Министерство науки и высшего образования Российской Федерации

ФГБОУ ВО «АЛТАЙСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ»

Институт цифровых технологий, электроники и физики

Кафедра вычислительной техники и электроники (ВТиЭ)

**Отчёт по производственной эксплуатационной практике**

Выполнил студент 595 гр.:

\_\_\_\_\_\_\_\_\_\_\_\_\_ Лаптев А.В.

Проверил: проф. д.т.н. каф. ВТиЭ

\_\_\_\_\_\_\_\_\_ Белозерских В.В.

  Оценка \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

  «\_\_\_\_\_\_» \_\_\_\_\_\_\_\_\_\_\_2022г.

Барнаул 2022

**Содержание**

[**1.** **ВВЕДЕНИЕ** 3](#_Toc107134385)

[**1.1.** **Общие цели производственной эксплуатационной практики** 3](#_Toc107134386)

[**1.2.** **Постановка задачи производственной эксплуатационной практики** 3](#_Toc107134387)

[**2.** **ОБЩИЕ СВЕДЕНИЯ** 4](#_Toc107134388)

[**2.1.** **Постановка задачи** 4](#_Toc107134389)

[**2.2.** **Описание выполненных работ** 4](#_Toc107134390)

[**3.** **ЗАКЛЮЧЕНИЕ** 7](#_Toc107134391)

1. **ВВЕДЕНИЕ**
   1. **Общие цели производственной эксплуатационной практики**

Целью проведения практики является приобретение студентами универсальных и профессиональных навыков, практического опыта, закрепление, систематизация и расширение теоретических знаний по использованию, администрированию, настройке и наладке программно-аппаратных комплексов. Производственная эксплуатационная практика ориентирована на формирование у бакалавров универсальных и профессиональных компетенций.

* 1. **Постановка задачи производственной эксплуатационной практики**

В ходе выполнения практики обучающийся приобретает навыки научно-исследовательской деятельности и решает одну или несколько профессиональных задач:

* изучение научно-технической информации, отечественного и зарубежного опыта по тематике исследования;
* математическое моделирование процессов и объектов на базе стандартных пакетов автоматизированного проектирования и исследований;
* проведение экспериментов по заданной методике и анализ результатов;
* проведение измерений и наблюдений, составление описания проводимых исследований, подготовка данных для составления обзоров, отчетов и научных публикаций;
* составление отчета по выполненному заданию, участие во внедрении результатов исследований и разработок.

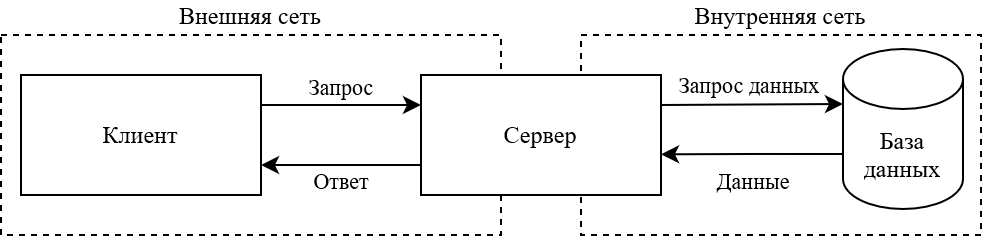
**Время прохождения практики:** 16.05.–11.06.2022.

1. **ОБЩИЕ СВЕДЕНИЯ**

Производственная эксплуатационная практика проходила в АО «БСКБ «Восток». Руководителям практики являлись: от кафедры – Белозерских Василий Вениаминович, от предприятия – Коршунов Дмитрий Сергеевич.

* 1. **Постановка задачи**

Создать простую тестовую базу данных на PostgreSQL из двух связанных таблиц. На C++ реализовать сервер и клиент (для клиента использовать Qt). Клиент должен получать данные из БД от сервера по стандарту REST API.  Для реализации API найти библиотеки с открытым исходным кодом и активной поддержкой со стороны сообщества разработчиков. Результаты работы продемонстрировать в виде рабочей информационной системы и отчёта руководителю практики от предприятия.



*Рис. 1. Схема обмена данными в информационной системе.*

В мои задачи, в рамках данного проекта, входили реализация клиента с использованием фреймворка Qt и налаживание его взаимодействия с сервером по внешней сети. Реализация сервера и настройка БД осуществлялись Осиповым А.В.

* 1. **Описание выполненных работ**

В качестве операционной системы, на которой работает клиент, была установлена отечественная операционная система AstraLinux «Орёл», один из дистрибутивов GNU/Linux. Для разработки клиентского приложения использовалась интегрированная среда разработки (IDE) Qt Creator, которая содержит в своем составе текстовый редактор для написания кода и компилятор gcc, а, также, поддерживает все необходимые модули для работы с фреймворком Qt на C++.

Суть работы клиента заключается в отправке запросов серверу на получение данных от сервера или отправку данных серверу. Также, клиент должен получать ответ от сервера, который содержит либо запрашиваемые данные (если был отправлен GET-запрос), либо отчет о статусе транзакции (завершилась успешно или с ошибками – если был отправлен POST-запрос). В качестве протокола обмена данных между клиентом и сервером используется HTTP. Передаваемая по этому протоколу информация хранится в текстовом формате JSON, а также в URI-параметрах запроса (для запросов на получение данных).

Для реализации всего необходимо для работы клиента функционала (создание и отправка HTTP-запросов, чтение/запись JSON, графический интерфейс) использовались только встроенные модули Qt. В частности, интерфейс был разработан в Qt Designer, для того, чтобы не загромождать остальное решение; HTTP-запросы, с использованием методов GET (получение данных с сервера) и POST (отправка данных на сервер), были реализованы с применением модуля QtNetwork и классов QNetworkAccessManager, QNerworkReply, QNetworkRequest, входящих в этот модуль; чтение и разбор (парсинг) JSON, а также запись JSON осуществлялись с помощью модуля QtCore и классов QJsonDocument, QJsonArray, QJsonObject и QJsonValue, которые входят в данный модуль.

Созданный клиент обладает следующими возможностями:

* получение информации о группах студентов и списка студентов в каждой из групп или для всех групп сразу, с использованием GET-запроса;
* получение записей о посещаемости для студентов выбранной группы/всех групп, с использованием GET-запроса;
* возможность добавлять, изменять и удалять данные о студентах в базе данных посредством отправки POST-запросов на сервер, а, также, добавлять новые отметки о посещаемости для конкретных даты и времени занятия и вносить изменения в посещаемость конкретного студента, путем проставления отметки в чекбоксе;
* возможность выводить вышеперечисленные данные в удобном для просмотра виде.

Работа с приложением осуществляется следующим образом:

После запуска приложения перед пользователем появляется главное окно, которое содержит три кнопки: «Просмотр посещаемости», «Список студентов», «Отметка посещаемости». После нажатия на одну из кнопок будет открываться одно из второстепенных окон (при этом главное окно скроется).

В окне для просмотра посещаемости предоставляется возможность выбора номера группы из выпадающего списка. После нажатия кнопки «Выбрать» в таблице будет выведен список группы, в котором для каждого студента будет указана дата и время занятия и отметка о посещаемости (присутствовал/отсутствовал). После нажатия кнопки «Назад» пользователь вновь перемещается в главное окно (окно просмотра посещаемости скрывается).

В окне для отметки посещаемости пользователь, помимо группы, указывает дату и время занятия, и, после подтверждения выбора (нажатия кнопки «Выбрать»), в таблицу выводится список студентов и столбец чекбоксов, в котором можно проставить флажок для тех студентов, которые присутствовали на занятии. По нажатию на кнопку «Сохранить» данные будут отправлены на сервер и занесены в БД. Кнопка «Назад» также возвращает пользователя в главное окно.

В окне просмотра и редактирования списка студентов пользователь также выбирает группу и подтверждает свой выбор. После этого, в таблицу выводятся данные о студентах и группе, в которой они обучаются. В правой части окна расположены три кнопки: «Добавить», «Изменить», «Удалить». После нажатия на кнопку «Добавить» появляется окно с формой, заполнив которую можно добавить студента в указанную группу. Для этого надо нажать кнопку «Добавить» в окне формы, для отмены операции добавления можно нажать «Отмена». Для удаления данных о студенте требуется выделить строку со студентом, которого требуется удалить, нажать соответствующую кнопку в правой части окна и подтвердить или отменить удаление, нажав, соответственно, «Удалить» или «Отмена» в появившемся окне. Для изменения информации о студенте нужно изменить данные в ячейках той строки, в которой выведены данные о нужном студенте и нажать кнопку «Изменить» справа. После этого, в появившемся окне подтвердить или отменить изменения, нажав, соответственно, «Изменить» или «Отмена». По нажатию кнопки «Назад» пользователь снова вернется в главное окно.

За время разработки клиентского приложения и его тестирования и документирования была создана часть программного комплекса, которая вместе с сервером и СУБД формирует информационную систему для контроля посещаемости студентов учебного заведения.

1. **ЗАКЛЮЧЕНИЕ**

В ходе производственной эксплуатационной практики было разработано клиентское приложение: программа, которая способна взаимодействовать с сервером (получать от него и передавать ему данные) по внешней сети и отображать их для пользователя.

Для достижения поставленной цели надо было решить следующие задачи:

1. Определение целевого назначения создаваемой информационной системы.
2. Изучение встроенных возможностей ПО, которые доступны для создания клиентской части информационной системы.
3. Выбор сетевого протокола и модулей для реализации взаимодействия с сервером.
4. Разработка алгоритма работы клиента.
5. Написание клиентской части программы.
6. Соединение клиента с сервером и тестирование их взаимодействия.

Созданное клиентское приложение позволяет пользователю получить доступ к таким объектам БД, как список студентов и групп и отметкам о посещаемости в удобном для просмотра и редактирования виде, посредством HTTP-запросов к серверу. Такой подход позволяет обезопасить БД от несанкционированных действий со стороны пользователя.

Таким образом, поставленные передо мной задачи были выполнены в срок и в полном объеме. Все необходимые компетенции в области разработки ПО для сетевого оборудования были освоены. Полученная на выходе информационная система, включающая сервер, реализованный Осиповым А.В., и СУБД, используемую для хранения данных о студентах и их посещаемости, с некоторыми доработками может быть использована для простого контроля посещаемости в учебных заведениях.