|  |  |
| --- | --- |
| **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:**

***Пул потоков, реализованный на основе критических секций с использованием WinAPI.***

по дисциплине ***Системное программирование***

Студент гр. СОИ.Б-41 \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ ( Мустафин Д. И. )

(подпись) (Ф.И.О.)

Руководитель \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ ( Фролов П.В. )

(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\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_Мусатфин Д.И, 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 сотрудников фитнесс центра***

***2. Техническое задание***

*Разработать систему, способную хранить и обрабатывать данные о клиентах, тренерах, «программах» и других услугах, которые предоставляет фитнес центр.*

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

***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](#_Toc59562551)

[**1.1 Общие сведения** 3](#_Toc59562552)

[**1.1.1 Наименование** 3](#_Toc59562553)

[**1.1.2 Исполнитель** 3](#_Toc59562554)

[**1.1.3 Перечень нормативно-технических документов, методических материалов, использованных при разработке ТЗ:** 3](#_Toc59562555)

[**1.1.4 Плановые сроки по созданию приложения** 3](#_Toc59562556)

[**1.2 Назначение и цели создания системы** 3](#_Toc59562557)

[**1.2.1 Назначение системы** 3](#_Toc59562558)

[**1.2.2 Цели создания системы** 3](#_Toc59562559)

[**1.3 Требования к системе** 3](#_Toc59562560)

[**1.3.1 Требования к системе в целом** 4](#_Toc59562561)

[**1.4.2 Требования к надежности** 4](#_Toc59562562)

[**1.4.3 Требования к функциям, выполняемым системой** 4](#_Toc59562563)

[**1.4.4 Требования к информационному обеспечению системы** 4](#_Toc59562564)

[**1.4.5 Требования к методическому обеспечению** 4](#_Toc59562565)

[**1.5 Состав и содержание работ по созданию(развитию) системы** 5](#_Toc59562566)

[**1.6 Порядок контроля и приемки системы** 5](#_Toc59562567)

[**1.6.1 Методы испытания системы** 5](#_Toc59562568)

[**1.6.2 Общие требования к приемке работ по стадиям** 5](#_Toc59562569)

[**1.6.3 Статус приемочной комиссии** 5](#_Toc59562570)

[**2.НАУЧНО-ИССЛЕДОВАТЕЛЬСКАЯ ЧАСТЬ** 6](#_Toc59562571)

[**2.1. Постановка задачи** 6](#_Toc59562572)

[**2.2 Описание предметной области** 6](#_Toc59562573)

[**2.2.1 Общие сведения** 6](#_Toc59562574)

[**2.2.2 Анализ аналогов.** 8](#_Toc59562575)

[**2.2.3 Концептуальная модель данных** 11](#_Toc59562576)

[**2.2.4 Перечень реализуемых функций.** 13](#_Toc59562577)

[**2.3 Сценарий использования** 13](#_Toc59562578)

[**2.4 Задача автоматизации** 14](#_Toc59562579)

[**2.5 Перечень задач, подлежащих решению в процессе разработки** 14](#_Toc59562580)

[**2.6 Обоснование выбора платформы для разработки** 14](#_Toc59562581)

[**3.ПРОЕКТНО-КОНСТРУКТОРСКАЯ ЧАСТЬ** 15](#_Toc59562582)

[**3.1. Разработка структуры приложения** 15](#_Toc59562583)

[**3.2. Разработка алгоритмов обработки информации** 16](#_Toc59562584)

[**3.3. Логическая схема базы данных** 18](#_Toc59562585)

[**3.4.Описание физической модели базы данных** 19](#_Toc59562586)

[**3.5.Разработка архитектуры приложения** 22](#_Toc59562587)

[**4.ПРОЕКТНО-ТЕХНОЛОГИЧЕСКАЯ ЧАСТЬ** 30](#_Toc59562588)

[4.1. Порядок развёртывания системы 30](#_Toc59562589)

[4.2. Краткое руководства пользователя 30](#_Toc59562590)

[4.3. Возможности улучшений 31](#_Toc59562591)

[ЗАКЛЮЧЕНИЕ 32](#_Toc59562592)

[СПИСОК ИСПОЛЬОВАННОЙ ЛИТЕРАТУРЫ 33](#_Toc59562593)

# **1.ТЕХНИЧЕСКОЕ ЗАДАНИЕ**

## **1.1 Общие сведения**

### **1.1.1 Наименование**

Приложение для сотрудников фитнесс центра “Упорство”

### **1.1.2 Исполнитель**

Студент КФ МГТУ Им. Н.Э. Баумана

Мустафин Денис Игоревич гр. СОИ.Б-51

### **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/DenisTR-KL/Kurasch-po-BD-2020, 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 о клиентах, услугах и работниках организации.

### **1.2.2 Цели создания системы**

Система должна хранить и обновлять данные о клиентах, тренерах, «программах» и других услугах, которые предоставляет фитнес центр.

## **1.3 Требования к системе**

### **1.3.1 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 центра.

### **1.4.3 Требования к функциям, выполняемым системой**

Программа должна обеспечивать реализацию следующих функций:

1. Завести клиента
2. Ввод/вывод информации о посещениях центра ирасписании
3. Регистрацию личной карты клиента
4. Перечень и подсчет предоставленных услуг.

### **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 документированию**

-ТЗ по проектируемой системе

-Расчетно-пояснительная записка

-Презентация Microsoft PowerPoint

# **2.НАУЧНО-ИССЛЕДОВАТЕЛЬСКАЯ ЧАСТЬ**

## **2.1. Постановка задачи**

Требуется разработать программу, для сотрудников фитнес центра

## **2.2 Описание предметной области**

### **2.2.1 Общие сведения**

Фитнес центр - организация сочетающая в себе спортивный зал, бассейн, кардио-зону, залы аэробики, сайкл студию, студии единоборств, студии йоги, групповые и танцевальные программы, а также аквааэробику.

Имеет площади для проведения разнообразных оздоровительных и фитнес тренировок, с использованием силовых упражнений и специального оборудования для кардио-тренировок.

Может иметь уличные площади, для проведений личных или групповых занятий на открытом воздухе, оказывать дополнительные услуги в виде солярия, услуг массажа, бани, сауны, хамама, салона красоты и тп.

Организация обеспечивает свободный доступ физических лиц к оборудованным площадям и тренажерам на основе разового платежа или на основании участия в членской системе(приобретение клубной карты).

В фитнес клубе могут проводить соревнования. Закрытые, только для участников или и городские. В некоторых случаях, соревнования большего масштаба.

Фитнес клуб за определённую плату может предоставить инструктора, который за определённую плату составит личную программу тренировок для оздоровления/укрепления организма.

На основании всего вышеизложенного можно сделать вывод, что клуб работает с большими массивами данных: Виды клубных программ, список участников клубных программ, список клиентов, список работающих в нём тренеров, график личных тренировок(для каждого тренера), график «дежурного тренера»(тренер, ответственный за тренажёрный зал и за то, что бы посетители центра не травмировали себя в следствии не правильного выполнения упражнений), график групповых программ(сайкл студия, групповые занятия: йога, боевые искусства, обычные групповые занятия), список прочего персонала.

Все эти данные нужно как то обрабатывать, для этого любой фитнес центр имеет специальную программу, в которой хранится большая часть этих данных.

Потребитель услуг центра и большая часть работников не видят это приложение и не пользуются им, этим поэтому можно сэкономить на красивом оформлении и выполнить его в минималистичном стиле, оставив только важный функционал, нужный для комфортной работы с программой.

В рамках курсовой работы приложение будет иметь функционал создания членов клуба, клубных программ. Содержать в себе расписание групповых занятий. Окно, в котором можно узнать цены на все предоставляемые услуги и клубные программы, что бы быстро отвечать на частые вопросы посетителей, что и сколько стоит.

### **2.2.2 Анализ аналогов.**

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

1. «WS. Фитнес»- Сервис для учёта клиентов и финансов для фитнес клубов.

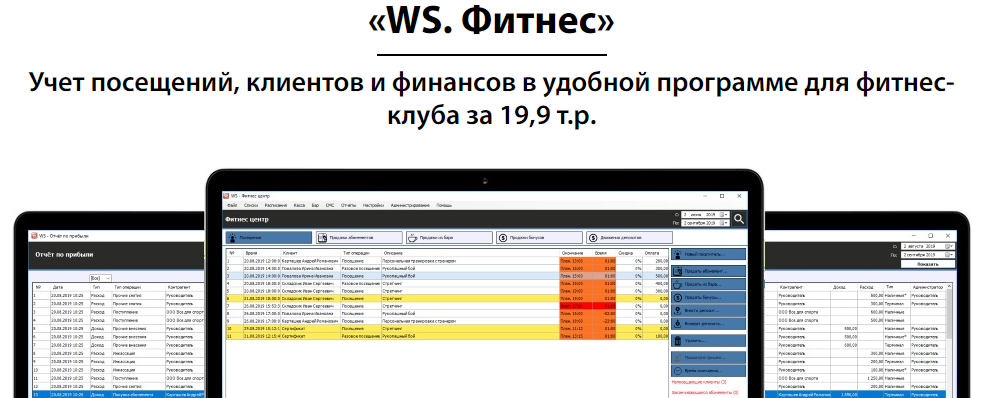


Рисунок 1 – сайт рекламы, описания и продажи приложения «WS. Фитнес»

На сайте относительно подробно описан весь функционал приложения.

Приложение даёт возможность создавать карточки клиента, добавлять туда фотографию человека.

Доп. Информацию о клиенте. Из какого источника он узнал о фитнес-клубе. Сведения о детях и приглашённых им друзьям.

Функция идентификации клиента по карте. К клиенту можно привязать карту и при ее сканировании программа автоматически идентифицирует клиента при создании нового посещения. Просмотр и редактирование абониментов клиента, возможность заморозить абонемент, клубную карту(Функция заморозки абонемента позволяет приостановить действие абонемента на определенный период.).

А так же хранит информацию о депозите клиента База Данных клиентов, напоминание о днях рождения(что бы предложить сидки или подарок для повышения лояльности клиента), скидочные системы, подарочные сертификаты и общий отчёт о клиенте.

2)Mobifitness

Система управления фитнес-клубом, спортивной студией, школой танцев или йоги.

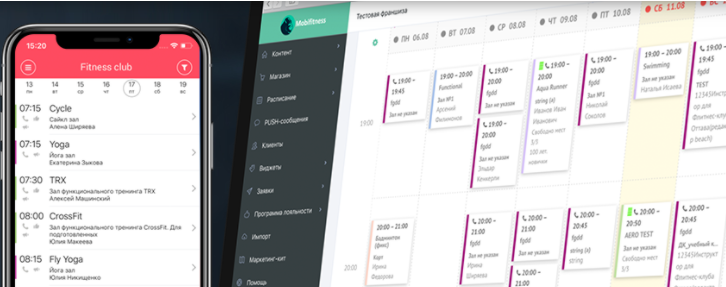


Рисунок 2 – Мобильное приложение и расписание в приложении Mobifitness

В отличии от прошлой программы, эта ориентирована не только на фитнес-центры, но и на более маленькие спортивные студии и школы танцев.

Она имеет:

* Учетную система для управления
* Мобильное приложение для коммуникации с клиентами
* Онлайн-запись для снижения нагрузки на рецепцию, для танцев и бойцовских клубов
* Онлайн-касса для ускорения работы администратора.
* Личный кабинет и интернет-магазин продаж.

Эта система продаётся целостным комплексом и имеет приятный, современный дизайн.

3)INEED CHAT

Этот сервис предоставляет почти готовое шаблонное приложение для бизнеса любого типа.

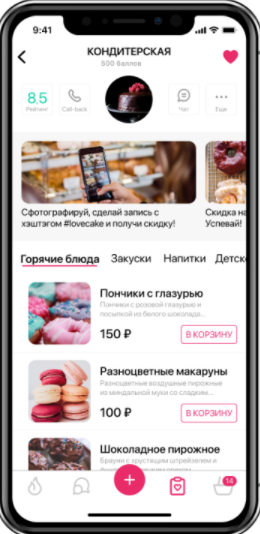


Рисунок 3 – Мобильное приложение от «INEED CHAT» на примере кондитерской

В отличии от прошлых примеров, это приложение не специализируется на услугах категории «Здоровье и медицина».

Оно проще и дешевле, но при этом, из-за ограниченности функционала, подойдёт только маленьким клубам бокса, карате или салонам красоты.

Фитнес-центрам оно не подойдёт.

Выводы: представлена была только часть аналогов. У многих интерфейс выполнен современно и стильно, у некоторых приложение выполнено удобно для пользователя, работника ли управляющего фитнес-центра.

Каждое приложение имеет интересные особенности, которые можно заимствовать.

### **2.2.3 Концептуальная модель данных**

В схеме присутствует сущность «Работник». 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.

Так же есть сущность «Групповая тренировка» она нужна для создания расписания. В расписании считается, что каждая тренировка является уникальной и проводится в определённый день недели, в определённое время. Клиент с картой имеет возможность посещать любую групповую тренировку.

Не хранится расписание уже проведённых тренировок и не ведётся учёт того, кто из клиентов на эту тренировку пришёл.

У «Клиента» может быть более одной «Клубной карты», но только одна из них может быть активна на данный момент.

Приобретённые услуги привязываются к карте. Работник может посмотреть услуги в «Каталоге услуг» и добавить их в список «Активных услуг» привязанных к конкретной карте клиента.

Ведётся учёт, в какой момент какой клиент в какое время вошёл и вышел из заведения.

Групповая тренировка и тренер связанны поле «Фамилия» в сущности тренер это первичный ключ (в теории взято, что у нас не работают два тренера с одной фамилией).

Столбец «Ссылка на тренера» в сущности «Групповая тренировка» внешний ключ, в неё нельзя добавить строку с тренером, которого не существует.

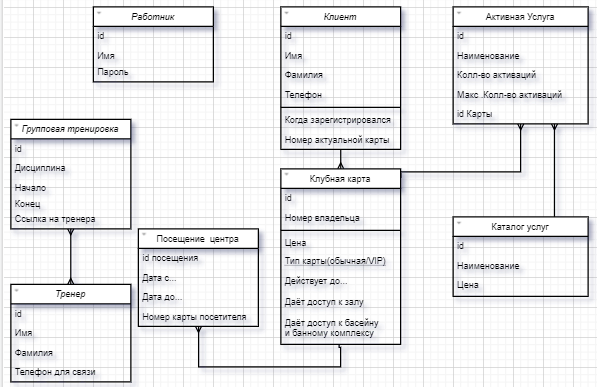
****

Рисунок 4 – Схема модели данных

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

Ввод и хранение личных данный о Клиентах и Тренерах. Имя, способ связи. Так же номер карты для клиента.

Хранение данных о расписании групповых занятий. Кто проводит, когда, и какое направление у тренировки.

Хранение данных об услугах, заказанных клиентом. Будет иметься общий прайс. Так же услугу можно будет использовать только в ограниченное время после приобретения. К примеру, один месяц, 3 месяца или год. После чего она «сгорает»

Карта клиента с двумя уровнями. Обычная и VIP. VIP карта будет давать доступ не только к тренажёрному залу, но и к «водной зоне».

Будет учитываться посещение центра клиентом. В какой момент пришёл, в какой ушёл. Чем он занимался в этой сущности учитываться не будет.

## **2.3 Сценарий использования**

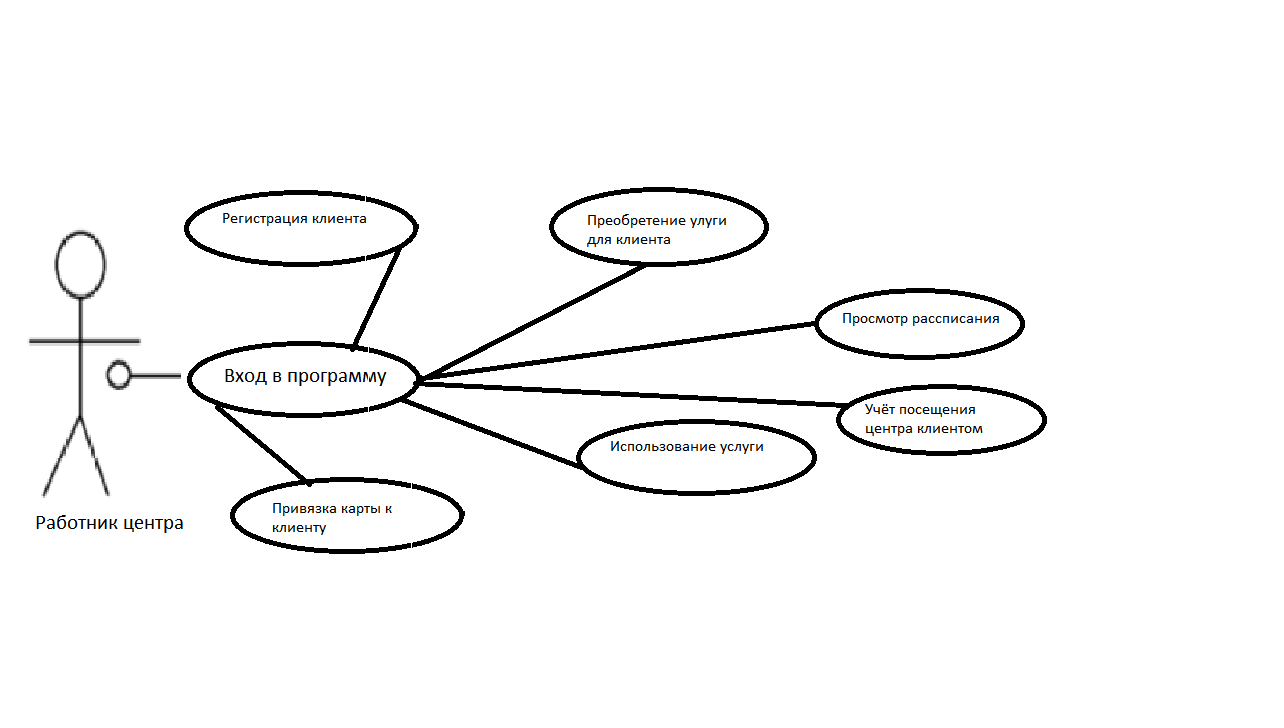


Рисунок 5 – Сценарий использования для работника.

## **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 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**

При разработке появилась потребность создать структуру приложения. Было принято решение, что приложение будет состоять из двух частей. Сервера, на котором будет храниться База Данных. И клиентского приложения, которое будет находиться на компьютере работника.

В роли сервера был использован бесплатный локальный веб-сервер MAMP(Macintosh, Apache, MySQL и PHP). Так как локальный сервер не требует затрат на хостинг. Так же обеспечивает мгновенную передачу данных на сервер и с сервера в приложение клиента.

Все запросы обрабатываются на компьютере работника, что бы излишне не нагружать сервер.

Приложение для взаимодействия с базой данных будет написало 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.

В итоге приложение выглядит так:

* Клиентская часть на Windows Forms, где будет работать пользователь и которая имеет простой и понятный для пользователя интерфейс.
* Серверная часть в виде веб-сервера MAMP. Где будет храниться база данных.

## **3.2. Разработка алгоритмов обработки информации**

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

Что бы это осуществить, нужно связать MAMP и Windows Forms, был использован коннектор «mysql connector» обеспечивающий связь сервера и программы-клиента.

Он предоставляет подключаемую библиотеку «MySql.Data.MySqlClient;», для клиентской части программы.

В приложении присутствуют кнопки, активирующие различные функции.

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**

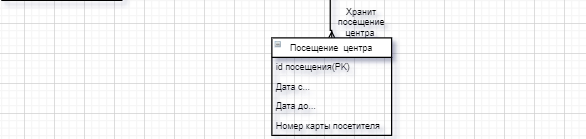
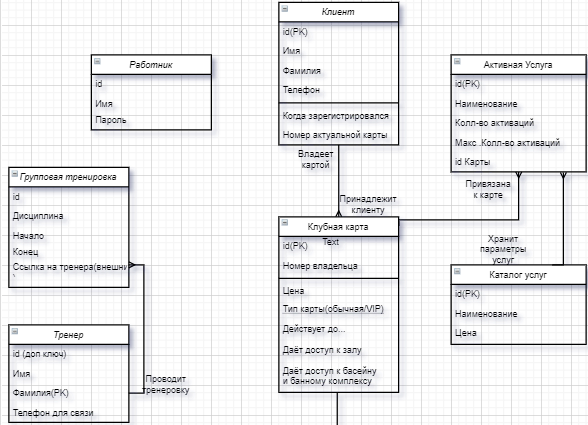


Рисунок 5 – Схема логической модели данных

## **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**

Приложение имеет следующую архитектуру:

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 руководства 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.

## **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 тренеровок

# **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