**ТЕХНИЧЕСКОЕ ЗАДАНИЕ**

«Healthcare» - программа для заказа лекарств

**Оглавление**

[**1.** **Цели и задачи приложения** 3](#_Toc199763194)

[**2.** **Функциональные требования** 3](#_Toc199763195)

[**3.** **Проектирование и описание структуры данных** 6](#_Toc199763196)

1. **Цели и задачи приложения**

**Цель:** Создание мобильного приложения «Healthcare», обеспечивающего пользователям удобный и быстрый доступ к записи на прием к медицинским специалистам и на медицинские услуги, предоставляющего информацию о посещениях, возможность управления записями и связь со службой поддержки.

**Задачи:**

* Упрощение процесса записи: предоставление интуитивно понятного интерфейса для записи на прием.
* Поиск врачей и услуг: обеспечение эффективного поиска врачей и услуг по различным критериям (специальность, ФИО врача, название услуги, клиника).
* Выбор времени и даты приема: предоставление актуальной информации о доступном времени приема.
* Управление записями: предоставление пользователю информации о запланированных, отмененных и прошедших приемах, а также возможность редактировать и отменять текущие записи.
* Выбор времени и даты приема: предоставление актуальной информации о доступном времени приема.
* Управление записями: предоставление пользователю информации о запланированных, отмененных и прошедших приемах.
* Информирование: отправка пользователю уведомлений о предстоящих приемах, изменениях в расписании и т.д.
* Обеспечение поддержки пользователей: возможность оперативного взаимодействия со службой поддержки через чат.

1. **Функциональные требования**
   1. Аутентификация

Неавторизованные пользователи должны быть ограничены в доступе к данным.

* 1. Регистрация

Для создания нового пользователя необходимо ввести: имя пользователя, электронную почту и пароль. Данные пользователя отправляются на сервер для регистрации.

* 1. Авторизация

Для авторизации пользователя необходимо ввести: электронную почту и пароль. Данные пользователя отправляются на сервер для авторизации.

* 1. Пользовательские сессии
* после успешной аутентификации данные пользователя сохраняется в системе для дальнейшего использования;
* электронная почта и пароль сохраняются в локальное хранилище, они используются для повторного входа и удаляются при выходе из системы;
* данные токена пользователя сохранятся в виде локальной переменной и очищаются при закрытии приложения.
  1. Валидация вводимых данных
* проверять поля на соответствие типу данных;
* проверять поля для ввода на пустоту, на длину (короткий текст и числа: не более 20 символов, пароль: не более 8 символов);
* email на корректность (соответствие паттерну "name@domenname.ru", где имя и доменное имя может состоять только из маленьких букв и цифр).
* проверять повторный ввод пароля на соответствие исходному;
* проверять даты на корректность;
  1. Навигация в приложении
* реализовать навигацию между всеми необходимыми экранами приложения
* для реализации навигации рекомендуется использовать фрагменты или библиотеки навигации;
* для перехода между основными экранами приложения (например, главный экран, профиль, корзина) рекомендуется использовать нижнюю навигационную панель.
  1. ПИН-код
* после аутентификации пользователю открывается экран ПИН-кода, на котором он может ввести свой ПИН-код;
* если пользователь успешно авторизован в системе, при повторном входе в приложение открывается Экран входа по ПИН-коду, в котором пользователь может ввести ПИН-код либо выйти из приложения для повторной аутентификации.
  1. Восстановление пароля
* пользователь может перейти к функции восстановления пароля либо с экрана входа, либо с экрана профиля;
* пользователь вводит электронную почту, на которую приходит шестизначный OTP-код;
* после отправки кода на почту приложение переводит пользователя на экран Ввода OTP-кода, где пользователь должен ввести корректный OTP-код;
* если OTP-код корректен, пользователь переходит на экран Изменения пароля, где вводит и подтверждает новый пароль.
  1. Локальное хранилище

Необходимо хранить данные пользователя, необходимые для автоматической аутентификации, в локальном хранилище приложения.

* 1. Профиль пользователя
* личные данные пользователя сохраняются в его профиле и отображаются на экране Личного профиля;
* на экране Личного профиля пользователь может изменить данные о пользователе: имя пользователя, электронную почту, пароль, аватар и т.д.;
* пользователь может загрузить аватар из галереи устройства.
  1. Отображение множества объектов данных пользователю
* данные должны отображаться в удобном для пользователя формате в виде горизонтального и вертикального списка;
* обновление списка объектов по жесту;
* детализация элементов списка:
* переход на подробное описание элемента по нажатию на элемент;
* вызов контекстного меню при долгом нажатии на элемент;
* действия при нажатии на детали на элементе списка (например, добавление в избранное при нажатии на кнопку на элементе списка)
* экран Получение подробностей об одном объекте.
  1. Фильтрация данных:
* поиск по ФИО врача: пользователь может вводить ФИО врача (или его часть) для поиска.
* поиск по названию услуги: пользователь может вводить название услуги (или его часть) для поиска.
* фильтрация по нескольким свойствам объекта данных (например, специальности, клинике) отдельно или одновременно.
  1. Изменение данных о объекте данных
* создание, редактирование и удаление объектов данных;
* система должна корректно аутентифицировать пользователя и позволять управлять только разрешенными объектами;
* изменение данных должно быть реализовано в удобном для пользователя виде.
  1. Работа со сложными объектами

Отображение, фильтрацию, создание, удаление, редактирование сложных объектов (имеющих две и более связи с другими таблицами) или сложных для восприятия (неструктурированных, или связующих таблиц) в удобном человеко-читаемом формате;

* 1. Работа с изображениями

Все изменяемые медиа должны храниться, создаваться и добавляться из сервиса Supabase.

* 1. Обработка ошибок
* необходимо корректно обрабатывать запросы к базе данных;
* в случае получения ошибки от сервера или отсутствия соединения с сетью Интернет необходимо отобразить соответствующий текст ошибки в диалоговом окне, которое должно закрываться только пользователем.
  1. Выход
* выход из пользовательской сессии;
* удаление данных пользователя из локального хранилища по нажатию соответствующей кнопки.
  1. Локализация
* использовать ресурсы Android там, где это возможно;
* приложение должно быть локализовано на русский и английский язык.
  1. Выбор времени и даты приема
* календарь для выбора даты приема;
* индикация занятых слотов времени;
* подтверждение записи:
  + отображение информации о записи (врач/услуга, дата, время, клиника).
  + подтверждение записи с возможностью добавления комментария (по желанию).
  + сохранение записи.
  1. Просмотр истории посещений
* отображение списка будущих и прошедших записей.
* отображение информации о записи (врач/услуга, дата, время, клиника, статус).
* детализация записи: просмотр подробной информации о записи (врач/услуга, дата, время, клиника, статус, комментарий).
  1. Просмотр и редактирование текущих записей
* Пользователь может просматривать свои текущие (запланированные) записи.
* Редактирование: пользователь может редактировать комментарий к записи (если он есть), время и дату.
* Отмена: пользователь может отменить запись (с подтверждением).
  1. Чат поддержки
* Предусмотреть стартовое приветственное сообщение от службы поддержки (автоматическое).
* Пользователь может отправлять текстовые сообщения в чат.
* Служба поддержки может отправлять текстовые сообщения в чат.
* Отображение даты и времени сообщений.

1. **Проектирование и описание структуры данных**

Примерная структура данных:

* 1. Пользователь (Users):
* user\_id (INT, ПЕРВИЧНЫЙ КЛЮЧ, AUTO\_INCREMENT): Уникальный идентификатор пользователя.
* email (VARCHAR(255), УНИКАЛЬНЫЙ): адрес электронной почты пользователя.
* phone\_number (VARCHAR(20), UNIQUE): номер телефона пользователя.
* password (VARCHAR(255)): Хеш пароля пользователя.
* first\_name (VARCHAR(255)): Имя пользователя.
* last\_name (VARCHAR(255)): Фамилия пользователя.
* middle\_name (VARCHAR(255), NULLABLE): Отчество пользователя (необязательное поле).
* date\_of\_birth (ДАТА, NULLABLE): Дата рождения пользователя.
* gender (ENUM(«мужской», «женский»), NULLABLE): Пол пользователя.
  1. Врачи (Doctors):
* doctor\_id (INT, ПЕРВИЧНЫЙ КЛЮЧ, AUTO\_INCREMENT): Уникальный идентификатор врача.
* first\_name (VARCHAR(255)): Имя врача.
* last\_name (VARCHAR(255)): Фамилия врача.
* middle\_name (VARCHAR(255), NULLABLE): Отчество врача (необязательное поле).
* speciality (VARCHAR(255)): Специальность врача.
* description (ТЕКСТ, ВОЗМОЖНО ПУСТОЕ ЗНАЧЕНИЕ): Дополнительная информация о враче.
* photo\_url (VARCHAR(255), NULLABLE): ссылка на фотографию врача.
* clinic\_id (INT, внешний ключ, ссылающийся на Clinics.clinic\_id): идентификатор клиники, в которой работает врач.
  1. Клиники (Clinics):
* clinic\_id (INT, ПЕРВИЧНЫЙ КЛЮЧ, AUTO\_INCREMENT): Уникальный идентификатор клиники.
* name (VARCHAR(255)): Название клиники.
* address (VARCHAR(255)): Адрес клиники.
* phone\_number (VARCHAR(20), NULLABLE): Номер телефона клиники.
  1. Запись на прием (Appointments):
* appointment\_id (INT, ПЕРВИЧНЫЙ КЛЮЧ, АВТОИНКРЕМЕНТ): Уникальный идентификатор записи на прием.
* user\_id (INT, внешний ключ, ссылающийся на Users.user\_id): идентификатор пользователя, записавшегося на прием.
* doctor\_id (Целое число, внешний ключ, ссылающийся на Doctors.doctor\_id): идентификатор врача, к которому записан пользователь.
* date (DATE): Дата приема.
* time (TIME): Время приема.
* clinic\_id (INT, внешний ключ, ссылающийся на Clinics.clinic\_id): идентификатор клиники, в которой будет проводиться прием.
* status (Целое число (запланировано, выполнено, отменен’), по умолчанию «запланировано»): статус записи.
* comment (ТЕКСТ, NULLABLE): Комментарий к записи.
* created\_at (TIMESTAMP): дата и время создания записи.
* updated\_at (TIMESTAMP): дата и время последнего обновления записи.
  1. Специальности (Specialties)
* specialty\_id (Целое число, первичный ключ, автоинкремент)
* name (VARCHAR(255))
  1. Услуги (Services)
* service\_id (INT, ПЕРВИЧНЫЙ КЛЮЧ, AUTO\_INCREMENT): Уникальный идентификатор услуги.
* name (VARCHAR(255)): Название услуги (например, «Консультация терапевта», «УЗИ брюшной полости»).
* description (ТЕКСТ, NULLABLE): Описание услуги.
* specialty\_id (INT, внешний ключ, ссылающийся на Specialties.specialty\_id): идентификатор специальности, к которой относится услуга (для удобства поиска и фильтрации).
* clinic\_id (INT, внешний ключ, ссылающийся на Clinics.clinic\_id, NULLABLE): идентификатор клиники, в которой предоставляется услуга.
* price (DECIMAL(10, 2), NULLABLE): стоимость услуги.