**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](#_Toc183475086)

[Концепция взаимодействия мобильного и десктопного приложения 4](#_Toc183475087)

[Система хранения данных 6](#_Toc183475088)

[Идентификация данных 10](#_Toc183475089)

[Передача уникального номера врачу 10](#_Toc183475090)

[Систематизация описания приложения 12](#_Toc183475091)

[Описание функций и структуры мобильного приложения 14](#_Toc183475092)

[Приложение 17](#_Toc183475093)

# Введение

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

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.

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.

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 децентрализованной структуру хранения данных о пациентах и всей сопутствующей информации, с возможностью сбора данных в десктопном приложении с различных устройств.
3. Разработать систему верификации данных, программа не должна принимать пациентов из непроверенных источников.
4. Хелпы описать, выбрать инструмент, инструкцию для врачей
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 приложения и веб-приложения — это два типа программного обеспечения, которые могут выполнять разные функции и иметь разные характеристики. 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.

При этом десктопные приложения сейчас все больше становятся связаны со следующими тенденциями:

- Облачные технологии. Облачные технологии позволяют десктопным приложениям синхронизировать данные и настройки с другими устройствами, обеспечивать резервное копирование и восстановление, уменьшать расходы на хранение и поддержку, повышать доступность и мобильность.

- Искусственный интеллект. ИИ добавляет десктопным приложениям новые возможности, такие как распознавание речи, изображений, лиц, эмоций, анализ данных, прогнозирование, рекомендации, автоматизация, персонализация и прочее.

- Кроссплатформенность. Она означает, что десктопные приложения могут работать на разных операционных системах, таких как Windows, Mac, Linux, Android и т. д. Это увеличивает аудиторию и удобство для пользователей, которые могут использовать свои любимые приложения на любом устройстве.

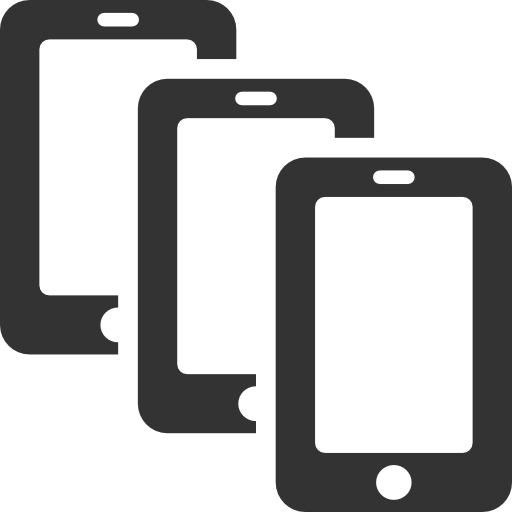
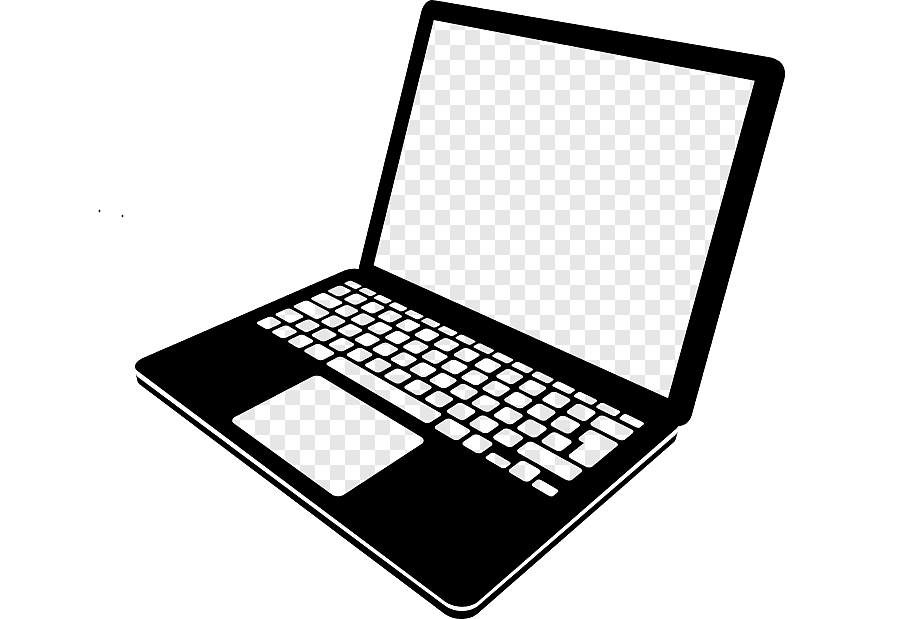
Интеграция с мобильными платформами означает, что десктопные приложения могут взаимодействовать с мобильными приложениями, которые работают на смартфонах, планшетах. Это увеличивает функциональность и удобство приложений, которые могут обмениваться данными, отправлять уведомления.

В разрабатываемой системе есть 2 типа приложений:

1. Десктопное.
2. Мобильное

Мобильное и десктопное приложения взаимодействуют друг с другом через облачное хранилище Яндекс Диск, далее просто ЯД. Эта настройка была выбрана из-за ограниченного бюджета, поскольку ЯД предоставляет бесплатный и простой в использовании сервис для передачи данных.

Яндекс Диск



**Задачи, выполняемые в десктопном приложении:**

Загрузка актуальных данных о обучающей таблице, хранящиеся в базе данных, далее БД, на ЯД.

Загрузка обновленной обучающей таблицы, новый список анкетированных пациентов с ЯД.

**Задачи, выполняемые в мобильном приложении:**

Загрузка данных о анкетированных пациентах с данного устройства на ЯД.

Загрузка обновленных данных о специальной обучающей таблице с ЯД.

**Алгоритм их взаимодействия:**

1. Врач заполняет анкету в мобильном приложении, которая в последствии будет загружена после проверки на ЯД в специальную папку.
2. Мобильное приложение смотрит дату последнего обновления специальной обучающей таблицы и при необходимости загружает ее.
3. Десктопное приложение может выкачать данные о новых анкетированных приложениях, обработать их и загрузить в таблицу Anketa\_Val\_New
4. Десктопное приложение отправляет измененную БД на ЯД.

Эта концепция взаимодействия обеспечивает синхронизацию данных между мобильным и десктопным приложениями, позволяя врачам получать доступ к самой актуальной информации о пациентах с любого устройства.

# Система хранения данных

Так как список характеристик, описывающих пациента и используемых в тестировании, может меняться, то нужно разработать гибкую систему хранения с возможностью изменения или добавления новых характеристик.

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

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

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

**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

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 одинаковые таблицы Anket\_Val\_Trust и Anket\_Val\_New. Когда врач-админ проверит пациента, то он в программе сможет добавить анкету пациента в обучающую таблицу.

Value

Value

Code

Id\_Char

Code

Id\_Anket

Id\_Char

Id\_Anket

**Anket\_Val\_Trust**

**Anket\_Val\_New**

Ниже представлено физическое описание БД:

Описание характеристики

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 |  |
| 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, учавствующих в процессе диагностики

|  |  |  |
| --- | --- | --- |
| Doctors | | |
| Тип | Поле | Комментарий |
| Целое | Id\_ Doctor |  |
| 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 десктопном приложении отдельно хранятся данные в JSON:

1. Заполненные анкеты из мобильных приложений, до момента пока врач-администратор не примет решение о дальнейшей судьбе анкеты. Как только json скачан, то он удаляется из ЯД

Дополнительно будет использована система бэкапов БД.

# Идентификация данных

К каждому врачу прикрепляется уникальный код, который он должен внести в передаваемые данные, чтобы можно было понять, кто их отправил. Такой код называется цифровой подписью.

Цифровые подписи используются для обеспечения целостности и подлинности электронных данных. Они работают следующим образом:

1. Врач получает уникальную неквалифицированную электронную подпись
2. Цифровая подпись прикрепляется к данным.

Если цифровая подпись действительна, получатель может быть уверен, что:

Данные не были подделаны или изменены после подписания.

Данные были отправлены именно тем врачом, который указан в цифровой подписи.

## Передача уникального номера врачу

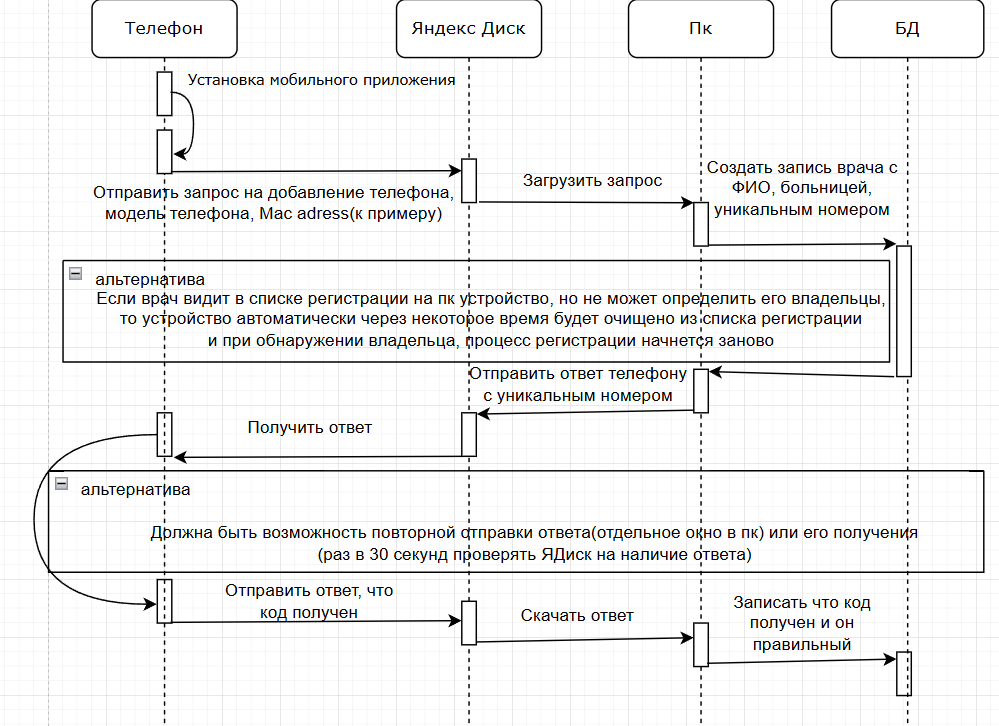
Данный процесс происходит сразу после установки приложения. Врач администратор вносит данные о новом враче в таблицу с помощью десктопного приложения, новый врач только дает необходимые личные данные. Больше врачи в процессе не задействованы, остальное происходит автоматически

Автоматизация процесса закрепления устройства за человеком важна по нескольким причинам:

Автоматизация устраняет необходимость в ручном процессе закрепления устройств, что экономит время и усилия врачей и администраторов.

Автоматизация снижает риск ошибок, которые могут возникнуть при ручном закреплении устройств. Это гарантирует, что устройства всегда закреплены за правильными врачами.

Для подтверждения уникального идентификационного номера врача используется следующий подход:



В дальнейшем, уникальный код врача используется при передаче данных, он передается вместе с данными пациентов и позволяет идентифицировать врача, с чьего устройства переданы данные.

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

# Систематизация описания приложения

Система создания хелпов в приложении для оценки вероятности рецидивов заболеваний желудка должна быть интуитивно понятной и предоставлять исчерпывающую информацию пользователю – врачу или медицинскому работнику, быть быстродейственной. Она должна быть функциональной, т.е должна содержать исчерпывающую информацию о всех аспектах работы приложения, от ввода данных до интерпретации результатов. Быть безопасной и надежной.

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, 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.

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 программного обеспечения происходил по следующим критериям:

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

Ниже представлена сравнительная таблица разных программ:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Программа | “SWYP” | Конвертация в pdf | Цена | Русский язык |
| HTMLHelp Workshop | нет | Да | бесплатно | Да |
| KEL CHM Creator | да | нет | бесплатно | Да |
| Pandoc | нет | да | бесплатно | Нет |
| HelpNDoc | да | да | бесплатно | Нет |
| 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.

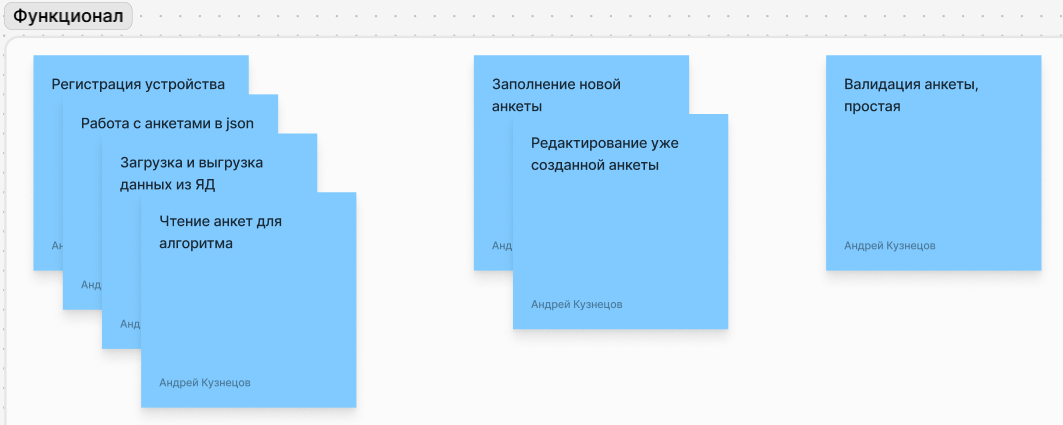
# Описание функций и структуры мобильного приложения

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

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

1. User story

Перед разработкой были определены основные функции, которыми должно обладать мобильное приложение. Работа с данными, заполнение анкеты, прогнозирование. Полный список приведен на фото.



1. Первый запуск приложения

При первом запуске приложения проверяется существование файла settings.json. Если файл был найдет то открывается доступ к главному меню. Если файл не существует то вызывается функция ProcessRegistration класса RegistrationPhone и сообщается пользователю что для полноценной работы нужен доступ к интернету (только в первый раз).

После регистрации устройства, приложение создает settings.json (тут Id доктора), в котором будут храниться настройки приложения, доп информация для работы приложения(например уникальный код врача).

Так же создается файл ankets.json, который имеет следующую структуру

[

{

“FIO” : “value”

“char” : “value”

…

“sending” = bool

“dateCreate” : “value”

},

{

“FIO” : “value”

“char” : “value”

…

“sending” = bool

“dateCreate” : “value”

},

…

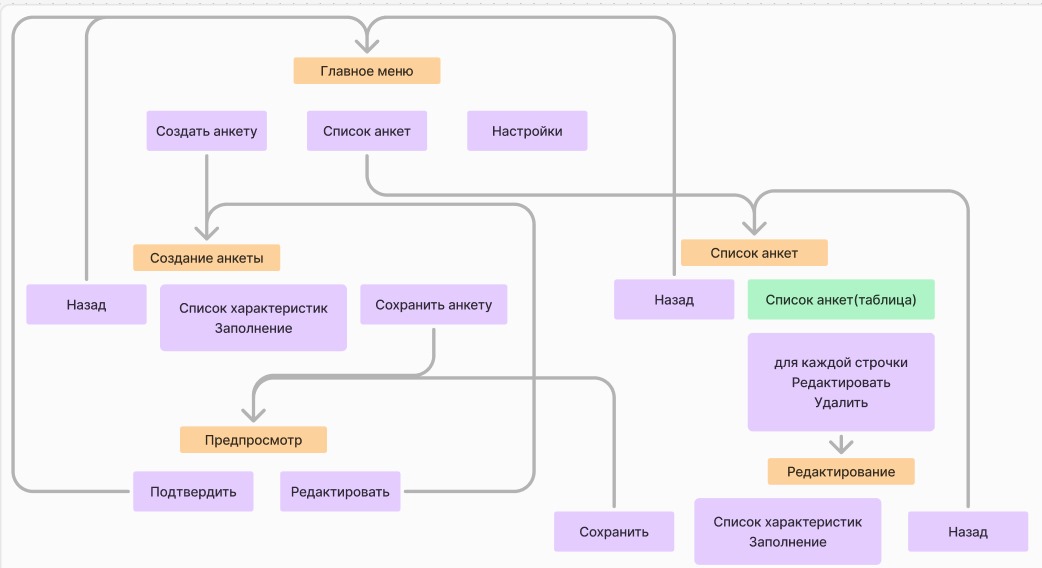
]

(sending – говорит, отправлялась ли анкета или нет)

(dateCreate – заполнять автоматически)

1. КНОПКИ И ОПИСАНИЕ КАЖДОГО ПРОЦЕССА

Перед начал разработки нужно было определить сценарии использования приложения со стороны врачей. Для этого была разработана информационная структура(см фото)



3.1 Заполнение анкеты:

В программе создается экземпляр 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)

Простая проверка анкеты:

Осуществляется при заполнении анкеты. Проверки характеристик на удовлетворение их физическим значениям (Например: пульс человека не может быть меньше 30 или больше 300)

3.2 Редактирование анкеты:

Выбирается анкета из списка, загружаются заполненные поля из json. После редактирования записываются назад в JSON.

Врач сам устанавливает, готова ли анкета к отправке на ЯД(полностью ли заполнена анкета), одно из полей анкеты. У врача будут сутки изменить свое мнение (достаточно убрать галочку). Анкета остается для просмотра.

В момент работы с характеристиками, при изменении значения сразу проверяются в Verification.CheckCharacteristic(name, value)

1. Автоматические процессы:

Работают при запуске программы и наличии интернета:

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

Формата:

{

“Patients” : [

{

…

},

…

]

“Id\_doctor” : “value”

“dateSend” : “value”

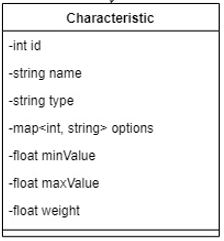
}

1. Регистрация пользователя (только при первом запуске)

5. Диаграмма классов

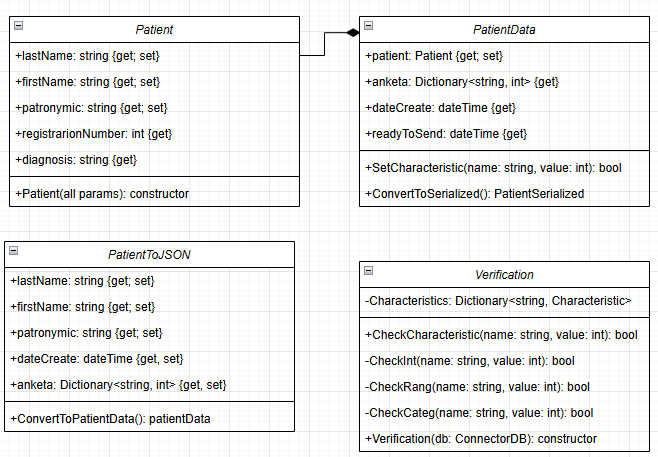
После того как требования к прототипу были определены была разработана диаграмма классов, которые нужны для реализации прототипа.

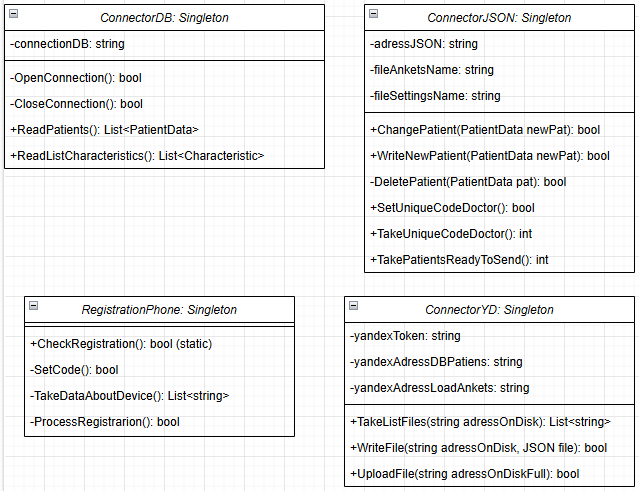
Характеристика



Пациент

Поле – проверили - сохранили





Заключение

Список использованных источников

# 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

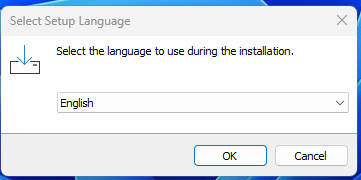
Тяжесть кровопотери

Степень активности кровотечения

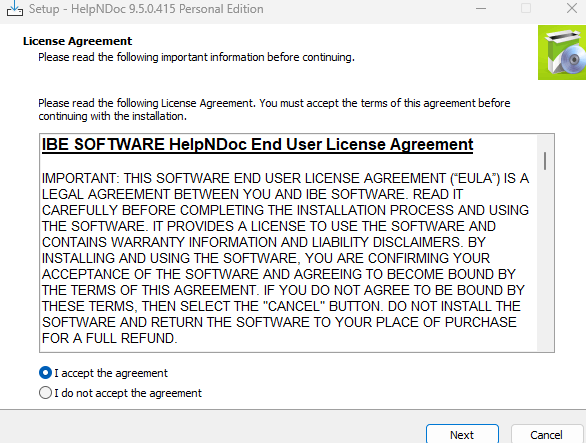
Сопутствующая патология

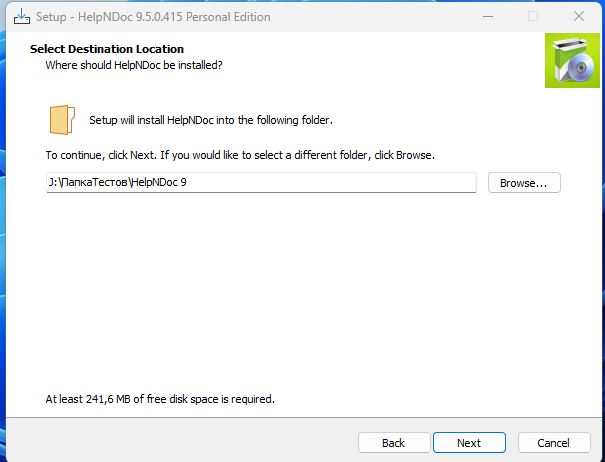
Установка:

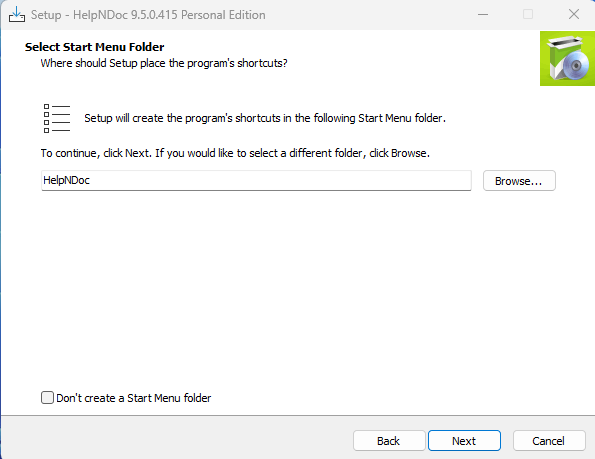
Ссылка для скачивания: <https://www.helpndoc.com/download/>

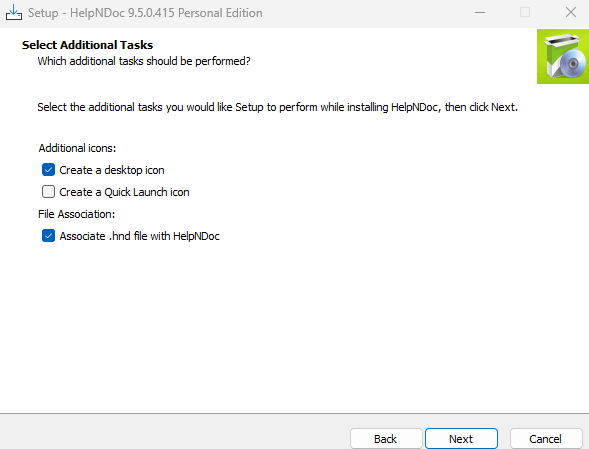


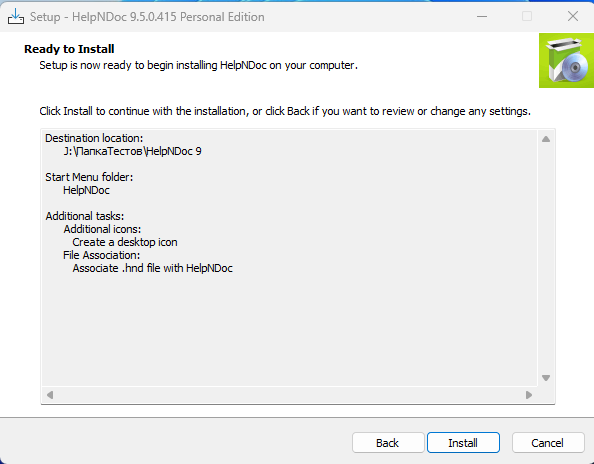
ПРОГРАММА ДОСТУПНА ТОЛЬКО НА АНГЛИСКОМ ЯЗЫКЕ

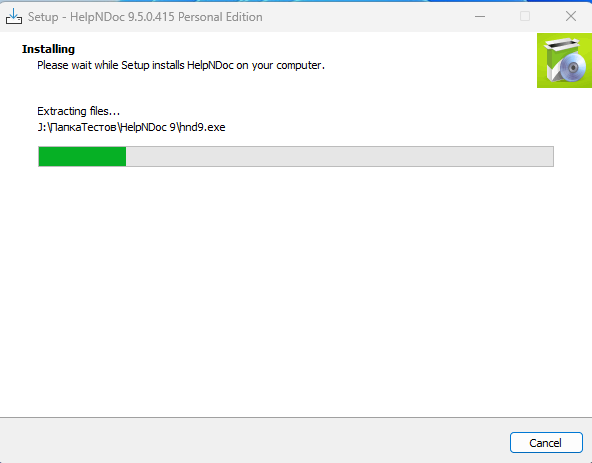


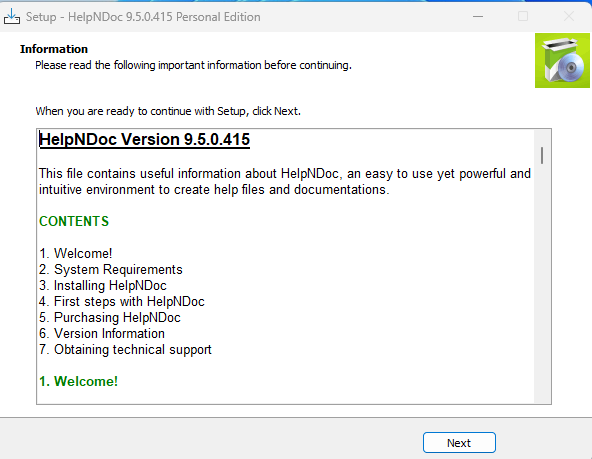


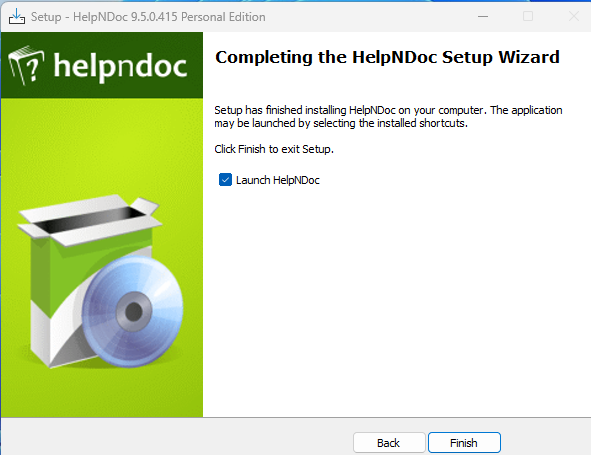












Установка завершена

Основной функционал, который может понадобиться:

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

Создание нового проекта-файла

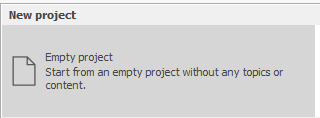
После открытия приложения нажать



В появившемся окне задать название проекта и язык(ОБЯЗАТЕЛЬНО РУССКИЙ):



Тип проекта:



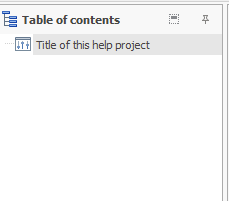
Нажать на кнопку:



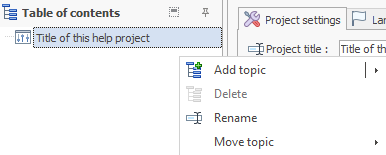
Пустой проект создан.

Создание оглавления:

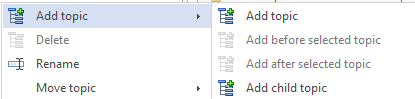
В левом меню:



Нажать правой кнопкой мыши, должно появиться следующее окно?



Вы наводите курсор на окно Add topic:



Затем выбираете одну из функций:

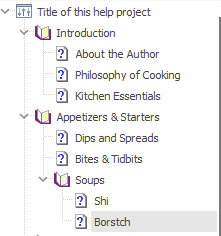
Add topic – создать топик

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

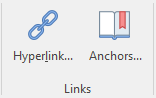
Перетащите курсором файл из папки в проект

Создание связи между главами (гиперссылки)

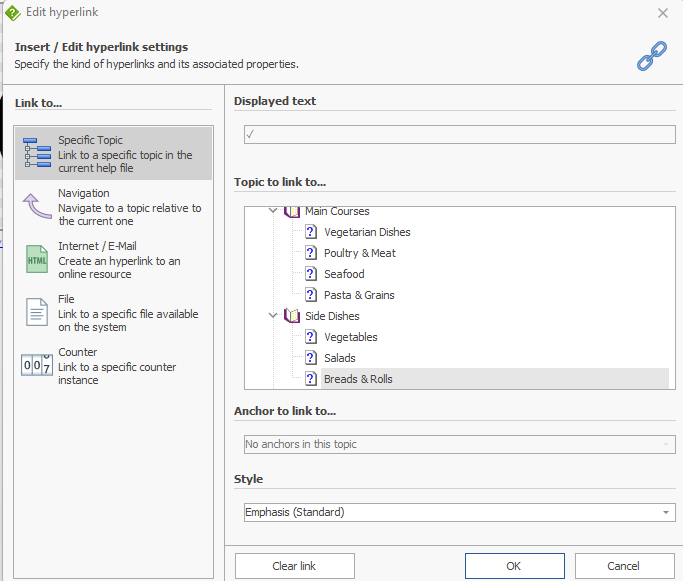
Открываете вкладку Insert в верхней части программы



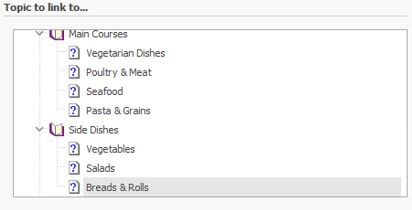
Справа есть инструмент Hyperlink:



Открыв его появляется новое окно:



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



Нажимаете кнопку OK

Ссылка успешно создана

