|  |  |
| --- | --- |
| **Gerb-BMSTU_01** | **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 сотрудников фитнесс центра***

по дисциплине ***Базы данных***

Студент гр. ИУК5.Б-51 \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ ( Антюхов А.А. )

(подпись) (Ф.И.О.)

Руководитель \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ ( Кириллов В.Ю. )

(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 ***Базы Данных***

Студент\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_Антюхов А.А., ИУК5.Б-51\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(фамилия, инициалы, индекс группы)

Руководитель\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Кириллов В. Ю. \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(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 учёта и 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

[**1.ТЕХНИЧЕСКОЕ ЗАДАНИЕ** 3](#_Toc59721922)

[**1.1 Общие сведения** 3](#_Toc59721923)

[**1.1.1 Наименование** 3](#_Toc59721924)

[**1.1.2 Исполнитель** 3](#_Toc59721925)

[**1.1.3 Перечень нормативно-технических документов, методических материалов, использованных при разработке ТЗ:** 3](#_Toc59721926)

[**1.1.4 Плановые сроки по созданию приложения** 3](#_Toc59721927)

[**1.2 Назначение и цели создания системы** 3](#_Toc59721928)

[**1.2.1 Назначение системы** 3](#_Toc59721929)

[**1.2.2 Цели создания системы** 3](#_Toc59721930)

[**1.3 Характеристики объекта автоматизации** 3](#_Toc59721931)

[**1.4 Требования к системе** 4](#_Toc59721932)

[**1.4.1 Требования к системе в целом** 4](#_Toc59721933)

[**1.4.2 Требования к надежности** 4](#_Toc59721934)

[**1.4.3 Требования к функциям, выполняемым системой** 4](#_Toc59721935)

[**1.4.4 Требования к информационному обеспечению системы** 4](#_Toc59721936)

[**1.4.5 Требования к методическому обеспечению** 4](#_Toc59721937)

[**1.5 Состав и содержание работ по созданию(развитию) системы** 5](#_Toc59721938)

[**1.6 Порядок контроля и приемки системы** 5](#_Toc59721939)

[**1.6.1 Методы испытания системы** 5](#_Toc59721940)

[**1.6.2 Общие требования к приемке работ по стадиям** 5](#_Toc59721941)

[**1.6.3 Статус приемочной комиссии** 5](#_Toc59721942)

[**1.7 Требования к составу и содержанию работ по подготовке объекта автоматизации к вводу системы в действие** 5](#_Toc59721943)

[**1.8 Требования к документированию** 6](#_Toc59721944)

[**1.9 Источники разработки** 6](#_Toc59721945)

[**2.НАУЧНО-ИССЛЕДОВАТЕЛЬСКАЯ ЧАСТЬ** 6](#_Toc59721946)

[**2.1. Постановка задачи** 6](#_Toc59721947)

[**2.2 Описание предметной области** 7](#_Toc59721948)

[**2.2.1 Общие сведения** 7](#_Toc59721949)

[**2.2.2 Анализ аналогов.** 8](#_Toc59721950)

[**2.2.3 Концептуальная модель данных** 10](#_Toc59721951)

[**2.2.4 Перечень реализуемых функций.** 12](#_Toc59721952)

[**2.3 Сценарий использования** 12](#_Toc59721953)

[**2.4 Задача автоматизации** 12](#_Toc59721954)

[**2.5 Перечень задач, подлежащих решению в процессе разработки** 12](#_Toc59721955)

[**2.6 Обоснование выбора платформы для разработки** 13](#_Toc59721956)

[**3.ПРОЕКТНО-КОНСТРУКТОРСКАЯ ЧАСТЬ** 14](#_Toc59721957)

[**3.1. Разработка структуры приложения** 14](#_Toc59721958)

[**3.2. Разработка алгоритмов обработки информации** 15](#_Toc59721959)

[**3.3. Логическая схема базы данных** 16](#_Toc59721960)

[**3.4.Описание физической модели базы данных** 17](#_Toc59721961)

[**3.5.Разработка архитектуры приложения** 20](#_Toc59721962)

[**4.ПРОЕКТНО-ТЕХНОЛОГИЧЕСКАЯ ЧАСТЬ** 29](#_Toc59721963)

[**4.1. Порядок развёртывания системы** 29](#_Toc59721964)

[**4.2. Краткое руководство пользователя** 29](#_Toc59721965)

[**4.3. Возможности улучшений** 35](#_Toc59721966)

[**ЗАКЛЮЧЕНИЕ** 36](#_Toc59721967)

[**СПИСОК ИСПОЛЬОВАННОЙ ЛИТЕРАТУРЫ** 36](#_Toc59721968)

# **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 загружается 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 местоположении персонала.

### **1.2.2 Цели создания системы**

Система должна хранить и обновлять данные о местоположении персонала.

## **1.3 Характеристики объекта автоматизации**

Объект автоматизации – обновление 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 окончанию 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 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

### **2.2.3 Концептуальная модель данных**

В схеме присутствует сущность «Уполномоченный сотрудник». Она нужна для того, что бы можно было обеспечить минимальный уровень защиты от не санкционированного доступа. Она позволяет связаться с базой данных и проверить присутствует ли в ней работник с нужным логином и паролем, который, собственно, и будет пользоваться приложением.

Так же есть сущность «Работник». Она нужна для хранения имён работников, которые будут перемещаться по предприятию. Предполагается, что работников с одинаковыми именами нет.

Сущность «Зона» нужна для хранения зон, между которыми будут перемещаться сотрудники. Предполагается, что зон с одинаковыми названиями нет.

Сущность «Перемещение сотрудника» представляет наибольшее значение. Она нужна для хранения данных о перемещении сотрудников между зонами.

Поля id сущностей «Работник» и «Зона» являются первичными ключами для данных сущностей.

Поля «id работника» и «id зоны» сущности «Перемещения сотрудников» являются внешними ключами, в данную сущность нельзя добавить работника или зону, которых не существует.

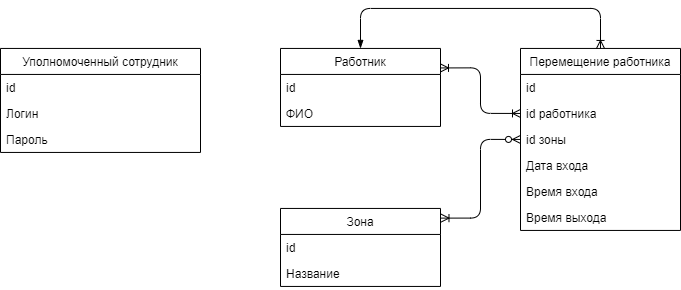


Рисунок 4 – Схема модели данных

### **2.2.4 Перечень реализуемых функций.**

Хранение данных о работниках предприятия.

Хранения данных о зонах предприятия.

Ввод и хранение данных о перемещениях работников предприятия между зонами. Будут учитываться дата и время входа и выхода.

## **2.3 Сценарий использования**



Рисунок 5 – Сценарий использования для работника.

## **2.4 Задача автоматизации**

Задачей автоматизации является упрощение процесса ведения учёта местоположения работников, 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 информацию 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 аналоги 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 `employee` WHERE `login` = @uL AND `pass` = @uP ", db.getConnection());

## **3.3. Логическая схема базы данных**

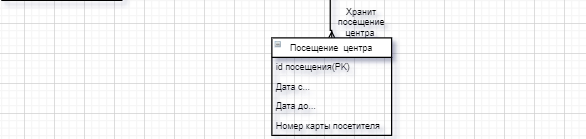
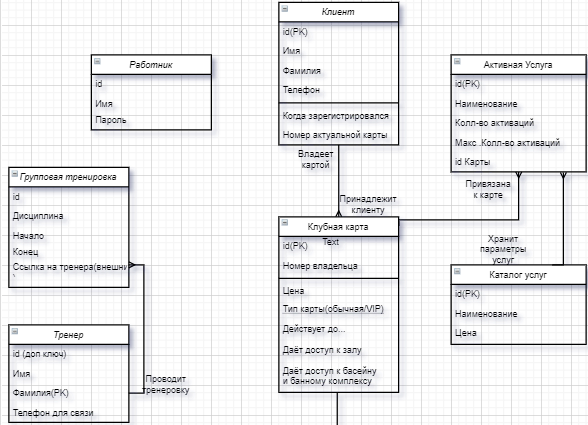


Рисунок 5 – Схема логической модели данных

## **3.4.Описание физической модели базы данных**

Опишем сущности, использованные в базе данных.

`employee`- Сущность для работника. Она имеет 3 поля.

* ID(Первичный ключ) тип данных int
* Логин тип данных varchar
* Пароль тип данных varchar

`customer`-Сущность клиента. В ней присутствует 6 полей

* ID(Первичный ключ) тип данных int
* Имя тип данных varchar(100)
* Фамилия тип данных varchar(100)
* Телефон тип данных varchar(11)
* Дата регистрации тип данных DATE
* И одну привязанную карту (для внешнего ключа)

Так как не имеет значения точное время регистрации клиента, я использовал тип данных DATA, вместо DATTIME.

Поле currentСard связанно с таблицей card и в него нельзя записать значение Id карты, которого нет в таблице card.

Для телефонного номера использовал тип данных varchar(11), так как ограничение в 11 символов не позволит записать туда не нужную информацию, при том что в целом удобнее работать с номером телефона, как с массивом символов.

`card`-Сущность карты имеет 7 полей

* ID(Первичный ключ) тип данных int
* Телефон владельца тип данных varchar(11)
* Цена приобретения тип данных int
* Тип карты тип данных varchar(100)
* Актуальность до… тип данных data
* Доступ в зал тип данных varchar(100)
* Доступ в бассейн тип данных varchar(100)

Не смотря на то, что существует тип данных MONEY, я использовал тип данных int, что бы избежать дополнительных конвертаций.

Тип карты varchar(100), так как в будущем можно добавить дополнительные виды карт, к примеру карту только для бассейна, без доступа к залу

Актуальность карты хранится в DATA,так как услуги салон предоставляет ровно год и полный последний день, а значит время нам не важно.

Несмотря на то, что сейчас доступно только два параметра для Доступа в зал и в бассейн. Yes/No, возможно будет придуманы и другие параметры для этих атрибутов, поэтому я не использовал BOOL

`serviceсatalog`-Сущность для хранения каталога услуг . Имеет 3 поля

* ID(Первичный ключ) тип данных int
* Название услуги тип данных varchar
* Цена тип данных int

Причина использования типа int вместо MONEY, такая же как в таблице card, сократить число преобразований.

active\_services- Таблица, куда помещаются все купленные услуги. Имеет 5 столбцов:

* ID(Первичный ключ) тип данных int
* Название услуги тип данных varchar
* Сколько раз использовали тип данных int
* Сколько раз можно использовать тип данных int
* ID карты клиента тип данных int

Логично, что раз мы используем для хранения названий в каталоге тип данных varchar, то здесь мы тоже будем использовать такой же тип данных.

ID карты клиента, хранит номер конкретной карты клиента, к которой привязана услуга имеющая фиксированное количество использований.

`visit` Таблица для хранения записей о визитах клиента

* ID(Первичный ключ) тип данных int
* Время входа тип данных datetime
* Время выхода тип данных datetime
* Номер карты клиента тип данных int

Так как нужно знать точное время входа и выхода, использован тип данных datetime.

Trainer таблица имеющая 4 поля и два ключа

* ID(Вторичный ключ) тип данных int
* Имя тип данных varchar
* Фамилия(Первичный ключ) тип данных varchar
* Телефон тип данных varchar(11)

В теории в центре не будет работать два работника с одной фамилией и одного ключа в поле «Фамилия достаточно», но так как в будущем возможно появление тренеров однофамильцев в таблице присутствует поле id тренера, что бы можно было сделать составной ключ.

«group\_training» Таблица имеющая 5 полей

* ID(Первичный ключ) тип данных int
* Дисциплину тренировки тип данных varchar
* Время начала тип данных datetime
* Время конца тип данных datetime
* Фамилия тренера (внешний ключ) тип данных varchar

В «Фамилия тренера» нельзя поместить значения, отсутствующее в таблице тренеров.

Время начала и конца datetime, что бы учитывать точные часы и минуты.

Первоначально планировалось довить поле «День недели», но что бы не нарушить принцип нормализации, день недели вычисляется отдельно при помощи соответствующего запроса.

## **3.5.Разработка архитектуры приложения**

Имеется несколько Windows Form, переход между ними осуществляется при помощи нажатий 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 входа активируется код, вызывающий такой запрос

"SELECT \* FROM `employee` WHERE `login` = @uL AND `pass` = @uP "

@uL и @uP поля, принимающие в себя логин и пароль из полей loginField.Text и passField.Text соответственно.

После ввода правильного пароля мы попадаем на главную форму.



Рисунок 7- Главная форма

Не будем на ней останавливаться, так как в ней отсутствует SQL код.



Рисунок 8- Окно регистрации клиента

После нажатия на кнопку «зарегистрировать» вызовется такой запрос

"INSERT INTO `customer`( `name`, `surname`, `phone`, `registrationDate`, `currentСard`) VALUES(@name, @surname, @phone, CURDATE(), 0)"

Поле `currentСard` присваивается 0, так как карта привязывается отдельно.



Рисунок 9- Регистрация и привязка клиента

При нажатии на кнопку «Привязать карту». Запускается 3 SQL запроса.

SELECT phone FROM `customer` WHERE customer.phone= @phone

Проверяет существование указанного номера телефона.

INSERT INTO `card`(`phone`, `price`, `card\_type`, `validity`, `access\_gum`, `access\_swimming`) VALUES(@phone , @price ,@card\_type ,CURDATE(), @access\_gum, @access\_swimming); UPDATE `card` SET `validity`= DATE\_ADD(`validity`, INTERVAL 1 year) WHERE [card.phone=@phone](mailto:card.phone=@phone);

В две строчки вносит данные в таблицу карт. Сначала в поле `validity` записывается текущая дата, поле чего при помощи UPDATE она увеличивается на год.

" UPDATE `customer` SET `currentСard`= (SELECT card.id FROM card WHERE card.phone = @phone111) WHERE customer.phone=@phone111 "

Самый сложный запрос, он вложенный и изменяет данные в поле «Привязанная карта» в сущности клиента, на id только что созданной карты.



Рисунок 10- Визит в центр.

Это окно визита. Первым делом проверяется, если ли гость в центре сейчас. Проверка вынесена в отдельную функцию

Запрос проверки:

"SELECT `out\_centr` FROM `visit` WHERE `card\_id` = @curdId AND `out\_centr` ='1999-12-30 00:00:00' "

Время по дефолту всегда устанавливается на 30 декабря 1999 года и если такая запись присутствует в таблице, то гость уже в центре.

Если такая запись присутствует, то её заменяют таким запросом:

"UPDATE `visit` SET `out\_centr`=CURRENT\_TIMESTAMP() WHERE `card\_id` = @curdId "

Заменяя 30 декабря 1999 года на точное время выхода.

Если же функция вернула false, то программавызывает запрос, создающий запись о входе клиента:

"INSERT INTO `visit`(`in\_centr`, `out\_centr`, `card\_id`) VALUES (CURRENT\_TIMESTAMP(), '1999-12-30' , @curdId )"



Рисунок 11 - Каталог услуг.

Данные в dataGridView выводятся при помощи самого обычного селекта

SELECT \* FROM `serviceсatalog` ORDER BY id

Услуга из каталога в список услуг добавляется таким запросом

"INSERT INTO `active\_services`(`name`, `used`, `can be used`, `idCardCustomer`) VALUES (@name, '0', @maxuse, @idCardCustomer) "



Рисунок 12 - Окно, в котором списывают услуги

Когда клиент хочет использовать услугу, работник открывает соответствующее окно. В нём он может выполнить поиск по услуге или списать её.

Запрос на поиск.

"SELECT `id`, `name`, `used`, `can be used`, `idCardCustomer` FROM `active\_services` WHERE idCardCustomer = " + idCardCudtomer + " GROUP BY active\_services.id"

Запрос на списание

"UPDATE `active\_services` SET `used`= `used`+1 WHERE idCardCustomer=" + idCardCudtomer + " AND id ="+ idServis



Рисунок 13 - Расписание тренировок

Вывод данных в dataGridView отлсчается от прошлого вывода данных, так как ещё выводит день недели, когда проводят тренировку. В начале я планировал создать ещё один столбец для дня недели, но потом понял чтоэто грубо нарушит правила нормализации

"SELECT `id`, `discipline`, `start`, `finish`, `trainer`, DAYname(start) FROM `group\_training`"

Нажатие «Поиск» вызывает команду:

"SELECT `id`, `discipline`, `start`, `finish`, `trainer`, DAYname(start) FROM `group\_training` WHERE discipline = '" + disciplineSearchName + "' ORDER BY id"

Нужно учитывать что перед каждым выводом данных в dataGridView, она отчищается при помощи метода dataGridView1.Rows.Clear();

# **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 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 14- Главное меню

В нём выберите интересующий вас пункт.

«Регистрация клиента» позволит внести данные о новом клиенте клуба

«Регистрация карты» осуществит привязку выданной карты к клиенту

«Посещение центра» регистрирует вход-выход клиентов центра.

«Перечень услуг» позволит увидеть весь список предоставляемых центром услуг и приобрести услуги клиенту

Окно «Использовать услугу» позволит списать услугу, ранее купленную клиентом

Окно «Расписание групповых» занятий покажет список ближайших тренировок и позволит найти тренировку по интересующей клиента дисциплине.



Рисунок 15- Окно входа

В это окно пользователь вводит данные для входа.



Рисунок 16- Главное меню

Главное меню позволяет выбрать нужное окно.



Рисунок 17- Окно регистрации клиента

Введите в поля имя, фамилию и телефон нового клиент, после чего нажмите кнопку «Зарегистрировать». Если клиент с таким телефонным номером не был зарегистрирован раньше, программа внесёт его в список клиентов.



Рисунок 18- Регистрация и привязка клиента

Поле регистрации клиента нужно привязать к клиенту карту. Впишите в поле «Номер телефона», номер телефона клиента, выберите тип его карты и нажмите «Привязать карту».



Рисунок 19- окно визита в центр.

Если считывающее карты устройство не работает, в этом окне вы можете вписать номер карты клиента, программа сама определит входит клиент или наоборот, покидает центр



Рисунок 20 - Каталог услуг.

При запуске этого окна перед вами появляется весь перечень услуг центра. Если считывающее карты устройство не работает, вручную введите номер карты клиента, номер услуги и количество приобретённых клиентом услуг выбранного типа.



Рисунок 21 - Окно, в котором списывают услуги

Окно позволяет и использовать услугу введите id услуги и номер карты клиента, после чего нажмите кнопку «Использовать услугу».



Рисунок 22 - Расписание тренировок

Эта вкладка позволит вам увидеть всё расписание групповых занятий и при необходимости найти ближайшие групповые занятия по нужной дисциплине.

## **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.

Упростить использование услуг.

Добавить поиск по тренеру и поиск по дням недели для групповых тренировок

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 ИСПОЛЬОВАННОЙ 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 пректирования баз данных. 2010