

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ  
Федеральное государственное автономное образовательное учреждение  
высшего образования

**«Южно-Уральский государственный университет  
(национальный исследовательский университет)»**

**Высшая школа электроники и компьютерных наук  
Кафедра системного программирования**

ДОПУСТИТЬ К ЗАЩИТЕ

Заведующий кафедрой, д.ф.-м.н.,  
профессор

\_\_\_\_\_ Л.Б. Соколинский

«\_\_\_»\_\_\_\_\_ 2024 г.

**Разработка кроссплатформенного приложения  
«Дневник пациента с эпилепсией»**

**ВЫПУСКНАЯ КВАЛИФИКАЦИОННАЯ РАБОТА  
ЮУрГУ – 09.03.04.2024.308-055.ВКР**

Научный руководитель,  
доцент кафедры СП, к.ф.-м.н.  
\_\_\_\_\_ А.Б.А. Алаасам

Автор работы,  
студент группы КЭ-404  
\_\_\_\_\_ Ю.В. Рейхерт

Ученый секретарь  
(нормоконтролер)  
\_\_\_\_\_ И.Д. Володченко  
«\_\_\_»\_\_\_\_\_ 2024 г.

Челябинск, 2024 г.

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ  
Федеральное государственное автономное образовательное учреждение  
высшего образования

**«Южно-Уральский государственный университет  
(национальный исследовательский университет)»**

**Высшая школа электроники и компьютерных наук  
Кафедра системного программирования**

УТВЕРЖДАЮ

Зав. кафедрой СП

\_\_\_\_\_ Л.Б. Соколинский

29.01.2024 г.

### **ЗАДАНИЕ**

**на выполнение выпускной квалификационной работы бакалавра**

студенту группы КЭ-404

Рейхерт Юрию Витальевичу,

обучающемуся по направлению

09.03.04 «Программная инженерия»

**1. Тема работы** (утверждена приказом ректора от 22.04.2024 г. № 764-13/12)  
Разработка кроссплатформенного приложения «Дневник пациента с эпилепсией».

**2. Срок сдачи студентом законченной работы:** 03.06.2024 г.

**3. Исходные данные к работе**

3.1. KIVY документация. [Электронный ресурс] URL:  
<https://kivy.org/doc/stable/> (дата обращения: 15.03.2024 г.).

3.2. Python документация. [Электронный ресурс] URL:  
<https://docs.python.org/3/> (дата обращения: 15.03.2024 г.).

**4. Перечень подлежащих разработке вопросов**

4.1. Изучение литературы по эпилепсии

4.2. Произвести обзор приложений по предметной области.

4.3. Разработать структуру приложения.

4.4. Реализовать приложение.

4.5. Провести тестирование.

**5. Дата выдачи задания:** 29.01.2024 г.

**Научный руководитель,**

доцент кафедры СП, к.ф.-м.н.

А.Б.А. Алаасам

**Задание принял к исполнению**

Ю.В. Рейхерт

## **ОГЛАВЛЕНИЕ**

ВВЕДЕНИЕ.....	4
1. АНАЛИЗ ПРЕДМЕТНОЙ ОБЛАСТИ .....	6
1.1. Обзор эпилепсии .....	6
1.2. Анализ аналогичных проектов .....	8
1.3. Анализ существующих решений для реализации проекта.....	11
2. ПРОЕКТИРОВАНИЕ .....	13
2.1. Требования к проектируемой системе.....	13
2.2. Варианты использования системы.....	14
2.3. Структура базы данных.....	16
3. РЕАЛИЗАЦИЯ .....	18
4. ТЕСТИРОВАНИЕ .....	27
ЗАКЛЮЧЕНИЕ .....	32
ЛИТЕРАТУРА.....	33

## **ВВЕДЕНИЕ**

### **Актуальность**

Эпилепсия является одним из самых сложных и распространенных заболеваний в обществах. Эта болезнь значительно влияет на способность больного вести обычную жизнь, особенно учитывая тот факт, что окончательного лечения этого заболевания на данный момент не существует.

Основное лечение заключается в использовании препаратов, которые подавляют мозговые сигналы, что помогает контролировать приступы, но не приводит к полному избавлению от них. Несмотря на технологический прогресс последних десятилетий, назначение противосудорожных препаратов и корректировка их дозировки врачами по-прежнему в значительной степени основываются на наблюдении за состоянием пациента и регистрации статистических данных, таких как количество ежедневных приступов, длительность каждого приступа, форма приступа [1] и т.д. Кроме того, эпилепсия включает в себя множество типов, каждый из которых может вызывать разные виды припадков. Таким образом, наблюдение приступа во время его возникновения является одним из основных методов диагностики и определения типа эпилепсии, а также оценки прогресса состояния пациента в ходе лечения. В связи с этим возникла потребность в инструментах, которые помогают наблюдать за пациентом и ежедневно фиксировать динамику его состояния [2].

В последние годы появились мобильные приложения, которые помогают пациентам или их близким записывать информацию о приступах. Несмотря на то, что эти приложения предоставляют важные функции, большинство из них все еще не хватает некоторых ключевых возможностей. К примеру, приложения могут работать только на определенной операционной системе, не иметь функции записи видео или не предоставлять достаточную статистику.

Именно поэтому разработка нового кроссплатформенного приложения для больных с эпилепсией является актуальной задачей.

Основной целью проекта будет проведение анализа существующих приложений данного типа, реализация и тестирование приложения удобного и практичного для пользователей разных платформ.

### **Постановка задачи**

Целью выпускной квалификационной работы является разработка кроссплатформенного приложения «Дневник пациента с эпилепсией». Для достижения поставленной цели необходимо решить следующие задачи:

- 1) изучение литературы по эпилепсии;
- 2) произвести обзор приложений по предметной области;
- 3) разработать структуру приложения;
- 4) реализовать приложение;
- 5) провести тестирование.

### **Структура и содержание работы**

Работа состоит из введения, четырех глав, заключения и списка литературы. Объем работы составляет 34 страницы, объем списка литературы – 19 источников.

В первой главе описывается анализ предметной области, выбор языка программирования, фреймворка, операционной системы, анализ конкурентов.

Вторая глава посвящена проектированию приложения, структуре базы данных, функциональным и нефункциональным требованиям, вариантам использования.

В третьей главе описывается реализация приложения, листинг, пример работы приложения.

Четвертая глава посвящена тестированию приложения.

# 1. АНАЛИЗ ПРЕДМЕТНОЙ ОБЛАСТИ

## 1.1. Обзор эпилепсии

Эпилепсия – группа хронических заболеваний, характеризующихся повторяющимися время от времени эпизодами неконтролируемого возбуждения нейронов мозга, что проявляется повторными непровоцируемыми приступами нарушений двигательных, чувствительных, вегетативных, мыслительных или психических функций. На эпилепсию, которой страдает около 50 миллионов человек во всем мире, приходится значительная доля глобального бремени болезней. По оценкам, доля общего населения с активной формой эпилепсии (то есть с продолжающимися припадками или потребностью в лечении) в тот или иной момент времени составляет от 4 до 10 на 1000 человек [3–4].

По оценкам, в глобальных масштабах эпилепсия ежегодно диагностируется у 5 миллионов человек. В странах с высоким уровнем дохода ежегодно регистрируется 49 новых случаев заболевания на 100 000 населения. В странах с низким и средним уровнем дохода этот показатель может достигать до 139 случаев на 100 000 населения. Это, по всей видимости, объясняется повышенным риском таких эндемичных заболеваний, как малярия и нейроцистицеркоз, повышенной частотой дорожно-транспортного травматизма, родовыми травмами, а также дефицитом медицинской инфраструктуры, программ профилактики и доступной медицинской помощи. Почти 80% страдающих эпилепсией проживают в странах с низким и средним уровнем дохода [3–4].

Эпилепсия разделяется на три вида по причинам ее развития. Для лучшего понимания темы вкратце стоит затронуть каждый из видов.

При идиопатической эпилепсии заболевание проявляется только эпилептическими приступами без других неврологических симптомов. Отсутствуют структурные повреждения головного мозга. Врачи предполагают, что идиопатическая эпилепсия имеет генетическую природу и с возрастом проходит.

При симптоматической эпилепсии причина приступов кроется в структурных повреждениях мозга, которые можно обнаружить во время обследования. Риск развития симптоматической эпилепсии повышается также при наличии метаболических заболеваний.

При криптогенной эпилепсии клиническая картина схожа с симптоматической формой болезни, но точно установить причину патологии невозможно. Для криптогенной эпилепсии нельзя однозначно сказать, чем она вызвана: генетическим нарушением или каким-либо состоянием организма [5].

Так как лечение эпилепсии предполагает длительный (в течение многих лет, а иногда и всей жизни) прием противоэпилептических препаратов (ПЭП), назначение противоэпилептической терапии должно быть основано на четких критериях диагностики. Перед назначением постоянной противоэпилептической терапии у пациентов с недавно диагностированной эпилепсией врач должен быть уверен в следующем.

1. У пациента возникают эпилептические приступы (это требует тщательного проведения дифференциального диагноза и исключения любых других состояний, напоминающих эпилептические приступы);

2. Эпилептические приступы, возникающие у больного, требуют назначения терапии (необходима точная диагностика эпилептического синдрома и типа приступов, определения их частоты и тяжести, вероятности рецидива или ремиссии, провоцирующих факторов, индивидуальных проблем, переживаний и вопросов, возникающих у конкретного пациента и членов его семьи, а также соотношения рисков и преимуществ противоэпилептической терапии).

3. Выбор ПЭП для конкретного пациента с определенными типами приступов сделан корректно [6].

## 1.2. Анализ аналогичных проектов

В настоящее время существует несколько приложений, разработанных для помощи больным эпилепсией. Однако, у них есть как свои преимущества, так и недостатки.

Одним из таких приложений является «Epilepsy Journal», которое предоставляет возможность записывать эпизоды эпилепсии и другие события, связанные с заболеванием. Оно позволяет пользователям отслеживать частоту и длительность приступов, вести дневник лекарств и контролировать прием медикаментов. Кроме того, оно предоставляет возможность создания отчетов для врача [7]. На рисунке 1 представлен пример работы данного приложения.

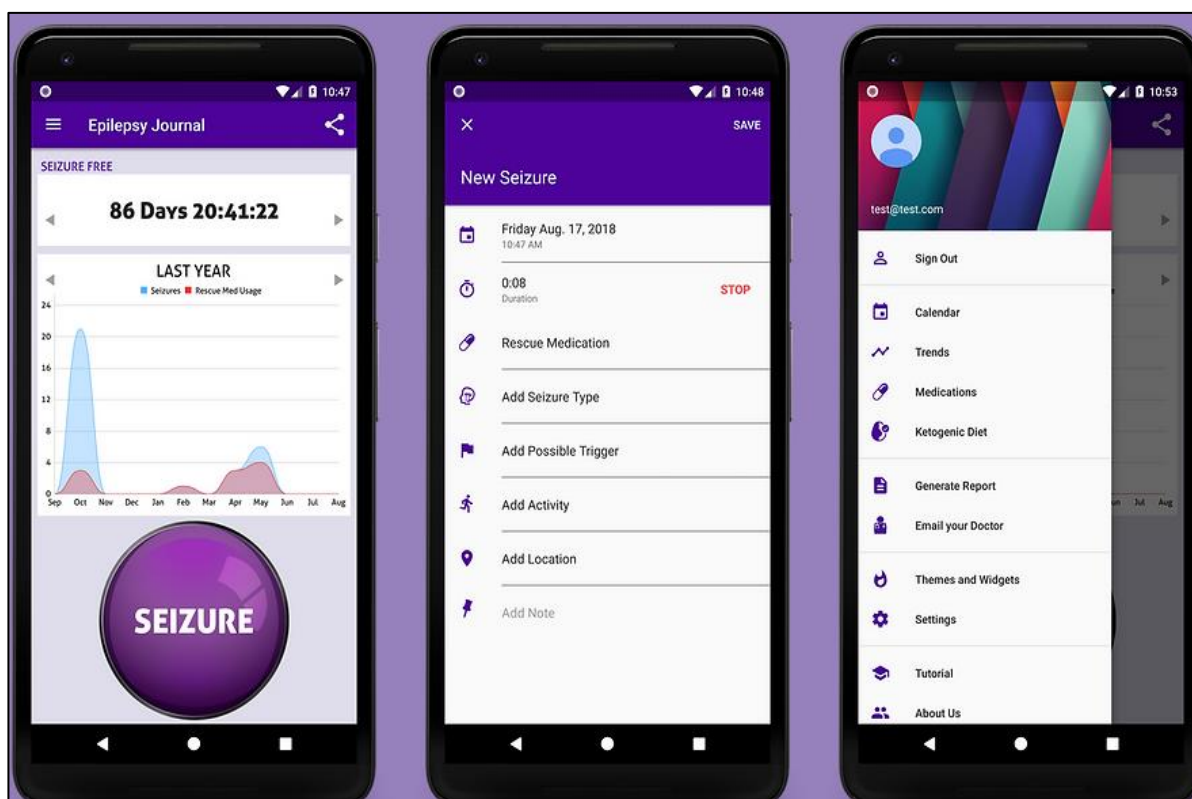


Рисунок 1 – Скриншот приложения «Epilepsy Journal»

Другим примером является «SeizAlarm», которое предупреждает окружающих о начале эпилептического приступа через отправку SMS-сообщений или активацию тревожного вызова. Это может быть полезным для случаев, когда пациент находится без посторонней помощи [8]. В об-



щем, это приложение предназначено для оповещения, а не для записи приступов или статистики о них. Основная проблема заключается в том, что оно может выдавать ложные тревоги, реагируя на некоторые действия, которые не являются приступами, такие как вибрации тела или падения, которые могут не быть связаны с приступами. Кроме того, приложение не может фиксировать некоторые виды приступов, например, приступы, проявляющиеся в виде потери сознания, застывания тела [9], или вибрации отдельных частей тела, а также разнообразные приступы, которые невозможно предсказать [10]. Пример интерфейса приложения «SeizAlarm» предоставлен на рисунке 2.



Рисунок 2 – Скриншот приложения «SeizAlarm»

Новое приложение для кроссплатформенной разработки будет иметь ряд преимуществ по сравнению с существующими. Оно будет доступно

как пользователям мобильных устройств, так и пользователям настольных операционных систем, что позволит большему числу людей воспользоваться его функциональностью. Помимо этого, приложение будет поддерживать ряд важных функций, таких как запись видео, возможность добавления комментариев, добавление нескольких пациентов, предоставление важной статистики и помощи в определении типа эпилептического приступа.

Приложение «EpiDiary» (рисунок 3) позволяет отслеживать эпилептические припадки во время сна, однако реализация данного приложения не позволяет вести запись видео с различных устройств.

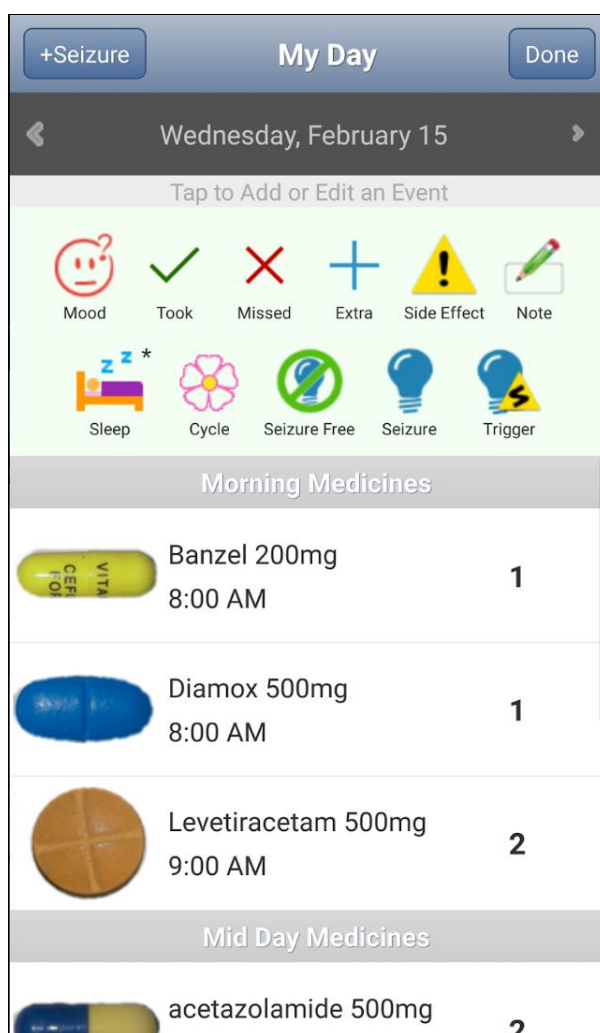


Рисунок 3 – Скриншот приложения «EpiDiary»

### 1.3. Анализ существующих решений для реализации проекта

Для разработки кроссплатформенного приложения «Дневник больного эпилепсией» важно выбрать подходящие инструменты и технологии. В данном разделе рассмотрим несколько языков программирования, фреймворков и сред разработки, чтобы определить наиболее подходящие решения для проекта.

JavaScript – это высокоуровневый и динамически типизированный язык программирования, созданный Бренданом Айком и выпущенный в 1995 году как сценарный язык для веб-браузеров. С тех пор JavaScript значительно увеличил свою популярность и стал доминирующим языком для веб-разработки [11]. Более того JS является достаточно популярным языком и для кроссплатформенной разработки, но имеет ряд минусов, которые не позволяют выбрать его в сравнении с другими языками. JavaScript это клиентский язык программирования и не является серверным языком, что крайне важно для создания кроссплатформенных приложений [12].

Java/Kotlin будут рассмотрены совместно, потому что имеют схожие преимущества и недостатки. Java – это простой, мощный и надежный объектно-ориентированный язык программирования. Он подходит для различных целей, таких как создание и управление Android-приложениями, веб-приложениями, серверными приложениями, встроенными системами, обработкой больших данных и многим другим [13]. Kotlin был разработан компанией JetBrains в 2017 году с целью увеличить продуктивность разработки на Java. Он стал официальным языком программирования в 2018 году [14]. У этих двух языков есть явные минусы для кроссплатформенной разработки. Так, например, Kotlin, хотя и растущий язык, имеет меньшее сообщество разработчиков по сравнению с Java и Python, что приводит к меньшему количеству библиотек и инструментов для разработки. Java, в свою очередь, хоть и отличный язык для кроссплатформенной разработки, требует более длинного и формального кода, в сравнении с Python, что может замедлить скорость разработки [15].

В конечном итоге, для разработки приложения был выбран язык программирования Python. Python – это язык программирования, который широко используется в интернет-приложениях, разработке программного обеспечения, науке о данных и машинном обучении [16]. Python известен своей простотой и легкостью в изучении, огромным сообществом разработчиков, поддержкой асинхронного программирования, мощным набором библиотек, таких как Django и Flask, Tkinter и Kivy. Возможность запуска кода Python без особых проблем на таких операционных системах как Windows, Linux, macOS является несомненным плюсом [17].

Для создания приложения выбран фреймворк Kivy. Kivy – это фреймворк для Python, который позволяет разрабатывать кросс-платформенные приложения с графическим интерфейсом. У фреймворка богатая библиотека виджетов, которые можно использовать для создания интерактивных элементов управления, анимации, графики и других компонентов приложения. Kivy работает на Linux, Windows, OS X, Android и iOS, что делает его удобным для разработки приложений, которые могут запускаться на разных устройствах. Также, имеет хорошую документацию и активное сообщество разработчиков, которые помогают решать проблемы и улучшать фреймворк [18]. Для разрабатываемого приложения используется SQLite. SQLite – это встраиваемая СУБД, которая хранит всю базу данных в одном файле [19].

### **Вывод по первой главе**

В результате обзора эпилепсии были выделены основные задачи для будущего приложения, был рассмотрен механизм диагностирования болезни, а также ее виды по причинам возникновения. Более того анализ конкурентов на рынке помог выявить слабые стороны существующих проектов, что поспособствует более детальной разработке нового. А анализ существующих решений для реализации помогает выбрать наиболее оптимальный язык программирования и фреймворк.

## **2. ПРОЕКТИРОВАНИЕ**

### **2.1. Требования к проектируемой системе**

#### **Функциональные требования**

В рамках работы были выделены следующие функциональные требования.

1. Пользователь должен иметь возможность создавать и входить в аккаунт.
2. Пациент должен иметь возможность редактировать данные своего аккаунта.
3. Пациент должен иметь возможность добавлять информацию о приступах.
4. У пациента должен быть уникальный код, который он сообщает врачу для того, чтобы врач мог его отследить.
5. Приложение должно быть адаптивным под разные размеры экрана.
6. Врач должен иметь возможность создавать и входить в аккаунт.
7. Врач должен иметь возможность добавлять пациентов.
8. Врач должен иметь возможность просматривать статистику приступов пациентов.

#### **Нефункциональные требования**

В рамках работы были выделены следующие нефункциональные требования.

1. Производительность. Приложение должно быть отзывчивым и масштабируемым для выполнения функциональных требований без значительного ухудшения производительности.
2. Безопасность. Приложение должно внедрять надежные меры безопасности для защиты данных пациентов, включая безопасную аутентификацию и удержание данных под контролем пациента.

3. Совместимость. Платформа должна быть совместимой с широким спектром устройств, чтобы обеспечить единообразный пользовательский опыт на разных платформах.

4. Масштабируемость. Архитектура должна поддерживать будущий рост и расширение, позволяя добавлять новые функции и увеличивать количество пользователей.

## 2.2. Варианты использования системы

Разработанная диаграмма вариантов использования представлена на рисунке 4.

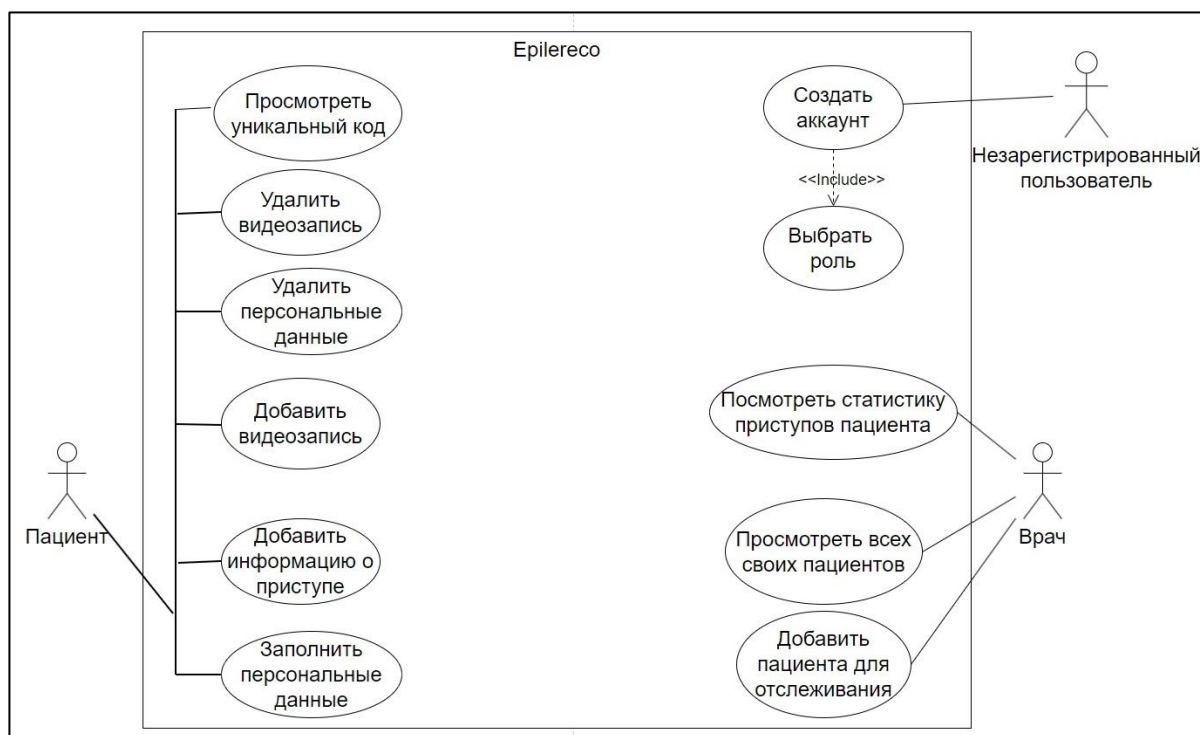


Рисунок 4 – Диаграмма вариантов использования

На диаграмме представлены актеры – пациент, врач и незарегистрированный пользователь, которые описаны ниже.

1. Пациент – это человек, больной эпилепсией, который использует приложение с целью облегчения контроля своего состояния или ведет наблюдение за родственником, страдающим от нее.

2. Незарегистрированный пользователь – человек, который пользуется приложением, не имея аккаунта. При регистрации пользователь выбирает роль пациента или врача для доступа к соответствующим функциям.

3. Врач – человек, который проводит профессиональное медицинское наблюдение за пациентом.

### **Краткое описание вариантов использования**

Незарегистрированный пользователь имеет возможность зарегистрироваться, если он не имеет аккаунта, с помощью формы. Для создания аккаунта незарегистрированный пользователь должен ввести уникальное имя пользователя, выбрать роль пациента или врача и придумать пароль, а так же подтвердить его.

Авторизоваться в системе: для того, чтобы пользователь получил доступ к функциям, ему необходимо авторизоваться в системе. Для входа в аккаунт пользователь должен ввести логин, выбрать роль пациента или врача, а так же ввести пароль.

При входе в свой аккаунт пациент имеет некоторые возможности, которые ему предоставляются.

Пациент может изменить данные о себе, удалив существующие данные, а после внести новые.

Пациент может добавить данные о себе, заполнив необходимые поля (имя, фамилия, отчество, дата рождения, пол). Также может внести информацию о своем приступе. После ввода даты, когда приступ начался, длительности этого приступа, его продолжительности и типа, информация сохраняется в базу данных для дальнейшей обработки. У каждого пациента есть свой уникальный номер, который присваивается при регистрации. Этот номер необходим для передачи его врачу, чтобы врач мог начать отслеживать статистику своего пациента. В версии на стационарном компьютере, пациент может записать видео, используя подключенную камеру. На мобильном устройстве для записи видео пациенту необходимо исполь-

зовать встроенное в устройство приложение. К каждому видео можно добавить текстовый комментарий при записи.

Пользователь может покинуть приложение и выйти из своего собственного аккаунта. В таком случае, он попадает на главную страницу.

Врач, в свою очередь, как и пациент, может зарегистрировать аккаунт и войти в него. Самым главным является возможность врача добавить пациента. Для этого пациент должен сообщить врачу свой уникальный номер. После добавления пациента, врач может просмотреть список уже записанных к нему пациентов, а также вывести статистику по их приступам. В статистике указываются такие данные как:

- 1) среднее количество приступов в день;
- 2) средняя продолжительность приступа за последние 10 и 30 дней;
- 3) самый длинный и самый короткий приступ пациента;
- 4) средняя продолжительность приступов.

### 2.3. Структура базы данных

База данных – это структурированный набор всех данных, расположенных в приложении. В данном случае, отношения между таблицами в базе данных организуются на уровне программирования. Ярким примером является связь с помощью уникального номера, который присваивается каждому пациенту. На основе этого уникального номера, с помощью программного кода, в таблицу добавляются или из нее извлекаются необходимые для работы данные.

В таблице 1 представлена структура таблиц и ее краткое описание, а на рисунке 5 изображена схема базы данных.

Таблица 1 – Описание таблиц БД

Структура таблицы	Данные таблицы
<code>cursor.execute('''CREATE TABLE IF NOT EXISTS patient (name TEXT, surname TEXT, patronymic TEXT, birth_date DATE, sex TEXT, number TEXT, number d TEXT)''')</code>	Персональные данные пациента
<code>cursor.execute('''CREATE TABLE IF NOT EXISTS user_p (username TEXT, password TEXT, role TEXT, number p TEXT)''')</code>	Регистрация пациента (User_P)



Структура таблицы	Данные таблицы
<code>cursor.execute('''CREATE TABLE IF NOT EXISTS user_d (username TEXT, password TEXT, role TEXT, number_d TEXT)''')</code>	Регистрация врача (User_D)
<code>cursor.execute('''CREATE TABLE IF NOT EXISTS video (number_p TEXT, video_name TEXT)''')</code>	Привязка видео к конкретному пациенту
<code>cursor.execute('''CREATE TABLE IF NOT EXISTS seizure (number_p INTEGER, patient_name TEXT, seizure_start DATE, seizure_duration INTEGER, seizure_type TEXT)''')</code>	Информация по приступам

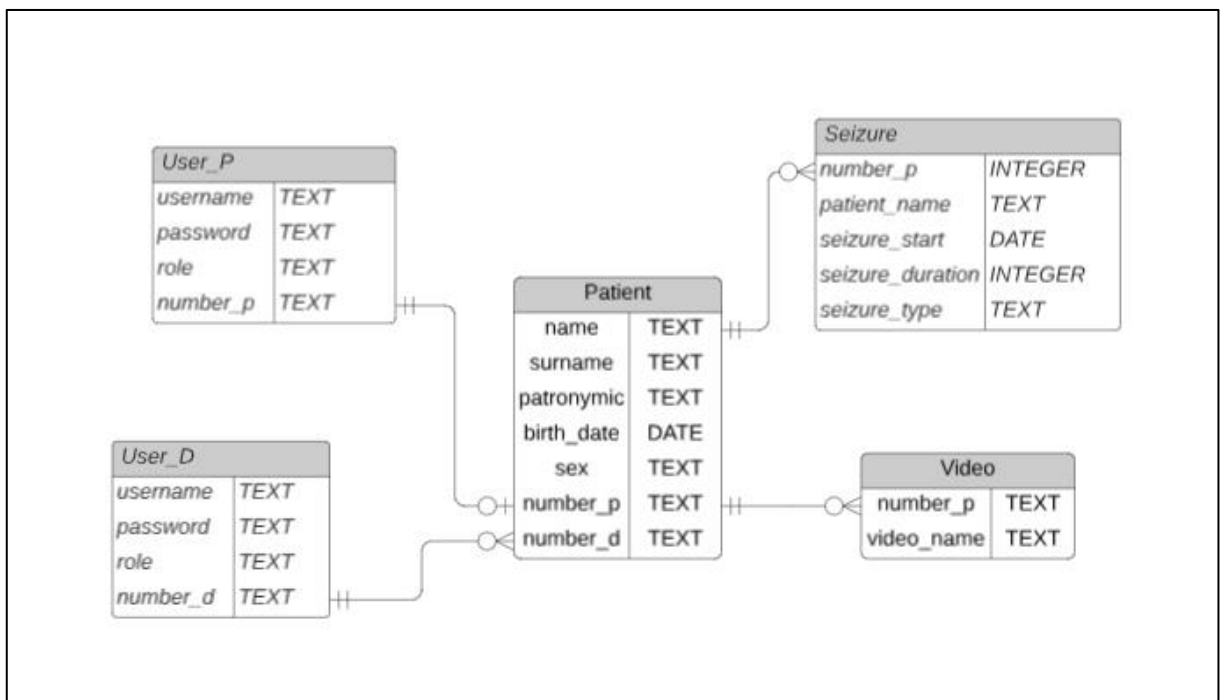


Рисунок 5 – Схема базы данных

### Вывод по второй главе

В результате проектирования были выявлены функциональные и нефункциональные требования к системе. На основе требований была составлена диаграмма вариантов использования приложения и определены основные актеры, которые будут взаимодействовать с приложением. Также была составлена схема базы данных.

### 3. РЕАЛИЗАЦИЯ

Первоочередной функцией должна быть регистрация пользователей, для доступа ко всем возможностям приложения (листинг 1).

#### Листинг 1 – Функция регистрации пользователя

```
def signup(self, username, password, confirm_password, role):
    if role == 'Patient':
        if is_username_available_p(username):
            if password == confirm_password:
                number = random.randint(1000000000, 9999999999)
                create_user_p(username, password, role, number)
                folder_name = str(number)
                user_data_dir = App.get_running_app().user_data_dir
                # Полный путь к новой папке
                full_path = os.path.join(user_data_dir, folder_name)
                if not os.path.exists(full_path):
                    os.makedirs(full_path)
                    print(f'Папка "{full_path}" успешно создана!')
                    self.error_message = "Welcome!"
                    App.get_running_app().root.current = 'login'
            else:
                self.error_message = "Passwords do not match!"
        else:
            self.error_message = "Patient already exists"
    else:
        if is_username_available_d(username):
            if password == confirm_password:
                number = random.randint(1000000000, 9999999999)
                create_user_d(username, password, role, number)
                folder_name = str(number)
                user_data_dir = App.get_running_app().user_data_dir
                # Полный путь к новой папке
                full_path = os.path.join(user_data_dir, folder_name)
                if not os.path.exists(full_path):
                    os.makedirs(full_path)
                    print(f'Папка "{full_path}" успешно создана!')
                    self.error_message = "Welcome!"
                    App.get_running_app().root.current = 'login'
            else:
                self.error_message = "Passwords do not match!"
        else:
            self.error_message = "Doctor already exists"
```

В рамках регистрации пользователя происходит проверка на правильность введенного пароля и на доступность имени пользователя. Помимо этого, пользователю присваивается уникальный номер и создается папка пользователя для хранения информации.

В рамках дизайна самой регистрации был разработан макет, представленный на рисунке 6.

Create an Account	
Username	
Password	
Confirm Password	
Select user type	
Sign Up	
Back	

Рисунок 6 – Регистрация пользователей

Также немаловажной являлась функция входа в аккаунт после проверки существования данного пользователя и корректности введенного пароля, которая была реализована следующим образом, который отображен в листинге 2.

#### Листинг 2 – Функция входа в аккаунт

```
def login(self, username, password, role):
    os_name = platform.system()
    if os_name == 'Linux':
        # Запрос разрешений для Android
        from android.permissions import request_permissions, Permission
        request_permissions([Permission.CAMERA,
        Permission.WRITE_EXTERNAL_STORAGE, Permission.INTERNET,
        Permission.READ_EXTERNAL_STORAGE])
    if role == 'Patient':
        if validate_login_p(username, password):
            set_current_user(role)
            set_current_number(username)
            if username == 'Admin':
                set_current_user(username)
                app = App.get_running_app()
                app.root.get_screen('main').main.set_user(username)
                app.root.current = 'main'
```

```

else:
    self.error_message = ""
    app = App.get_running_app()
    app.root.get_screen('patient').patient.set_user(username)
    app.root.current = 'patient'
else:
    self.error_message = "Login failed!"
elif role == 'Doctor':
    if validate_login_d(username, password):
        set_current_user(role)
        set_current_number_d(username)
        self.error_message = ""
        app = App.get_running_app()
        app.root.get_screen('doctor').doctor.set_user(username)
        app.root.current = 'doctor'
    else:
        self.error_message = "Login failed!"
else:
    self.error_message = "Choose your role!"

```

Для добавления персональной информации и для того, чтобы узнать свой уникальный номер, необходимый для отслеживания врачом, пациент выбирает соответствующую функцию, которая имеет следующее отображение, продемонстрированное на рисунке 7.

Name
Surname
Patronymic
Birth Date
Select Sex
Save info
Show unique code
Delete info
Back

Рисунок 7 – Экран добавления персональных данных

Сам же врач имеет следующий экран просмотра и добавления пациентов, который показан на рисунках 8 и 9.

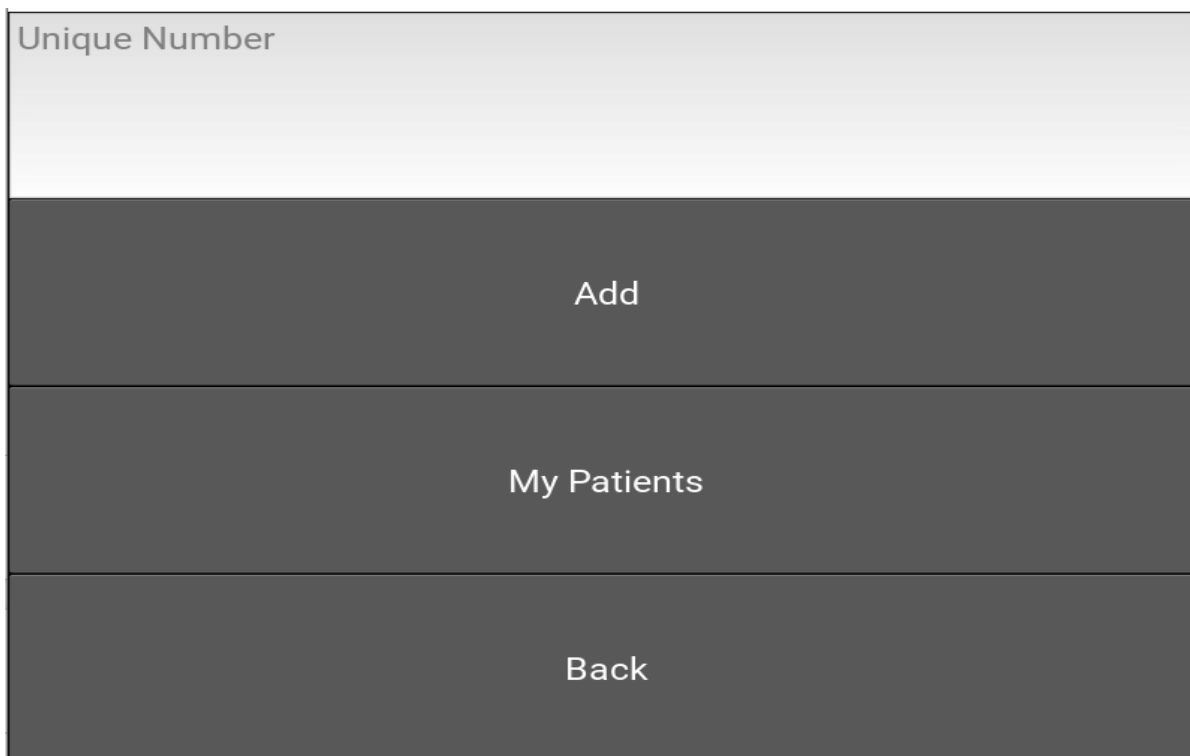


Рисунок 8 – Добавление нового пациента для отслеживания

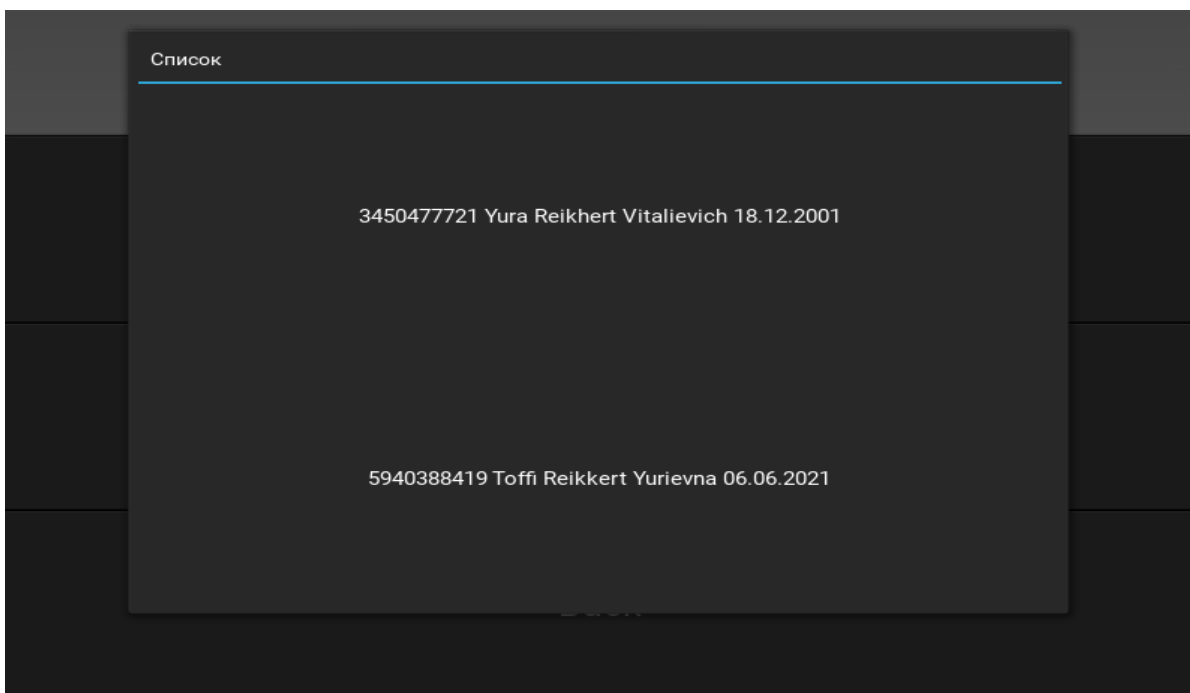


Рисунок 9 – Просмотр закрепленных за врачом пациентов

Экран проигрывания и удаления видео изображен на рисунке 10. Пациент может как выбрать видео для проигрывания из существующих на его устройстве, так и удалить, если запись была сделана случайно или по-

теряла свою актуальность. Сам код, используемый для реализации возможности проигрывания видео, указан в листинге 3.

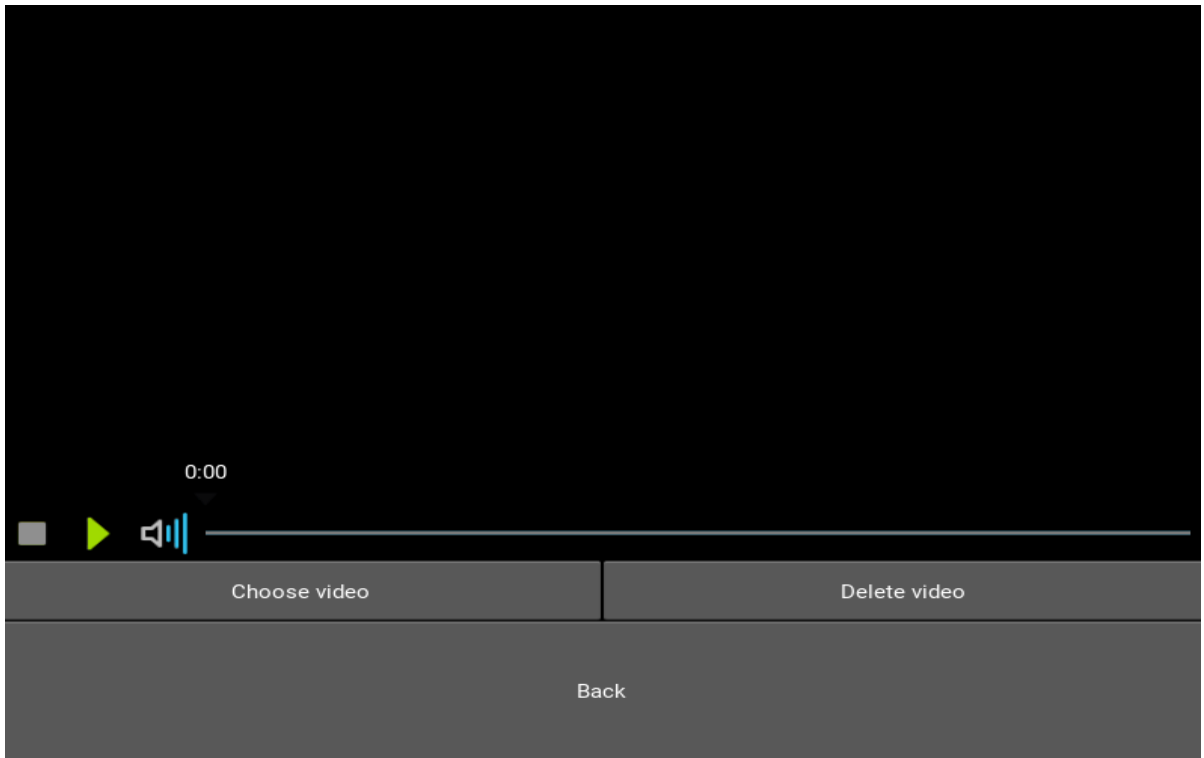


Рисунок 10 – Окно видеоплеера

### Листинг 3 – Функции проигрывания видео

```
def play_video(self, instance):
    os_name = platform.system()
    # self.info_message = os_name
    if os_name == 'Linux':
        from plyer import filechooser
        filechooser.open_file(title="Выберите видео",
on_selection=self.handle_selected_file)
    else:
        self.setup_file_chooser_for_other_os()
@mainthread
def handle_selected_file(self, selection):
    if selection:
        path = selection[0]
        self.setup_file_chooser(path)
    else:
        self.info_message = 'Файл не выбран'
def setup_file_chooser(self, path):
    file_chooser = FileChooserListView(path=path)
    self.popup = Popup(title="Выберите видео", content=file_chooser,
size_hint=(0.8, 0.8))
    self.popup.open()
    file_chooser.bind(selection=self.on_file_selected)
def on_file_selected(self,instance, selection):
    if selection:
        selected = selection[0]
        with connect():
            number = get_current_number()
```

```

valid_name = number
filename = os.path.basename(selected)
os_name = platform.system()
if os_name == 'Windows':
    if filename.startswith(valid_name):
        self.ids.video_player.source = selected
        self.ids.video_player.state = 'play'
        self.info_message = ''
    else:
        self.info_message = 'Choose YOUR video'
else:
    self.ids.video_player.source = selected
    self.ids.video_player.state = 'play'
    self.info_message = ''
self.popup.dismiss()
self.popup.open()
def setup_file_chooser_for_other_os(self):
    with connect():
        number = get_current_number()
        user_data_dir = App.get_running_app().user_data_dir
        user_folder_path = os.path.join(user_data_dir, str(number))
        self.setup_file_chooser(user_folder_path)

```

Ради наглядности представлена структура бэкенд и фронтенд части приложения на рисунке 11.

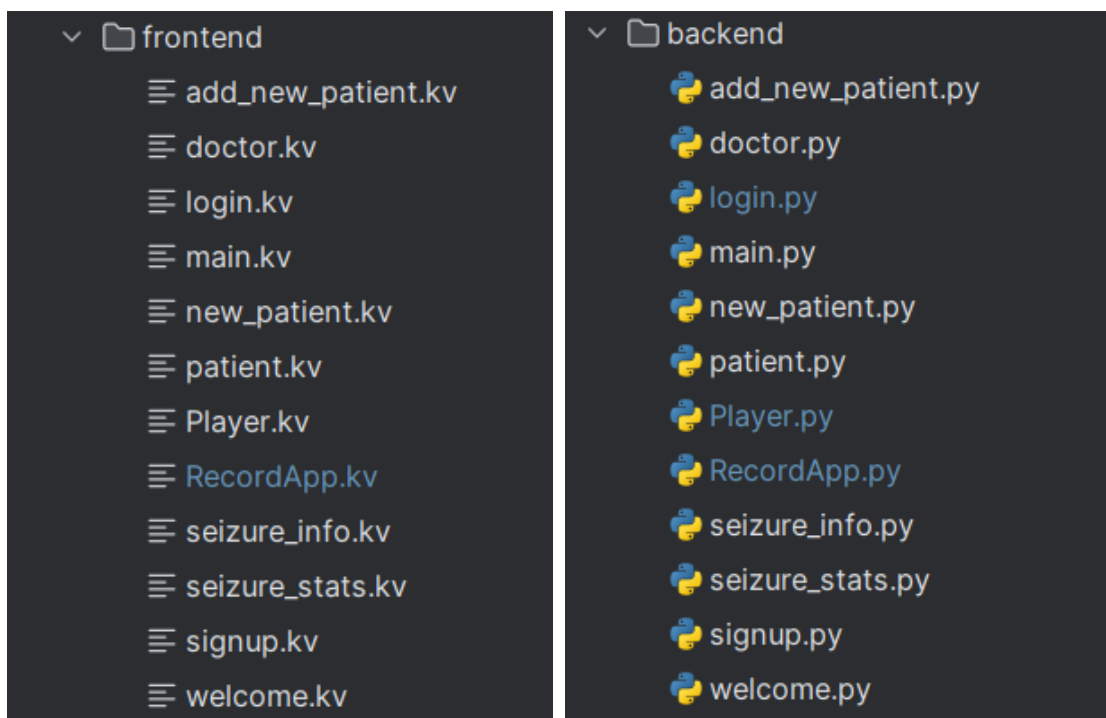


Рисунок 11 – Структура бэкенд-приложения(слева) и фронтенд-приложения(справа)

На рисунке видно, что использовались файлы с расширением «py» и «kv». Файлы с идентичными названиями в совокупности образуют один

экран. Так, например, экран, представленный ранее на рисунке 9, образуется файлами «Player.py» и «Player.kv». А пары файлов «doctor.py», «doctor.kv» и «main.py», «main.kv» образуют экраны, на которые может попасть пользователь после авторизации в зависимости от выбора роли. Все файлы вместе собираются за счет одного основного. Такая структура помогает разделить функции на отдельные кластеры логически, что повышает читаемость кода и упрощает работу с ним.

Для сборки пакета, необходимого для запуска приложения на операционной системе Android, использовалась утилита «Buildozer». Buildozer – это инструмент разработки, позволяющий превращать приложения Python в бинарные пакеты, готовые к установке на любую из множества платформ, включая мобильные устройства. Для корректной работы с «Buildozer» необходим отдельный файл «buildozer.spec», в нем отображаются необходимые для работы приложения конфигурации.

Функция записи видео была реализована с использованием библиотеки OpenCV, которая была выбрана не случайно, ведь это мощная библиотека с открытым исходным кодом для компьютерного зрения и обработки изображений, что позволит впоследствии использовать ее же для внедрения в систему «Умный дом».

OpenCV это не что иное как открытая библиотека для работы с алгоритмами компьютерного зрения, машинным обучением и обработкой изображений. Написана на C++, но существует также для Python, JavaScript, Ruby и других языков программирования. Работает на Windows, Linux и MacOS, iOS и Android. В самом приложении экран выглядит так, как это отображено на рисунке 12. А меню выбора файла для загрузки показано на рисунке 13.

Интерфейс мобильного приложения не отличается от версии на стационарных компьютерах. Интерфейс мобильной версии показан на рисунке 14.



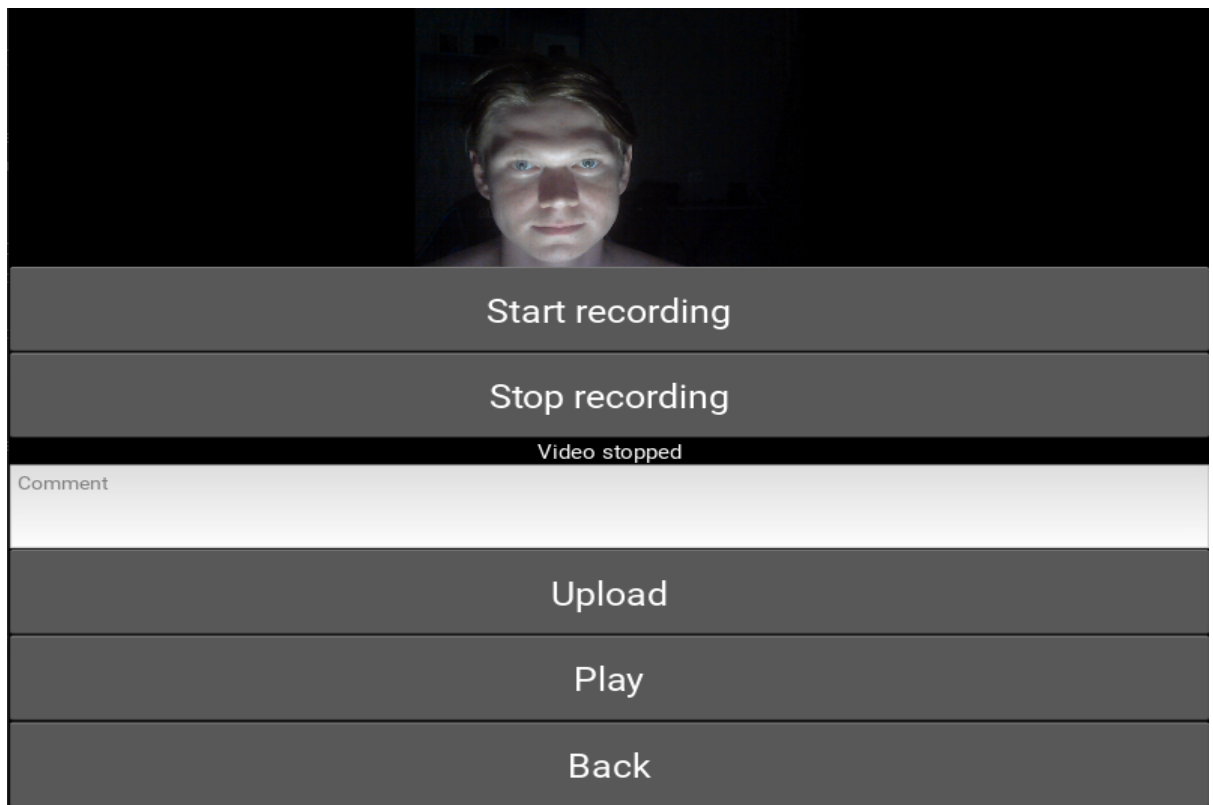


Рисунок 12 – Экран записи и загрузки видео

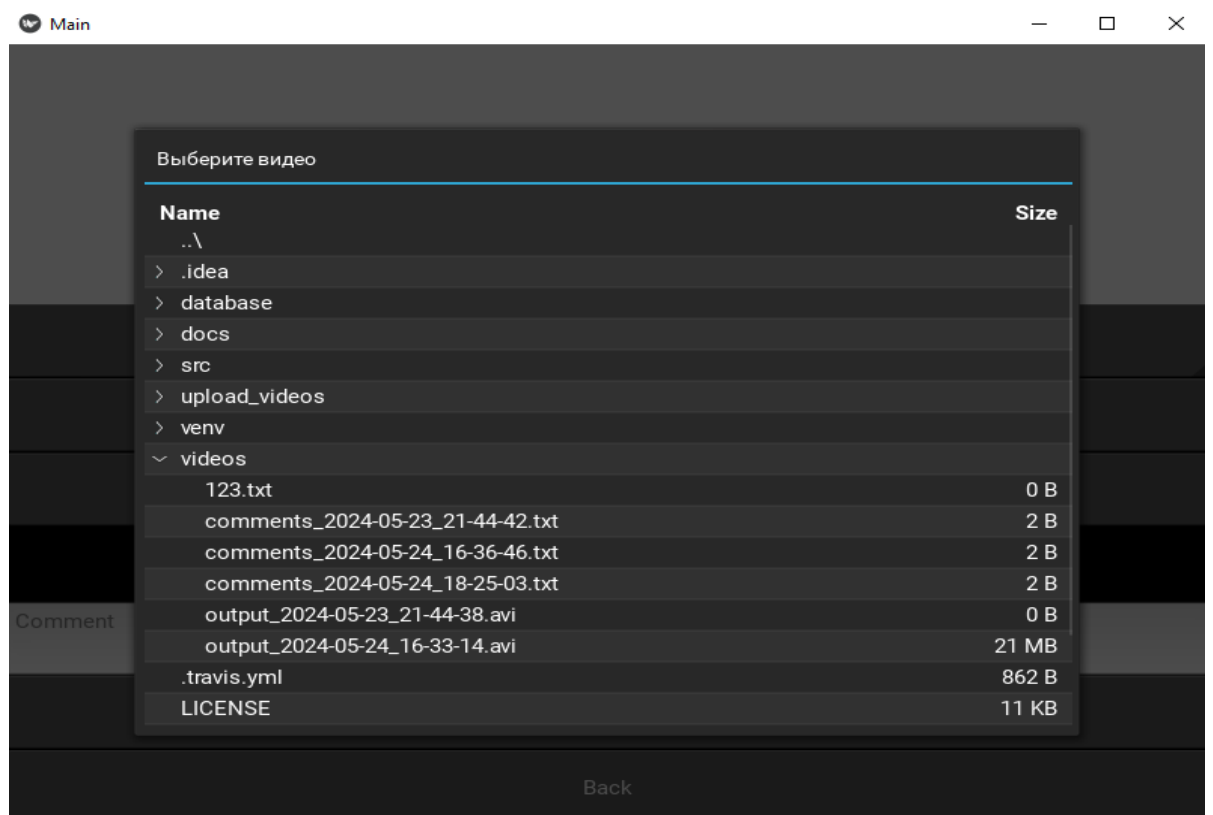


Рисунок 13 – Выбор видео для загрузки

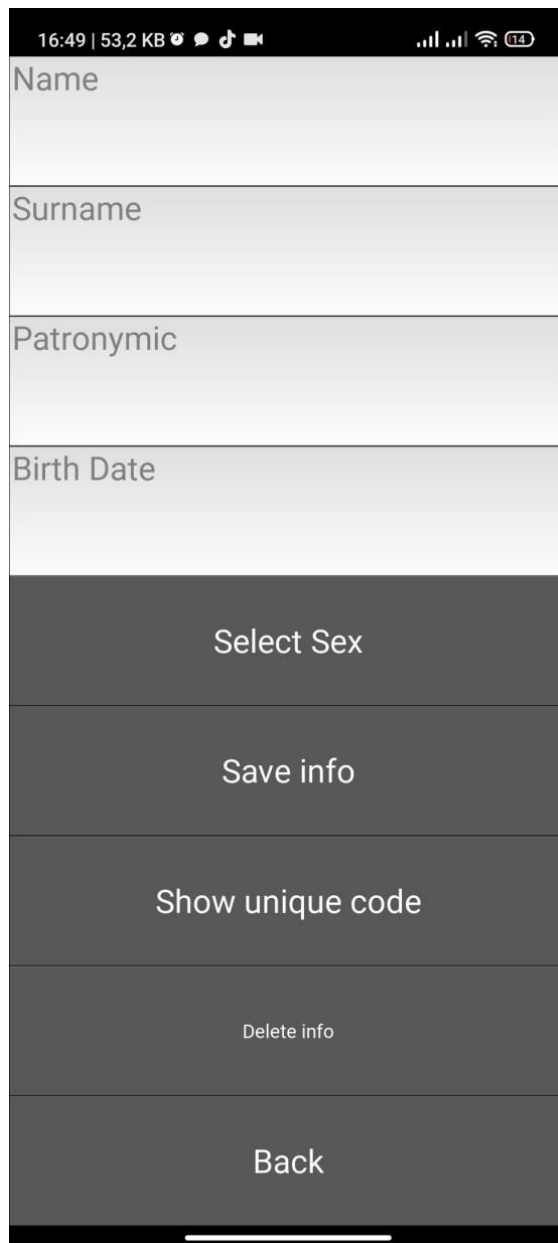


Рисунок 14 – Интерфейс приложения для Android

### **Вывод по третьей главе**

В рамках данной главы были реализованы части приложения, отвечающие за его внешний вид и выполнение задач необходимых для функционирования приложения. Реализована база данных.

#### 4. ТЕСТИРОВАНИЕ

Функциональное тестирование – процесс тестирования программного обеспечения, который проверяет соответствие программной системы функциональным требованиям и спецификациям.

Для тестирования системы была создана тестовая база данных. Она состоит из четырех пользователей: 2 пациента и 2 врача. Используя методологию функционального тестирования, была проверена работа приложения. Результаты тестирования приведены в таблице 2.

Таблица 2 – Результаты функционального тестирования

№	Название теста	Шаги	Ожидаемый результат	Тест пройден?
1	Регистрация с корректными данными	1. Нажать кнопку «Регистрация». 2. Ввести логин. 3. Ввести пароль. 4. Подтвердить пароль. 5. Выбрать роль. 6. Нажать кнопку «Зарегистрироваться».	Пользователь успешно прошел регистрацию	Да
2	Регистрация с некорректным подтверждением пароля	1. Нажать кнопку «Регистрация». 2. Ввести логин. 3. Ввести пароль. 4. Ввести некорректные данные в поле подтверждения пароля. 5. Выбрать роль. 6. Нажать кнопку «Зарегистрироваться».	Появляется сообщение о некорректном подтверждении пароля	Да
3	Регистрация с существующим логином	1. Нажать кнопку «Регистрация». 2. Ввести логин существующего пользователя. 3. Ввести пароль. 4. Подтвердить пароль. 5. Выбрать роль. 6. Нажать кнопку «Зарегистрироваться».	Появляется сообщение о том, что аккаунт уже существует	Да

Продолжение таблицы 2

№	Название теста	Шаги	Ожидаемый результат	Тест пройден?
4	Регистрация без заполнения всех полей	1. Нажать кнопку «Регистрация». 2. Ввести необходимые данные, но пропустить различные поля. 3. Нажать кнопку «Зарегистрироваться».	Уведомление о незаполненных полях	Да
5	Авторизация с корректными данными	1. Нажать кнопку «Войти». 2. Ввести личные данные. 3. Ввести пароль. 4. Выбрать роль. 5. Нажать кнопку «Войти».	Пользователь успешно авторизовался	Да
6	Авторизация с некорректными данными	1. Нажать кнопку «Войти». 2. Ввести некорректные данные логина или пароля. 3. Выбрать роль. 4. Нажать кнопку «Войти».	Уведомление системы о некорректных данных	Да
7	Добавление информации о при-ступе	1. Авторизоваться как пациент. 2. Нажать кнопку «Добавить информацию о при-ступе». 3. Ввести информацию о при-ступе. 4. Нажать кнопку «Сохранить».	Добавление информации о при-ступе в базу данных	Да
8	Добавление информации о при-ступе, если пациент не добавил личную информацию	1. Авторизоваться как пациент. 2. Нажать кнопку «Добавить информацию о при-ступе». 3. Ввести информацию о при-ступе. 4. Нажать кнопку «Сохранить».	Уведомление системы об отсутствии персональных данных пользователя	Да
9	Добавление информации о пациенте	1. Авторизоваться как пациент. 2. Нажать кнопку «Личные данные». 3. Заполнить поля. 4. Нажать кнопку «Сохранить».	Данные о пользователе сохранены в базу данных	Да

№	Название теста	Шаги	Ожидаемый результат	Тест пройден?
10	Добавление некорректной информации о пациенте	1. Авторизоваться как пациент. 2. Нажать кнопку «Личные данные». 3. Заполнить поля некорректно или оставить их пустыми. 4. Нажать кнопку «Сохранить».	Уведомление системы о некорректности введенных данных или о том, что поля пусты	Да
11	Удаление информации о пациенте	1. Авторизоваться как пациент. 2. Нажать кнопку «Личные данные». 3. Нажать кнопку «Удалить».	Удаление личной информации из базы данных	Да
12	Попытка заполнить информацию о пациенте, если она уже заполнена	1. Авторизоваться как пациент. 2. Нажать кнопку «Личные данные». 3. Заполнить поля. 4. Нажать кнопку «Сохранить».	Система уведомляет о том, что данные уже были записаны	Да
13	Запись видео с устройства	1. Авторизоваться как пациент. 2. Нажать кнопку «Добавить информацию о приходе». 3. Нажать кнопку «Видео». 4. Нажать кнопку «Начать запись». 5. Добавить комментарии в поле при необходимости. 6. Нажать кнопку «Остановить запись».	Видео записывается и сохраняется на устройство	Да
14	Попытка остановить запись видео, если запись не начиналась	1. Авторизоваться как пациент. 2. Нажать кнопку «Добавить информацию о приходе». 3. Нажать кнопку «Видео». 4. Нажать кнопку «Остановить запись».	Уведомление системы о том, что запись видео не начиналась	Да

## Продолжение таблицы 2

№	Название теста	Шаги	Ожидаемый результат	Тест пройден?
15	Загрузка видео с устройства	<ol style="list-style-type: none"> <li>1. Авторизоваться как пациент.</li> <li>2. Нажать кнопку «Добавить информацию о приходе».</li> <li>3. Нажать кнопку «Видео».</li> <li>4. Нажать кнопку «Загрузить».</li> <li>5. Выбрать видео.</li> </ol>	Видео загружается в папку пользователя	Да
16	Удаление видео с устройства	<ol style="list-style-type: none"> <li>1. Авторизоваться как пациент.</li> <li>2. Нажать кнопку «Добавить информацию о приходе».</li> <li>3. Нажать кнопку «Видео».</li> <li>4. Нажать кнопку «Проиграть видео».</li> <li>5. Нажать кнопку «Удалить видео».</li> <li>6. Выбрать видео.</li> </ol>	Видео удаляется с устройства	Да
17	Просмотр видео с устройства	<ol style="list-style-type: none"> <li>1. Авторизоваться как пациент</li> <li>2. Нажать кнопку «Добавить информацию о приходе».</li> <li>3. Нажать кнопку «Видео».</li> <li>4. Нажать кнопку «Проиграть видео».</li> <li>5. Нажать кнопку «Выбрать видео».</li> <li>6. Выбрать видео.</li> </ol>	Видео проигрывается в видеоплеере	Да
18	Добавление пациента для отслеживания	<ol style="list-style-type: none"> <li>1. Авторизоваться как врач.</li> <li>2. Нажать кнопку «Добавить пациента».</li> <li>3. Ввести уникальный номер пациента.</li> <li>4. Нажать кнопку «Добавить».</li> </ol>	Пациент добавляется врачу для отслеживания	Да
19	Добавление пациента для отслеживания с некорректным номером пациента	<ol style="list-style-type: none"> <li>1. Авторизоваться как врач.</li> <li>2. Нажать кнопку «Добавить пациента».</li> <li>3. Ввести неправильный номер пациента.</li> <li>4. Нажать кнопку «Добавить».</li> </ol>	Система уведомляет, что такого пациента нет	Да

№	Название теста	Шаги	Ожидаемый результат	Тест пройден?
20	Просмотр списка всех отслеживаемых пациентов	1. Авторизоваться как врач. 2. Нажать кнопку «Добавить пациента». 3. Нажать кнопку «Мои пациенты».	Выходит список отслеживаемых пациентов	Да
21	Просмотр статистики приступов отслеживаемого пациента	1. Авторизоваться как врач. 2. Нажать кнопку «Просмотреть статистику приступов». 3. Выбрать пациента. 4. Нажать кнопку «Показать статистику».	Выходит окно со статистикой пациента	Да
22	Просмотр статистики приступов неотслеживаемого пациента	1. Авторизоваться как врач. 2. Нажать кнопку «Просмотреть статистику приступов». 3. Выбрать не своего пациента. 4. Нажать кнопку «Показать статистику».	Выходит уведомление о том, что пациент выбран неправильно	Да
23	Просмотр статистики приступов пациента, который не вносил данные приступов	1. Авторизоваться как врач. 2. Нажать кнопку «Просмотреть статистику приступов». 3. Выбрать пациента, у которого нет данных о приступах. 4. Нажать кнопку «Показать статистику»	Выходит уведомление о том, что у пациента нет статистики	Да
24	Просмотреть уникальный код пациента	1. Авторизоваться как пациент. 2. Нажать кнопку «Личные данные». 3. Нажать кнопку «Показать уникальный код».	Пациенту отображается его уникальный код, который присваивается ему при регистрации	Да

### Вывод по четвертой главе

В третьей главе представлены результаты функционального тестирования, подтверждающего работоспособность приложения и его соответствие функциональным требованиям.

## **ЗАКЛЮЧЕНИЕ**

В ходе выполнения работы были выполнены следующие задачи.

1. Выполнен анализ предметной области и аналогичных проектов.
2. Разработана структура приложения, его функциональные составляющие.
3. Создана и внедрена база данных SQLite.
4. Реализована логика управления данными.
5. Проведено тестирование приложения.

В настоящий момент происходит доработка приложения, добавление новых функций. В будущем мы планируем разработать программное обеспечение с серверной и клиентской частью, где данные могут безопасно синхронизироваться между сервером и клиентом, а также позволять врачу отслеживать статистику по любому пациенту в любое время. Расширить возможности можно будет, добавив функцию записи видео и с мобильных устройств, что упростит пользование приложением. Также планируется провести дальнейшее тестирование на других платформах и операционных системах. В будущем так же будет реализована панель администратора.

В ходе работы были освоены методы разработки кроссплатформенных приложений с использованием фреймворка Kivy, был изучен язык программирования Python.



## ЛИТЕРАТУРА

1. Рудакова И.Г., Котов С.В., Беловва Ю.А. Эпилепсия. Основы диагностики и терапии. // Альманах клинической медицины, 2004. – №.7. – С. 303–315.
2. Глоба О.В., Кузенкова Л.М., Баранов А.А. Диагностика и лечение эпилепсии: Возможности и трудности. // МС, 2010. – №5-6. – С. 41–46.
3. Всемирная организация здравоохранения. Эпилепсия. [Электронный ресурс] URL: <https://www.who.int/ru/news-room/fact-sheets/detail/epilepsy> (дата обращения: 27.05.2024 г.).
4. Эпилепсия. [Электронный ресурс] URL: <https://gemotest.ru/info/spravochnik/zabolevaniya/epilepsiya/> (дата обращения: 27.05.2024 г.).
5. MedEx. Эпилепсия: причины, симптомы, лечение. [Электронный ресурс] URL: <https://mymedex.ru/news/epilepsiya-prichiny-simptomu-lechenie.html> (дата обращения: 27.05.2024 г.).
6. Воронкова К.В., Пылаева О.А., Косякова Е.С., Мазальская О.В., Голосная Г.С., Проваторова М.А., Королева Н.Ю., Ахмедов Т.М., Ананьева Т.В., Петрухин А.С. Современные принципы терапии эпилепсии. // Журнал неврологии и психиатрии им. С.С. Корсакова. 2010. – Т. 6. – С. 24–36.
7. Epilepsy Journal – App Store. [Электронный ресурс] URL: <https://apps.apple.com/ru/app/epilepsy-journal/id1450726202> (дата обращения: 15.03.2024 г.).
8. SeizAlarm: Seizure Detection. [Электронный ресурс] URL: <https://apps.apple.com/ru/app/seizalarm-seizure-detection/id978475280> (дата обращения: 15.03.2024 г.).
9. Щедеркина И.О., Шадрова А.А., Колтунов И.Е., Заваденко Н.Н. Детская абсансная эпилепсия, особенности оценки течения и исходов. // Эпилепсия и пароксизмальные состояния, 2017. – №. 1. – С. 69–78.

10. Невро-Мед. Классификация эпилептических приступов. [Электронный ресурс] URL: [https://www.nevromed.ru/stati/klassifikaciya\\_epilepticheskikh\\_p](https://www.nevromed.ru/stati/klassifikaciya_epilepticheskikh_p) (дата обращения: 27.05.2024 г.).
11. Что такое JavaScript (JS)? [Электронный ресурс] URL: <https://aws.amazon.com/ru/what-is/javascript/> (дата обращения: 27.05.2024 г.).
12. Кроссбраузерная совместимость: Обеспечение согласованности при разработке на JavaScript. [Электронный ресурс] URL: <https://open.zerba.academy/krossbrauzernaya-sovmestimost-razrabotka-javascript/> (дата обращения: 15.03.2024 г.).
13. Что такое Java? [Электронный ресурс] URL: <https://aws.amazon.com/ru/what-is/java/#:~:text=Java%20-%20это%20многоплатформенный%2C%20объектно-,большими%20данными%20и%20серверных%20технологий> (дата обращения: 27.05.2024 г.).
14. Бреслав А. JetBrains: «Прямой монетизации языков программирования, скорее всего, уже не будет». [Электронный ресурс] URL: <https://www.forbes.ru/tehnologii/345781-andrey-breslav-jetbrains-pryamou-monetizacii-yazykov-programmirovaniya-skoree> (дата обращения: 27.05.2024 г.).
15. 20 важных плюсов и минусов Java. [Электронный ресурс] URL: [https://ablison.com/ru/важные-плюсы-и-минусы-java/#Cons\\_of\\_Java](https://ablison.com/ru/важные-плюсы-и-минусы-java/#Cons_of_Java) (дата обращения: 27.05.2024 г.).
16. Python документация. [Электронный ресурс] URL: <https://docs.python.org/3/> (дата обращения: 15.03.2024 г.).
17. Java vs Python vs Kotlin: Battle of the Best [Электронный ресурс] URL: <https://www.aegisofttech.com/articles/java-vs-python-vs-kotlin-battle-of-the-best.html> (дата обращения: 15.03.2024 г.).
18. KIVY документация. [Электронный ресурс] URL: <https://kivy.org/doc/stable/> (дата обращения: 15.03.2024 г.).
19. Skillfactory Media. [Электронный ресурс] URL: <https://blog.skillfactory.ru/glossary/sqlite/> (дата обращения: 15.03.2024 г.).