ОГЛАВЛЕНИЕ

[1. Техническое задание 5](#_Toc72197076)

[1.1 Наименование системы 5](#_Toc72197077)

[1.2 Основания для разработки 5](#_Toc72197078)

[1.3 Исполнитель 5](#_Toc72197079)

[1.4 Назначение и цель разработки системы 5](#_Toc72197080)

[1.5 Содержание работы 5](#_Toc72197081)

[1.5.1 Задачи, подлежащие решению 5](#_Toc72197082)

[1.5.2 Требования к архитектуре АСОИ 6](#_Toc72197083)

[1.5.3 Требования к составу программных компонентов 6](#_Toc72197084)

[1.5.4 Требования к прикладным программам 7](#_Toc72197085)

[1.5.5 Требования к входным/выходным данным 7](#_Toc72197086)

[1.5.6 Требования к временным характеристикам 7](#_Toc72197087)

[1.5.7 Требования к составу технических средств 7](#_Toc72197088)

[1.6 Этапы разработки 8](#_Toc72197089)

[1.7 Техническая документация, предъявляемая по окончании работы 8](#_Toc72197090)

[1.8 Дополнительные условия 9](#_Toc72197091)

[2. НАУЧНО–Исследовательская часть 10](#_Toc72197092)

[2.1 Постановка задачи 10](#_Toc72197093)

[2.2 Перечень задач, подлежащих решению в процессе разработки 11](#_Toc72197094)

[2.3 Анализ существующих аналогов 12](#_Toc72197095)

[2.4 Обоснование выбора языка, среды и платформы разработки 14](#_Toc72197096)

[2.5 Выводы 17](#_Toc72197097)

[3. ПРОЕКТНО-КОНСТРУКТОРСКАЯ ЧАСТЬ 19](#_Toc72197098)

[3.1 Разработка структуры приложения 19](#_Toc72197099)

[3.2 Разработка алгоритмов обработки информации 20](#_Toc72197100)

[3.3 Логическая схема базы данных 21](#_Toc72197101)

[3.4 Описание физической модели базы данных 21](#_Toc72197102)

[3.5 Разработка архитектуры приложения 23](#_Toc72197103)

[3.6 Предлагаемые функции приложения для реализации 28](#_Toc72197104)

# Техническое задание

## Наименование системы

Настоящее Техническое задание определяет требования и порядок создания клиент-серверного приложения для определения местоположения сотрудников.

## Основания для разработки

Основанием для разработки приложения для определения местоположения сотрудников является задание в соответствии с рабочей программой дисциплины «Сетевые технологии в автоматизированных системах обработки информации и управления».

## Исполнитель

Исполнителем проекта является студент Калужского филиала МГТУ им. Н. Э. Баумана, факультета ИУК «Информатики и управления», кафедры ИУК5 «Системы обработки информации», группы ИУК5-61Б, Антюхов Алексей Александрович.

## Назначение и цель разработки системы

Разрабатываемое приложение предназначено для хранения и обновления данных о местоположении персонала.

Целью создания системы является закрепление полученных навыков работы с программными интерфейсами для обеспечения обмена данными между процессами.

## **Содержание работы**

### Задачи, подлежащие решению

В ходе реализации программного продукта необходимо разработать приложение-клиент, которое будет подключаться к серверу с удаленных компьютеров, разработать приложение-сервер, предназначенное для параллельной обработки запросов и работающее в ОС Windows, реализовать взаимодействие двух частей приложения с использованием механизма сокетов.

### Требования к архитектуре АСОИ

Архитектура системы должна быть достаточно гибкой, т.е. допускать относительно простое, без коренных структурных изменений, наращивания функциональности в соответствии с расширением задач её применения.

Должны быть обеспечены безопасность функционирования системы, надежная защита данных от ошибок.

Должен быть обеспечен эргономичный доступ пользователей к сервисам и результатам функционирования АСОИ на основе современных пользовательских интерфейсов.

### Требования к составу программных компонентов

Программный продукт должен представлять собой протестированное клиент-серверное приложение, размещенное на удаленном сервере.

Для передачи данных между сервером и клиентом будет использоваться технология Windows Sockets API (далее Winsock).

Будут использоваться следующие библиотеки:

Winsock 2.0 – для функционирования Winsock

MySQLClient – для функционирования БД

Для представления визуальной части приложения будут использоваться стандартные аппаратные компоненты среды разработки Microsoft Visual Studio – Windows Forms (далее Winfroms).

Каждое окно и поведение включённых в него элементов будет располагаться в отдельном файле с соответствующим названием. Для серверной части приложений клиента и сервера будет выделено по одному файлу. Также клиент и сервер будут содержать один главный файл и функцией main(), в котором будут прописаны начальные инструкции приложений.

### **Требования к прикладным программам**

Для работы БД необходимо приложение с поддержкой технологии MySQL. В моём случае будет использоваться MAMP.

### Требования к входным/выходным данным

Входные данные должны быть логичными и отображать то, что пользователь хочет получить в результате работы программного продукта.

Выходные данные должны максимально правильно и полностью соответствовать действиям конечных пользователей.

выходные данные описать полностью по таблицам из запросов

и входные

### Требования к временным характеристикам

Система должна обеспечивать возможность одновременной работы 10 пользователей при следующих характеристиках времени отклика системы:

* подключение к серверу – не более 5 сек;
* получение ответа от сервера по запросу – не более 10 сек
* время, необходимое для функционирования протокола, передачи данных без разрыва соединения по истечению таймаута – не более 2 минут

### Требования к составу технических средств

Для построения программного продукта необходимы следующие аппаратно-технические и программные средства:

* Персональный компьютер (ПЭВМ)
* В качестве серверной и клиентской частей допускается использование ПК, удовлетворяющего мин требованиям для ОС MS Windows 10
* Наличие сетевого оборудования для функционирования локальной ВС

## Этапы разработки

На стадии разработки технического задания должен быть выполнен этап разработки, согласования и утверждения настоящего технического задания.

На исследовательской стадии должен быть выполнен обзор существующих аналогов программному продукту; выбор программного обеспечения, библиотек для создания продукта.

На стадии проектирования компонентов программного продукта выполняется проектирование архитектуры системы; состава и методов взаимодействия компонентов; алгоритмов обработки и представления данных.

На стадии реализации производится разработка и тестирование спроектированной программы, ввод ее в эксплуатацию, а также оформление технической документации.

## Техническая документация, предъявляемая по окончании работы

Должны быть разработаны следующие программные документы:

1. Расчетно-пояснительная записка:

* техническое задание;
* научно-исследовательская часть;
* проектирований компонентов программного продукта;
* проектно-технологическая часть.

2. Графическая часть.

## Дополнительные условия

Дополнительных условий не предъявляется.

# НАУЧНО–Исследовательская часть

## Постановка задачи

Предприятие — самостоятельный, организационно-обособленный хозяйствующий субъект с правами юридического лица, который производит и сбывает товары, выполняет работы, оказывает услуги. Часто в качестве частичных синонимов термина «предприятие» выступают понятия «фирма», «компания», «корпорация». Распространёнными примерами предприятий являются завод, фабрика, фермерское хозяйство, артель, производственный кооператив и другие.

Офис или контора — помещение, здание, комплекс зданий, в котором работают служащие предприятия (фирмы). В офисе (конторе) принимают клиентов, хранят и обрабатывают документы, архивы и тому подобное.

Главный офис — офис, в котором находится руководство предприятия, место пребывания центрального аппарата компании или предприятия, где размещаются топ-менеджмент компании или предприятия, дирекция, секретариат и другие важные административные подразделения.

Фабрика— промышленное предприятие, основанное на применении машин, характеризующееся крупномасштабным производством. Как правило, состоит из одного или нескольких расположенных неподалёку друг от друга строений, в которых располагаются цеха, сформированные по функции или по виду выпускаемой продукции, а также складские и офисные помещения.

Фабрики имеют большие площади, на них может работать несколько сотен рабочих одновременно.

Территория фабрики может быть разделена на зоны, в которых сотрудники выполняют различную работу, которые служат для определённых отличных друг от друга целей. Например, зона нарезки, зона сушки, зона упаковки, зона хранения.

Каждый работник имеет определённую должность и, соответственно, определённые обязанности и зону работы. Например, стеклодув и художник-декоратор ёлочных игрушек.

За таким большим количеством персонала нужно следить, чтобы эффективность работы не падала ниже определённой нормы, чтобы сотрудники не входили в места, в которых им не положено находиться. Видеонаблюдение может не дать полной картины происходящего на предприятии (слепые зоны), особенно в случае экстренной ситуации, например, из-за плохой видимости. Назначение на должность людей, контролирующих работу других сотрудников, может быть слишком затратно, также имеет место человеческий фактор, например, плохой сон или подкуп\коррупция. Использование же систем контроля местоположения сотрудников позволяет узнать местоположение каждого отдельного сотрудника предприятия в реальном времени, историю его перемещений, его время, проведённое эффективно, за работой.

Главной задачей курсовой работы является проектирование и создание веб-приложения с использованием базы данных для реализации возможностей обновления местоположения сотрудников и просмотра статистики по перемещениям в выбранной форме.

## Перечень задач, подлежащих решению в процессе разработки

Для выполнения поставленной задачи необходимо определить оптимальную архитектуру для построения веб-приложения и средства реализации программного продукта, соответствующего данной архитектуре. Также необходимо исследовать существующие аналоги приложений для кинотеатров, разработать концептуальные и физические модели проектирования базы данных для приложения, разработать базу данных для хранения необходимой информации, спроектировать и разработать визуальную часть приложения и провести её тестирование.

## Анализ существующих аналогов

Разрабатываемая программа не является уникальной и имеет аналоги. Я не могу получить полный доступ к конкретным программам, которые используют в реальных предприятиях. Поэтому я возьму в качестве аналогов рекламные предложения подобных систем.

1. «GPShome» - Сервис для контроля сотрудников в офисе внутри закрытых помещений.



Рисунок 1 – демо-версия личного кабинета сервиса «GPShome»

На сайте относительно подробно описан весь функционал сервиса.

Сервис GPShome.ru предлагает решение для отслеживания сотрудников компании. Можно контролировать местоположение каждого сотрудника в режиме онлайн, получать статистику и визуализацию по всем перемещениям сотрудников за любой выбранный период. Можно формировать широкий перечень отчетов о фактическом нахождении всех сотрудников. Можно контролировать приход и уход сотрудников в офис и из офиса. Контролировать время нахождения на рабочем месте. Если в компании существует ограничения для сотрудников на посещение ими определенных помещений, то сервис позволяет собирать статистику по посещению этих помещений, а также подавать сигнал тревоги при нахождении неавторизированных сотрудников в этих помещениях.

1. Система RealTrac

Модульная система мониторинга персонала для зданий и офисов.



Рисунок 2 – направления позиционирования персонала системы RealTrac «Производство»

В отличии от прошлой системы, эта ориентирована на большие производственные предприятия, а не на офисные помещения и здания.

Решаемые задачи:

* Контроль местоположения рабочих на территории предприятия
* Контроль уровней доступа рабочих
* Контроль рабочего времени, времени, проведенного в геозоне
* Аварийное оповещение персонала в случае ЧП
* Контроль эвакуации персонала

## Обоснование выбора языка, среды и платформы разработки

В качестве основного языка разработки был выбран С# и разработка с использованием Winforms в Microsoft Visual Studio.

Основные языки программирования для разработки десктопных приложений для MS Windows – C++, С#, Java. Далее проведу небольшое сравнение.

C# в сравнении с C++ и Java имеет поддержку аппаратных средств для работы с графическим интерфейсом внутри основной среды разработки MS Visual Studio. В случае с C++ и Java необходимо дополнительное ПО в виде библиотек, фреймворков или сред разработки с необходимым функционалом.

C++ является наиболее производительным языком программирования из представленных. Но в данном случае приложение не будет сильно загружать систему, производительность приложения будет упираться в пропускную способность локальной сети и скорость работы ПК с данными., поэтому жертва скоростью и удобством программирования на C# и Java в угоду производительности на C++ не целесообразна. В то же время на MS Windows язык программирования C# будет более производительным, чем Java, так как разрабатывался специально под Windows.

Остальные отличия касаются функционала и его реализации в конкретном языке, поэтому затрагивать их не имеет смысла.

Среди сред разработки под Windows выделяются MS Visual Studio и MS Visual Studio Code. Они были разработаны компанией Microsoft специально под Windows и являются наиболее удобными, производительными, с богатым функционалом.

VS Code поддерживает множество языков программирования (можно расширить установкой соответствующих плагинов и библиотек) и содержит в себе среду их выполнения, по сути являясь усовершенствованным редактором кода. Это больше подходит для поддержки готового кода, тестирования, веб-разработки.

VS «из коробки» также поддерживает все основные языки программирования за исключением Java (можно расширить установкой соответствующих плагинов и библиотек), но в то же время имеет в разы больший функционал в сравнении с VS Code. Например, создание и работа с БД, конструктор графического интерфейса, фреймворк. VS имеет аппаратную поддержку таких технологий как WinForms и WinSock. Именно эти технологии я собираюсь использовать.

VS Code может работать с WinSock, но не может полноценно работать с WinForms, так как может только скомпилировать код. Мне же нужна возможность удобно редактировать графический интерфейс моего приложения. Соответственно мой выбор – MS Visual Studio.

Подробнее про WinSock. Это технология поддерживается только ОС Windows и является частью среды разработки VS. .NET имеет аппаратную поддержку данной технологии в лице класса System.Net.Sockets.Socket, поэтому для работы достаточно прописать using System.Net.Sockets;

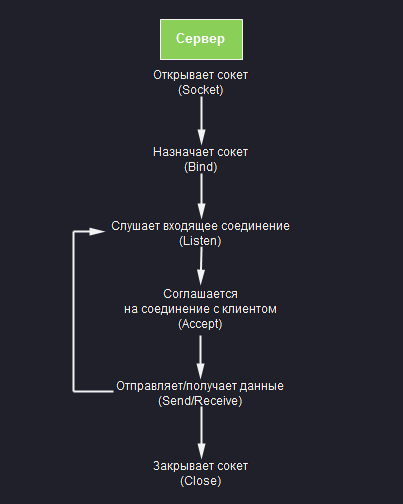


Рисунок 3 – Алгоритм работы серверного сокета (переделать под алгоритм блок-схем)

* Windows Forms имеет несколько уровней визуализации
* Разработка на XAML даёт возможность выводить свойства элементов окон в отдельные стили, по аналогии с CSS. Значит, чтобы изменить внешний вид окон достаточно заменить файл со стилями
* Можно строить веб-подобные приложения на основе страниц, что дает возможность пользоваться навигацией.
* Независимость от разрешения. WPF берет на себя компоновку элементов интерфейса подстраивая его под разные разрешения.
* Windows Forms достаточно просто связать с базой данных.

Рассмотреть виндовс линукс мак ос, выбрать.

Основными ОС для ПК на сегодняшний день являются MS Windows, Mac OS и Linux OS. Проведём сравнение.

Сразу же отклоню Mac OS, так как она поставляется только с техникой компании Apple, которая имеет цену намного выше, чем аналогичная под управлением другой ОС. В данном случае переплата в несколько раз за излишнюю производительность и отсутствие возможности дальнейшего апгрейда нецелесообразна. Linux бесплатна, базовая версия Windows стоит 200$ в официальном магазине компании Microsoft.

Linux имеет много версий и вариаций, но подавляющее большинство из них имеют большее быстродействие, чем Windows, при одинаковых комплектующих. Но в то же время ОС на базе Linux из-за их разнообразия может не иметь нужных драйверов для установленных комплектующих, когда под Windows драйвера будут всегда. Т.е. мы жертвуем некоторой производительностью ради удобства установки и работы с ОС.

Также стоит отметить, что Windows является самой популярной в мире ОС. Вероятно, не человека, который бы был знаком с техникой, но не пользовался Windows. Linux же из трёх представленных ОС является наименее популярной и установлена на менее чем 5% ПК по всему миру. Чтобы полноценно пользоваться Linux нужно время на адаптацию, в случае с Windows это не требуется.

## Выводы

Таким образом, исходя из требований к реализуемой системе, рассмотрения возможностей наиболее подходящих инструментов, вариантов разработки и последующего их сравнения было решено использовать следующие решения:

Основной язык программирования — C#. Для разработки приложения была выбрана интегрированная среда разработки Microsoft Visual Studio. В качестве базы данных — MySQL.

# ПРОЕКТНО-КОНСТРУКТОРСКАЯ ЧАСТЬ

## Разработка структуры приложения

Приложение состоит из трёх частей. Из клиента, написанного с использованием Winforms, сервера и реляционной базы данных MySQL. То есть я использую «трёхуровневую архитектуру».

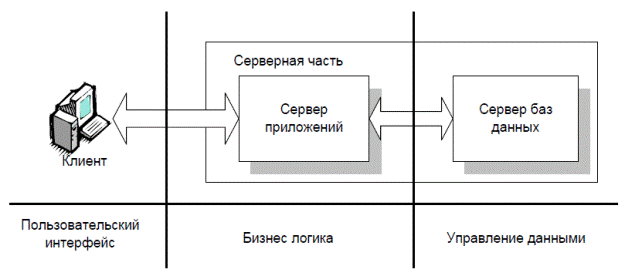


Рисунок 4 - Макет структуры приложения

Пользователь через интерфейс приложения-клиента выбирает действие, которое он хочет произвести. Клиент через сокет отправляет серверу сообщение с операцией, которую требуется выполнить, и входные данные для составления запроса к БД.

Вся работа по подготовке запросов происходит на сервере. Полученные от клиента входные данные он подставляет в макет нужного запроса и отправляет его БД.

БД хранит данные и отвечает на приходящие от сервера запросы.

Сначала необходимо разработать БД и сервер и наладить между ними подключение. Так можно выявить основные проблемы в процессе получения данных от БД. Если обмен данными между сервером и БД происходит стабильно и ожидаемым образом, значит ошибок в этой части структуры нет. Далее необходимо разработать клиент и его взаимодействие с сервером. Так как сервер исправно получает данные от БД, нужно создать интерфейс для передачи этих данных от сервера на клиент через сокет.

Приложение для взаимодействия с базой данных написано на языке C# при помощи Windows Forms. Windows Forms позволяет разрабатывать интеллектуальные клиенты. Интеллектуальный клиент — это приложение с полнофункциональным графическим интерфейсом, простое в развертывании, способное работать при наличии или отсутствии подключения к Интернету и использующее более безопасный доступ к ресурсам на локальном компьютере по сравнению с традиционными приложениями Windows.

## Разработка алгоритмов обработки информации

Все функции приложения активируются после нажатия соответствующих кнопок.

При помощи функций коннектора mySqlConnection.Open(); и mySqlConnection.Close(); либо при помощи созданного класса DB, в который встроены эти функции, открывается и закрывается доступ к базе данных.

После открытия доступа используя, MySqlCommand и MySqlDataAdapter, программа получает доступ к базе данных. Считывает их от туда таблицы, добавляет или изменяет строки в таблицах, согласно написанному на SQL коду.

Пример команды:

MySqlCommand command = new MySqlCommand("SELECT \* FROM `users` WHERE `login` = @ln AND `password` = @pswd", db.getConnection());

алгоритм заполнения и алгоритм получения информации (максимально в общем, как всё в теории должно работать)

!!!! с использованием сетевых технологий – алгоритм работы сокета

## Логическая схема базы данных

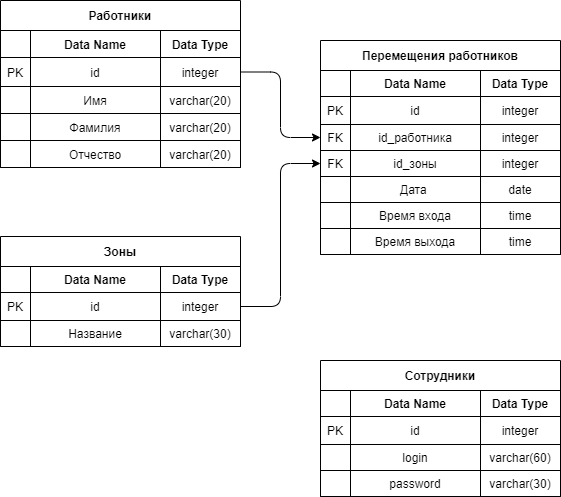


Рисунок 5 – Схема логической модели данных

## Описание физической модели базы данных

Опишем сущности, использованные в базе данных.

`users`- Сущность для сотрудников, работающих с приложением. Она имеет 3 поля.

* ID(Первичный ключ) тип данных bigint(20) (serial)
* Логин тип данных varchar(60)
* Пароль тип данных varchar(30)

`members` - Сущность работника. В ней присутствуют 4 поля

* ID(Первичный ключ) тип данных bigint(20) (serial)
* Имя тип данных varchar(20)
* Фамилия тип данных varchar(20)
* Отчество тип данных varchar(20)

`rooms` - Сущность работника. В ней присутствуют 2 поля

* ID(Первичный ключ) тип данных bigint(20) (serial)
* Название тип данных varchar(30)

`st` - Сущность перемещения работников. Имеет 6 полей

* ID (Первичный ключ) тип данных bigint(20) (serial)
* id работника (Внешний ключ) тип данных int(11)
* id зоны (Внешний ключ) тип данных int(11)
* Дата входа/выхода тип данных date
* Время входа тип данных time
* Время выхода тип данных time

На данный момент есть понимание, что было бы целесообразнее использовать для даты и времени входа/выхода 2 поля типа datetime и уже из них по отдельности извлекать дату и время.

## Разработка архитектуры приложения

Имеется несколько Windows Form, переход между ними осуществляется при помощи нажатии кнопок. При нажатии соответствующих кнопок в окне программы активируются функции, содержащие в себе код на языке SQL.

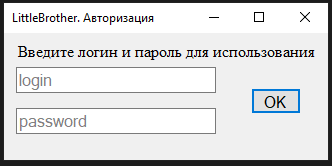


Рисунок 6 - Окно входа

При нажатии на кнопку OK активируется код, вызывающий такой запрос

"SELECT \* FROM `users` WHERE `login` = @ln AND `password` = @pswd"

@ln и @pswd поля, принимающие в себя логин и пароль из полей login\_textBox.Text и PasswordInput\_textBox.Text соответственно.

После ввода правильного пароля мы попадаем на главную форму.



Рисунок 7- Главная форма

В нём мы помещаем в соответствующие comboBox-ы полные имена сотрудников с помощью запроса "SELECT CONCAT(`fName`, ' ', `sName`, ' ', `tName`) AS fullName FROM `members`" и названия помещений (зон) с помощью запроса "SELECT `room` FROM `rooms`".

Также в этом окне при нажатии кнопки «Добавить» выполняется запрос

UPDATE `st` SET `time\_out` = CURRENT\_TIME()

WHERE `member\_id` = (SELECT id FROM `members`

WHERE CONCAT(`fName`, ' ', `sName`, ' ', `tName`) = 'names\_comboBox.Text')

ORDER BY `id` DESC LIMIT 1;

INSERT INTO `st` (`id`, `member\_id`, `room\_id`, `date`, `time\_in`, `time\_out`)

VALUES (NULL,

(SELECT id FROM `members` WHERE CONCAT(`fName`, ' ', `sName`, ' ', `tName`) = 'names\_comboBox.Text'),

(SELECT id FROM `rooms` WHERE `room` = 'rooms\_comboBox.Text'),

CURRENT\_DATE(),

CURRENT\_TIME(),

NULL);

Красным цветом помечены вставки значений полей главной формы, данный запрос в приложении разбивается на несколько строк, которые после объединяются в один цельный запрос.

При нажатии кнопки «Получить» открывается форма, соответствующая выбранному виду отчёта в соответствующем поле. Каждая из этих форм при открытии выполняет запрос, используя для вставки в него текущий текст из поля «ФИО» области «СВОДКА О ПЕРЕМЕЩЕНИЯХ».



Рисунок 8 – Форма «Маршрут сотрудника»

При открытии форма «Маршрут сотрудника» выполняет запрос:

SELECT DATE\_FORMAT(`date`, '%d.%m.%Y') AS date,

(SELECT CONCAT(`fName`, ' ', `sName`, ' ', `tName`) FROM `members`

WHERE CONCAT(`fName`, ' ', `sName`, ' ', `tName`) = '\_memberName') AS name,

(SELECT `room` FROM `rooms` WHERE rooms.id = st.room\_id) AS room,

`time\_in`, `time\_out`,

TIMEDIFF(`time\_out`, `time\_in`) AS duration

FROM `st` INNER JOIN `members` AS m

WHERE CONCAT(`fName`, ' ', `sName`, ' ', `tName`) = '\_memberName'

AND st.member\_id = m.id

AND `time\_out` IS NOT NULL

#AND st.date = CURRENT\_DATE() - 1



Рисунок 9 – Форма «Пребывание сотрудника в зонах»

При открытии форма «Пребывание сотрудника в зонах» выполняет запрос:

SELECT `room`, SEC\_TO\_TIME(SUM(duration\_inSEC)) AS duration FROM

(SELECT (SELECT `room` FROM `rooms` WHERE rooms.id = st.room\_id) AS room,

CASE WHEN `time\_in` < `time\_out` THEN

TIMESTAMPDIFF(second, `time\_in`, `time\_out`)

ELSE

86400 + TIMESTAMPDIFF(second, `time\_in`, `time\_out`) END AS duration\_inSEC

FROM `st`

INNER JOIN `members` AS m

WHERE CONCAT(`fName`, ' ', `sName`, ' ', `tName`) = '\_memberName'

AND st.member\_id = m.id

AND `time\_out` IS NOT NULL

AND WEEK(st.date) = WEEK(CURRENT\_DATE())) as secc

GROUP BY secc.room



Рисунок 10 – Форма «Переработка сотрудника»

При открытии форма «Переработка сотрудника» выполняет запрос:

SELECT (SELECT CONCAT(`fName`, ' ', `sName`, ' ', `tName`) FROM `members`

WHERE CONCAT(`fName`, ' ', `sName`, ' ', `tName`) = '\_memberName') AS fullName,

28800/3600 AS plan,

SUM(duration\_inSEC) /3600 AS fact,

(SUM(duration\_inSEC) - 28800) /3600 AS overtime\_inSEC FROM

(SELECT (SELECT `room` FROM `rooms` WHERE rooms.id = st.room\_id ) AS room,

CASE WHEN `time\_in` < `time\_out` THEN

TIMESTAMPDIFF(second, `time\_in`, `time\_out`)

ELSE

86400 + TIMESTAMPDIFF(second, `time\_in`, `time\_out`) END AS duration\_inSEC

FROM `st`

INNER JOIN `members` AS m

WHERE CONCAT(`fName`, ' ', `sName`, ' ', `tName`) = '\_memberName'

AND st.member\_id = m.id

AND st.room\_id <> 2

AND `time\_out` IS NOT NULL

AND WEEK(st.date) = WEEK(CURRENT\_DATE()) ) as secc

GROUP BY `fullName`

В последних трёх запросах присутствует такая часть, как \_memberName. Это переменная типа string, которая хранит в себе текст из поля «ФИО» области «СВОДКА О ПЕРЕМЕЩЕНИЯХ» главной формы. Соответственно, в каждый из этих трёх запросов подставляется полное имя рабочего, указанное в соответствущем поле «ФИО».

Вывод данных этих запросов производится в dataGridView.

После каждого последующего нажатия копки «Получить» текущее окно сводки не закрывается автоматически и не меняет содержащихся в нём данных. Создаётся новое окно с выбранными параметрами. «Старые(-ое)» окна(-о) надо закрыть самому в случае необходимости.

## Предлагаемые функции приложения для реализации