнено следующее:

- Разработан функциональный прототип, включающий в себя модули для заполнения и редактирования анкеты пользователя,

- Алгоритм обработки данных и расчета вероятности рецидива,

- Система безопасной передачи уникального идентификатора врачу,

- Система безопасной передачи данных между различными устройствами в системе.

Прототип мобильного приложения демонстрирует потенциал развития концепции и определяет основные архитектурные решения для дальнейшей разработки полноценного приложения, включая выбор системы хранения данных и механизмы идентификации. Были определены ключевые User Story и разработана диаграмма классов, иллюстрирующая взаимодействие компонентов системы.

В целом, работа подтвердила возможность создания мобильного приложения для оценки вероятности рецидивов заболеваний желудка, определив перспективные направления дальнейшей разработки и требуемые ресурсы.

# **Список литературы**

1. Марушка А.А. Технологии создания мобильных приложений // Актуальные направления научных исследований XXI века: теория и практика. 2015. № 8-1 (19-1). С. 21-24
2. Рахимов Б. К. Мобильные приложения и Интернет-ресурсы // International scientific and practical conference world science. 2016. № 3-7. С. 37-38.
3. Зуева С.В., Кривоногов С.В. Аспекты и перспективы развития современных информационных технологий // Карельский научный журнал. 2015. № 3 (12). С. 10-12.
4. Чиркова В.М. Перспективы внедрения мобильных приложений при подготовке студентов-медиков к профессиональной деятельности // Карельский научный журнал. 2020. № 1(30). Т.9. С. 43-46.
5. Кудратиллаев М. Б. ПРИМЕНЕНИЯ ТЕХНОЛОГИЙ 5G В СОВРЕМЕННОЙ МИРОВОЙ МЕДИЦИНЕ //Международный научный форум. – 2022. – Т. 1. – С. 915-917.
6. Kudratillaev M. B., Yakhshiboev R. E. ANALYSIS OF INNOVATIVE EQUIPMENT FOR THE DIAGNOSIS OF ASTROENTEROLOGICAL DISEASES //Open Access Repository. – 2023. – Т. 4. – №. 03. – С. 13-23.
7. Яхшибоев Р. Э. РАЗРАБОТКА АППАРАТНО-ПРОГРАММНОГО КОМПЛЕКСА “SALIVA” ДЛЯ ПЕРВИЧНОЙ ДИАГНОСТИКИ ЖЕЛУДОЧНОКИШЕЧНЫХ ЗАБОЛЕВАНИЙ //International Bulletin of Applied Science and Technology. – 2023. – Т. 3. – №. 2. – С. 25-33
8. Яхшибоев Р. Э. РАЗРАБОТКА ПРОГРАММНО-АППАРАТНОГО КОМПЛЕКСА НА ОСНОВЕ НЕЙРОННОЙ СЕТИ ДЛЯ ПЕРВИЧНОЙ ДИАГНОСТИКИ ГАСТРОЭНТЕРОЛОГИЧЕСКИХ ЗАБОЛЕВАНИЙ //Journal of new century innovations. – 2023. – Т. 20. – №. 1. – С. 108-119.
9. . Ziyadullaev D. et al. Development of a traditional transport system based on the bee colony algorithm //E3S Web of Conferences. – EDP Sciences, 2023. – Т. 365. – С. 01017.
10. Никитин П. В., Мурадянц А. А., Шостак Н. А. Мобильное здравоохранение: возможности, проблемы, перспективы // Клиницист. 2015. Т. 9. № 4. С. 13–21.
11. Кучерявый А. Е., Кучерявый Е. А., Прокопьев А. В. Самоорганизующиеся сети. СПб.: Любавич. 2011. 312 с.
12. Гимранов Р. Р., Киричек Р. В., Шпаков М. Н. Технология межмашинного взаимодействия LoRa // Информационные технологии и телекоммуникации. 2015. № 2 (10). С. 62–73. URL: <http://www.sut.ru/doci/nauka/review/2-15.pdf>
13. Блинников М. А., Пирмагомедов Р. Я. Оптимизация нагрузки на сети связи общего пользования, вызванной трафиком медицинских наносетевых приложений // Информационные технологии и телекоммуникации. 2016. Т. 4. № 3. С. 22–30. URL: <http://www.sut.ru/doci/nauka/review/20163/22-30.pdf>
14. Шеряков, Е. В. Мобильные приложения: от идеи до реализации. – Санкт-Петербург: Кристалл, 2021. – 400 с.

# **Приложение**

Вариант структуры контекстной справки

Help

О программе

 Инструкция работы с мобильным приложением

Регистрация пользователя и его смартфона

Заполнение анкеты пациента

Принятие решений на основе прогноза рецидива

Взаимодействие с Яндекс-диск

Инструкция работы с десктопным приложением

...

Характеристики анкеты

Длительность кровотечения

Анамнез

Операции в анамнезе

Способствующие факторы

Рвота кровью

Melena

Коллапс

Потеря сознания

Лечебные мероприятия до настоящего осмотра

Состояние больного во время осмотра

Пульс

АД

ДЦК

Эритроциты

Гемоглобин

Локализация язвы

Размер язвы

Состояние язвенной поверхности

Содержимое желудка и 12 п. кишки

Деформация привратника и 12 п. кишки:

Группа крови

Частота дыхания

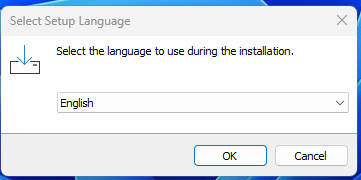
Тяжесть кровопотери

Степень активности кровотечения

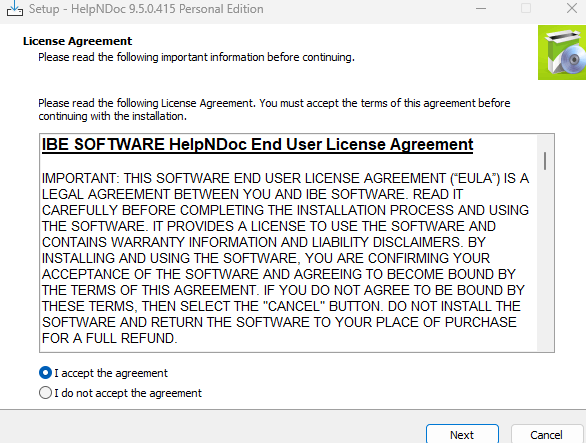
Сопутствующая патология

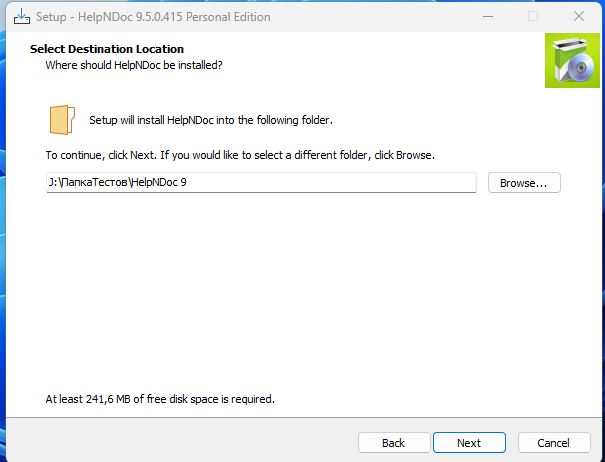
Установка:

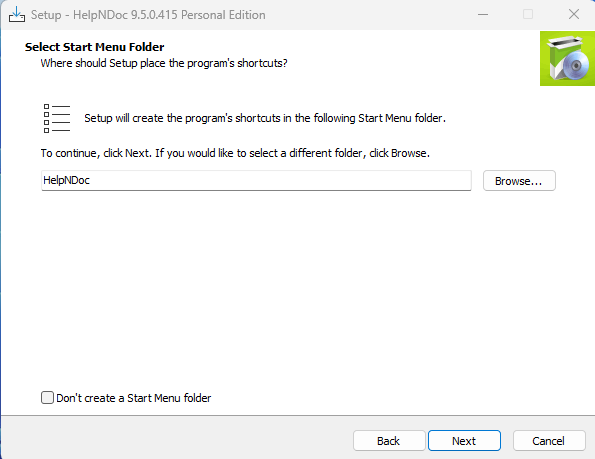
Ссылка для скачивания: <https://www.helpndoc.com/download/>

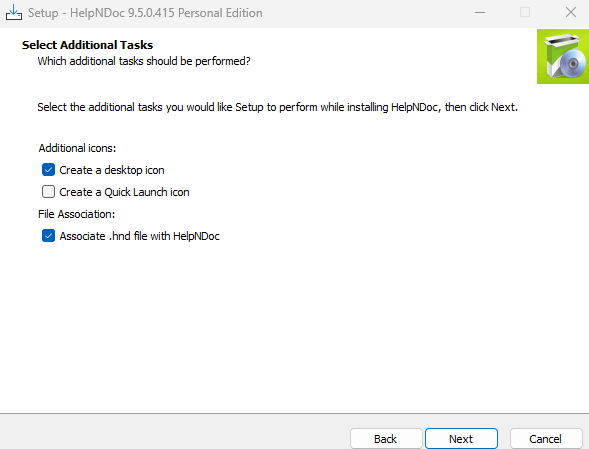


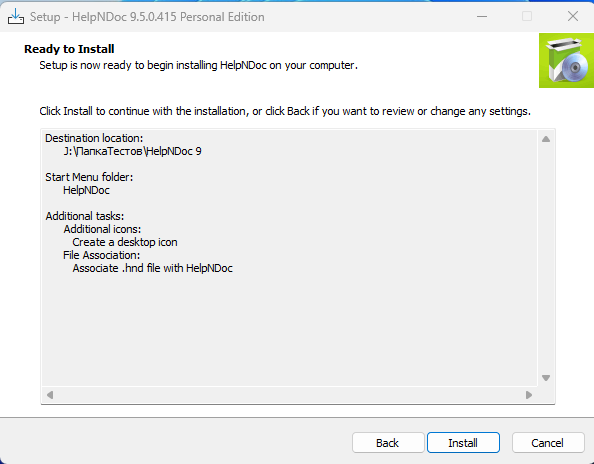
ПРОГРАММА ДОСТУПНА ТОЛЬКО НА АНГЛИСКОМ ЯЗЫКЕ

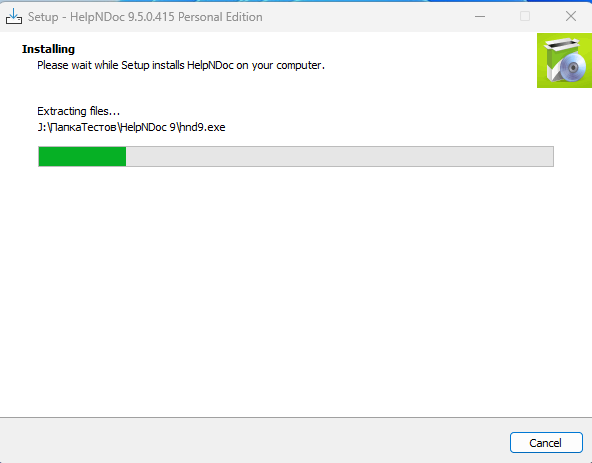


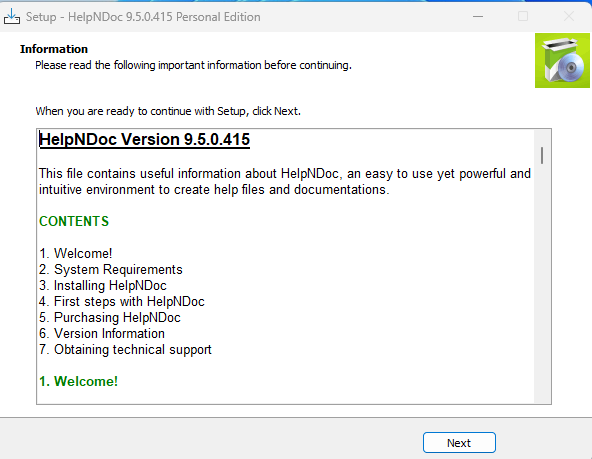


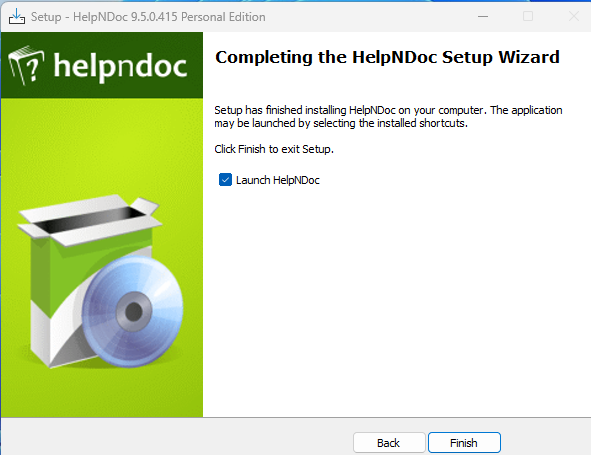












Установка завершена

Основной функционал, который может понадобиться:

1. Создание нового проекта-файла;
2. Создание оглавления;
3. Добавление картинок;
4. Создание связи между главами (гиперссылки).

Должна быть общая папка, в которой будут лежать сам проект документации, прилагающиеся к нему файлы (картинки и другое)

**Создание нового проекта-файла**

После открытия приложения нажать

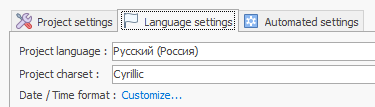
В появившемся окне задать название проекта.



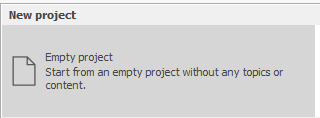


В закладке «Языковые настройки (Language setting) установить язык Project Language = РУССКИЙ(Россия)) и набор символов – кирилица:

Project charset=Cyrillic



Тип проекта:



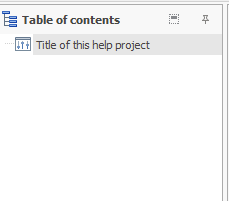
Нажать на кнопку:



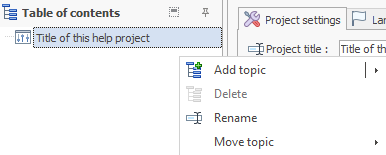
Пустой проект создан.

**Создание оглавления:**

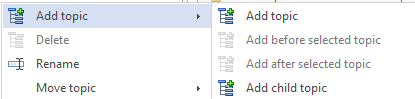
В левом меню:



Нажать правой кнопкой мыши, должно появиться следующее окно?



Вы наводите курсор на окно Add topic:



Затем выбираете одну из функций:

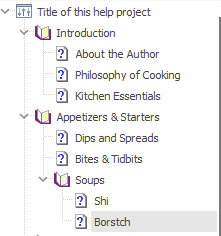
Add topic – создать топик

Add before selected topic – добавить топик перед выделенного

Add after selected topic – добавить топик после выделенного

Add child topic – добавить подраздел в главу

Используя этот инструмент вы можете получить следующее оглавление:



**Добавление картинок**

Создайте отдельную папку, где будут лежать картинки и сам проект

Поместите в папку с картинками, фото которое нужно добавить.

Дайте фото понятное название, по которому можно понять о чем фото (давление, виды язв…)

Откройте главу на которой нужно разместить фото

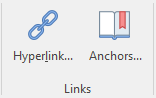
Перетащите курсором файл из папки в проект

**Создание связи между главами (гиперссылки)**

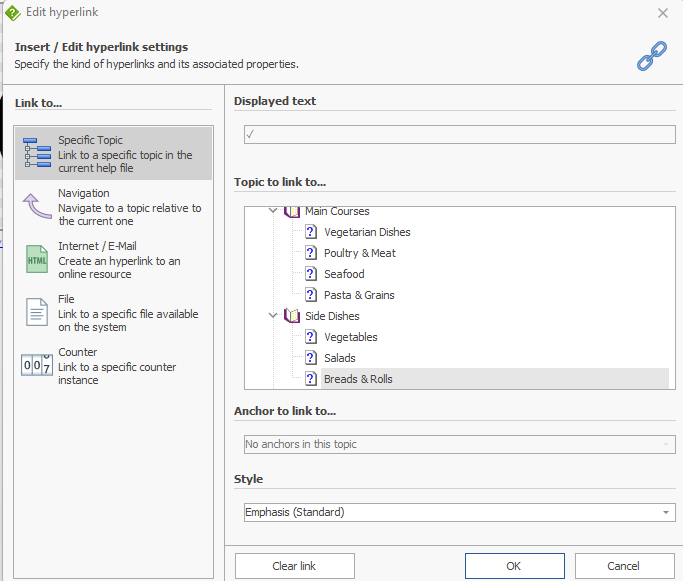
Открываете вкладку Insert в верхней части программы



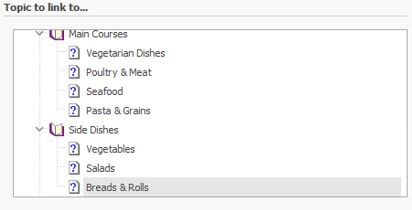
Справа есть инструмент Hyperlink:



Открыв его появляется новое окно:



В появившемся окне, выбираете на какую главу хотите создать ссылку:



Нажимаете кнопку OK

Ссылка успешно создана

