**Министерство образования, науки и молодежной политики Нижегородской области**

**Государственное бюджетное профессиональное образовательное учреждение**

**«Нижегородский Губернский колледж»**

Методическая комиссия «Информатика и вычислительная техника»

Допущен к защите:

преподаватель

\_\_\_\_\_\_\_\_\_\_\_\_В.А. Авакян,

\_\_\_\_\_\_\_\_\_\_\_\_Д.А. Дергачёв,

«21» марта 2022 г.

**ОТЧЕТ**

**ПО УЧЕБНОЙ ПРАКТИКЕ**

ПМ.01 РАЗРАБОТКА МОДУЛЕЙ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ ДЛЯ КОМПЬЮТЕРНЫХ СИСТЕМ

Студент: Иконникова А.В.

Специальность, группа: 09.02.07, 44П

Нижний Новгород

2022 г.

**СОДЕРЖАНИЕ**

[ВВЕДЕНИЕ 3](#_Toc98594383)

[1 «РАЗРАБОТКА ПРОГРАММНЫХ МОДУЛЕЙ» 5](#_Toc98594384)

[1.1 Проектирование системы и работа с базой данных 5](#_Toc98594385)

[1.2 Разработка приложения 8](#_Toc98594386)

[1.3 Тестирование 15](#_Toc98594387)

[2 «РАЗРАБОТКА МОБИЛЬНЫХ ПРИЛОЖЕНИЙ» 17](#_Toc98594388)

[2.1 Анализ аспектов среды разработки и выбор платформы 17](#_Toc98594389)

[2.2 Проектирование и разработка мобильного приложения 20](#_Toc98594390)

[СПИСОК ИСПОЛЬЗУЕМЫХ ИСТОЧНИКОВ И ЛИТЕРАТУРЫ 27](#_Toc98594391)

# **ВВЕДЕНИЕ**

Программа профессионального модуля является частью программы подготовки специалистов среднего звена в соответствии с ФГОС по специальности 09.02.07 «Информационные системы и программирования».

В части освоения основного вида деятельности «Разработка модулей программного обеспечения для компьютерных систем» и соответствующих профессиональных компетенций (ПК):

ПК 1.1. Формировать алгоритмы разработки программных модулей в соответствии с техническим заданием;

ПК 1.2. Разрабатывать программные модули в соответствии с техническим заданием;

ПК 1.3. Выполнять отладку программных модулей с использованием специализированных программных средств;

ПК 1.4. Выполнять тестирование программных модулей;

ПК 1.5. Осуществлять рефакторинг и оптимизацию программного кода;

ПК 1.6. Разрабатывать модули программного обеспечения для мобильных платформ.

Цели и задачи профессионального модуля – требования к результатам освоения профессионального модуля. С целью овладения указанным видом профессиональной деятельности и соответствующим профессиональными компетенциями обучающийся в ходе освоения профессионального модуля должен:

Иметь практический опыт:

* В разработке кода программного продукта на основе готовой спецификации на уровне модуля;
* В использовании инструментальных средств на этапе отладки программного продукта;
* В проведении тестирования программного модуля по определенному сценарию;
* В использовании инструментальных средств на этапе отладки программного продукта;
* В разработке мобильных приложений.

Уметь:

* Осуществлять разработку кода программного модуля на языках низкого и высокого уровней;
* Создавать программу по разработанному алгоритму как отдельный модуль;
* Выполнять отладку и тестирование программы на уровне модуля;
* Осуществлять разработку кода программного модуля на современных языках программирования;
* Уметь выполнять оптимизацию и рефакторинг программного кода;
* Оформлять документацию на программные средства.

Знать:

* Основные этапы разработки программного обеспечения;
* Основные принципы технологии структурного и объектно-ориентированного программирования;
* Способы оптимизации и приемы рефакторинга;
* Основные принципы отладки и тестирования программных продуктов.

# **1 «РАЗРАБОТКА ПРОГРАММНЫХ МОДУЛЕЙ»**

## **1.1 Проектирование системы и работа с базой данных**

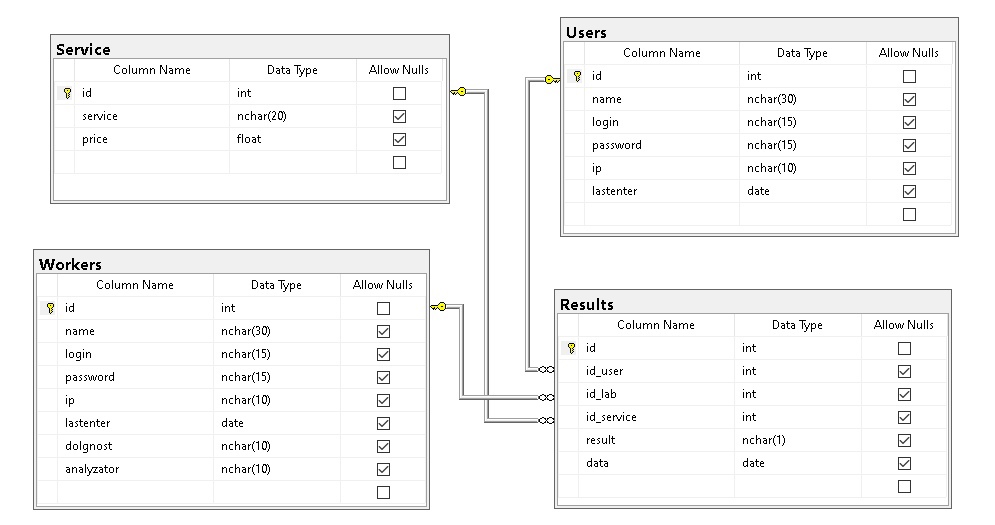
Предметная область – лабораторные анализы и их результаты. Нужно предусмотреть функционал для разных пользователей, а именно для администратора и лаборанта. Администратор может просматривать любые данные и редактировать их. Лаборант же может просматривать не все данные (например, логины и пароли пациентов и других сотрудников) и может редактировать данные только в таблице с результатами. На начальном этапе были предусмотрены данные, благодаря которым можно было воссоздать структуру. В итоге в системе получилось 4 основных блока (таблицы): сотрудники (все данные о них, в том числе данные для авторизации), пациенты (данные, которые использовались при записи результатов, а также для авторизации в мобильном приложении), услуги (список анализов и их цены, которые также нужны для описания результатов) и сами результаты, которые объединяют данные с трёх таблиц (кто сдавал анализ, кто его проводил, когда и какой результат), те же данные используются в мобильном приложении для просмотра пациентами своих данных).

Рисунок 1.1.1 – Er-диаграмма

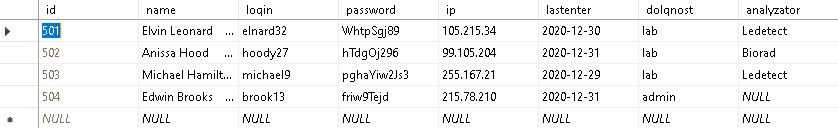
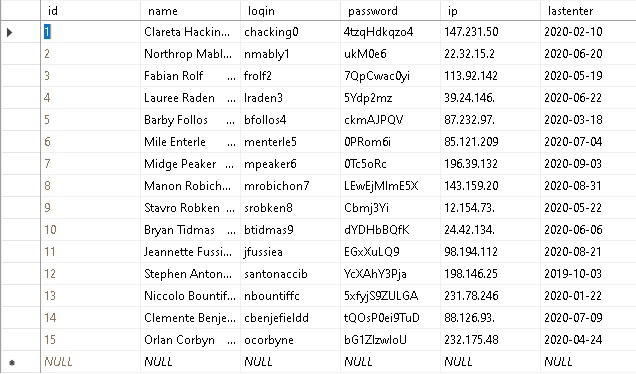


Рисунок 1.1.4 – Таблица «Услуги»

Рисунок 1.1.3 – Таблица «Пациенты»

Рисунок 1.1.2 – Таблица «Сотрудники»

****

Рисунок 1.1.5 – Таблица «Результаты»

## **1.2 Разработка приложения**

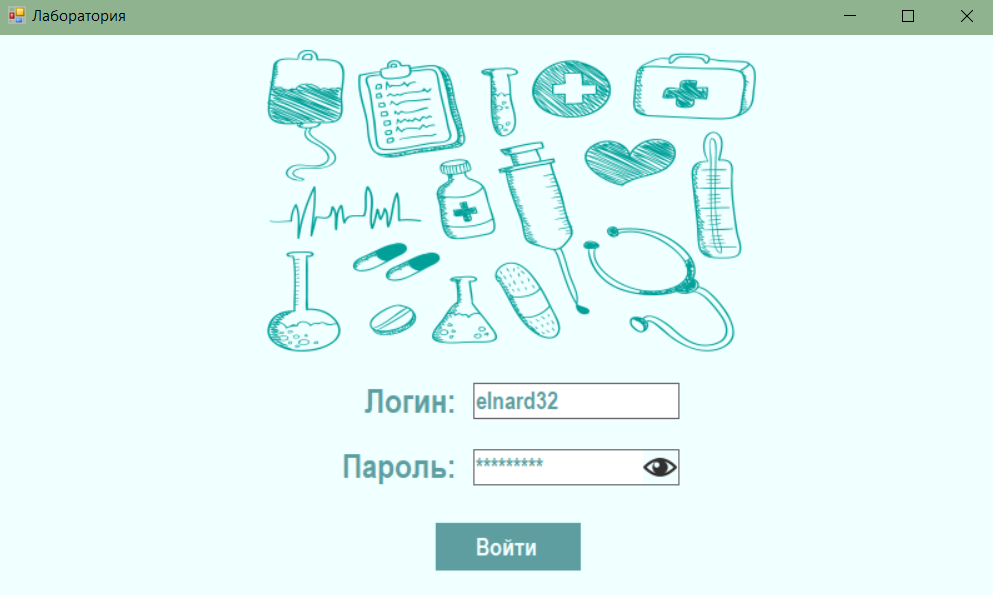
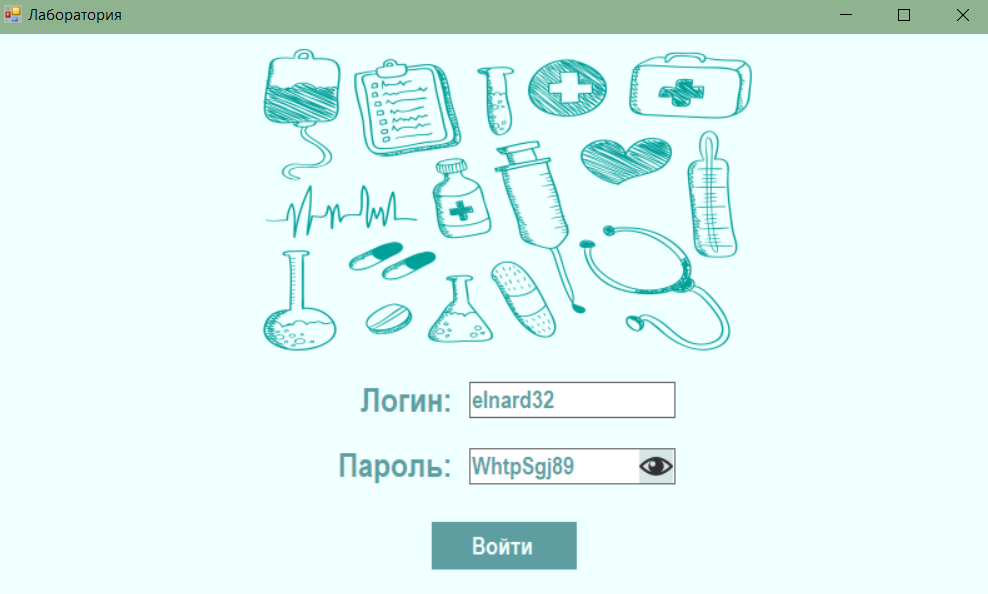
При запуске программы появляется окно авторизации, на котором есть картинка, 2 поля (для ввода логина и пароля) и кнопка вход, при нажатии на которую происходит проверка данных и открытие нужного окна (для администратора или для лаборанта).

Рисунок 1.2.1 – Форма авторизации

Также у поля для ввода пароля есть кнопка «глаз», которая при удержании показывает пароль для его проверки. При отпускании кнопки пароль опять зашифровывается звёздочками. При нажатии кнопки «Вход» логин сравнивается со значением в базе данных, проверяется пароль для конкретной записи, в глобальную переменную (в отдельном классе) заносится значение поля «Должность». Далее эта переменная будет условием редактирования таблиц и выборки при загрузке форм. Если пароль не совпадает с логином или логина нет в базе данных – появляется уведомление, побуждающее проверить и заполнить поля снова.

Рисунок 1.2.2 – Форма авторизации (проверка пароля)

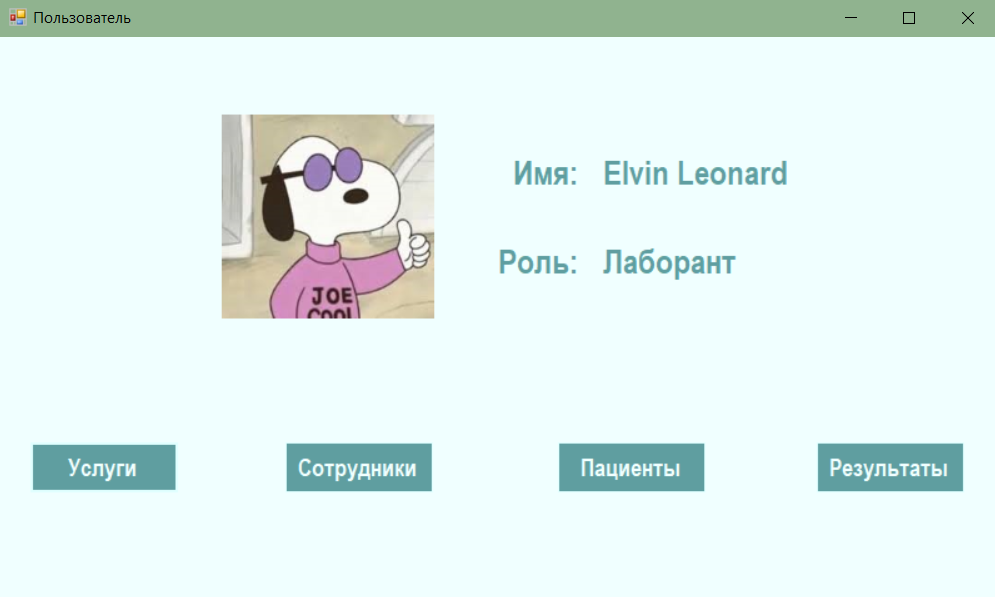
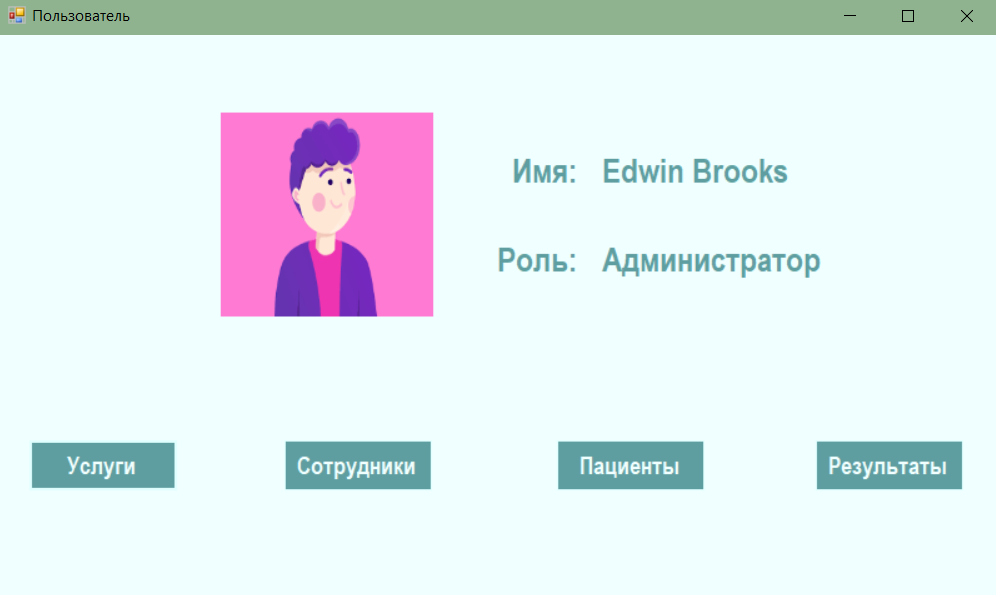
После входа открывается форма «Пользователь». В зависимости от логина и должности указывается имя пользователя и его роль. Так как форма одна, данные на ней динамически изменяются в зависимости от учётной записи. Также на форме есть аватар пользователя и 4 кнопки перехода на другие формы.

Рисунок 1.2.3 – Форма пользователя (лаборант)

Рисунок 1.2.4 – Форма пользователя (администратор)

Форма услуг для всех одинаковая, основное отличие, что администратор может добавлять/редактировать/удалять записи, а лаборант только просматривать записи. И лаборант, и администратор могут воспользоваться поиском, сортировкой и фильтрацией. Поиск осуществляется по нажатию клавиши «Enter», если поле пустое, то выборка происходит по умолчанию всех записей, после нажатия клавиши поле очищается, сортировка отображает записи по возрастанию (услуги по алфавиту) и по убыванию (услуги по алфавиту с я до а). Фильтр отображает услуги до 200 рублей и отдельно услуги более 200 рублей. Каждая услуга имеет свой код, который потом будет использоваться для сбора анализов. Кнопка «Добавить» осуществляет добавление текста из полей в таблицу (создаёт запись). Кнопка «Редактировать» по первому нажатию добавляет значения из выбранной строки в соответствующие поля для их дальнейшего редактирования. Чтобы подтвердить изменения нужно нажать на кнопку второй раз, после чего поля очищаются. Кнопка «Удалить» осуществляет удаление записи из таблицы. Все эти 3 кнопки доступны только администратору, если ими воспользуется лаборант, то появится сообщение о нехватке прав доступа.

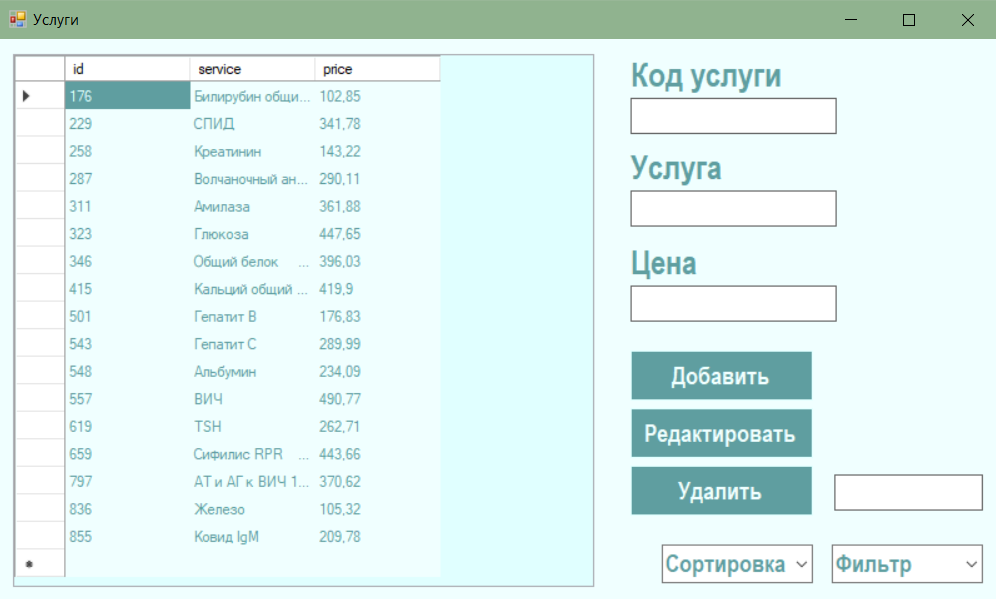
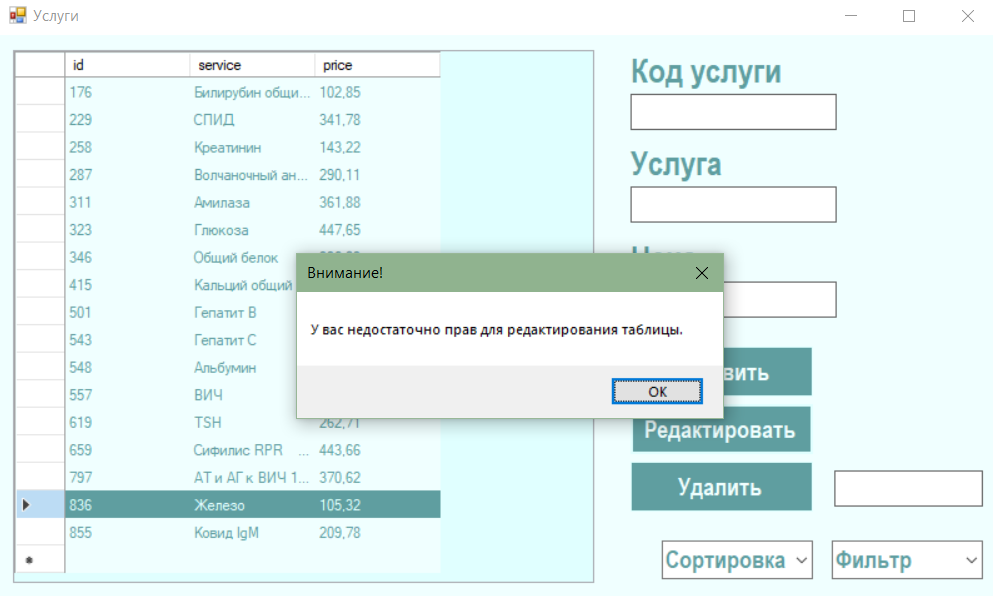
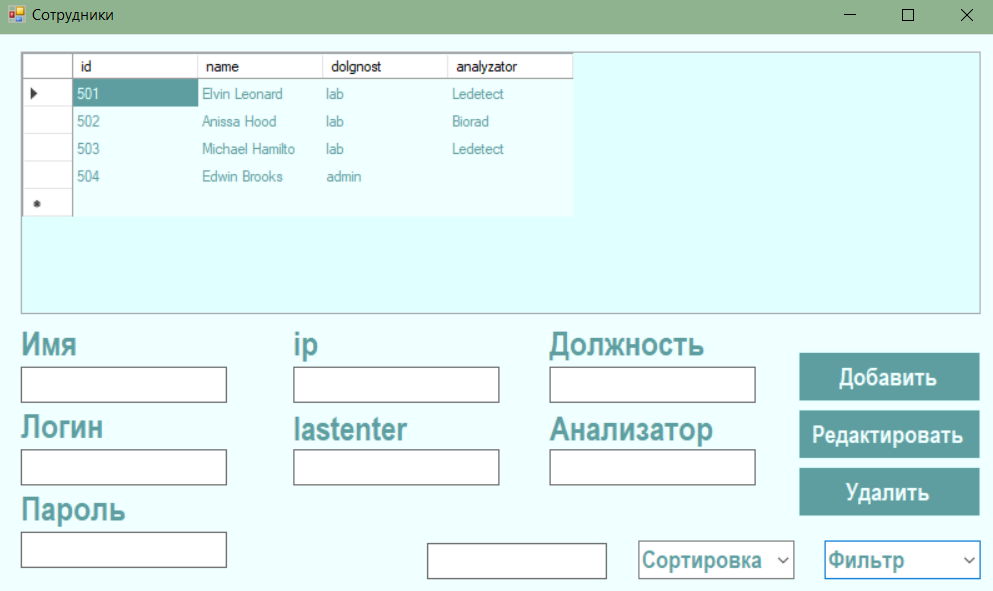
Рисунок 1.2.5 – Форма услуг

Рисунок 1.2.6 – Форма услуг, ошибка доступа

Форма «Сотрудники» имеет ту же структуру, что и услуги. На форме представлен dataGridView для отображения таблицы из базы данных, поля для добавления/редактирования записей, кнопки для добавления, редактирования и удаления записей, поле поиска данных, сортировка и фильтр. Также редактирование доступно только администратору. Сортировка осуществляется по имени (по возрастанию и убыванию), фильтр отображает либо лаборантов, либо администратора. Ещё одно из различий для двух видов пользователей – это отображение таблицы для соответствующих ролей. Если лаборант открывает эту форму, то ему доступны для просмотра только номер пользователя, его имя, должность и анализатор, на котором работает лаборант. Если форму открывает администратор – ему доступны все поля.

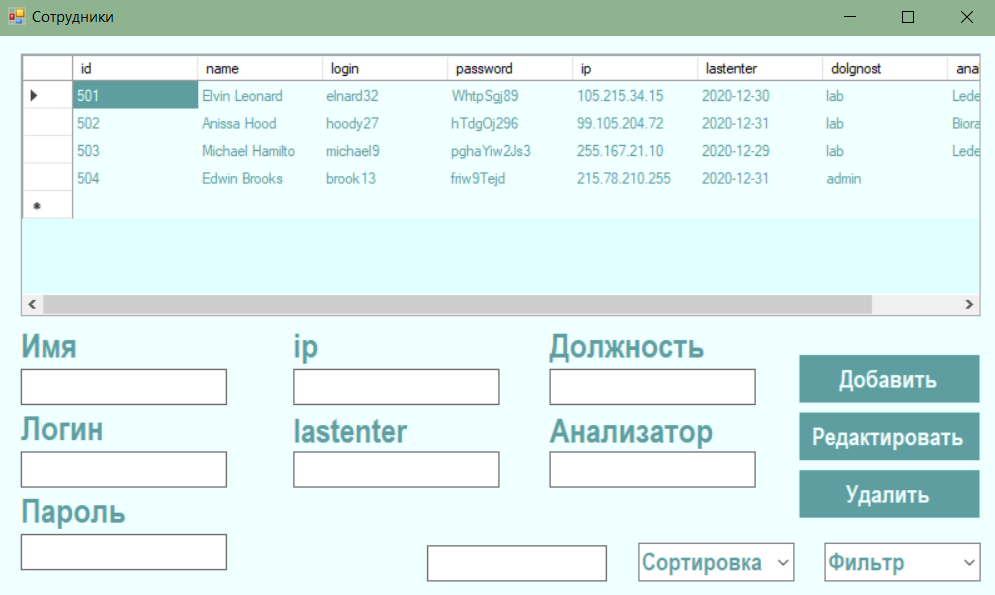
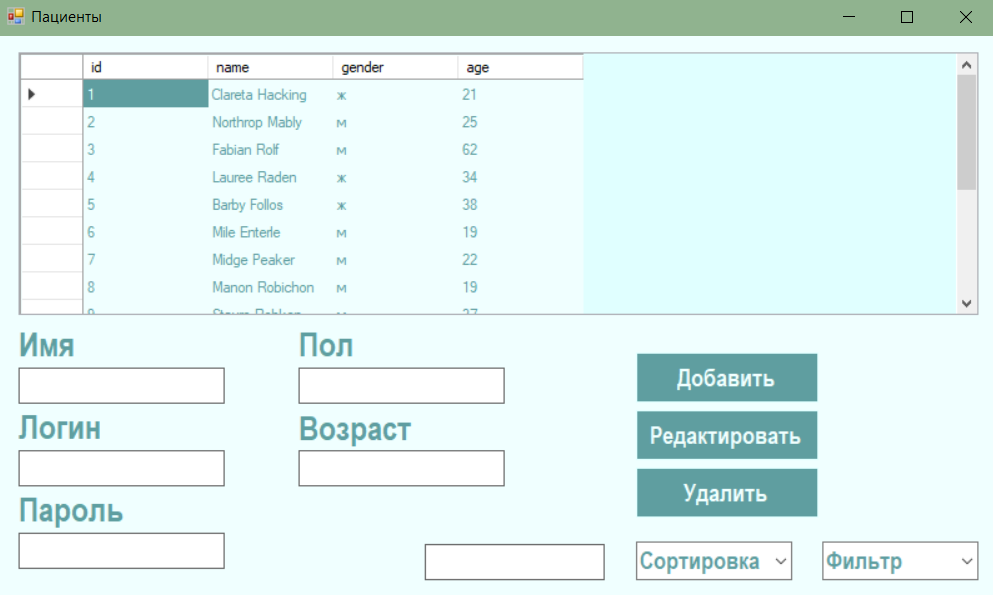
Рисунок 1.2.7 – Форма сотрудников (лаборант)

Рисунок 1.2.8 – Форма сотрудников (администратор)

Форма «Пациенты» повторяет форму «Сотрудники». Сортировка происходит по имени (по возрастанию и по убыванию), фильтр осуществляется по возрасту (до 18, от 18 до 45, 45 и более). Если на форму заходит лаборант, то ему доступны только номер пациента, его имя, пол и возраст, если администратор – то ему доступны все поля. Также лаборанту недоступно редактирование таблицы.

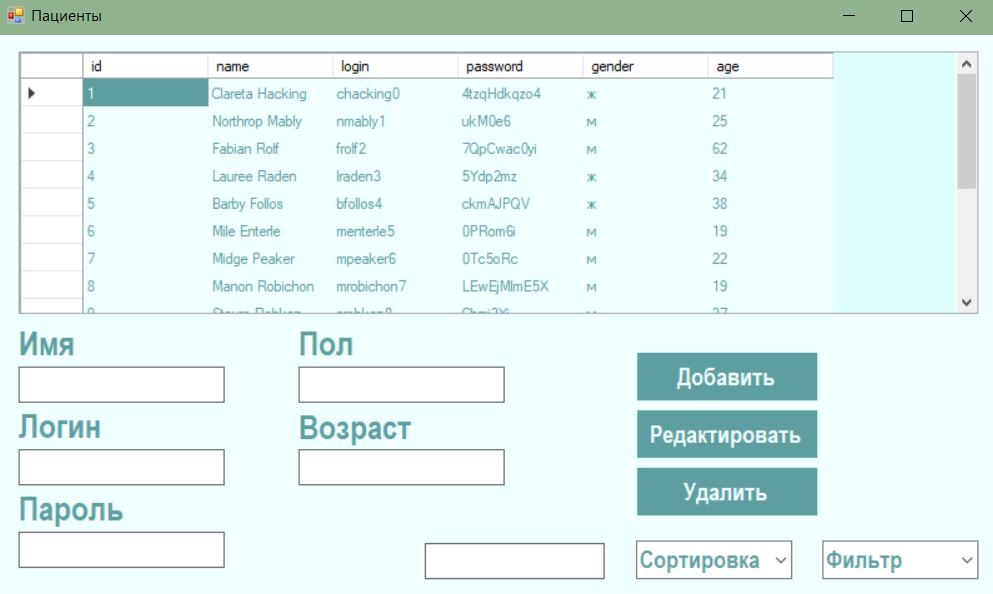
Рисунок 1.2.9 – Форма пациентов (лаборант)

Рисунок 1.2.10 – Форма пациентов (администратор)

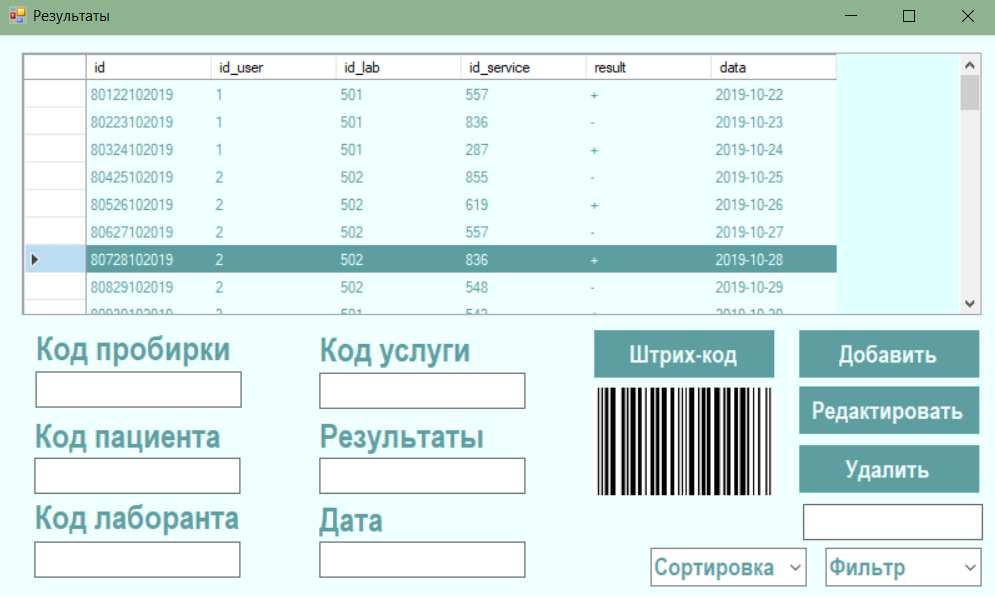
Форма результаты сохраняет коды услуги, пациента и лаборанта, а также дату анализа и его результат. На форме есть dataGridView для отображения таблицы, поля для ввода/редактирования данных, кнопки для редактирования таблицы, которые доступны как для администратора, так и для лаборанта. Также реализован поиск, сортировка по дате (по возрастанию и по убыванию) и фильтр по положительным и по отрицательным результатам. Помимо этого, осуществлён штрих код, который формируется по кнопке и выделенной строке в таблицу. Штрих код хранит в себе значение первого поля (номер пробирки из 11 цифр). Пока штрих код просто отображается на экране, но в будущем его можно будет сохранить для отчёта.

Рисунок 1.2.11 – Форма результатов

## **1.3 Тестирование**

**Аннотация теста**

|  |  |
| --- | --- |
| Название проекта | Проект Лаборатории «lab» |
| Рабочая версия | 1.0 |
| Имя тестирующего | Иконникова А.В. |
| Дата теста | 18.03.2022 |

**Тестовый пример #1:**

|  |  |
| --- | --- |
| Тестовый пример # | TC\_UI\_1: Тестовый случай пользовательского интерфейса |
| Приоритет тестирования | Средний |
| Заголовок | Воспользуйтесь фильтрацией списка |
| Краткое изложение теста | Проверить работу фильтра вместе с сортировкой и отдельно |
| Этапы теста | 1. Авторизироваться в системе. 2. Перейти на страницу «Услуги» или «Результаты». 3. Воспользоваться фильтрацией вместе с сортировкой. 4. Воспользоваться фильтрацией отдельно. |
| Тестовые данные | - |
| Ожидаемый результат | Исправная работа фильтра при остальных факторах. |
| Фактический результат | Фильтрация на этих двух формах работает только вместе с сортировкой, если сортировка не участвует, то фильтр не работает. |
| Статус | Неудачный |
| Предварительное условие | - |
| Постусловие | Состояние системы после выполнения теста должно быть стабильным. |
| Примечания | Ошибка связана с построением запроса, в котором необходимо указать условие фильтрации. |

После теста, используя отладку, было найдено место ошибки, а именно запрос. Для исправления этой ошибки нужно прописать условие в тексте запроса. После этого дефект был устранён.

**Тестовый пример #2:**

|  |  |
| --- | --- |
| Тестовый пример # | TC\_UI\_2: Тестовый случай пользовательского интерфейса |
| Приоритет тестирования | Высокий |
| Заголовок | Подтвердите страницу авторизации с действительным именем пользователя и паролем. |
| Краткое изложение теста | Проверить работу формы авторизации, проверить исключения, вызванные неправильным логином или паролем. |
| Этапы теста | 1. Ввести логин. 2. Ввести пароль. 3. Нажать кнопку «Войти». |
| Тестовые данные | Из базы данных. |
| Ожидаемый результат | При правильных введённых данных – переход на форму пользователя. При ошибке ввода появляется сообщение, которое предупреждает пользователя. |
| Фактический результат | Совпадает с ожидаемым. |
| Статус | Удачный |
| Предварительное условие | - |
| Постусловие | Состояние системы после выполнения теста должно быть стабильным. |
| Примечания | - |

# **2 «РАЗРАБОТКА МОБИЛЬНЫХ ПРИЛОЖЕНИЙ»**

## **2.1 Анализ аспектов среды разработки и выбор платформы**

В ходе разработки программного обеспечения необходимо внимательно отнестись к выбору среды разработки (IDE – Integrated Development Environment). Основная причина выбора той или иной IDE зависит от языка программирования и целевой платформы. Поскольку для разработки android приложений чаще всего используется язык программирования Java, то ниже будут представлены наиболее популярные IDE предназначенные для разработки приложений на Java для платформы Android.

На сегодняшний день существует множество Android IDE, используемых разработчиками. Одними из наиболее популярных IDE для разработки приложений для платформы Android являются: Android Studio, Eclipse, NetBeans IDE, Xamarin.Android, AIDE, Komodo, IntelliJ IDEA.

Eclipse – свободная интегрированная среда разработки модульных кроссплатформенных приложений. Однако, чтобы начать программировать под Android, необходимо установить плагин ADT (Android Development Tools). Eclipse является платформо-независимым продуктом, за исключением библиотеки SWT.

Komodo IDE – это коммерческая среда разработки, построенная на Komodo Edit, а также имеющая ряд дополнительных функций, таких как отладка кода (включая удаленную), рефакторинг, работа с VCS, HTTP Inspector, Rx Toolkit, интегативный шелл, юнит-тесты, интеграцию с другими продуктами компании (Stackato, Perl Dev Kit и TCL Dev Kit) и т.д. Данная среда разработки обеспечивает подсветку синтаксиса и сворачивание блоков для разнообразных языков.

IntelliJ IDEA представляет собой высокотехнологичный комплекс тесно интегрированных инструментов программирования, включающий интеллектуальный редактор исходных текстов с развитыми средствами автоматизации, мощные инструменты рефакторинга кода, встроенную поддержку технологий J2EE, механизмы интеграции со средой тестирования Ant/JUnit и системами управления версиями, уникальный инструмент оптимизации и проверки кода Code Inspection, а также инновационный визуальный конструктор графических интерфейсов.

Xamarin – это платформа с открытым исходным кодом, предназначенная для построения современных производительных приложений для iOS, Android и Windows с .NET. Платформа Xamarin представляет собой уровень абстракции, который обеспечивает управление взаимодействием между общим кодом и кодом базовой платформы. Приложения Xamarin.Android компилируются из языка C# в промежуточный язык (IL), который при запуске приложения претерпевает Just-in-Time-компиляцию (JIT) в машинную сборку. Приложения Xamarin.Android работают в среде выполнения Mono параллельно с виртуальной машиной среды выполнения Android (ART). Xamarin предоставляет привязки .NET к пространствам имен Android.\* и Java.\*. Среда выполнения Mono обращается к этим пространствам имен с использованием управляемых вызываемых оболочек (MCW) и предоставляет среде выполнения ART вызываемые программы-оболочки Android (ACW), благодаря чему обе среды могут вызывать код друг друга.

NetBeans IDE – бесплатная интегрированная среда разработки с открытым исходным кодом для разработчиков программного обеспечения. Среда предоставляет все средства, необходимые для создания профессиональных десктоп приложений, корпоративных, мобильных и веб-приложений на платформе Java, а также C/C++, PHP, JavaScript, Groovy и Ruby.

Android Studio – интегрированная среда разработки, основанная на программном обеспечении IntelliJ IDEA от компании JetBrains, предназначенная для работы с платформой Android. Android Studio поддерживает различные виды сборок и генерацию нескольких .apk файлов, а также разработку приложений для Android Wear и Android TV. Кроме того, статический анализатор кода (Lint) данной среды разработки позволяет находить проблемы производительности, а также несовместимости версий. Android Studio доступна для Windows, OS X и Linux.

Проведя анализ некоторых наиболее используемых IDE, предназначенных для работы с платформой Android для разработки приложений рекомендуется использовать среду разработки Android Studio. На текущий момент данная IDE является наиболее используемой при разработке мобильных приложений, кроме того, Android Studio считается официальным средством разработки Android приложений.

## **2.2 Проектирование и разработка мобильного приложения**

Предметная область мобильного приложения – лабораторные анализы и их результаты. То есть конечный пользователь (пациент) должен иметь доступ к своему аккаунту, где будут размещены его анализы, дата и результаты этих анализов. Помимо этого, должен быть реализован поиск по записям, фильтрация по результату анализа и сортировка по дате (возрастание и убывание). Также должен быть доступ для администратора, который имеет возможность добавить, редактировать и удалить данные, а также имеет доступ к базе пользователей (логины, пароли, роли для авторизации) и может редактировать список пользователей.

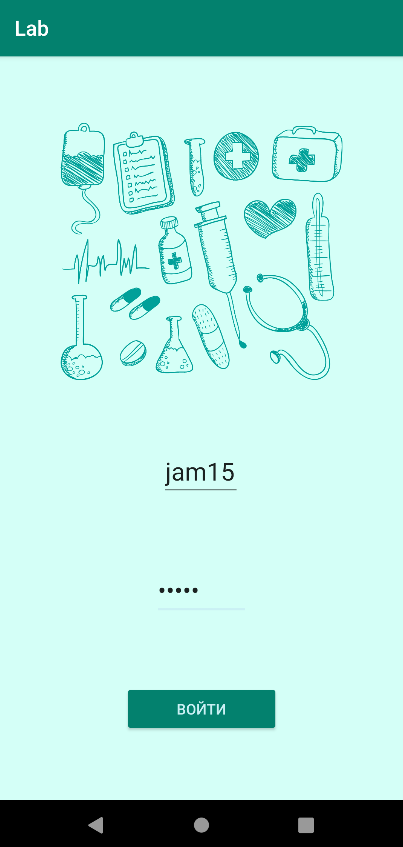
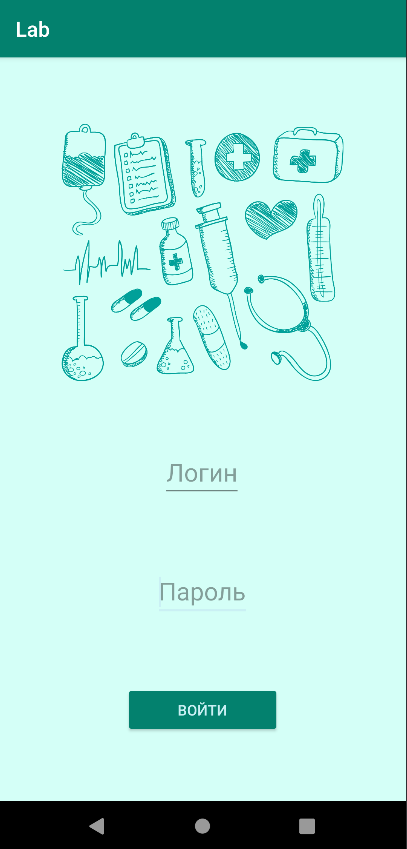
Исходя из задания можно начать проектировать систему. Первое – это авторизация. Основные объекты, которые понадобятся при построении интерфейса: EditText (для ввода текста, в данном случае логин и пароль), Button (кнопка для входа) и ImageView (для логотипа).

Рисунок 2.2.1 – Окно авторизации

Рисунок 2.2.2 – Окно авторизации 2

Также для пользователя реализованы подсказки в виде светло-серых слов на полях (Логин и Пароль) и защита пароля (черные круги вместо символов).

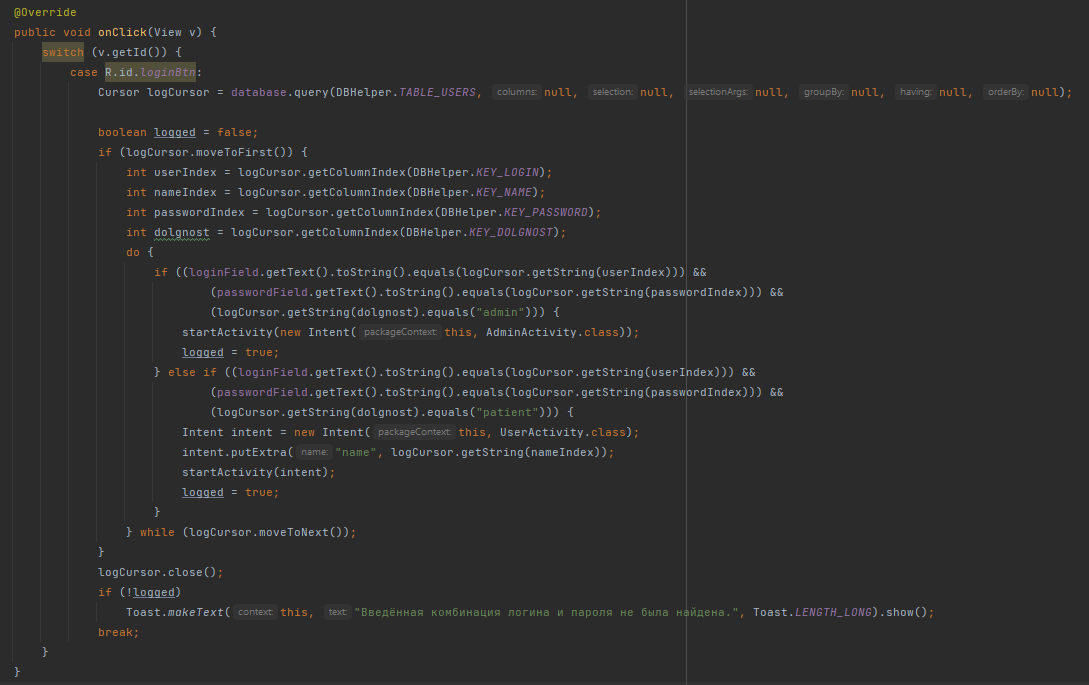
Алгоритм кода построен таким образом, что значение из поля «Логин» сравнивается со значением в базе данных, сравниваются пароли и уже из базы по логину и паролю становится известной роль пользователя (либо пациент, либо админ), которая помещается в отдельную переменную. Эта переменная является основой для разделения функционала.

Рисунок 2.2.3 – Код для авторизации

Если роль пользователя «Администратор», то при нажатии кнопки «Войти» пользователю открывается окно с результатами всех пациентов и возможностью их редактировать, также есть кнопка «Назад», которая возвращает на форму авторизации и кнопка «База пользователей», которая открывает окно с данными каждого пользователя. Также на форме есть поля ввода для добавления или редактирования записей, которые тоже имеют подсказки. Кнопка «Удалить» очищает всю базу данных. Для того, чтобы удалить какую-то конкретную запись из таблицы предусмотрена кнопка «Удалить» напротив каждой записи в таблице. Для вывода данных из таблицы используется инструмент «TableLayout».

Рисунок 2.2.4 – Окно администратора

Кнопки запрограммированы таким образом, что для добавления не нужно прописывать id, он ставится автоматически, а редактирование как раз осуществляется по id. То есть, если такой id уже есть в базе, то данные изменяются в соответствии с заполненными полями. После нажатия кнопки добавить/редактировать поля очищаются.

Рисунок 2.2.5 – Код кнопки «Добавить»

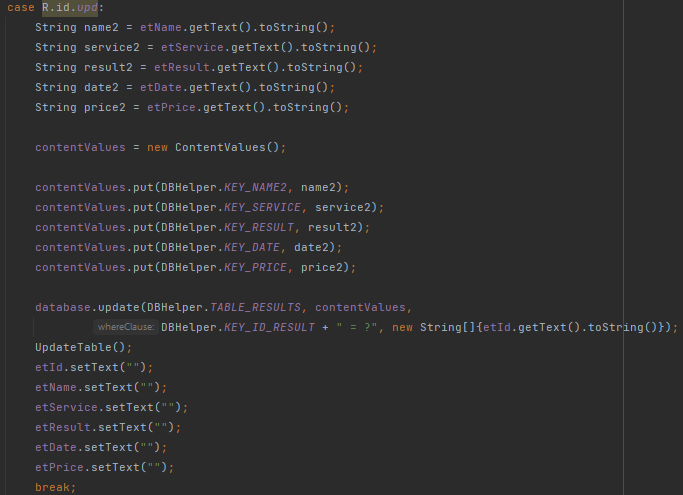
Также на каждой форме есть метод «UpdateTable», который используется после нажатия кнопок, связанных с редактированием базы.

Рисунок 2.2.6 – Код кнопки «Редактировать»

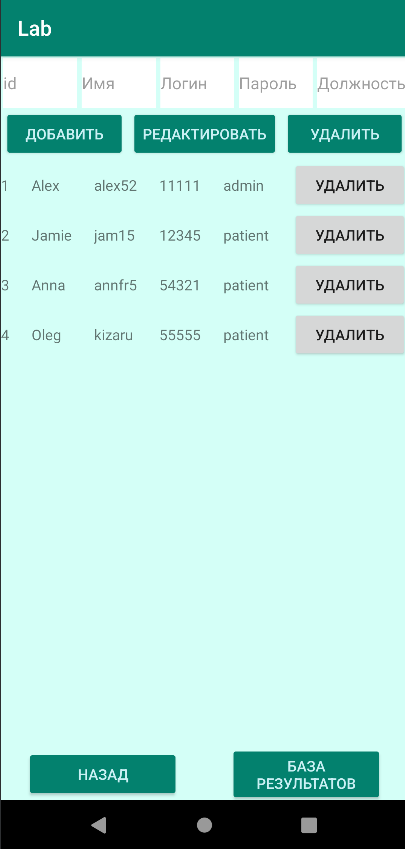
При нажатии кнопки «База пользователей», администратор переходит на форму редактирования и просмотра пользователей. По функционалу она идентична окну «База результатов», только работа происходит с другой таблицей.

Рисунок 2.2.7 – Окно администратора (редактирование пользователей)

Если же роль пользователя «Пациент», то ему открывается окно с результатами его анализов. Для этого в коде предусмотрена переменная, которую можно использовать в нескольких активностях, а именно имя, по которому осуществляется выборка в таблицу. Благодаря этому пользователь видит только свои анализы, а результаты других – нет. Также у пользователя нет инструментала для редактирования таблицы, но вместо этого у него есть поле для поиска и фильтр и сортировка, а также кнопка «Назад», которая возвращает на форму авторизации. Пользователю видны его имя, название анализа, дата анализа и цену, по которой осуществлялась услуга. Фильтр работает на выборку через результат, то есть можно выбрать либо все положительные результаты, либо все отрицательные. Сортировка же выводит все записи либо по возрастанию цены, либо по убыванию. Таким образом все 3 инструмента могут работать как по отдельности, так и вместе.

Рисунок 2.2.8 – Окно пользователя



Рисунок 2.2.12 – Окно пользователя (поиск)

Рисунок 2.2.11 – Окно пользователя (поиск)

Рисунок 2.2.10 – Окно пользователя (цена по возрастанию)

Рисунок 2.2.9 – Окно пользователя (отрицательные результаты)



Рисунок 2.2.13 – Окно пользователя (сортировка и фильтр одновременно)

# **СПИСОК ИСПОЛЬЗУЕМЫХ ИСТОЧНИКОВ И ЛИТЕРАТУРЫ**

Монографическая литература:

1. Влацкая И.В. Проектирование и реализация прикладного программного обеспечения [Электронный ресурс] учебное пособие / И.В. Влацкая, Н.А. Заельская, Н.С. Надточий. — Электрон. текстовые данные. — Оренбург: Оренбургский государственный университет, ЭБС АСВ, 2020. — 119 c. — 978-5-7410-1238-3. — Режим доступа: http://www.iprbookshop.ru/54145.html (дата обращения 28.08.2020)
2. Федорова Г.Н. Разработка программных модулей программного обеспечения для компьютерных систем: учебник. Среднее профессиональное образование, профессиональная подготовка / Г. Н. Федорова. – М.: Академия, 2020.–336 с.
3. Журавлёва И.А. Системное и прикладное программное обеспечение [Электронный ресурс] лабораторный практикум / И.А. Журавлёва, П.К. Корнеев. — Электрон. текстовые данные. — Ставрополь: Северо-Кавказский федеральный университет, 2017. — 132 c. — 2227-8397. —
4. Подбельский В. Язык C#. Базовый курс. Издание второе, переработанное и дополненное. Издательство: Финансы и статистика, 2020.–408 с.

Интернет-ресурсы:

1. Выбор среды разработки для создания android приложений – Режим доступа <https://infourok.ru/vibor-sredi-razrabotki-dlya-sozdaniya-android-prilozheniy-3028586.html>.
2. SQLite и Android – Режим доступа <http://developer.alexanderklimov.ru/android/sqlite/cathouse2.php>.