

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РЕСПУБЛИКИ КАЗАХСТАН
Некоммерческое акционерное общество
«АЛМАТИНСКИЙ УНИВЕРСИТЕТ ЭНЕРГЕТИКИ И СВЯЗИ»
Кафедра IT-инжиниринг

ДОПУЩЕН К ЗАЩИТЕ
Заведующий кафедрой
PhD, доцент

_____ Т.С. Картбаев
« ____ » _____ 2019 г.

ДИПЛОМНЫЙ ПРОЕКТ

На тему: Разработка информационной системы для автоматизации работы администратора фитнес-клуба

Специальность 5В060200 – «Информатика»

Выполнил Сейлхан М.Б. Группа ИНФ-15-2
Научный руководитель ст.преп. Ворогушина М.А.

Консультанты:

по экономической части: к.э.н., доцент _____ *А.И.* Бекишева А.И.
« 15 » 05 2019 г.

по безопасности жизнедеятельности: ст. преп. _____ *Ш.Ш.* Бекбасаров Ш.Ш.
« 15 » 05 2019 г.

по применению
программного обеспечения: ст. преп. _____ *М.Н.* Майкотов М.Н.
« 15 » 05 2019 г.

Нормоконтролер: PhD, ст. преп. _____ *Ж.К.* Алимсеитова Ж.К.
« 15 » 05 2019 г.

Рецензент: зам.дир. ИИиТТ _____ Алиев А. С.
« ____ » _____ 2019 г.

Алматы 2019

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РЕСПУБЛИКИ КАЗАХСТАН
Некоммерческое акционерное общество
«АЛМАТИНСКИЙ УНИВЕРСИТЕТ ЭНЕРГЕТИКИ И СВЯЗИ»

Институт систем управления и информационных технологий

Кафедра IT-инжиниринг

Специальность 5В060200 – «Информатика»

ЗАДАНИЕ

на выполнение дипломного проекта

Студенту: Сейлхан Мирасу Батыржанулы

Тема проекта: Разработка информационной системы для автоматизации работы администратора фитнес-клуба

Утверждена приказом по университету № 84 от «26» 10 2018г.

Срок сдачи законченного проекта «24» мая 2019 г.

Исходные данные к работе (требуемые параметры результатов исследования (проектирования) и исходные данные объекта): Руководство системы менеджмента качества на предприятии; международные стандарты ИСО-9001, данные преддипломной практики.

Перечень вопросов, подлежащих разработке в дипломном проекте, или краткое содержание дипломного проекта:

- а) Провести исследование предметной области;
- б) Разработка информационной системы для автоматизации работы администратора фитнес-клуба;
- в) Технико-экономическое обоснование проекта;
- г) Вопросы безопасности жизнедеятельности и охраны труда;

Перечень графического материала (с точным указанием обязательных чертежей): представлены 10 таблиц, 36 иллюстрации.

Основная рекомендуемая литература:

- 1 Гамма Э., Приемы объектно-ориентированного проектирования // Э.Гамма, Р.Хелм, Р.Джонсон. — СПб: Питер, 2001. — 368 с.
- 2 UML диаграммы. [Электронный ресурс] // Google Inc. — Электрон. дан. — [Б.м.], 2016 — URL: <https://docplayer.ru/29295497-Uml-diagrammy-diagramma-klassov-2-diagramma-precedentov-22.html>.

3 Бекишева А.И. Методические указания к выполнению экономической

3. Бекишева А.И. МЕТОДИЧЕСКИЕ УКАЗАНИЯ К ВЫПОЛНЕНИЮ ЭКОНОМИЧЕСКОЙ ЧАСТИ ДИПЛОМНОЙ РАБОТЫ ДЛЯ БАКАЛАВРОВ СПЕЦИАЛЬНОСТИ 5В0703 - ИНФОРМАЦИОННЫЕ СИСТЕМЫ – Алматы: АУЭС; 2013. –24 с.

4. С.Е. МАНАНБАЕВА, А.С. БЕГИМБЕТОВА ОХРАНА ТРУДА. МЕТОДИЧЕСКИЕ УКАЗАНИЯ К ВЫПОЛНЕНИЮ РАСЧЕТНО-ГРАФИЧЕСКИХ РАБОТ ДЛЯ СТУДЕНТОВ - БАКАЛАВРОВ ВСЕХ СПЕЦИАЛЬНОСТЕЙ. - Алматы: АУЭС, 2013 - 17 с.

Консультации по работе с указанием относящихся к ним разделов проекта

Раздел	Консультант	Сроки	Подпись
Экономическая часть	Бекишева А.И.	01.04.19-15.05.19	<i>[Signature]</i>
Безопасности жизнедеятельности	Бекбасаров Ш.Ш	01.04.19-15.05.19	<i>[Signature]</i>
Программная часть	Майкотов М.Н.	01.04.19-15.05.19	<i>[Signature]</i>
Нормоконтролер	Алимсеитова Ж.К	01.04.19-13.05.19	<i>[Signature]</i>

График
подготовки дипломного проекта

Наименование разделов, перечень разрабатываемых вопросов	Сроки представления научному руководителю	Примечание
Получение информации о предметной области	01.11.18-20.12.18	<i>Выполнено</i>
Проектирование	21.12.18-20.02.19	<i>Выполнено</i>
Разработка приложения	21.02.19-30.04.19	<i>Выполнено</i>
Тестирование	31.04.19-15.05.19	<i>Выполнено</i>
Внедрение и поддержка	15.05.19-20.05.19	<i>Выполнено</i>

Дата выдачи задания « 25 » октября 2018 г.

Заведующий кафедрой _____ Т.С. Картбаев

Научный руководитель проекта *[Signature]* Ворогушина М.А.

Задание принял к исполнению студент *[Signature]* Сейлхан М.Б.

Андатпа

Дипломдық жобасында талаптар мен міндеттерге сәйкес фитнес-клуб әкімшісінің жұмысын автоматтандыру үшін ақпараттық жүйенің дамуы жүргізілді.

Зерттеу тақырыбы фитнес-клубтың дамыған ақпараттық жүйесі болып табылады. Жалпы алғанда, жұмыс, шешілген мәселелерді шешуге мүмкіндік беретін ақпараттық жүйені құруға бағытталған. Өмір сүру қауіпсіздігі және еңбек жағдайлары қарастырылады. Дипломдық жобаның құнын және өтелу кезеңін есептелді.

Аннотация

В данном дипломном проекте согласно требованиям и заданию была осуществлена разработка информационной системы для автоматизации работы администратора фитнес-клуба.

Предметом исследования является разрабатываемая информационная система для фитнес-клуба. В целом работа направлена на создание информационной системы, позволяющей решать выявленные проблемы. Рассмотрены вопросы безопасности жизнедеятельности и условия труда. Произведен расчет затрат дипломного проекта и срок окупаемости.

Annotation

In this thesis project in accordance with the requirements and tasks was carried out the development of an information system for automating the work of the fitness club administrator.

The subject of the research is a developed information system for a fitness club. In general, the work is aimed at creating an information system that allows to solve problems solved. The issues of life safety and working conditions are considered. The calculation of the cost of the graduation project and the payback period.

Содержание

Введение.....	5
1 Теоритическая часть.....	6
1.1 Задачи и этапы проектирования информационной системы.....	6
1.2 Виды информационных систем в управлении.....	9
1.3 Роль управления информационными системами в бизнесе.....	11
1.4 Преимущества управления информационными системами	13
1.5 Тенденции развития автоматизированных систем в сфере фитнеса..	15
2 Проектная часть.....	17
2.1 Анализ предметной области..	17
2.2 Обзор существующих аналогов разрабатываемой системы.....	22
2.3 Обоснование программного обеспечения для разработки.....	23
2.4 Моделирование бизнес-процессов.....	25
2.5 Проектирование информационной системы с использованием	
UML.....	28
2.6 Моделирование базы данных.....	31
3 Практическая часть.....	34
3.1 Создание базы данных.....	34
3.2 Информационное наполнение.....	36
3.3. Выполнение запросов к базе данных.....	36
3.4 Реализация программного интерфейса.....	38
4 Экономическое обоснование разработки проекта.....	51
4.1 Трудоемкость разработки программного продукта.....	51
4.2 Расчет затрат на разработку ПП.....	52
4.3 Трудовые ресурсы, задействованные в работе.....	53
4.4 Определение возможной (договорной) цены ПП.....	56
4.5 Оценка социально - экономических результатов	57
5 Безопасность жизнедеятельность.....	59
5.1 Исходные данные.....	59
5.2 Расчетная часть.....	60
5.3 Выбор кондиционера для помещения.....	63
Заключение.....	64
Список литературы.....	62
Приложение А.....	63
Приложение Б.....	72
Приложение В.....	79

Введение

В настоящее время фитнес-клубы, спортивные ассоциации и другие организации, занимающиеся подготовкой и переподготовкой клиентов, становятся все более популярными. В связи с этим возникают определенные проблемы по решению специфических задач.

Целью дипломной работы является создание информационной системы для автоматизации работы фитнес-центров, позволяющей решать различные задачи данной предметной области.

Для достижения поставленной цели были определены задачи:

- сделать обзор предметной области;
- рассмотреть существующие аналоги;
- изучить процессы, протекающие в системе;
- выявить основную проблему и определить факторы, ее создающие;
- выявить основные роли пользователей ИС и описать их функции;
- построить диаграммы потоков данных ИС;
- построить функциональную модель ИС;
- спроектировать и создать базу данных;
- разработать интерфейс ИС.

Объектом исследования является деятельность фитнес-клуба и других подобных организаций, включающая в себя различные процессы, начиная от первого посещения клиентом клуба, продажи ему абонемента и так далее до каждого занятия тренировочного процесса. Предметом исследования является разрабатываемая информационная система для автоматизации работы администраторов фитнес-клубов. В целом работа направлена на создание информационной системы, позволяющей решать выявленные проблемы.

1 Теоритическая часть

1.1 Задачи и этапы проектирования информационной системы

Проектирование ИС охватывает три основные области:

- проектирование объектов данных, которые будут реализованы в базе данных;
- проектирование программ, экранных форм, отчетов, которые будут обеспечивать выполнение запросов к данным;
- учет конкретной среды или технологии, а именно: топологии сети, конфигурации аппаратных средств, используемой архитектуры (файл-сервер или клиент-сервер), параллельной обработки, распределенной обработки данных и т.п.

Проектирование информационных систем всегда начинается с определения цели проекта. В общем виде цель проекта можно определить как решение ряда взаимосвязанных задач, включающих в себя обеспечение на момент запуска системы и в течение всего времени ее эксплуатации:

- требуемой функциональности системы и уровня ее адаптивности к изменяющимся условиям функционирования;
- требуемой пропускной способности системы;
- требуемого времени реакции системы на запрос;
- безотказной работы системы;
- необходимого уровня безопасности;
- простоты эксплуатации и поддержки системы.

Согласно современной методологии, процесс создания ИС представляет собой процесс построения и последовательного преобразования ряда согласованных моделей на всех этапах жизненного цикла (ЖЦ) ИС. На каждом этапе ЖЦ создаются специфичные для него модели – организации, требований к ИС, проекта ИС, требований к приложениям и т.д. Модели формируются рабочими группами команды проекта, сохраняются и накапливаются в репозитории проекта. Создание моделей, их контроль, преобразование и предоставление в коллективное пользование осуществляется с использованием специальных программных инструментов – CASE-средств.

Процесс создания ИС делится на ряд этапов, ограниченных некоторыми временными рамками и заканчивающихся выпуском конкретного продукта (моделей, программных продуктов, документации и пр.).

Обычно выделяют следующие этапы создания ИС: формирование требований к системе, проектирование, реализация, тестирование, ввод в действие, эксплуатация и сопровождение.

Начальным этапом процесса создания ИС является моделирование бизнес-процессов, протекающих в организации и реализующих ее цели и задачи. Модель организации, описанная в терминах бизнес-процессов и бизнес-функций, позволяет сформулировать основные требования к ИС. Это

фундаментальное положение методологии обеспечивает объективность в выработке требований к проектированию системы. Множество моделей описания требований к ИС затем преобразуется в систему моделей, описывающих концептуальный проект ИС. Формируются модели архитектуры ИС, требований к программному обеспечению (ПО) и информационному обеспечению (ИО). Затем формируется архитектура ПО и ИО, выделяются корпоративные БД и отдельные приложения, формируются модели требований к приложениям и проводится их разработка, тестирование и интеграция.

Целью начальных этапов создания ИС, выполняемых на стадии анализа деятельности организации, является формирование требований к ИС, корректно и точно отражающих цели и задачи организации-заказчика.

Чтобы специфицировать процесс создания ИС, отвечающей потребностям организации, нужно выяснить и четко сформулировать, в чем заключаются эти потребности. Для этого необходимо определить требования заказчиков к ИС и отобразить их на языке моделей в требования к разработке проекта ИС так, чтобы обеспечить соответствие целям и задачам организации.

Задача формирования требований к ИС – одна из наиболее ответственных, трудно формализуемых и наиболее дорогих и тяжелых для исправления в случае ошибки. Современные инструментальные средства и программные продукты позволяют достаточно быстро создавать ИС по готовым требованиям. Но зачастую эти системы не удовлетворяют заказчиков, требуют многочисленных доработок, что приводит к резкому удорожанию фактической стоимости ИС. Основной причиной такого положения является неправильное, неточное или неполное определение требований к ИС на этапе анализа.

На этапе проектирования, прежде всего, формируются модели данных. Проектировщики в качестве исходной информации получают результаты анализа. Построение логической и физической моделей данных является основной частью проектирования базы данных. Полученная в процессе анализа информационная модель сначала преобразуется в логическую, а затем в физическую модель данных.

Параллельно с проектированием схемы базы данных выполняется проектирование процессов, чтобы получить спецификации (описания) всех модулей ИС. Оба эти процесса проектирования тесно связаны, поскольку часть бизнес-логики обычно реализуется в базе данных (ограничения, триггеры, хранимые процедуры). Главная цель проектирования процессов заключается в отображении функций, полученных на этапе анализа, в модули информационной системы. При проектировании модулей определяют интерфейсы программ: разметку меню, вид окон, горячие клавиши и связанные с ними вызовы.

Конечными продуктами этапа проектирования являются:

– схема базы данных (на основании ER-модели, разработанной на этапе анализа);

– набор спецификаций модулей системы (они строятся на базе моделей функций).

Кроме того, на этапе проектирования осуществляется также разработка архитектуры ИС, включающая в себя выбор платформы (платформ) и операционной системы (операционных систем). В неоднородной ИС могут работать несколько компьютеров на разных аппаратных платформах и под управлением различных операционных систем. Кроме выбора платформы, на этапе проектирования определяются следующие характеристики архитектуры:

- будет ли это архитектура «файл-сервер» или «клиент-сервер»;
- будет ли это 3-уровневая архитектура со следующими слоями: сервер, ПО промежуточного слоя (сервер приложений), клиентское ПО;
- будет ли база данных централизованной или распределенной. Если база данных будет распределенной, то какие механизмы поддержки согласованности и актуальности данных будут использоваться;
- будет ли база данных однородной, то есть, будут ли все серверы баз данных продуктами одного и того же производителя (например, все серверы только Oracle или все серверы только DB2 UDB). Если база данных не будет однородной, то какое ПО будет использовано для обмена данными между СУБД разных производителей (уже существующее или разработанное специально как часть проекта);
- будут ли для достижения должной производительности использоваться параллельные серверы баз данных (например, Oracle Parallel Server, DB2 UDB и т.п.).

Этап проектирования завершается разработкой технического проекта ИС.

На этапе реализации осуществляется создание программного обеспечения системы, установка технических средств, разработка эксплуатационной документации.

Этап тестирования обычно оказывается распределенным во времени.

После завершения разработки отдельного модуля системы выполняют автономный тест, который преследует две основные цели:

- обнаружение отказов модуля (жестких сбоев);
- соответствие модуля спецификации (наличие всех необходимых функций, отсутствие лишних функций).

После того как автономный тест прошел успешно, модуль включается в состав разработанной части системы и группа сгенерированных модулей проходит тесты связей, которые должны отследить их взаимное влияние.

Далее группа модулей тестируется на надежность работы, то есть проходят, во-первых, тесты имитации отказов системы, а во-вторых, тесты наработки на отказ. Первая группа тестов показывает, насколько хорошо система восстанавливается после сбоев программного обеспечения, отказов аппаратного обеспечения. Вторая группа тестов определяет степень устойчивости системы при штатной работе и позволяет оценить время

безотказной работы системы. В комплект тестов устойчивости должны входить тесты, имитирующие пиковую нагрузку на систему.

Затем весь комплект модулей проходит системный тест – тест внутренней приемки продукта, показывающий уровень его качества. Сюда входят тесты функциональности и тесты надежности системы.

Последний тест информационной системы – приемо-сдаточные испытания. Такой тест предусматривает показ информационной системы заказчику и должен содержать группу тестов, моделирующих реальные бизнес-процессы, чтобы показать соответствие реализации требованиям заказчика.

Необходимость контролировать процесс создания ИС, гарантировать достижение целей разработки и соблюдение различных ограничений (бюджетных, временных и пр.), привело к широкому использованию в этой сфере методов и средств программной инженерии: структурного анализа, объектно-ориентированного моделирования, CASE-систем.

Основными задачами, решению которых должна способствовать методология проектирования корпоративных ИС, являются следующие:

- обеспечивать создание корпоративных ИС, отвечающих целям и задачам организации, а также предъявляемым требованиям по автоматизации деловых процессов заказчика;

- гарантировать создание системы с заданным качеством в заданные сроки и в рамках установленного бюджета проекта;

- поддерживать удобную дисциплину сопровождения, модификации и наращивания системы;

- обеспечивать преемственность разработки, т.е. использование в разрабатываемой ИС существующей информационной инфраструктуры организации (задела в области информационных технологий) [1].

1.2 Виды информационных систем в управлении

Информационные системы управления - одна из нескольких информационных систем, которые используются в бизнесе. Чтобы лучше понять информационные системы управления, давайте рассмотрим различные типы информационных систем, доступных в бизнесе.

Системы обработки транзакций. Эти системы были разработаны для сбора, обработки и хранения транзакций, которые происходят в повседневной деятельности компании. Система также может быть использована для отмены или изменения транзакций, выполненных в прошлом, если возникнет такая необходимость. Одним из свойств этой системы, которое позволяет им эффективно работать, является возможность точной записи нескольких транзакций, даже если разные транзакции происходят одновременно. Они созданы для обработки больших объемов транзакций. Примеры включают системы управления запасами, системы начисления заработной платы, системы обработки заказов и т. д. Системы поддержки принятия решений.

Эти системы помогают лицам, принимающим решения, принимать лучшие решения, генерируя статистические прогнозы на основе проанализированных данных. Хотя это не устраняет необходимость в суждении менеджера, оно значительно улучшает качество решения, предлагая прогнозы, которые помогают определить наилучший курс действий. Эти системы собирают информацию из нескольких источников для целей принятия решений. Примерами таких систем являются совместная работа на компьютере, системы поддержки принятия групповых решений, системы логистики и системы финансового планирования.

Исполнительные информационные системы. Этот инструмент, также известный как Система поддержки исполнительной власти, используется для представления данных по всему предприятию высшим руководителям. Эти системы предоставляют быстрые и простые в использовании отчеты, которые представлены в графических дисплеях, которые легко сравнивать. Их можно рассматривать как специализированные системы поддержки принятия решений, поскольку они предоставляют информацию, необходимую для повышения качества решений. В связи с высокими ожиданиями от такой системы, эти системы должны быть высоко индивидуализированы, следовательно, они обычно изготавливаются на заказ для конкретных клиентов. Они также настраиваются в соответствии с конкретными потребностями клиентов.

Информационные системы управления. Эти системы используют информационные технологии, чтобы помочь менеджерам обеспечить бесперебойную и эффективную работу организации. Информация, собираемая этими системами, структурирована таким образом, чтобы менеджеры могли легко оценить текущие результаты деятельности компании по сравнению с предыдущими результатами. Некоторые из распространенных типов информационных систем управления включают в себя системы управления процессами, системы управления персоналом, системы продаж и маркетинга, системы управления запасами, системы автоматизации делопроизводства, системы планирования ресурсов предприятия, системы учета и финансирования, а также системы управленческой отчетности.

Компоненты управленческих информационных систем. Для эффективной доставки информации, необходимой лицам, принимающим решения, системы управленческой информации должны иметь необходимые компоненты для сбора, обработки, хранения и извлечения информации, когда это необходимо. Для этого в этих системах используются следующие четыре компонента:

Информационная система. Это комбинация программного обеспечения, оборудования, персонала и инфраструктуры. Этот компонент помогает в сборе данных, которые хранятся в MIS. Аппаратное обеспечение включает компьютеры, сканеры, принтеры и сетевые устройства. Элементы программного обеспечения включают корпоративное программное обеспечение компании и любое другое программное обеспечение,

используемое для работы сети компании. Этот компонент позволяет сотрудникам взаимодействовать с системой, что позволяет собирать информацию.

Система управления базами данных. Этот компонент в основном состоит из компьютерных программ, которые помогают в хранении и извлечении данных. Конечно, он также включает в себя реальные физические базы данных, где информация хранится после того, как она была захвачена. Существует несколько различных систем управления базами данных, которые можно использовать в информационных системах управления. Пригодность систем будет зависеть от объема данных, которые необходимо будет обработать и сохранить в системе. Существуют небольшие системы управления базами данных, которые могут комфортно работать на персональных компьютерах, и есть огромные, которым потребуются более крупные и более сложные машины, такие как мэйнфрейм-компьютеры. Узнайте больше о технологиях баз данных.

Интеллектуальная система. Этот компонент связан с обработкой собранных данных и представлением их способом, который легко понять. Все, начиная от обработки данных и заканчивая отображением данных, призвано облегчить руководителям процесс принятия решений, касающихся бизнеса. Иногда его называют бизнес-аналитикой, которая хранит человеческие знания и использует логику для формулирования быстрых решений для будущих проблем, где шаблоны соответствуют

Система исследований. Этот компонент связан с выявлением основных проблем управления в организации и выработкой альтернативных решений, которые могли бы быть достаточными в конкретной ситуации. Это помогает обеспечить анализ всех возможных вариантов и принятие наилучшего решения. Лучшее решение не всегда самое очевидное. Этот компонент информационных систем управления гарантирует, что наилучшее решение будет достигнуто даже в этих случаях.

1.3 Роль управления информационными системами в бизнесе

Основная роль информационных систем управления заключается в составлении отчетов о деловых операциях с целью поддержки принятия решений. Это делается для того, чтобы организация управлялась лучше и эффективнее, чтобы она могла в полной мере реализовать свой потенциал и получить конкурентное преимущество.

Давайте посмотрим на некоторые другие роли, которые играют информационные системы управления в организации.

Быстро предоставлять информацию лицам, принимающим решения. Независимо от того, является ли это маркетинговым, финансовым или операционным вопросом, менеджерам необходим быстрый доступ к информации, чтобы они могли принимать правильные решения, которые окажут положительное влияние на результаты деятельности компании.

Информационные системы управления усиливают это путем стратегического хранения огромного количества информации о компании в центральном месте, к которому менеджеры могут легко получить доступ через сеть. Это означает, что менеджеры из разных отделов имеют доступ к одной и той же информации, поэтому они смогут принимать решения, которые в совокупности помогут быстрее решить проблемы компании.

Информационные системы управления также помогают в сборе данных. Данные о повседневных операциях в компании собираются и объединяются с данными из внешних источников. Это обеспечивает здоровые и функциональные отношения между дистрибьюторами, торговыми точками и любыми другими участниками цепочки поставок. Это также помогает отслеживать производительность, так как производственные и торговые показатели будут записываться, и храниться в центральной базе данных, к которой могут обращаться все члены MIS. Доступ к этой информации также помогает обеспечить раннее обнаружение проблем и быстрое принятие решений с использованием самой последней информации.

Содействовать сотрудничеству на рабочем месте. В любой крупной компании существует множество ситуаций, в которых требуется принять участие от нескольких лиц или отделов, прежде чем могут быть приняты решения. Без эффективного канала связи эти решения могут занять очень много времени. Даже при наличии хороших каналов связи, если различные заинтересованные стороны не имеют доступа ко всем доступным данным, процесс может столкнуться с некоторыми препятствиями до его завершения. Информационные системы управления обеспечивают, чтобы все члены группы принятия решений имели доступ ко всем данным, которые необходимы для принятия решения, даже если они работают из разных физических мест.

Для запуска возможных сценариев в разных бизнес-средах. Перед принятием решения, которое повлияет на общее положение бизнеса, необходимо принять много мер предосторожности. Необходимо проверить и убедиться, что компания не пострадает после принятия решения. Информационные системы управления позволяют руководителям запускать сценарии «что если», чтобы они могли видеть, как какое-либо из важных показателей в бизнесе будет затронуто данным решением. Данные представлены в виде простых для понимания отчетов и графиков, облегчающих интерпретацию. Например, менеджер по персоналу сможет рассказать, что произойдет с доходом, производством, продажами и даже прибылью после сокращения числа работников в производственном отделе. Другим примером может быть влияние изменения цены на прибыльность. После того, как руководители смогут понять, будет ли решение выгодным для компании, легче будет принимать правильные решения, которые не оставят компанию в хаосе.

Информационные системы управления дают точные прогнозы о положении компании в краткосрочной и долгосрочной перспективе.

Большинство решений, принимаемых топ-менеджерами в компаниях, влияют на стратегии компании. В результате некоторые из них могут нуждаться в некоторых изменениях в целях или стратегиях компании. Большинство систем управления информацией снабжены функциями анализа тенденций, которые позволят вам спрогнозировать эффективность бизнеса с текущей конфигурацией и то, как они будут затронуты после внесения любых изменений, которые вы рассматриваете. Информационные системы управления, которые не имеют функции анализа тенденций, по-прежнему предоставляют вам достаточно информации для точного проведения анализа с использованием внешних инструментов.

Информационные системы управления помогают отслеживать выполнение конкретных решений в компании. Прежде чем принять решение, руководители используют эти системы для прогнозирования ожиданий от конкретного решения. Если они решат продолжить внесение изменений, необходимо будет следить за производительностью, чтобы убедиться, что вы на пути к достижению желаемых результатов. Информационные системы управления дают подробные отчеты и рекомендации, так что оценка целей проходит гладко и эффективно. Вы получаете данные, которые показывают, оказали ли ваши решения желаемый эффект. Если нет, вы сможете принять необходимые корректирующие меры заранее, чтобы вы могли вернуться на правильный путь.

Улучшить отчетность компании. Одной из причин, по которой крупные информационные компании предпочитают управленческие информационные системы, является эффективность функций отчетности. Решения могут быть приняты быстро, потому что информация представлена в удобном для понимания формате. Тот факт, что система доступна для людей из разных частей организации, делает ее эффективным средством отчетности и коммуникации. Результаты могут быть переданы коллегам со всеми необходимыми дополнительными данными. Также возможно создание кратких исполнительных резюме, которые суммируют всю ситуацию для рассмотрения старшими руководителями компании в ситуациях, которые требуют их одобрения.

1.4 Преимущества управления информационными системами

Использование информационных систем управления дает много преимуществ. Некоторые из этих преимуществ помогают руководству работать легче, а остальные помогают организации в целом.

Давайте внимательнее посмотрим, что вы выиграете от использования MIS. Все заинтересованные стороны в компании имеют доступ к единой базе данных, которая содержит все данные, которые будут необходимы для повседневной работы. Если MIS используется для управления проектом, подрядчик, клиент и консультант смогут достичь высокого уровня прозрачности, следовательно, будет легко развить доверие. Операции также

будут более плавными, поскольку информация всегда будет легко доступна, а методы сбора данных, такие как формы или вопросники, будут стандартизированы.

Сотрудники и другие заинтересованные стороны в организации смогут тратить больше времени на выполнение продуктивных задач. Это потому, что большая часть их времени экономится благодаря более эффективной информационной системе. В противном случае это время было бы потрачено на настройку или поиск традиционных систем записи информации, таких как формы и файлы. В результате компания может сэкономить на затратах на рабочую силу, и в то же время производство большего объема продукции за фиксированный промежуток времени теперь будет расходоваться продуктивно. Еще одним преимуществом информационных систем управления является то, что они предоставляют возможности инструментов обработки данных, которые помогают значительно улучшить качество решений, принимаемых в компании. Большинство систем управленческой информации имеют встроенные инструменты обработки данных, которые могут делать выводы на основе входных данных, полученных из различных источников. Это помогает лучше планировать управление материальными ресурсами, распределение рабочей силы и даже общее выполнение проекта. Благодаря гибкости, которую обеспечивает использование мобильных устройств, таких как планшетные компьютеры и смартфоны, системы управления информацией обеспечивают более легкое и более тесное взаимодействие сотрудников с информацией о ходе выполнения любого процесса в организации. Это также обеспечивает более высокую степень точности сбора данных, поскольку на мобильных устройствах будет возможно записывать прогресс в течение меньших этапов в течение дня, а не записывать один раз в конце дня. В результате, руководство может получить лучшее представление о прогрессе благодаря доступности самой последней информации. Входы и модификации в этих системах регистрируются, и авторы отмечают. Время внесения изменений также записывается для дальнейшего использования. Это означает, что компания может достичь более высокой степени ответственности, поскольку все действия всегда можно отследить до отдельных лиц, которые их инициировали. Это также означает, что наиболее эффективных сотрудников также можно легко определить, поскольку такая информация, как производственные показатели за смену и отчеты о продажах, всегда доступна и хорошо представлена в системе.

Информационные системы управления помогают сократить объем бумажной работы, с которой приходится иметь дело департаментам, благодаря центральной базе данных, доступной из сети компании. Это означает, что, помимо упрощения и ускорения процессов, компания может обходиться без бумаги и в то же время сокращать выбросы углекислого газа. Счета также снижаются, поскольку потребность в таких предметах, как обычная бумага, чернила и картриджи с тонером, будет значительно снижена. Транспортные расходы также снижаются, так как не будет необходимости

отправлять документы туда и обратно для утверждения и подписи. Место на полках будет сохранено и использовано для других задач. Отходы компании также будут сокращены, когда компания станет безбумажной.

Информационные системы управления предоставляют отчеты о доходах, отчеты о производительности для сотрудников, отчеты об отслеживании расходов и многое другое.

Информационные системы управления с точки зрения высшего руководства помогают создать общее представление о финансовом положении компании. Эти системы могут также предоставлять общие отчеты о состоянии для конкретных проектов в рамках организации[2]. Это позволяет высшим руководителям и менеджерам легко определить, находится ли компания на пути к достижению своих целей.

Благодаря информационным системам управления компания получает конкурентное преимущество. Это связано с тем, что операции выполняются быстрее и плавнее, и поэтому результаты достигаются быстрее и эффективнее. Клиенты будут довольны предоставлением услуг, потому что они будут получать ответы, которые они ищут быстрее, и сотрудники будут мотивированы, потому что большинство задач станет легче благодаря лучшему доступу к данным.

MIS помогает устранить лишние роли. Когда информация хранится эффективно, можно идентифицировать части системы, которые не нужны. Это означает, что любые усилия, которые были дублированы, устранены, следовательно, компания может лучше использовать имеющиеся ресурсы.

1.5 Тенденции развития автоматизированных систем в сфере фитнеса

Развитие автоматизированных систем не стоит на месте. На сегодняшний день можно утверждать, что только при наличии комплексной системы возможна эффективная деятельность современного предприятия []. За последние несколько лет, появились большое количество прикладных программных решений для тренажерных залов. На сегодняшний день в самых продвинутых системах предусмотрена автоматизация следующих процессов:

- работа с клиентами;
- работа с абонементом;
- работа с финансами;
- работа со складом;
- работа с отчетностью;
- работа с персоналом;
- учет маркетинговых предприятий (СМС и e-mail рассылка);
- обмен данными с другими информационными базами;
- работа с дополнительными услугами и т.д.;
- возможность подключения необходимого оборудования;
- работа с аналитикой.

Важным условием является обеспечение возможности:

- расширения функционала;
- изменения и настройки интерфейса для более удобного и понятного использования;
- расширение необходимых настроек программного обеспечения;
- развитие интеграции с различным специализированным оборудованием (торговым, охранным, регистрационным и т.п.).

2 Проектная часть

2.1 Анализ предметной области

Тренажерный зал «MaxGym» оказывает услуги в сфере фитнеса и силовых упражнений. Предприятие находится по адресу г. Алматы, мкр Кокжиек д44. Общая площадь составляет 500 м². Для удобства клиентов зал разделён на 2 части:

Зал для фитнеса. Общая площадь зала составляет около 140 м². Зал сдаётся в аренду, в свободное время клиенты могут посещать его и использовать весь инвентарь. Зал оснащён различными видами спортивного оборудования таких как:

- степ-платформы;
- коврики гимнастические;
- гири;
- гимнастические мячи;
- скакалки;
- гантели для аэробики;
- штанги для аэробики и т. д.

Зал силовых упражнений. Общая площадь составляет около 280м².

Данный зал предназначен только для основных клиентов. Зал оснащён различными видами спортивного оборудования таких как:

- беговые дорожки;
- велотренажеры;
- тяга сверху;
- жим ногами;
- сгибание ног лёжа;
- сгибание ног сидя;
- жим лежа;
- комплекс для персональных тренировок и т.д.

В целях безопасности тренажёры размещены так, чтобы не мешать клиентам, занимающимся на ближайших тренажерах. Организационная структура тренажерного зала «MaxGym». Директор управляет предприятием, решает все вопросы, касающиеся управления деятельностью предприятия, отслеживает и анализирует статистику по деятельности предприятия. Имеет право первой подписи, распоряжается имуществом предприятия. Осуществляет прием и увольнение работников предприятия. Администраторы консультируют клиентов, регистрируют клиентов, осуществляют оформление абонементов и отслеживание посещений по ним, помогают клиенту выбрать более подходящее решение, формируют необходимые отчеты.

Главный бухгалтер ведёт бухгалтерский учёт. Тренеры следят за порядком в залах, в случае необходимости помогают клиентам, консультируют их по возникающим в ходе тренировки вопросам, ведут учет личных достижений клиентов, проводят индивидуальные занятия,

отслеживают состояние тренажеров.

Обслуживающий персонал содержит в чистоте и порядке все залы, а также обеспечивает исправное состояние различных систем обеспечения: отопительной системы в зимний период и исправность электросетей в залах и подсобных комнат. Организационная структура предприятия, представлена на рисунке 2.1



Рисунок 2.1 – Организационная структура

В тренажерном зале имеются различные виды абонементов. Абонемент представляет собой карточку с фамилией, инициалами и видом выбранного абонемента и периодом его действия. Прежде чем начать занятия новому клиенту необходимо обратиться к администратору, который консультирует клиента и оформляет подходящий клиенту абонемент. Клиент, который уже зарегистрирован должен предъявить абонемент и подтвердить достоверность информации абонемента.

Администратору в этом случае нужно проверить информацию о клиенте и абонементах и отметить посещение. Клиенты с абонементами, срок которых истек, на занятия не допускаются. В тренажерном зале учитываются пожелания клиентов, для увеличения посещаемости введены различные виды абонементов, таких как:

Абонементы для дневных занятий. Посещать зал разрешается с 10:00 до 17:00. Стоимость таких абонементов. Количество посещений Стоимость/на месяц Абонемент на 4 занятия 2000 тг. Абонемент на 8 занятий 4000. Абонемент на 12 занятий 1100 руб. 23 2. Абонементы для вечерних занятий. тг. Посещать зал разрешается с 17:00 до 22:00. Количество посещений Стоимость/на месяц Абонемент на 4 занятия 2400 тг. Абонемент на 8 занятий 4800 тг. Абонемент на 12 занятий 6000 тг. Цены на абонементы приведены в таблице 2.1

Таблица 2.1 – Цены на абонемент

Название абонемента	посещений	Цена тг
Дневной	4	2000
Дневной	8	4000
Вечерний	8	4800
Вечерний	12	6000

Специальные абонементы. Специальные абонементы действуют на ограниченный круг лиц, таких как: студенты, школьники, семейные пары. Для привлечения клиентов созданы выгодные условия посещения зала. Посещать зал разрешается в любое удобное время (по графику работы тренажерного зала). Стоимость/на месяц Школьный (При предъявлении дневника) 3000 тг. Студенческий (При предъявлении студ. билета) 5000 тг. Безлимитные абонементы. При приобретении безлимитного абонемента клиент может посещать зал в любое время и день недели (по графику работы тренажерного зала). Отличаются они лишь сроком, на который они приобретаются. Цены на специальные абонементы указаны в таблице 2.2

Таблица 2.2 – Специальные абонементы

Название абонемента	посещений	Цена тг
Школьный	12	3000
Студенческий	12	5000
Безлимит	1 месяц	8000

Администратор тренажерного зала консультирует клиентов по возникающим вопросам, помогает подобрать наиболее подходящий вариант для клиента, регистрирует клиентов, абонементы, аренду и т.д. Администратор сидит за стойкой сразу же при входе в тренажерный зал. Должностные обязанности администратора представлены на рисунке 2.2

Директор тренажерного зала осуществляет общее руководство, формирует различные отчеты и ведет необходимую статистику. Должностные обязанности директора тренажерного зала представлены на рисунке 2.3

Тренер тренажерного зала ведет учет личных достижений клиентов, консультирует клиентов по вопросам использования спортивного инвентаря, проводит индивидуальные тренировки, отслеживает исправность инвентаря и т.п. Должностные обязанности тренера тренажерного зала представлены на рисунке 2.4



Рисунок 2.2 — Должностные обязанности администратора

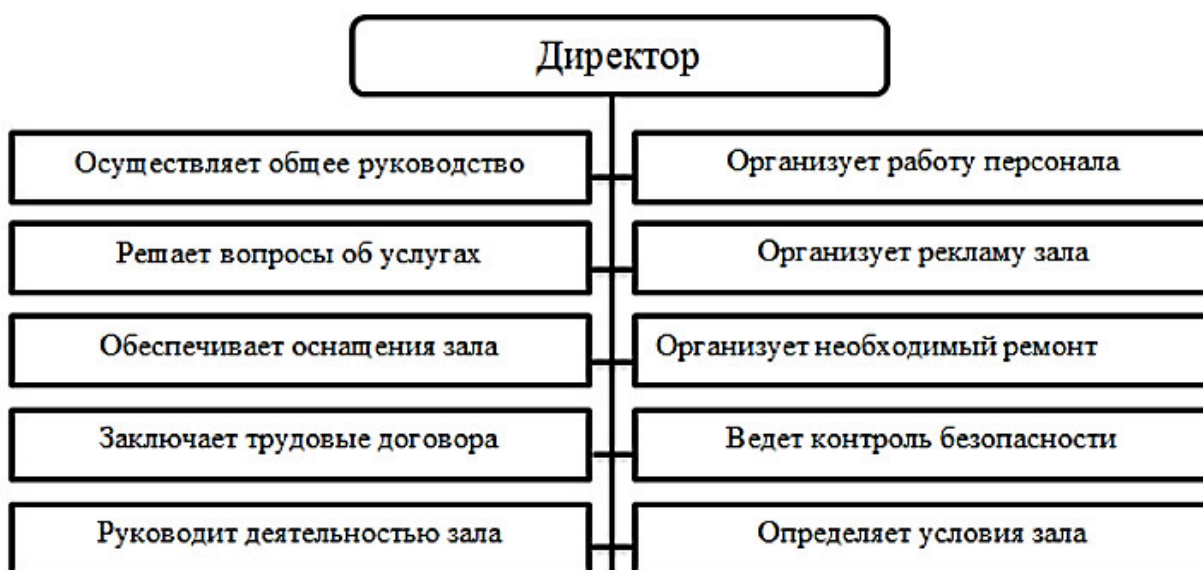


Рисунок 2.3 – Должностные обязанности директора тренажерного зала

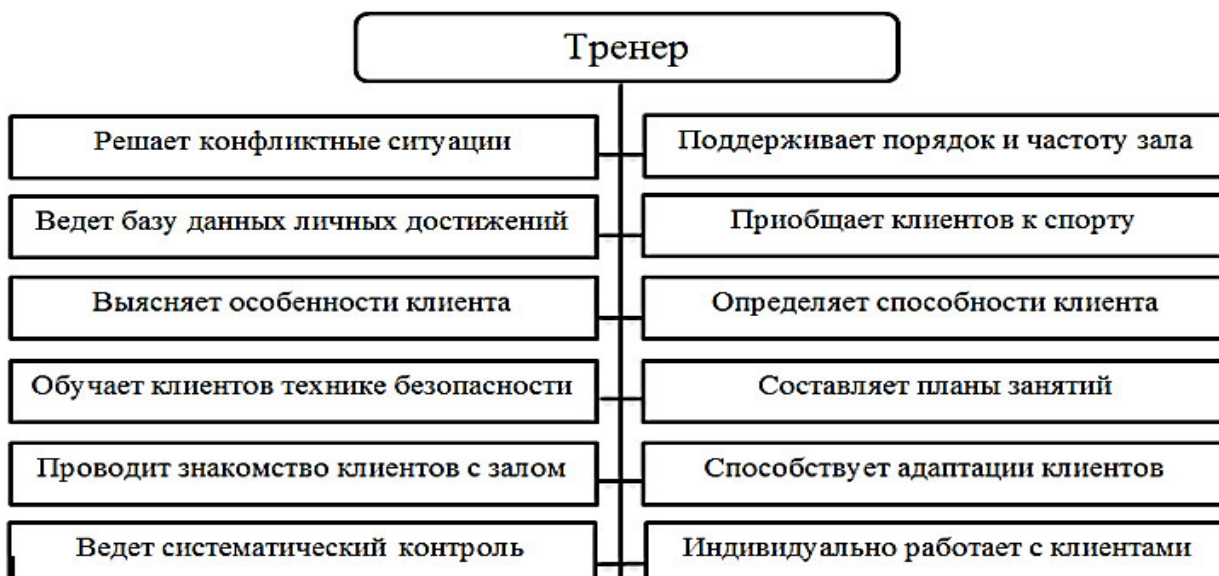


Рисунок 2.4 – Должностные обязанности тренера тренажерного зала

На момент становления предприятия в тренажёрном зале не было программы для учёта клиентов, абонементов, составления программы и вывода отчётов. Вся запись клиентов осуществлялась на бумажных носителях (журналах) ручным способом. А для учёта клиентов и составления отчётов использовался пакет программ Microsoft Office. Отчёты составлялись в электронных таблицах MS Excel. Ведение личных достижений осуществлялись ручным способом в журналах и электронных таблицах MS Excel. С увеличением числа клиентов, увеличилось время, необходимое для выполнения необходимых операций, возросла нагрузка на сотрудников тренажерного зала, выполняющих различные функции.

Недостатки процесса обслуживания клиента:

- Большие временные затраты на регистрацию клиентов, абонементов, аренды помещения зала;
- невозможность качественного консультирования клиентов из-за длительного поиска или невозможности оперативного доступа к необходимой информации;
- неэффективная технология хранения и оперативного поиска необходимой информации;
- возможность ошибочных действий по допуску клиентов к услугам (закончился абонемент по времени или по количеству раз).

В ходе анализа текущей информационной модели было принято решение автоматизировать основные процессы (ведение базы клиентов, базы абонементов, спортивного инвентаря и т.д.), а также автоматизировать процесс учета личных достижений клиентов. В целях анализа текущего состояния информационной системы рассмотрим причинно-следственную диаграмму, которая позволит формализовать и структурировать причины возникновения того или иного события, а также устанавливать причинно-следственные связи. При работе с данной методикой анализ проблемы подразумевает, что причины её возникновения могут быть одной из так называемых «5М».

Рассмотрим ошибки и недостатки существующей системы обработки данных, проанализируем причины их возникновения в разрезе пяти следующих факторов:

- Человек причины связанные с человеческими факторами ошибки при регистрации клиентов и оформлении документов, отслеживании срока действия абонементов, ошибочный расчет стоимости услуг из-за ручного способа их выполнения, ошибки при заполнении и формировании отчётов;
- машины и оборудование причины, связанные с оборудованием: технологии формирования итоговых результатов деятельности предприятия трудоемки и не обеспечивают получение достоверных данных;
- материал причины связанные с материалами велики затраты на ведение, передачу, хранение и учёт на бумажных носителях;
- методы причины связанные с организацией процессов долгая

обработка информации ручным путем (регистрация клиентов, регистрация абонементов, регистрация аренды, контроль оплаты, регистрация посещений, ведение личных достижений, неудобства в учете изменения цен на абонементы, аренду и т.д.);

- измерения причины», связанные с методами измерения: сложность и медлительность расчётов необходимых показателей (неисправность инвентаря, занятость тренажёрного зала и пр.).

Проанализировав несовершенство существующей технологии работы сотрудников тренажёрного зала, приходим к выводам о необходимости ее совершенствования путем автоматизации основных бизнес-процессов, которая позволит решить вышеперечисленные проблемы.

2.2 Обзор существующих аналогов разрабатываемой системы

10to8. Ведущая система онлайн-бронирования и планирования встреч, которая помогает компаниям более эффективно общаться с клиентами, сокращая количество неявок и предотвращая конфликты по расписанию. Он может обрабатывать громкие встречи и имеет такие функции, как двусторонний чат, встроенный в календарь; интеграция с онлайн-календарями; автоматизированные интеллектуальные SMS и электронные оповещения; ротационное управление, чтобы легко координировать доступность персонала и назначения; аудиторские функции; приборная панель и отчетность; и многое другое.

Ум тело. Программное обеспечение для управления бизнесом для компаний, работающих в сфере здравоохранения и оздоровления, с функциями, которые автоматизируют бизнес-задачи, включая сбор платежей, подтверждения, перепланирование и обработку заработной платы. Он также имеет маркетинговые инструменты, которые позволяют организациям поддерживать связь с клиентами и интегрировать рекламные акции в точках продаж. Среди его многочисленных возможностей - автоматические уведомления, подтверждения и напоминания, назначение в облаке через веб-сайт или мобильные приложения, отчетность и отслеживание эффективности бизнеса, а также прямая ссылка на страницу в социальных сетях с обновлениями в режиме реального времени.

Планировщик дзен. Популярный инструмент управления бизнесом и планирования, разработанный для фитнеса. Облачная платформа упрощает бизнес-операции и функции управления членством благодаря инструментам и функциям, которые отслеживают посещаемость и прогресс участников, позволяют осуществлять онлайн-планирование, включают автоматическое выставление счетов и интегрированную обработку платежей, предоставляют настраиваемые отчеты и аналитику, а также предлагают клиентское самообслуживание через мобильные приложения и участников портал.

Virtuagym. Универсальное решение для тренеров, фитнес-клубов и тренеров для управления повседневной деятельностью своих клубов через

онлайн-платформу управления. Он создан для того, чтобы профессионалы в области фитнеса могли сосредоточиться на эффективном тренинге по фитнесу и одновременно вести успешный бизнес. Он предлагает индивидуальные фирменные приложения и онлайн-сообщество для продвижения бренда клуба или тренера и повышения вовлеченности клиентов. Он имеет гибкие параметры членства, аппаратную интеграцию, режимы онлайн-платежей, функцию управления членством и автоматизированную систему регистрации.

Trainerize. Программное решение для фитнеса для спортивных залов и фитнес-центров, которое может управлять повседневными делами. Программное обеспечение помогает тренерам управлять индивидуальными клиентами, строить индивидуальные планы тренировок, отслеживать их эффективность. Trainerize предоставляет обширную библиотеку видеофильмов, где тренеры могут изучать и обучать клиентов различным упражнениям. В дополнение к упражнениям, инструкторы могут использовать свой смартфон для мониторинга потребления пищи клиентом или отправлять клиентам предписанные планы питания в виде файлов PDF. Программное обеспечение также имеет встроенные функции автоматического обмена сообщениями и уведомлениями, а также модуль обмена сообщениями приложений, позволяющий проводить сеансы через Skype.

Coacha. Разработанный для спортивных клубов, Coacha - это полнофункциональное решение для соблазнения спортсменов и менеджеров, мониторинга спортивных результатов, отслеживания посещаемости, обработки членских взносов и других. С помощью этого программного обеспечения вы можете создавать профили спортсменов и аккуратно упорядочивать их по категориям на основе спортивных возможностей, пола и других факторов. Для беспрепятственного общения, программное обеспечение позволяет пользователям отправлять массовые электронные письма или использовать SMS-трансляцию, чтобы привлечь большое количество спортивных групп или тренеров.

2.3 Обоснование выбора программного обеспечения для разработки

В качестве СУБД для разрабатываемой информационной системы используется СУБД MS SQL Server Server, возможности которой ограничены, пожалуй, лишь аппаратными ресурсами. В MS SQL Server Server включены все новейшие разработки по безопасному хранению, обработке и конечному представлению данных. База данных MS SQL Server содержит следующие виды файлов:

- Управляющие файлы (Control files) - содержат метаданные о самой базе данных. Эти файлы очень важны для базы данных. Без них не могут быть открыты файлы данных и поэтому не может быть открыт доступ к информации базы данных;
Файлы управления (файлы управления) добавление метаданных для данных базы данных. Эти файлы очень важны для базы данных. Это может быть обща

информация.

Файлы данных (файлы данных). Иметь информацию базы данных. Интернетжурналы (снова файлы в Интернете) позволяют восстановить базу данных после сбоя. Когда база данных завершена, данные могут быть получены из данных в этом файле.

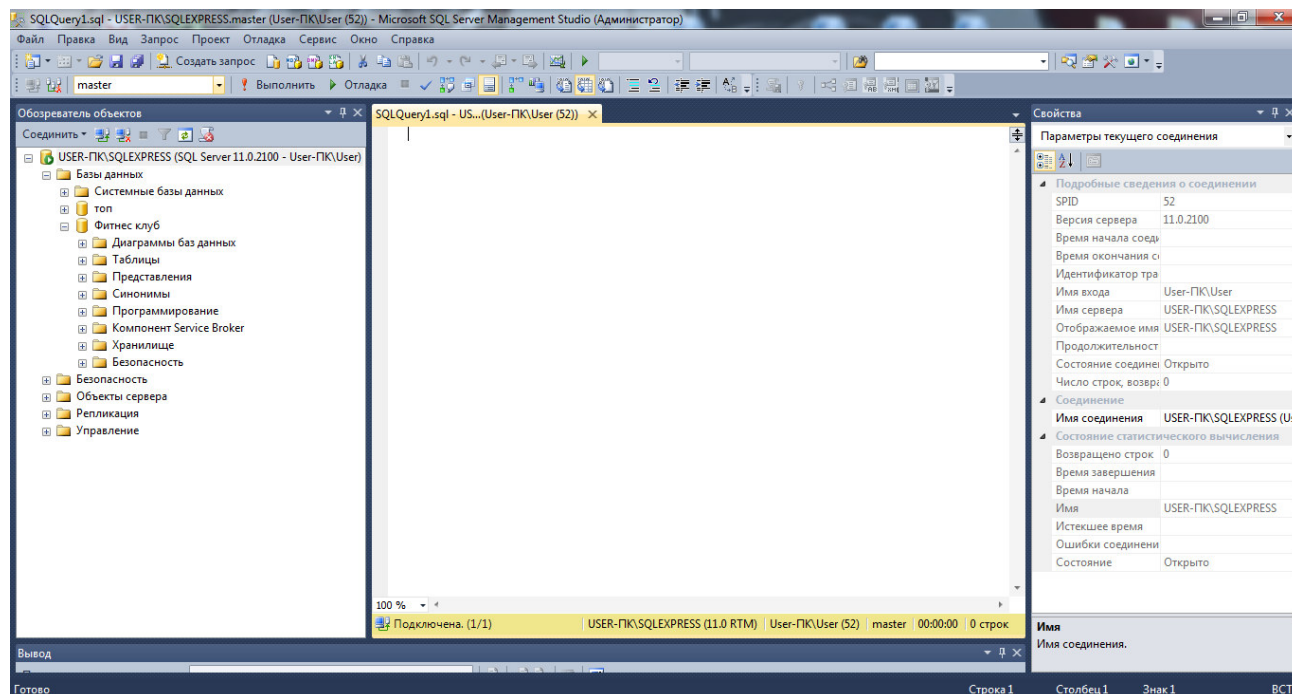


Рисунок 2.5 – Главное окно MS SQL Server Server

Так как данный инструмент хорошо интегрируется с MS SQL Server Server (рисунок 2.5), выбираем его в качестве графического инструмента SQL для разработки базы данных. MS SQL Server Server предоставляет разработчикам возможность открывать и просматривать созданные ранее структуры, а также выполнять запросы и формировать отчеты с использованием репозитория отчетов.

Для разработки программного обеспечения использовалось объектно-ориентированное, аспектно-ориентированное, императивное и структурное язык программирование Python. Он включает в себя богатую стандартную библиотеку, простоту и легко читаемый код. Язык программирование Python в данное время очень быстро набирает свою популярность. Эталонном реализации считается интерпретатор CPython, который поддерживает очень много платформ и пользоваться им без каких либо ограничений. Для выполнение своей скорости компиляции используют в проектах PyPy компиляции JIT. Так же он поддерживает динамическую типизацию, что означает при исполнении определяет тип переменной и вместе того чтобы присваивать переменную будет лучше связывать его значения с некоторыми именами. Однако большая часть библиотеки состоит из других модулей. Есть много способов разбирать эту коллекцию. Некоторые модули написаны на С и

встроены в интерпретатор Python'a, другие написаны на Питоне и импортированы в форме исходников. Некоторые модули обеспечивают интерфейс к высокоуровневым возможностям Питона, таким как печать трассировки стека; доступ к специфичным функциям операционной системы, таким как доступ к определенному оборудованию. Для того чтобы создать интерфейс программного обеспечения на Python, потребовалась установить библиотеку RQT5. Имеется два различных интерфейса, один для администратора, а второй для тренера. Администратор имеет право добавить, изменить или удалить клиента и сотрудника. У тренера есть ограничения он может только управлять своими клиентами, может добавить, изменить или удалить.

2.4 Моделирование бизнес-процессов

Фитнес клуб работает в соответствии с законами РК и положениями, которые разработало администрация фитнес клуба. Эти ограничения действуют на протяжении всей его деятельности.

На входе процесса Организация деятельности фитнес клуба – Запрос клиента. Это означает, что вся его деятельность связана с работой со клиентами. На выходе процесса Абонемент и Договор об оказании услуг. Механизмы, которыми управляется процесс: Персонал, Оборудование и Информационная система (ИС).

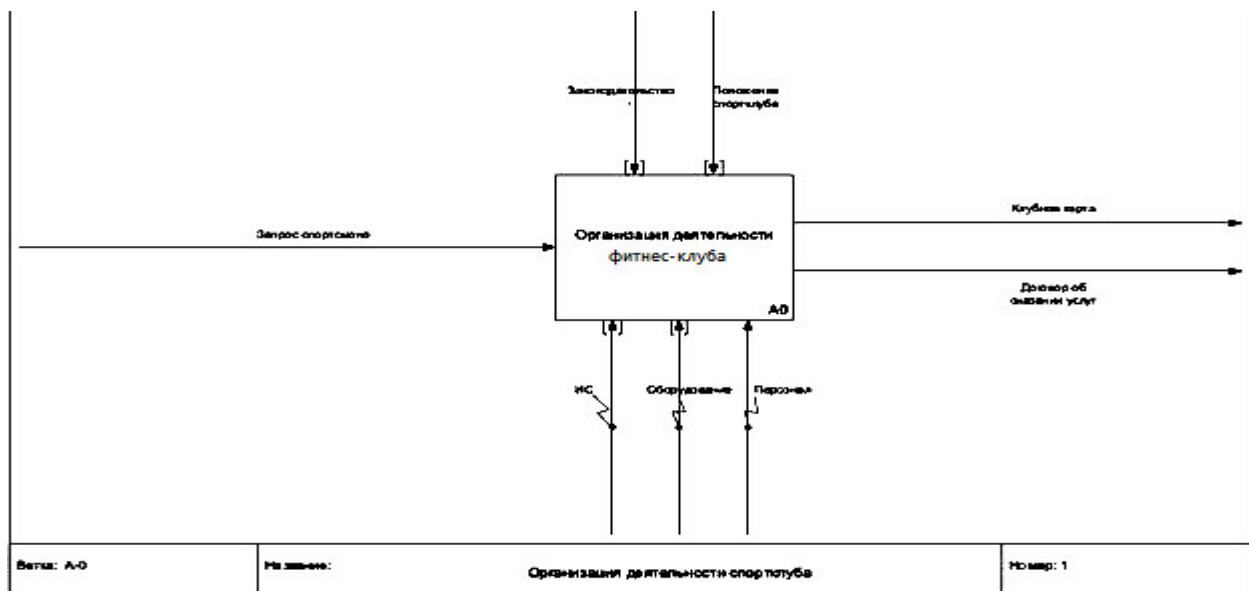


Рисунок 2.6 - Контекстная диаграмма IDEF0

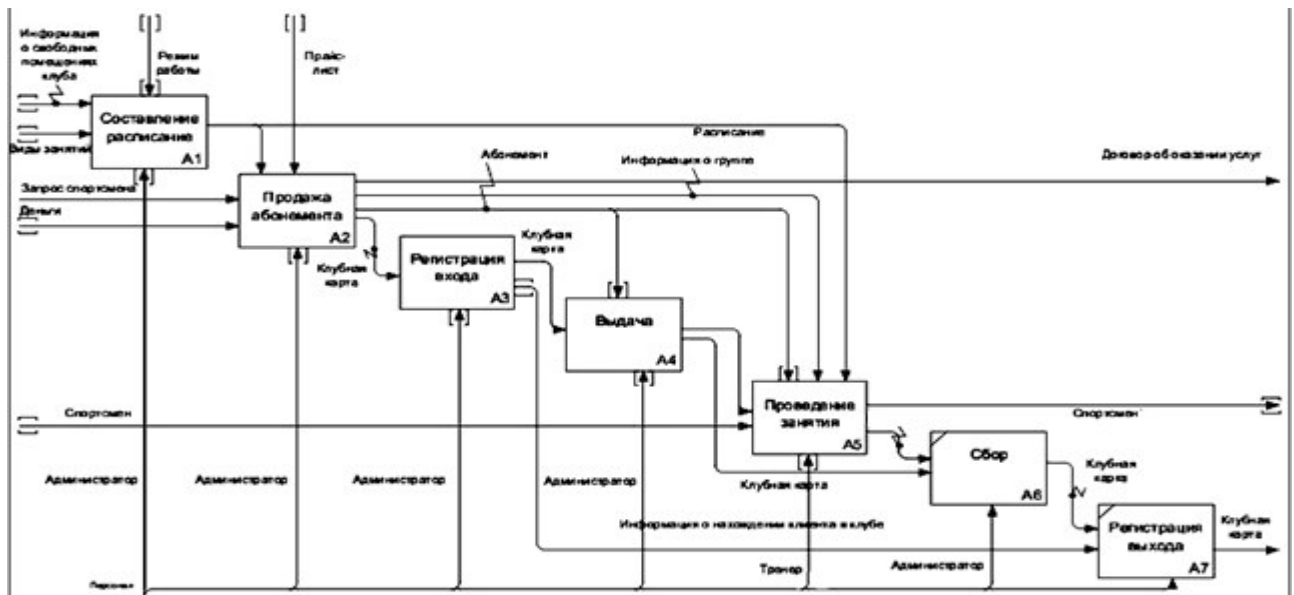


Рисунок 2.7 - Декомпозиция контекстной диаграммы IDEF0

Подпроцесс Выбор вида занятия (A11) имеет на входе объект Виды занятий. На выходе – Информацию о выбранном виде занятий.

Подпроцесс Выбор тренера (A12) имеет на входе объект Виды занятий. На выходе – Информацию о выбранном тренере.

Подпроцесс Выбор группы (A13) имеет на входе объект Информация о тренере. На выходе – Информацию о группе, для которой составляется расписание.

Под процесс Выбор даты занятия (A14) имеет на входе объект Информация о группе. На выходе – Информацию о выбранной дате занятия. Подпроцесс Выбор помещения (A15) имеет на входе объекты:

- Информация о выбранной дате и Информацию о свободных помещениях фитнес центр. На выходе – Информацию о выбранном виде занятий;

- Подпроцесс Ввод времени окончания (A16) имеет на входе объект Информация о выбранном помещении. На выходе – Информацию о времени начала занятия.

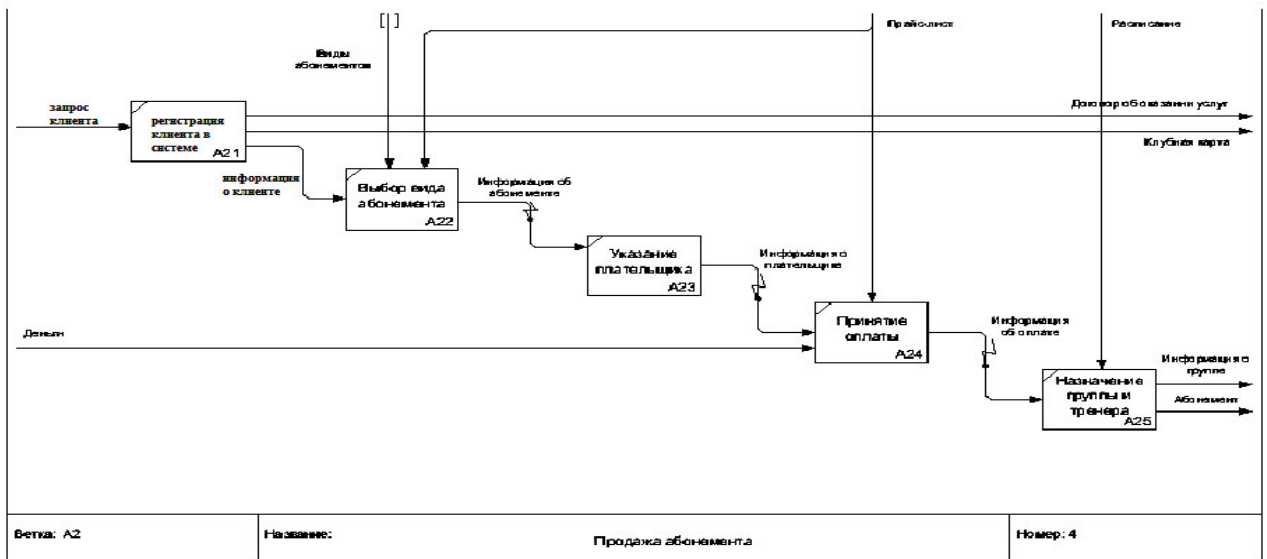


Рисунок 2.8 - Декомпозиция подпроцесса Составление расписания

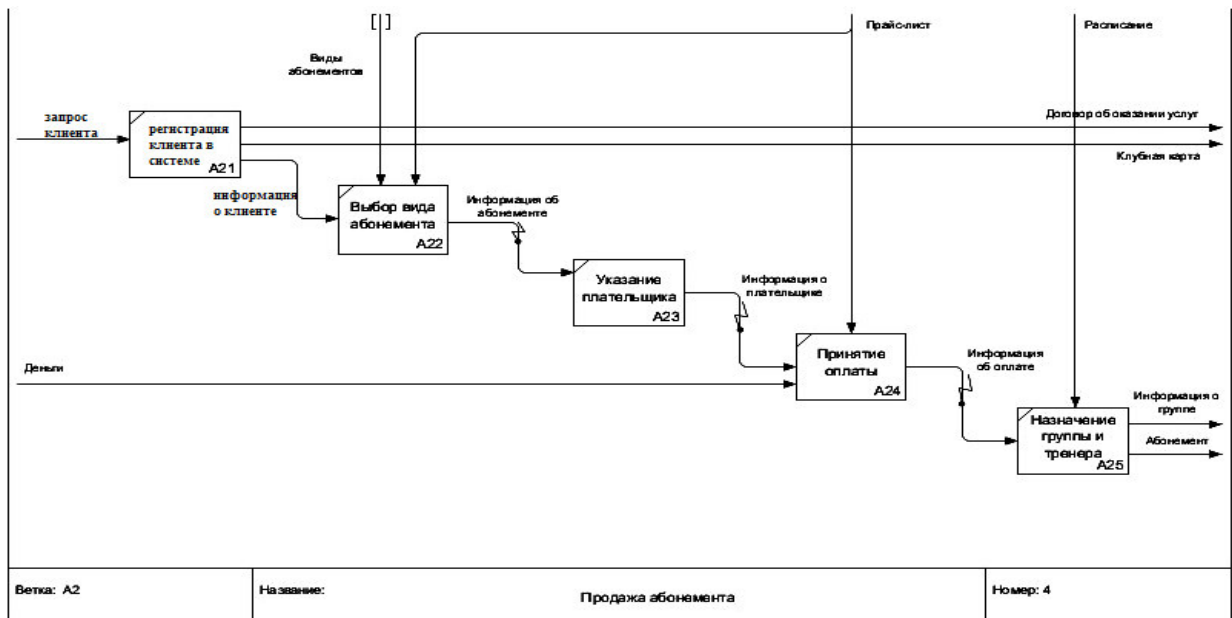


Рисунок 2.9 - Декомпозиция подпроцесса Продажа абонемента

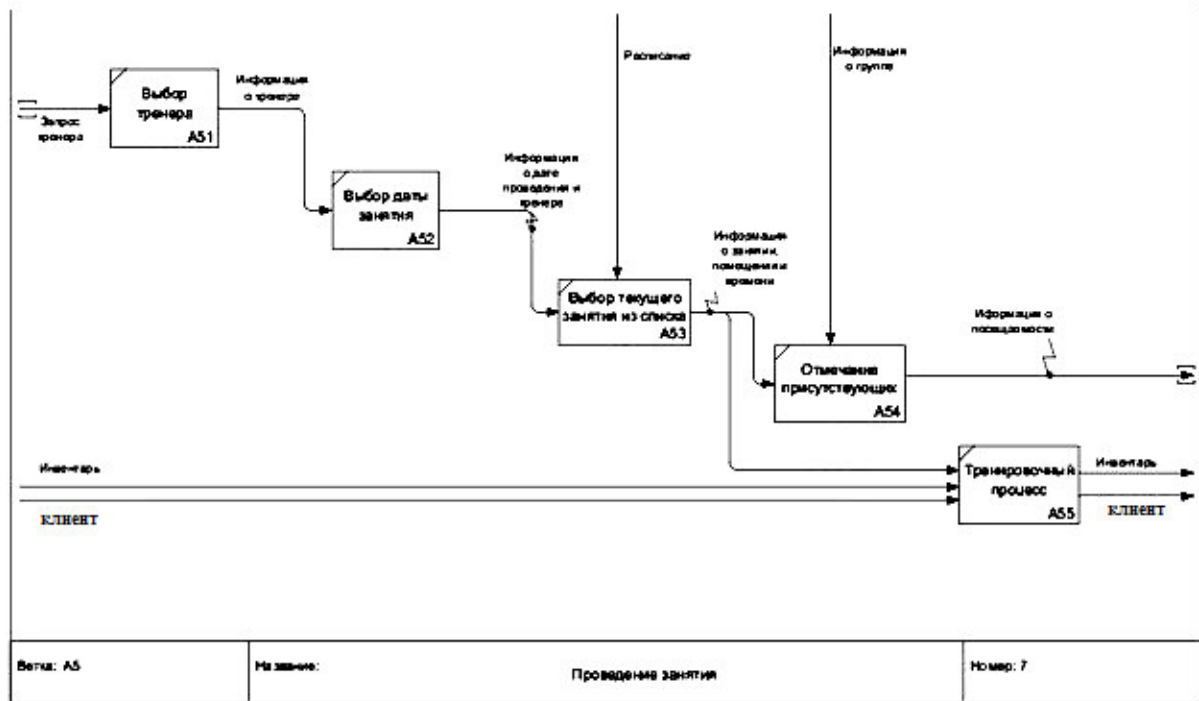


Рисунок 2.10 - Декомпозиция проведение занятия

2.5 Проектирование информационной системы с использованием UML

Информационная система предусматривает две роли пользователя:

- администратор;
- тренер.

Роль Администратора в информационной системе наиболее всеобъемлющая. Если пользователь находится в системе с этой ролью, то ему доступны все ее функции (просмотр данных о клиентах, а также задолженности и оплаты услуг, управление абонементом клиентов, продажи абонементов, составление расписания и пр.). Эту роль занимает администратор фитнес центра. Все доступные Администратору действия в системе отражены на диаграммах вариантов использования (Use case diagram), разработанные с помощью программы Rational Rose. Общая диаграмма отражает основные подсистемы Администратора, далее эта диаграмма разбивается на более подробные, отражающие возможности каждой подсистемы. Такое разбиение диаграмм связано с тем, что Администратор выполняет достаточно большое количество действий в системе. В первую очередь, Администратор занимается регистрацией посещений клиентов и продажей абонементов[8]. На диаграмме вариантов использования отражены соответствующие подсистемы: Управление абонементом и Регистрация посещения. Также основной его работой является Составление расписания и ведение базы клиентов. Для Администратора фитнес центра есть возможность

Просмотра информации о клиентах, а именно: Просмотр истории посещений, Просмотр информации об абонеентах, Просмотр личных и фитнес сведений.

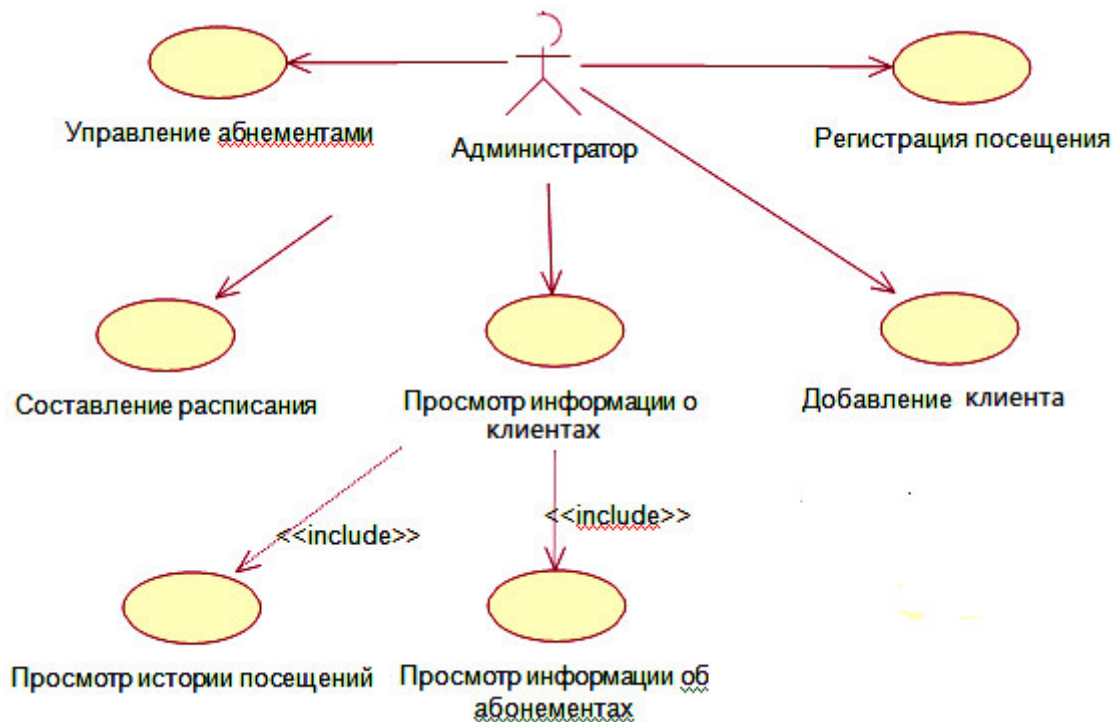


Рисунок 2.11 - Диаграмма основных подсистем администратора

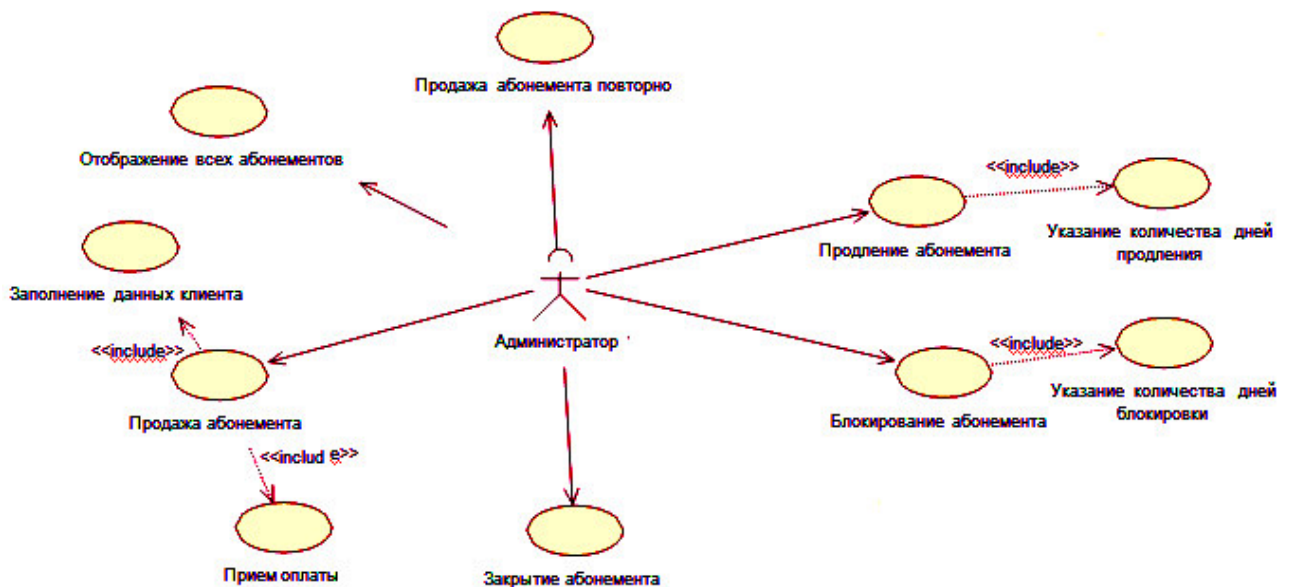


Рисунок 2.12 - Диаграмма вариантов использования подсистемы

Диаграмма последовательности «Добавление нового клиента» (рис.2.12) показывает, какие действия должен совершить Администратор для добавления клиента в справочник. На Форме «Создание клиента» он вводит

ФИО клиента, основные данные, информацию об опекунах. Если клиент состоял уже в других фитнес клубах, то Администратор вводит сведения о клиенте. Новый клиент добавляется в «Клиенты».

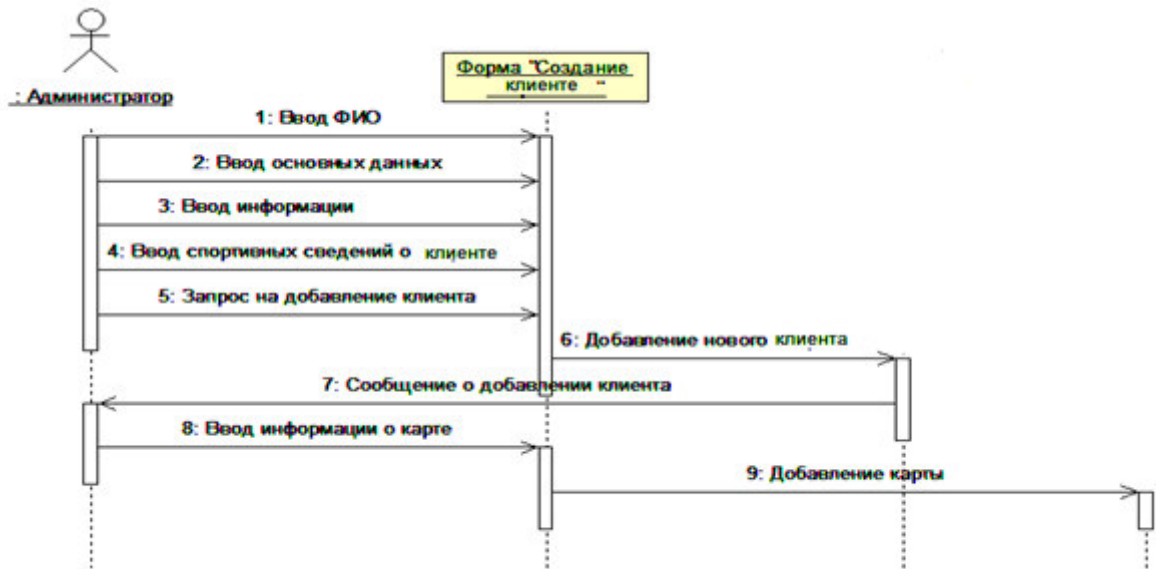


Рисунок 2.13 - Диаграмма последовательности «Добавление нового клиента»

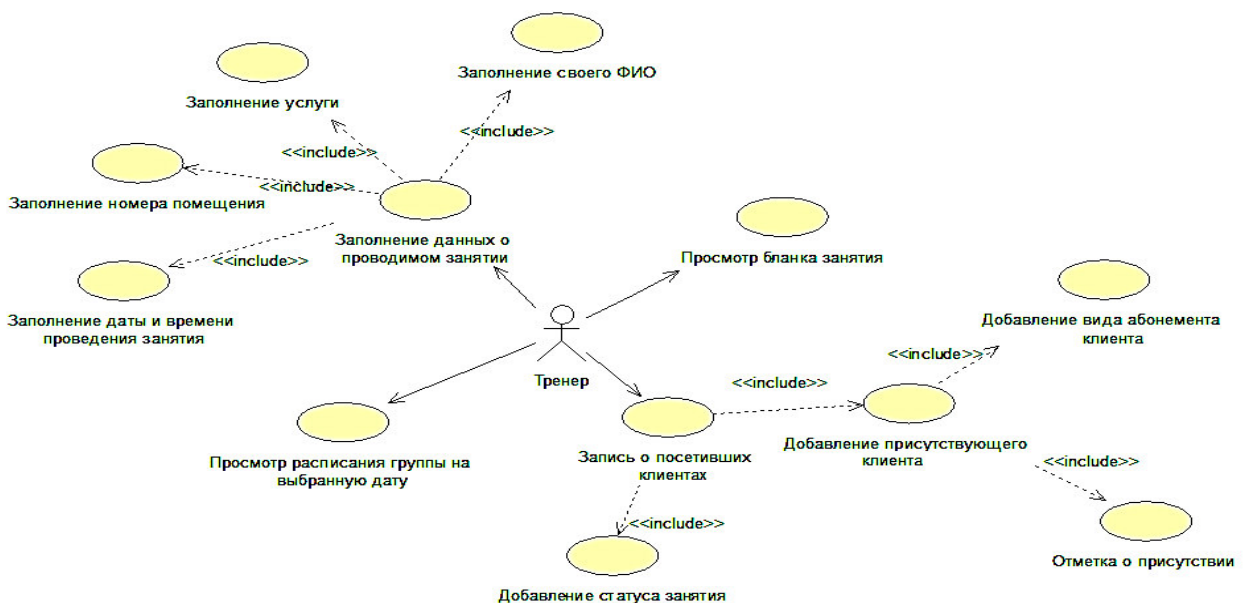


Рисунок 2.14 - Диаграмма вариантов использования тренера

ИС разделена на 3 основных функциональных блока: АРМ администратора, АРМ тренера и Расписание. Первый блок содержит в себе следующие функции:

- просмотр информации о спортсмене;
- продажа абонемента;

- изменение абонемента;
- управление инвентарем;
- регистрация посещения;
- создание спортсмена.

Второй функциональный блок АРМ тренера включает в себя 3 основные функции: заполнение данных о проводимом занятии, отметка присутствующих и просмотр бланка занятия.

Третий функциональный блок включает в себя работу с расписанием, содержащую следующие функции:

- создание занятия;
- редактирование занятия;
- выбор периода отображения;
- отображение для выбранного тренера/группы.

В целом функционирование информационной системы схематично изображено на рис. 2.15

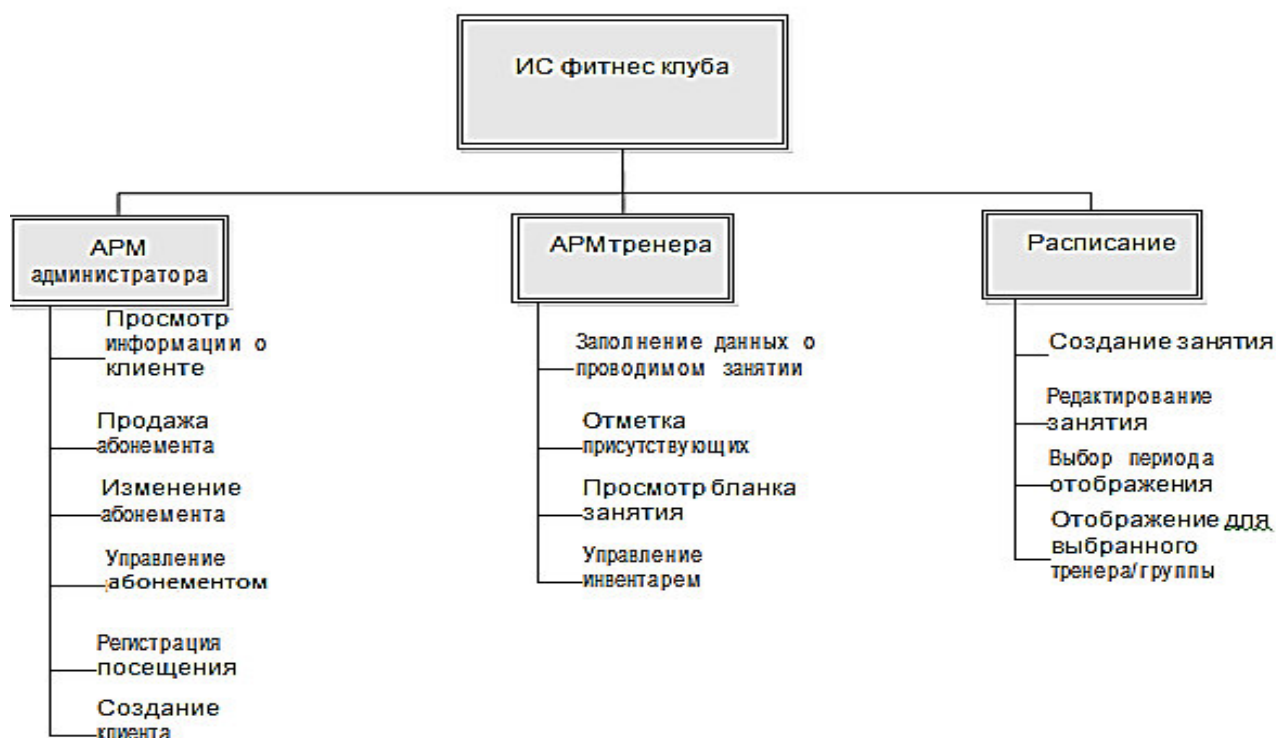


Рисунок 2.15 - Функционирование ИС

2.6 Моделирование базы данных

Администратор тренажёрного зала консультирует клиентов по возникающим вопросам, помогает подобрать наиболее подходящий вариант для клиента, регистрирует клиентов, абонементы, аренду и т.д. Администратор сидит за стойкой сразу же при входе в тренажёрный зал.

Качество разработанной БД всецело зависит от качества выполнения отдельных этапов ее проектирования. Огромное значение имеет качественная разработка логической модели данных, так как она, с одной стороны, обеспечивает адекватность базы данных предметной области, а с другой стороны, определяет структуру физической БД и, следовательно, ее эксплуатационные характеристики.

Логическая модель отражает логические связи между элементами данных вне зависимости от их содержания и среде хранения. На рисунке 2.16 приведена логическая модель базы данных.

Связи между сущностями в большей части имеют характер «многие к одному» и только в редких случаях – «один к одному». Выбор как этих сущностей с атрибутами, так и связей между ними вполне оправдан и является рациональным решением.

Выполняем преобразование логической модели в физическую (реляционную) модель.

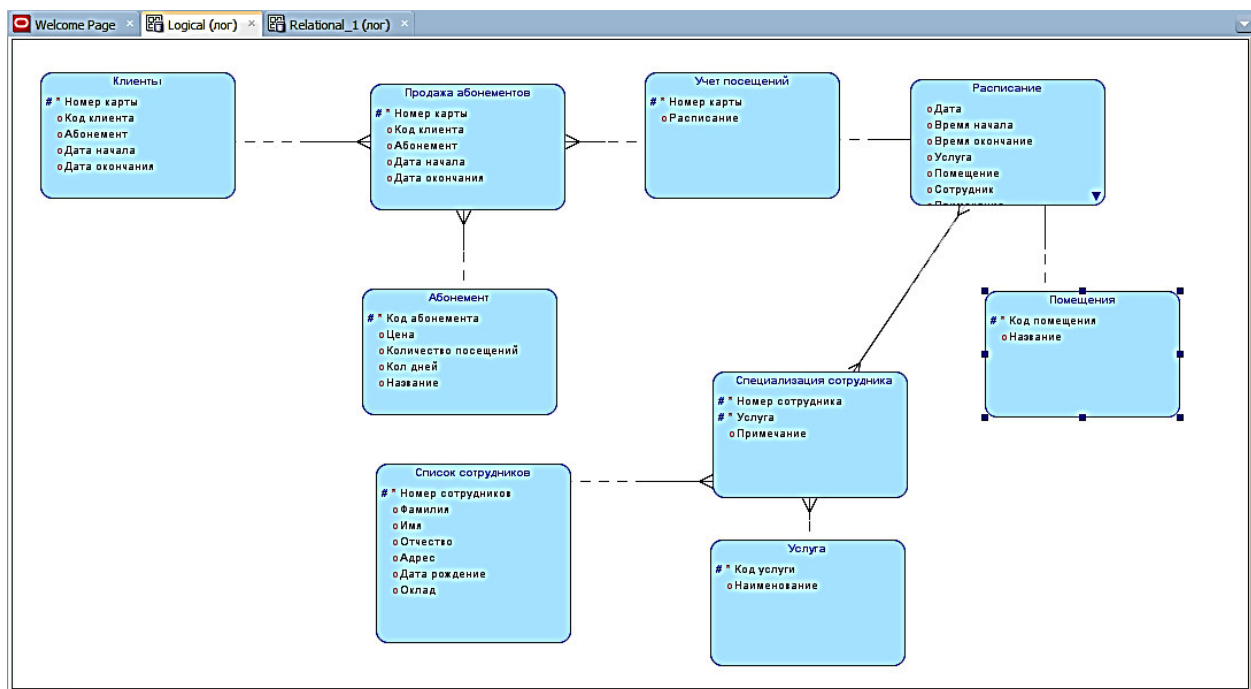


Рисунок 2.16 – Логическая модель

Реляционная модель, представленная на рисунке 2.18 отличается от логической тем, что в реляционной можно увидеть типы данных атрибутов, также наглядно представлено по каким ключевым полям связаны наши будущие таблицы (а в данном случае – сущности), какие атрибуты перейдут от родительских сущностей к дочерним и др.

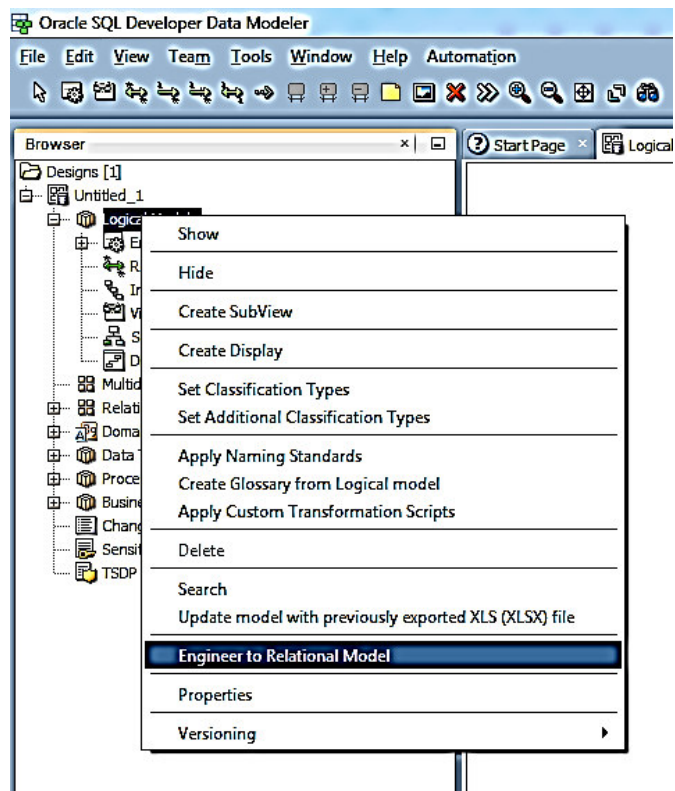
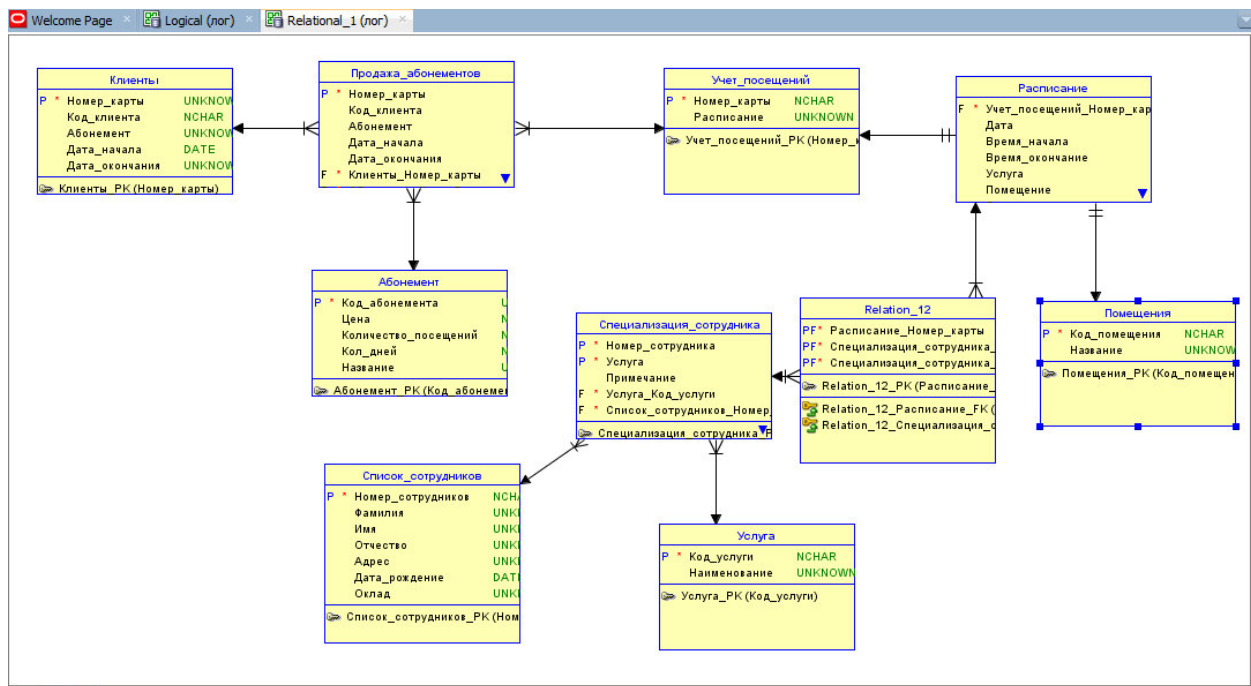


Рисунок 2.17 – Преобразование логической модели в реляционную



На рисунке 2.18 - Реляционная модель базы данных

3 Практическая часть

3.1 Создание базы данных

Для визуального проектирования архитектуры базы данных выбран инструмент MS SQL Server Server. Данная программа позволяет визуально рисовать таблицы, и строить взаимосвязи с ними. Многие ошибки проектирования архитектуры БД можно избежать при таком подходе проектирования. Это единое интегрированное решение, для управления информацией, доступное на различных открытых платформах и поддерживаемое множеством сторонних приложений. Используя MS SQL Server Server, организации смогут управлять всей корпоративной информацией, а также оперативно и уверенно адаптироваться ко все более изменчивой конкурентной среде. Чтобы предоставить такие возможности, в новом выпуске расширены уникальные механизмы Sql, обеспечивающие кластеризацию баз данных, автоматизацию центров обработки данных и управление рабочими нагрузками.

Создаем базу данных фитнес-клуба (рисунок 3.1).

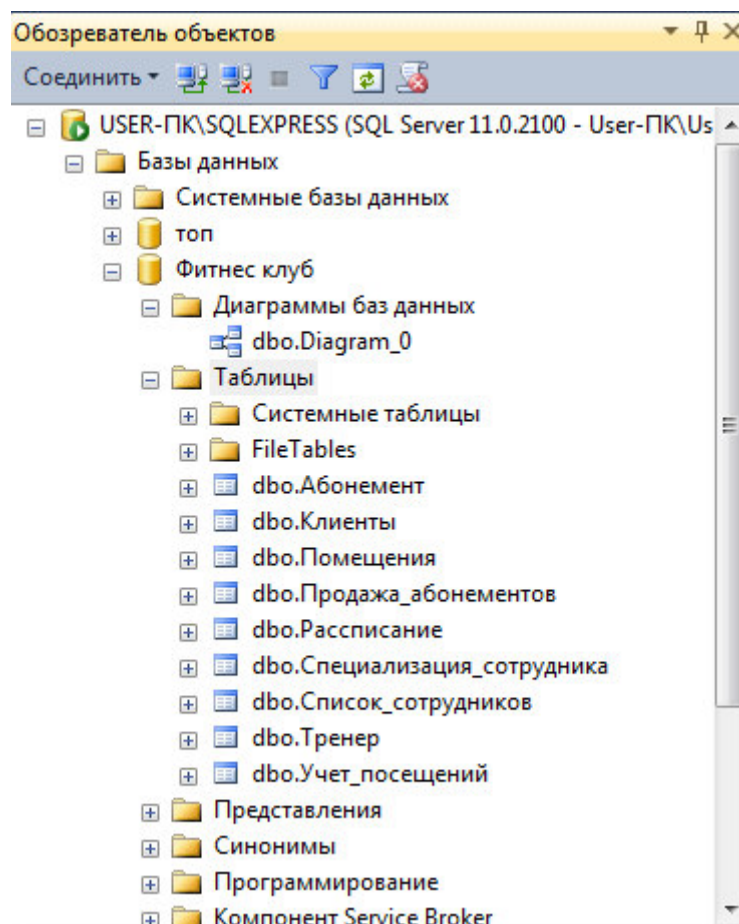


Рисунок 3.1 - Создание базы данных

	Имя столбца	Тип данных	Разрешить ...
▶	Код_клиента	int	<input type="checkbox"/>
	Имя	varchar(50)	<input checked="" type="checkbox"/>
	Фамилия	varchar(50)	<input checked="" type="checkbox"/>
	Отчество	varchar(50)	<input checked="" type="checkbox"/>
	Комментарий	varchar(50)	<input checked="" type="checkbox"/>
	Дата	date	<input checked="" type="checkbox"/>
	Город	char(25)	<input checked="" type="checkbox"/>
	Спортивная_организац...	char(25)	<input checked="" type="checkbox"/>
	Код_тренера	int	<input checked="" type="checkbox"/>
	Код_вес_категорий	int	<input checked="" type="checkbox"/>
			<input type="checkbox"/>

Рисунок 3.2 - Создание таблиц

	Имя столбца	Тип данных	Разрешить ...
▶	Код_клиента	int	<input type="checkbox"/>
	Код_абонемента	int	<input checked="" type="checkbox"/>
	Код_вида_абонемента	int	<input checked="" type="checkbox"/>
	Код_сотрудника	int	<input checked="" type="checkbox"/>
	Комментарий	varchar(50)	<input checked="" type="checkbox"/>
	Сумма	money	<input checked="" type="checkbox"/>
			<input type="checkbox"/>

Рисунок 3.3 - Создание таблиц

	Имя столбца	Тип данных	Разрешить ...
▶	номер_сотрудникоа	int	<input checked="" type="checkbox"/>
	Фамилия	nchar(10)	<input checked="" type="checkbox"/>
	Имя	varchar(50)	<input checked="" type="checkbox"/>
	Отчество	varchar(50)	<input checked="" type="checkbox"/>
	Адрес	varchar(50)	<input checked="" type="checkbox"/>
	Дата_рождения	varchar(50)	<input checked="" type="checkbox"/>
	Оклад	varchar(50)	<input checked="" type="checkbox"/>
			<input type="checkbox"/>

Рисунок 3.4 - Создание таблиц

3. 2 Информационное наполнение

После создания структуры таблиц следующим этапом является их заполнение данными (рисунки 3.5-3.7) .

	Код_вида_або...	Длительность_...	Длительность_...	Длительность_...	Допустимы_...	Количество_заморозки	Дней_заморо...	Групповой
1	12 занятий	30 дней	1 месяц	да	3 заморозки абонеента	3 дня	нет	
2	12 занятий	30 дней	1 месяц	да	3 заморозки абонеента	3 дня	нет	
3	Без ограничен...	30 дней	1 месяц	да	6 заморозок абонеента	6 дней	нет	
4	Танцы	30 дней	1 месяц	нет	нет	нет	да	
5	Бокс	30 дней	1 месяц	нет	нет	нет	дп	
6	Дзюдо	30 дней	нет	нет	нет	нет	да	
»*	NULL	NULL	NULL	NULL	NULL	NULL	NULL	

Рисунок 3.5 - Заполнение таблиц

	Код_клиента	Имя	Фамилия	Отчество	Комментарий	Дата	Город	Спортивная_о...	Код_тренера
▶	1124	Александр	Ким	Сергеевич	Ходит по вече...	2019-01-08	Алматы	... МахГум	... 2342
	1126	Адлет	Бейсембаев	Дулатов	Просит перене...	2019-01-06	Алматы	... МахГум	... 2345
	1129	Никита	Поддубнов	Игорев	NULL	2019-01-09	Алматы	... МахГум	... 2347
	1132	Маулен	Тазыбай	Арман	Просит перене...	2019-01-11	Алматы	... МахГум	... 2330
	1136	Кайрат	Аскарлов	Арманулы	NULL	2019-01-13	Алматы	... МахГум	... 2355
*	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

Рисунок 3.6 - Заполнение таблиц

	Код_клиента	Код_абонеме...	Код_вида_або...	Код_сотрудни...	Комментарий	Сумма
▶	1124	1	2	2246	NULL	6000,0000
	1126	2	2	2246	NULL	6000,0000
	1129	3	2	2246	NULL	6000,0000
	1132	4	2	2246	NULL	6000,0000
	1136	5	2	2246	NULL	6000,0000
*	NULL	NULL	NULL	NULL	NULL	NULL

Рисунок 3.7 - Заполнение таблиц

3.3. Выполнение запросов к базе данных

После создания и заполнения таблиц выполним несколько запросов.

```

USER-ПК\SQLXP...луб - dbo.Тренер  USER
|
| select * from тренер
| where Код_тренера = '2330'|

```

	Код тренера	Сотрудник
1	2330	Рамзан

Рисунок 3.8 - Запрос в MS SQL Server

USER-ПК\SQLXPRES...уб - dbo.Клиенты USER-ПК\SQLXPRES...луб - dbo.Тренер SQLQuery6.sql - нет соединения* USER-ПК\SQLXPRES...клуб - Diagram_0*

```

select *
  /*b.код_клиента,
   b.дата,
   b.код_вес_категорий,
   t.код_тренера
   q.код_сотрудника
   q.фио
   */
from клиенты b
 join выдача_инвентаря c
 on b.код_клиента = c.код_клиента
 left join тренер t
 on t.код_тренера = b.код_тренера
 join сотрудник q
 on c.код_сотрудника = q.код_сотрудника
 left join Вес_категории l
 on l.Код_вес_категорий = b.Код_вес_категорий
 where b.код_вес_категорий like '3267'
 and b.дата >= '2019-01-08'

```

100 %

	Код_клиента	Контрагент	Комментарий	Дата	Город	Спортивная_организация	Код_тренера	Код_вес_категорий	Номер_док_выд_инвент	Код_сотрудника	Код_...
1	1129	NULL	NULL	2019-01-09	Алматы	МахГум	2347	3267	1039	2236	1129

Рисунок 3.9 - Запрос в MS SQL Server

USER-ПК\SQLXPRES...клуб - Diagram_0* SQLQuery2.sql - US...(User-Г

```

select * from Сотрудник
 where Код_сотрудника = '2246'

```

	Код_сотрудника	ФИО	Дата_рождения	Пол	Должность
1	2246	Ким Александр Сергеевич	1997-04-06	муж	Администратор

Рисунок 3.10 - Запрос в MS SQL Server Server

3.4 Реализация программного интерфейса

Для разработки программного обеспечения использовалась библиотека RQT5 для Python. В этой библиотеке есть много возможностей, одно из них создавать интерфейс и связывать кнопки, таблицы не обращаясь к коду. Но чтобы назначать кнопки конечно нам потребуется прописать скрипты для их реализации. Используем QT Designer:

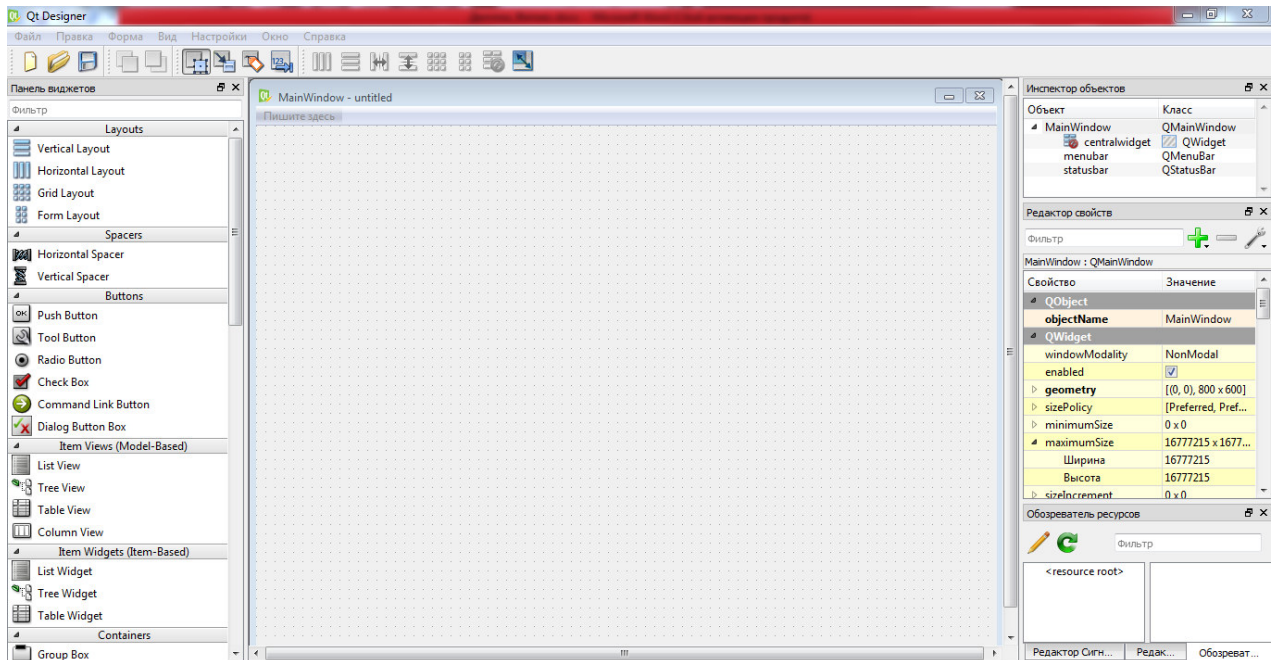


Рисунок 3.11 - Главное окно QT Designer

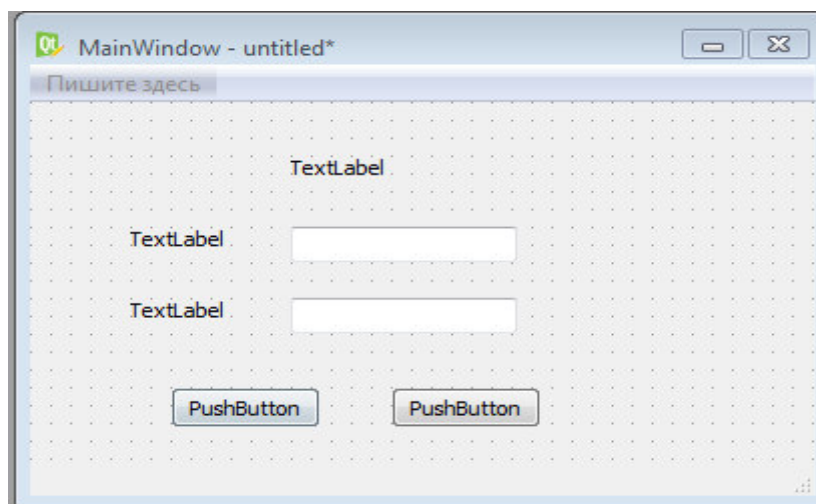


Рисунок 3.12 - Создание интерфейса Вход

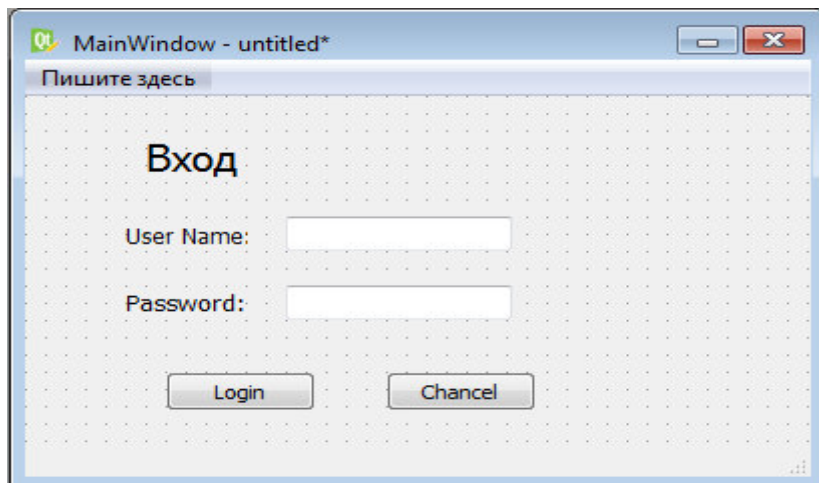


Рисунок 3.13 - Назначение свойства

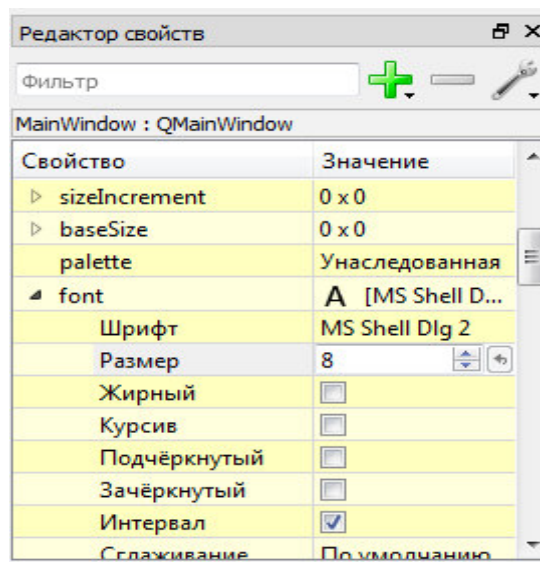


Рисунок 3.14 - Редактор свойств


```

login.py
1  from PyQt5 import QtCore, QtGui, QtWidgets
2  from database import Db #importing database.py
3  from home import Ui_MainWindow
4  from signup import Ui_Dialog
5  import Projects_Gym2
6
7
8  class Ui_Dialog2(QtWidgets.QDialog):
9
10     self.label = QtWidgets.QLabel(Dialog)
11     self.label.setGeometry(QtCore.QRect(130, 160, 131, 21))
12     self.label.setObjectName("label")
13     self.label_2 = QtWidgets.QLabel(Dialog)
14     self.label_2.setGeometry(QtCore.QRect(130, 190, 151, 21))
15     self.label_2.setObjectName("label_2")
16     self.txtUsername = QtWidgets.QLineEdit(Dialog)
17     self.txtUsername.setGeometry(QtCore.QRect(300, 160, 191, 27))
18     self.txtUsername.setObjectName("txtUsername")
19     self.txtPassword = QtWidgets.QLineEdit(Dialog)
20
21     ##### make the password invisible #####
22     self.txtPassword.setEchoMode(QtWidgets.QLineEdit.Password)
23     #####
24     self.txtPassword.setGeometry(QtCore.QRect(300, 190, 191, 27))
25     self.txtPassword.setObjectName("txtPassword")
26
27     self.btnLogin = QtWidgets.QPushButton(Dialog)
28     self.btnLogin.setGeometry(QtCore.QRect(210, 250, 71, 41))
29     self.btnLogin.setObjectName("btnLogin")
30     ##### Login Button funtion #####
31     self.btnLogin.clicked.connect(self.loginCheck)
32
33     #####
34     self.btnSignup = QtWidgets.QPushButton(Dialog)
35     self.btnSignup.setGeometry(QtCore.QRect(290, 250, 81, 41))

```

Рисунок 3.15 - Скрипт интерфейса Вход



Рисунок 3.16 - Интерфейс Вход

Создаем интерфейс программы для администратора через PQT5. В главном меню находим List widget и ставим по порядку в три столбца. Первый лист нам нужен для просмотра всех сотрудников, второй лист нужен для просмотра всех клиентов, а третий для календаря. Первые две таблицы будем связывать в дальнейшем с базами данных и будет отображать. Так же найдем в панели виджетов Label он нужен для заголовков list widget. Для третьего листа нужно найти в панели Date edite показана вся панель виджетов на рис. 3.17 и получаем первые эскизы интерфейса администратора программного обеспечения.

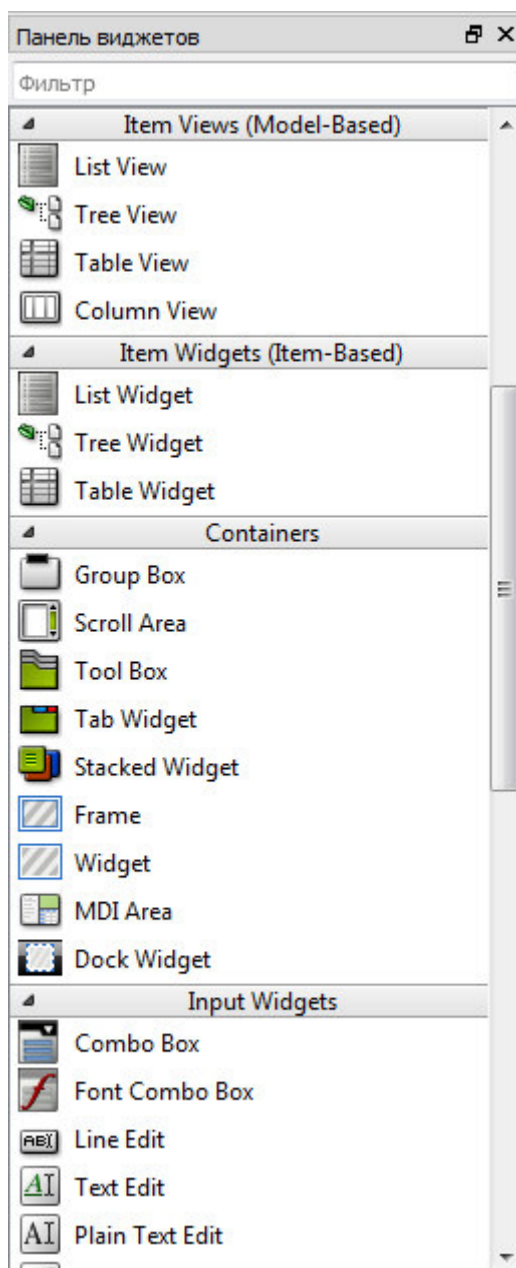


Рисунок 3.17 – Панель виджетов

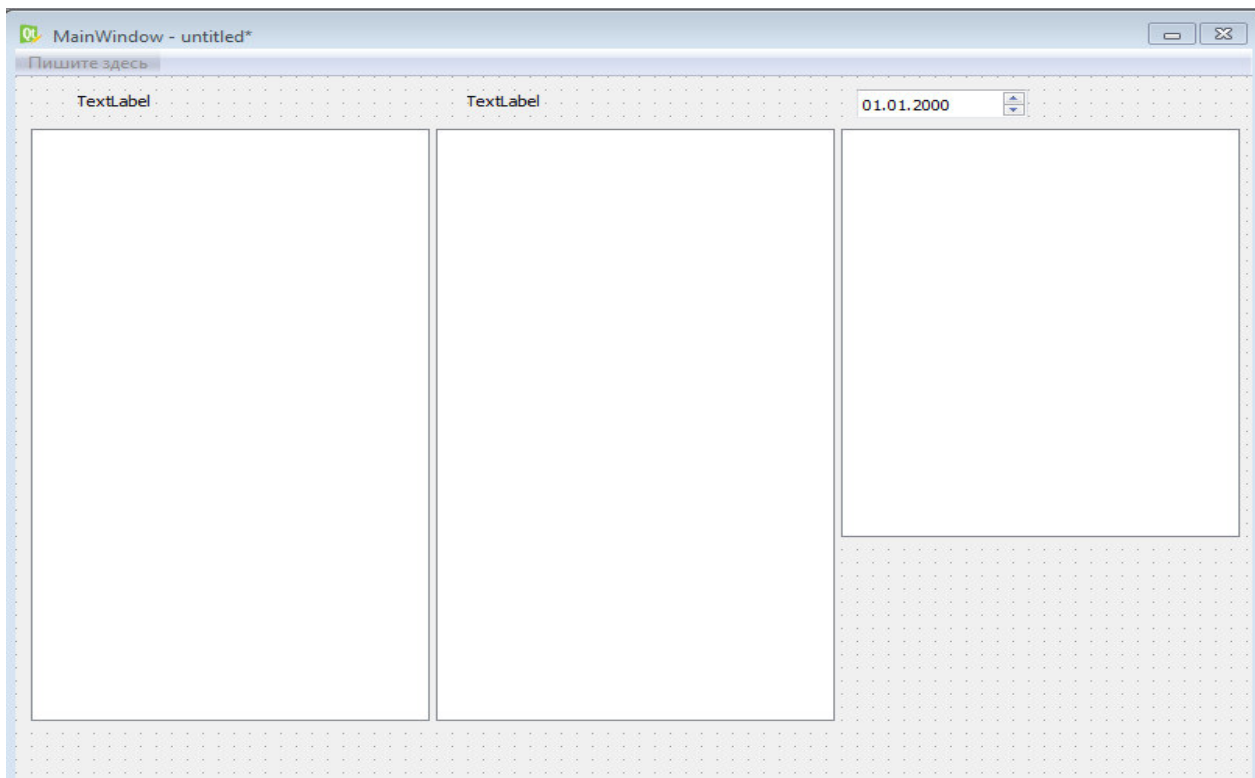


Рисунок 3.18 – Разработка интерфейса администратора

Далее присваиваем имена для кнопки и дать название. Находим виджет Push button и ставим в Mainwindow для реализации кнопок добавить клиента, изменить или удалить. Так же необходимы кнопки создание нового клиента, изменить или удалить. Смотреть на рис. 3.19

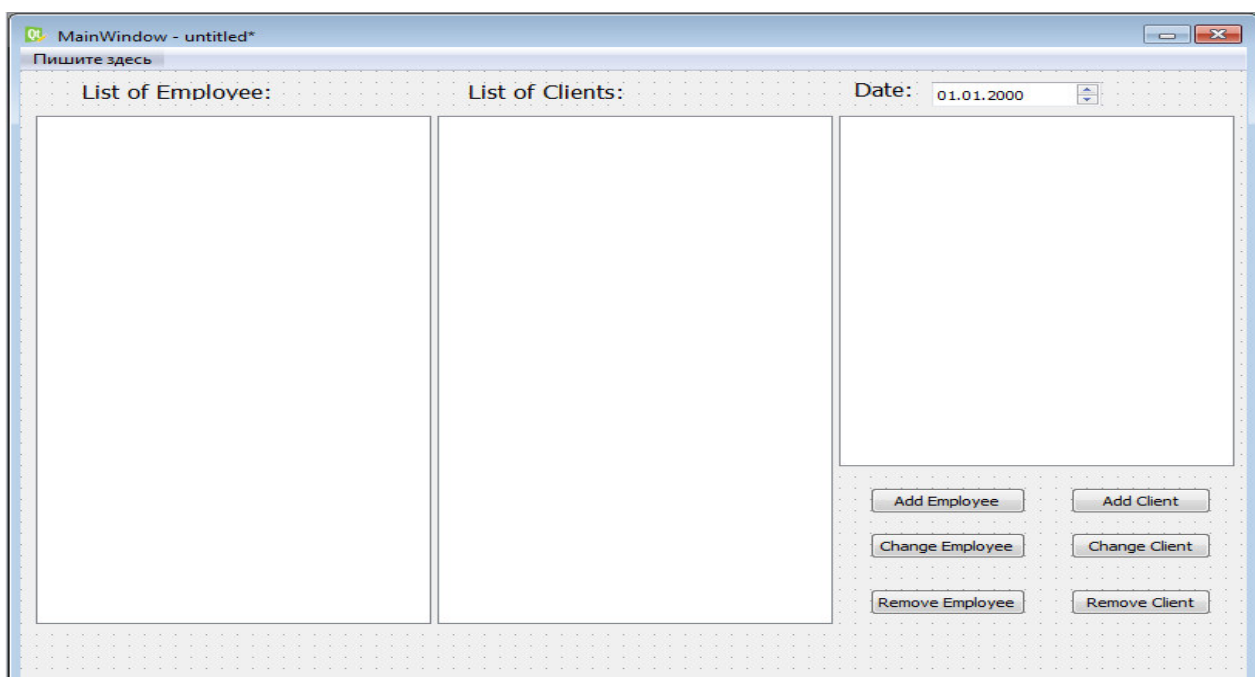


Рисунок 3.19 – Реализация кнопок и заголовков интерфейса

В главном меню Gt Designer открываем редактор свойств QMainWindow и там можно будет выбрать свойства. Изменяем шрифт кнопок, размер, цвет интерфейса. Редактор свойств QMainWindow показан на рис. 3.20

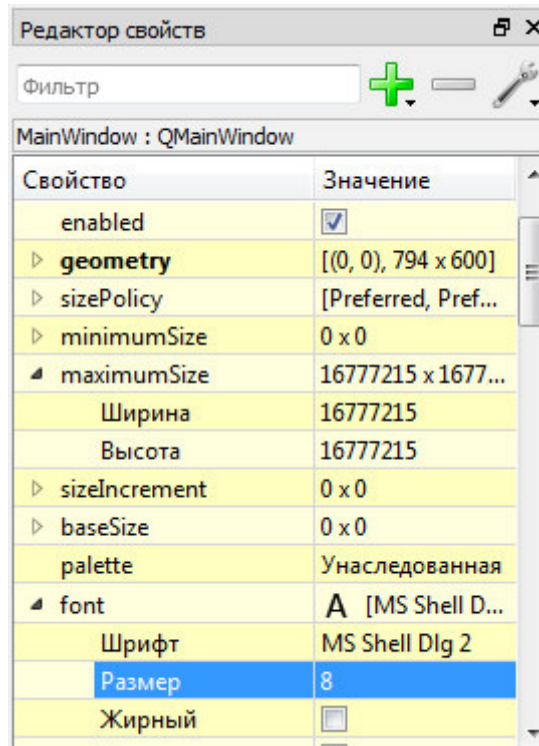


Рисунок 3.20 – Редактор свойств

Далее сохраняем файл и даем название и конвертируем из формата Project_Gym2.ui в Project_Gym2.PY, через блокнот создаем в формате (converter.bat). Для того чтобы запустить файл в Python. Показано на рис. 3.21

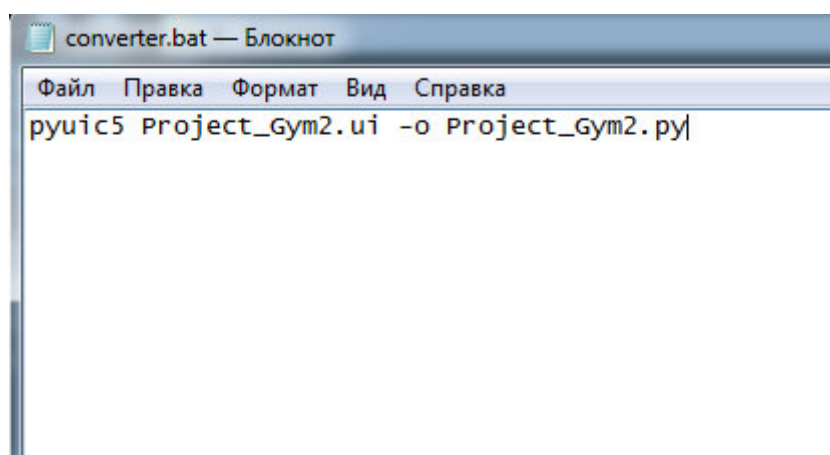


Рисунок 3.21 – Блокнот в формате (converter.bat)

После открытие файла в наш List of Employees связываем с нашей базой сотрудников которые создавалось ранее MS SQL Server Server. Показано на рис. 3.22

```
import sqlite3
# import MySQLdb as mdb
import pymysql as mdb

class Db:
    def __init__(self):
        self.db = mdb.connect(host="localhost", user="cbuser", passwd="1234567", db="sports")
        #

    def insertEmployeeTable(self, id_coach, id_employee, Firstname, Lastname, Patronymic, Phone, Address, City, Course, Date_registered, Gender, Login, Password):

        self.db = mdb.connect(host="localhost", user="cbuser", passwd="1234567", db="sports")

        self.id_coach = id_coach
        self.id_employee = id_employee
        self.Firstname = Firstname
        self.Lastname = Lastname
        self.Patronymic = Patronymic
        self.Phone = Phone
        self.Address = Address
        self.City = City
        self.Course = Course
        self.Date_registered = Date_registered
        self.Gender = Gender
        self.Login = Login
        self.Password = Password

        c = self.db.cursor()

        c.execute("""INSERT INTO employee2 (id_coach,
            id_employee,
            FirstName,
            Lastname,
```

Рисунок 3.22 – Скрипт для List of Employees

Так же связываем с базой данных List of Clients для получение информации наших клиентов. Показано рис. 3.23

```
self.db.commit()
self.db.close()

def insertClientTable(self, id_client, Firstname, Lastname, Patronymic, Phone, Address, City, Course, id_coach, id_subscription, Gender):

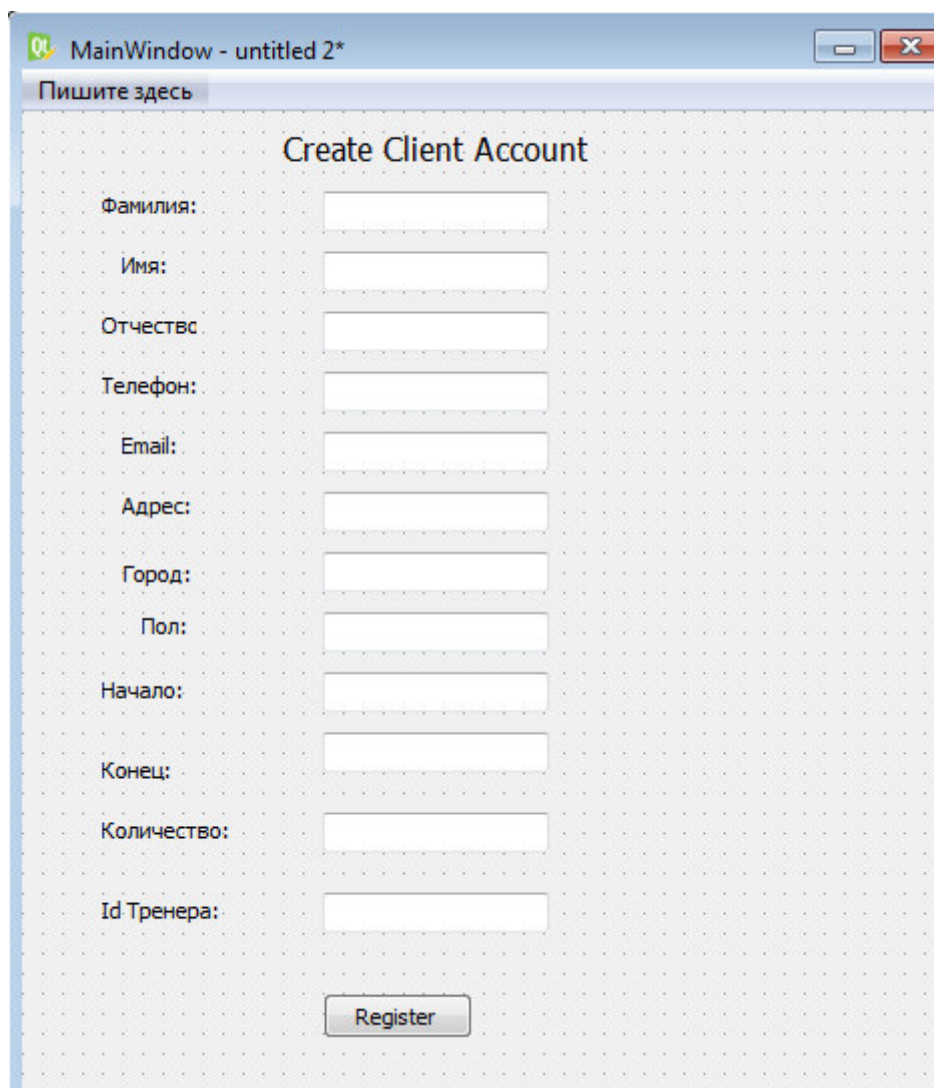
    self.db = mdb.connect(host="localhost", user="cbuser", passwd="1234567", db="sports")

    self.id_client = id_client
    # self.id_employee = id_employee
    self.Firstname = Firstname
    self.Lastname = Lastname
    self.Patronymic = Patronymic
    self.Phone = Phone
    self.Address = Address
    self.City = City
    self.Course = Course
    self.id_coach = id_coach
    self.id_subscription = id_subscription
    # self.Date_registered = Date_registered
    self.Gender = Gender
    # self.Login = Login
    # self.Password = Password

    self.ccc = self.db.cursor()

    self.ccc.execute("INSERT INTO client VALUES(%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s)" % (
        str(self.id_client),
        str(self.Firstname),
        str(self.Lastname),
        str(self.Patronymic),
        str(self.Phone),
        str(self.Address),
        str(self.City),
```


Рисунок 3.25 – Новое окно для добавление сотрудников
Присвоим название ярлыкам и кнопки в редакторе свойств. Показано на рис. 3.26



The screenshot shows a window titled "MainWindow - untitled 2*" with a standard Windows-style title bar. Below the title bar is a menu bar with the text "Пишите здесь". The main content area has a light gray background with a dotted grid pattern. At the top center of this area is the text "Create Client Account". Below this text are several input fields, each with a label to its left: "Фамилия:", "Имя:", "Отчество:", "Телефон:", "Email:", "Адрес:", "Город:", "Пол:", "Начало:", "Конец:", "Количество:", and "Id Тренера:". Each label is followed by a white rectangular input box. At the bottom center of the form is a button labeled "Register".

Рисунок 3.26 – Окно для добавление сотрудников

Далее конвертируем и открываем файл в Python и назначаем кнопки для добавление нового сотрудника. Связываем с базой данных с нашей таблицей сотрудники в MS SQL Server Server с окном Change Client Account. Показано на рис. 3.27

```
def registerButton(self):
    id_coach = 5_#self.id_coach.text()
    id_employee = 5_#self.id_employee.text()
    Firstname = self.txtName.text()
    Lastname = self.txtUsername.text()
    Patronymic = self.txtPatronymic.text()
    Phone = self.txtPhone.text()
    Address = self.txtAddress.text()
    City = self.txtCity.text()
    Course = self.txtCourse.text()
    Date_registered = self.txtDateRegistered.text()
    Gender = self.txtGender.text()
    Login = self.txtLogin.text()
    Password = self.txtPassword.text()

    password = self.txtPassword.text()
    password2 = self.txtPassword2.text()
    if self.checkFields(Firstname, Lastname, Phone, password):
        self.showMessage("Error", "All fields must be filled")
    else:
        if(self.checkPassword(password, password2)):
            insertDb = Db()
            insertDb.insertEmployeeTable(id_coach, id_employee, Firstname, Lastname, Patronymic, Phone, Address, City, Course, Date_registered, Gender, Login, P
            self.showMessage("Success", "Registration successful")
            if self.flag == '1':
                self.clearField()
            else:
                self.showMessage("Error", "Passwords doesn't match")

def registerClientButton(self):
    id_client = 8_#self.id_coach.text()
    Ui Dialog > registerClientButton()
```

Рисунок 3.27 – Скрипт реализации кнопки добавление нового сотрудника

Теперь при нажатии кнопки Add Employee выводится новое окно где прописывается вся информация о новом сотруднике имя, фамилия, город, id тренера и тд. Показано на рис. 3.28

Dialog

Create Client Account

Фамилия:

Имя:

Отчество:

Телефон:

Email:

Адрес:

Город:

Курс:

Пол:

Абонимент:

Начала:

Конец:

Количество:

ID тренера:

Register

Рисунок 3.28 – Окно для создания нового сотрудника

Так же релизовались кнопки Change Employee, создавалось новое окно в Qt Designer и прописывался скрипт в Python. Показано конечный итог нового окна кнопки Change Employee на рис. 3.29

Фамилия:	Seilkhan
Имя:	Miras
Отчество:	Batyrzhanuly
Телефон:	87473259796
Email:	miras@mail.ru
Адрес:	geologov 19
Город:	Almaty
Курс:	Dzydo
Пол:	mael
Дата Регистрац.:	2019-05-22 11:21:47
Login:	joo
Password:	joo

Update

Рисунок 3.29 – Окно кнопки Change Client Account

Интерфейс для тренера создавалось таким же образом через Qt Designer. Созавалось новое окно, добавление ярлыков, кнопок PushButton, через редактор свойств изменялись свойства. Прописывался код в Python связывали с базой данных MS SQL Server Server с таблицей сотрудники. Так же реализовались кнопки как Add Client и Change Client. Далее на рис. 3.30-3.33 показано конечный итог интерфейса программы.

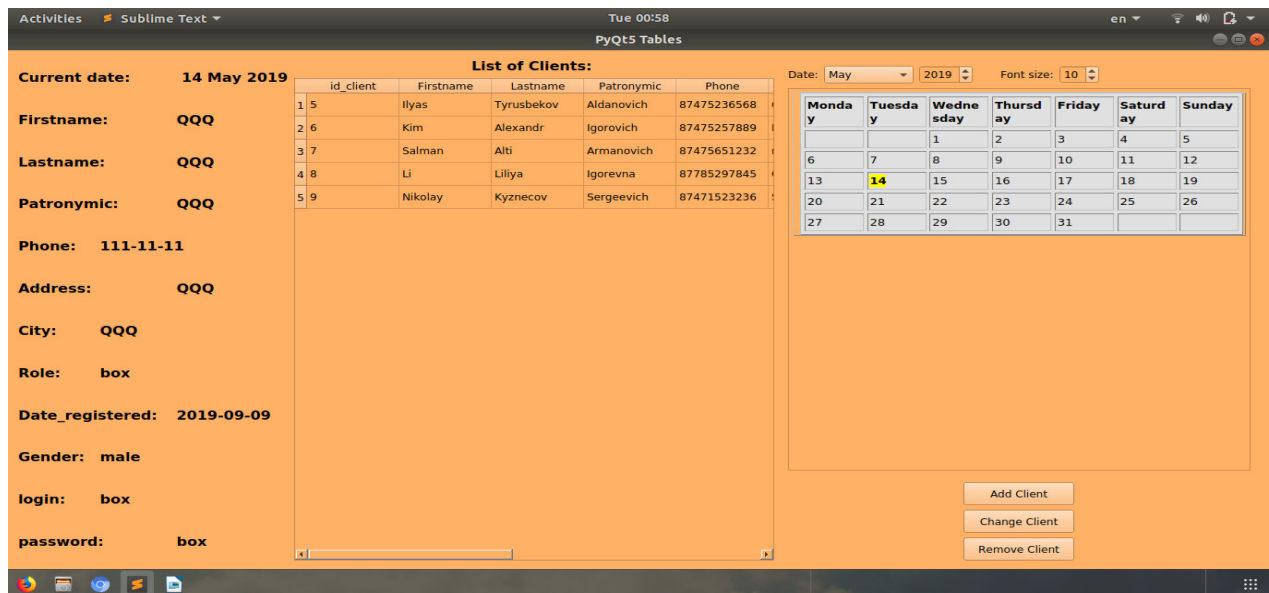


Рисунок 3.30 - Интерфейс тренера



Рисунок 3.31 – Окно кнопки Add Client

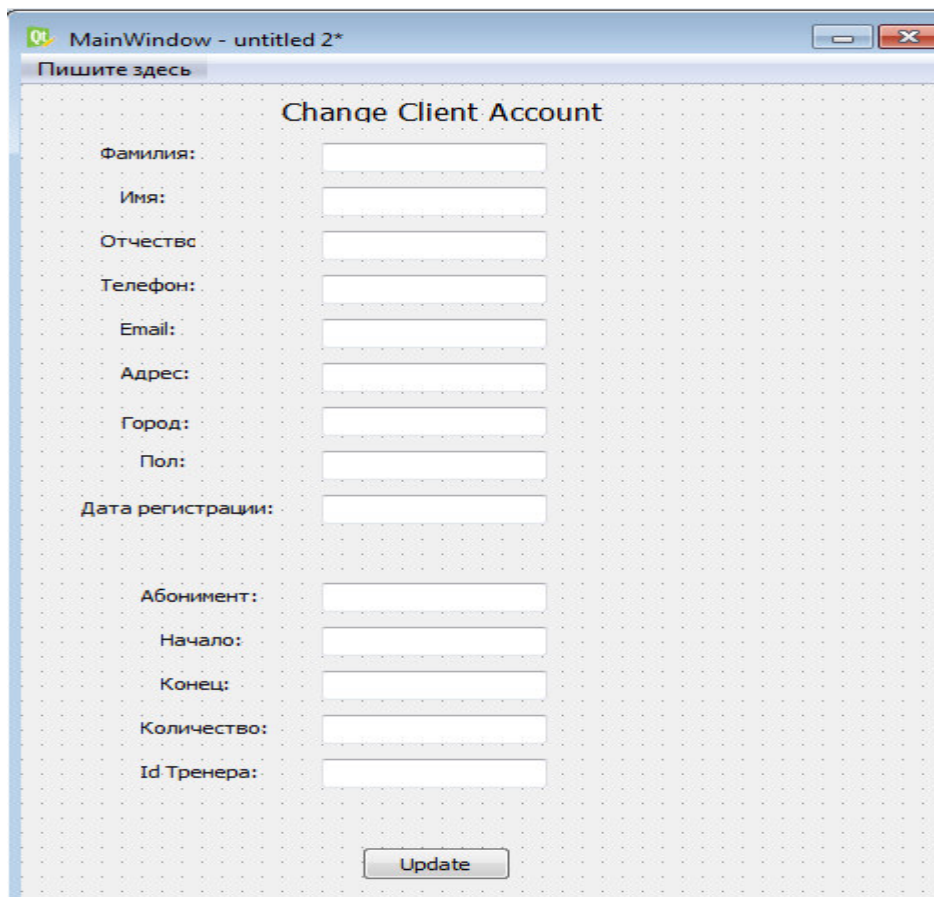


Рисунок 3.32 – Реализация кнопки Change Client



Рисунок 3.33 – Окно кнопки Change Client

4 Экономическое обоснование разработки проекта

4.1 Трудоемкость разработки программного продукта

Тема дипломного проекта – «Разработка информационной системы для автоматизации работы администратора фитнес-клуба».

Целью экономического обоснования дипломного проекта является расчет и анализ затрат, необходимых для создания и реализации информационной системы, изучить затраты на проектирование и функционирование приложения; определить условия и сроки окупаемости затрат. Обосновать рациональность проектирования данного приложения.

В данной главе производится экономический расчет коммерческой стоимости исследования проекта. Расчеты учитывают расходы на создание и внедрение информационной системы.

Для реализации проекта необходимы финансовые, материальные и трудовые ресурсы;

План проектирования и реализации проекта предусматривает 3 этапа в течении 1 месяца (4 недели).

Первый этап:

- постановка задачи;
- разработка содержания работы.

Второй этап:

- создание программного продукта.

Третий этап:

- тестирование;
- отладка;
- внедрение.

Таблица 4.1 - Распределение работ по этапам и видам и оценка их трудоемкости

Этап разработки ПП	Вид работы на данном этапе	Трудоемкость разработки ПП, чел.× ч.
Первый этап	Постановка задачи; Разработка содержания работы	1×45
Второй этап	Создание программного продукта	1×97
Третий этап	Тестирование; Отладка; Внедрение.	1×50
ИТОГО трудоемкость выполнения дипломной работы		192

4.2 Расчет затрат на разработку ПП

Таблица 4.2 – Затраты на программное обеспечение и оборудование

Наименование ресурса	Единица измерения	Количество израсходованного материала	Цена за единицу, тг	Сумма, тг
Ноутбук ACER Core i3 3320M	шт	1	69000	69000
Лазерный принтер HP Deskjet Ink Advantage 2515	шт	1	42000	42000
Программное обеспечение Python	шт	1	бесплатно	
Итого затраты на программное обеспечение и оборудование				111000

Таблица 4.3 - Материальные затраты

Наименование материального ресурса	Единица измерения	Количество израсходованного материала	Цена за единицу, тг	Сумма, тг
Бумага	шт	3	1100	3300
Картридж	шт	1	2500	2500
Итого затраты на материальные ресурсы				5800

Общая сумма затрат на материальные ресурсы (Z_m) определяется по формуле:

$$Z_m = \sum_{i=1}^n P_i \times C_i \quad (4.1)$$

где P_i - расход i -го вида материального ресурса, натуральные единицы;
 C_i - цена за единицу i -го вида материального ресурса, тг;
 i - вид материального ресурса;
 n - количество видов материальных ресурсов.

$$Z_m = \sum_{i=1}^n P_i * C_i = ((1100 \times 3) + 2500) = 5800 \text{ тг.}$$

Необходимо рассчитать затраты на электроэнергию по форме, приведенной в таблице 4.4.

Общая сумма затрат на электроэнергию ($Z_{Э}$) рассчитывается по формуле:

$$Z_{э} = \sum_{i=1}^n M_i * K_i * T_i * Ц \quad (4.2)$$

где M_i - паспортная мощность i -го электрооборудования, кВт;
 K_i - коэффициент использования мощности i -го электрооборудования (принимается $K_i=0.7, 0.9$);
 T_i - время работы i -го оборудования за весь период разработки ПП ч;
 $Ц$ - цена электроэнергии, тг/кВт*ч;
 i - вид электрооборудования;
 n - количество электрооборудования.

$$Z_{э1} = 0,45 \cdot 0,7 \cdot 192 \cdot 16,53 = 999,7 \text{ тг.} \approx 1000 \text{ тг.}$$

$$Z_{э2} = 0,33 \cdot 0,7 \cdot 2 \cdot 16,53 = 7,6 \text{ тг} \approx 8 \text{ тг.}$$

$$Z_{э} = Z_{э1} + Z_{э2} = 1000 + 8 = 1008 \text{ тг.}$$

Таблица 4.4 - Затраты на электроэнергию

Наименование оборудования	Паспортная мощность, кВт	Коэффициент использования мощности	Время работы оборудования для разработки ПП, ч	Цена электроэнергии, тг/кВт*ч	Сумма, тг
Ноутбук ACER Core i3 3320M	0,45	0,7	192	16,53	1000
Лазерный принтер HP Deskjet Ink Advantage 2515	0,33	0,7	2	16,53	8
ИТОГО затраты на электроэнергию					1008

4.3 Трудовые ресурсы, задействованные в работе

Для расчёта общей суммы затрат на оплату заработной платы $Z_{тр}$ воспользуемся формулой:

$$Z_{тр} = \sum_{i=1}^n ЧС_i \cdot T_i, \quad (4.3)$$

где n – количество разработчиков приложения;
 $ЧС_i$ - часовая ставка i -го работника, тг;
 T_i – трудоемкость разработки ПП, чел.*ч;

i - категория работника.

Трудоемкость разработки ПП определяется по данным таблицы 5.1.

Часовая ставка инженера-разработчика составляет 900 (тг/ч),
трудоемкость разработки – 192 ч.

Результаты расчёта основной заработной платы представлены в таб. 4.5

Таблица 4.5– Результаты расчёта затрат основной заработной платы

Наименование содержания работ	Исполнитель	Трудоемкость норма-час	Заработная плата за час, тг/час	Сумма ЗП, тенге
Тех. задание		9	900	8100
Моделирование		36		32400
Программирование		97		87300
Тестирование		34		30600
Внедрение		16		14400
Итого		192		172800

Также необходимо рассчитать отчисления на социальный налог, который составляет 9,5% (согласно статье 485 НК РК) от дохода работника. Социальные отчисления определим по следующей формуле:

$$З_{сзи} = (З_{тр} - З_{по}) \cdot 0,095, \quad (4.4)$$

где ЗПО – пенсионный отчисления, 10% от общего фонда оплаты труда, тенге. Рассчитаем пенсионные отчисления по следующей формуле:

$$З_{по} = З_{тр} \cdot 0,1 \quad (4.5)$$

Используя формулы 3.5 и 3.9 получаем:

$$З_{по} = 172800 \cdot 0,1 = 17280 \text{ тенге}$$
$$З_{сзи} = (172800 - 17280) \cdot 0,095 = 14774 \text{ тенге}$$

Сумма годовых амортизационных отчислений определяется по формуле:

$$A = \text{Перв.стоимость} \cdot \text{Норма амортизации}/100$$

Амортизационные отчисления приведены в таблице 4.6.

Годовые нормы амортизации ОФ принимаются по налоговому кодексу РК или определяются, исходя из возможного срока полезного использования.

$$NA_i = 100/TN_i, \quad (4.6)$$

где TN_i - возможный срок использования i -го ОФ, год.

Таблица 4.6 - Амортизация основных фондов (ОФ)

Наименование оборудования и ПО	Стоимость оборудования и ПО, тг	Годовая норма амортизации, %	Срок полезного использования оборудования и ПО, лет	Сумма амортизации в год, тг	Сумма амортизации в месяц, тг
Ноутбук ACER Core i3 3320M	69 000	20	5	13 800	1150
Лазерный принтер HP Deskjet Ink Advantage 2515	42 000	20	5	8 400	700
Программное обеспечение Android Studio	Распространяется бесплатно				
ИТОГО амортизация основных фондов					1850

$$NA_{об} = 100 / 5 = 20;$$

$$A_H = (69\,000 * 20) / 100 = 13\,800 \text{ тг};$$

$$A_M = (42\,000 * 20) / 100 = 8\,400 \text{ тг};$$

Сумма амортизации за один месяц = $A / 12$.

Сумма амортизационных отчислений за месяц равна 1850 тг.

В статью «Прочие затраты» включаются расходы на арендную плату, включая коммунальные платежи, канцелярские и прочие хозяйственные расходы. Стоимость аренды помещения на месяц равна 118 000 тг. (в эту сумму включены коммунальные услуги).

Арендная плата рассчитывается по формуле:

$$AP = Ca * S, \tag{4.7}$$

где Ca – срок аренды;

S – стоимость аренды за 1 месяц;

$$AP = 118\,000 * 1 = 118\,000 \text{ тг}.$$

Расходы на интернет, месячная оплата которого составляет 6300 тг равны:

$$P_{и} = 6300 * 1 = 6\,300 \text{ тг};$$

Прочие хозяйственные расходы составляют 5000 тг;

$$\text{Прочие затраты} = 118\,000 + 6300 + 5000 = 129\,300 \text{ тг}.$$

На основании полученных данных по отдельным статьям в таблице 4.7 приведена смета затрат на разработку ПП

Таблица 4.7 - Смета затрат на разработку ПП

Статьи затрат	Сумма, тг
Материальные затраты, в том числе:	
Материалы	5800
Электроэнергия	1008
Затраты на оплату труда.	172800
Отчисления на социальные нужды.	14774
Амортизация основных фондов.	1850
Прочие затраты.	129300
Итого по смете	325532

4.4 Определение возможной (договорной) цены ПП

Величина возможной (договорной) цены ПП должна устанавливаться с учетом эффективности, качества и сроков ее выполнения на уровне, отвечающем экономическим интересам заказчика (потребителя) и исполнителя.

Договорная цена (Цд) для прикладных ПП рассчитывается по формуле:

$$Цд = З_{\text{нир}} \times \left(1 + \frac{P}{100}\right), \quad (4.8)$$

где З_{НИР} - затраты на разработку ПП (из таблицы 4.7), тг;
 P - средний уровень рентабельности ПП. %

$$Цд = З_{\text{нир}} \times \left(1 + \frac{P}{100}\right) = 325532 \times \left(1 + \frac{20}{100}\right) = 390638 \text{ тг.}$$

Далее определяется цена реализации с учетом налога на добавленную стоимость (НДС), ставка НДС устанавливается законодательно Налоговым Кодексом РК. На 2019 год ставка НДС установлена в размере 12%.

Цена реализации с учетом НДС рассчитывается по формуле:

$$Цр = Цд + Цд \times \text{НДС}. \quad (4.9)$$

$$Цр = Цд + Цд \times \text{НДС} = 390638 + 390638 * 0,12 = 437514,56 \text{ тенге.}$$

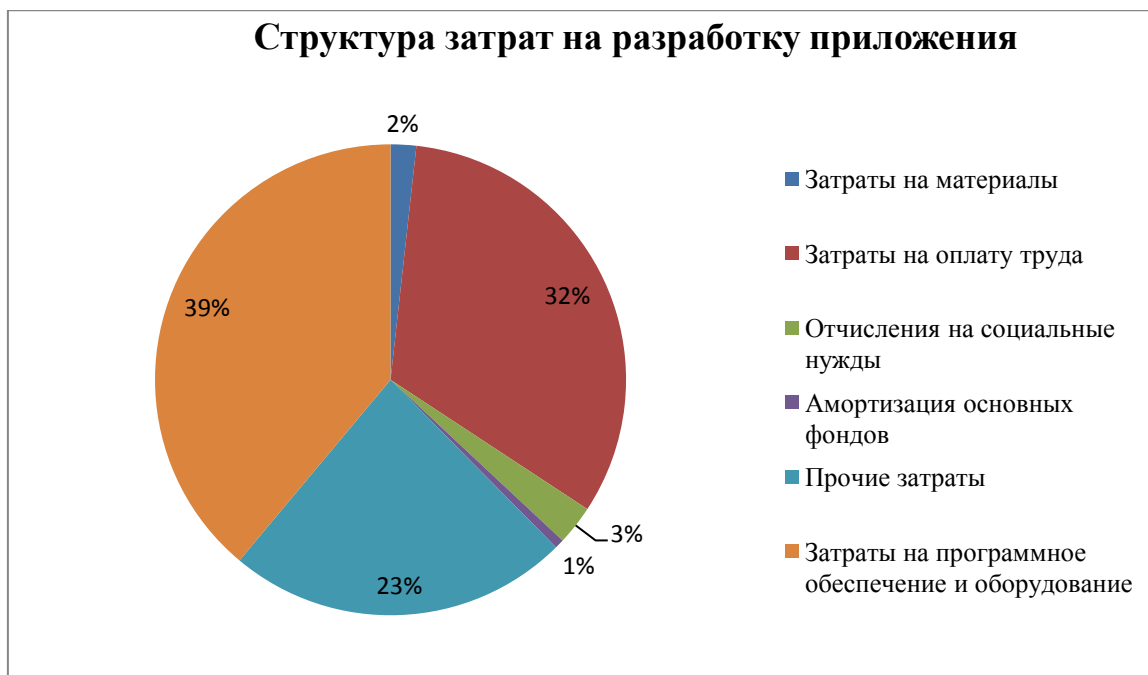


Рисунок 4.1 - Структура затрат на разработку приложения

4.5 Оценка социально - экономических результатов функционирования ПП

К социально-экономическим показателям функционирования мобильной информационной системы является обработка таких факторов как:

- качество процесса управления;
- длительность и сроки проектирования программного продукта;
- расходы на реализацию приложения и эксплуатацию продукта;
- количеству разработчиков.

Рентабельность составит 20%.

Рассчитанную возможную цену ПП необходимо сравнить с ценами на аналогичные разработки.

Таблица 4.8 – Расчет срок окупаемости

Период	Первоначальные затраты	Денежный нарастающим итогом	поток
1	325532	110 000	
2	325532	110 000	
3	325532	110 000	
Срок окупаемости		3 месяцев	

Суммарные расходы на разработку аналитической информационной системы считаются как сумма фонда заработной плат, затрат на социальное страхование, накладных расходов и расходов на материалы и комплектующие.

Срок разработки данного проекта составил 30 дней или 192 часа.

Итоговая стоимость разработки приложения составила (определили по вышеуказанной методике), в том числе НДС – 46876,56 (12%). Предполагаемая цена продажи составила 437514,56 тг, прибыль от продажи составит 110 000 тг. в месяц.

Срок окупаемости проекта составит 3 месяцев.

5 Безопасность жизнедеятельность

Дипломный проект посвящен созданию информационной системы, которая будет автоматизировать работу администратора фитнес-клуба «МахGym». Это небольшая организация, которая предлагает услуги фитнес-силовых занятий. В помещении, где располагается администратор отсутствует шум, так как ведется исключительно продажа абонементов и обслуживание клиентов, поэтому шума нет. Площадь помещения составляет 40 м². Длина составляет - 5 метров, а ширина - 8 метров, а высота – 3 метра. Есть достаточно естественного освещения и искусственное освещение также присутствует. Относительно естественного освещения: есть два окна, расположенные по ширине помещения, габаритами, равными 3 метра в длину и 1 метр в высоту. Касательно искусственного освещения: в помещении есть 6 ламп. Света в данном помещении достаточно для нормальной организации труда. С вентиляцией в помещении дела обстоят иначе, окна не открываются, а кондиционер, который поставлен, не справляется. Поэтому будет делаться расчет вентиляции и подбираться соответствующий кондиционер, необходимый данному типу помещения. Каждый год вырастает потребность в правильном кондиционировании помещений, кондиционирование необходимо для поддержания правильных параметров воздуха.

5.1 Исходные данные

Таблица 5.1(а) –Исходные данные

Город		Алматы
Параметры помещения		Длина-5м, ширина – 8м, высота – 3м
Оборудование	количество	10
	Мощность Р _{об} , кВт/ч	1,5
	КПД	0,8
мощ. N ос.уст., Вт/м ²		40
Вид источников света		Лампы накаливания
Окна	Кол-во	2
	площадь 1 окна, м ²	3

Таблица 5.1(б) –Исходные данные

	Расположение	СЗ
	Вид	Жалюзи, <u>метал переплет</u> , загрязнение незначительное
Расчетное время суток, ч.		12-13
Температура, °С	летом	24
	зимой	20
Вид положения работы		сидя

Сотрудники: 3 мужчины, 1 женщина.

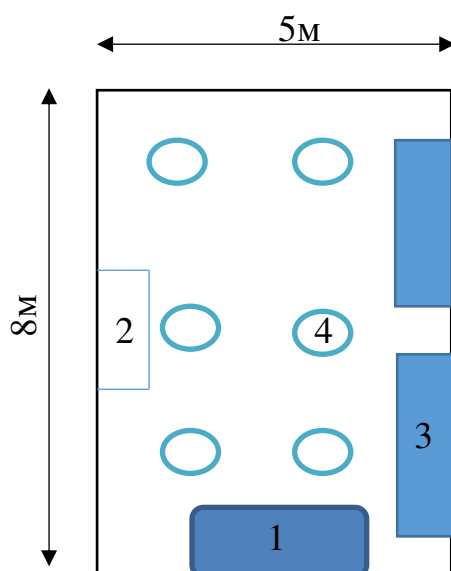


Рисунок 5.1- Помещение

5.2 Расчетная часть

Теплопоступления и теплопотери в результате разности температур определяются по формуле:

$$Q_{огр} = V_{пом} \times X_0 (t_{Нрасч} - t_{Врасч});$$

$$V_{пом} = 120 \text{ м}^3;$$

$$X_0 = 0.42 \text{ Вт/м}^3 \text{ } ^\circ\text{С}; X_0 \text{ – удельная тепловая характеристика};$$

Для теплого времени года:

$$t_{Нрасч} = 29,4 \text{ } ^\circ\text{С};$$

$$t_{Врасч} = 24 \text{ } ^\circ\text{С};$$

$$Q_{огр} = 120 \cdot 0,42 \cdot 5,4 = 272.16 \text{ Вт};$$

Для холодного времени года:

$$t_{Нрасч} = -9 \text{ } ^\circ\text{С};$$

$$t_{\text{Врасч}} = 20 \text{ }^{\circ}\text{C};$$

$$Q_{\text{отр}} = 120 \cdot 0,42 \cdot (-29) = -1461,6 \text{ Вт};$$

Теплопоступление от солнечного излучения через остекление определяется по формуле:

$$Q_p = (q_1 F_{01} + q_2 F_{02}) \times \beta_{\text{с.з.}};$$

$\beta_{\text{с.з.}}$ – коэффициент теплопропускания;

$$\beta_{\text{с.з.}} = 0,15;$$

$$Q_p = q_1 F_0 \times \beta_{\text{с.з.}} = (q_{\text{пр}} + q_{\text{р}}) \times K_1 \times K_2 \times \beta_{\text{с.з.}} \times n \times S_0;$$

$q_{\text{пр}}$; $q_{\text{р}}$ – тепловые потоки от прямой рассеянной радиации, Вт/м².

По таблице для СЗ после полудня в 12-13 ч. при расположении 3:

$$q_{\text{пр}} = 63 \text{ Вт/м}^2; q_{\text{р}} = 65 \text{ Вт/м}^2;$$

$F_0 = n S_0 = 2 \cdot 3 = 6 \text{ м}^2$ – площадь светового проема (n – число окон; S_0 – площадь 1 окна);

K_1 – коэффициент затемнения остекления переплетами (K_{C1} – для облученных проемов);

$$K_{C1} = 0,72;$$

K_2 – коэффициент загрязнения остекления;

$$K_2 = 0,9.$$

Тогда:

$$Q_p = (63 + 65) \cdot 0,72 \cdot 0,9 \cdot 0,72 \cdot 0,15 \cdot 6 = 53,74 \text{ Вт}$$

По таблице летом при 24 $^{\circ}\text{C}$ один мужчина выделяет явного тепла 67 Вт, а общего – 102 Вт. Женщина выделяет 85% от нормы тепловыделений взрослого мужчины. Тогда выделение явного тепла в помещении составит:

$$Q_{\text{л}}^{\text{я}} = 67 \cdot 3 + 67 \cdot 1 = 268 \text{ Вт.}$$

А выделение общего тепла:

$$Q_{\text{л}}^{\text{о}} = 102 \cdot 3 + 102 \cdot 1 \cdot 0,85 = 392,7 \text{ Вт.}$$

По таблице зимой при 20 $^{\circ}\text{C}$ один мужчина выделяет явного тепла 82 Вт, а общего – 103 Вт. Тогда выделение явного тепла в помещении составит:

$$Q_{\text{з}}^{\text{я}} = 82 \cdot 3 + 82 \cdot 1 = 328 \text{ Вт.}$$

А выделение общего тепла аспирационный тепловой кондиционер:

$$Q_{\text{з}}^{\text{о}} = 103 \cdot 3 + 103 \cdot 1 \cdot 0,85 = 396,55 \text{ Вт};$$

Теплопоступление от ламп определяется по формуле:

$$Q_{\text{осв}} = \eta \cdot N_{\text{осв}} \cdot F_{\text{пол}}, \text{ Вт};$$

где η – коэффициент перехода электрической энергии в тепловую (для люминесцентных ламп $\eta=0,5-0,6$);

$N_{\text{осв}}$ – установленная мощность ламп ($N=60 \text{ Вт/м}^2$);

$$F_{\text{пол}} – \text{площадь пола: } F_{\text{пол}} = 5 \cdot 8 = 40 \text{ м}^2.$$

Тогда:

$$Q_{\text{осв}} = 0,5 \cdot 40 \cdot 60 = 800 \text{ Вт.}$$

Тепло, выделяемое производственным оборудованием, определяется по формуле:

$$Q_{об} = N_{ум} \cdot K \quad (5.1)$$

$$Q_{об} = 1.5 \cdot 0.8 \cdot 10 = 12 \text{ кВт.}$$

Теплопритоки, возникающие за счет находящейся оргтехники, - это 30% мощности оборудования:

$$Q_{орг} = 1.5 \cdot 10 \cdot 0.3 = 4.5 \text{ кВт.}$$

На основании выполненных расчетов составим баланс теплоступлений в помещении:

$$Q_{изб} = Q_p + Q^a + Q_{осв} + Q_{об} + Q_{орг} + Q_{огр}$$

Лето:

$$Q_{изб} = 53.74 + 268 + 800 + 12000 + 4500 + 272.16 = 17893.9 \text{ Дж.}$$

Зима:

$$Q_{изб} = 53.74 + 328 + 800 + 12000 + 4500 + 1461.6 = 19143.34 \text{ Дж}$$

Так как тепловой баланс для лета больше зимнего теплового баланса, то рассчитаем теплонапряженность воздуха по формуле:

$$Q_H = \frac{Q_{избзима} \cdot 860}{V_{пом}} = \frac{19143.34 \cdot 860}{120} = 137 \text{ ккал/м}^3;$$

При $Q_H > 20 \text{ ккал/м}^3$, $\Delta t = 8 \text{ }^\circ\text{C}$,

Определение количества воздуха, необходимое для поступления в помещение:

$$L = \frac{Q_{изб} \cdot 860}{C \cdot \Delta t \cdot \gamma} = \frac{19143.34 \cdot 860}{0.24 \cdot 8 \cdot 1206} = 7109.9 \text{ м}^3/\text{час};$$

где $C = 0.24 \text{ ккал/(кг}^\circ\text{C)}$ – теплоемкость воздуха;

$\gamma = 1.206 \text{ кг/м}^3$ – удельная масса приточного воздуха.

Определение кратности воздухообмена:

$$N = \frac{L}{V_{пом}} = \frac{7109.9}{120} = 59.25 \text{ час}^{-1};$$

5.3 Выбор кондиционера для помещения

- 1 – Стол;
- 2 – дверь;
- 3 – окно;
- 4 – лампочка;
- 5 – внутренний блок;
- 6 – внешний блок.

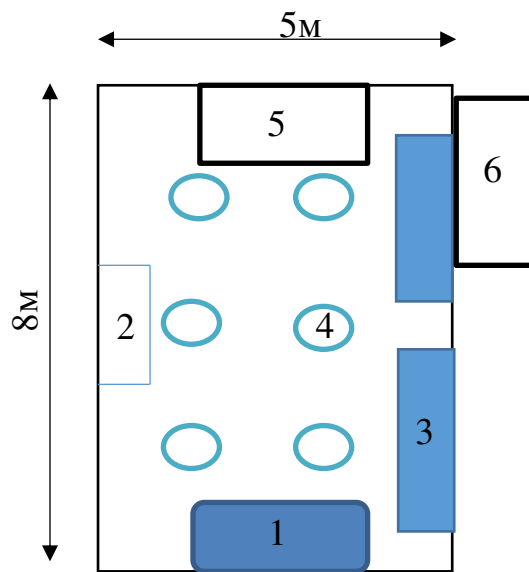


Рисунок 5.2 – Схема расположения кондиционера в производственном помещении

Заключение

В дипломном проекте разработана информационная система для автоматизации работы администратора фитнес-клуба. В ходе разработки были решены следующие задачи:

- выполнен анализ предметной области, в результате чего были выявлены процессы, подлежащие автоматизации;
- выполнены обзор и анализ существующих разработок-аналогов, приведены достоинства и недостатки рассмотренных аналогов. Было принято решение о разработке собственного программного средства;
- разработан алгоритм решения задачи;
- в соответствии с заданной предметной областью была спроектирована база данных, реализованная при помощи MS SQL Server, а так же создано программное обеспечение, реализованное с помощью языка Python и библиотеки PQT5, Qt Designer;
- проведена отладка и тестирование программного средства.

Информационная система отвечает требованиям заказчика, которые он предъявлял к функционалу и пользовательскому интерфейсу. Программный продукт прошел апробацию и внедрен в производство (фитнес-клуб «MaxGym»).

Список литературы

- 1 Проектирование информационных систем. Практикум: Учебное пособие / В.И.Грекул, Н.Л.Коровкина, Ю.В. Куприянов. – М.: Национальный Открытый Университет «ИНТУИТ», 2012. – 187 с.
- 2 Тенденции развития автоматизированных систем. Электронный ресурс: <https://studfiles.net/preview/962732>.
- 3 Избачков Ю.С. Информационные системы: Учебник для вузов / Ю.С. Избачков, В.Н. Петров. – 2-е изд. – СПб.: Питер, 2005. – 656
- 4 Волкова В.Н. Информационные системы: Учеб. пособие / Под ред. В.Н. Волковой, Б.И. Кузина. – СПб: ГТУ, 2001. – 216 с.
- 5 Гвоздева Т.В. Проектирование информационных систем: учеб. пособие / Т.В. Гвоздева, Б.А. Баллод. – Ростов-на-Дону: Феникс, 2009. – 508 с.
- 6 Буч Г., Якобсон А., Рамбо Дж UML 2.0. – СПб.: Питер, 2006. – 735 с.
- 7 Гамма Э. Приемы объектно-ориентированного проектирования // Э.Гамма, Р.Хелм, Р.Джонсон. – СПб: Питер, 2001. – 368 с.
- 8 Душин В.К. Теоретические основы информационных процессов и систем: Учебник / В.К. Душин. – М.: Издательско-торговая корпорация «Дашков и Ко», 2006. – 348 с.
- 9 1С:Предприятие. Конфигурирование и администрирование. – М.: ООО «1С-Публишинг», 2001.
- 10 Бондарь А. Microsoft SQL Server 2014. – СПб: БХВ-Петербург, 2015. – 592 с.
- 11 [Бен-Ган Ицик](#). Microsoft SQL Server 2012: T-SQL Fundamentals. – М.: Эксмо, 2015. – 400 с.
- 12 МакГрат М. Программирование на Python для начинающих. – М.: Эксмо, 2015. – 192 с.
- 13 Мэтиз Э. Изучаем Python. Программирование игр, визуализация данных, веб-приложения. – СПб.: Питер, 2018. – 496 с.
- 14 Бекишева А.И. Методические указания к выполнению экономической части дипломной работы для бакалавров специальности 5В0703 - Информационные системы – Алматы: АУЭС; 2013. – 24 с.
- 15 Мананбаева С.Е., Бегимбетова А.С. Охрана труда. Методические указания к выполнению расчетно-графических работ для студентов - бакалавров всех специальностей. - Алматы: АУЭС, 2013 - 17 с.

Приложение А (обязательное)

Техническое задание

1 Общие сведения

Полное наименование системы:

Разработка информационной системы для автоматизации работы администратора фитнес-клуба.

Наименование разработчика и заказчика работ

Разработчик:

- Сейлхан Мирас Батыржанулы

Адрес: республика казахстан, город Алматы

телефон: 8-747-535-15-39

Заказчик:

- Фитнес-клуб MaxGym

Сроки начала и окончания работ:

- Дата начала работ: 15.01.19;

- Дата окончания работ: 15.05.19.

1.2 Порядок предъявления заказчику работ

Работа будет принята поэтапно, со своевременными корректировками и пожеланиями для ее эффективной работы на предприятии.

Назначение и цели создания системы.

Назначение системы. ИС для автоматизации работы администратора фитнес-клуба:

- ведение и учет денежных средств;

- клиентский учет в электронном виде.

Характеристика объекта автоматизации.

Объектом автоматизации является набор всех существующих процессов, которые осуществляет администратор по взаиморасчетам с клиентами, а также ряда дополнительных работ, выполняющих функции информационной поддержки, контроля, а также нормативного регулирования объекта автоматизации. Функциональная система должна включать в себя все задачи, которые необходимы для автоматизации, обеспечение совместной работы всех подсистем для достижения целостной работы.

Требования к режимам функционирования должны поддерживаться следующие режимы для бесперебойной работы:

- сетевой режим функционирования;

- автономный режим функционирования.

Продолжение приложение А

1.3 Требования к жизненному циклу системы

Система должна иметь длительный жизненный цикл, должна быть разработана в соответствии с современными и инновационными решениями в области программирования и разработки новых ИС. Данная ИС должна быть открытой и подлежать внедрению новых функциональных возможностей. Программное средство и информационная система должна работать бесперебойно и без отказов, безотказно выполнять все функции и запросы к системе. Необходимым условием можно считать ведение журнала событий, также должна быть функция разграничения прав доступа. Весьма нужным условием считается восстановление данных и ее резервное копирование. Все уровни ИС должны быть подключены к источнику бесперебойного питания в целях сохранной работы данных в информационных системах.

1.4 Требования к внешнему виду системы

Обязательным условием является многооконный режим, простой и понятный интерфейс, цветовая гамма, гармонично и приятно сочетающаяся, наличие контекстно-зависимой помощи. Для сохранности информации при сбоях и авариях необходимыми являются носители информации, создание резервной копии хранимых данных, создание резервной копии программного обеспечения. Должна быть возможность восстановить утерянные данные с помощью импорта данных.

Таблица А.1 – Состав и содержание работ по созданию системы

Стадия работ	Выполняемые работы	Сроки	Итоги выполнения работы
Формирование требований	Исследование предметной области	выполнено	Отчет о результатах исследования
	Разработка специального технического задания	выполнено	Утверждение данного технического задания с заказчиком
Проектирование	Разработка логической и физической модели подсистемы Разработка эскизов и макетов предполагаемой подсистемы	1 месяц с начала выполнения работ	Техническое обеспечение системы. Выбор и утверждение используемых программных средств

Продолжение приложение А

Продолжение таблицы А1

Поставка программно-технических средств	Поставка программных и технических средств для грамотной работы	2 месяца с начала выполнения работ	Акты
Разработка программных средств	Разработка, отладка и тестирование программных средств программы и выявление возможных ошибок при эксплуатации	3 месяца с начала выполнения работ	Создана вся документация на подсистему, перенесена на машиночитаемые носители
Приемка работ	Показ проделанной работы и тестирование на машине исполнителя	4 месяца с начала выполнения работ	Акт о приеме работ, протоколы теста программ

1.5 Порядок контроля и приемки системы

Система при ее полной готовности должна тестироваться на стенде или оборудовании Исполнителя, при этом работу должны принимать представители заказчика и представители исполнителя. При проведении испытаний и тестов Исполнитель предъявляет все свои разработанные материалы и всю программную и эксплуатационную документацию. При успешном проведении тестовых работ приемная комиссия подписывает протокол испытания с перечисленными ошибками или коррективами. После корректировок проводятся повторные испытания данной разработанной системы.

1.6 Требования к документированию

Вся информация и программные коды должны быть предоставлены в бумажном виде и на носителе, программная реализация должна быть в электронном виде. Заказчику должна быть передана документация проектная, конструкторская, сопроводительная, а также предложения по дальнейшей поддержке разрабатываемой системы.

Приложение Б (обязательное)

Текст программы

```
#!/usr/bin/python

import sys

from PyQt5.QtCore import *
from PyQt5.QtGui import *
from PyQt5.QtWidgets import *
import login

#
*****
****
# ***** Tab Login_pswd
*****
#
*****
****

class QQ_App(QMainWindow):

    def __init__(self):
        super().__init__()
        self.title = 'PyQt5 tabs - pythonspot.com'
        self.left = 0
        self.top = 0
        self.width = 300
        self.height = 200
        self.setWindowTitle(self.title)
        self.setGeometry(self.left, self.top, self.width, self.height)

        self.table_widget = MyTableWidget(self)
        self.setCentralWidget(self.table_widget)

        # self.show()
```

Продолжение приложение Б

```
class MyTableWidget(QWidget):

    def __init__(self, parent):
        super(QWidget, self).__init__(parent)
        self.layout = QVBoxLayout(self)

        # Initialize tab screen
        self.tabs = QTabWidget()
        self.tab1 = QWidget()
        self.tab2 = QWidget()
        self.tabs.resize(300,200)

        # Add tabs
        self.tabs.addTab(self.tab1,"Coach")
        self.tabs.addTab(self.tab2,"Admin")

        # Create first tab
        self.textName = QLineEdit(self)
        self.textPass = QLineEdit(self)
        self.buttonLogin = QPushButton('Login', self)
        self.buttonLogin.clicked.connect(self.handleLogin)
        self.tab1.layout = QVBoxLayout(self)
        self.tab1.layout.addWidget(self.textName)
        self.tab1.layout.addWidget(self.textPass)
        self.tab1.layout.addWidget(self.buttonLogin)
        self.tab1.setLayout(self.tab1.layout)

        # Create second tab
        self.textName = QLineEdit(self)
        self.textPass = QLineEdit(self)
        self.buttonLogin = QPushButton('Login', self)
        self.buttonLogin.clicked.connect(self.handleLogin)
        self.tab2.layout = QVBoxLayout(self)
        self.tab2.layout.addWidget(self.textName)
        self.tab2.layout.addWidget(self.textPass)
        self.tab2.layout.addWidget(self.buttonLogin)
        self.tab2.setLayout(self.tab2.layout)

        # Add tabs to widget
        self.layout.addWidget(self.tabs)
        self.setLayout(self.layout)
```

Продолжение приложение Б

```
def handleLogin(self):
    if (self.textName.text() == " and
        self.textPass.text() == "):

        print("--==--")
        # print(dir(self.event))

    else:
        QMessageBox.warning(
            self, 'Error', 'Bad user or password')

    @pyqtSlot()
    def on_click(self):
        print("\n")
        for currentQTableWidgetItem in self.tableWidget.selectedItems():
            print(currentQTableWidgetItem.row(), currentQTableWidgetItem.column(),
                currentQTableWidgetItem.text())

#
*****
****
# ***** Login_pswd
*****
#
*****
****

class Login_pswd(QDialog):
    def __init__(self, parent=None):
        super(Login_pswd, self).__init__(parent)
        self.textName = QLineEdit(self)
        self.textPass = QLineEdit(self)
        self.buttonLogin = QPushButton('Login', self)
        self.buttonLogin.clicked.connect(self.handleLogin)

        layout = QVBoxLayout(self)
        layout.addWidget(self.textName)
        layout.addWidget(self.textPass)
        layout.addWidget(self.buttonLogin)

    def handleLogin(self):
        if (self.textName.text() == 'foo' and
```


Продолжение приложение Б

```
self.textPass.text() == 'bar'):
    self.accept()
else:
    QMessageBox.warning(
        self, 'Error', 'Bad user or password')
```

```
class Window(QMainWindow):
    def __init__(self, parent=None):
        super(Window, self).__init__(parent)
        # self.ui = Ui_MainWindow()
        # self.ui.setupUi(self)
```

```
#
*****
****
# ***** Calendar
*****
#
*****
****
```

```
class Calendar(QWidget):
    def __init__(self, parent=None):
        super(Calendar, self).__init__(parent)
        # self.initUI2()

    def initUI2(self):
        my_calendar = QCalendarWidget(self)
        my_calendar.setGridVisible(True)
        my_calendar.move(10, 20)
        my_calendar.clicked[QDate].connect(self.show_date)
        self.my_label = QLabel(self)
        date = my_calendar.selectedDate()
        self.my_label.setText(date.toString())
        self.my_label.move(10, 220)
        self.setGeometry(100,100,320,270)
        self.setWindowTitle('Calendar')
        self.show()
```

Продолжение приложение Б

```
def show_date(self, date):  
    self.my_label.setText(date.toString())
```

```
#  
*****  
****  
# ***** MainWindow  
*****  
#  
*****  
****
```

```
class MainWindow(QMainWindow):  
    def __init__(self):  
        super(MainWindow, self).__init__()  
  
        self.selectedDate = QDate.currentDate()  
        self.fontSize = 10  
  
        centralWidget = QWidget()  
  
        dateLabel = QLabel("Date:")  
        monthCombo = QComboBox()  
  
        for month in range(1, 13):  
            monthCombo.addItem(QDate.longMonthName(month))  
  
        yearEdit = QDateTimeEdit()  
        yearEdit.setDisplayFormat('yyyy')  
        yearEdit.setDateRange(QDate(1753, 1, 1), QDate(8000, 1, 1))  
  
        monthCombo.setCurrentIndex(self.selectedDate.month() - 1)  
        yearEdit.setDate(self.selectedDate)  
  
        self.fontSizeLabel = QLabel("Font size:")  
        self.fontSizeSpinBox = QSpinBox()  
        self.fontSizeSpinBox.setRange(1, 64)  
        self.fontSizeSpinBox.setValue(10)  
  
        self.editor = QTextBrowser()
```


Продолжение приложение Б

```
tableFormat.setColumnWidthConstraints(constraints)

table = cursor.insertTable(1, 7, tableFormat)

frame = cursor.currentFrame()
frameFormat = frame.frameFormat()
frameFormat.setBorder(1)
frame.setFrameFormat(frameFormat)

format = cursor.charFormat()
format.setFontPointSize(self.fontSize)

boldFormat = QTextCharFormat(format)
boldFormat.setFontWeight(QFont.Bold)

highlightedFormat = QTextCharFormat(boldFormat)
highlightedFormat.setBackground(Qt.yellow)

for weekDay in range(1, 8):
    cell = table.cellAt(0, weekDay-1)
    cellCursor = cell.firstCursorPosition()
    cellCursor.insertText(QDate.longDayName(weekDay), boldFormat)

table.insertRows(table.rows(), 1)

while date.month() == self.selectedDate.month():
    weekDay = date.dayOfWeek()
    cell = table.cellAt(table.rows()-1, weekDay-1)
    cellCursor = cell.firstCursorPosition()

    if date == QDate.currentDate():
        cellCursor.insertText(str(date.day()), highlightedFormat)
    else:
        cellCursor.insertText(str(date.day()), format)

    date = date.addDays(1)

    if weekDay == 7 and date.month() == self.selectedDate.month():
        table.insertRows(table.rows(), 1)

cursor.endEditBlock()
```

Продолжение приложение Б

```
self.setWindowTitle("Calendar for %s %d" %
(QDate.longMonthName(self.selectedDate.month()), self.selectedDate.year()))

def setfontSize(self, size):
    self.fontSize = size
    self.insertCalendar()

def setMonth(self, month):
    self.selectedDate = QDate(self.selectedDate.year(), month + 1,
        self.selectedDate.day())
    self.insertCalendar()

def setYear(self, date):
    self.selectedDate = QDate(date.year(), self.selectedDate.month(),
        self.selectedDate.day())
    self.insertCalendar()

#
*****
****
# *****
App_Dialog( *****
#
*****
****

class App_Dialog(QWidget):

    def __init__(self):
        super().__init__()
        self.title = 'PyQt5 input'
        self.left = 10
        self.top = 10
        self.width = 640
        self.height = 480
        # self.initUI()

    def initUI3(self):
        self.setWindowTitle(self.title)
        self.setGeometry(self.left, self.top, self.width, self.height)

        self.getInteger()
```

Продолжение приложение Б

```
self.getText()
self.getDouble()
self.getChoice()

self.show()

def getInteger(self):
    i, okPressed = QDialog.getInt(self, "Get integer","Percentage:", 28, 0,
100, 1)
    if okPressed:
        print(i)

def getDouble(self):
    d, okPressed = QDialog.getDouble(self, "Get double","Value:", 10.50, 0,
100, 10)
    if okPressed:
        print(d)

def getChoice(self):
    items = ("Red","Blue","Green")
    item, okPressed = QDialog.getItem(self, "Get item","Color:", items, 0,
False)
    if okPressed and item:
        print(item)

def getText(self):
    text, okPressed = QDialog.getText(self, "Get text","Your name:",
QLineEdit.Normal, "")
    if okPressed and text != "":
        print(text)

#
*****
****
# ***** App
*****
#
*****
****

class App(QWidget):
```

Продолжение приложение Б

```
def __init__(self):
    super().__init__()
    self.title = 'PyQt5 button - pythonspot.com'
    self.left = 10
    self.top = 10
    self.width = 320
    self.height = 200
    self.result = 0
    self.App_Dialog = App_Dialog()
    self.Calendar = Calendar()
    self.Login_pswd = Login_pswd()
    self.ui = login.Ui_Dialog2()
    self.MainWindow = MainWindow()
    self.QQ_App = QQ_App()
    self.initUI()

def initUI(self):
    self.setWindowTitle(self.title)
    self.setGeometry(self.left, self.top, self.width, self.height)

    button = QPushButton('Dialog', self)
    button.setToolTip('This is an example button')
    button.move(10,170)
    button.clicked.connect(self.blue_click)

    button = QPushButton('Log in', self)
    button.setToolTip('This is an example button')
    button.move(100,170)
    button.clicked.connect(self.login_func)

    button = QPushButton('Scheduler', self)
    button.setToolTip('This is an example button')
    button.move(100,70)
    button.clicked.connect(self.scheduler)

    button = QPushButton('Quit', self)
    button.setToolTip('This is an example button')
    button.move(230,170)
    button.clicked.connect(self.quit_App)

    button = QPushButton('Close', self)
    button.setToolTip('This is an example button')
```

Продолжение приложение Б

```
button.move(230,70)
    button.clicked.connect(self.close_app)

    self.show()

@pyqtSlot()
def blue_click(self):
    self.App_Dialog.initUI3()

@pyqtSlot()
def scheduler(self):
    # self.Calendar.initUI2()

    self.MainWindow.resize(640, 256)
    self.MainWindow.show()

@pyqtSlot()
def login_func(self):
    self.Dialog = QDialog()
    self.ui.setupUi(self.Dialog)
    self.Dialog.show()
    res = self.ui.loginCheck()
    if type(res) == tuple():
        if res[0] == 1:
            self.Dialog.close()
    print(self.ui.loginCheck(), '-----')
    print(self.ui.exec_())

@pyqtSlot()
def quit_App(self):
    QApplication.quit()

@pyqtSlot()
def close_app(self):
    self.Dialog.close()

# login = Login_pswd()
# l = login.exec_()
# print(l)

# if l == QDialog.Accepted:
```


Продолжение приложение Б

```
# window = Window()
# window.show()
```

```
#
```

```
*****
```

```
# ***** Main
```

```
*****
```

```
#
```

```
*****
```

```
if __name__ == '__main__':
```

```
    app = QApplication(sys.argv)
```

```
    ex = App()
```

```
    sys.exit(app.exec_())
```

Приложение В (справочное)

Акт о внедрении

АКТ

О принятии к внедрению результатов дипломной работы студента
специальности 050060200 – Информатика НАО АУЭС

Сейлхан Мирас Батыржанулы

(фамилия, имя, отчество студента)

На тему: Разработка информационной системы для автоматизации работы
администратора фитнес-клуба

(тема дипломной работы)

Настоящая работа посвящена администратору тренажёрного зала консультирует клиентов по возникающим вопросам, помогает подобрать наиболее подходящий вариант для клиента, регистрирует клиентов, абонементы, аренду и т. д. Администратор сидит за стойкой сразу же при входе в тренажерный зал. Качество разработанной БД всецело зависит от качества выполнения отдельных этапов ее проектирования. Огромное значение имеет качественная разработка логической модели данных, так как она, с одной стороны, обеспечивает адекватность базы данных предметной области, а с другой стороны, определяет структуру физической БД и, следовательно, ее эксплуатационные характеристики. Результаты дипломного проекта Сейлхан Мираса было установлена программа для фитнес-клуба MaxGym.

Эффект от внедрения приходится на сокращение работы администратора фитнес-клуба и анализ необходимых данных по клиентам и сотрудникам, а так же оптимизация процессов контроля обработки этих данных.

Руководитель предприятия

(учреждения, отдела)

Токаренко М. В. (расшифровка подписи)

М.П. предприятия

