Федеральное государственное бюджетное образовательное учреждение

высшего образования

«Саратовский государственный технический университет имени Гагарина Ю.А.»

Институт прикладных информационных технологий и коммуникаций

Кафедра «Информационно-коммуникационные системы

и программная инженерия»

**КУРСОВАЯ РАБОТА**

по дисциплине

«Технологии разработки корпоративных информационных систем»

**Разработка модуля прогнозирования для корпоративной системы 6 городской больницы**

Выполнил: студент группы б1-ИВЧТ-31

Кузнецов Андрей Алексеевич

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

подпись студента

Руководитель:

к.т.н., доцент кафедры ИКСП

Безруков Алексей Иосифович

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

подпись руководителя

Курсовая работа защищена на оценку

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Саратов 2025

Оглавление

Оглавление

**Элементы оглавления не найдены.**

Введение

В современной медицинской практике одной из ключевых проблем является предотвращение рецидивов заболеваний желудка, таких как язвенная болезнь, которые могут привести к серьезным осложнениям, включая кровотечения. Несмотря на успехи в лечении, повторные случаи заболеваний остаются значимой угрозой для пациентов, особенно при отсутствии своевременной диагностики и прогнозирования. В связи с этим разработка инструментов, позволяющих оценивать вероятность рецидивов на основе анализа клинических данных, становится крайне востребованной. Такие инструменты не только повышают качество медицинской помощи, но и способствуют накоплению и систематизации знаний, что особенно важно для обучения молодых врачей.

Цель дипломной работы заключается в разработке мобильного приложения, предназначенного для оценки вероятности рецидивов заболеваний желудка. Приложение является частью корпоративной системы 6-й городской больницы и направлено на интеграцию с десктопным решением для создания единой платформы обмена данными и опытом между врачами.

Актуальность темы обусловлена не только необходимостью повышения качества медицинских услуг в России, но и глобальными тенденциями в здравоохранении, где использование технологий для прогнозирования заболеваний становится стандартом. В условиях растущей нагрузки на медицинские учреждения и ограниченных ресурсов, эффективные решения в области прогнозирования могут сыграть решающую роль в управлении здравоохранением.

Цели и задачи:

1. Разработка концепции взаимодействия мобильного и десктопного приложений, включая обмен данными через облачное хранилище.
2. Проектирование системы хранения данных, обеспечивающей гибкость и масштабируемость для адаптации к изменяющимся требованиям.
3. Реализация мобильного приложения с функционалом заполнения анкет пациентов, проверки данных и прогнозирования рисков.
4. Оценка удобства и эффективности предложенного решения на основе обратной связи от потенциальных пользователей — врачей.

Глава 1.

**1.1. Общая структура приложения**

Разрабатываемое мобильное приложение предназначено для оценки вероятности рецидивов заболеваний желудка и является частью корпоративной системы 6-й городской больницы. Оно взаимодействует с десктопным приложением через облачное хранилище (Яндекс Диск), обеспечивая синхронизацию данных и доступ к актуальной обучающей базе.

**Основные модули приложения:**

1. **Модуль регистрации и аутентификации**
   * Регистрация нового врача в системе.
   * Привязка мобильного устройства к учетной записи.
   * Автоматическая загрузка уникального идентификатора.
2. **Модуль работы с данными пациентов**
   * Заполнение анкет пациентов.
   * Проверка корректности введенных данных.
   * Локальное хранение анкет в формате JSON.
3. **Модуль прогнозирования**
   * Оценка риска рецидива на основе введенных данных.
   * Отображение результатов в удобном формате.
4. **Модуль синхронизации с облаком**
   * Загрузка новых анкет на Яндекс Диск.
   * Скачивание обновленной обучающей таблицы.
5. **Модуль справочной системы (Help’ы)**
   * Контекстные подсказки по заполнению анкет.
   * Руководство пользователя.

**1.2. Диаграмма классов мобильного приложения**

Диаграмма классов отражает ключевые сущности системы и их взаимодействие. Основные классы:

**1.2.1. Классы для работы с характеристиками пациентов**

* Characteristic – хранит описание характеристик (название, тип, допустимые границы значений).
* CharacteristicType – хранит описание типов характеристик (числовой, ранговый, категориальный).

**Связи:**

* PatientData использует Characteristic для хранения значений.

**1.2.2. Классы для хранения данных пациентов**

* Patient – содержит основные сведения о пациенте (ФИО, дата рождения и др.).
* PatientData – связывает пациента с его медицинскими показателями.
* PatientToJSON – обеспечивает сериализацию и десериализацию данных в JSON.

**Связи:**

* Patient агрегирует PatientData.
* PatientToJSON преобразует Patient в JSON формат из объекта программы и обратно.

**1.2.3. Классы для взаимодействия с внешними сервисами**

* PhoneData – отвечает за работу с общесистемными переменными, созданием необходимых файлов при первом запуске.
* ConnectorYD – обеспечивает загрузку/выгрузку данных в Яндекс Диск.
* ConnectorDB – обеспечивает загрузку/выгрузку данных в Яндекс Диск.
* ConnectorLocalFiles – работает с локальными файлами.

**Связи:**

* PhoneData считывает данные с помощью ConnectorLocalFiles.
* ConnectorJSON сохраняет анкеты перед отправкой в облако, считывает при запуске приложения.

В данной главе рассмотрена архитектура мобильного приложения, включая его модульную структуру, диаграмму классов и ключевые процессы. Приложение обеспечивает удобный ввод данных, их проверку, локальное хранение и синхронизацию с облаком. Диаграммы наглядно демонстрируют взаимодействие компонентов системы, что подтверждает корректность выбранного подхода к проектированию.

Глава 2.

**2.1. Архитектура экранов и их назначение**

Приложение построено по принципу одностраничного интерфейса с четким разделением функциональных блоков. Каждый экран решает конкретную задачу в логической цепочке работы системы.

**2.1.1. Главный экран (хаб)**

**Назначение:**

* Центральная точка управления всеми функциями приложения
* Обеспечивает навигацию между модулями

**Ключевые функции:**

* Проверка состояния синхронизации данных с Яндекс Диском
* Управление фоновыми процессами обновления данных

**Взаимодействие с данными:**

* При запуске приложения запрашивает у PatientToJSON количество локальных анкет
* Проверяет через ConnectorYD наличие новых данных в облаке

**2.1.2. Экран настроек**

**Назначение:**

* содержит технические параметры работы системы и функции управления учетной записью

**Ключевые функции:**

* содержит кнопку, позволяющую начать процесс регистрации устройства, если оно не зарегистрировано

**2.1.3. Экран регистрации устройства**

**Назначение:**

* Первичная привязка мобильного устройства к врачу в системе
* Загрузка уникального идентификатора из облачного хранилища

**Логика работы:**

1. При первом запуске проверяется наличие поля unic\_code в файле settings.json
2. Если поле отсутствует - активируется процесс регистрации
3. Формируется запрос к Яндекс Диску на получение файла содержащего ID врача
4. Полученные данные сохраняются локально в зашифрованном виде в файле settings.json

**Взаимодействие с другими компонентами:**

* Использует модуль ConnectorYD для облачных операций
* После успешной регистрации передает управление экрану настроек

2.2. Рабочие экраны приложения

**2.2.1. Экран создания/редактирования анкеты**

**Назначение:**

* Ввод и валидация медицинских данных пациента
* Подготовка информации для прогнозирования
* Отображение результатов прогнозирования

**Логика обработки данных:**

1. Для каждого поля вызывается проверка введенного значения
2. Корректные данные передаются в объект PatientData
3. При сохранении формируется JSON-структура

**Динамическая анкета:**

Приложение реализует динамическую систему формирования анкет, которая автоматически адаптируется под обновленные требования. При каждом запуске проверяется актуальная версия структуры анкет, хранящаяся на Яндекс Диске в файле dataUlcer.db. Если обнаружены изменения (добавление новых полей, модификация допустимых значений или удаление устаревших параметров), приложение автоматически:

1. загружает новую конфигурацию,
2. обновляет локальный шаблон анкеты,
3. модифицирует интерфейс ввода данных.

Это позволяет гибко подстраиваться под меняющиеся медицинские протоколы без необходимости обновления всего приложения, обеспечивая постоянную актуальность собираемых данных. Для существующих анкет сохраняется обратная совместимость - отсутствующие в старых записях новые поля помечаются как "не указано" при отображении

**2.2.2. Экран списка анкет**

**Назначение:**

* Управление локальной базой пациентов
* Подготовка к синхронизации с облаком

**Работа с данными:**

* Получает список из ankets.json
* Фильтрует по статусу отправки (sending: false), так же имеет поле поиска пациентов по ФИО
* При выборе конкретной анкеты загружает полные данные через ID

**Критичные процессы:**

* Помечает анкеты для отправки
* Управляет очередью синхронизации

**2.2.3. Экран справки**

**Особенности реализации:**

* Загружает данные из локального хранилища
* Для контекстных подсказок использует поле Help из Characteristic
* Автоматически обновляет справочные материалы при синхронизации

Заключение

Список литературы

Приложение А