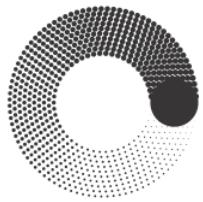


ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ
УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ



МОСКОВСКИЙ ПОЛИТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ

Факультет Информационных технологий
Кафедра Информатики и информационных технологий

направление подготовки

09.03.02 «Информационные системы и технологии»

ЛАБОРАТОРНАЯ РАБОТА № 2

Дисциплина: Теория информационных процессов и систем

Тема: Разработка диаграммы классов и объектов для выбранной информационной системы

Выполнил: студент группы 241-334

Мельников Георгий Валерьевич

(Фамилия И.О.)

Дата, подпись 27.09.25

(Дата)

(Подпись)

Проверил: Рябчикова Анна Валерьевна

(Фамилия И.О., степень, звание)

(Оценка)

Дата, подпись _____

(Дата)

(Подпись)

Замечания: _____

Москва 2025

Лабораторная работа № 2 «Разработка диаграммы классов и объектов для выбранной информационной системы»

Тема «Пропускная система контроля доступа посетителей на режимный объект, организованная на основе пропускного пункта».

Целью данной лабораторной работы является выделение сущностей и отношений между ними, а также в данной лабораторной работе, в соответствии с выбранной темой (которую вы выбрали в прошлой лабораторной работе), вы должны будете полностью описать и разработать диаграмму классов и объектов для своей системы, определите основные виды деятельности (назначение основных операций) рассматриваемого предмета исследования.

Описание классов

1. Класс Visitor (Посетитель)

Атрибуты:

- id : Int - уникальный идентификатор посетителя (например, внутренний код в системе).
- name : String - ФИО посетителя.
- docNumber : Int - номер документа, удостоверяющего личность.

Методы:

- arrive() - событие прибытия на КПП.
- provideDocument() - предоставление документов охране/оператору.

2. Класс Checkpoint (КПП, зона приёма)

Атрибуты:

- location : String - местоположение КПП (например: «КПП №1, въезд со стороны парковки»).
- shiftOperator : String - имя или идентификатор сотрудника, который обслуживает пост.

Методы:

- acceptVisitor(visitor : Visitor) - регистрация факта прихода посетителя.

- `routeToTerminal()` - перенаправление посетителя на терминал регистрации (оператор указывает, куда пройти).

3. Класс RegistrationTerminal (Терминал регистрации / VMS)

Атрибуты:

- `terminalId : String` - уникальный идентификатор терминала.
- `ip : String` - сетевой адрес терминала (для связи с сервером).
- `operatorId : String` - ID сотрудника охраны, работающего за этим терминалом.

Методы:

- `registerVisitor(data)` - регистрация посетителя (ввод данных, сканирование документа, фотография).
- `queryVisitorDB(query)` - проверка по базе (например, не находится ли посетитель в «чёрном списке»).
- `printBadge(pass)` - печать/выдача пропуска (карточки, бейджа или QR-кода).

4. Класс SystemServer (Сервер системы)

Атрибуты:

- `serverId : String` - уникальный ID сервера.
- `ip : String` - IP-адрес сервера.
- `status : String` - текущее состояние (например: «online», «maintenance»).

Методы:

- `authenticateVisitor(data) : boolean` - проверка данных посетителя (разрешить/запретить вход).
- `logEvent(event)` - запись события (прибытие, вход, тревога) в журнал.
- `issueTemporaryKey()` - генерация временного электронного ключа/кода доступа для пропуска.

5. Класс VisitorsDatabase (База данных посетителей)

Атрибуты:

- `dbUri : String` - адрес базы данных.
- `lastBackup : Date` - дата последнего резервного копирования.

Методы:

- `getVisitor(id)` - поиск данных посетителя по ID.
- `saveVisitRecord(record)` - сохранение записи о визите (кто, когда, куда входил).
- `queryBlackList()` - проверка, не включён ли посетитель в список ограничений.

6. Класс ReaderTurnstile (Считыватель / турникет)

Атрибуты:

- `readerId : String` - уникальный ID считывателя.
- `type : enum{RFID, QR, Barcode}` - тип устройства (RFID-карта, QR-код, штрихкод).

Методы:

- `readTag() : TagData` - считывание данных с пропуска/карточки.
- `sendAuthRequest(key)` - отправка считанного ключа на контроллер СКУД для проверки.

7. Класс ACS_Controller (Контроллер СКУД)

Атрибуты:

- `controllerId : String` - ID контроллера.
- `zones : List<Zone>` - список зон, доступ в которые контролирует данный контроллер.

Методы:

- `verifyAccess(tag) : boolean` - проверка, имеет ли данный ключ право доступа.
- `controlGate(gateId, action)` - подача команды воротам/турникету (например, открыть/закрыть).

8. Класс BarrierGate (Шлагбаум / турникет)

Атрибуты:

- `gateId : String` - ID ворот/турникета.
- `state : enum{Open, Closed}` - текущее состояние.

Методы:

- open() - открыть проход.
- close() - закрыть проход.

9. Класс SecurityPanel_CCTV (Панель охраны / CCTV)

Атрибуты:

- consoleId : String - идентификатор пульта охраны.
- cameras : List<Camera> - список подключённых камер.

Методы:

- showFeed(cameraId) - отобразить видеопоток с указанной камеры.
- raiseAlarm(event) - вызвать тревогу (например, при несанкционированной попытке прохода).

Описание связей

1. Visitor - Checkpoint

- **Тип:** зависимость (*dependency*)
- **Почему:** посетитель зависит от функционала КПП, но не является его частью.

2. Checkpoint - RegistrationTerminal

- **Тип:** направленная ассоциация (*association*)
- **Почему:** КПП направляет посетителя к терминалу регистрации.

3. RegistrationTerminal - SystemServer

- **Тип:** направленная ассоциация
- **Почему:** терминал отправляет данные посетителя на сервер для проверки.

4. SystemServer - VisitorsDatabase

- **Тип:** ассоциация (двусторонняя)
- **Почему:** сервер читает/записывает записи в БД; взаимодействие идёт в обе стороны.

5. RegistrationTerminal - ReaderTurnstile

- **Тип:** направленная ассоциация
- **Почему:** терминал выдаёт временный пропуск/ключ, который затем используется на считывателе.

6. ReaderTurnstile - ACS_Controller

- **Тип:** направленная ассоциация
- **Почему:** считыватель передаёт считанные данные контроллеру для проверки.

7. SystemServer - ACS_Controller

- **Тип:** ассоциация (двусторонняя)
- **Почему:** сервер конфигурирует контроллер (правила доступа), контроллер передаёт события обратно на сервер.

8. ACS_Controller - BarrierGate

- **Тип:** направленная ассоциация
- **Почему:** контроллер управляет состоянием шлагбаума/турникета (открыть/закрыть).

9. SystemServer - SecurityPanel_CCTV

- **Тип:** направленная ассоциация
- **Почему:** сервер передаёт события/сигналы тревог на панель охраны.

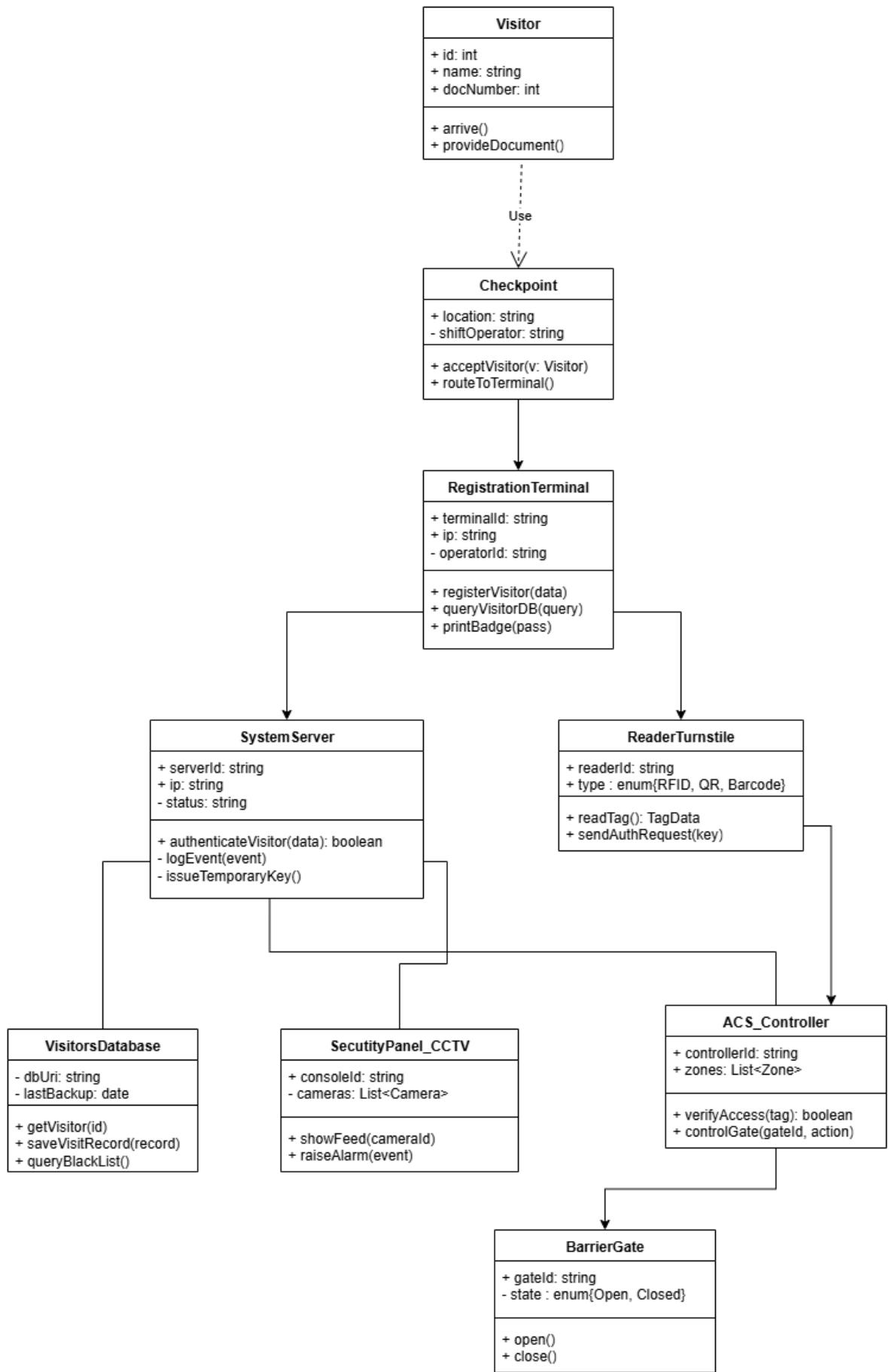


Рисунок 1. Диаграмма классов

Построение диаграммы объектов.

Эта диаграмма позволяет моделировать экземпляры сущностей, которые содержатся в диаграмме классов. На диаграмме объектов показаны множества объектов и отношений между ними в некоторый момент времени. В качестве класса для разработки диаграммы объектов выберем класс “Visitor”.

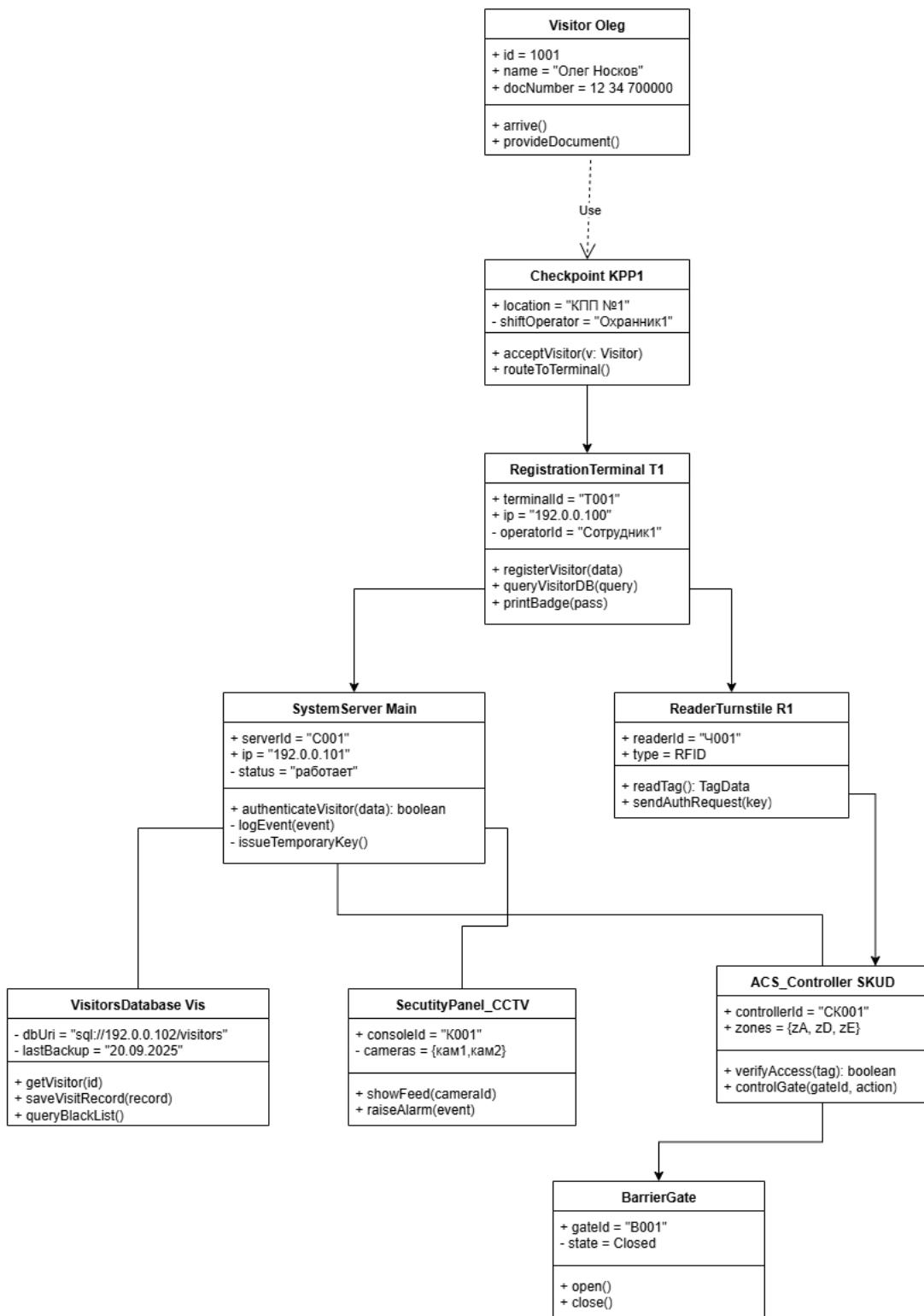


Рисунок 2. Диаграмма объектов