**Министерство образования Иркутской области**

Государственное бюджетное профессиональное

образовательное учреждение Иркутской области

«Иркутский авиационный техникум»

(ГБПОУИО «ИАТ»)

|  |  |
| --- | --- |
| КП.09.02.03.22.193.03 ПЗ |  |

ИНФОРМАЦИОННАЯ СИСТЕМА

«ИДЕНТИФИКАЦИЯ ЛИЧНОСТИ»

|  |  |  |
| --- | --- | --- |
| Председатель ВЦК: | \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  (подпись, дата) | (М.А. Кудрявцева) |
| Руководитель: | \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  (подпись, дата) | (М.А. Кудрявцева) |
| Студент: | \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  (подпись, дата) | (И.А. Громыко) |

Иркутск 2022

**Содержание**

ВВЕДЕНИЕ 3

1. Описание предметной области 5

2. Анализ инструментальных средств разработки 6

3. Техническое задание 11

4. Проектирование ИС 12

4.1. Структурная схема ИС 12

4.2. Функциональная схема ИС 16

4.3. Проектирование базы данных 19

4.4. Проектирование интерфейса 22

5. Разработка ИС 24

5.1. Разработка интерфейса ИС 24

5.2. Разработка базы данных ИC..................................................................................25

5.3. Разработка ИС 26

6. Документирование программного продукта 32

6.1. Руководство пользователя ИС 32

Заключение 37

СПИСОК ИСПОЛЬЗУЕМЫХ ИСТОЧНИКОВ 38

Приложение А – Техническое задание 40

Приложение Б – Листинг 44

Приложение В – Листинг..............................................................................................46

**Введение**

В наше время идентификация личности актуальна как никогда. В различных компаниях и даже в магазинах или обучающих учреждениях находятся камеры для безопасности и выявлений нарушений или слежения за людьми, но что бы выявить нарушителя тратиться много времени для распознавания, но с помощью программного продукта для идентификации личности человека можно будет распознать нарушителя в считанные секунды с помощью общей базы данных о личностях.

Системы обнаружения и распознавания лиц применяются для идентификации людей в биометрических системах безопасности или как дополнительный сервис, например, в социальных сетях. Такие системы обеспечивают высокую точность распознавания, что позволяет повысить уровень безопасности какого-либо устройства или сервиса, а также предоставить пользователям удобные возможности по поиску запечатленных на фото людей.

Целью курсового проекта является разработка информационной системы «Идентификация личности».

Основной задачей программного продукта будет предназначаться для идентификации личности с фотографий для обозначения и выявления личности в разных структурах с помощью базы данных.

В процессе выполнения курсового проекта необходимо решить следующие задачи для реализации программного продукта:

* провести анализ предметной области и представить его описание;
* выполнить анализ программных продуктов, используемых при разработке информационной системы (далее – ИС) и обосновать выбор платформы для разработки программного продукта;
* разработать техническое задание;
* выполнить проектирование информационной системы;
* спроектировать базу данных;
* разработать структуру и интерфейс;
* разработать информационную систему;
* провести тестирование и отладку программного продукта;
* написать руководство пользователя для информационной системы.

1. **Описание предметной области**

Распознавание лиц – это одни из наиболее перспективных методов биометрической бесконтактной идентификации человека. В наше время технология распознавания лиц наиболее часто используется в системах видеонаблюдения, контроля доступа в предприятиях, или на разнообразных мобильных и облачных платформах.

Работа системы распознавания личности начинается с получения изображения и в дальнейшем обработка его по алгоритму:

* обработка фотографии;
* поиск лиц фотографии;
* выделение найденных лиц;
* сохранение лиц в отдельные файлы;
* поиск лиц в базе данных;
* вывод информации о найденных лиц.

Распознавание лиц используется в разных отраслях и технологиях. Данная технология уже используется в устройствах для распознавания и идентификации для авторизации пользователя или идентификации личности в органах исполнительной власти или же организациях.

1. **Анализ инструментальных средств разработки**

Инструменты разработки программного продукта определяют будущий результат.

Проектировать структуру программного продукта удобно через Draw.io его дизайн – через онлайн-сервис Figma. Приложение будет состоять из двух частей – клиентская и серверная. Для реализации клиентской части отлично подойдет инструмент для разработки программного продукта как PyQt6, a проектирование базы (далее – БД) данных будет реализована с помощью MySQL Workbench.

MySQL Workbench — инструмент для визуального проектирования [баз данных](https://ru.wikipedia.org/wiki/%D0%91%D0%B0%D0%B7%D1%8B_%D0%B4%D0%B0%D0%BD%D0%BD%D1%8B%D1%85), интегрирующий проектирование, моделирование, создание и эксплуатацию БД в единое бесшовное окружение для системы баз данных MySQL. В проекте используется для создания ER-диаграммы БД.

Draw.io –– это удобное бесплатное онлайн-приложение для создания диаграмм для рабочих процессов, BPM, организационных, сетевых диаграмм, блок-схемм (флоучарты), UML и принципиальных электросхем. В проекте используется для создания прототипа страниц.

Figma – бесплатный удобный онлайн-сервис для дизайнеров, веб-разработчиков и маркетологов. Он предназначен для создания прототипов сайтов или приложений, иллюстраций и векторной графики. В редакторе можно настроить совместную работу, вносить и обсуждать правки, причём как в браузере, так и через приложение на компьютере. Популярен, в частности, для разработки прототипа и дизайна сайта или приложения.

Программный продукт будет содержать в себе информацию – её необходимо хранить, изменять, структурировать и использовать. Это реализуется благодаря базе данных. Были рассмотрены следующие варианты реализации СУБД:

1. MySQL.

2. PostgreSQL.

MySQL – свободная реляционная система хранения и управления базами данных. Разработку и поддержку MySQL осуществляет корпорация Oracle, получившая права на торговую марку вместе с поглощённой Sun Microsystems, которая ранее приобрела шведскую компанию MySQL AB. Продукт распространяется как под GNU General Public License, так и под собственной коммерческой лицензией. Помимо этого, разработчики создают функциональность по заказу лицензионных пользователей. Именно благодаря такому заказу почти в самых ранних версиях появился механизм репликации. На сегодняшний день является самой популярной серверной базы данных (далее – БД), за счёт своей простоты, скорости работы и внушительного функционала.

PostgreSQL – свободная объектно-реляционная система хранения и управления базами данных. Существует в реализациях для множества UNIX-подобных платформ, включая AIX, различные BSD-системы, HP-UX, IRIX, Linux, macOS, Solaris/OpenSolaris, Tru64, QNX, а также для Microsoft Windows. Работает только на одном движке – Storage Engine. Все таблицы представлены в виде объектов, они могут наследоваться, а все действия с таблицами выполняются с помощью объектно-ориентированных функций. Обладает открытым исходным кодом, при этом старается максимально соответствовать стандарту SQL.

Для наглядности сравнения вариантов реализации базы данных была составлена таблица 1.

Таблица 1 – Сравнение средств реализации базы данных

|  |  |  |
| --- | --- | --- |
| Название СУБД | MySQL | PostgreSQL |
| Большое кол-во типов данных | + | + |
| Популярность | + | - |
| Отказоустойчивость | - | + |
| Не требует удаленного сервера | - | - |
| Простота использования | + | - |
| Портативность | - | - |

Таким образом, в качестве базы данных для будущего продукта была выбрана MySQL, так как она предоставляет весь необходимый функционал для разработки продукта, и при этом она проста в изучении и овладении, достаточно быстрая, предоставляет возможность гибкой настройки.

Для взаимосвязи баз данных и северной части продукта необходимо использовать серверный язык, им выступил язык программирования – Python.

Python — высокоуровневый язык программирования общего назначения с динамической строгой типизацией и автоматическим управлением памятью, ориентированный на повышение производительности разработчика, читаемости кода и его качества, а также на обеспечение переносимости написанных на нём программ. Язык является полностью объектно-ориентированным – всё является объектами. Необычной особенностью языка является выделение блоков кода пробельными отступами. Синтаксис ядра языка минималистичен, за счёт чего на практике редко возникает необходимость обращаться к документации. Сам же язык известен как интерпретируемый и используется в том числе для написания скриптов.

Таким образом, Python будет более лучшим вариантом, так как он имеет большое количество библиотек и имеет инструменты для работы с БД. В сравнении с языком Php, у Python работа с программными модулями проще, а также его сообщество более активное.

Для разработки программного продукта рассмотрены следующие инструментальные средства разработки программных продуктов:

1. PyCharm.

2. Spyder.

3. Visual Studio Code.

PyCharm – одна из лучших полнофункциональных IDE, предназначенных именно для Python. Существуют, как и версия Community(бесплатная), так и Professional (платная). PyCharm доступен на Windows, Mac OS X и Linux. PyCharm уже сразу после установки поддерживает разработку на Python — можете открыть новый файл и сразу же писать код. Вы сможете запускать и отлаживать код прямо в PyCharm. Также в IDE есть поддержка проектов и системы управления версиями. Преимущества: поддержка всего и хорошее сообщество: можно редактировать, запускать и отлаживать Python-код сразу, как скачал и установил PyCharm. Недостатки: из-за большого количества встроенных инструментов может медленно загружаться на слабых компьютерах.

Spyder — свободная и кроссплатформенная интерактивная IDE для научных расчетов на языке Python, обеспечивающая простоту использования функциональных возможностей и легковесность программной части.

Spyder является частью модуля spyderlib для Python, основанного на PyQt4, pyflakes, rope и Sphinx, предоставляющего мощные виджеты на PyQt4, такие как редактор кода, консоль Python (встраиваемая в приложения) и графический редактор переменных (в том числе списков, словарей и массивов).

Отличительной особенностью Spyder является наличие проводника переменных. Он позволяет просмотреть значения переменных в форме таблицы прямо внутри IDE. Преимущества: он бесплатный, с открытым исходным кодом и доступен на Windows, macOS и Linux. Недостатки: опытные разработчики на Python считают Spyder недостаточно функциональным для работы и предпочитают другую более функциональную IDE.

Visual Studio Code – текстовый редактор, разработанный Microsoft для Windows, Linux и macOS. Позиционируется как «лёгкий» редактор кода для кроссплатформенной разработки веб- и облачных приложений. Включает в себя отладчик, инструменты для работы с Git, подсветку синтаксиса, IntelliSense. Имеет широкие возможности для кастомизации: пользовательские темы, сочетания клавиш и файлы конфигурации. Распространяется бесплатно, разрабатывается как программное обеспечение с открытым исходным кодом, но готовые сборки распространяются под проприетарной лицензией.

Сравнение IDE для разработки программного продукта наглядно представлено в таблице 2.

Таблица 2 – Сравнение IDE для разработки программного продукта

|  |  |  |  |
| --- | --- | --- | --- |
| Название IDE | Spyder | PyCharm | Visual Studio Code |
| Распространяется бесплатно | + | - | + |
| Автоматическое сохранение | + | + | - |
| Подсказки по коду | - | + | + |
| Интеграция с системой контроля версия (GIT) | - | + | + |
| Возможность расширения функционала | - | + | + |
| Подходит под Python-разработку | + | + | + |
| Комфортное использование на слабых ПК | + | - | + |

Для разработки программного продукта в конечном итоге средой разработки была выбрана программа Visual Studio Code по ряду причин:

* Среда имеет значительную часть функционала IDE.
* Поддержка большинства ОС: Windows, Linux и MacOS.
* Имеет встроенный мощный механизм автозаполнения – IntelliSense.
* Имеет обширное количество расширений и дополнений.
* Интегрирован система контроля версий Git.
* Имеется встроенные отладчики для кода.
* Хорошая «отзывчивость».
* Встроенный инструмент для отладки.

После рассмотрения всех инструментов разработки для проекта были выбраны такие инструменты как MySQL и Python-фреймворк PyQt5, а инструменты для проектирования и разработки программного продукта – Draw.io, MySQL Workbench и Visual Studio Code.

1. **Техническое задание**

В начале разработки создавалась техническое задание, в котором указывались основные требования.

Для создания технического задания использовался стандарт ГОСТ 19.

Согласно ГОСТ 19 техническое задание должно включать следующие разделы:

Введение.

1. Общие сведения.

2. Назначение и цели создания системы.

3 Требования к системе в целом.

3.1. Требования к структуре и функционированию системы.

3.2. Требования к надежности.

3.3. Требования к безопасности.

3.4. Требования к эксплуатации, техническому обслуживанию, ремонту и

хранению компонентов системы.

4. Требования к документированию.

5. Состав и содержание работ по созданию системы.

Техническое задание на разработку информационной системы представлено в Приложение А.

**4. Проектирование ИС**

**4.1. Структурная схема ИС**

Проектирование информационной системы происходит при помощи CASE средств, которые позволяют за короткий срок создавать схемы и реляционные модели программ.



Рисунок 1 – Структурная схема ИС

На рисунке 1 изображена Use Case View, которая показывает структурную схему ИС «Идентификация личности» для пользователя.

Она отображает действия, выполняемые пользователем, программным продуктом и администратором. «Пользователь», «Программный продукт» и «Администратор» являются – актерами. «Идентификация личности», «Просмотреть базу данных», «Вставка фотографии», «Авторизация» – действия пользователя. «Проверка пользователя для авторизации», «Распознавание лица», «Идентификация

лица», «Поиск информации по лицу», «Вывод информации о личности – действия программного продукта. «Авторизация», «Внесение изменений в базу данных», «Добавление данных в базу данных», «Удаление данных в базе данных», «Выгрузка

данных из базы данных» – действия администратора.

На данной диаграмме присутствуют:

* актеры (согласно определенным на этапе анализа предметной области исполнителям);
* прецеденты (согласно определенным на этапе анализа функциональным требованиям к проекту, выделенным из деятельности в рамках предметной области и сравнительного анализа аналогов);
* связи (возможность тех или иных актеров использовать те или иные прецеденты).

Таким образом, представленная диаграмма демонстрирует взаимодействие основных объектов информационной системы и их действия.

Диаграмма взаимодействия - UML-диаграмма, на которой показаны действия, состояния которых описано на диаграмме состояний. Под взаимодействием понимается спецификация исполняемого поведения в виде координированного последовательного и параллельного выполнения подчинённых элементов - вложенных видов взаимодействия и отдельных действий, соединённых между собой потоками, которые идут от выходов одного узла ко входам другого.

Диаграммы взаимодействия используются при моделировании бизнес-процессов, технологических процессов, последовательных и параллельных вычислений. На рисунке 2 изображена диаграмма взаимодействия.

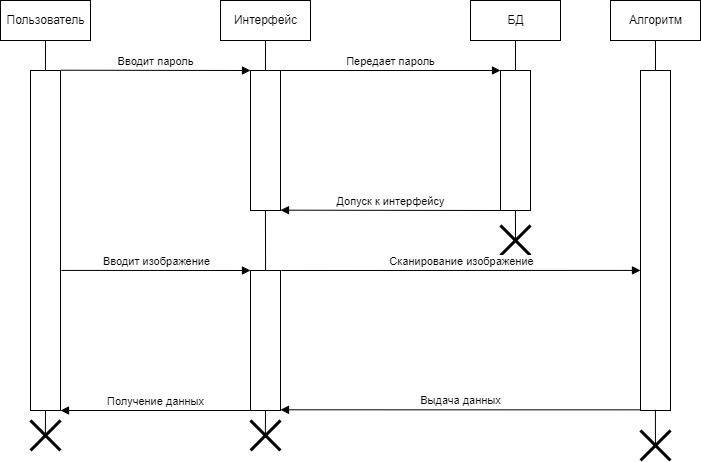


Рисунок 2 – Диаграмма взаимодействия

В итоге проектирования диаграммы взаимодействия были выделены основные возможные действия пользователя с программным продуктом.

Следующей была создана диаграмма компонентов. На этой диаграмме представляются компоненты – независимые модули ПО, скрывающие свою реализацию и взаимодействующие друг с другом через интерфейсы.

Компонент – это физически существующая часть системы, которая обеспечивает реализацию классов и отношений, а также функционального поведения моделируемой программной системы. А модуль — это часть программной системы, требующая памяти для своего хранения и процессора для исполнения. Диаграмма компонентов представлена на рисунке 3.

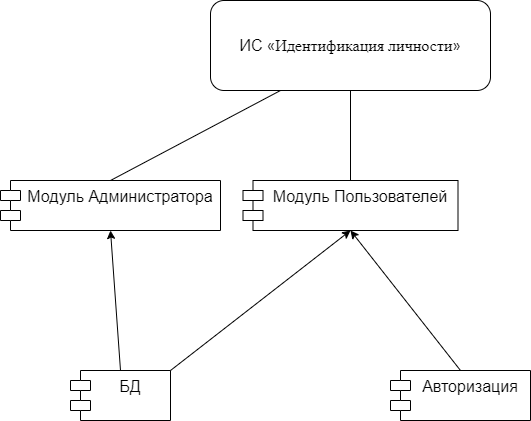


Рисунок 3 – Диаграмма компонентов

На диаграмме изображены основные компоненты и промежуточные, между пользователем и базой данных.

Далее была разработана диаграмма развертывания (рисунок 4).

Диаграмма развертывания – это диаграмма, на которой представлены узлы выполнения программных компонентов реального времени, а также процессов и объектов. Физическое представление программной системы не может быть полным, если отсутствует информация о том, на какой платформе и на каких вычислительных средствах она реализована. Для общего понимания системы в целом и была разработана эта диаграмма.

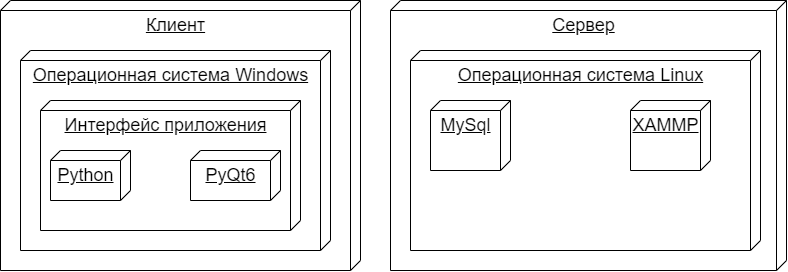


Рисунок 4 – Диаграмма развертывания

4.2 Функциональная схема ИС

На этапе проектирования создания функциональной модели, которая содержит контекстную диаграмму и диаграмму декомпозиции.

Контекстная диаграмма – это модель, представляющая систему как набор иерархических действий, в которой каждое действие преобразует некоторый объект или набор объектов.

На контекстной диаграмме на рисунке 5 показаны входные данные, управление механизм, выходные данные и функция.

В центре диаграммы показана функция информационная система «Идентификация личности». Входные данные: «Фотографии». Управление: «Алгоритм распознавания». Механизм «ПО». Выходные данные «Неисправное устройство», «Информация о человеке, «Распознанное лицо».

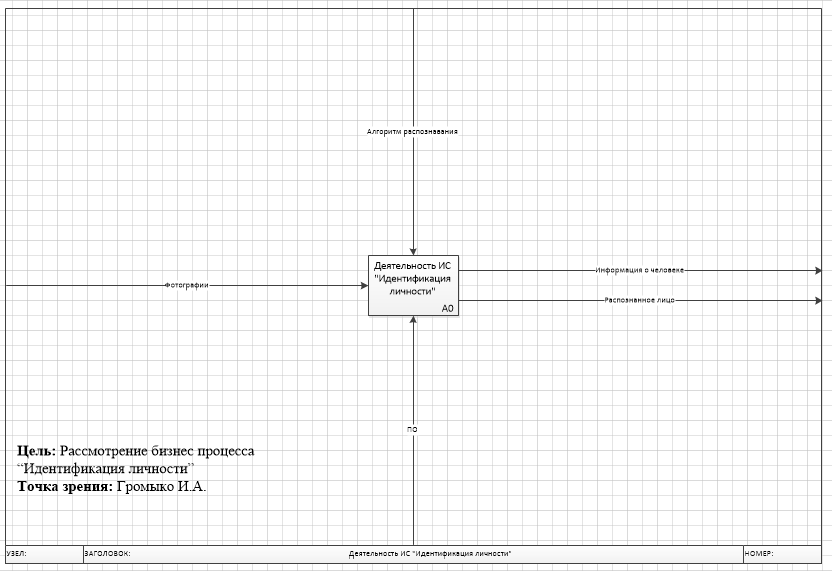


Рисунок 5 – Контекстная диаграмма

Диаграмма декомпозиции – это разбиение функции, т.е. «Авторизации и обработки данных».

На рисунке 6 показана диаграмма декомпозиций, которая расписывает функцию работы приложения. На этой диаграмме показаны такие функции как: «Сканирование фото», «Поиск лиц», «Анализ лиц на присутствие в базе данных», «Вывод информации о людях с фото из базы данных». А также на этой диаграмме присутствуют входные данные, выходные данные, управления и механизмы.

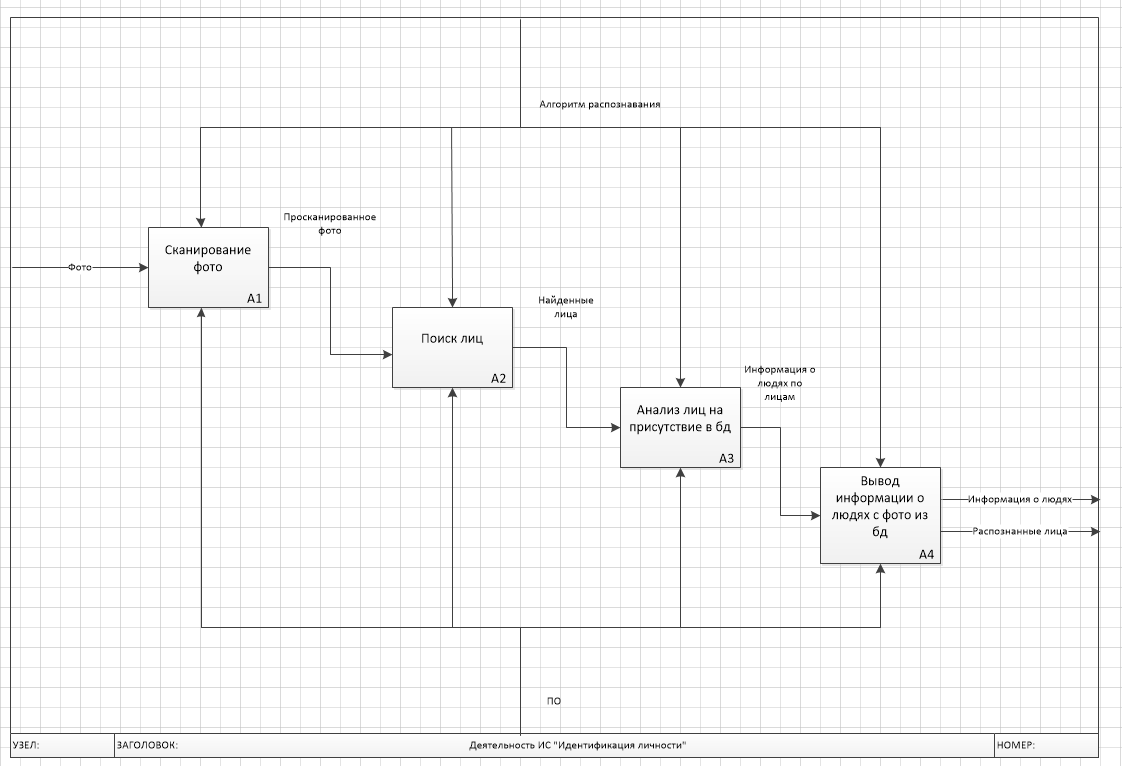


Рисунок 6 – Диаграмма декомпозиции

Таким образом, представленные диаграммы демонстрируют взаимодействие между основными функциями информационной системы «Идентификация личности» и механизмами, влияющими на выполнение той или иной функции.

Диаграмма классов – демонстрируют общую структуру иерархии классов системы, их коопераций, атрибутов, методов, интерфейсов и взаимосвязей между ними.

На рисунке 7 показана диаграмма классов, которая демонстрирует классы и атрибуты. На данное диаграмме показаны такие классы как: «Персонал», «Администратор», «Пользователь». Каждый класс имеет свои раздельные атрибуты и методы действия.

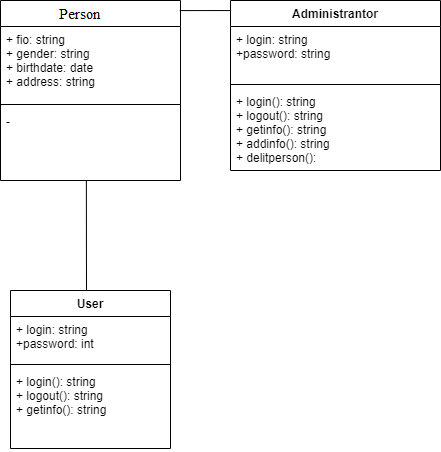


Рисунок 7 – Диаграмма классов

Также было необходимо построить диаграмму потоковых данных DFD, так как это один из основных инструментов структурного анализа и проектирования информационных систем. Диаграмма потоковых данных DFD – это нотация, предназначенная для моделирования информационных систем с точки зрения хранения, обработки и передачи данных. Данная диаграмма изображена на рисунке 8.

На диаграмме видно, откуда поступает, где обрабатывается и где храниться данные. Диаграмма помогает представить систему функционирования приложения в целом.

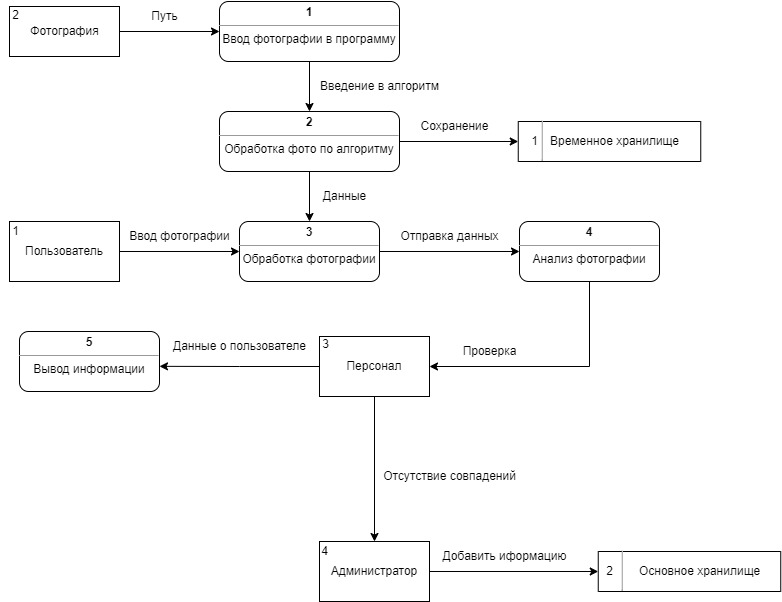


Рисунок 8 – Диаграмма потоков данных

4.3 Проектирование базы данных

Цель инфологического моделирования – обеспечение наиболее естественных для человека способов сбора и представления той информации, которую предполагается хранить в создаваемой базе данных рисунок 9.

Основными конструктивными элементами инфологических моделей являются сущности, связи между ними и их свойства (атрибуты).

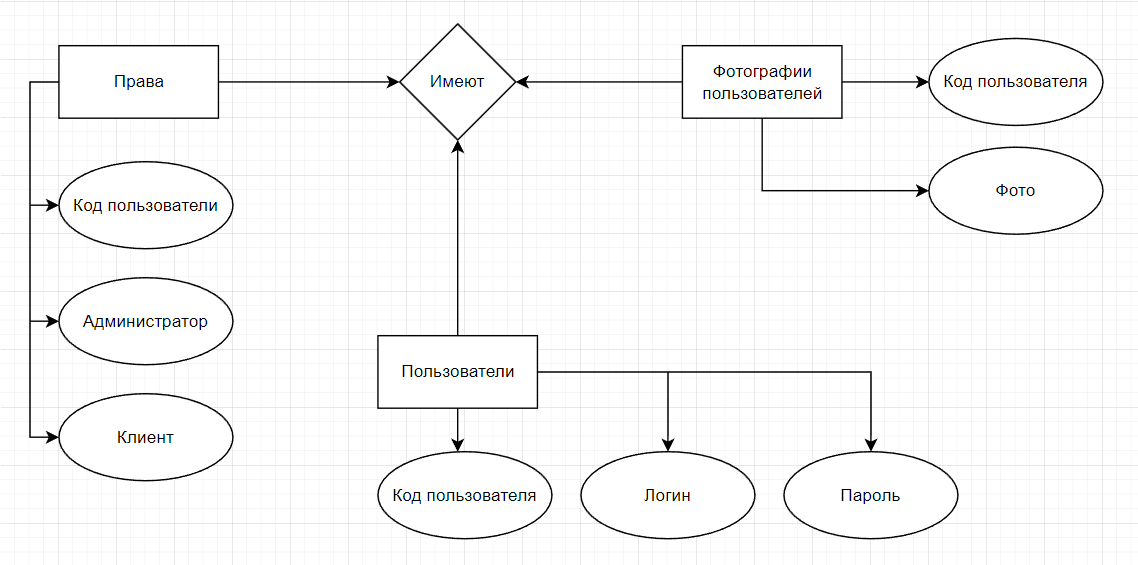


Рисунок 9 – Инфологическая модель базы данных

База данных информационной системы является важнейшей часть. Без ее реализации программный продукт не может называться информационной системой. Перед разработкой базы данных необходимо её спроектировать.

Для проектирования базы данных использовался инструмент «MySQL Workbench».

На рисунке 10 показана даталогическая модель. Даталогическая модель предполагает определение состава и взаимосвязей таблиц, отражающих содержание информационных сущностей инфологической модели в терминах конкретной СУБД.

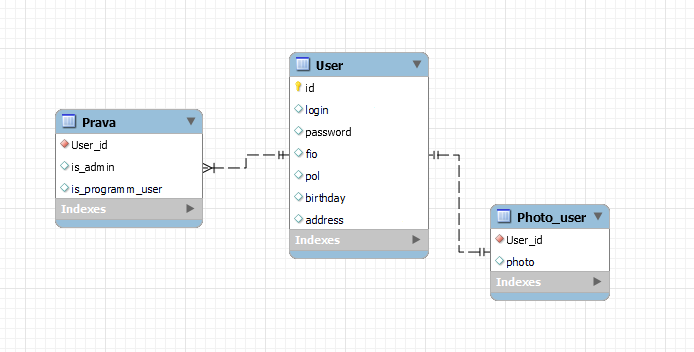


Рисунок 10 – Даталогическая модель базы данных

На рисунке 11 изображена схема данных информационной системы «Идентификация личности».

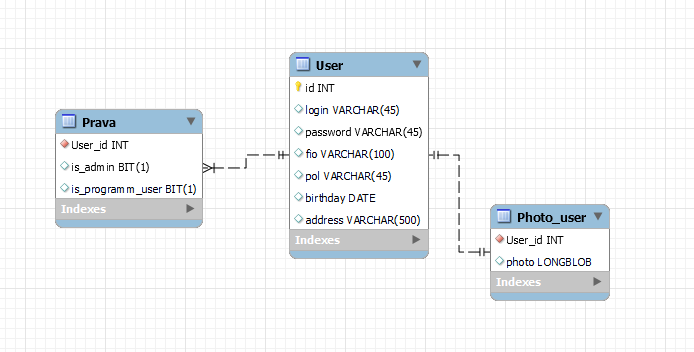


Рисунок 11 – ЕR-модель

На схеме базы данных представлены три таблицы: User, Prava Photo\_user. Таблицы: User, Prava, Photo\_user имеют связи, запрещается удалять или изменять родительский ключ, когда существует один или несколько ссылающихся на него дочерних ключей. В дочерних таблицах поля id являются вторичными ключами своих таблиц.

Таблица 3 – Таблица «пользователь»

|  |  |  |
| --- | --- | --- |
| Поле | Тип данных | Описание |
| id | INT(11) | Код пользователя |
| login | VARCHAR(45) | Логин |
| password | VARCHAR(45) | Пароль |
| fio | VARCHAR(100) | ФИО |
| pol | VARCHAR(45) | Пол |
| birthday | DATE | Дата рождения |
| address | VARCHAR(500) | Адрес проживания |

Таблица 4 – Таблица «права»

|  |  |  |
| --- | --- | --- |
| Поле | Тип данных | Описание |
| User\_id | INT(11) | Код пользователя |
| is\_admin | int(1) | Проверка на администратора |
| is\_programm\_user | int(1) | Проверка на пользователя программы |

Таблица 5 – Таблица «фото пользователя»

|  |  |  |
| --- | --- | --- |
| User\_id | INT(11) | Код пользователя |
| photo | longblob | Фото пользователя |

После завершения разработки прототипа базы данных информационной системы «Идентификация личности» получиться готовая схема, по которой будет разрабатываться база данных информационной системы.

# **4.4. Проектирование интерфейса**

Пользовательский интерфейс – это средства взаимодействия между человеком и компьютером. Говоря простыми словами, интерфейс – внешняя часть программы или устройства, с которыми работает пользователь.

Интерфейсы являются основой взаимодействия всех современных информационных систем. Если интерфейс какого-либо объекта, стандартизирован, легок в освоении пользователем, это даёт возможность модифицировать сам объект, не перестраивая принципы его взаимодействия с другими объектами.

Данный прототип интерфейса был построен в дизайнере Figma.

На рисунке 12 представлена главная страница информационной системы, на которой отображены кнопки взаимодействия и основные блоки. Информация будет выводится в назначенных для них полях.



Рисунок 12 – Главная страница

На рисунке 13 представлена главная страница с панелью просмотра найденных лиц на фото, с которой можно просмотреть информацию о сотрудниках или отдельных личностях.

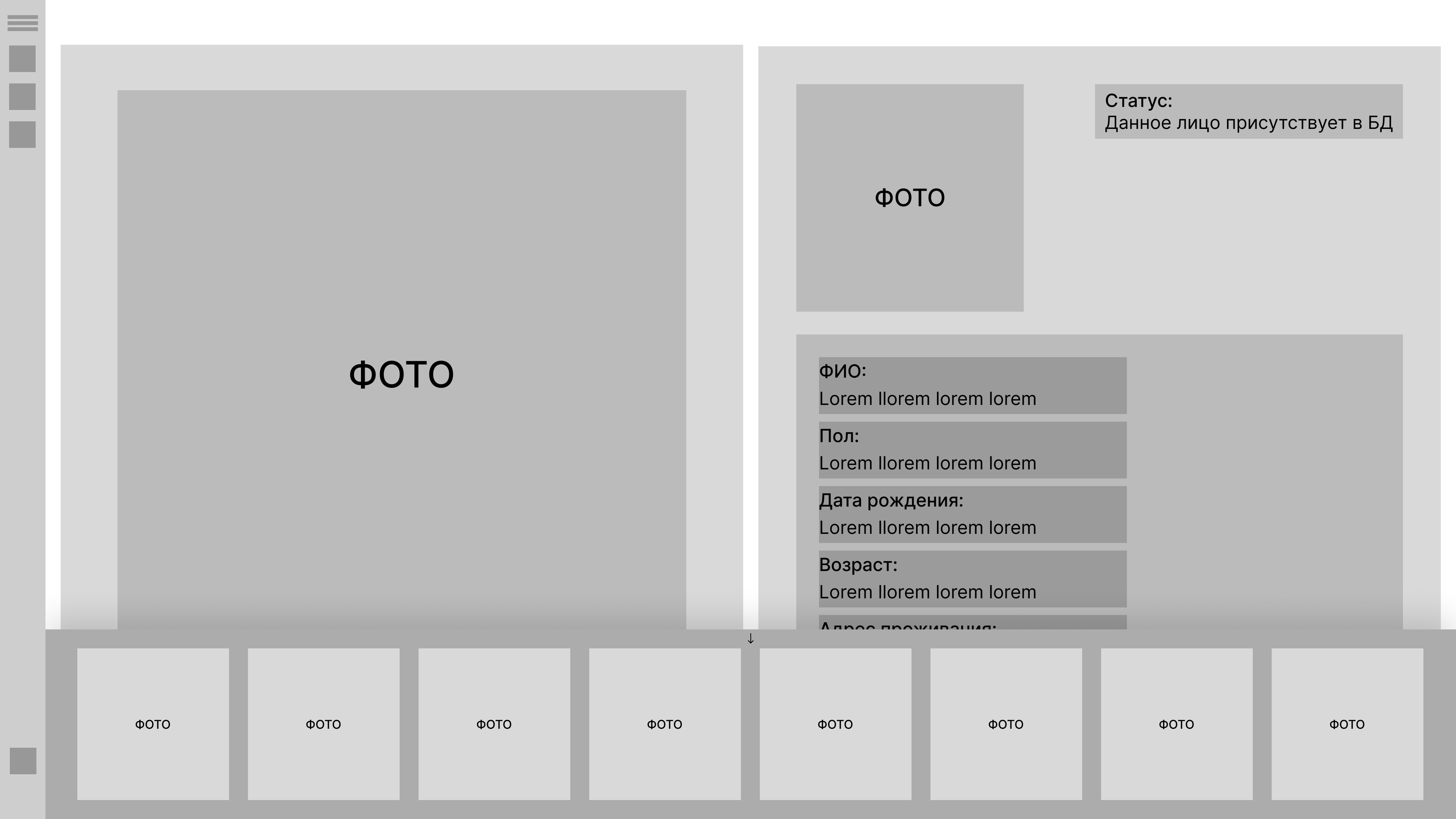


Рисунок 13 – Главная страница с просмотром лиц

Во время разработки прототипа интерфейса был учтен проведенный анализ аналогов, что помогло реализовать представленный вид прототипа информационной системы «Идентификация личности».

**5 Разработка ИС**

**5.1 Разработка интерфейса ИС**

Структура работы представляет собой набор из нескольких файлов формата \*.py с возможностью переходов между ними по средствам открытия новых форм, начиная с главной страницы и перехода в окно авторизации.

Используя, PyQt5 и QSS было разработано главное окно (рисунок 14).

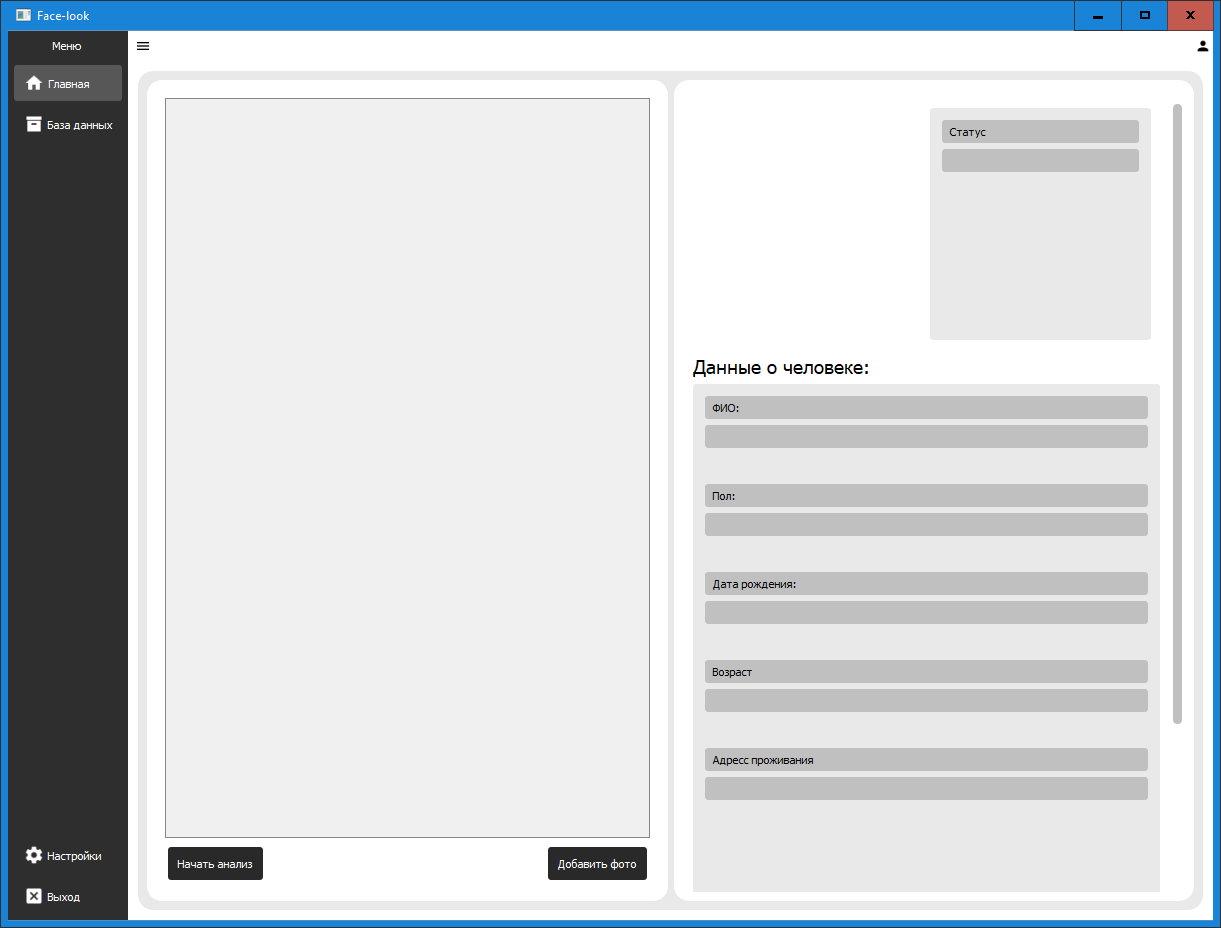


Рисунок 14 – Главное окно

Благодаря, Python, PyQt5 и QSS были реализованы стили и переключения страниц. Также была реализована страница для авторизации пользователя (рисунок 15)

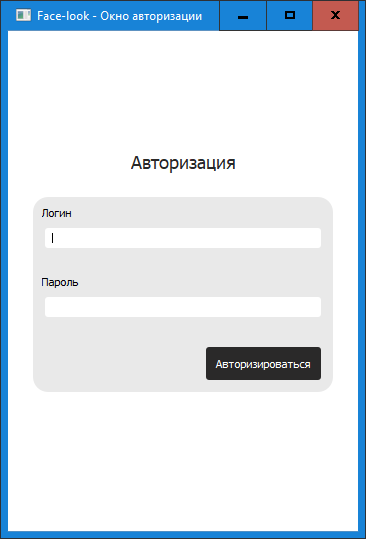


Рисунок 15 – Окно авторизации

Окно авторизации разрабатывалось таким образом что бы было понятно и удобно для пользователей.

Чтобы открыть окно с авторизацией, достаточно нажать на кнопку в правом верхнем углу основного окна приложения виде иконки человека.

# **5.2. Разработка базы данных ИС**

Разработка базы данных информационной системы «Идентификация личности» реализовывалась в СУБД MySQL.

База данных информационной системы «Идентификация личности» состоит из 3 таблиц.

Структуры таблиц соответствуют схеме базы данных из пункта 4.3.

На рисунках 15, 16, 17 представлены таблицы user, prava, photo\_user.

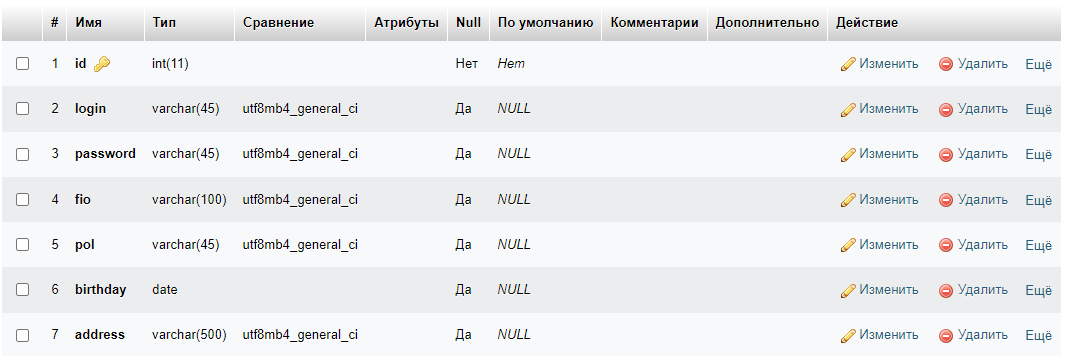


Рисунок 15 – Представление таблицы user



Рисунок 16 – Представление таблицы prava

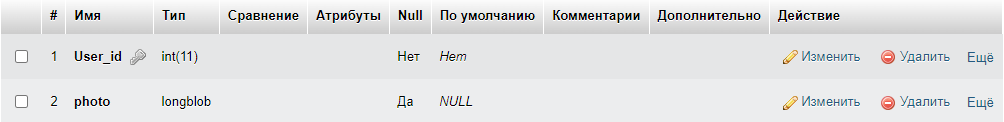


Рисунок 17 – Представление таблицы photo\_user

# **5.3. Разработка ИС**

При разработке данной информационной системы создаются файлы, из которых выстраивается репозиторий (рисунок 18).

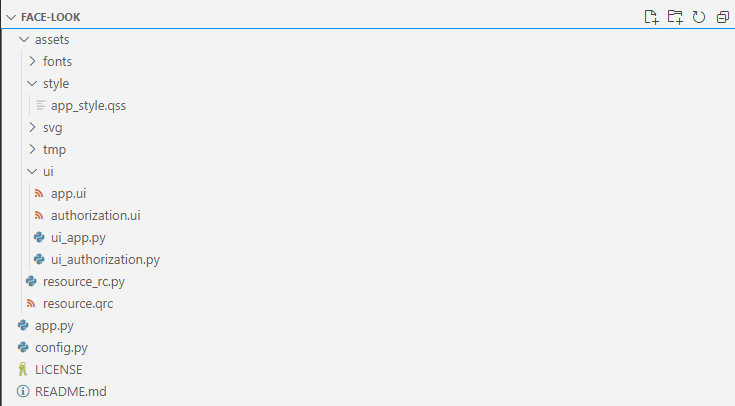


Рисунок 18 – Репозиторий проекта

На рисунке 19 представлен код вывода данных из таблицы “user” и “photo\_user” для клиента. Для начала, новая вкладка в окне приложения, в котором прописываются данные для объявления и записываются в таблицу, далее с помощью команды for и sql запросов в данную таблицу выводится информация, которая хранится в базе данных.



Рисунок 19 – Код вывода данных

На рисунке 20 представлен код отправки данных. В коде миграции используется функция, которая добавляет столбцы с определённым типом данных в базу данных (рисунок 21).



Рисунок 20 – Код для отправки данных

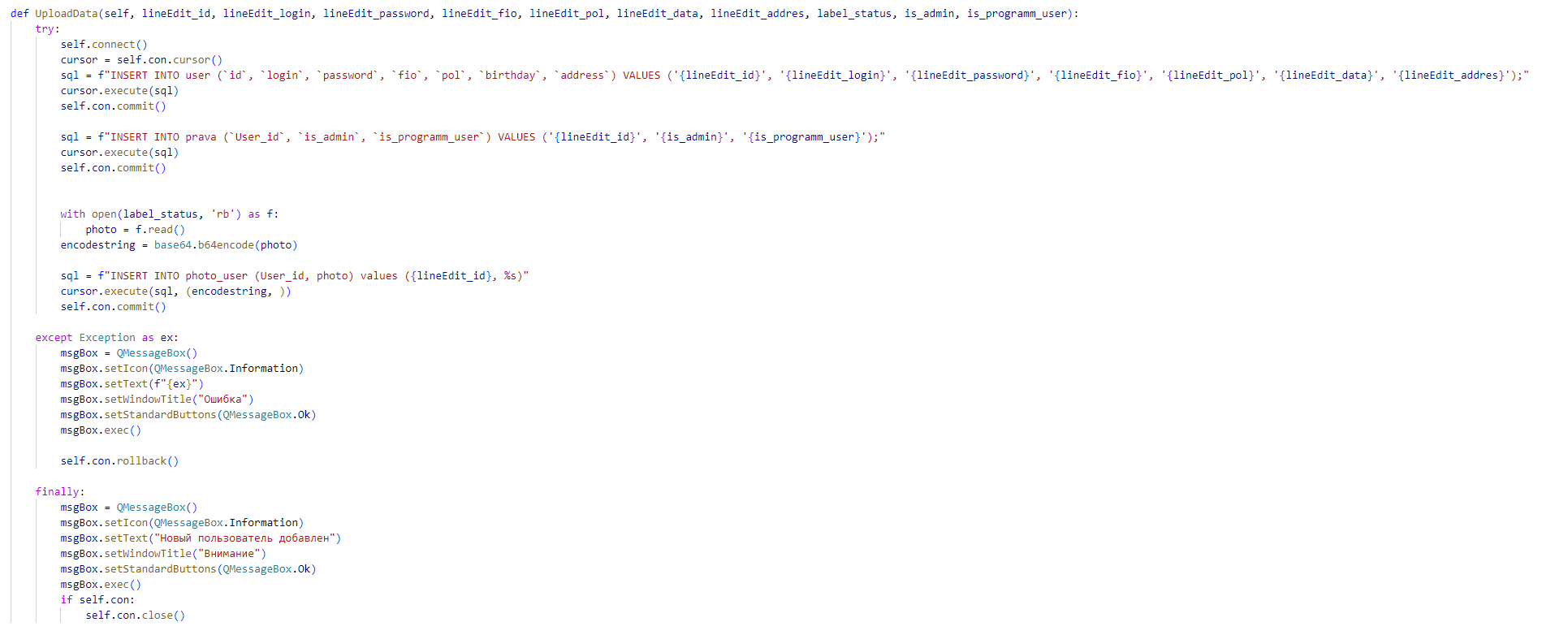


Рисунок 21 – Код добавление данных в базу данных

На рисунке 22 представлен код удаления данных о пользователях. В начале выбирается пользователь из таблицы которого хотят удалить из базы данных (рисунке 23) и дальше проходит сверки id пользователя со всеми таблицами где есть вторичный ключ, то данные удаляются всех таблиц (рисунок 24).



Рисунок 22 – Код удаления информации из базы данных

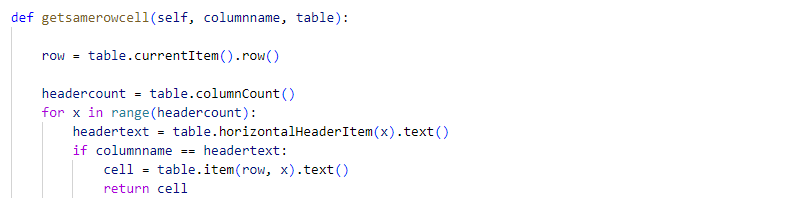


Рисунок 23 – Код поиска id пользователя выбранного в таблице

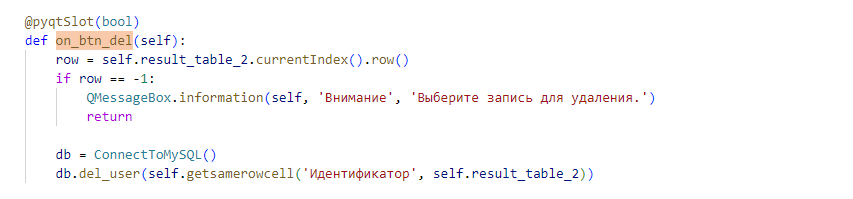


Рисунок 24 – Код отправки id в функции для удаления

На рисунке 25 представлен код изменение данных о пользователях. В начале выбирается строка с пользователем в таблице на которой происходили изменения, затем происходит чтение данных из линии (рисунке 26) и дальше проходит сверка id пользователя со всеми таблицами где есть вторичный ключ, и данные изменяются во всех таблицах.

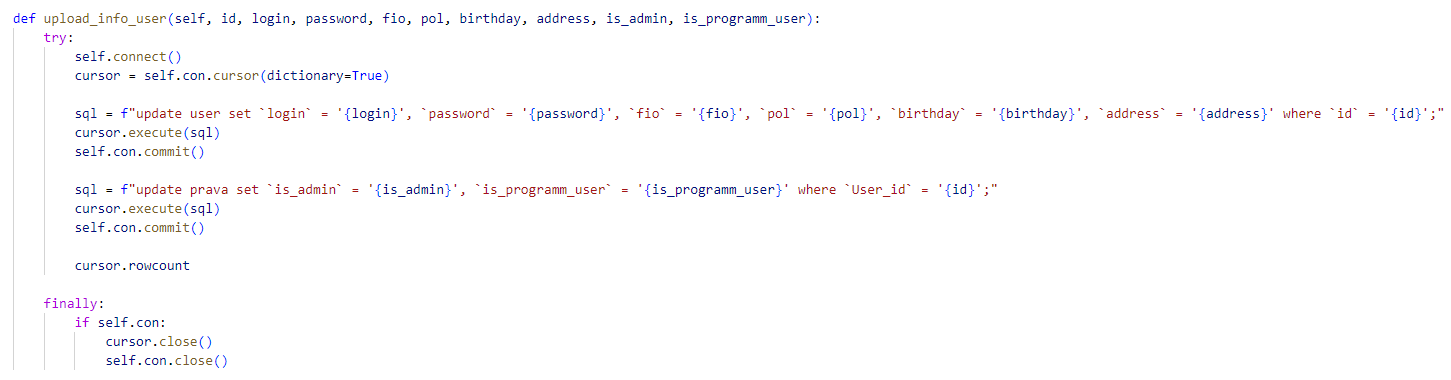


Рисунок 25 – Код отправки данных на изменение

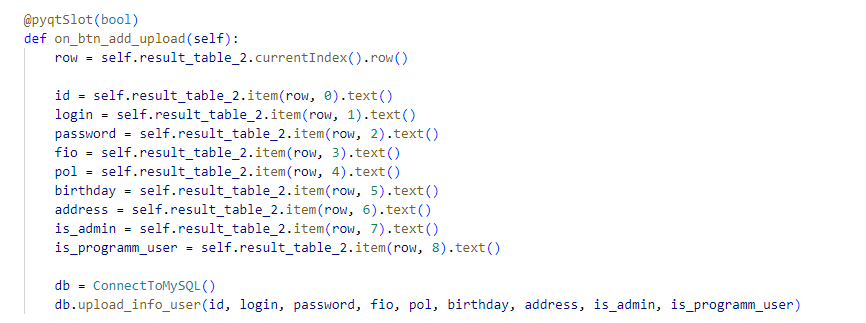


Рисунок 26 – Код чтения измененной линии

На рисунке 27 представлен код идентификации лица. После того как клиент выбрал фото для распознавания, начинается цикл вывода всех фотографий из базы данных и выбранное лицо проходит через алгоритм для нахождения уникальных лицевых точек. По средством сравнения фотографии из базы данных и выбранного лица клиентом через уникальные лицевые точки происходит сверка на то, являются данные лица похожими, если сверка оказалась положительной, то делается запрос на вывод информации из бд по идентификатору пользователя.



Рисунок 26 – Код распознавания и идентификации пользователя

## **6. Документирование программного продукта**

## **6.1. Руководство пользователя ИС**

Информационная система «Идентификация личности» устроена так, чтобы интерфейс был понятен для каждого пользователя, что позволит спокойно взаимодействовать с системой.

Для того что бы начать пользовать программным продуктом, необходимо запустить проект и перейти в меню во вкладку “Настройки”, в открытой вкладки вписать данные для подключения к базе данных и нажать на кнопку подключиться (рисунок 27).

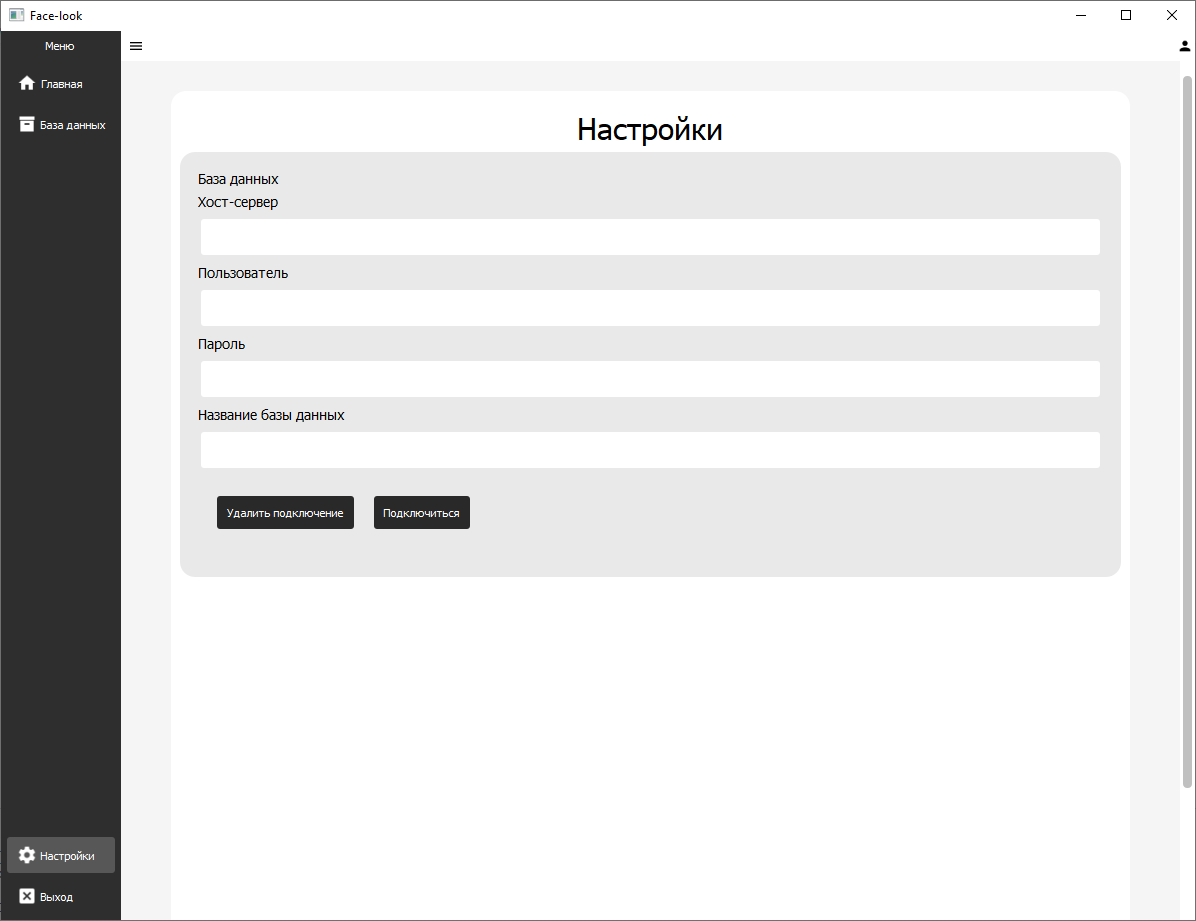


Рисунок 27 – Главное окно в меню “Настройки”

Далее нужно перезагрузить проект посредству его закрытия и открытия.

При открытии проекта, пользователю будет доступна главная страница, где необходимо авторизоваться, прежде чем начать эксплуатацию, для этого пользователь может открыть окно авторизации нажав на иконку в верхнем правом углу. В открытом окне пользователю необходимо вписать данные для входа в поля предназначенные для этого (рисунок 28).

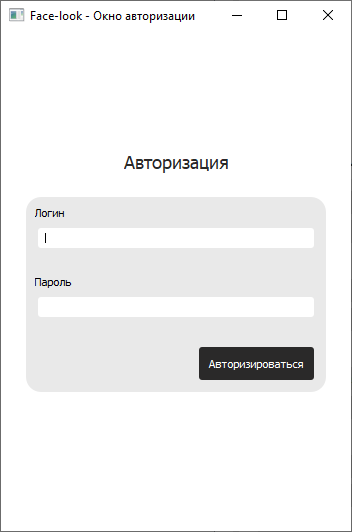


Рисунок 28 – Окно авторизации

Пользователи, у которых уже есть, аккаунт с правами клиента имеют возможность прогрузить таблицу “user” из базы данных в меню “База данных” и увидеть информацию о пользователях и их фотографию (рисунок 29).

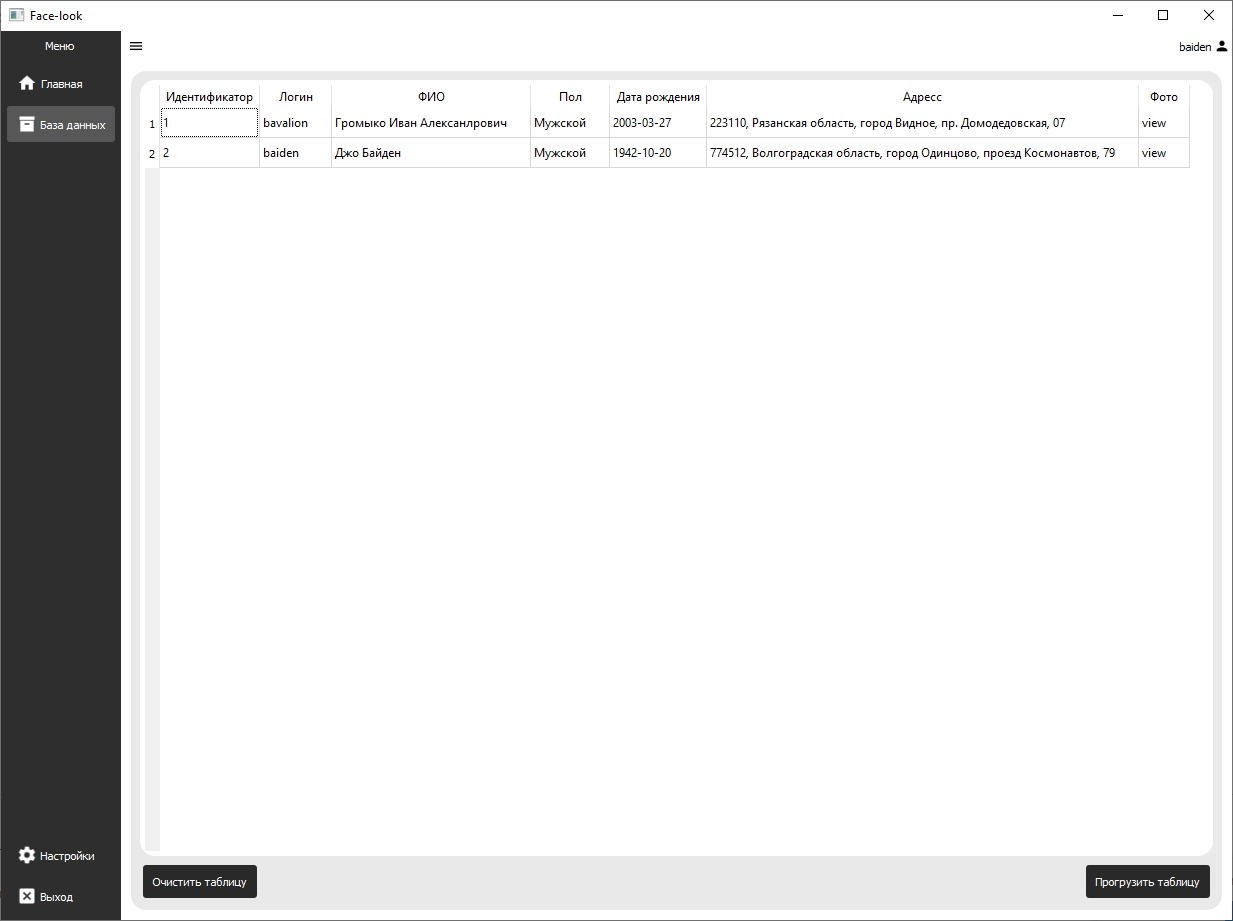


Рисунок 29 – Главное окно в меню “База данных”

В меню “Главная” пользователь может добавить изображение нажав на кнопку “Добавить фото” и проверить человека на наличие присутствия в базе данных и получить тем самым ответ в виде обновленного статуса и выводом данных о найденном человеке (рисунок 30).

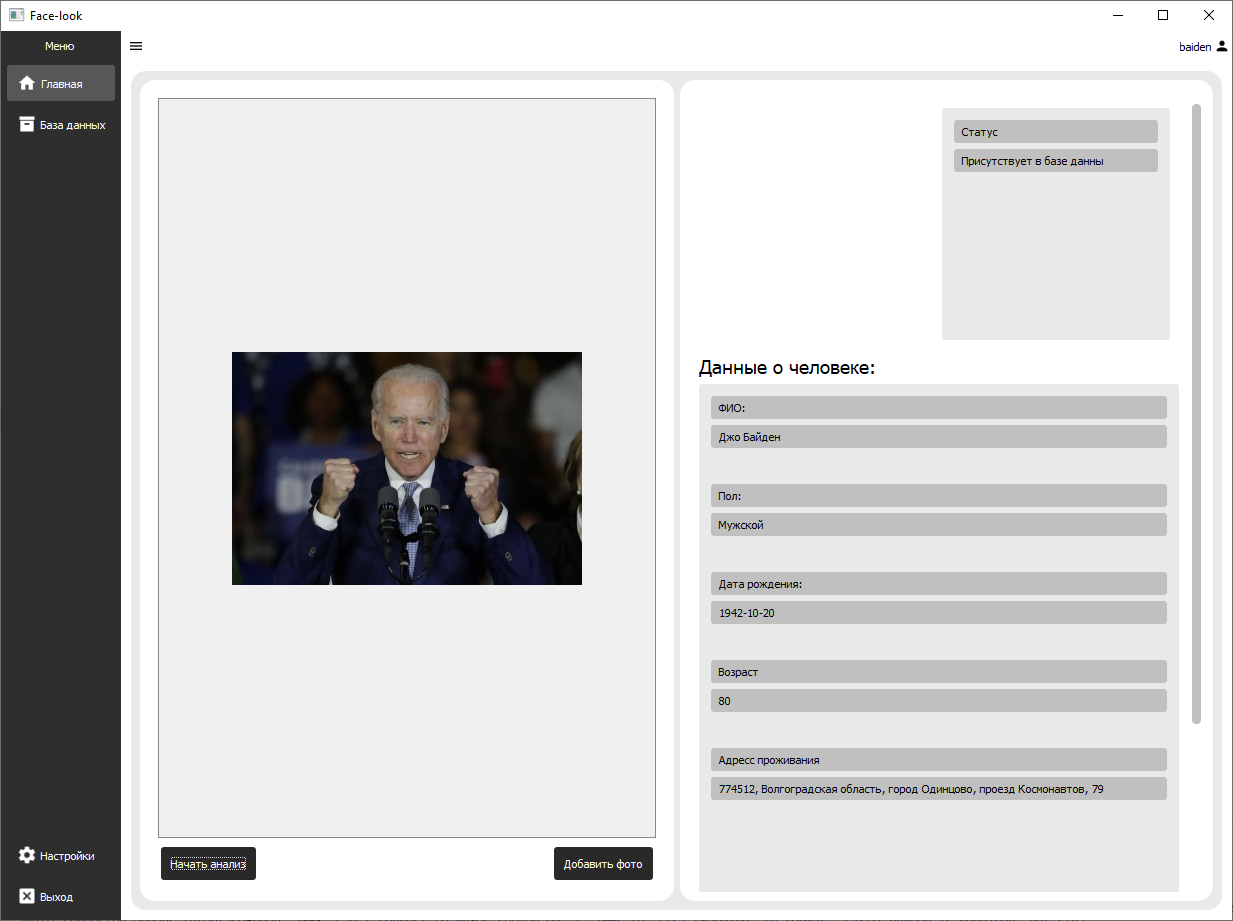


Рисунок 30 – Главное окно в меню “Главная” с выводом информации

Пользователи, у которых есть права администратора, могут перейти во вкладку “Добавить” и начать создавать, изменять и удалять данные в базе данных и создавать выгрузку всей информации в файл \*.xlsx (рисунок 31, 32).

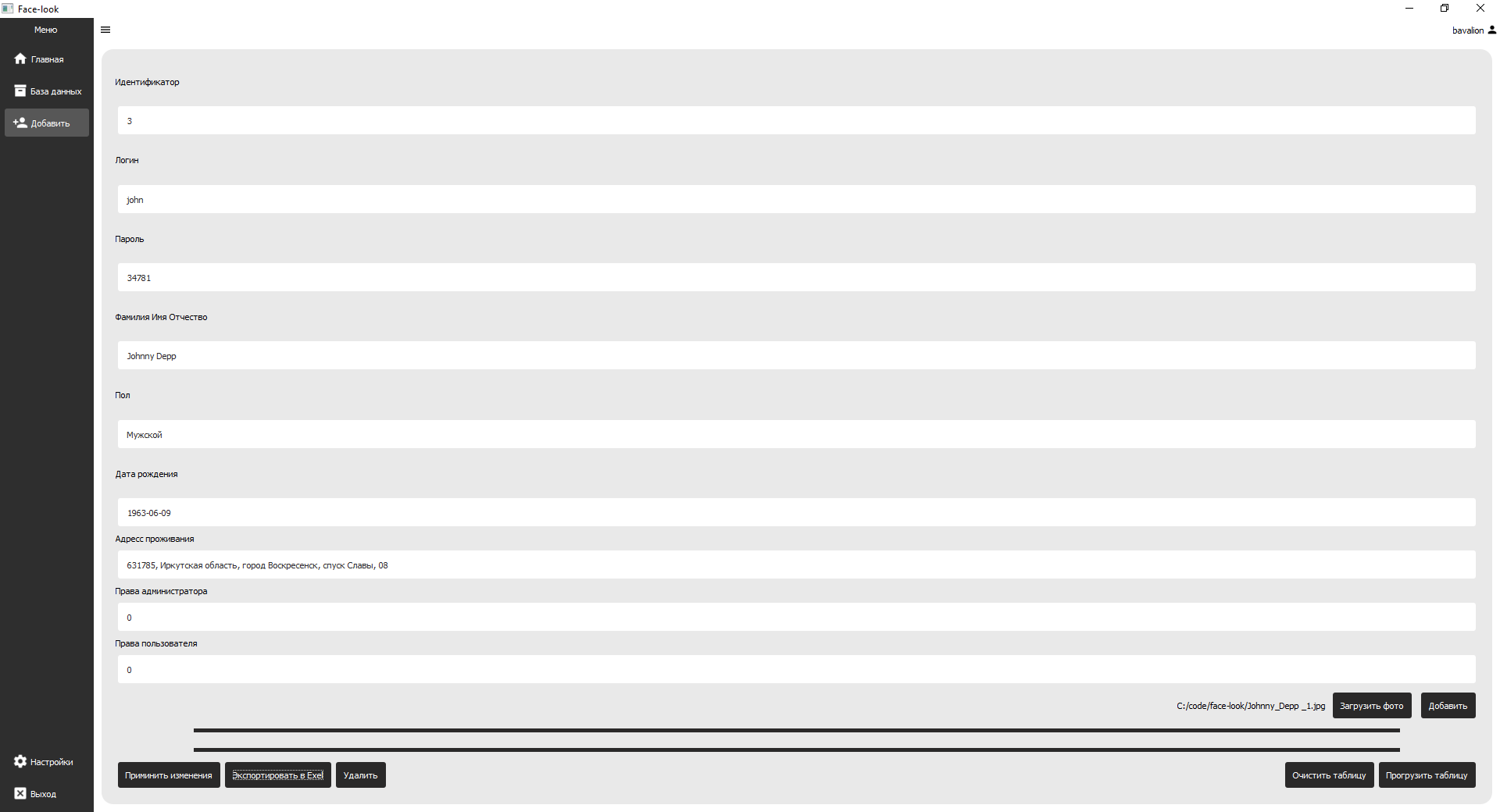


Рисунок 31 – Главное окно в меню “Добавить” с функцией добавления

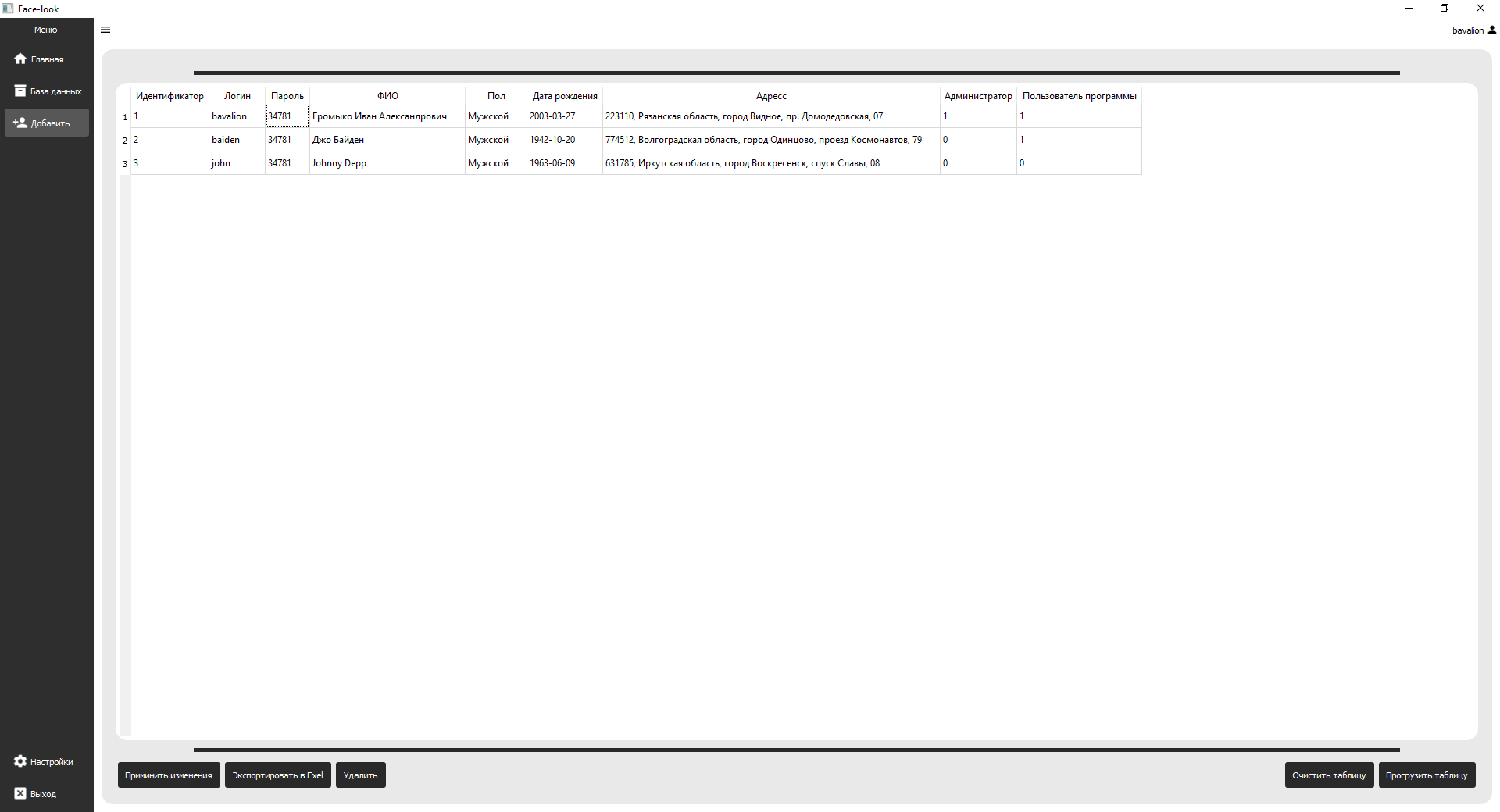


Рисунок 31 – Главное окно в меню “Добавить” с функциями удаления, применения и экспорта

**Заключение**

В ходе создания курсового проекта были решены следующие задачи:

1. провести анализ предметной области и представить его описание;
2. выполнить анализ программных продуктов, используемых при разработке ИС;
3. разработать техническое задание;
4. выполнить проектирование информационной системы;
5. разработать структуру и интерфейс;
6. разработать информационную систему;
7. провести тестирование и отладку;
8. написать руководство пользователя ИС.

Был определен и реализован следующий функционал ИС:

− авторизация и регистрация;

− просмотр данных;

− манипулирование данных.

Были рассмотрены возможные реализации ИС с использованием разных технологий и языков программирования, но в результате анализа инструментальных средств разработки выбор сделан в пользу языка программирования python и среды разработки Visual Studio Code.

В дальнейшем в ИС планируется расширение функционала и совершенствования интерфейса, тем самым набирая большее сообщество пользователей, а также и актуальность программного продукта.

[СПИСОК ИСПОЛЬЗУЕМЫХ ИСТОЧНИКОВ](file:///G:\КП__Ильин%20Артур.docx#_Toc89852065)

1. github.com : Face Recognition. – URL: https://github.com/ageitgey/face\_recognition (дата обращения: 05.11.2022). – Текст: электронный.
2. Python 3 для начинающих : Строки. Функции и методы строк. – URL: https://pythonworld.ru/tipy-dannyx-v-python/stroki-funkcii-i-metody-strok.html (дата обращения: 24. 09.2022). – Текст: электронный.
3. Qt DOCUMENTATION : Публичные функции. – URL: https://doc.qt.io/qt-6/qpixmap.html (дата обращения: 12.10.2022). – Текст: электронный.
4. Riverbank Computing : pyrcc5. – URL: https://www.riverbankcomputing.com/static/Docs/PyQt5/resources.html#pyrcc5 (дата обращения: 12.10.2022). – Текст: электронный.
5. Riverbank Computing : pyuic5. – URL: https://www.riverbankcomputing.com/static/Docs/PyQt5/designer.html#pyuic5 (дата обращения: 12.10.2022). – Текст: электронный.
6. Riverbank Computing : Using the Generated Code. – URL: https://www.riverbankcomputing.com/static/Docs/PyQt5/designer.html#pyuic5 (дата обращения: 08.10.2022). – Текст: электронный.
7. Riverbank Computing : Что такое PyQt?. – URL: https://www.riverbankcomputing.com/software/pyqt/ (дата обращения: 08.10.2022). – Текст: электронный.
8. YouTube : Face Recognition with MYSQL Database in Python | Jupyter | Open CV| Xampp Server. – URL: https://www.youtube.com/watch?v=tBP3orL7bIo&list=WL&index=5&ab\_channel=DataScienceDevil%27s (дата обращения: 05.11.2022). – Видео: электронный.
9. YouTube : Python для начинающих. Урок 17.: Наследование (ООП), функция super(), Полиморфизм. – URL: https://www.youtube.com/watch?v=-UOfovZosj8&list=WL&index=8&t=338s&ab\_channel=alishev (дата обращения: 05.10.2022). – Видео: электронный.
10. ВикипедиЯ : PyCharm. – URL: https://ru.wikipedia.org/wiki/PyCharm (дата обращения: 24.09.2022). – Текст: электронный.
11. ВикипедиЯ : Python. – URL: https://ru.wikipedia.org/wiki/Python (дата обращения: 24. 09.2022). – Текст: электронный.
12. ВикипедиЯ : Spyder (IDE). – URL: https://ru.wikipedia.org/wiki/Spyder\_(IDE) (дата обращения: 24. 09.2022). – Текст: электронный.
13. ВикипедиЯ : Visual Studio Code. – URL: https://ru.wikipedia.org/wiki/Visual\_Studio\_Code (дата обращения: 24. 09.2022). – Текст: электронный.

# **Приложение А – Техническое задание**

**Министерство образования Иркутской области**

Государственное бюджетное профессиональное образовательное учреждение

Иркутской области

«Иркутский авиационный техникум»

(ГБПОУИО «ИАТ»)

|  |  |
| --- | --- |
|  |  |

ТЕХНИЧЕСКОЕ ЗАДАНИЕ

ИНФОРМАЦИОННАЯ СИСТЕМА

«ИДЕНТИФИКАЦИЯ ЛИЧНОСТИ»

|  |  |  |
| --- | --- | --- |
| Руководитель: | \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  (подпись, дата) | (М.А. Кудрявцева) |
| Студент: | \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  (подпись, дата) | (И.А. Громыко) |

Иркутск 2022

**1 Общие сведения**

Наименование работы: информационная система «Идентификация личности».

Исполнитель: студент иркутского авиационного техникума, группы ПКС-19-3, Громыко И.А.

Разработка информационной системы проходит в рамках курсового проекта по МДК.03.01 «Технология разработки программного обеспечения», на основании приказа №37-у от 21 сентября 2022 года.

Сроки разработки информационной системы с 21.09.2022 по 06.12.2022 года.

**2 Назначение и цели создания системы**

Назначение информационной системы «Идентификация личности» заключается в распознавании лиц и хранении информации о людях. Для пользователя необходимы такие функции, как добавление информации, изменение информации о людях в базу данных, просмотр информации о людях, находящихся на изображениях.

# **3 Требования к системе в целом**

## **3.1 Требования к структуре и функционированию системы**

Функции информационной системы:

1. раздел «Личности»;
   1. отображение информации о личностях (ФИО, Пол, Возраст);
   2. регистрация новых личностей;
   3. удаление личностей;
2. раздел «Оборудование»;
   1. отображение информация об оборудовании;
   2. добавление нового оборудования (камеры);

**3.2 Требования к надежности**

Для обеспечения надежности необходимо проверять корректность получаемых данных и реализовать валидность полей. Входные данные поступают в виде значений c клавиатуры. Эти значения отображаются в отдельных полях таблицы.

**3.3 Требования к безопасности**

Для обеспечения безопасности в информационной системе, необходимо реализовать разграничение прав доступа по средствам пароля.

**3.4 Требования к эксплуатации, техническому обслуживанию, ремонту и**

**хранению компонентов системы**

Минимальные системные требования для рабочей станции:

1. Процессор: Intel Core i5 / AMD XP 2200+;
2. Оперативная память: 8 Гб;
3. Жёсткий диск: 2 Гб;
4. Операционная система: Windows.

Минимальные системные требования для серверной станции:

1. Процессор: Intel Core 2 Duo 4 2.0Ghz / AMD XP 2200+;
2. Оперативная память: 4 Гб;
3. Жёсткий диск: 2 Гб;
4. Операционная система: Windows, Linux.

**4 Требования к документированию**

Основным документам, регламентирующими использование информационной системы является руководство пользователя.

Основным документам, регламентирующими разработку информационной системы является техническое задание.

**5 Состав и содержание работ по созданию системы**

В таблице 1 представлены плановые сроки начала и окончания работы по созданию информационной системы.

Таблица 1 – Плановые сроки по созданию информационной системы

|  |  |  |  |
| --- | --- | --- | --- |
| №  п/п | Содержание этапов разработки | Сроки выполнения | |
| Начало | Окончание |
| 1 | Предпроектное исследование предметной области (выбор темы, постановка цели, задач, описание области применения, исследование предметной области) | 12.09.22 | 21.09.2022 |
| 2 | Разработка технического задания (выбор архитектуры программного обеспечения, выбор типа пользовательского интерфейса, выбор языка и среды программирования) | 22.09.22 | 26.09.2022 |
| 3 | Проектирование программного обеспечения.  (разработка структурной и функциональной схемы ПО, проектирование базы данных (инфологическое, ER-модель, физическая модель) | 27.09.22 | 25.10.2022 |
| 4 | Разработка (программирование) и отладка программного продукта | 26.10.22 | 28.11.2022 |
| 5 | Составление программной документации (оформление руководство пользователя) | 29.11.22 | 06.12.2022 |

**Приложение Б – Листинг авторизация**

class Authorization(QtWidgets.QWidget):

def \_\_init\_\_(self):

super(Authorization, self).\_\_init\_\_()

self.appUI = Ui\_Authorization()

self.appUI.setupUi(self)

self.auth\_user = self.appUI.pushButton

self.status = self.appUI.label\_4

self.auth\_user.clicked.connect(self.on\_auth\_user)

@pyqtSlot(bool)

def on\_auth\_user(self):

self.user\_login = self.appUI.lineEdit\_login\_auth.text()

self.user\_password = self.appUI.lineEdit\_pass\_auth.text()

self.connect = ConnectToMySQL()

result = self.connect.connect\_auth\_user()

rules = self.connect.rules()

for item in result:

for i in rules:

self.username = item['login']

self.window = MainWindow()

if (not self.user\_login) or (not self.user\_password):

self.status.setText('Вы не ввели логин или пароль')

elif (str(self.user\_login) != str(item['login']) or str(self.user\_password) != str(item['password']) ):

self.status.setText('Введен неправильно логин или пароль')

else:

if (str(self.user\_login) == str(item['login']) and str(self.user\_password) == str(item['password']) and (item['id'] == i['User\_id'])):

if (item['id'] == i['User\_id'] and i['is\_programm\_user'] == 1):

self.status.clear()

window.btn\_add\_1.hide()

window.btn\_add\_2.hide()

window.label\_user.setText(self.username)

self.close()

if (item['id'] == i['User\_id'] and i['is\_admin'] == 1):

self.status.clear()

window.btn\_add\_1.show()

window.btn\_add\_2.show()

window.label\_user.setText(self.username)

self.close()

if (item['id'] == i['User\_id'] and i['is\_admin'] == 0 and i['is\_programm\_user'] == 0):

self.status.setText('У пользователя нет прав для входа')

break

**Приложение В – Листинг добавления данных в таблицы.**

@pyqtSlot(bool)

def on\_btn\_add\_user\_in\_bd(self):

lineEdit\_login = self.appUI.lineEdit\_login.text()

lineEdit\_id = self.appUI.lineEdit\_id.text()

lineEdit\_password = self.appUI.lineEdit\_password.text()

lineEdit\_fio = self.appUI.lineEdit\_fio.text()

lineEdit\_pol = self.appUI.lineEdit\_pol.text()

lineEdit\_data = self.appUI.lineEdit\_data.text()

lineEdit\_addres = self.appUI.lineEdit\_addres.text()

lineEdit\_admin = self.appUI.lineEdit\_admin.text()

lineEdit\_polsovatel = self.appUI.lineEdit\_polsovatel.text()

label\_status = self.appUI.label\_status.text()

if (label\_status == ''):

msgBox = QMessageBox()

msgBox.setIcon(QMessageBox.Information)

msgBox.setText("Вы не добавили фото!\nДобавьте обязательно фотографию")

msgBox.setWindowTitle("Внимание")

msgBox.setStandardButtons(QMessageBox.Ok)

msgBox.exec()

elif (lineEdit\_id == "" or lineEdit\_password == "" or lineEdit\_fio == "" or lineEdit\_pol  == "" or lineEdit\_data == "" or lineEdit\_addres == "" or lineEdit\_admin == "" or lineEdit\_polsovatel == ""):

msgBox = QMessageBox()

msgBox.setIcon(QMessageBox.Information)

msgBox.setText("Вы не заполнили все строки данными")

msgBox.setWindowTitle("Внимание")

msgBox.setStandardButtons(QMessageBox.Ok)

msgBox.exec()

else:

db = ConnectToMySQL()

db.UploadData(lineEdit\_id, lineEdit\_login, lineEdit\_password, lineEdit\_fio, lineEdit\_pol, lineEdit\_data, lineEdit\_addres, label\_status, str(lineEdit\_admin), str(lineEdit\_polsovatel))

def UploadData(self, lineEdit\_id, lineEdit\_login, lineEdit\_password, lineEdit\_fio, lineEdit\_pol, lineEdit\_data, lineEdit\_addres, label\_status, is\_admin, is\_programm\_user):

try:

self.connect()

cursor = self.con.cursor()

sql = f"INSERT INTO user (`id`, `login`, `password`, `fio`, `pol`, `birthday`, `address`) VALUES ('{lineEdit\_id}', '{lineEdit\_login}', '{lineEdit\_password}', '{lineEdit\_fio}', '{lineEdit\_pol}', '{lineEdit\_data}', '{lineEdit\_addres}');"

cursor.execute(sql)

self.con.commit()

sql = f"INSERT INTO prava (`User\_id`, `is\_admin`, `is\_programm\_user`) VALUES ('{lineEdit\_id}', '{is\_admin}', '{is\_programm\_user}');"

cursor.execute(sql)

self.con.commit()

with open(label\_status, 'rb') as f:

photo = f.read()

encodestring = base64.b64encode(photo)

sql = f"INSERT INTO photo\_user (User\_id, photo) values ({lineEdit\_id}, %s)"

cursor.execute(sql, (encodestring, ))

self.con.commit()

except Exception as ex:

msgBox = QMessageBox()

msgBox.setIcon(QMessageBox.Information)

msgBox.setText(f"{ex}")

msgBox.setWindowTitle("Ошибка")

msgBox.setStandardButtons(QMessageBox.Ok)

msgBox.exec()

self.con.rollback()

finally:

msgBox = QMessageBox()

msgBox.setIcon(QMessageBox.Information)

msgBox.setText("Новый пользователь добавлен")

msgBox.setWindowTitle("Внимание")

msgBox.setStandardButtons(QMessageBox.Ok)

msgBox.exec()

if self.con:

self.con.close()