



Министерство науки и высшего образования Российской Федерации
Калужский филиал
федерального государственного бюджетного
образовательного учреждения высшего образования
«Московский государственный технический университет имени Н.Э. Баумана
(национальный исследовательский университет)»
(КФ МГТУ им. Н.Э. Баумана)

ФАКУЛЬТЕТ _ ИУ-КФ «Информатика и управление»

КАФЕДРА _ ИУ5-КФ «Системы обработки информации»

РАСЧЕТНО - ПОЯСНИТЕЛЬНАЯ ЗАПИСКА

к курсовой работе на тему:

Приложение для определения местоположения сотрудников

по дисциплине Архитектура автоматизированных систем обработки
информации и управления

Студент гр. ИУК5-51Б _____ (Антюхов А.А.)
(подпись) (Ф.И.О.)

Руководитель _____ (Смирнов М.Е.)
(подпись) (Ф.И.О.)

Оценка руководителя _____ баллов _____
30-50 (дата)

Оценка защиты _____ баллов _____
30-50 (дата)

Оценка проекта _____ баллов _____
(оценка по пятибалльной шкале)

Комиссия: _____ (_____)
(подпись) (Ф.И.О.)

_____ (_____)
(подпись) (Ф.И.О.)

_____ (_____)
(подпись) (Ф.И.О.)

Калуга 2020
Калужский филиал

федерального государственного бюджетного образовательного учреждения высшего образования
«Московский государственный технический университет имени Н.Э. Баумана
(национальный исследовательский университет)»
(КФ МГТУ им. Н.Э. Баумана)

УТВЕРЖДАЮ

Заведующий кафедрой ИУ5-КФ

_____ (_____)

«___» _____ 20__ г.

ЗАДАНИЕ **на выполнение курсового проекта**

по дисциплине Архитектура автоматизированных систем обработки информации и управления

Студент _____ Антюхов А.А., ИУК5-51Б _____
(фамилия, инициалы, индекс группы)

Руководитель _____ Смирнов М.Е. _____
(фамилия, инициалы)

График выполнения проекта: 25% к 4 нед., 50% к 7 нед., 75% к 10 нед., 100% к 14 нед.

1. Тема курсового проекта

Приложение для определения местоположения сотрудников

2. Техническое задание

Разработать приложение, способное добавлять в БД данные о перемещении сотрудников, что имитирует работу ключ-карты, и выводить эти данные.

3. Оформление курсового проекта

3.1. Расчетно-пояснительная записка на _____ листах формата А4.

3.2. Перечень графического материала КП (плакаты, схемы, чертежи и т.п.) _____

Дата выдачи задания «___» _____ 20__ г.

Руководитель курсового проекта _____ / _____ /
(подпись) (Ф.И.О.)

Задание получил _____ / _____ / «___» _____ 20__ г.
(подпись) (Ф.И.О.)

Оглавление

| | |
|---|--|
| <i>Архитектура автоматизированных систем обработки информации и управления</i> | <i>0</i> |
| 1.ТЕХНИЧЕСКОЕ ЗАДАНИЕ | Ошибка! Закладка не определена. |
| 1.1 Общие сведения | Ошибка! Закладка не определена. |
| 1.1.1 Наименование | Ошибка! Закладка не определена. |
| 1.1.2 Исполнитель..... | Ошибка! Закладка не определена. |
| 1.1.3 Перечень нормативно-технических документов, методических материалов, использованных при разработке ТЗ:..... | Ошибка! Закладка не определена. |
| 1.1.4 Плановые сроки по созданию приложения | Ошибка! Закладка не определена. |
| 1.2 Назначение и цели создания системы | Ошибка! Закладка не определена. |
| 1.2.1 Назначение системы | Ошибка! Закладка не определена. |
| 1.2.2 Цели создания системы | Ошибка! Закладка не определена. |
| 1.3 Характеристики объекта автоматизации..... | Ошибка! Закладка не определена. |
| 1.4 Требования к системе..... | Ошибка! Закладка не определена. |
| 1.4.1 Требования к системе в целом..... | Ошибка! Закладка не определена. |
| 1.4.2 Требования к надежности..... | Ошибка! Закладка не определена. |
| 1.4.3 Требования к функциям, выполняемым системой | 22 |
| 1.4.4 Требования к информационному обеспечению системы | 22 |
| 1.4.5 Требования к методическому обеспечению | 25 |
| 1.5 Состав и содержание работ по созданию(развитию) системы..... | Ошибка! Закладка не определена. |
| 1.6 Порядок контроля и приемки системы | Ошибка! Закладка не определена. |
| 1.6.1 Методы испытания системы..... | Ошибка! Закладка не определена. |
| 1.6.2 Общие требования к приемке работ по стадиям | Ошибка! Закладка не определена. |
| 1.6.3 Статус приемочной комиссии..... | Ошибка! Закладка не определена. |
| 1.7 Требования к составу и содержанию работ по подготовке объекта автоматизации к вводу системы в действие | Ошибка! Закладка не определена. |
| Никаких особых требований по подготовке программы к вводу в эксплуатацию не требуется | Ошибка! Закладка не определена. |
| 1.8 Требования к документированию..... | Ошибка! Закладка не определена. |
| 1.9 Источники разработки | Ошибка! Закладка не определена. |
| 2.НАУЧНО-ИССЛЕДОВАТЕЛЬСКАЯ ЧАСТЬ | 7 |
| 2.1. Постановка задачи | 7 |
| 2.2 Описание предметной области | 7 |
| 2.2.1 Общие сведения | 7 |
| 2.2.2 Анализ аналогов..... | 9 |
| 2.3 Сценарий использования..... | 11 |

| | |
|---|--|
| 2.4 Задача автоматизации | 11 |
| 2.5 Перечень задач, подлежащих решению в процессе разработки | 11 |
| 2.6 Обоснование выбора платформы для разработки | 12 |
| 3.ПРОЕКТНО-КОНСТРУКТОРСКАЯ ЧАСТЬ..... | 13 |
| 3.1. Разработка структуры приложения | 13 |
| 3.2. Разработка алгоритмов обработки информации | 14 |
| 3.3.Разработка архитектуры приложения | 15 |
| Функции: closeButton_Click, closeButton_MouseEnter, closeButton_MouseLeave, topPanel_MouseMove, topPanel_MouseDown, mainPanel_MouseMove, back_Click и mainPanel_MouseDown я уже описал выше, поэтому описывать их второй раз не буду..... | 19 |
| 3.5.Разработка внешнего вида и дизайна приложения | 20 |
| 4.ПРОЕКТНО-ТЕХНОЛОГИЧЕСКАЯ ЧАСТЬ | Ошибка! Закладка не определена. |
| 4.1. Порядок развёртывания системы..... | Ошибка! Закладка не определена. |
| 4.2. Краткое руководство пользователя..... | Ошибка! Закладка не определена. |
| 4.3. Возможности улучшений | Ошибка! Закладка не определена. |
| ЗАКЛЮЧЕНИЕ | 26 |
| СПИСОК ИСПОЛЬЗОВАННОЙ ЛИТЕРАТУРЫ | 26 |

1.ТЕХНИЧЕСКОЕ ЗАДАНИЕ

1.1 Общие сведения

1.1.1 Наименование

Приложение для учёта и определения местоположения сотрудников «Мышь»

1.1.2 Исполнитель

Студент КФ МГТУ Им. Н.Э. Баумана

Антюхов Алексей Александрович гр. ИУК5-51Б

1.1.3 Перечень нормативно-технических документов, методических материалов, использованных при разработке ТЗ:

-Методические указания по выполнению курсовой работы по дисциплине “Базы данных”, автором которых является В.Ю. Кириллов.

-На основании Технического задания.

1.1.4 Плановые сроки по созданию приложения

Начало работ – 1 сентября 2020 года, окончание работ -1 декабря 2020 года.

По мере выполнения работы результаты загружаются на сервис bitbucket по адресу <https://github.com/VektorHL/LittleBrother>, где заказчик может следить за выполнением работ и вносить коррективы, а также презентуются в личных встречах с заказчиком.

1.2 Назначение и цели создания системы

1.2.1 Назначение системы

Автоматизация процесса сбора и обработки данных местоположений персонала.

1.2.2 Цели создания системы

Система должна хранить и обновлять данные о местоположении персонала.

1.3 Характеристики объекта автоматизации

Объект автоматизации – учёт местоположения персонала.

1.4 Требования к системе

1.4.1 Требования к системе в целом

Структуру систему составляют

-СУБД

- Приложение на ПК

1.4.2 Требования к надежности

Доступ к программе должен иметь только уполномоченный сотрудник.

1.4.3 Требования к функциям, выполняемым системой

Программа должна обеспечивать реализацию следующих функций:

- 1) Обновление местоположения сотрудника
- 2) Вывод информации о местоположении сотрудника в течение рабочего дня
- 3) Вывод информации об эффективности работы сотрудника

1.4.4 Требования к информационному обеспечению системы

Хранение и обслуживание данных в приложении должно происходить с помощью одной из существующих систем управления базами данных. Информация должна сохранять целостность, не должно возникать аномалий при попытке удаления, изменения и добавления данных.

1.4.5 Требования к методическому обеспечению

Разработку системы требуется сопровождать расчетно-пояснительной запиской и графической частью. Расчетно-пояснительная записка состоит из следующих частей:

1. Техническое задание
2. Исследовательская часть.
3. Проектно-конструкторская часть.

4. Проектно-технологическая часть.

1.5 Состав и содержание работ по созданию(развитию) системы

Первый этап – изучение предметной области, определение состава и структуры данных. В результате прохождения первого этапа определяются: логическая структура данных, ТЗ, описание объекта автоматизации и целей разработки.

Второй этап – анализ пользовательских сценариев работы. В результате создается пользовательский интерфейс.

Третий этап – анализ взаимодействия веб-приложения с серверной компонентой. В результате определяются: архитектура приложения и структура классов.

1.6 Порядок контроля и приемки системы

1.6.1 Методы испытания системы

Испытания проектируемой системы проводятся заказчиком.

1.6.2 Общие требования к приемке работ по стадиям

Место проведения приемки работ КФ МГТУ им. Н.Э. Баумана.

Сроки проведения приёмки работ – 1 октября, 15 октября, 1 ноября, 15 ноября, 1 декабря и другие даты, согласованные с заказчиком и исполнителем

1.6.3 Статус приемочной комиссии

Прием работы на всех этапах осуществляется непосредственным руководителем курсовой работы. По окончании разработки работоспособное приложение демонстрируется на комиссии.

1.7 Требования к составу и содержанию работ по подготовке объекта автоматизации к вводу системы в действие

Особые требования к подготовке перед развёртыванием программы отсутствуют

1.8 Требования к документированию

-ТЗ по проектируемой системе

-Расчетно-пояснительная записка

-Презентация Microsoft PowerPoint

1.9 Источники разработки

Техническое задание выполнено на основе следующего стандарта ГОСТ-34.602-89

2. НАУЧНО-ИССЛЕДОВАТЕЛЬСКАЯ ЧАСТЬ

2.1. Постановка задачи

Требуется разработать программу для сотрудников производства/офиса.

2.2 Описание предметной области

2.2.1 Общие сведения

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.

В рамках курсовой работы приложение будет иметь возможность ведения учёта местоположения сотрудников, окна для просмотра истории их перемещений, просмотра времени, проведённого внутри посещённых зон, просмотра эффективного рабочего времени.

2.2.2 Анализ 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.

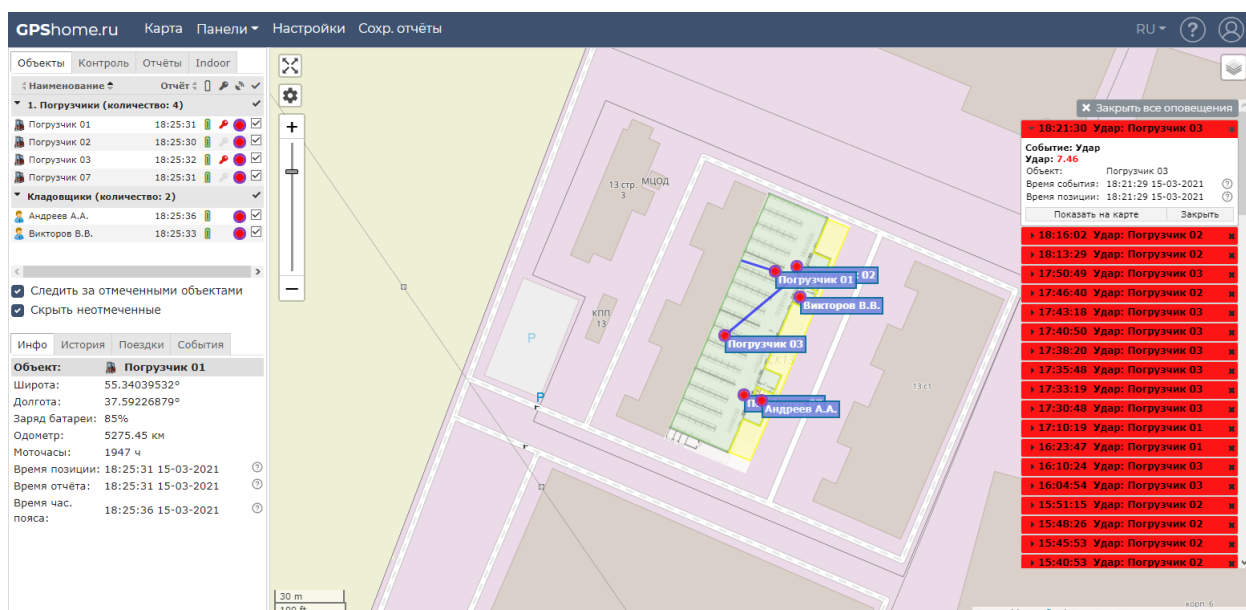


Рисунок 1 – демо-версия личного кабинета сервиса «GPShome»

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 существуют 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.

2) Система RealTrac

Модульная система мониторинга персонала для зданий и офисов.

ПРИНЦИП РАБОТЫ И КОМПОНЕНТЫ

Позиционирование персонала на промышленном предприятии может осуществляться по трем направлениям:

ЗОНАЛЬНОЕ ПОЗИЦИОНИРОВАНИЕ



Определение местоположения рабочего или техники с точностью до метра или геозоны.

- Точность: +/- 20 метров
- Дальность радиобнаружения: до 100 метров
- Конфигурация зоны: от 20 метров до зоны*

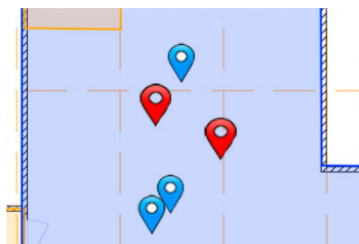
* - возможна настройка зон по требованию Заказчика

- Канал обратной связи: BLE или другой протокол передачи данных*

* - Под требования Заказчика

- Ограничения: Работа только в зоне радиовидимости точек доступа
- Затраты на внедрение: низкие

ГЛОБАЛЬНОЕ ПОЗИЦИОНИРОВАНИЕ



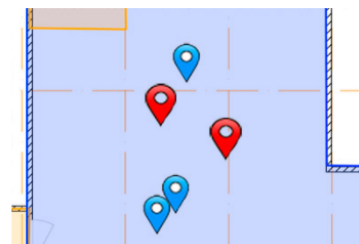
Определение местоположения рабочего или техники с точностью до метра.

- Точность: +/- 1-5 метров
- Дальность радиобнаружения: не ограничена
- Конфигурация зоны: от 10 м2
- Канал обратной связи: Wi-Fi, LoRa или другой протокол передачи данных*

* - Под требования Заказчика

- Ограничения: Работа в условиях видимости минимум 4х спутников
- Затраты на внедрение: низкие

ГИБРИДНОЕ ПОЗИЦИОНИРОВАНИЕ



Объединяет в себе преимущества зонального и глобального позиционирования.

- Точность: +/- 1-5 (20) метров
- Дальность обнаружения: не ограничена
- Конфигурация зоны: от 10 м2
- Канал обратной связи: Wi-Fi, LoRa или другой протокол передачи данных*

* - Под требования Заказчика

- Ограничения: В помещении работает зональное позиционирование. На открытой территории работает глобальное позиционирование
- Затраты на внедрение: средние

Рисунок 2 – направления позиционирования персонала системы RealTrac «Производство»

В отличие 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

2.3 Сценарий использования

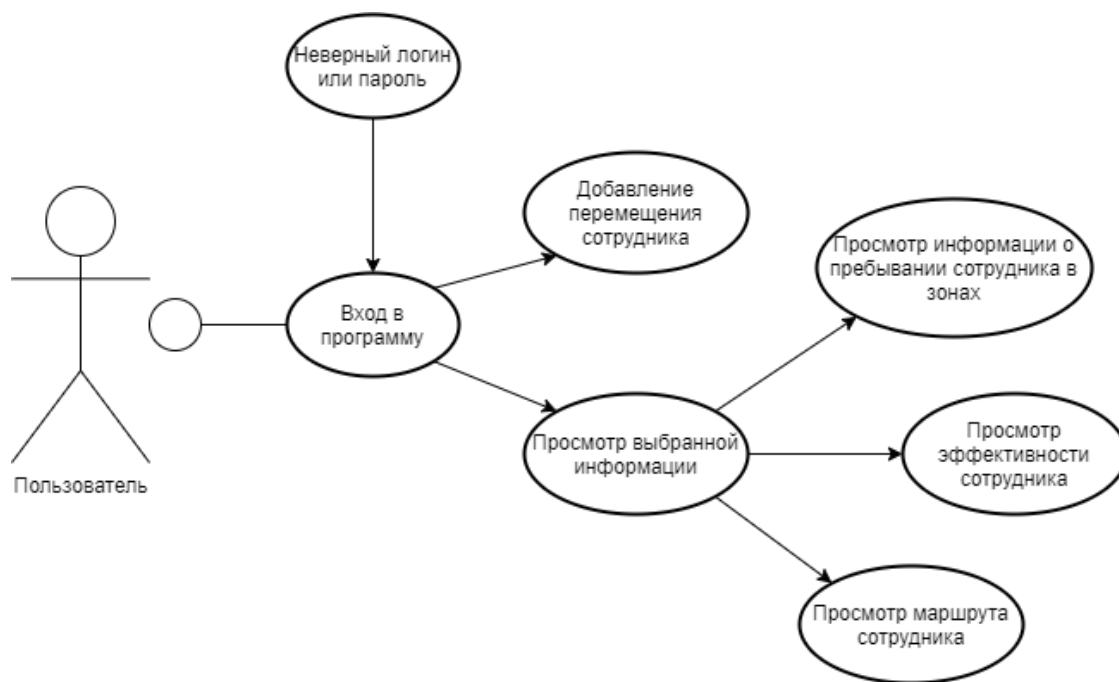


Рисунок 4- Сценарий использования

2.4 Задача автоматизации

Задачей автоматизации является упрощение процесса регистрации клиента, предоставление удобного способа обработки и просмотра данных для администратора.

2.5 Перечень задач, подлежащих решению в процессе разработки

Для создания системы необходимо:

1. Изучить предметную область и на основе полученной информации определить структуру данных, необходимую для выполнения поставленных задач.
2. Разработать план поэтапного формирования
3. Создать или изучить готовые алгоритмы необходимых операций.

4. Определить взаимодействия между базой данных и программой

5. Создать интерфейс взаимодействия пользователя с программой

2.6 Обоснование выбора платформы для разработки

Основная платформа для разработки это Windows Forms на языке C# в Visual Studio.

- Windows Forms имеет несколько уровней визуализации
- Разработка на XAML даёт возможность выводить свойства элементов окон в отдельные стили, по аналогии с CSS. Значит, что бы 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.

3.ПРОЕКТНО-КОНСТРУКТОРСКАЯ ЧАСТЬ

3.1. Разработка структуры приложения

Приложение состоит из двух частей. Из клиента, написанного в Windows Forms и реляционной база данных MySQL. То есть я использую «двухзвенную архитектуру».

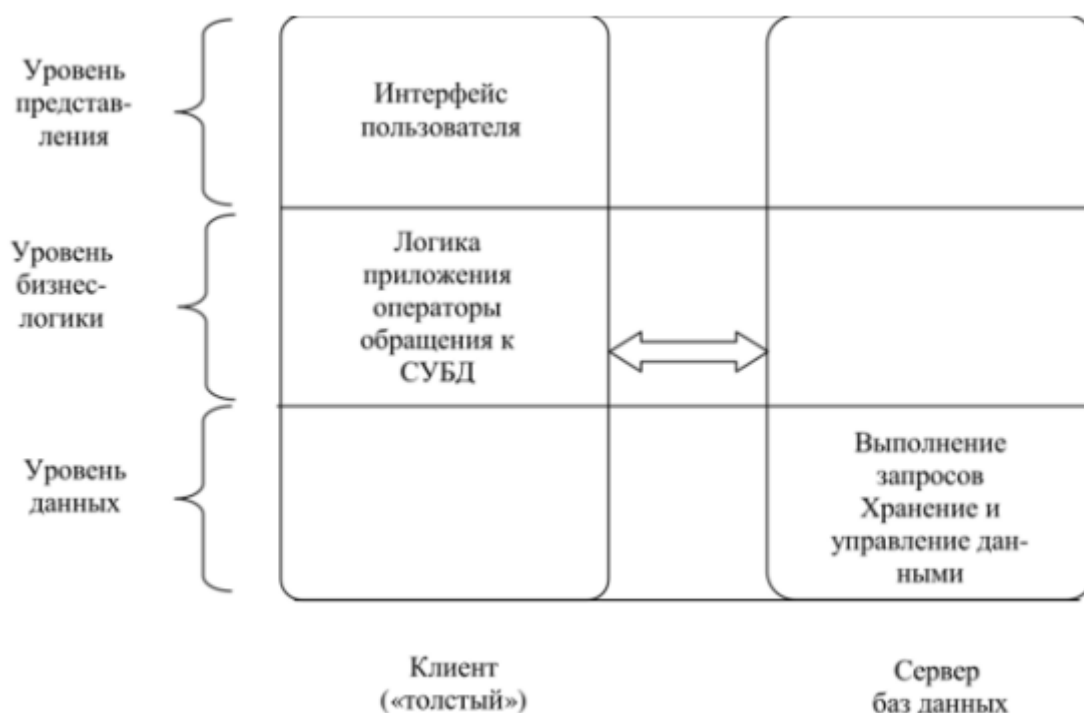


Рисунок 5- Макет структуры приложения

База данных хранит данные. Вся работа по подготовке запросов происходит на клиенте

Это даёт следующие преимущества:

- Поддержка многопользовательского режима работы с общими данными большого объёма.
- Обеспечение сохранности и целостности данных – в базе данных проще обеспечить защиту и целостность данных по сравнению с рабочими станциями клиентов.

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.

3.2. 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());
```

Приложение должно принимать новые данные, обновлять уже имеющиеся данные, выводить данные согласно условиям поиска.

При работе с базой данных, первое, что необходимо сделать, установить связь с этой базой. Для этого я скачал специальный «connector»,

он устанавливается в Visual Studio, он идёт как «плагин» и позволяет связать моё приложение с базой данных. Я использовал коннектор «mysql connector».

После того как я скачал «mysql connector», я подключил его к студии через файл MySqlData.dll.

3.3.Разработка архитектуры приложения

Приложение имеет следующую архитектуру:

Имеется несколько окон Windows Form, переход между ними осуществляется при помощи нажатий кнопок. При нажатии соответствующих кнопок в окне программы активируются функции, либо переключающие окна, либо закрывающие программу, либо содержащие в себе код на языке SQL.

Начнём с описания разработанного класса.

Класс DB служит для связи клиента с базой данных:

```
class DB
{
    MySqlConnection connection = new
    MySqlConnection("server=localhost; port=3306; username=root;
    password=root; database=littlebrother");
    public void openConnection()
    public void closeConnection()
    public MySqlConnection getConnection()
}
```

Я создаю объект на основе класса «MySqlConnection» и в параметры прописываю строку, которая вмещает в себя все характеристики для подключения. Так же класс содержит 3 функции «openConnection()» открывает соединение, если оно закрыто. Вторая closeConnection()»закрывающая соединение, и третья «getConnection()»для проверки наличия соединения.

Теперь опишу похожие участки кода программы, встречающиеся во всех окнах.

Заккрытие программы:

```
private void closeButton_Click(object sender, EventArgs e)

private void closeButton_MouseEnter(object sender, EventArgs e)

private void closeButton_MouseLeave(object sender, EventArgs e)
```

Для закрытия программы создано три функции. Первая функция закрывает программу, вторая подсвечивает крестик зелёным при наведении на него. Третья функция меняет цвет на исходный если не закрывать приложение.

Передвижение окна:

```
Point lastPoint;

private void mainPanel_MouseDown(object sender, MouseEventArgs
e)

private void mainPanel_MouseMove(object sender, MouseEventArgs
e)
```

Концепция: панель перемещается в то место, где сейчас находится мышка, но перемещается только в том случае, если левая клавиша мыши зажата

Создаёт объект класса «Point». И функции для отслеживания положения курсора и нажатия ЛКМ. Вторая функция осуществляет передвижение окна за курсором.

Смена окна:

Так как в программе реализованы 8 видов смены окна, но они все почти не отличаются, покажу этот код на примере кнопки «Назад»

```
private void back_Click(object sender, EventArgs e)
```

После нажатия кнопки, текущая форма «скрывается» и создаётся новый экземпляр формы Main, после чего открывается только что созданное окно.

SQL запросы.:

Вызов SQL запросов происходит после нажатия на соответствующую кнопку, приведу в качестве примера кнопку входа в стартовом окне авторизации:

```
private void ButtonLogin_Click(object sender, EventArgs e)
```

Программа записывает данные из текстовых полей loginField.Text; и passField.Text; В переменные типа string. После этого создаётся объект написанного мной класса DB и выделим под него память. Создаю объекты DataTable и MySqlConnection, выделяю под них память.

Что бы использовать «MySqlConnection» нужно подключить адаптер в заголовочном файле прописав «using MySql.Data.MySqlClient;», это подключит библиотеку, которая содержит все нужные мне дополнительные классы.

«MySqlCommand command» объект, который позволяет прописать команду для выборки данных из базы данных, он принимает SQL запросы. «db.getConnection()» нужен, что бы указать к какой базе данных мы подключаемся.

«command.Parameters.Add» позволяет мне записать параметры в sql запрос.

Адаптер может выполнить команду и сформировать в себе массив данных, который потом он заполняет в объект Table, в котором мы можем посчитать колл-во элементов.

IF/ELSE проводит отбор. Если количество юзеров больше чем 0, то значит, человек с таким логином и паролем существует и ему можно открыть доступ в программу.

Вывод данных из БД в dataGridView на примере окна useService:

После открытия окна, программы вызывает функцию «LoadFullData();»

В этой функции я решил использовать другой способ подключения к БД, без использования класса BD

Ключевая разница только в форме представления кода, « MySqlCommand command » так же принимает строку кода и ссылку на БД, но открывать доступ к БД приходится отдельно.

Здесь я использую « MySqlDataReader », он позволяет прочитать данные из БД. И LIST с массивом string.

Решил использовать именно цикл while, так как метод .Read возвращает true при успешной работе. А значит не нужно приписывать количество итераций

Цикл foreach будет вносить данные в таблицу, пока он не закончатся.

Все остальные участки кода отличаются условиями if/else, SQL запросами, и количеством использованных функций в конкретном окне. Так как я несколько раз создавал функции, в которые помещал весь код из кнопки. Следственно кнопка активировала функцию и выполняла код. Но принципиально весь остальной код не отличаются методом решения задач, поэтому я не буду помещать его в исследовательскую или проектно-технологическую части. Напишу сюда полное описание функций код одной из форм. В качестве примера возьму форму регистрации.

```

public partial class RegistrForm : Form
{
    public RegistrForm()
    {
        private void closeButton_Click(object sender, EventArgs e)
        private void closeButton_MouseEnter(object sender, EventArgs e)
        private void closeButton_MouseLeave(object sender, EventArgs e)
        private void topPanel_MouseMove(object sender, MouseEventArgs e)
        private void topPanel_MouseDown(object sender, MouseEventArgs
e)
        private void mainPanel_MouseMove(object sender, MouseEventArgs
e)
        private void mainPanel_MouseDown(object sender, MouseEventArgs
e)
        private void customerNameField_MouseEnter(object sender,
EventArgs e)
        private void customerNameField_MouseLeave(object sender,
EventArgs e)
        private void customerSurnameField_MouseEnter(object sender,
EventArgs e)
        private void customerSurnameField_MouseLeave(object sender,
EventArgs e)
        private void customerPhoneField_MouseEnter(object sender,
EventArgs e)
        private void customerPhoneField_MouseLeave(object sender,
EventArgs e)
        private void buttonRegistr_Click(object sender, EventArgs e)
        public Boolean isUserExist()
        private void back_Click(object sender, EventArgs e)
        private void buttonRegistr_ClickAktion()
    }
}

```

Функции: closeButton_Click, closeButton_MouseEnter, closeButton_MouseLeave, topPanel_MouseMove, topPanel_MouseDown, mainPanel_MouseMove, back_Click и mainPanel_MouseDown я уже описал выше, поэтому описывать их второй раз не буду.

Функции customerNameField_MouseEnter и customerNameField_MouseLeave. При старте окна в поле имени записано «Введите имя», customerNameField_MouseEnter отвечает за то, что при наведении мыши на поле содержащее текст «Введите имя», поле NameField удаляло в себе текст. Функция customerNameField_MouseLeave отвечает за то, что если это поля осталось пустым, то после того как курсор покинет его, поле NameField опять заполняло себя текстом «Введите имя».

Аналогично работают функции для фамилии и телефона.

Функция `buttonRegistr_Click` активируется после нажатия на кнопку «зарегистрироваться», она запускает SQL запрос, зарегистрирован ли уже клиент с подобным номером телефона. Запрос : "SELECT * FROM `customer` WHERE `phone` = @cP "

Если такого пользователя нет, то она активирует другую функцию `buttonRegistr_ClickAktion()`. Эта функция соединяется с базой данных используя класс `DB` и при помощи SQL запроса добавляет запись в базу данных . Запрос: "INSERT INTO `customer`(`name`, `surname`, `phone`, `registrationDate`, `currentCard`) VALUES(@name, @surname, @phone, CURDATE(), 0)"

Имя, фамилия и телефон передаются при помощи параметров класса `MySQLCommand`.

3.5.Разработка внешнего вида и дизайна приложения

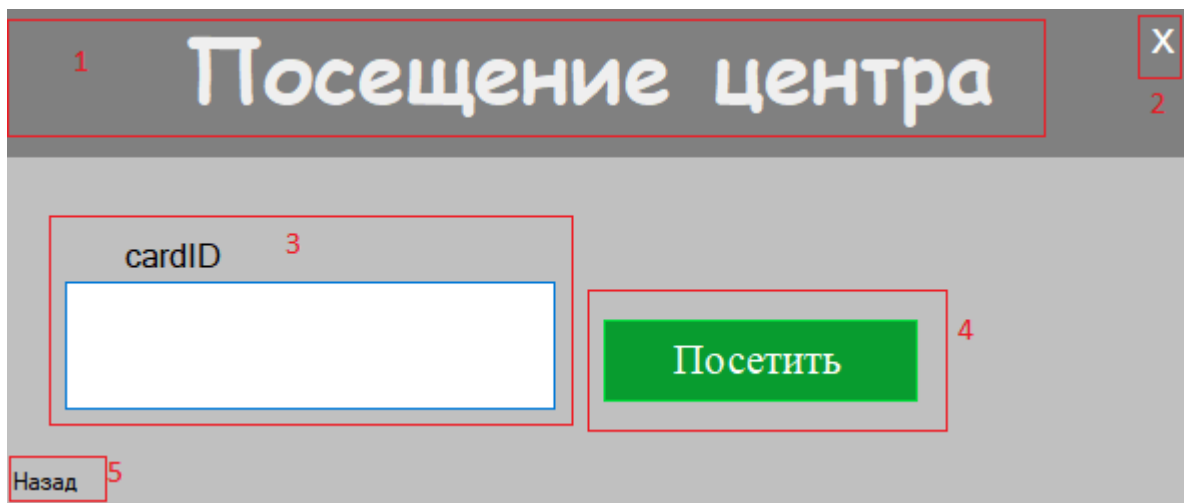


Рисунок 6- Макет окна «визит в центр».

Это окно служит в качестве симуляции работы устройства, которое должно вводит информацию после контакта с картой клиента. Так как это симуляция, она должна быть выполнена с минимальным количеством деталей.

Здесь есть несколько сегментов

1. `TopPanel` С названием окна
2. Крестик для выключения программы

3. Поле для ввода номера карты(в реальности оно заполняется автоматически после контакта карты и специального сканера)
4. Кнопка регистрации посещения.
5. Кнопка «назад», для перехода в главное меню

Рисунок 7 - Макет окна в котором списывают услуги

Окно списка услуг является одним из самых сложных. Оно состоит из:

1. TopPanel С названием окна
2. Крестик для выключения программы
3. DataGridView для вывода информации из базы данных
4. Блок для поиска услуг по карте. Состоит из надписи, поля для ввода номера карты и кнопки, которая активирует поиск
5. Блок для использования услуги. Состоит из поясняющих надписей, полей для ввода номера карты и id услуги. И кнопки, активирующей код.
6. Кнопка «назад», для перехода в главное меню.

4.ПРОЕКТНО-ТЕХНОЛОГИЧЕСКАЯ ЧАСТЬ

4.1. Порядок развёртывания системы

Что бы программа начала работу запустите её через .exe файл, нужно понимать, что она не будет корректно работать в том случае, если виртуальный сервер выключен.

4.2. Краткое руководство пользователя

После запуска программы введите свой логин и пароль, после этого вы попадёте в главное меню.

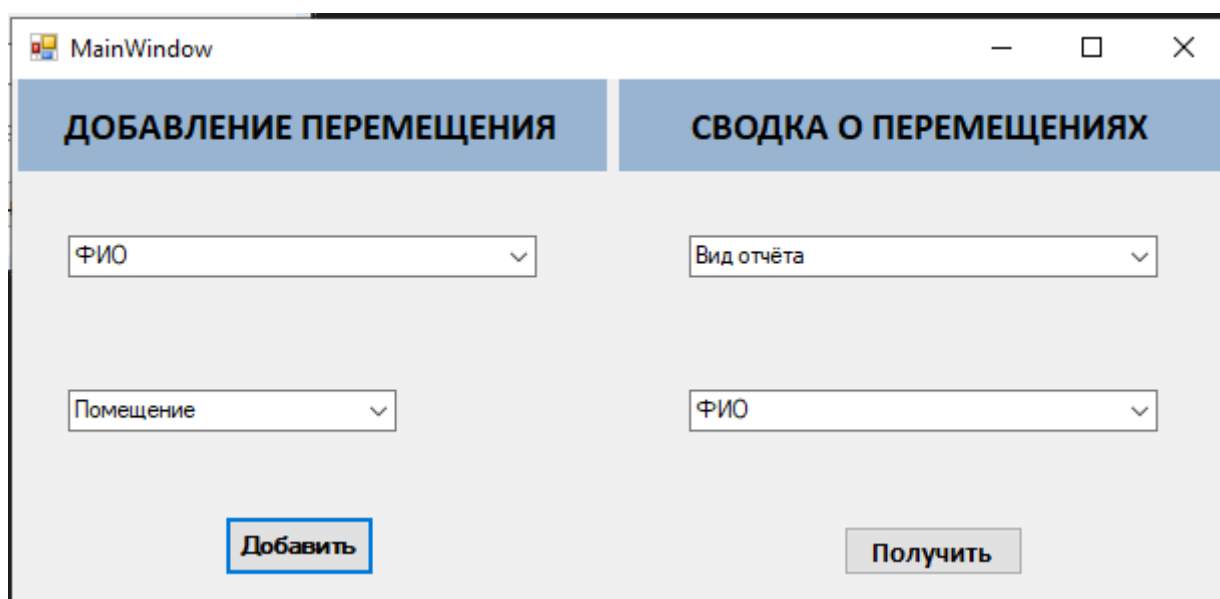


Рисунок 11 - Главное меню

Для добавления перемещения работника необходимо в левой области «ДОБАВЛЕНИЕ ПЕРЕМЕЩЕНИЯ» выбрать полное имя нужного сотрудника в поле «ФИО» и зону в поле «Помещение», в которую он должен переместиться.

Для просмотра сводок необходимо в правой области «СВОДКА О ПЕРЕМЕЩЕНИЯХ» выбрать вид желаемого отчёта в поле «Вид отчёта» и

полное имя сотрудника в поле «ФИО» и нажать кнопку «Получить». После нажатия кнопки может быть открыто одно из трёх окон: «Маршрут сотрудника», «Пребывание сотрудника в зонах», «Переработка сотрудника».



The screenshot shows a window titled 'member_way' with a blue header bar containing the text 'МАРШРУТ СОТРУДНИКА'. Below the header is a table with 7 columns: an empty column, 'Дата', 'ФИО', 'Зона', 'Время входа', 'Время выхода', and 'Длительность'. The table contains 8 rows of data for the employee 'Вася Пупкин Иванович' on various dates in 2021. The first row is highlighted in blue. Below the table is a large grey rectangular area.

| | Дата | ФИО | Зона | Время входа | Время выхода | Длительность |
|---|------------|----------------------|--------------------|-------------|--------------|--------------|
| ▶ | 09.03.2021 | Вася Пупкин Иванович | [ВНЕ РАБОЧЕЙ ЗОНЫ] | 21:20:40 | 21:36:03 | 00:15:23 |
| | 09.03.2021 | Вася Пупкин Иванович | Цех 1 | 21:36:03 | 21:43:18 | 00:07:15 |
| | 09.03.2021 | Вася Пупкин Иванович | [ВНЕ РАБОЧЕЙ ЗОНЫ] | 21:43:18 | 21:48:42 | 00:05:24 |
| | 09.03.2021 | Вася Пупкин Иванович | Цех 1 | 21:48:42 | 13:18:17 | -08:30:25 |
| | 10.03.2021 | Вася Пупкин Иванович | [ВНЕ РАБОЧЕЙ ЗОНЫ] | 13:26:00 | 13:27:20 | 00:01:20 |
| | 10.03.2021 | Вася Пупкин Иванович | [ВНЕ РАБОЧЕЙ ЗОНЫ] | 13:27:21 | 13:32:45 | 00:05:24 |
| | 10.03.2021 | Вася Пупкин Иванович | Сортировка | 13:32:45 | 14:02:57 | 00:30:12 |
| * | | | | | | |

Рисунок 12 – Окно «Маршрут сотрудника»

Данное окно выводит историю перемещения выбранного сотрудника между зонами предприятия. На данный момент в рамках курсовой работы в БД находится малое количество записей, и история выводится за всё время существования БД. Но это легко можно исправить при необходимости.

| | Зона | Общая длительность |
|---|--------------------|--------------------|
| ▶ | [ВНЕ РАБОЧЕЙ ЗОНЫ] | 00:27:31 |
| | Сортировка | 00:30:12 |
| | Цех 1 | 15:36:50 |
| * | | |

Рисунок 13 – Окно «Пребывание сотрудника в зонах»

Данное окно выводит количество времени в формате «чч:мм:сс», которое сотрудник пробыл в посещённых им зонах. На данный момент в рамках курсовой работы период счёта ограничен одной неделей.

| | ФИО | План, чч | Фактически, чч | Переработка, чч |
|---|----------------------|----------|----------------|-----------------|
| ▶ | Вася Пупкин Иванович | 8,0000 | 16,5758 | 8,5758 |
| * | | | | |

Рисунок 14 – Форма «Переработка сотрудника»

Данное окно выводит: ФИО сотрудника; количество часов, которое работник должен отработать для выполнения плана; количество фактически отработанных часов на момент запроса сводки; количество часов, которые были отработаны сверх нормы (переработка) на момент запроса сводки. Если

столбец переработки показывает отрицательное число, значит выбранный сотрудник ещё не выполнил норму. В рамках курсовой работы подсчёт ведётся не за рабочий день, а за всё время.

4.3. Возможности улучшений

Разработанное приложение можно улучшить многими способами.

Сделать более приятный интерфейс.

Добавить дополнительную защиту и более подробные уведомления, когда работник пытается внести не корректные данные в таблицу.

Добавить ограничение для каждого сотрудника на посещение определённых зон, в которых ему быть не положено.

Добавить более сложную структуру зон.

Решить потенциальную проблему утечки памяти при помощи использования паттерна Singleton.

ЗАКЛЮЧЕНИЕ

В ходе выполнения курсовой работы была разработана программа выполняющая следующий функционал:

- Регистрация клиента
- Привязка карты
- Учёт информации о посещениях центра
- Регистрацию личной карты клиента
- Фиксация продажи и использования дополнительных услуг центра
- Показывает расписание групповых занятий центра

В ходе выполнения я более детально изучил Windows Forms и научился связывать их в реляционными базами данных на языке SQL

СПИСОК ИСПОЛЬЗОВАННОЙ ЛИТЕРАТУРЫ

1. Антон А. Н. Информатика и информационные технологии. 2001
2. Диго С. М. Создание баз данных в среде СУБД Учебное пособие. 2001.
3. Туманов В.Е. Основы проектирования реляционных баз данных. 2011
4. Интернет Энциклопедия
<https://ru.wikipedia.org/wiki/Фитнес-клуб>
5. Илюшечкин В. М. Основы использования и проектирования баз данных. 2010