**РОСЖЕЛДОР**

**ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ**

**ВЫСШЕГО ОБРАЗОВАНИЯ**

**СИБИРСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ**

**ПУТЕЙ СООБЩЕНИЯ (СГУПС)**

|  |  |  |  |
| --- | --- | --- | --- |
| **К защите:** |  |  | |
| **Заведующий кафедрой** | **Информационные** | |
| **технологии транспорта** | | |
|  | д-р техн. наук, проф. | |
|  |  | В. И. Хабаров | |
| *подпись* |  | *инициалы, фамилия* | |
|  |  |  | |
| *дата* |  |  | |

**ВЫПУСКНАЯ КВАЛИФИКАЦИОННАЯ РАБОТА (БАКАЛАВРСКАЯ РАБОТА)**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Тема:** | Разработка модуля сетевого взаимодействия для тренажёрного | | | | | |
|  | комплекса оперативного персонала сортировочной горки | | | | | |
|  | |  | БР.БИСТ.03.2022 |  |  |
|  | |  | *шифр документа* |  |  |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  |  | **Выполнил** |  |  |  | **Руководитель** |
|  |  | В.А. Березикова |  |  |  | канд. техн. наук, доц.  Е. Б. Тарасов |
| *подпись* |  | *инициалы, фамилия* |  | *подпись* |  | *инициалы, фамилия* |
|  |  |  |  |  |  |  |
| *дата* |  |  |  | *дата* |  |  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Консультанты по разделам** |  |  |  |  |
| 1 |  |  |  |  |
|  |  |  |  |  |
| 2 |  |  |  |  |
|  |  |  |  |  |
| 3 |  |  |  |  |
|  |  |  |  |  |
| Нормоконтролер работы |  |  |  | ст. преп.  Т. А. Распопина |
|  |  | *подпись* |  |  |
|  |  |  |  |  |
|  |  | *дата* |  |  |

**2022 г.**

**СИБИРСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ**

**ПУТЕЙ СООБЩЕНИЯ (СГУПС)**

Факультет: Бизнес-информатики

Кафедра: Информационные технологии транспорта

Направление: 09.03.02 «Информационные системы и технологии»

Профиль: Интеллектуальные транспортные системы

|  |  |
| --- | --- |
|  | ***УТВЕРЖДАЮ****: зав. кафедрой «Информационные технологии транспорта»*  д-р техн. наук, проф.  В. И. Хабаров |
|  | *«\_\_\_\_»\_\_\_\_\_\_\_\_\_\_ 2022 г.* |

**З А Д А Н И Е**

**на выполнение выпускной квалификационной работы**

|  |  |  |  |
| --- | --- | --- | --- |
| студенту | Березиковой Валерии Алексеевне | | |
|  |  | | |
| 1. Тема «Разработка модуля сетевого взаимодействия для тренажёрного комплекса оперативного персонала сортировочной горки» утверждена приказом № 203/с от «30» мая 2022 г. | | | |
| 2. Задание выдано «12» мая 2022 г. | | | |
| 3. Срок сдачи законченной работы на кафедру «17» июня 2022 г. | | | |
| 4. Исходные данные: данные, полученные в ходе прохождения преддипломной практики | | | |
| 5. Содержание расчетно-пояснительной записки | | | |
| Наименование разделов и вопросов | | Примерное количество страниц | График (сроки) выполнения |
| Введение | | 1 | 13.05.2022 |
| Аналитическое исследование | | 5 | 13.05.2022 |
| Проектирование информационной системы | | 13 | 15.05.2022 |
| Создание приложения | | 37 | 28.05.2022 |
| Заключение | | 1 | 01.06.2022 |

6. Содержание и объемы графической части

|  |  |  |
| --- | --- | --- |
| Наименование графического документа (чертежа, схемы, графика) | Количество  листов  формата А1 | График  (сроки)  выполнения |
| Презентация PowerPoint | 15 | 05.05.2022 |

7. Консультанты по разделам

|  |  |  |
| --- | --- | --- |
| Наименование  раздела | Фамилия, И. О.  консультанта | Подпись консультанта,  дата выдачи задания |
|  |  |  |

|  |  |  |
| --- | --- | --- |
| Руководитель |  | Е. Б. Тарасов |
|  | *(подпись, фамилия, И.О.)* |  |
| Задание к использованию принял |  | В. А. Березикова |
|  | *(подпись студента)* |  |

УДК 004.41

**АННОТАЦИЯ**

В работе 38 страниц, 23 рисунка, 9 таблиц, 10 источников.

Ключевые слова: *тренажерный комплекс, модуль, сетевое взаимодействие.*

Предметная область - тренажерный комплекс оперативного персонала сортировочной горки. Модуль сетевого взаимодействия работает в фоновом режиме без вмешательства пользователя. Программный модуль принимает сообщения, организует очереди, определяет с помощью маршрутной таблицы получателя и отсылает сообщения. Для хранения сообщений и маршрутной таблицы используется база данных.

**ABSTRACT**

The work contains 38 pages, 23 figures, 9 tables, 10 sources.

Keywords: *training complex, module, networking.*

The subject area is a training complex for the operational personnel of the classification yard. The networking module runs in the background without any user intervention. The program module accepts messages, organizes queues, determines the recipient using the routing table and sends the messages. A database is used to store messages and a routing table.

**ОПРЕДЕЛЕНИЯ, ОБОЗНАЧЕНИЯ И СОКРАЩЕНИЯ**

ПО – программное обеспечение.

ПМ – программный модуль.

ИС – информационная система.

ДСПГ – дежурный по горке.

СЦБ – сигнализация, централизация и блокировка.

Оперативный персонал – персонал компании, находящийся на смене и выполняющий свои непосредственные обязанности.

Сортировочная горка – устройство на сортировочных железнодорожных станциях для ускоренного расформирования подвижных составов путем скатывания вагонов с уклона.

ТК (тренажерный комплекс) – система связанных автоматизированных рабочих мест, предназначенных для обучения и тестирования персонала в рамках предметной области.

ОПСГ – оперативный персонал сортировочной горки.

АРМ – автоматизированное рабочее место.

C# – объектно-ориентированный язык программирования высокого уровня.

MS – компания Microsoft.

ER (Entity Relationship) – модель «сущность-связь».

ПТЭ – правила технической эксплуатации железных дорог Российской Федерации.

Брокер сообщений – приложение, являющееся посредником между приложением-источником и приложением-приёмником, преобразуя сообщение по протоколу и передавая его.

**СОДЕРЖАНИЕ**

[Введение 6](#_Toc106539989)

[1 Аналитическое исследование 7](#_Toc106539990)

[1.1 Обзор предметной области 7](#_Toc106539991)

[1.3 Решение проблемы 10](#_Toc106539992)

[1.4 Постановка цели, задачи 11](#_Toc106539993)

[1.5 Вывод об аналитическом исследовании 12](#_Toc106539994)

[2 Проектирование информационной системы 13](#_Toc106539995)

[2.1 Моделирование бизнес-процессов информационной системы 13](#_Toc106539996)

[2.2 Описание бизнес-процессов до и после внедрения приложения 18](#_Toc106539997)

[2.3 Структура ИС и ее средства разработки 20](#_Toc106539998)

[2.4 Обзор и сравнение технологий 22](#_Toc106539999)

[2.5 Построение концептуальной модели базы данных 25](#_Toc106540000)

[2.6 Вывод о проектировании системы 28](#_Toc106540001)

[3 Реализация спроектированного модуля 30](#_Toc106540002)

[3.1 Процесс разработки, описание методов 30](#_Toc106540003)

[3.2 Интеграция базы данных в программу 31](#_Toc106540004)

[3.3 Тестирование приложения 35](#_Toc106540005)

[Заключение 36](#_Toc106540006)

[Список использованных источников 37](#_Toc106540007)

# **ВВЕДЕНИЕ**

Любая организация представляет собой систему. Каждая система имеет свою собственную структуру и сложность. Однако никакую систему невозможно представить без связей между её элементами. В век информатизации все данные организаций хранятся и реализуются с помощью информационных систем. Информационные системы в свою очередь состоят из компьютеров, компьютерных сетей и программ, связанных и сообщающихся между собой.

Тренажерные комплексы представляют собой совокупность автоматизированных рабочих мест. Для полноты информации о ситуации на станции, все изменения, сигналы и сообщения, появляющиеся в результате действий одного из операторов, должны быть получены и отображены на рабочих компьютерах других участников процесса.

Информационные технологии стремительно развиваются, не стоит на месте и прогресс в сетевом проектировании. Связь и взаимодействие в контексте компьютеров – сложный процесс, протоколы и инструменты работы с которым непрерывно совершенствуются. Поэтому зачастую подходы, используемые в организациях, устаревают и требуют замены на новые современные решения, обеспечивающие более высокий уровень отказоустойчивости, быстродействия и удобства дальнейшего расширения функционала.

В связи с этим требуется пересмотреть реализацию программного модуля сетевого взаимодействия тренажерного комплекса оперативного персонала сортировочной горки

В ходе выпускной квалификационной работы был разработан модернизированный модуль сетевого взаимодействия тренажерного комплекса оперативного персонала сортировочной горки на полигоне ОАО «РЖД». Также разработанный модуль возможно внедрить в другой тренажерный комплекс или другую информационную систему.

# **1 Аналитическое исследование**

## 1.1 Обзор предметной области

Предметной областью в данной выпускной квалификационной работе является тренажерный комплекс оперативного персонала сортировочной горки – это модель сортировочной горки с возможностью отработки основных навыков работы дежурного по станции и операторов сортировочной горки в учебном режиме. Тренажер является разработкой федерального государственного бюджетного образовательного учреждения высшего образования «Сибирский Государственный Университет Путей Сообщений» (ФГБОУ ВО «СГУПС»), заказчиком продукта является открытое акционерное общество «Российские Железные Дороги» (ОАО «РЖД»).

ТК ОПСГ разработан для операторов горочного комплекса с целью практической отработки навыков исполнения основных должностных обязательств [1].

Работа тренажерного комплекса включает следующие процессы:

1. моделирование реальной ситуации железнодорожной станции;
2. выполнение обучаемыми на автоматизированных рабочих местах надлежащих действий по разрешению ситуации. Действия выполняются согласно правилам технической эксплуатации железных дорог;
3. проверка инструктором корректности решений, принятых обучаемыми, а также соответствия их действий ПТЭ с дальнейшей оценкой и выводами по работе персонала сортировочной горки в моделируемой ситуации.

Так как в рамках данной работы планируется разработка ПО, рассмотрим составляющие существующей программно-аппаратной архитектуры ТК ОПСГ, она состоит из следующих программных модулей:

* ПМ «Сервер динамики»;
* ПМ «Сервер приложений»;
* Система компьютерной генерации изображений (СКГИ) – 3D модель;
* ПМ «Эмулятор СЦБ»;
* ПМ «Эмулятор АРМ ДСПГ» и «Эмулятор АРМ ОСГ»;
* ПМ «АРМ инструктора»;
* ПМ «Система обучения»;
* «Модуль обновления».

Схема структуры программно-аппаратного комплекса ТК ОПСГ показана на рисунке 1.1.

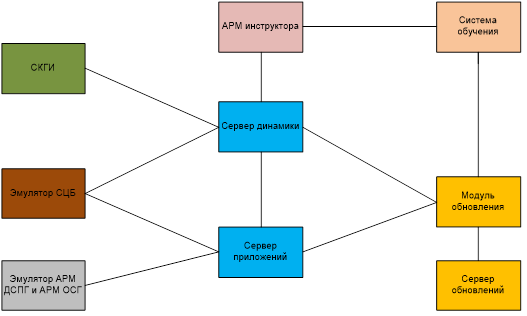


Рисунок 1.1 – Структура аппаратно-программного обеспечения тренажерного комплекса оперативного персонала сортировочной горки

На данный момент сетевое взаимодействие на тренажёре обеспечивает ПМ «Эмулятор СЦБ», рисунок 1.2. В его задачи входит:

* обеспечение протокола взаимодействия с пультами;
* обеспечение передачи сигналов о состоянии кнопок, коммутаторов управления стрелочными переводами и замедлителями программному модулю «Сервер приложений»;
* обеспечение передачи сигналов о занятости блок-участков для отображения на светосхеме пультов.

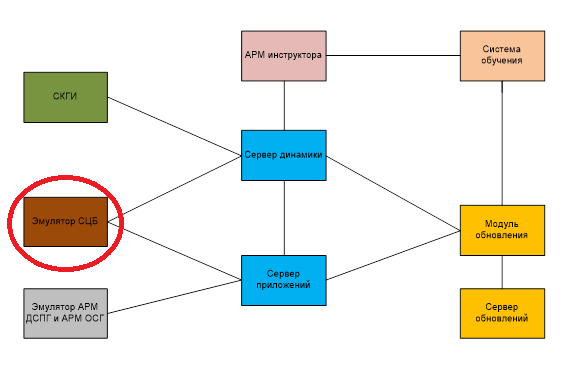


Рисунок 1.2 – Местонахождение модуля сетевого взаимодействия в программно-аппаратной структуре ТК ОПСГ

**1.2 Определение проблемы предметной области**

Рассмотрим подробнее принцип работы существующего сетевого взаимодействия эффективности коммуникации был разработан регламент переговоров, обеспечивающий четкость и понятность каждого сообщения или команды.

Рассмотрим процесс работы модуля:

* клиенты, которыми являются программные модули тренажерного комплекса, такие как: АРМ ДСПГ и ОСГ, сервер динамики, СКГИ и так далее отправляют в программу сообщения разного типа;
* все сообщения скапливаются в единой очереди;
* разбор очереди и обработка сообщений происходит по таймеру, раз в 4 миллисекунды;
* с помощью таблицы маршрутизации, в которой указаны отправители, типы сообщений и получатели, выполняется отсылка сообщений по назначению.

Основные проблемы модуля сетевого взаимодействия, следующие:

* трудность расширяемости, поддержки модуля;
* так как все сообщения скапливаются в единой очереди, возможны задержки в передачи сигналов, а также полный отказ в работе сетевой коммуникации, что приводит к остановке тренажера;
* с маршрутной таблицей и типами сообщений трудно работать ввиду отсутствия базы данных;
* присутствует низкая отказоустойчивость по причине отправки всех настроек единым для каждого компонента файлом непосредственно при загрузке всей системы;
* устаревший код.

## 1.3 Решение проблемы

Для устранения описанных проблем было принято решение модернизировать модуль сетевого взаимодействия с помощью внедрения в него базы данных и усовершенствования кода программы с использованием современных технологий разработки.

Брокер сообщений – программное приложение, используемое в качестве посредника между источником и приёмником, оно преобразует сообщение по протоколу передачи данных. Задачами брокера, помимо передачи и преобразования сообщения, также являются:

* проверка сообщения на ошибки;
* маршрутизация соответствующему приемнику(ам);
* сохранение сообщений в базе данных;
* разбиение сообщения на несколько маленьких, а затем агрегирование ответов приёмников и отправка результата источнику.

Брокер сообщений для повышения эффективности может реализовывать стратегии упорядоченной рассылки и определение приоритетности, балансировать нагрузку и прочее.

С внедрением усовершенствованного модуля, включающего в себя базу данных, решится ряд вышеперечисленных проблем, появится возможность исключить часто воспроизводимые ошибки или уменьшить вероятность их появления.

## 1.4 Постановка цели, задачи

В ходе выпускной квалификационной работы необходимо разработать улучшенный модуль сетевого взаимодействия ТК ОПСГ.

Разрабатываемый программный модуль должен соответствовать следующим требованиям:

* наличие базы данных;
* определение очереди для сообщения;
* передача сообщения в очередь;
* распределение сообщений по получателям с помощью маршрутной таблицы;
* обеспечение быстродействия и отсутствия задержек в передаче сообщений;
* ведение журнала необработанных сообщений.

Задачи, необходимые для достижения поставленной цели:

* провести анализ предметной области;
* изучить сетевую структуру тренажера;
* проанализировать требования к системе;
* спроектировать базу данных для информационной системы;
* рассмотреть различные реализации брокеров сообщений и выбрать наиболее подходящую для решения поставленных задач;
* разработать модуль сетевого взаимодействия с учетом всех требований и особенностей предметной области;
* проверить соответствие разработанного ПО всем требованиям;
* протестировать разработанный модуль при необходимости исправить его недостатки.

## 1.5 Вывод об аналитическом исследовании

По результатам проведения анализа предметной области был выявлен следующий список проблем существующего модуля сетевого взаимодействия, возникающий при обучении операторов железнодорожной станции на ТК ОПСГ:

* трудность расширяемости, поддержки модуля;
* низкий уровень отказоустойчивости ввиду неэффективной организации сообщений в очереди;
* низкая скорость передачи данных при высокой нагрузке системы;
* отсутствие БД, что затрудняет работу с данными;
* необходимость обновления кода.

Для решения вышеописанных проблем требуется разработать улучшенный модуль сетевого взаимодействия для тренажерного комплекса оперативного персонала сортировочной горки.

# **2 Проектирование информационной системы**

## 2.1 Моделирование бизнес-процессов информационной системы

Рассмотрим существующие методологии моделирования бизнес-процессов, необходимые для проектирования информационных систем.

*Методология IDEF0.*

Неудивительным является то, что внешне диаграммы, построенные с помощью методологии IDEF0, несколько похожи на диаграммы, реализованные с помощью методологии SADT, ведь методология IDEF0 является подмножеством методологии SADT.

IDEF0 также, как и SADT, позволяет простроить функциональную модель описания бизнес-процессов предметной области. Различие заключается в том, что IDEF0 более детально останавливается на взаимодействии объектов между друг другом, нежели на последовательности действий отдельных объектов.

Моделирование происходит по принципу «черного ящика» – известно состояние объекта до какого-либо процесса и после него, но сам процесс скрыт от глаз читателя диаграммы. Более подробно процессы раскрываются в диаграммах декомпозиции.

На диаграмме IDEF0, в отличие от диаграммы потоков данных DFD, отображаются сигналы управления. Описанная методология используется для моделирования всех процессов [2].

*Диаграмма потоков данных DFD.*

DFD в дословном переводе с английского языка является диаграммой потоков данных. DFD акцентирует внимание на передаче данных между процессами, способами ее обработки и хранения. Диаграммы поток данных активно используются при разработке программных продуктов. Еще одной сферой использования данной нотации является анализ системы в рамках документооборота. Методология наглядно представляет:

* места хранения данных;
* процесс обмена документации;
* ошибки организации бизнес-процессов и прочее.

DFD не описывает бизнес-процесс, так как не учитывает времени выполнения этого процесса, а также условий, при которых процесс может пойти по альтернативному «сценарию»[3].

*Методология UML.*

Универсальный язык моделирования (UML) является одной из самых популярных методологий для графического описания бизнес-процесс. Большинство разработчиков программного обеспечения используют его при проектировании и сопровождении своих продуктов.

UML состоит из интегрированного набора диаграмм, разработанных для помощи разработчикам систем и программного обеспечении в определении, визуализации, конструировании и документировании результатов разработки программного обеспечения, а также для бизнес-моделирования[4].

*Методология ARIS.*

Методология ARIS представляет собой набор модулей, использующийся для рассмотрения бизнес-процессов сразу с нескольких точек зрения:

* организационной;
* функциональной;
* обрабатываемых данных;
* структуры бизнес-процессов;
* продуктов и услуг[5].

Из рассмотренных методологий для моделирования бизнес-процессов приложения автоматизированного контроля регламента переговоров была выбрана методология UML, так как диаграммы в данной методологии предоставляют всю необходимую информацию о бизнес-процессах предметной области, достаточную для разработки информационной системы.

Для моделирования бизнес-процессов в UML необходимо построить следующие диаграммы:

* диаграмму вариантов использования системы;
* диаграмму последовательности процесса автоматизированного контроля регламента переговоров;
* диаграмму активностей автоматизированного контроля регламента переговоров;
* диаграмму состояний информационной системы;
* диаграмму классов предметной области.

На рисунке 2.1 представлена диаграмма вариантов использования системы автоматизированного контроля регламента переговоров.

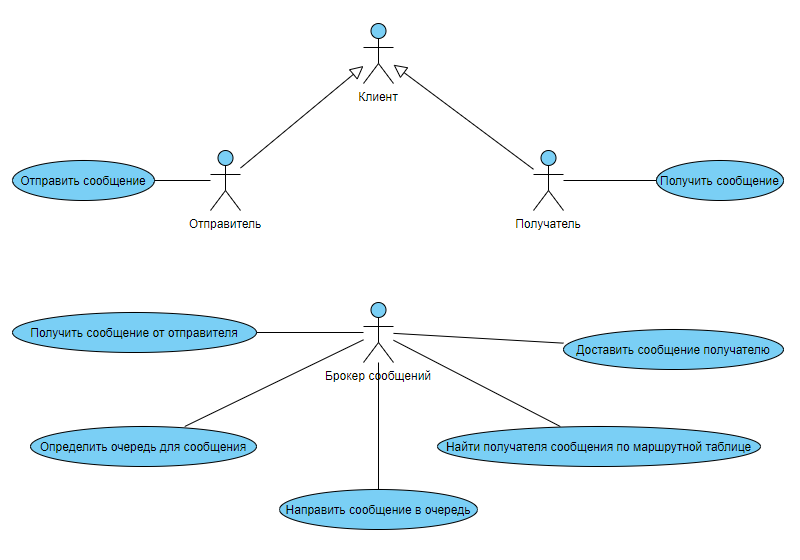


Рисунок 2.1 – Диаграмма вариантов использования приложения автоматизированного контроля регламента переговоров

Процесс предполагает участия трёх актёров: отправителя, получателя и брокера сообщений. Выполнение основных функций – ответственность брокера сообщений. От клиентов не требуется никаких дополнительных действий, кроме отправки сообщений.

На рисунке 2.2 представлена диаграмма последовательности процесса сетевого взаимодействия ТК ОПСГ. Данная диаграмма более подробно раскрывает диаграмму вариантов использования, показанную на рисунке 2.1. С помощью диаграммы последовательности можно подробнее рассмотреть причинно-следственные и временные связи между актерами и действиями, совершаемыми ими.

Рисунок 2.3 изображает диаграмму состояний информационной системы. Данная диаграмма показывает то, как состояния объекта «Сообщение» меняется в ходе процесса.

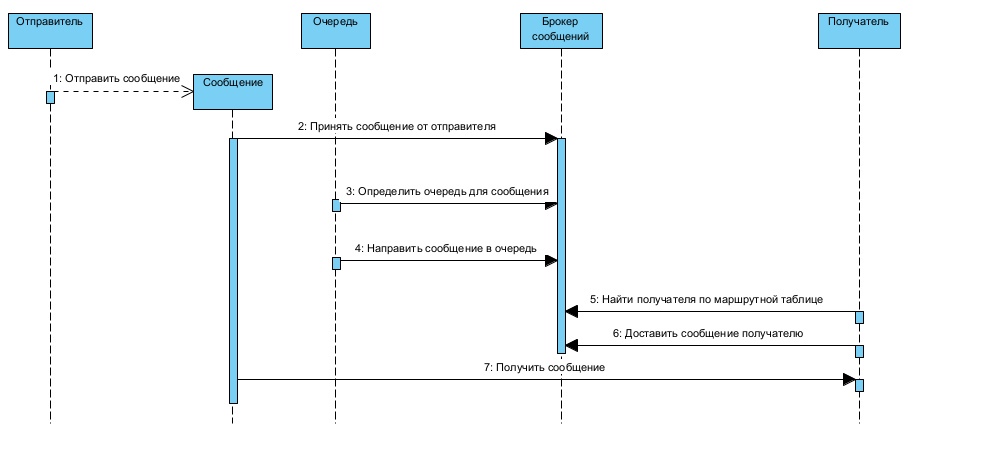


Рисунок 2.2 – Диаграмма последовательности процесса контроля регламента

На рисунке 2.4 представлена диаграмма активностей процесса автоматизированного контроля регламента переговоров. Эта диаграмма отличается использованием так называемых «дорожек», на которых располагаются действия участвующих в процессе сущностей.

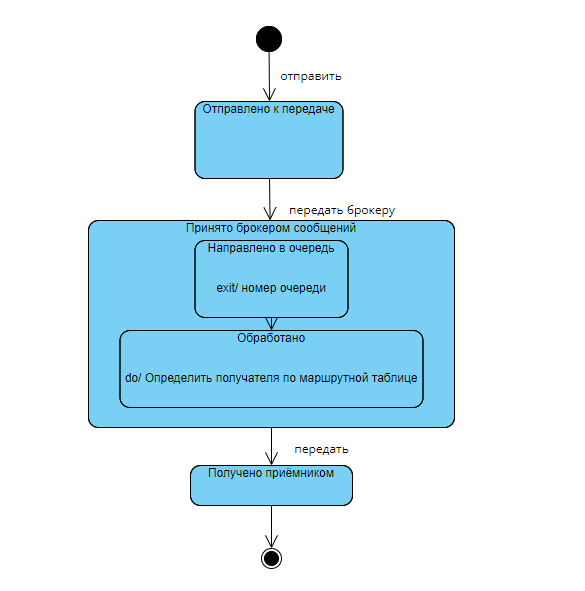


Рисунок 2.3 – Диаграмма состояний объекта «Сообщение»

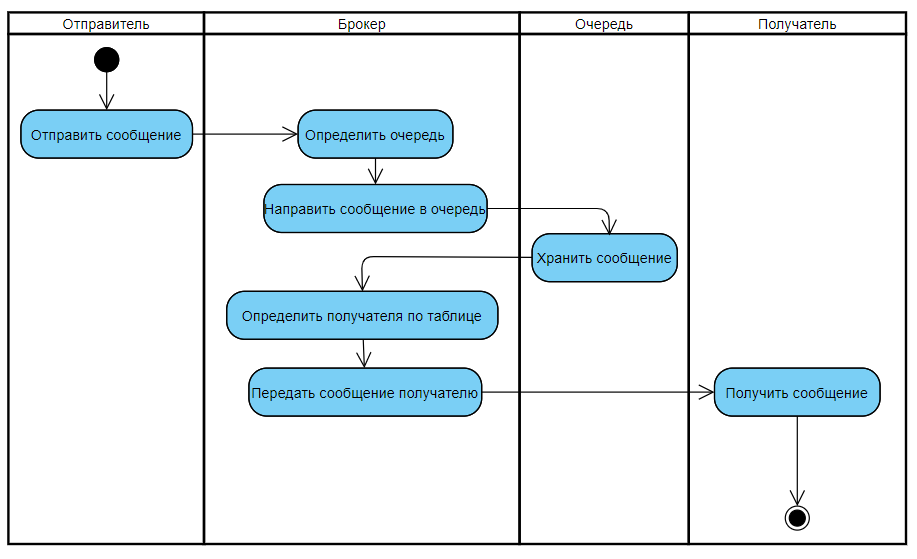


Рисунок 2.4 – Диаграмма активностей процесса передачи сообщений

На рисунке 2.5 изображена диаграмма классов предметной области «Система автоматизированного контроля регламента переговоров».

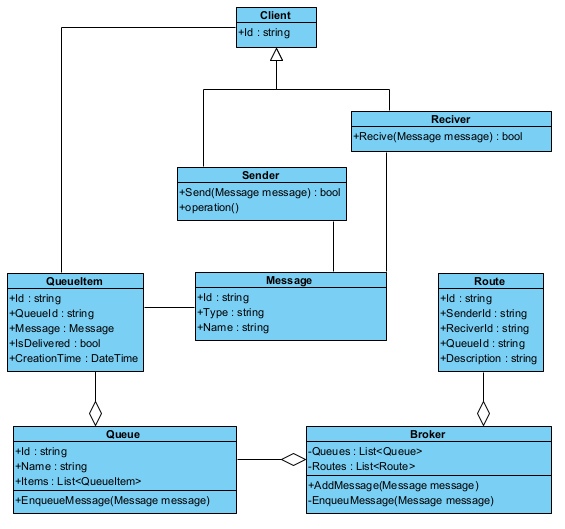


Рисунок 2.5 – Диаграмма классов предметной области

## 2.2 Описание бизнес-процессов до и после внедрения приложения

Рассмотрим, как выглядит процесс передачи сообщений и сигналов на данный момент. Схема процесса показана на рисунке 2.6. До внедрения разрабатываемого модуля отсутствовала база данных ИС, а также возникал ряд проблем при высокой нагрузке системы из-за низкой эффективности кода.

Теперь рассмотрим процесс сетевого взаимодействия после модернизации модуля с помощью внедрения брокера сообщений и БД, он претерпит некоторые изменения. Схема нового процесса показана на рисунке 2.7.

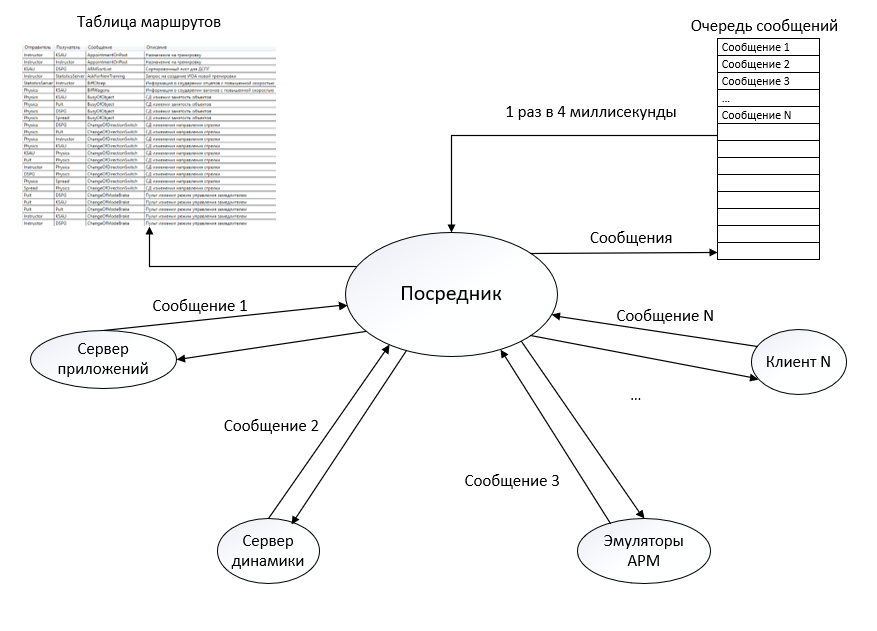


Рисунок 2.6 – Схема сетевого взаимодействия до внедрения модуля

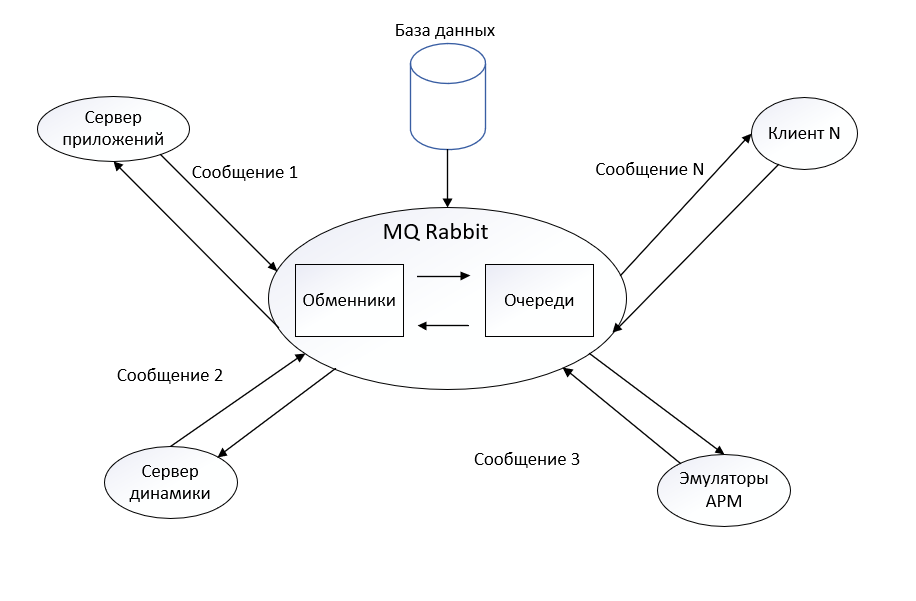


Рисунок 2.7 – Схема сетевого взаимодействия после внедрения модуля

Внедрение нового разрабатываемого модуля обеспечит решение существующих проблем, что в свою очередь повысит качество обучения оперативного персонала сортировочной горки.

## 2.3 Структура ИС и ее средства разработки

Модуль сетевого взаимодействия разрабатывается для ТК ОПСГ, но также может использован для других комплексов, где важно сетевое взаимодействие. Рассмотрим список требований к функционалу модуля:

* расширяемость;
* удобство поддержки;
* современный подход к разработке;
* отказоустойчивость;
* наличие базы данных.

Таким образом, система включает в себя следующие части:

* база данных для хранения данных о сообщениях, клиентах и маршрутах;
* программа-посредник, реализующая отправку сообщений.

Техническое задание предполагает разработку приложения автоматизированного контроля регламента переговоров тренажерного комплекса оперативного персонала сортировочной горки на языке программирования C#.

На данный момент C# является довольно распространенным языком программирования, поэтому существует множество сред (IDE) для разработки на нём. Для выбора наиболее подходящей IDE было произведено сравнение наиболее популярных вариантов, был проведен анализ их достоинств и недостатков и соответствия требованиям.

Рассматриваемые альтернативы: Eclipse, Visual Studio, Project Rider, Visual Studio Code, MonoDevelop[6]. Для сравнения были выбраны следующие критерии: наличие бесплатной версии, простота в освоении, поддержка выбранного языка программирования, широкая функциональность, а также наличие инструментов автоматизации, компилятора, интерпретатора Соответствие альтернатив требованиям наглядно показано в таблице 2.1.

Таблица 2.1 – Сравнение альтернативных сред разработки

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Критерий | Eclipse | Visual Studio | Project Rider | VS Code | MonoDevelop |
| Наличие бесплатной версии | + | + | - | + | + |
| Широкая функциональность | + | + | + | - | - |
| Простота в освоении | - | + | + | + | + |
| Поддержка выбранного языка программирования | + | + | + | + | + |
| Наличие инструментов автоматизации, компилятора, интерпретатора | + | + | + | + | + |

Все описанные среды разработки поддерживают выбранный язык программирования, поддерживаются их разработчиками и обладают качественными компилятором, интерпретатором, отладчиком и инструментами автоматизации.

Для реализации технической составляющей выпускной квалификационной работы IDE должна обладать большой функциональностью. Поэтому среды разработки Visual Studio Code, MonoDevelop не подходят, т.к. обладают недостаточным функционалом.

Среды разработки Visual Studio и Eclipse являются сложными в освоении, что может замедлить разработку приложений.

Project Rider является платной средой разработки, что исключает данную IDE из списка рассматриваемых сред разработки.

В результате выбора среды разработки:

* Visual Studio Code и MonoDevelop не подходят, т.к. обладают недостаточным функционалом;
* Project Rider является платным продуктом;
* Eclipse является сложной в освоении.

Самой подходящей средой разработки для написания технической составляющей выпускной квалификационной работы является Visual Studio.Выбранная среда разработки:

* поддерживает разработку на выбранном для работы языке программирования (C#);
* адекватна для решения задач, предусмотренных технической частью выпускной квалификационной работы, в полной мере;
* имеет качественный встроенный компилятор, интерпретатор, отладчик и инструменты автоматизации;
* современна, поддерживается разработчиками;
* имеет обширную документацию;
* удобна в использовании;
* имеет бесплатную версию;
* предусмотрена программой обучения исполнителя, что нивелирует сложность.

## 2.4 Обзор и сравнение технологий

Так как было принято решение использовать готовый брокер сообщений, следующим шагом необходимо рассмотреть популярные варианты и выбрать наиболее подходящий для реализации поставленных нами задач.

*Apache Kafka.*

Kafka – это брокер сообщений с открытым исходным кодом, который был разработан и сейчас поддерживается в первую очередь фондом Apache Software Foundation при содействии сообщества разработчиков приложений с открытым кодом. Основные характеристики:

* акцент на потоковом контенте, работа с большими потоками данных;
* основные возможности: обеспечение сохранности сообщений и их многократная повторная обработка;
* хостинг на месте и поддержка сторонних модулей.

Kafka обеспечивает оптимизированную потоковую обработку событий, при этом связь между потребителями реализуется по модели «Публикации — подписки». События могут быть разбиты на топики — эта возможность позволяет лучше организовать схемы взаимодействия распределенного приложения. Кроме того, события делятся между несколькими серверами в пределах одного кластера – благодаря этому достигается высокая устойчивость и производительность системы[7].

Kafka заточен на обработку потоков данных, поэтому получается высокопроизводительная система, которая выполняет сложную обработку больших объемов данных. Функционал маршрутизации этих потоков относительно ограничен по сравнению с другими брокерами сообщений.

*RabbitMQ.*

RabbitMQ – это еще один брокер сообщений с открытым кодом.

Основные характеристики:

* акцент на обмене сообщениями с возможностью поддержки больших потоков данных;
* основная особенность – расширенный функционал маршрутизации;
* хостинг на месте и поддержка сторонних модулей.

RabbitMQ также использует модель «Публикации – подписки», отправляя объекты сообщений в двоичной форме в различные именованные очереди, которые могут динамически создаваться и уничтожаться. Он может работать как автономно, так и в составе кластера, позволяет наращивать достаточную мощность для достижения любой степени избыточности или безопасности данных[8]*.*

Данный брокер способен справляться практически с любыми нагрузками и может эффективно масштабироваться вместе с приложением по мере роста базы пользователей.

*SQS/SNS.*

SNS и SQS представляют собой примеры двух разных подходов к распределенному обмену сообщениями. SNS в значительной степени ориентирован на доставку сообщений. С помощью модели «Публикации – подписки» он позволяет быстро передавать сообщения множеству клиентов, например мобильным устройствам, конечным точкам HTTPS или другим сервисам. SQS, напротив, приоритетом ставит успешную доставку и обработку сообщений отдельными клиентами.

Основные характеристики:

* возможность передавать широковещательные сообщения и работать по модели «Публикации – подписки»;
* быстрая настройка;
* отсутствие хостинга.

SNS транслирует одно и то же сообщение множеству получателей, а SQS распределяет организованные в очередь сообщения среди отдельных подписчиков.

В SNS применяется режим push-уведомлений, который позволяет автоматизировать ответы, получаемые специальными мобильными приложениями, совместимыми с данной системой. SQS больше ориентирован на механизм опроса с поддержкой некоторых дополнительных функций, управляемых событиями[9]. Сравнение альтернатив продемонстрированно в таблице 2.2.

Таблица 2.2 – Сравнение брокеров сообщений

|  |  |  |  |
| --- | --- | --- | --- |
| Критерий | Apache Kafka | MQ Rabbit | SNS/SQS |
| Локальный хостинг | + | + | - |
| Обеспечение сохранности сообщений | + | - | + |
| Поддержка сложной маршрутизации | - | + | + |
| Работа с большими потоками данных | + | + | - |
| Быстрая установка | - | - | + |

В результате анализа рассматриваемых альтернатив, их достоинств и недостатков, был выбран брокер MQ Rabbit, обеспечивающий:

* обмен сообщениями с возможностью поддержки больших потоков данных;
* расширенный функционал маршрутизации;
* хостинг на месте и поддержку сторонних модулей.

Главное достоинство этого брокера по сравнению с остальными – хорошая расширяемость, так как этот критерий является важным для достижения нашей цели.

## 2.5 Построение концептуальной модели базы данных

ER-модель – модель данных, позволяющая описывать концептуальные схемы предметной области. ER-модель используется при высокоуровневом (концептуальном) проектировании баз данных. С её помощью можно выделить ключевые сущности и обозначить связи, которые могут устанавливаться между этими сущностями.

Во время проектирования баз данных происходит преобразование ER-модели в конкретную схему базы данных на основе выбранной модели данных. ER-модель представляет собой формальную конструкцию, которая сама по себе не предписывает никаких графических средств её визуализации.

В соответствии с анализом предметной области и принципом работы выбранного брокера сообщений можно спроектировать следующие выделенные сущности: «Клиент», «Сообщение», «Очередь», «Список очередей», «Маршрутная таблица», «Очередь необработанных сообщений». В таблицах 2.3 – 2.8 представлено описание каждой сущности.

Таблица 2.3 – Описание сущности «Сообщение»

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Поле | Тип данных | Обязательное | PK | Описание |
| Id | string | + | + | Уникальный идентификатор |
| Type | string | + | - | Тип сообщения |
| MessageName | string | + | - | Наименование сообщения |

Таблица 2.4 – Описание сущности «Клиент»

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Поле | Тип данных | Обязательное | PK | Описание |
| Id | string | + | + | Уникальный идентификатор |
| Name | string | + | - | Наименование клиента |

Таблица 2.5 – Описание сущности «Очередь»

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Поле | Тип данных | Обязательное | PK | Описание |
| Id | string | + | + | Уникальный идентификатор |
| Name | string | + | - | Наименование очереди |

Таблица 2.6 – Описание сущности «Список очередей»

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Поле | Тип данных | Обязательное | PK | Описание |
| Id | string | + | + | Уникальный идентификатор |
| QueueNamesId | string | + | - | Ссылка на id очереди, в которую помещается сообщение |
| IsDelivered | bool | + | - | Флаг об успешной доставке |
| Message | string | + | - | Ссылка id сообщения |
| CreationTime | datetime | + | - | Время добавления сообщения в очередь |

Таблица 2.7 – Описание сущности «Маршрутная таблица»

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Поле | Тип данных | Обязательное | PK | Описание |
| Id | string | + | + | Уникальный идентификатор |
| SenderId | string | + | - | Ссылка на id клиента, являющегося отправителем |
| RecieverId | string | + | - | Ссылка на id клиента, являющегося получателем |
| QueueId | string | + | - | Ссылка id очереди |
| Description | string | - | - | Описание операции |

Таблица 2.8 – Описание сущности «Очередь необработанных сообщений»

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Поле | Тип данных | Обязательное | PK | Описание |
| Id | string | + | + | Уникальный идентификатор |
| ErrorReason | string | + | - | Причина по которой не было доставлено сообщение |
| AttemptTime | datetime | + | - | Время попытки отправки сообщения |
| QueueId | string | + | - | Ссылка id очереди |

По таблицам, описывающим сущности, была составлена ER-диаграмма, графически отображающая их связи, рисунок 2.8.

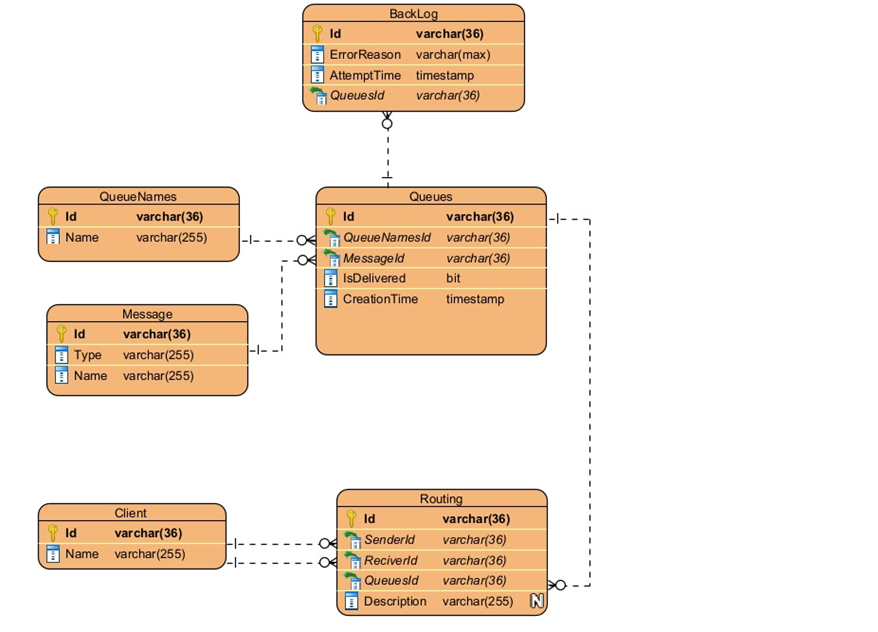


Рисунок 2.8 – ER-диаграмма базы данных модуля сетевого взаимодействия

## 2.6 Вывод о проектировании системы

В ходе проектирования модуля сетевого взаимодействия ТК ОПСГ было выполнено следующее:

1. были смоделированы бизнес-процессы системы в нотации UML, а именно построены следующие диаграммы;
2. диаграмма вариантов использования системы;
3. диаграмма последовательности процесса взаимодействия брокера и клиентов;
4. диаграмма активностей процесса передачи сообщения;
5. диаграмма состояний объекта «Сообщение»;
6. диаграмма классов предметной области;
7. описаны бизнес-процессы до и после внедрения системы;
8. описаны структура ИС и средства ее разработки;
   1. модули, входящие в приложение системы автоматизированного контроля регламента переговоров ТК ОПСГ;
   2. приведены язык программирования, на котором необходимо разработать приложение, и протоколы, по которым должно происходить подключение к серверу разработчиков и загрузки файлов, а также для передачи файлов между компьютерами тренажерного комплекса по локальной сети;
9. проведен обзор технологий реализации сетевого взаимодействия;
10. спроектирована база данных для рассматриваемой информационной системы.

# **3 Реализация спроектированного модуля**

## 3.1 Процесс разработки, описание методов

Перейдем непосредственно к разработке системы. Методы, обеспечивающие работу модуля, показаны в таблице 3.1.

Таблица 3.1 – Описание методов модуля

|  |  |  |  |
| --- | --- | --- | --- |
| Название метода | Входные параметры | Выходные параметры | Описание метода |
| Send | Сообщение класса Message | Булевая переменная | Метод реализует отправку сообщений от клиента-отправителя брокеру сообщений |
| Receive | Сообщение класса Message | Булевая переменная | Метод реализует отправку сообщения брокером получателю |
| AddMessage | Сообщение класса Message | void | Метод получения нового сообщения брокером |
| EnqueueMessage | Сообщение класса Message | void | Метод добавления нового сообщения брокером в какую-либо из очередей. Определение очереди ведется по маршрутной таблице. Также метод определяет приоритет сообщений |

## 3.2 Интеграция базы данных в программу

Для работы с базой данных будем использовать Entity Framework. Для начала необходимо создать сущности, отображающие таблицы БД. Для начала была создана базовая сущность EntityBase, от которой будут наследоваться все остальные. Код класса на рисунке 3.1.

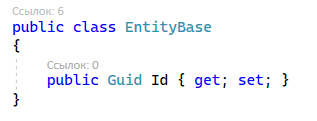


Рисунок 3.1 – Класс EntityBase

Далее были добавлены все необходимые сущности, их код на рисунках 3.2 – 3.7.

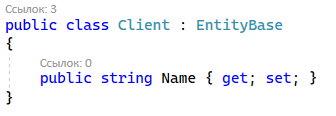


Рисунок 3.2 – Класс Client

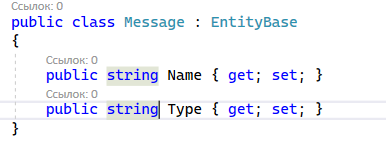


Рисунок 3.3 – Класс Message

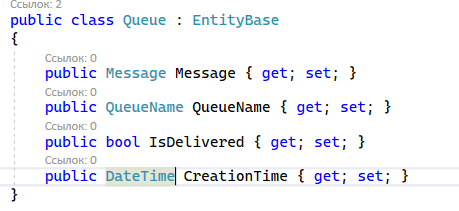


Рисунок 3.4 – Класс Queue

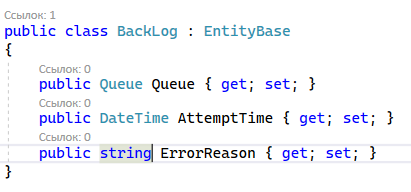


Рисунок 3.5 – Класс BackLog

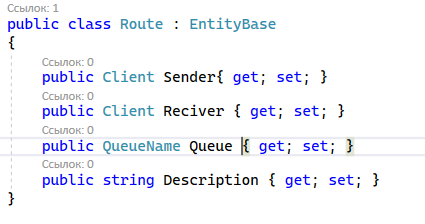


Рисунок 3.6 – Класс Route

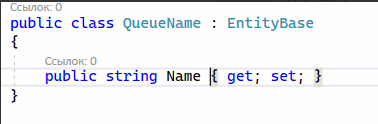


Рисунок 3.7 – Класс QueueName

Далее необходимо создать контекст базы данных, рисунок 3.8. Это нужно для того, чтобы обеспечить работу с БД.

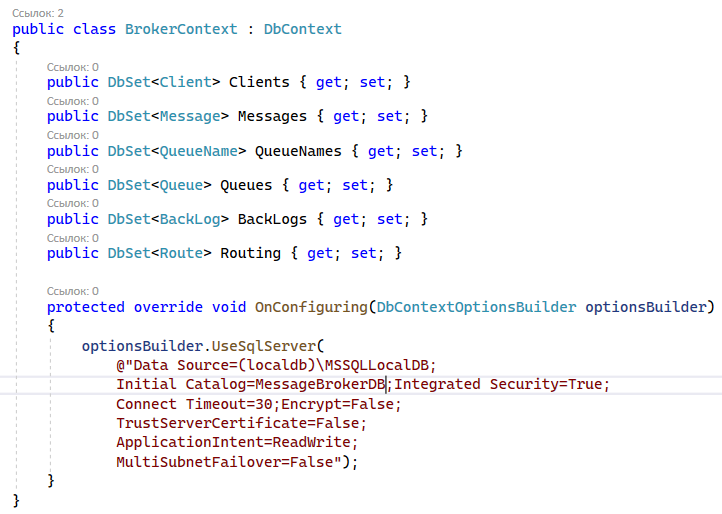


Рисунок 3.8 – Класс BrokerContext

Для изменения версии базы данных безопаснее всего использовать механизм миграций. С помощью него ведется журнал изменений, используя который есть возможно быстро перемещаться между версиями базы данных. Для создания миграций используется инструмент EF Core Tools, позволяющий автоматизировать процесс создания скриптов для обновления структуры данных. Этот инструмент самостоятельно определяет разницу между текущей версией БД и той, что соответствует коду, и генерирует скрипт для устранения этой разницы.

Чтобы создать новую миграцию, нужно использовать команду *Add-Migration*. Команда *Update-Database* применяется для использования миграции[10]. Скрипт понижения версии – на рисунке 3.9. Скрипт повышения версии БД, то есть применения новой миграции, показан на рисунке 3.10.



Рисунок 3.9 – Снижение версии БД



Рисунок 3.10 – Применение миграции БД

## 3.3 Тестирование приложения

Так как приложение работает в фоновом режиме и участие пользователя не требуется, его графический интерфейс отсутствует. Следовательно, для тестирования работы системы проследим за движением посланного сообщения с помощью вывода его состояний на консоль. Вывод будет представлен в виде логов.

Для примера будет послано сообщение типа TrainingCreated, название сообщения – Создание тренировки. Отправителем сообщения будет Сервер приложений, получателем – АРМ инструктора. После получения брокером сообщения от отправителя, сообщение помещается в очередь под названием, одноимённым с получателем.

Выводы консоли, показывающие путь сообщения, отображены на рисунках 3.11 – 3.13.

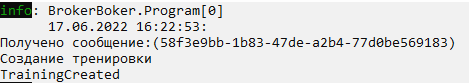


Рисунок 3.11 – Сообщение получено брокером



Рисунок 3.12 – Сообщение помещено в очередь

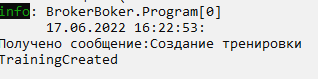


Рисунок 3.13 – Сообщение доставлено получателю

# **ЗАКЛЮЧЕНИЕ**

В рамках выпускной квалификационной работы было разработано приложение системы автоматизированного контроля регламента для ТК ОПСГ. Для достижения поставленной цели были выполнены следующие задачи:

* проведен анализ предметной области;
* изучена сетевая структура тренажера;
* проанализированы требования к системе;
* спроектирована база данных для информационной системы;
* рассмотрены различные реализации брокеров сообщений и выбрана наиболее подходящую для решения поставленных задач;
* разработан модуль сетевого взаимодействия с учетом всех требований и особенностей предметной области;
* проверено соответствие разработанного ПО всем требованиям;
* разработанный модуль протестирован.

После этого модуль будет в полной мере готова к внедрению в ТК ОПСГ.

# **СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ**

1. Факультет «Бизнес-информатика» - Абитуриенту / Официальный сайт ВУЗа «СГУПС» URL: https://www.stu.ru/education/index.php?page=674 (дата обращения: 11.04.2022).
2. Методология IDEF0 / Электронный образовательный ресурс ITteach.RU; [Электронный ресурс]. – URL: https://itteach.ru/bpwin/metodologiya-idef0 (дата обращения 12.04.2022).
3. Что такое диаграмма DFD и как ее создать? / Сервис для интеллектуального построения диаграмм Lucidchart; [Электронный ресурс]. – URL: https://www.lucidchart.com/pages/ru/диаграмма-dfd (дата обращения 12.04.2022).
4. Что такое унифицированный язык моделирования? / Сервис для интеллектуального построения диаграмм Lucidchart; [Электронный ресурс]. – URL: https://www.lucidchart.com/pages/ru/uml (дата обращения 12.04.2022).
5. Методология ARIS / Сайт группы компаний Концерн R-Про; [Электронный ресурс]. – URL: https://www.r-p-c.ru/page/metodologiya-aris.html (дата обращения 12.04.2022).
6. Орлов Е. Лучшие IDE для C#-разработчика / Е. Орлов – сайт международного IT-колледжа DevEducation; [Электронный ресурс]. – URL: https://spb.deveducation.com/blog/luchshie-ide-dlya-c-razrabotchika/ (дата обращения 05.05.2022).
7. Kafka, RabbitMQ или AWS SNS/SQS: какой брокер выбрать? / Сайт Habr. [Электронный ресурс]. – URL: https://habr.com/ru/post/573358/ (дата обращения 15.05.2022).
8. Что такое Apache Kafka: как устроен и работает брокер сообщений. / Сайт сообщества Selectel. [Электронный ресурс]. – URL: https://selectel.ru/blog/apache-kafka/ (дата обращения 20.05.2022).
9. RabbitMQ – самый распространеннный брокер сообщений с открытым исходным кодом. / Официальный сайт RabbitMq. [Электронный ресурс]. – URL: https://www.rabbitmq.com/ (дата обращения 24.05.2022).
10. Документация по Entity Framework. / Официальный сайт Microsoft Docs. [Электронный ресурс]. – URL: https://docs.microsoft.com/ru-ru/ef/ (дата обращения 30.05.2022).