Министерство науки и высшего образования Российской Федерации
Федеральное государственное бюджетное образовательное
учреждение высшего образования
«Петрозаводский государственный университет»
Институт математики и информационных технологий
Кафедра информатики и математического обеспечения

Баженов Никита Александрович

Отчет по учебному курсу «Верификация Программного Обеспечения»

Тестирование сервисов видеонаблюдения в рамках выполнения проекта НИР «Разработка программно-аппаратного комплекса производственного мониторинга на предприятии»

Направление 09.04.02 — Информационные системы и технологии

Программа «Управление данными»

Преподаватель:
к.ф.–м.н., доцент К. А. Кулаков

Петрозаводск

2019

Содержание

1 Объект тестирования

Видеоаналитика создает широкий спектр приложений с огромным потенциалом для взаимодействия с обществом и при реализации различных видеосервисов . Географически распределенная архитектура общедоступных облаков и границ, которые простираются до камер, является единственным возможным подходом для удовлетворения строгих требований в реальном времени крупномасштабной аналитики живого видео.

Многие текущие системы видеоаналитики описывают и объясняют объектно- ориентированную концепцию на основе происходящих событий в рамках унифицированных языков моделирования и проектирования. Разработка сервисно-ориентированной системы, основанной на событиях, которые происходят во время извлечения видеоданных, может быть полезна для разработчиков различных видеоприложений. Зачастую создаваемый макет и взаимодействие между компонентами видеосистемы не может быть ясным и понятным, когда клиентом является разработчик видео приложения. Кроме того, события в видеоданных могут быть сложными и универсальными, что позволит наиболее детально изучить полученные данные для оказания услуг и построения сервисов. Тестирование такого вида систем позволяет определить проблемные участки кода

Задачами тестирования видеосистемы является:

- Наиболее полное представление сведений о системе конечному заказчику
- Повышение качества модулей видеосистемы путем выполнения тестирования различных видов
- Выявление ошибок и неоптимизированных участков в системе с целью их дальнейшего исправления

Основные определения системы:

- Модуль - программная реализация набора функций (функции), предоставляюших ту или иную возможность;
- Подсистема - набор модулей, взаимодействующих между собой;
- Система - самостоятельное приложение, состоящее из подсистем;
- Сервис - цифровая услуга, обеспечивающая доступ к одной или многим возможностям системы для решения возникающих у пользователя задач;

- Слой - условное обозначение набора компонент, взаимодействующих между собой;
- Событие - явление, необходимое для дальнейшего анализа и интерпретации и достаточное для сохранения в файловой системе и/или базе данных;
- Простое событие - событие, являющееся простым т.е нетребующим дополнительных объяснений на уровне рассматриваемой архитектуры;
- Составное событие - событие, включающее в себя цепочку из нескольких последовательных событий, образующих одно сложное.

Объектом тестирования является система видеонаблюдения, состаящая из сервисов и осуществляющая мониторинг за производственным оборудованием. Сервисы представляют набор модулей, работающих отдельно и выполняющих свои функции (отображение в веб, работа с различными наборами датчиков, модули видеокамеры, модули мониторинга событий, модули генерации сложных событий). Таким образом, будет происходить тестирование как всей системы в целом, так и каждого отдельного модуля в частности. Сервисы для пользователей в виде графиков, изображений и контекстной информации строятся в вебе.

Система состоит из следующих сервисов (моделей взаимодействия с пользователем, т.е реализации механизмов, обеспечивающих доступ ко многим возможностям ПАК для решения различных пользовательских задач):

1. Модуль 1: Подключение к камере (camera_connection);
2. Модуль 2: Распознавание присутствия человека (human_detection);
3. Модуль 3 (служеб.): Отправка событий в базу данных, сканирование, самодиагностика видеокамер (состоит из 3 подмодулей).

Используемые технологии в системе (жирным выделены технологии, в рамках которых необходимо проводить тестирование):

- **Python (3+)**;
- QML, C++;
- **Bash**
- **MongoDB**;

- ClickHouse;
- Apache, PHP, JS;
- RabbitMQ (Broker, Pub-Sub).

Используемые библиотеки языка Python:

- cv2 (OpenCV: алгоритмы распознавнания, отрисовка изображения);
- multiprocessing, multithreading (распределение ресурсов компьютера на несколько видеопотоков);
- time, datetime (синхронизация по времени);
- pymongo, MongoClient, bson, json (работа с коллекциями событий и объектами в БД MongoDB);
- pytesseract (Сервис 1);
- caffe (Сервис 2).

Основные функции объекта тестирования:

- Просмотр изображения с одной камеры;
- Выполнение функций в многопоточном режиме (просмотр изображения с нескольких камер, корректная и своевременная работа алгоритмов);
- Корректность выполнения конфигураций (sh, ini) из коллекции начальных параметров;
- Корректноость распознавания человека в кадре (вероятность определения, близкая к 1);
- Корректноость генерации и отправления данных в MongoDB (вероятность определения, близкая к 1);
- Корректноость сканирования ip-адресов через имеющиеся mac-адреса;
- Корректноость самодиагностики и возможности подключения к камерам;
- Корректноость распознавания человека в кадре (вероятность определения, близкая к 1);

2 Стратегия тестирования

Структура объекта тестирования

Описание модулей системы

Сервисы реализуются за счет программных модулей. Программная архитектура тестируемых модулей системы представлена на Рис. 1

- Модуль подключения к видеокамере camera_connection;
- Модуль генерации событий eventsgenerator;
- Модуль сканирования видеокамер scan;
- Модуль самодиагностики видеокамер selfdiagnosis;
- ~~Модуль кода ошибок errorrecognition~~;
- ~~Модуль поворота головки head_angle~~;
- Модуль определения человека human_detection;
- ~~Модуль нейронной сети nn_training~~;
- ~~Модуль определения расстояния до человека human_distance_estimator~~;
- ~~Модуль удара шпинделя об деталь spindle_bump~~.

Тестирование будет проводиться только для приведенных модулей по следующим причинам:

- Некоторые из модулей не являются достаточно проработанными. То есть, они нуждаются в полноценной разработке, а только потом в тестировании;
- Определенное количество функций данных модулей являются "простыми то есть в них маловероятна возможность допущения какой-либо ошибки;
- Данные модули не являются критичными для системы (то есть, не являются основополагающими), в отличие от модулей, которые будут протестированы в рамках данной системы;

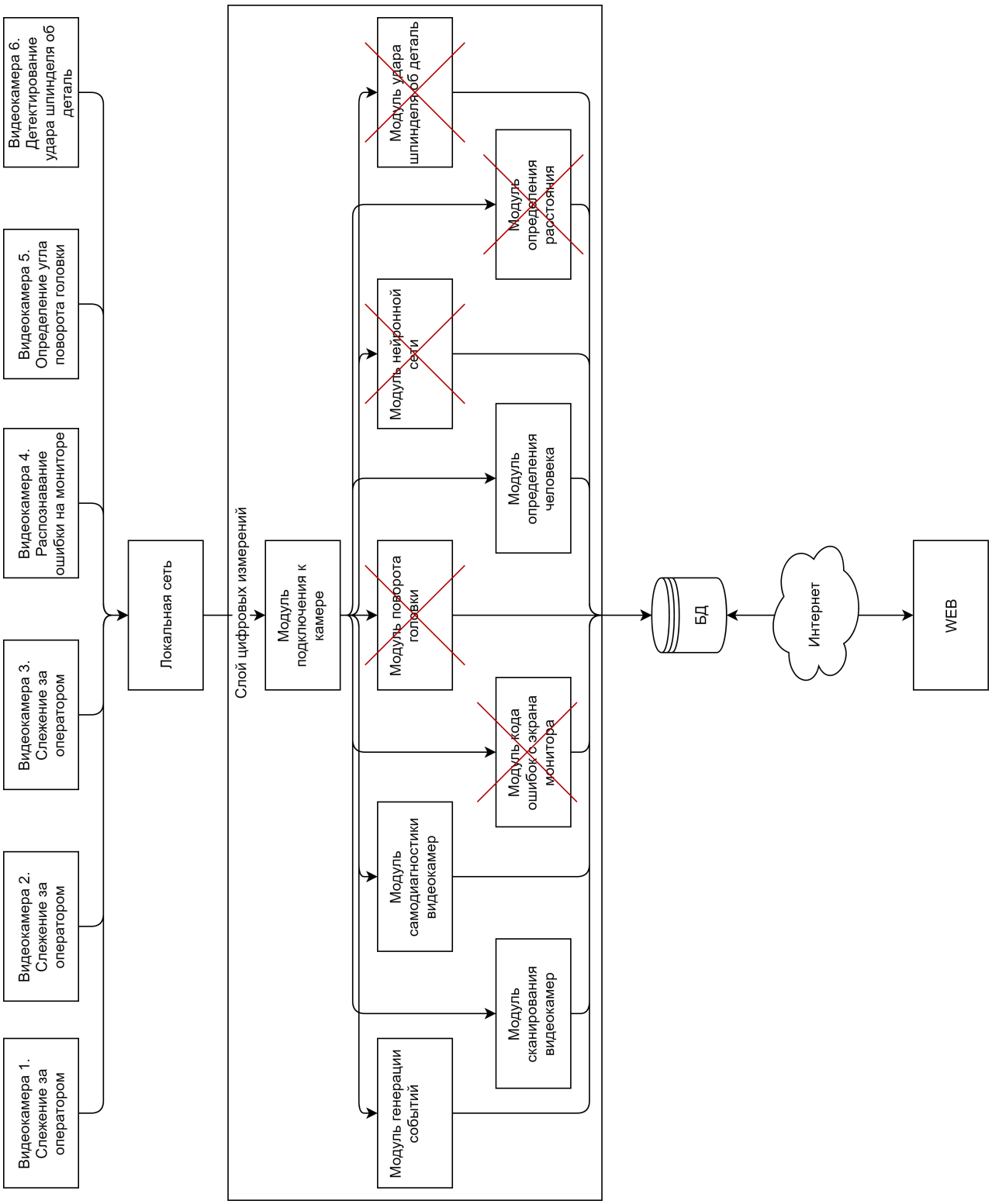

Рис. 1: Программная архитектура тестируемых модулей системы

- Система объемная, она не может быть в полной мере протестирована в рамках прохождения данного направления.

Таким образом, в рамках данной работы будут протестированы следующие модули (с описанием):

1. Модуль подключения к видеокамере camera_connection - модуль осуществляет под-

ключение к камере(-ам) с различными параметрами и настройками из конфигурации;

2. Модуль генерации событий eventsgenerator - модуль, организующий мониторинг событий (на уровне скрипта - генерирующий события) и отправляющий их в базу данных MongoDB;

3. Модуль сканирования видеокамер scan - модуль сканирования локальных IP-адресов и их преобразования в физические адреса (требует привилегий sudo);

4. Модуль самодиагностики видеокамер selfdiagnosis - модуль самодиагностики работы видеокамер;

5. Модуль определения человека human_detection - графический модуль отображения распознавания человека и вывода сообщений. Состоит из двух подмодулей: подключение к камере с получением потока и инициализацией нейронной сети, распознавание человека с помощью нейронной сети. Работа и обучение нейронной сети на данном уровне рассматриваться не будет.

Программная архитектура функций в модулях системы представлена на Рис. 2. Модуль camera_connection представляет следующие функциональные компоненты:

- main() - основная функция модуля, берет информацию из конфигураций для подключения к камерам, организует многопоточность просмотра изображений с камеры
- cam() - определяет к какому количеству камер и с какими параметрами необходимо подключиться
- **connect**(name, mac, protocol, login, password, port, width, height, fps, frames, savedir, obj) – выбор настроек, тестирование и подключение к камере(-ам) с сохранением пути до хранилища с видеоданными
- **stream**(cap, width, height, fps, frames, dir, name, obj) – обработка видеопотока(-ов) в режиме реального времени с сохранением видео, записью скриншотов и обработкой простых и сложных событий (в зависимости от используемого модуля)
- stream_unstable() - то же, что и stream, с лругим алгоритмом отрисовки кадров (в случае нестабильности потока)
- timenow() - определяет текущую дату и время

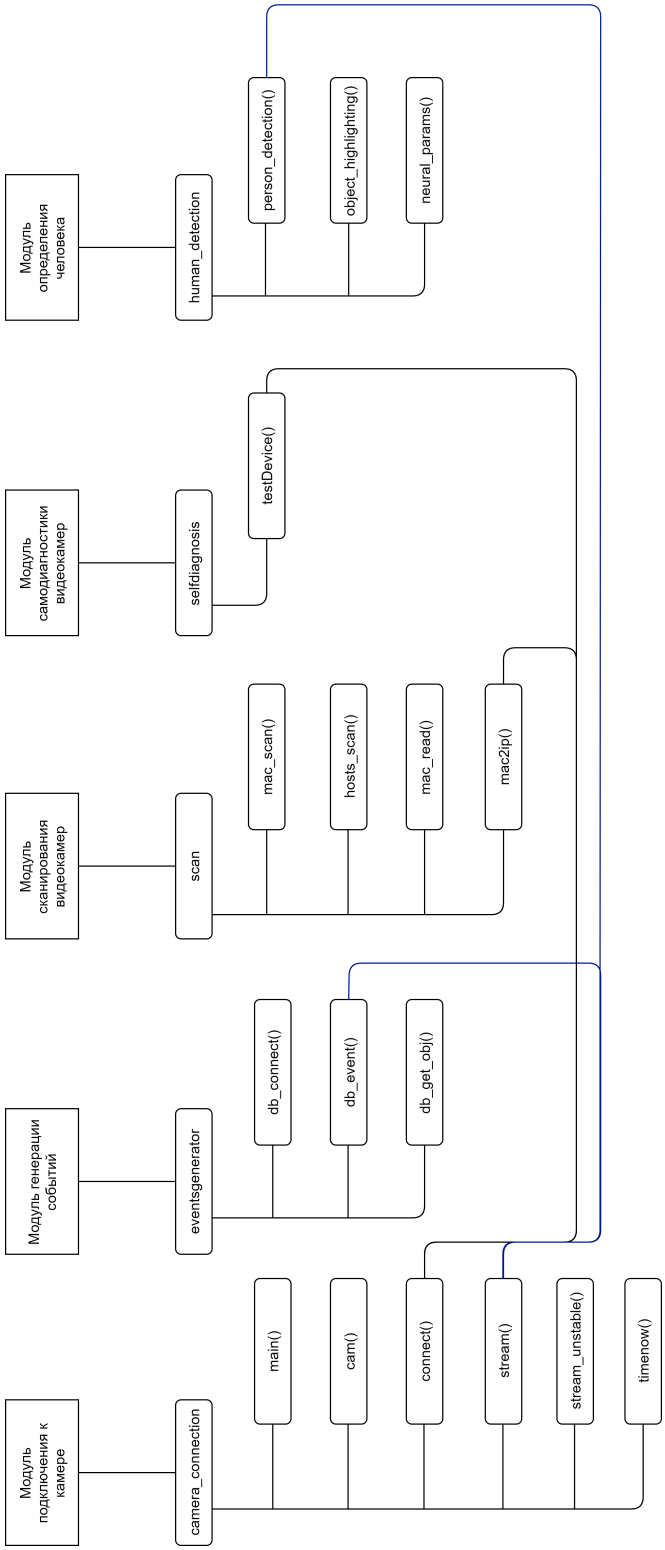

Рис. 2: Программная архитектура модулей и их функций

Модуль eventsgenerator представляет следующие функциональные компоненты:

- db_connect() - подключение к БД MongoDB
- db_event(n, msg, name) – отправка событий в БД MongoDB
- get_obj(id) – поиск и получение объекта коллекции базы данных по id

Модуль scan представляет следующие функциональные компоненты:

- mac_scan(hosts) - сканирование хостов адресного пространства
- mac_host() - список хостов локального адресного пространства без root-доступа
- mac_read(file) - функция чтения MAC-адресов
- mac2ip(mac) - функция, преобразующая MAC-адрес в IP-адрес в случае, если он доступен в локальной сети

Модуль selfdiagnosis представляет следующие функциональные компоненты:

- testDevice(source) - функция проверки работоспособности видеокамеры (досягаемость камеры и извлечение видеопотока)

Модуль human_detection представляет следующие функциональные компоненты:

- person_detection()
- object_highlighting()
- neural_params()

Модуль представляет следующие функциональные компоненты:

- connect(name, mac, protocol, login, password, port, weight, height, fps, frames, savedir, obj)- соединение с камерой
- stream(cap, width, height, fps, frames, dir, name, obj)- инициализация работы видеопотока с записью видеофайлов, дальнейший вызов функции распознавания человека
- person_detection(frame, net, CLASSES, COLORS, text) - выделение человека в прямоугольник и вывод сообщения на экран (скриншот)

Стратегия блочного тестирования

Первый вид тестирования, которому будет подвержена система - блочное тестирование. Набор входных параметров для функций connect, stream модуля camera_connection:

- name -название видеокамеры
- mac – физическое устройство видеокамеры

- protocol – протокол подключения к камере (напр., rtsp, http)
- login – логин для подключения к камере
- password – пароль для подключения к камере
- port – порт подключения к камере
- width – ширина получаемого кадра
- height – высота получаемого кадра
- fps – количество получаемых кадров в секунду во время подключения к камере
- frames – количество кадров для обработки в модуле
- savedir – директория сохранения видеофайла (видеоизображения)
- obj – объект из базы данных MongoDB
- cap – экземпляр класса захвата видео (VideoCapture)
- width – ширина получаемого кадра во время передачи потока
- height – высота получаемого кадра во время передачи потока
- fps – количество получаемых кадров в секунду во время подключения к камере во время передачи потока
- frames – количество кадров для обработки в модуле
- dir – директория сохранения видеофайла (видеоизображения)
- name – название события для сохранения в базе данных MongoDB
- obj – объект из базы данных MongoDB

II. Блочное тестирование будет проводиться для функций модуля eventsgenerator:
Набор входных параметров для функций connect, stream модуля eventsgenerator:

- n - номер события (общий для всех сервисов)
- msg - сообщение события
- name - отправитель события

- id - идентификатор конфигурации

III. Блочное тестирование будет проводиться для функций модуля scan:

Набор входных параметров для функций connect, stream модуля scan:

- hosts - список хостов для сканирования (напр., '192.168.1.0/24')
- file - имя файла для чтения (шаблон: AA:BB:CC:DD:EE:FF, каждый MAC-адрес на новой строке)
- mac - физический адрес устройства для поиска

IV. Блочное тестирование будет проводиться для функций модуля selfdiagnosis:

Набор входных параметров для функций testSource модуля selfdiagnosis:

- source - источник получения видеопотока (в локальном случае: 0..n, в глобальном случае: PROTOCOL://LOGIN:PASSWORD@IP:PORT)

V. Блочное тестирование будет проводиться для функций модуля human_detection:

Набор входных параметров для функций connect, stream, person_detection модуля human_detection:

- cap – экземпляр класса захвата видео (VideoCapture)
- width – ширина получаемого кадра
- height – высота получаемого кадра
- fps – количество получаемых кадров в секунду во время подключения к камере
- frames – количество кадров для обработки в модуле
- dir – директория сохранения видеофайла (видеоизображения)
- name – название события для сохранения в базе данных MongoDB
- obj – объект из базы данных MongoDB
- name -название видеокамеры
- mac – физическое устройство видеокамеры
- protocol – протокол подключения к камере (напр., rtsp, http)
- login – логин для подключения к камере

- password – пароль для подключения к камере
- frame — текущий кадр
- net — ссылка нейронную сеть
- CLASSES — заранее подготовленный список классов объектов, участвует в выводе на экран
- COLORS – цвета объектов интерфейса
- text – сообщение о присутствии человека

Стратегия интеграционного тестирования

Второй этап - интеграционное тестирование. Для проведения интеграционного тестирования необходимо определить все возможные входные данные, соответствующие им ожидаемые результаты.

Интеграционное тестирование будет проведено для следующих взаимодействий между модулями:

Схема интеграционного тестирования для соответствующих тестов представлена на Рис. 3. В схеме представлены последовательные шаги, которые будут реализовывать интеграционное тестирование.

- Шаг 1 - Тест 5
- Шаг 2 - Тест 7
- Шаг 3 - Тест 6
- Шаг 4 - Тест 8
- Шаг 5 - Тест 3
- Шаг 6 - Тест 2
- Шаг 7 - Тест 1
- Шаг 8 - Тест 4

Более подробное описание проводимых тестов в разделе примеров тестов.

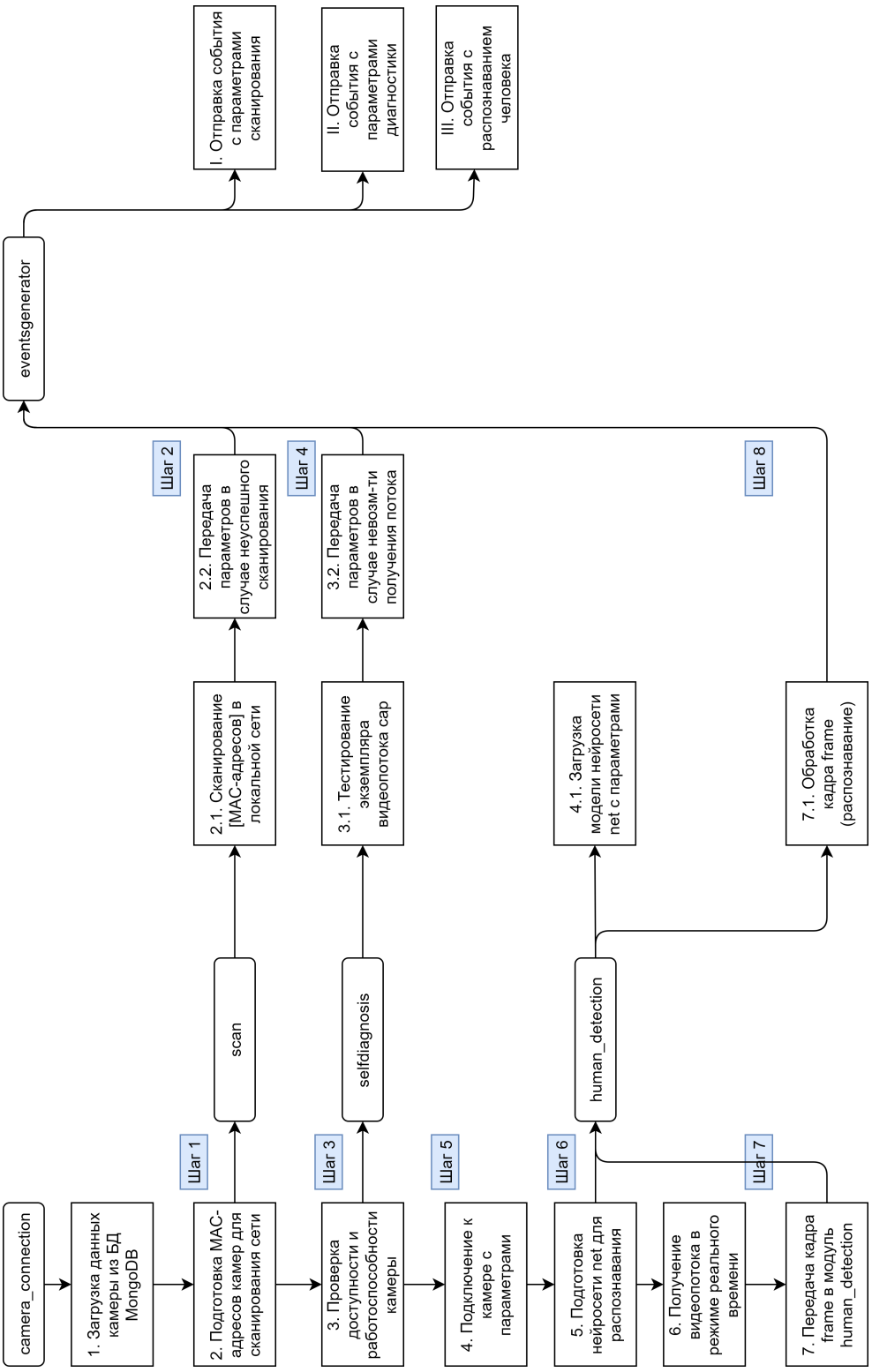

Рис. 3: Схема интеграционного тестирования

Стратегия аттестационного тестирования

Аттестационное тестирование будет проводиться методом «живого человека». В роли такого человека выступает сам автор тестирования.

Тестирующий человек, по заранее заданным инструкциям (TestCases), производит тре-

буемые действия и сверяется с заранее заданными результатами. Тест считается пройденным, если ожидаемый результат совпадает с фактическим результатом. В противном случае тест считается не пройденным.

Стратегия нагрузочного тестирования

Четвертый этап – нагрузочное тестирование. Нагрузочное тестирование – это процесс умышленной нагрузки системы, с целью определения показателей производительности, времени отклика, проверки соответствия требованиям, которые были предъявлены к данной системе или отдельному устройству. Целью данного тестирования является оценка производительности и работоспособности тестируемого модуля.

Нагрузочные тесты:

1. Подключение к 10 камерам одновременно
2. Подключение к удаленной по местоположению камере
3. Распознавание человека с очень высокой точностью 99
4. Создание и отправка события большой длины
5. Сканирование ip-адреса в сети с большим количеством одновременно работающих устройств
6. Одновременное сканирование большого количества mac-адресов

Критерий прохождения тестов

Тест считается успешно пройденным, если ожидаемый и фактический результаты совпадают. Если тест завершается неудачей, то перед принятием решения целесообразно проверить правильность самого теста. Если тест завершился неудачей и тест реализован правильно, то производится заключение о найденной ошибке. Тестирование считается пройденным, если во время его прохождения не выявлено критических ошибок, а процент не пройденных тестов меньше 1

Критерий приостановки тестов

Тестирование должно быть приостановлено, если количество не пройденных тестов превысит 10% от их общего количества. Тестирование должно быть приостановлено при обнаружении критических ошибок.

Критерий возобновления тестирования

Тестирование возобновляется после исправления ошибок, выявленных при предыдущем тестировании.

Оборудование для проведения тестирования

Для проведения тестирования используется настольный компьютер с установленной программой на язык программирования Python3.

3 Детальный план тестирования

3.1 Блочное тестирование

Общие параметры тестирования для модулей. Module camera_connection: 1) connect(..) – функция не выдает значений. После завершения происходит переход к функции stream(..). stream(..) – выдает видеопоток подключенной камеры, выводит сообщения

2) В случае успеха функция connect() закончит свою работу и перейдет к функции stream(). В случае неуспеха будет выдана ошибка, связанная с каким-либо из параметров во время подключения (например, видеопоток камеры занят и недоступен в данный момент). В случае успеха функция stream() вернет видеопоток (т.е. изображение с камеры), получаемое в режиме реального времени. В случае неуспеха будет выдана ошибка, связанная с прерыванием получения потока и/или другими сторонними ошибками, независящими от работы модуля.

3) В случае если к камере невозможно подключиться по тем или иным причинам будет выдана ошибка о невозможности подключения. В том числе в базу данных MongoDB будет отправлено событие.

Module eventsgenerator: 1) db_event(n, msg, name) – получение кода об успешной загрузке события (возвращение id нового события в случае, если его создание прошло успешно) get_obj(id) – получение объекта в json-виде в случае успеха, получение объекта типа None в случае неуспеха.

2) В случае успешного отправления события выведется id созданного события. В случае неудачного отправления события выведется ошибка.

3) В случае отсутствия доступа к базе данных событие не будет отправлено. Неправильная работа модуля детектируется на уровне реализации. В данном случае: каждому

n соответствует номер события (от 1 и так далее) с определенным набором параметров (каждый параметр относится к определенному виду событий).

Module scan: 1) macs – список из найденных MAC-адресов ips – список из соответствующих MAC-ам IP-адресов

2) В случае успешного сканирования вернется список из mac и ip адресов. В случае неуспешного сканирования вернется 2 пустых списка.

3) В случае невозможности поиска адресов или сбоя работы модуля nmap будет возвращена ошибка. Неправильная работа модуля детектируется на уровне реализации. В данном случае это скан портов с помощью специальных ключей nmap.

Module selfdiagnosis: 1) Выводит сообщение об успехе и возвращает единицу, если камера доступна и к ней можно подключиться. Выводит сообщение о неуспехе и возвращает ноль, если камера недоступна и/или к ней нельзя подключиться. Отправляет событие в базу данных о попытке подключения к недоступной камере.

2) В случае успешной попытки подключения к камере вернется 1. В случае неуспешной попытки подключения к камере вернется 0.

3) В случае отсутствия доступа к камере основной модуль не будет запущен, а в базе данных появится событие. Неправильная работа модуля детектируется на уровне реализации. Ввиду простоты функции с единственным параметром наличие аномальной работы модуля не подозревается.

Module human_detection: 1) В случае успешного распознавания присутствия человека будет возвращено число 1. В случае неуспешного распознавания присутствия человека (вероятно, его отсутствие) будет возвращено число 0. Начинает работать обнаружение присутствия человека. Если человек в кадре то создается событие «Human detected» , и скриншот события можно просмотреть в Web-интерфейсе. Через некоторое время опять происходит проверка, если человек ушел ,то уже создается событие Human leaving, скриншот также можно просмотреть.

2) В случае успеха функция connect() закончит свою работу и перейдет к функции stream(). В случае неуспеха будет выдана ошибка, связанная с каким-либо из параметров во время подключения (например, видеопоток камеры занят и недоступен в данный момент). В случае успеха функция stream() вернет видеопоток (т.е. изображение с камеры), получаемое в режиме реального времени. В случае неуспеха будет выдана ошибка, связанная с прерыванием получения потока и/или другими сторонними ошибками, независящими от работы модуля. (это в подключении и получении потока).

6) Модуль требователен к ресурсам памяти видеокарты (GPU) и процессору (CPU),

вследствие чего может появиться нехватка ресурсов компьютера. В таком случае произойдет вынужденная остановка сервиса с диагностическим сообщением об ошибке.

Таблица 1: Модуль подключения к камерам (camera_connection)

Тест	**1**
Цель теста (описание)	Тест проверяет корректность полученного из базы данных и используемого в модуле значения name - название видеокамеры
Тип теста	Общий (блочный)
Объект тестирования (модуль, интерфейс или функциональность)	connect()
Входные параметры	name - строка
Косвенные данные	Нет
Ожидаемый результат	Значение получено из базы данных, произошло успешное подключение к видеокамере с параметром name

Таблица 2: Модуль подключения к камерам (camera_connection)

Тест	**2**
Цель теста (описание)	Тест проверяет корректность полученного из базы данных и используемого в модуле значения mac – физическое устройство видеокамеры
Тип теста	Общий (блочный)
Объект тестирования (модуль, интерфейс или функциональность)	connect()
Входные параметры	mac - строка
Косвенные данные	Нет
Ожидаемый результат	Значение получено из базы данных, произошло успешное подключение к видеокамере с параметром mac

Таблица 3: Модуль подключения к камерам (camera_connection)

Тест	**3**
Цель теста (описание)	Тест проверяет корректность полученного из базы данных и используемого в модуле значения protocol – протокол подключения к камере (напр., rtsp, http)
Тип теста	Общий (блочный)
Объект тестирования (модуль, интерфейс или функциональность)	connect()
Входные параметры	protocol - строка
Косвенные данные	Нет
Ожидаемый результат	Значение получено из базы данных, произошло успешное подключение к видеокамере с параметром protocol

Таблица 4: Модуль подключения к камерам (camera_connection)

Тест	**4**
Цель теста (описание)	Тест проверяет корректность полученного из базы данных и используемого в модуле значения login – логин для подключения к камере
Тип теста	Общий (блочный)
Объект тестирования (модуль, интерфейс или функциональность)	connect()
Входные параметры	login - строка
Косвенные данные	Нет
Ожидаемый результат	Значение получено из базы данных, произошло успешное подключение к видеокамере с параметром login

Таблица 5: Модуль подключения к камерам (camera_connection)

Тест	**5**
Цель теста (описание)	Тест проверяет корректность полученного из базы данных и используемого в модуле значения password – пароль для подключения к камере
Тип теста	Общий (блочный)
Объект тестирования (модуль, интерфейс или функциональность)	connect()
Входные параметры	password - строка
Косвенные данные	Нет
Ожидаемый результат	Значение получено из базы данных, произошло успешное подключение к видеокамере с параметром password

Таблица 6: Модуль подключения к камерам (camera_connection)

Тест	**6**
Цель теста (описание)	Тест проверяет корректность полученного из базы данных и используемого в модуле значения port – порт подключения к камере
Тип теста	Общий (блочный)
Объект тестирования (модуль, интерфейс или функциональность)	connect()
Входные параметры	port – целое число
Косвенные данные	Нет
Ожидаемый результат	Значение получено из базы данных, произошло успешное подключение к видеокамере с параметром port

Таблица 7: Модуль подключения к камерам (camera_connection)

Тест	**7**
Цель теста (описание)	Тест проверяет корректность полученного из базы данных и используемого в модуле значения width – ширина получаемого кадра
Тип теста	Общий (блочный)
Объект тестирования (модуль, интерфейс или функциональность)	connect()
Входные параметры	width – целое число
Косвенные данные	Нет
Ожидаемый результат	Значение получено из базы данных, произошло успешное подключение к видеокамере с параметром width

Таблица 8: Модуль подключения к камерам (camera_connection)

Тест	**8**
Цель теста (описание)	Тест проверяет корректность полученного из базы данных и используемого в модуле значения height – высота получаемого кадра
Тип теста	Общий (блочный)
Объект тестирования (модуль, интерфейс или функциональность)	connect()
Входные параметры	height – целое число
Косвенные данные	Нет
Ожидаемый результат	Значение получено из базы данных, произошло успешное подключение к видеокамере с параметром height

Таблица 9: Модуль подключения к камерам (camera_connection)

Тест	**9**
Цель теста (описание)	Тест проверяет корректность полученного из базы данных и используемого в модуле значения fps – количество получаемых кадров в секунду во время подключения к камере
Тип теста	Общий (блочный)
Объект тестирования (модуль, интерфейс или функциональность)	connect()
Входные параметры	fps – целое число
Косвенные данные	Нет
Ожидаемый результат	Значение получено из базы данных, произошло успешное подключение к видеокамере с параметром fps

Таблица 10: Модуль подключения к камерам (camera_connection)

Тест	**10**
Цель теста (описание)	Тест проверяет корректность полученного из базы данных и используемого в модуле значения frames – количество кадров для обработки в модуле
Тип теста	Общий (блочный)
Объект тестирования (модуль, интерфейс или функциональность)	connect()
Входные параметры	frames – целое число
Косвенные данные	Нет
Ожидаемый результат	Значение получено из базы данных, произошло успешное подключение к видеокамере с параметром frames

Таблица 11: Модуль подключения к камерам (camera_connection)

Тест	**11**
Цель теста (описание)	Тест проверяет, как поведет себя модуль в случае получения некорректно полученного из базы данных и используемого в модуле значения savedir – директория сохранения видеофайла (видеоизображения)
Тип теста	Негативный (блочный)
Объект тестирования (модуль, интерфейс или функциональность)	connect()
Входные параметры	savedir – строка
Косвенные данные	Нет
Ожидаемый результат	Значение получено из базы данных. Произошло успешное подключение к видеокамере с параметром savedir. Но в момент сохранения модуль выводит ошибку о том, что директория не существует

Таблица 12: Модуль подключения к камерам (camera_connection)

Тест	**12**
Цель теста (описание)	Тест проверяет корректность полученного из базы данных и используемого в модуле значения obj – объект из базы данных MongoDB
Тип теста	Негативный (блочный)
Объект тестирования (модуль, интерфейс или функциональность)	connect()
Входные параметры	obj – объект в виде строки
Косвенные данные	Нет
Ожидаемый результат	Значение получено из базы данных. Сохранение с пмощью id в obj в базу данных происходит с ошибкой из-за отсутствия в базе данных.

Таблица 13: Модуль подключения к камерам (camera_connection)

Тест	**13**
Цель теста (описание)	Тест проверяет корректность полученного из базы данных и используемого в модуле значения cap – экземпляр класса захвата видео (VideoCapture)
Тип теста	Негативный (блочный)
Объект тестирования (модуль, интерфейс или функциональность)	stream()
Входные параметры	cap - экземпляр класса
Косвенные данные	Нет
Ожидаемый результат	Значение получено из базы данных. Успешное получение видеопотока невозможно из-за его некорректности во время обработки библиотекой OpenCV.

Таблица 14: Модуль подключения к камерам (camera_connection)

Тест	**14**
Цель теста (описание)	Тест проверяет корректность полученного из базы данных и используемого в модуле значения width – ширина получаемого кадра во время передачи потока
Тип теста	Негативный (блочный)
Объект тестирования (модуль, интерфейс или функциональность)	stream()
Входные параметры	width - целое число
Косвенные данные	Нет
Ожидаемый результат	Значение получено из базы данных. Модуль выводит ошибку из-за некорректно полученного значения width.

Таблица 15: Модуль подключения к камерам (camera_connection)

Тест	**15**
Цель теста (описание)	Тест проверяет корректность полученного из базы данных и используемого в модуле значения height – высота получаемого кадра во время передачи потока
Тип теста	Негативный (блочный)
Объект тестирования (модуль, интерфейс или функциональность)	stream()
Входные параметры	height - целое число
Косвенные данные	Нет
Ожидаемый результат	Значение получено из базы данных. Модуль выводит ошибку из-за некорректно полученного значения width.

Таблица 16: Модуль подключения к камерам (camera_connection)

Тест	**16**
Цель теста (описание)	Тест проверяет корректность полученного из базы данных и используемого в модуле значения fps – количество получаемых кадров в секунду во время подключения к камере во время передачи потока
Тип теста	Общий (блочный)
Объект тестирования (модуль, интерфейс или функциональность)	stream()
Входные параметры	fps - целое число
Косвенные данные	Нет
Ожидаемый результат	Значение получено из базы данных. Не произошло успешного подключения к видеокамере из-за некорректного параметра fps

Таблица 17: Модуль подключения к камерам (camera_connection)

Тест	**17**
Цель теста (описание)	Тест проверяет корректность полученного из базы данных и используемого в модуле значения name – название события для сохранения в базе данных MongoDB
Тип теста	Негативный (блочный)
Объект тестирования (модуль, интерфейс или функциональность)	stream()
Входные параметры	name - строка
Косвенные данные	Нет
Ожидаемый результат	Значение получено из базы данных. Не произошло успешного подключения к видеокамере из-за некорректности параметра name

3.2 Интеграционное тестирование

Взаимодействие модулей camera_connection и human_detection

Таблица 18: Модуль подключения к камерам (camera_connection) и модуль распознавания человека (human_detection)

Тест	**1**
Цель теста (описание)	Тест проверяет корректность передачи видеокадра frame из модуля camera_connection в модуль human_detection
Тип теста	Общий (интеграционный)
Объект тестирования (модуль, интерфейс или функциональность)	camera_connection() + human_detection()
Входные параметры	frame - кадр
Косвенные данные	Нет
Ожидаемый результат	Значение получено из первоначального модуля camera_connection() и успешно обработано модулем human_detection() без появления непредвиденных ошибок и аномалий

Таблица 19: Модуль подключения к камерам (camera_connection) и модуль распознавания человека (human_detection)

Тест	**2**
Цель теста (описание)	Тест проверяет корректность передачи экземпляра нейросети net из модуля camera_connection в модуль human_detection
Тип теста	Общий (интеграционный)
Объект тестирования (модуль, интерфейс или функциональность)	camera_connection() + human_detection()
Входные параметры	net - экземпляр нейросети
Косвенные данные	Нет
Ожидаемый результат	Значение получено из первоначального модуля camera_connection() и успешно обработано модулем human_detection() без появления непредвиденных ошибок и аномалий

Таблица 20: Модуль подключения к камерам (camera_connection) и модуль генерации событий (eventsgenerator)

Тест	**3**
Цель теста (описание)	Тест проверяет корректность передачи параметров событий из модуля camera_connection в модуль eventsgenerator
Тип теста	Общий (интеграционный)
Объект тестирования (модуль, интерфейс или функциональность)	camera_connection() + eventsgenerator()
Входные параметры	name, savedir, cameraobj, cameraname
Косвенные данные	Нет
Ожидаемый результат	Значение получено из первоначального модуля camera_connection() и успешно обработано модулем human_detection() без появления непредвиденных ошибок и аномалий. Успешно отправлено событие и сохранено в базе данных.

Таблица 21: Модуль распознавания человека (human_detection) и модуль генерации событий (eventsgenerator)

Тест	**4**
Цель теста (описание)	Тест проверяет корректность передачи параметров событий из модуля human_deteection в модуль eventsgenerator
Тип теста	Общий (интеграционный)
Объект тестирования (модуль, интерфейс или функциональность)	human_detection() + eventsgenerator()
Входные параметры	name, savedir, eventid, eventname
Косвенные данные	Нет
Ожидаемый результат	Значение получено из первоначального модуля camera_connection() и успешно обработано модулем human_detection() без появления непредвиденных ошибок и аномалий. Успешно отправлено событие и сохранено в базе данных.

Таблица 22: Модуль подключения к камерам (camera_connection) и модуль сканирования (scan)

Тест	**5**
Цель теста (описание)	Тест проверяет корректность передачи mac-адресов из модуля camera_connection в модуль scan()
Тип теста	Общий (интеграционный)
Объект тестирования (модуль, интерфейс или функциональность)	camera_connection() + scan()
Входные параметры	mac[] - массив mac-адресов
Косвенные данные	Нет
Ожидаемый результат	Значение получено из первоначального модуля camera_connection() и успешно обработано модулем scan() без появления непредвиденных ошибок и аномалий. Возвращен список из ip-адресов для подключения.

Таблица 23: Модуль подключения к камерам (camera_connection()) и модуль самодиагностики (selfdiagnosis)

Тест	**6**
Цель теста (описание)	Тест проверяет корректность передачи экземпляра камеры cap из модуля camera_connection в модуль selfdiagnosis()
Тип теста	Общий (интеграционный)
Объект тестирования (модуль, интерфейс или функциональность)	camera_connection() + selfdiagnosis()
Входные параметры	cap - экземпляр видеокамеры
Косвенные данные	Нет
Ожидаемый результат	Значение получено из первоначального модуля camera_connection() и успешно обработано модулем selfdiagnosis() без появления непредвиденных ошибок и аномалий. Возвращено сообщение о возможности/невозможности подключения к видеокамере.

Таблица 24: Модуль сканирования (scan()) и модуль генерации событий (eventsgenerator)

Тест	**7**
Цель теста (описание)	Тест проверяет корректность передачи параметров сканирования из модуля scan() в модуль eventsgenerator()
Тип теста	Общий (интеграционный)
Объект тестирования (модуль, интерфейс или функциональность)	scan() + eventsgenerator()
Входные параметры	camera, status, eventid, eventname
Косвенные данные	Нет
Ожидаемый результат	Значение получено из первоначального модуля scan() и успешно обработано модулем eventsgenerator() без появления непредвиденных ошибок и аномалий.

Таблица 25: Модуль самодиагностики (selfdiagnosis()) и модуль генерации событий (eventsgenerator)

Тест	**8**
Цель теста (описание)	Тест проверяет корректность передачи параметров самодиагностики из модуля selfdiagnosis() в модуль eventsgenerator()
Тип теста	Общий (интеграционный)
Объект тестирования (модуль, интерфейс или функциональность)	selfdiagnosis() + eventsgenerator()
Входные параметры	mac, ip, status, eventid, eventname
Косвенные данные	Нет
Ожидаемый результат	Значение получено из первоначального модуля selfdiagnosis() и успешно обработано модулем eventsgenerator() без появления непредвиденных ошибок и аномалий.

3.3 Аттестационное тестирование

Таблица 26: Модуль camera_connection()

Тест	**1**
Цель теста (описание)	Тест проверяет корректность подключения к видеокамере в случае, когда разрешение (ширина и высота) кадра задана меньше стандартного в модуле camera_connection()
Тип теста	Общий (аттестационный)
Объект тестирования (модуль, интерфейс или функциональность)	Функциональность модуля camera_connection()
Входные параметры	width, height
Косвенные данные	Нет
Ожидаемый результат	Значение обработано в модуле camera_connection() и на выходе получено видеоизображение с параметрами width height.

Таблица 27: Модуль camera_connection()

Тест	**2**
Цель теста (описание)	Тест проверяет корректность подключения к видеокамере в случае, когда количество кадров в секунду отличается от стандартного в модуле camera_connection()
Тип теста	Общий (аттестационный)
Объект тестирования (модуль, интерфейс или функциональность)	Функциональность модуля camera_connection()
Входные параметры	fps
Косвенные данные	Нет
Ожидаемый результат	Значение обработано в модуле camera_connection() и на выходе получено видеоизображение с параметром fps.

Таблица 28: Модуль camera_connection()

Тест	**3**
Цель теста (описание)	Тест проверяет корректность подключения к видеокамере в случае, когда получение кадров с камеры является нестабильным в модуле camera_connection()
Тип теста	Общий (аттестационный)
Объект тестирования (модуль, интерфейс или функциональность)	Функциональность модуля camera_connection()
Входные параметры	fps_unstable
Косвенные данные	Нет
Ожидаемый результат	Значение обработано в модуле camera_connection() и на выходе получено видеоизображение с нестабильным fps.

Таблица 29: Модуль camera_connection()

Тест	**4**
Цель теста (описание)	Тест проверяет корректность подключения к видеокамере в случае, когда разрешение (ширина и высота) кадра задана в виде нечитаемых параметров в модуле camera_connection()
Тип теста	Негативный (аттестационный)
Объект тестирования (модуль, интерфейс или функциональность)	Функциональность модуля camera_connection()
Входные параметры	width, height
Косвенные данные	Нет
Ожидаемый результат	Значение обработано в модуле camera_connection(). На выходе не получено видеоизображения ввиду некорректности данных. Возникла ошибка.

Таблица 30: Модуль camera_connection()

Тест	**5**
Цель теста (описание)	Тест проверяет корректность подключения к видеокамере в случае, когда количество кадров в секунду задано в виде числа с плавающей точкой вместо целочисленного значения в модуле camera_connection()
Тип теста	Негативный (аттестационный)
Объект тестирования (модуль, интерфейс или функциональность)	Функциональность модуля camera_connection()
Входные параметры	fps
Косвенные данные	Нет
Ожидаемый результат	Значение обработано в модуле camera_connection(). Появление ошибки о некорректности заданного fps

Таблица 31: Модуль camera_connection()

Тест	**6**
Цель теста (описание)	Тест проверяет корректность подключения к видеокамере в случае, когда получение кадров с камеры является нестабильным с плавающей точкой в модуле camera_connection()
Тип теста	Негативный (аттестационный)
Объект тестирования (модуль, интерфейс или функциональность)	Функциональность модуля camera_connection()
Входные параметры	fps_unstable
Косвенные данные	Нет
Ожидаемый результат	Значение обработано в модуле camera_connection(). Появление ошибки о некорректности заданного fps

3.4 Нагрузочное тестирование

Таблица 32: Модуль camera_connection()

Тест	**1**
Цель теста (описание)	Тест проверяет корректность работы модуля в случае подключения к большому количеству видеокамер одновременно в модуле camera_connection()
Тип теста	Общий (нагрузочный)
Объект тестирования (модуль, интерфейс или функциональность)	Нагрузочная функциональность модуля camera_connection()
Входные параметры	cameras[]
Косвенные данные	Нет
Ожидаемый результат	Стабильная работа модуля camera_connection(). Все 10 камер показывают изображение в режиме реального времени.

Таблица 33: Модуль camera_connection()

Тест	**2**
Цель теста (описание)	Тест проверяет успешную передачу видеопотока в режиме реального времени, находящуюся на большом расстоянии от источника получения видео в модуле camera_connection()
Тип теста	Общий (нагрузочный)
Объект тестирования (модуль, интерфейс или функциональность)	Нагрузочная функциональность модуля camera_connection()
Входные параметры	camera(ip, port)
Косвенные данные	Нет
Ожидаемый результат	Стабильная работа модуля camera_connection(). Видеоизображение успешно показывается без особых (непредвиденных) задержек.

Таблица 34: Модуль human_detection()

Тест	**3**
Цель теста (описание)	Тест проверяет успешное распознавание человека с точностью 99% и более в модуле human_detection()
Тип теста	Общий (нагрузочный)
Объект тестирования (модуль, интерфейс или функциональность)	Нагрузочная функциональность модуля human_detection()
Входные параметры	camera(ip, port)
Косвенные данные	Нет
Ожидаемый результат	Стабильная работа модуля human_detection(). Человек успешно распознается (вероятность отображается на экране).

Таблица 35: Модуль human_detection()

Тест	**4**
Цель теста (описание)	Тест проверяет успешную проверку событий в базу данных MongoDB большой длины в модуле eventsgenerator
Тип теста	Общий (нагрузочный)
Объект тестирования (модуль, интерфейс или функциональность)	Нагрузочная функциональность модуля human_detection()
Входные параметры	eventid
Косвенные данные	Нет
Ожидаемый результат	Стабильная работа модуля eventsgenerator(). Событие успешно создалось в базе данных.

Таблица 36: Модуль human_detection()

Тест	**5**
Цель теста (описание)	Тест проверяет корректность сканирования локальной сети с большим количеством устройств в модуле scan()
Тип теста	Общий (нагрузочный)
Объект тестирования (модуль, интерфейс или функциональность)	Нагрузочная функциональность модуля scan()
Входные параметры	mac
Косвенные данные	Нет
Ожидаемый результат	Стабильная работа модуля scan(). Найден необходимый MAC-адрес устройства.

Таблица 37: Модуль human_detection()

Тест	**6**
Цель теста (описание)	Тест проверяет корректность сканирования одновременно нескольких MAC-устройств в модуле scan()
Тип теста	Общий (нагрузочный)
Объект тестирования (модуль, интерфейс или функциональность)	Нагрузочная функциональность модуля scan()
Входные параметры	macs[]
Косвенные данные	Нет
Ожидаемый результат	Стабильная работа модуля scan(). Найдены необходимые MAC-адреса устройства.

3.5 Покрытие кода тестами

$$T_cov = \frac{L_tc}{L_code} * 100\% \quad (1)$$

- Tcov - тестовое покрытие;
- Ltc - - количество строк кода, покрытых тестами;
- Lcode - общее количество строк кода.

Тогда

$$T_cov = (355/625) * 100 \quad (2)$$

3.6 Примеры реализации тестов

```
import camera_connection

def test_camera():
        x = eventsgenerator.get_obj(cid)
    print("X: ", x)
    ps = []
        pass = x['password'])
        camera_handle_pass = "1H3fy0guyhGGnbfx"
    assert pass == camera_handle_pass, "Should be equal"
        print("From DataBase:", pass)
        print("From handle:", camera_handle_pass)

if __name__ == "__main__":
    test_sum()
    print("Test 10 is passed")
```

4 Журнал тестирования

Матрица трассируемости представлена на Рис. 4.

Матрица трассируемости	Тест	Тип теста	Вид теста
Корректность передачи данных во время взаимодействия с базой данных MongoDB	1	Общий	Блочный
Корректность передачи данных во время взаимодействия с базой данных MongoDB	2	Общий	Блочный
Корректность передачи данных во время взаимодействия с базой данных MongoDB	3	Общий	Блочный
Корректность передачи данных во время взаимодействия с базой данных MongoDB	4	Общий	Блочный
Корректность передачи данных во время взаимодействия с базой данных MongoDB	5	Общий	Блочный
Корректность передачи данных во время взаимодействия с базой данных MongoDB	6	Общий	Блочный
Корректность передачи данных во время взаимодействия с базой данных MongoDB	7	Общий	Блочный
Корректность передачи данных во время взаимодействия с базой данных MongoDB	8	Общий	Блочный
Корректность передачи данных во время взаимодействия с базой данных MongoDB	9	Общий	Блочный
Корректность передачи данных во время взаимодействия с базой данных MongoDB	10	Общий	Блочный
Успешная обработка неадекватных данных во время взаимодействия с базой данных MongoDB	11	Негативный	Блочный
Успешная обработка неадекватных данных во время взаимодействия с базой данных MongoDB	12	Негативный	Блочный
Успешная обработка неадекватных данных во время взаимодействия с базой данных MongoDB	13	Негативный	Блочный
Успешная обработка неадекватных данных во время взаимодействия с базой данных MongoDB	14	Негативный	Блочный
Успешная обработка неадекватных данных во время взаимодействия с базой данных MongoDB	15	Негативный	Блочный
Успешная обработка неадекватных данных во время взаимодействия с базой данных MongoDB	16	Негативный	Блочный
Успешная обработка неадекватных данных во время взаимодействия с базой данных MongoDB	17	Негативный	Блочный
Целостность передачи данных, отсутствие ошибок и проблемных мест между модулями	18	Общий	Интеграционный
Целостность передачи данных, отсутствие ошибок и проблемных мест между модулями	19	Общий	Интеграционный
Целостность передачи данных, отсутствие ошибок и проблемных мест между модулями	20	Общий	Интеграционный
Целостность передачи данных, отсутствие ошибок и проблемных мест между модулями	21	Общий	Интеграционный
Целостность передачи данных, отсутствие ошибок и проблемных мест между модулями	22	Общий	Интеграционный
Целостность передачи данных, отсутствие ошибок и проблемных мест между модулями	23	Общий	Интеграционный
Целостность передачи данных, отсутствие ошибок и проблемных мест между модулями	24	Общий	Интеграционный
Целостность передачи данных, отсутствие ошибок и проблемных мест между модулями	25	Общий	Интеграционный
Функциональность модуля camera_connection: изменение width, height	26	Общий	Аттестационный
Функциональность модуля camera_connection: изменение fps	27	Общий	Аттестационный
Функциональность модуля camera_connection: изменение unstable_fps	28	Общий	Аттестационный
Функциональность модуля camera_connection: нестабильные значения width, height	29	Негативный	Аттестационный
Функциональность модуля camera_connection: нестабильные значения fps	30	Негативный	Аттестационный
Функциональность модуля camera_connection: нестабильные значения unstable_fps	31	Негативный	Аттестационный
Обработка сразу нескольких видеопотоков (до 10)	32	Общий	Нагрузочный
Подключение к удаленным видеокамерам	33	Общий	Нагрузочный
Успешное распознавание человека с точностью 99%	34	Общий	Нагрузочный
Корректная обработка длинных значений при взаимодействии с базой данных MongoDB	35	Общий	Нагрузочный
Нахождение IP-адресов с помощью MAC-адресов в загруженной локальной сети	36	Общий	Нагрузочный
Сканирование большого количества MAC-адресов одновременно	37	Общий	Нагрузочный

Рис. 4: Матрица трассируемости требований

4.1 Блочное тестирование

17 тестов - 15 успешно

Таблица 38: Блочные тесты

№ теста	**Дата**	**Результат**	**Номер ошибки в журнале**
1	01.12.2019	Пройден	-
2	01.12.2019	Пройден	-
3	01.12.2019	Пройден	-
4	01.12.2019	Пройден	-
5	01.12.2019	Пройден	-
6	01.12.2019	Не пройден	Ошибка 1
7	01.12.2019	Пройден	-
8	01.12.2019	Пройден	-
9	01.12.2019	Пройден	-
10	01.12.2019	Пройден	-
11	01.12.2019	Не пройден	Ошибка 2
12	01.12.2019	Пройден	-
13	01.12.2019	Пройден	-
14	01.12.2019	Пройден	-
15	01.12.2019	Пройден	-
16	01.12.2019	Пройден	-
17	01.12.2019	Пройден	-

4.2 Интеграционное тестирование

8 тестов - 8 успешно

Таблица 39: Интеграционные тесты

№ теста	Дата	Результат	Номер ошибки в журнале
1	01.12.2019	Пройден	-
2	01.12.2019	Пройден	-
3	01.12.2019	Пройден	-
4	01.12.2019	Пройден	-
5	01.12.2019	Пройден	-
6	01.12.2019	Пройден	-
7	01.12.2019	Пройден	-
8	01.12.2019	Пройден	-

4.3 Аттестационное тестирование

3 теста - 3 успешно

Таблица 40: Аттестационные тесты

№ теста	Дата	Результат	Номер ошибки в журнале
1	02.12.2019	Пройден	-
2	02.12.2019	Пройден	-
3	02.12.2019	Пройден	-
4	02.12.2019	Пройден	-
5	02.12.2019	Пройден	-
6	02.12.2019	Пройден	-

4.4 Нагрузочное тестирование

6 тестов - 3 успешно

Таблица 41: Нагрузочные тесты

№ теста	Дата	Результат	Номер ошибки в журнале
1	03.12.2019	Не пройден	Ошибка 3
2	03.12.2019	Пройден	-
3	03.12.2019	Пройден	-
4	03.12.2019	Не пройден	Ошибка 4
5	03.12.2019	Не пройден	Ошибка 5
6	03.12.2019	Пройден	-

5 Журнал найденных ошибок

1. Ошибка 1: Взятое значение port из БД не является числом, вследствие чего модуль прекращает работу
2. Ошибка 2: Взятое значение savedir из БД не существует, вследствие чего модуль прекращает работу
3. Ошибка 3: Подключение к нескольким камерам одновременно приводит к ошибке, вследствие чего модуль прекращает работу
4. Ошибка 4: Сообщение большой длины для сохранения в БД MongoDB приводит к ошибке отправки события, вследствие чего работа модуля сильно замедляется
5. Ошибка 5: Большое количество устройств локальной сети сильно замедляет работу модуля взятие IP-адресов

6 Результаты

Данное тестирование помогло работе проекта выявить ошибки в системе. В ходе блочного, интеграционного, аттестационного и нагрузочного тестирования модулей системы видеонаблюдения в рамках выполнения проекта НИР было выявлено 5 некритических ошибок. Ошибки №1, 2 были исправлены (блочное тестирование). Предположительно, работа системы с технической точки зрения является работоспособной, все заявленные функции выполняются без ошибок. В ходе проведения тестирования были выявлены недостатки, которые требуют рассмотрения и устранения в будущем.