## Проектно-конструкторская часть

### Разработка структуры приложения

У разрабатываемой системы есть 2 источника информации:

1. Видеопоток.
2. База данных.

База данных заполняется как пользователем, так и в автономном режиме, путем сбора и обработки данных с видеокамер.

Пользователю доступно ручное заполнение базы данных через графический интерфейс следующими записями:

1. Биометрические данные (добавление\изменение); выбирается сотрудник и загружается его фотография, которая обрабатывается внутренними механизмами обработки информации, результаты обработки записываются в базу данных.
2. Контрольно-пропускные пункты (добавление\изменение\удаление).
3. Камеры (добавление\удаление).

Так же база данных заполняется системой: видео, захваченное с камеры, обрабатывается механизмами обработки информации, результаты обработки записываются в базу данных.

Система определена двумя компонентами – интерфейс и управляющие подсистемы.

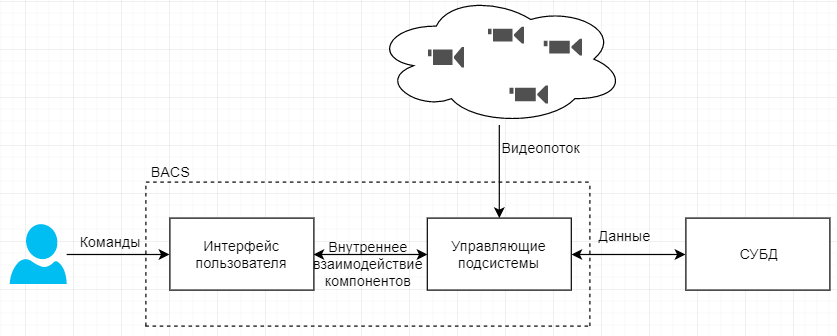


Рисунок 1- Структурная схема системы

Задачи управляющих подсистем определены в техническом задании.

Последовательность разработки компонентов системы:

1. Проектирование структуры базы данных.
2. Выбор архитектуры приложения.
3. Создание пользовательского интерфейса.
4. Создание управляющих подсистем.
5. Ручное тестирование каждой подсистемы.
6. Локализация интерфейса.
7. Тестирование взаимодействия подсистем.
8. Нагрузочное тестирование системы в целом.

### Логическая схема базы данных

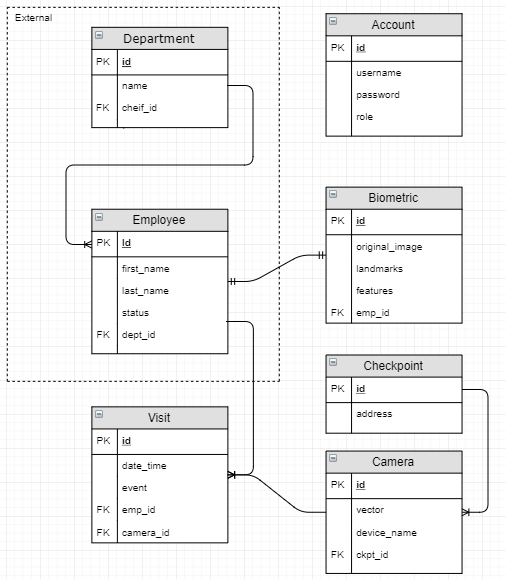


Рисунок 2 – Логическая схема базы данных

Логическая схема базы данных состоит из следующих таблиц:

1. Employee и Department – являются внешними таблицами отдела кадров.
2. Biometric – биометрия. Эта таблица содержит в себе эталонное изображение (original\_image) по которому, внутри системы, определяются характерные точки лица и биометрические признаки, уникальные для каждого сотрудника.
3. Checkpoint – кпп, содержит в себе адрес (address) пропускного пункта.
4. Camera – камера, описывается именем устройства (device\_name) и направлением установки (vector)-вход\выход.
5. Visit – посещение, характеризуется датой и временем (data\_time) и событием прохода конкретного работника (event).

### Разработка архитектуры приложения

Для разрабатываемой системы было принято решение использовать многоуровневую архитектуру приложения (N-layers).

Выбранная архитектура состоит из следующих уровней:

* + - 1. DAL (data access layer) – уровней доступа к данным.
      2. BLL (business-logic layer) – уровень бизнес-логики.
      3. APP (application) – уровень приложения.

Уровень доступа к данным определяет объекты технологии ORM, описывающие каждую сущность базы данных, а также классы и интерфейсы, содержащие набор CRUD (create\update\delete) функционала, для описанных сущностей.

Уровень бизнес-логики содержит в себе функциональные бизнес модели, такие как функционал настройки всегда приложения. Этот уровень реализует функции обработки данных – сортировку, группировку, выборку, объединенные в отдельные сервисы верхнего доступа к данным.

Уровень приложения содержит в себе файлы интерфейса приложения, локализации, а также первичный функционал обработки данных, поступающих от пользователя. На этом уровне определены подсистемы логгирования, перехвата видеопотока, принятия решений.

Передача данных между всеми подсистемами осуществлена при помощи шаблона проектирования DTO – data transfer object. Концепцией этого шаблона является создание объектов данных, не содержащих никакого поведения. Эти объекты реализованы в классах, лежащих на уровне BLL.

Для взаимодействия с СУБД PostgreSQL v10 была выбрана программная библиотека SQLAlchemy, служащая для синхронизации объектов, определенных на DAL уровне, и записей базы данных.

### Разработка интерфейса взаимодействия пользователя с системой.

На основе требований к ПО и разрабатываемой системе, описанных в техническом задании, было принято решение использовать кроссплатформенную среду для разработки графических интерфейсов программ, использующих библиотеку Qt – Qt Designer.

Локализация интерфейса осуществляется при помощи инструментов Poedit, Qt Linguist сразу на 3 языка – русский, английский, китайский.

Само графическое представление системы состоит из следующих компонент и поведений:

* + 1. При первоначальном запуске программы предлагается прописать строку подключения формата «postgresql://user:password@hostname/ databasename», которая далее будет записана в локальный файл конфигурации приложения.
    2. Окно авторизации.
    3. Главное окно.
    4. Окно настроек.

## Разработка алгоритмов обработки информации.

Основной цикл обработки информации – обработка видеопотока. Основные этапы обработки данных видеопотока:

* 1. При запуске главного окна производится выгрузка всех видеоустройств.
  2. Инициализируется системный поток, в котором циклически определяется список всех видеоустройств, подключенных к серверу.
  3. Для каждого видеоустройства (из п.1) инициализируется отдельный системный поток.

В каждом таком потоке инициализируется цикл с этапами:

1. Проверка наличия устройства: если его нет в списке подключенных устройств, то пишется лог о недоступности данного устройства, если при повторной проверке устройство найдено – пишется лог о подключении нового устройства.
2. Проверка готовности видеоустройства к работе.
3. Фиксация кадра.
4. Полученный кадр направляется в подсистему обработки информации
5. Подсистема обработки информации обрабатывает кадр, выводит его в интерфейс главного окна, а полученные выделенные биометрические данные передаются подсистеме принятия решений.
6. Подсистема принятия решений сверяет эталонные биометрические данные с выделенными. При нахождении достаточно схожих (функционал расстояния Кульбака-Лейблера) происходит логгирования события обнаружение работника, проверяется его статус, подается команда открытия турникетов, фиксируется событие прохода работника.

Нетривиальные алгоритмы обработки информации заключены в подсистеме обработки видеопотока.

Эта подсистема разбита на следующие модули:

1. Детектирование лиц на изображении.
2. Выделение характерных точек лица.
3. Нормализация поворота лица.
4. Выделение биометрических признаков.

Модуль детектирование лиц на изображении содержит в себе ранее обученную нейронную сеть Mobilenet SSD. Нейронная сеть реализована на Tensorflow object detection api, обучена на датасете «WiderFace». Эта нейронная сеть детектирует все лица в кадре. При этом на кадр наносится прямоугольники, ограничивающие лица людей (bounding boxes). Далее этот кадр подается в интерфейс главного окна. Среди полученных лиц ищется лицо, которое занимает большую площадь изображения, чем другие. При этом происходит проверка – если эта площадь больше определенного значения (это значение определяется техническим инженером и записывается в настройки системы), то данное лицо получает статус кандидата на получение пропуска.

Далее происходит выделение характерных точек лица-кандидата. Выделение производится за счет обученной модели ансамбля деревьев регрессии. Данная модель реализована в кроссплатформенной библиотеке Dlib. Выделяются 68 характерных точек.

После этого происходит поворот лица. Алгоритм базируется на геометрической структуре лица (68 точек характерных точек) и получении канонического выравнивания лица, основанное на перемещении, масштабе и вращении. Так же алгоритм выравнивает грани изображения так, чтобы они были примерно одинакового размера. Алгоритм реализован при помощи средств библиотек opencv, numpy.

Повернутое лицо передается на следующий модуль – выделение биометрических признаков. За работоспособность этого модуля отвечает нейронная сеть Facenet, обученная на «VGGFace2» датасете. Принцип ее работы основан на выделении «мест» для каждого лица в евклидовом пространстве, которое она научилась строить при обучении. На выходе нейронной сети образуется вектор размерностью R128, который как раз-таки и является вектор биометрических признаков, который записывается в базу данных к каждому работнику и сравнивается с такими же векторами в подсистеме принятия решений.

Все компоненты подсистемы обработки информации видеопотока подобраны исходя из условия real-time работы системы и условия надежности.

Так же был проведен тест компонент подсистемы обработки видеопотока, который показал 20FPS при 5 лицах в кадре.