**МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ**

**ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ**

**УЧРЕЖДЕНИЕ ВЫСШЕГО ПРОФЕССИОНАЛЬНОГО ОБРАЗОВАНИЯ**

**«Рязанский государственный радиотехнический университет»**

**(ФГБОУ ВПО «РГРТУ», РГРТУ)**

Факультет ВТ К защите

Кафедра ВПМ Зав. кафедрой \_\_\_\_\_\_\_\_\_ Пылькин А.Н.

Специальность 230105

« \_\_\_ » 2015 г.

#### 

#### ПОЯСНИТЕЛЬНАЯ ЗАПИСКА

К ДИПЛОМНОМУ ПРОЕКТУ НА ТЕМУ

**«Разработка программных средств обеспечивающих функционирование виртуального предприятия»**

Дипломник \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Кирюшин А.В.

Руководитель проекта \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Антипов В.А.

Консультант по

экономической части \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Кутузова И.В.

Консультант кафедры \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Антипов В.А.

Консультант

по безопасности

и экологичности проекта \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Голованчикова Л.М.

Рязань 2015

**Задание**

**Задание**

**Реферат**

Пояснительная записка 139 с., 10 табл., 34 рис., 23 источников.

ВИРТУАЛЬНОЕ ПРЕДПРИЯТИЕ, ПРОГРАММНОЕ СРЕДСТВО, МЕНЕДЖЕР ЗАДАЧ, ИНФОРМАЦИОННАЯ СИСТЕМА, ЧАТ

Цель проекта — создание системы обеспечивающей функционирование виртуального предприятия.

Средства разработки — среда программирования *Microsoft Visual* *Studio*.*NET,Microsoft SQL Server 2008*, язык программирования *С#,SQL*.

Назначение разработанных средств — обеспечение взаимодействия сотрудников внутри виртуального предприятия.

Область применения — виртуальные предприятия любого направления работы .

**Abstract**

The explanatory note consists of 139 pages, 10 tables, 34 figures, 23 sources.

VIRTUAL ENTERPRISE, SOFTWARE TOOL, TASK MANAGER, INFORMATION SYSTEM, CHAT

The purpose of the project – creating a system of operating the virtual enterprise.

Development tools are the programming environment *Microsoft Visual Studio 2010 EXPRESS*, a database management system *Microsoft SQL Server 2012 EXPRESS*.

The scope - virtual enterprises of all areas of work.

**ОГЛАВЛЕНИЕ**

ВВЕДЕНИЕ..............................................................................................................9

1. ТЕХНИКО-ЭКОНОМИЧЕСКОЕ ОБОСНОВАНИЕ ТЕМЫ........................13

1.1. Актуальность и практическая ценность разработки...............................13

1.2. Сравнение существующих аналогов........................................................15

1.3. Выбор средств разработки.........................................................................17

2. ТЕОРЕТИЧЕСКАЯ ЧАСТЬ.............................................................................19

2.1. Понятие виртуальной организации и цели ее создания...........................19

2.1.1. Виртуальное предприятие *«VIRTEC Project»*………**…..…**.....…....22

*2.1.2.* Виртуальное предприятие *«AGI-Wilgelm»*…………………....…….24

*2.1.3.* Виртуальное предприятие *«*Virtual University Enterprises*»*…...……25

2.2. Виды виртуальных организаций и их участники.....................................26

2.3. Особенности управления организацией, достоинства и недостатки виртуальных организаций....................................................................................33

2.4. Принцип построения виртуальных организаций....................................37

3. ПРОЕКТИРОВАНИЕ СИСТЕМЫ...................................................................38

3.1. Проектирование базы данных...................................................................38

3.1.1. Анализ предметной области................................................................38

3.1.2. Выделение сущностей..........................................................................39

3.1.3. Выявление связей между сущностями...............................................41

3.1.4. Формирование набора предварительных отношений.......................44

3.1.5. Разработка физической модели базы данных...................................48

3.1.6. Разработка средств поддержания целостности данных....................51

3.2. Основные технологии, применяемые при разработке системы.............55

3.3. Разработка пользовательского интерфейса............................................57

4.ОПИСАНИЕ, ПРОГРАММНАЯ РЕАЛИЗАЦИЯ АЛГОРИТМОВ.............

4.1. Разработка алгоритмов.........................................................................63

4.2. Проектирование классов программы.................................................60

4. ТЕХНИЧЕСКАЯ ДОКУМЕНТАЦИЯ............................................................66

4.1. Описание применения...............................................................................66

4.2. Условия применения...................................................................................66

4.3. Входные и выходные данные.....................................................................67

4.4. Руководство программиста.........................................................................67

4.4.1. Общие сведения о программе.............................................................67

4.4.2. Настройка программы..........................................................................67

4.5. Обращение к программе.............................................................................68

4.6. Руководство пользователя..........................................................................68

4.6.1. Интерфейс программы.........................................................................68

5. ТЕСТИРОВАНИЕ СИСТЕМЫ........................................................................75

5.1. Общие положения.......................................................................................75

5.2. Разработка методики проведения испытаний...........................................75

5.3. Результаты проведения испытаний...........................................................76

5.3.1. Условия тестирования..........................................................................77

5.3.2. Тестирование входных данных...........................................................77

5.3.3. Тестирование выходных данных........................................................77

6. БЕЗОПАСНОСТЬ И ЭКОЛОГИЧНОСТЬ ПРОЕКТА...................................79

6.1. Характеристика помещения и описание рабочего места........................79

6.1.1. Шум....................................................................................................81

6.1.2. Микроклимат.................................................................................... 82

6.1.3. Воздействие электрического тока................................................... 83

6.2. Оптимизация зрительных условий труда на рабочем месте................ 84

6.2.1. Основные требования к освещению с учётом труда.....................84

6.2.2. Обоснование организации освещения и нормативного уровня освещённости рабочего места............................................................................. 86

6.2.3. Средства и способы обеспечения требуемой освещённости равномерности светораспределения....................................................................88

6.2.4. Расчёт освещения рабочего места..................................................89

6.3. Обеспечение пожарной безопасности......................................................92

6.4. Экологичность проекта.............................................................................97

7. ЭКОНОМИЧЕСКАЯ ЧАСТЬ..........................................................................98

7.1. План-график разработки.....................................................................98

7.2. Составление сметы затрат на разработку.........................................99

7.2.1. Материальные затраты..................................................................99

7.2.2. Затраты на оплату труда.............................................................103

7.2.3. Амортизация основных фондов.................................................104

7.2.4. Прочие затраты...............................................................................105

7.3. Расчет проектной цены подписки на ПО.........................................103

7.4. Расчет и выводы по эффективности предложений.........................104

ЗАКЛЮЧЕНИЕ...................................................................................................105

Библиографический список...............................................................................107

Приложение. Листинг наиболее значимых частей программы......................109

**ВВЕДЕНИЕ**

Характерной чертой начала нового тысячелетия является интенсивная интеграции стран в мировое информационное сообщество. И большую роль в этом продолжает играть всемирная сеть Интернет. Поэтому интерес представляют те положительные изменения, которые происходят в ее эволюции.

Примечательно, что в последние годы наблюдается наступление нового этапа в жизни сети Интернет, когда она превращается в платформу наподобие обычного персонального компьютера. Миру, в котором пользователи Интернета могли только читать информацию, вероятнее всего наступает конец, поскольку в скором времени сеть объединит множество web-узлов, развернутых на самых разных системах и предоставит пользователям возможность читать, публиковать и аннотировать информацию, используя для этого голосовые средства, программы распознавания рукописного текста и другие подобного рода системы. Программное обеспечение этой платформы по существу станет как бы «программным двигателем», который будет способствовать изменениям Интернета и повышению его мощи.

Все эти изменения способствовали возникновению новых бизнес-структур, которые с помощью Сети могут устанавливать партнерские отношения, находясь в любой точке земного шара. Это, в свою очередь, позволило территориально рассредоточенным предпринимателям или компаниям создавать единый или базовый уровень компетенции и при помощи инфокоммуникационных технологий (ИКТ) разрабатывать эффективные технологические процессы производства товаров и услуг. В зависимости от решаемых задач такого рода объединение партнеров по бизнесу получило название «виртуального», а сформированные структуры стали именоваться «виртуальной командой», «виртуальным предприятием», «виртуальной корпорацией» и т.п. Необходимо особо подчеркнуть, что в их базе лежит временная сеть, объединяющая независимых предпринимателей или компании, которые, используя ИКТ, делятся опытом, затратами и успехами, достигнутыми на рынке. Каждый из партнеров содействует решению общей задачи только в пределах своего уровня компетенции.

Сказанное выше заставляет провести анализ того состояния, в котором находятся уже созданные или планируемые к созданию бизнес-структуры, использующие для повышения эффективности своей деятельности такие сети, как Интернет, Интранет и др. ,постараться определить их эффективность и перспективу выживаемости, дать характеристику менеджмента в условиях использования высокоэффективных ИКТ.

**Актуальность дипломного исследования** обуславливается тем, что современный рынок требует постоянного создания и внедрения в деятельность предприятий новых методов управления, способных обеспечивать как достоверный контроль текущей работы предприятия, так и видение перспектив принимаемых стратегических решений. Однако большинство предприятий имеет проблемы с организацией управления своей деятельности. Многие стратегические и тактические просчеты с недостаточно квалифицированным менеджментом. Это означает, что традиционные методы менеджмента уже в значительной мере себя исчерпали, и необходимо искать новые подходы к управлению. Речь должна идти о принципиально новом качестве управления: в решении теоретических и прикладных задач управления наиболее плодотворным является программный подход к динамическому процессу решения проблем взаимодействия системы с внешней средой.

В современном мире выживаемость любой компании, ее устойчивое положение на рынке услуг определяется уровнем конкурентоспособности. В свою очередь конкурентоспособность связана с двумя факторами- уровнем цены и уровнем качества оказываемых услуг. Причем второй фактор постепенно выходит на первое место.

На основе вышеизложенного **целью дипломного проекта**следует определить разработку программного обеспечения повышающего качество оказываемых услуг и увеличения эффективности коммуникации сотрудников предприятия.

Таким образом, **практическая ценность разрабатываемого программного средства** заключается в решении следующего перечня задач:

1. Обеспечение коммуникации сотрудников;
2. Возможность анализа вклада каждого участника в проект;
3. Менеджер задач позволяющий отслеживать все задачи предприятия и назначать ответственных за их выполнение;
4. Быстрое информирование всех необходимых сотрудников.

Краткое содержание работы.

1. В разделе «Технико-экономическое обоснование» анализируются существующие программы, предназначенные для решения аналогичных задач, делается вывод о необходимости разработки ПО, а также определяются основные средства разработки и технологии, необходимые для решения поставленной задачи.

2. В разделе «Проектирование системы» выполняется анализ предметной области, рассматривается процесс проектирования базы данных и разработки основных алгоритмов программы, проектируется интерфейс пользователя.

3. В разделе «Техническая документация» приводится документация программы, состоящая из руководства пользователя и руководства программиста. Руководство пользователя описывает порядок работы оператора с программой, руководство программиста содержит описание действий для установки и настройки программы.

4. В разделе «Тестирование системы» описываются методы и результаты проверки разработанного программного обеспечения на соответствие исходным требованиям.

5. В разделе «Экономическая часть» выполняется планирование процесса разработки, осуществляется расчет сметы затрат на разработку программного обеспечения и делается вывод о практическом значении и целесообразности выполнения данной разработки.

6. В разделе «Безопасность и экологичность проекта» рассматриваются меры по обеспечению безопасной работы за компьютером при использовании программы с точки зрения безопасности жизнедеятельности.

7. В заключении анализируются полученные результаты и определяются направления дальнейшего развития разработанного ПО.

В результате разработки были созданы программные средства обеспечивающие коммуникацию между сотрудниками виртуального предприятия и позволяющие им давать задачи друг другу.

**1.** **ТЕХНИКО-ЭКОНОМИЧЕСКОЕ ОБОСНОВАНИЕ ТЕМЫ**

**1.1. Актуальность и практическая ценность разработки**

Целью данного дипломного проекта является разработка программного обеспечения для обеспечения функционирования виртуального предприятия.

Современные организации постоянно ищут пути совершенствования эффективности, и поэтому постоянно оценивают уровень своих возможностей и недостатков. Присутствие в глобальной сети и совершенствование технологий предоставляет больше и больше вариантов эксплуатации систем для управления бизнесом.

Сегодня не оставляет никакого сомнения, что Интернет значительно изменил те правила, по которым происходило взаимодействие между организациями. Устраняя географические границы, Интернет позволил организациям обмениваться информацией и управлять транзакциями без требования работать с клиентом лицом к лицу. Эта технология, вместе с технологией обмена сообщениями, дали рождение технологии порталов и виртуальных торговых площадок. Порталы позволяют организациям, которые заинтересованы в бизнесе, быстро находить партнеров и поставщиков. Участие в торговле на бизнес-портале позволяет вести бизнес со всеми организациями, участвующими в работе портала.

В мире бизнеса слияния и объединения происходят ежедневно. Чаще всего объединяющиеся организации имеют различные системы управления, настроенные на бизнес-практику каждой из компаний. Заставить одну из организаций принять внутреннюю бизнес-систему другой для того, чтобы работать в одном поле данных, зачастую очень затратно и губительно для бизнеса. Люди, принимающие решения о слиянии, часто подразумевают, что требование о сосуществовании этих систем является главным для осуществления такого объединения. Интеграция между приложениями часто требуется и в обыденной жизни предприятия. Специальные программы, обслуживающие техпроцессы и финансовые приложения, часто используют гетерогенные платформы и должны быть объединены в единое информационное пространство. В этих случаях требуется недорогое, легко внедряемое и гибкое решение. Именно здесь и находят свое применение технологии обмена сообщениями, так как они практически не требуют вносить какие-либо изменения в работающие приложения для обеспечения надежной коммуникации между ними. Таким образом, они являются эффективным решением для обеспечения коммуникации между любым видами приложений на различных платформах.

Чтобы обеспечить информационный обмен между организациями, необходимо подготовить для них единое информационное пространство, где они могут обмениваться информацией об оказываемых услугах, поставляемых материалах, а также запрашивать необходимую информацию у других участниках.

Кроме того, все организации объединяются в единую виртуальную организацию (BO), которая состоит из юридически независимых предприятий, учреждений и отдельных предпринимателей. Объединенные группы главным образом обеспечивают свои внутренние потребности и их внешнее поведение подобно одиночной корпорации. Подобная корпорация отказывается от центрального офиса, вместо этого она управляется благодаря информационным и коммуникационным технологиям. Виртуальная организация – это новое объединение, сформированное за счет взносов ресурсов от обычных независимых предприятий.

Кроме того необходимо разделять весь поток информации внутри виртуальной организации на соответствующие группы, объединенные по общим признакам, так, чтобы каждый член виртуальной организации получал только те информационные сообщения, в которых она заинтересована. Наиболее близко, данную проблему решает система, основанная на технологии Публикация/Подписка. Данная система позволяет одним агентам опубликовывать информацию определенного типа, а другим – принимать те сообщения, на которые они подписались.

Разрабатываемая технология позволит сократить издержки при передаче информации между взаимодействующими предприятиями и обеспечить оперативную доставку сообщений. Также данная система обеспечит высокую масштабируемость, что позволит обмениваться информацией агентам, расположенным на значительном расстоянии друг от друга.

Современные виртуальные предприятия представляют сложные мультиагентные системы нуждающиеся в грамотном управлении и четком разделении обязанностей между агентами этой системы. Этого можно достичь используя систему позволяющую добавлять, редактировать, устанавливать сроки выполнения задач, а также закреплять задачи за конкретным агентом. Вторым аспектом успешного управления виртуальным предприятием является мониторинг деятельности агентов. Это можно обеспечить созданием аналитической системы отслеживающей активность сотрудников.

Внедрение данного информационного пространства даст следующие преимущества:

* уменьшение издержек;
* значительное повышение производительности;
* удобство работы с данными;
* возможность отслеживать слабые места предприятия;
* единый интерфейс взаимодействия между организациями;
* возможность работы с территориально разрозненными организациями.

## 1.2. Сравнение существующих аналогов

В настоящее время на рынке услуг программного обеспечения существует множество решений для планирования задач компании. Также имеется много программ в сфере коммуникации. Разрабатываемый нами продукт совмещает в себе два этих направления. При разработке данного дипломного проекта был произведен анализ этих программных продуктов и выделены плюсы и минусы каждого из них.

Первое рассмотренное решение называется “Synkra”.

Плюсы данного решения будут следующие:

* быстрая регистрация компании с помощью почты;
* возможность установления сроков выполнения задачи и сотрудника который будет ее выполнять;
* полностью бесплатно;
* не требуется обучения для работы с программой;
* аналитический раздел позволяющий отслеживать вклад каждого участника в проект.

Недостатки данного решения при решении нашей задачи следующие:

* нет возможности подписки на оповещения от других участников проекта;
* нет возможности личной переписки участников организации;
* нет возможности посмотреть кто дал задачу конкретному сотруднику.

Второе решение называется “Wrike” и имеет следующие достоинства:

* оповещение всех сотрудников компании с помощью ленты новостей;
* быстрая регистрация компании с помощью почты;
* возможность установления сроков выполнения задачи и сотрудника который будет ее выполнять.

Недостатки:

* лицензия программы требует 42$ в месяц;
* нет возможности личной переписки участников организации;
* нет возможности посмотреть кто дал задачу конкретному сотруднику.

Поскольку ни одна из рассмотренных программ не отвечает предъявляемым требованиям, было принято решение самостоятельной разработки программного обеспечения.

Для реализации поставленной задачи необходимо выбрать средства разработки (среда программирования, СУБД, технологии работы с сетью).

**1.3. Выбор средств разработки**

В качестве средств разработки были выбраны те технологии, которые наиболее полно соответствуют требованиям разрабатываемой системы. Для реализации клиентской части программного продукта была выбрана среда разработки Microsoft Visual Studio 2013.

Microsoft Visual Studio - мощная среда разработки, обеспечивающая высокое качество кода на протяжении всего цикла разработки программного обеспечения, от проектирования до внедрения. На сегодняшний день является одним из лучших средств разработки приложений. С каждой новое версией эта среда приобретает все больше и больше полезных функций.

Благодаря этому достигаются многие желательные свойства:

* удобный интерфейс;
* приятный внешний вид;
* всестороннее взаимодействие с любыми СУБД;
* легкость использования компонентов;
* легкость в написании кода;
* возможность разработки нескольких частей одного приложения на разных языках;
* возможность бесшовной отладки такого приложения;
* возможность написать класс на одном языке, а его потомков - на других языках.

Microsoft SQL Server 2008 была выбрана в качестве СУБД для дипломного проекта. Она обеспечивает платформу данных с приемлемым для данной задачи уровнем безопасности, производительности, масштабируемости и продуктивности разработчиков. SQL Server является всеобъемлющим, интегрированным сквозным решением.

SQL Server предоставляет мощные инструменты, уменьшая сложность создания, развёртывания, управления и использования, данных предприятия и аналитических приложений на платформах от мобильных устройств до информационных систем предприятия.

Благодаря исчерпывающему набору функций, взаимодействию с существующими системами и автоматизации типовых задач, Microsoft SQL Server 2008 предоставляет полное решение в области хранения данных для организаций всех масштабов. Главное объединяющее достоинство данных программных продуктов заключается в том, что в реализации разработки нашего проекта практически не нужно материальных

затрат на их покупку, т.к. можно с легкостью обойтись бесплатными версиями Express.

**2. ТЕОРЕТИЧЕСКАЯ ЧАСТЬ**

**2.1. Понятие виртуальной организации и цели ее создания.**

Виртуальная организация - новейшая и потенциально наиболее важная форма бизнес-организаций из тех, что появились в течение последних десятилетий. Новая модель стала возможной в результате развития наисовременнейших технологий, самой значимой из которых является Интернет. Виртуальные предприятия представляют собой группы людей, совместно занимающихся общим делом, независимо от их физического местонахождения, пересекая границы предприятий и стран, в реальном времени (синхронно) или в отсроченном режиме (асинхронно). Они (и предприятия, и люди) могут быстро реагировать на изменения рынка при критически низких затратах с точки зрения традиционного бизнеса.

Жизненный цикл любого виртуального предприятия проходит пять фаз.

1. Фаза выявления перспективной предпринимательской возможности.

Чтобы избежать провала, следует удостовериться в фактическом существовании той или иной предпринимательской возможности и оценить ее перспективность. Таким образом, на данном этапе жизненного цикла виртуального предприятия происходит обоснование целесообразности его создания.

2. Фаза создания виртуального предприятия.

Для реализации выявленной перспективной предпринимательской возможности следует определить конкретных участников будущего проекта: группу индивидуумов (организаций). На этой стадии ведутся переговоры по поводу распределения ролей и ответственности между партнерами, разрабатывается план предпринимательской деятельности.

3. Фаза функционирования виртуального предприятия.

На протяжении данной фазы жизненного цикла виртуального предприятия осуществляется согласованная и спланированная на предыдущей стадии предпринимательская деятельность партнеров, в ходе которой они координируют свои действия и поддерживают постоянную связь.

4. Фаза ликвидации виртуального предприятия.

 После завершения фазы функционирования у виртуального предприятия могут остаться некоторые невыполненные обязательства, и в этом случае события развиваются по следующему сценарию. Обязательства остаются в силе, но виртуальное предприятие приостанавливает функционирование до появления требований по этим обязательствам, то есть находится в неактивном состоянии. При получении таких требований виртуальное предприятие временно возобновляет свою деятельность и, выполнив соответствующие обязательства, возвращается в неактивное состояние.

5. Фаза распада виртуального предприятия.

  Когда все обязательства виртуального предприятия выполнены, начинается завершающая стадия его жизненного цикла, на которой партнерские отношения между его участниками разрываются, а само оно распадается, то есть прекращает свое существование. [1]

Виртуальными организациями следует руководить на «пять с плюсом» и менеджмент в них играет едва ли не самую главную роль. Первые компании, всерьез занимавшиеся электронной коммерцией потерпели крах(Вульф 1999, Портенджер 2002) из-за плохого менеджмента и финансирования. В данной среде большое влияние приобретает принцип B2B(business-to-business,поддержка компаний друг другом).

Виртуальные организации имеют много тех же особенностей, что и традиционные: им необходимы финансы, их штат состоит из людей. Но виртуальные организации несут в себе огромный потенциал, и для того чтобы раскрыть его полностью, от виртуальных менеджеров требуется гораздо больше профессионализма, чем от менеджеров традиционных компаний.

Такие виртуальные предприятия могут и будут постоянно перестраивать свою конфигурацию и архитектуру процессов, чтобы сохранять максимальную эффективность в условиях динамичного рынка. Благодаря своей способности создавать и эксплуатировать более новаторские и целенаправленные службы при меньших капиталовложениях, в более сжатые сроки и со значительно меньшим финансовым риском, они составят серьезную конкуренцию крупным традиционным корпорациям.

Internet предоставляет любому индивидууму возможность обмениваться информацией с любым человеком в любом уголке мира, и это позволяет создавать общность людей по интересам, для которых расстояние не имеет значения (частью этого сценария является электронная коммерция). В то же самое время, современные технологии поддержки бизнес-процессов позволяют им пересекать границы компании и вновь «соединяться» через Internet. Тем самым появляется перспектива сотрудничества между экономическими субъектами для организации совместных предприятий и возможность динамической перестройки их конфигурации по мере необходимости. В конечном итоге, это приведет к мощному прорыву в области производительности, организации, международных обменов и экономического роста благодаря следующим факторам:

1.Инновационные продукты и услуги, обеспечивающие эффективное и малозатратное обслуживание клиентов по всему миру за счет объединения средств коммуникации, электронной коммерции и автоматизации бизнес-процессов;

2.Структурированные и описанные процессы, где клиент является непосредственным участником, а процессы на базе workflow прослеживают транзакции, пересекая границы подразделений, компаний и предприятий;

3.Эффективно выстроенные организации, предоставляющие наилучшее обслуживание за счет реализации комплексных бизнес-процессов на базе workflow, несмотря на внутреннюю структуру, адаптируемую к рыночным нуждам;

4.Динамичное взаимовыгодное сотрудничество между предприятиями и индивидуумами; лучшее обслуживание рынка будет основываться на оптимальной производительности каждого из объединившихся партнеров, которая достигается за счет управления бизнес-процессами на базе новейших информационных технологий.[2]

Виртуальная организация бизнес-процессов характеризуется наличием большого числа форм, включающих например сеть стабильных поставок, которая работает как единое предприятие ,или же компаний, действующих разобщенно, независимо от того, входит ли она в общий контракт поставок, или же ее функции заключаются в обмене необходимой информацией. Компания OMNI, например, осуществляет обмен информацией со 186 размещенными в разных точках мира фирмами и исключает таким образом обязательное использование транспортных средств. Рассмотрим эффективность виртуальной организации на примере некоторых функционирующих виртуальных предприятий.

**2.1.1. Виртуальное предприятие *«VIRTEC Project»***

«VIRTEC Project» разработан инженерным факультетом (Сан Карлос) Университета Сан Паоло в Бразилии. Он объединяет девять малых и средних предприятий, которые обеспечили проект финансированием, технологиями и сервисом. Эти предприятия функционировали в области электроники, производства металло-керамических изделий, полимерных материалов, механики, механотроники (японское название отрасли промышленности, образовавшейся в результате слияния общего и электротехнического машиностроения), жидкостных систем, прикладного программного обеспечения и сервиса.

Работа данной группы компаний была построена на кооперативной основе, где каждый партнер имел доверительные отношения друг с другом и доступ к создаваемой инфраструктуре нового бизнеса. Задача состояла не только в принятии участия в глобальном бизнесе, но и в построении производственного процесса. Это важно отметить, поскольку компании в пределах виртуальной организации должны знать не только о том, как они должны действовать, но и где они должны осуществлять свои действия.

Для решения проблемы была разработана матрица базовой компетенции, позволившая дифференцировать между партнерами все виды деятельности, включая получение прибыли каждым участником проекта.

Каждая компания специализировалась на производстве одного продукта, например, одна — на изготовлении полиуретанового молотка, который очень быстро вырабатывался при его использовании; другая — сосредотачивалась на производстве полимерной резины и т.д.

В результате своей деятельности виртуальная компания VIRTEC разработала несколько новых продуктов, таких как вегетируемая (разлагающаяся в природных условиях) полиуретановая резина. Эта резина используется для целей механического ассемблирования, покрытия продуктов, в сфере медицинской диагностики и т.д. Разработанный тип резины оказался дешевле и имел более высокий по времени жизненный цикл. В настоящее время созданная резина продается в странах Европы и Америки.

Используя матрицу компетенции, компания VIRTEC показала хороший пример того, что виртуальная форма организации позволяет сократить временные затраты на разработку и производство нового продукта, уменьшить его себестоимость, а также повысить качественные параметры, в сравнении с аналогичным товаром других производителей.

***2.1.2. Виртуальное предприятие «AGI-Wilgelm»***

Немецкая компания «AGI businessmedia productions GmbH» была основана тремя студентами в 1966 г. в г.Штутгарте, Германия. Бизнес заключался в разработке и реализации Интернет-приложений и их концепций. «AGI» считается крупнейшим мультимедийным агентством Германии. Ежегодная прибыль компании составляет 4 млн евро. Структура компании представлена пятью директорами, 60 служащими и 25 работниками по найму. В 1997 г. Торгово-промышленная палата г. Штутгарта (IHK)совместно с «AGI» приступила к разработке проекта, получившего название «IHK», одна из задач которого состояла в разработке WEB-сайта, содержащего базы данных по роду деятельности партнеров. Реализация проекта повысила имидж компании, и к ней за помощью обратился зоологический сад «Wilhelma» (г. Штутгарт) с просьбой разработать в Интернете при условии обеспечения независимости и приемлемости для клиентов с учетом их потребностей. В перечень задач были включены такие вопросы, как снижение загруженности администрации вопросами клиентов, интенсификация взаимосвязей между клиентами зоосада и обеспечении их необходимой информацией и др. [3].

Дизайн проекта был разработан на основе концепции жизненного цикла виртуальной организации, включающей фазу ориентации компании, разработку системы взаимосвязей с клиентами, решение проблемы координации и реализации продукции. Все вопросы предполагалось решать через web-сайт. Были решены вопросы операционных издержек, формирования базовой компетенции и социологические аспекты проблемы, что позволило построить результативную и эффективную виртуальную компанию. Преимущество ее выразилось в следующих параметрах активности:

* организовано производство в цифровом формате дополнительной рекламной продукции посредством внедрения HTML-редактора и других программных продуктов;
* клиентам стали доступны многочисленные данные о деятельности партнеров виртуальной компании «IHK-Wilhelma»;
* оптимизация процессов позволила интенсифицировать взаимосвязи с клиентами через электронную почту;
* исключены посредники между компанией и потребителями;
* разработана электронная библиотека, ставшая доступной потребителям продукции и услуг, которые предлагаются партнерами виртуальной компании.

Все эти и ряд других преимуществ позволили существенно повысить конкурентноспособность виртуальной компании и интенсифицировать свой бизнес-процесс.

***2.1.3. Виртуальное предприятие «*Virtual University Enterprises*»***

Виртуальная компания«Virtual University Entyerprises» (VUE, подразделение компании NCS Pearson,Inc.) была образована в 1997 году и является сравнительно молодой виртуальной компанией, предоставляющей услуги по авторизованному тестированию [4].

Использование базовых основ виртуальной организации, таких как электронная сетевая инфраструктура, концентрация компетенции, системы взаимодействия с клиентами и т.д., позволило создать и оперативно функционировать более чем 2500 центрам компании, расположенных в 110 странах мира. Эти центры проводят экзамены по программам сертификаций таких компаний, как Check Point Technologies, Cisco Systems,CompTIA, Ericsson, Generation, Informix, Linux Professional Institute (LPI),Lucent Technologies, Microsoft, Novell, PTC, RSA Security, IBM, Tivoli и др.

Новыми тестовыми центрами«Virtual University Enterprises», функционирующими в России, стали «АстроСофт» и «Академия АйТи».

«Академия АйТи», например, готовит специалистов к сдаче большинства экзаменов по программам сертификации таких фирм, как Microsoft, Novell, Oracle, Caldera, Lotus, Avaya и т.д.

**2.2. Виды виртуальных организаций и их участники**

Существует несколько позиций, на основании которых различают виртуальные организации.

Представители первого подхода выделяют три основных вида виртуальных организаций:

1. с централизованным типом управления, при котором один из агентов управляет процессом (уясняет задачу, выдает задания другим агентам, обобщает результаты и принимает решения);

2. с распределенным типом управления, где знания и ресурсы распределяются между агентами, но сохраняется общий орган командного управления, принимающий решения в конфликтных ситуациях;

3.  с децентрализованным типом управления, при котором все управленческие процессы осуществляются только за счет локальных взаимодействий между агентами.

Для эффективного функционирования всей партнерской сети, работающей над выполнением виртуального проекта, агенты-партнеры должны базироваться на единой методологической основе и согласованно вести хозяйственную деятельность.

Если между партнерами уже существуют налаженные производственные связи и ясно, кто из них какую продукцию (услуги) и в какие сроки способен произвести и доставить в назначенную точку, то такое предприятие нельзя считать виртуальным.

Его эффективность при выполнении уникального заказа определяется простой способностью оперативно разместить частные задания у партнеров (или успешностью переговоров агентов по организации процесса выполнения заказа), а также добросовестностью выполнения партнерами своих обязательств.

Представители второго подхода, рассматривая возможные варианты организации внешнеэкономической деятельности с использованием Интернет-технологий, опираются на классические международные организационно-правовые формы. Указывают следующие группы критериев классификации виртуальных организаций:

1. юридические;

2. географические;

3. хозяйственно-экономические;

4. системно-сетевые.

Выделяются два главных класса виртуальных организаций: виртуальные корпорации и виртуальные товарищества (партнерства).

Виртуальная корпорация представляет собой электронное объединение капиталов (ресурсов) различного типа - финансового, технологического, человеческого (в частности, интеллектуального) в интересах выполнения сложных уникальных проектов, создания продукции мирового класса и максимально полного удовлетворения требований заказчика. Как и ее реальный правовой прототип, она способствует решению двух фундаментальных проблем рыночной экономики:

1. привлечению капитала для выполнения уникальных проектов или распределению бизнес-процессов с целью повышения конкурентоспособности продукции;

2. распределению риска в инвестиционных проектах.

Основная внешнеэкономическая цель создания виртуальной корпорации - объединение ключевых технологий и опыта партнеров разных стран для проведения более эффективных действий на мировом рынке. Виртуальная корпорация характеризуется определенной независимостью от участников (возможностью легкой смены партнеров), наличием опосредованного механизма управления (делегированием полномочий). Она предполагает договорные взаимоотношения между всеми узлами организационной сети и формирование их совместной собственности[[5]](http://www.vevivi.ru/best/Osobennosti-deyatelnosti-virtualnykh-organizatsii-ref158978.html" \l "_ftn9" \o "). Нередко виртуальная корпорация формируется в виде материнской виртуальной организации с сетью дочерних виртуальных филиалов, отделений и т.п.

Виртуальное товарищество (партнерство) представляет собой компьютерно - интегрированную (искусственную) организацию лиц, вместе ведущих дело (находящихся в отношениях кооперации, т.е. выполняющих совместную работу и координирующих действия) с целью извлечения прибыли, будучи географически удаленными друг от друга.

Следует отметить, что по своему содержанию понятие виртуального товарищества достаточно близко к представлениям о виртуальной рабочей группе. В данном случае каждый партнер, в той или иной степени участвуя в управлении и контроле за деятельностью виртуальной организации, несет индивидуальную ответственность за результаты работы, причем потеря партнера означает распад виртуальной организации.

Как и в случае обычного товарищества, для виртуального партнерства можно выделить две базовые формы: полное виртуальное товарищество, когда все партнеры равноправны в управлении предприятием и несут равную ответственность по ее обязательствам; ограниченное виртуальное партнерство, где один из партнеров имеет большие полномочия по контролю и управлению компанией и несет неограниченную ответственность по ее обязательствам, а другие партнеры не занимаются контролем и не отвечают по обязательствам партнерства.

Следующие исследователи читают, что среди важнейших критериев интеграции виртуальных организаций необходимо выделить объединение хозяйственной деятельности, отраслевую общность и юридическую самостоятельность. Опираясь на эти критерии, можно ввести основные типы виртуальных объединений: от наиболее «слабого» - виртуальной ассоциации до гипотетически самого «сильного» - виртуального треста.

В виртуальной ассоциации партнеры, находящиеся на удалении друг от друга, сотрудничают лишь при выполнении общих операций или функций. Здесь имеются две ключевые структурные характеристики: взаимозависимость между составляющими операциями и распределение ответственности между участниками.

Виртуальный консорциум близок по своим характеристикам к виртуальной ассоциации. Как правило, он создается в интересах реализации крупных программ или выполнения инновационных проектов. В рамках виртуального консорциума могут электронным способом объединяться предприятия разных стран, отраслей и форм собственности. При этом интеграция предполагает в первую очередь совместное выполнение функций и построение распределенной сети бизнес-процессов.

Виртуальный картель - это компьютерно - интегрированная форма объединения юридически самостоятельных организаций одной отрасли, имеющих соглашения о ценах, объемах производства, рынках сбыта.

Виртуальный синдикат представляет собой разновидность виртуального картеля, в рамках которого помимо вышеуказанных соглашений существует единый орган сбыта продукции участников виртуального объединения. Цель вступления в виртуальный синдикат - получить выигрыш от централизации сбыта. Синдикаты обычно образуются в отраслях с массовой однородной продукцией.

В свою очередь виртуальный пул означает полученное электронным путем временное объединение разных компаний (возможно разных отраслей), где для вступающих в пул организаций устанавливаются правила распределения общих расходов и прибыли, поступающей в единый фонд, распределяемый по установленной пропорции.

Виртуальный концерн предполагает электронное объединение компаний одной или нескольких отраслей на основе централизации научно-технических и производственных функций, сбыта, финансов, учета и пр. Участники делегируют концерну часть своих функций - тех, которые не могут выполнить сами, но остаются юридически самостоятельными.

Наконец, наиболее сильной формой компьютерной интеграции организаций оказывается виртуальный трест, когда все стороны деятельности входящих в него организаций объединяются, а сами они теряют юридическую и хозяйственную самостоятельность.

Одной из комплексных форм виртуальных предприятий является виртуальная финансово-промышленная группа. Как и обычная финансово-промышленная группа, она состоит из ряда неоднородных юридических лиц, удаленных друг от друга, которые полностью или частично объединяют ресурсы с применением Интернет-технологий на основе договора о создании финансово-промышленной группы для технологической или экономической интеграции. Как видно из самого названия, в состав виртуальной финансово-промышленной группы входят разные промышленные, торговые и финансовые учреждения (банки, страховые компании и пр.). Следует отметить, что обычно срок функционирования такого метапредприятия не ограничен рамками выполнения тех или иных проектов[6].

Для того чтобы деятельность виртуальной организации была продуктивной, необходимо координировать деятельность участников (агентов) с помощью уполномоченного на то органа, осуществляющего управленческие воздействия и регламентирующего деятельность предприятий-партнеров - органа-координатора. Его появление вызвано следующими причинами: высокая степень сложности процессов за счет кооперации различных организаций, наличие нескольких распределенных источников информации, которые необходимо скоординировать в режиме реального времени, нестабильная структура системы организации, расширение пространства поиска решений вследствие повышающейся комплексности процессов.

Обычно в виртуальную организацию входят: орган-координатор, занимающийся регламентацией деятельности виртуального предприятия; предприятия-поставщики, производители, транспортные и др. предприятия, принимающие участие, как в технологическом процессе, так и в процессе по доведению продукта до потребителя. Введение заказчика в систему взаимоотношений между субъектом и объектом управления дает ему возможность в режиме реального времени отслеживать работу предприятия и корректировать все действия: начиная от корректировки проекта товара или услуги до его производства и доставки.

Орган-координатор на организации виртуального типа должен выполнять следующие функции:

1.  разработка институций для виртуальной организации;

2.  поиск клиентов и анализ их запросов;

3.  синтез товарно-производственных решений, отвечающих требованиям запросов;

4.  регламентацию и корректировку деятельности предприятий, входящих в виртуальную организацию;

5.  управление ресурсами данных предприятий.

Особенностью функционирования виртуальной организации является то, что ее агенты могут работать над выполнением одного или нескольких проектов одновременно.

Агенты (организации-партнеры), осуществляющие работу над проектом, могут быть как сотрудниками одной организации, так и независимыми организациями, взаимодействующими друг с другом и заказчиком. Заметим, что среди агентов виртуальной организации необходимо выделить ответственных исполнителей (отвечающих за управление выделенной частью проекта) и исполнителей (непосредственно выполняющих работу).

Решить проблему управления предприятиями-партнерами виртуального предприятия можно в несколько этапов[[7]](http://www.vevivi.ru/best/Osobennosti-deyatelnosti-virtualnykh-organizatsii-ref158978.html" \l "_ftn11" \o "):

1.  этап идентификации виртуальной организации;

2.  этап формализации компетенций предприятий-партнеров;

3.  этап создания ролевой структуры и дифференциации статусов участников виртуальной сети.

Определимся с каждым из этих этапов более подробно.

1. Идентификацию любой виртуальной организации можно рассматривать в двух аспектах: идентификация агентов (предприятий-партнеров) виртуальной организации и идентификация виртуальной организации как единого проекта.

Процесс идентификации виртуальной организации является более простым, чем процесс идентификации ее членов. Для него необходимо определить цели и сроки виртуального проекта; определить мотивацию участников проекта; выделить основных участников и ответственного за проект; сообщить всем потенциальным участникам цели и сроки реализации проекта; запланировать и распределить работы и роли в виртуальном проекте; реализовать процесс включения в проект нового агента.

Орган-координатор может перераспределять работы наиболее активным предприятиям-партнерам сети, постоянно поддерживающим контакт в рамках данного сообщества. Как правило, любой из пассивных участников может покинуть организацию, в то время как любой новый член обычно пребывает в пассивном состоянии, анализируя происходящие в сети действия со стороны.

Таким образом, категории участников виртуальной сети можно представить следующим образом: орган-координатор: модератор и лидеры; активные участники; пассивные участники.

2. Поиск и привлечение потенциальных участников виртуальной сети является одним из самых важных моментов при инициализации проекта. В процессе формализации компетенций будущих предприятий-партнеров виртуальной организации целесообразно использовать каталог ключевых компетенций, включающий основные показатели каталога (внутренние и внешние компетенции, ключевые факторы успеха) и перечень компетенций участника виртуальной организации[[8]](http://www.vevivi.ru/best/Osobennosti-deyatelnosti-virtualnykh-organizatsii-ref158978.html" \l "_ftn12" \o ").

В дальнейшем поиск может осуществляться по другим критериям: опты работы на рынке, регион и т.д. Оценить опыт потенциального агента в рамках данной компетенции возможно с помощью системы статусов или структуры описаний компетенций участников виртуальной организации.

3. Как в традиционной, так и в виртуальной организациях любой участник может выполнять разные роли в зависимости от его задействованности в виртуальном проекте. Участники виртуального пула могут выполнять следующие роли:

1. Орган-координатор: разработка институций для виртуальной организации; поиск клиентов и анализ их запросов; синтез товарно-производственных решений, отвечающих требованиям запросов; регламентация и корректировка деятельности предприятий, входящих в виртуальную организацию; управление ресурсами данных предприятий;

2. Эксперт: носитель специфических знаний; отвечает на вопросы участников виртуальной организации; разрабатывает новые продукты;

3. Партнер выполняет определенные задания в рамках виртуального проекта с учетом своей ключевой компетенции[[9]](http://www.vevivi.ru/best/Osobennosti-deyatelnosti-virtualnykh-organizatsii-ref158978.html" \l "_ftn13" \o ").

**2.3. Особенности управления организацией, достоинства и недостатки виртуальных организаций**

Для планирования, организации и координации деятельности виртуальных предприятий необходимы и соответствующие управленческие подходы. При создании виртуальных организаций нередко появляются организации, которые концентрируют свои усилия исключительно на управлении компетенциями третьей стороны. В данном случае такая организация должна обладать как минимум следующими способностями: уметь идентифицировать и привлекать ключевые компетенции, необходимые для реализации проекта (аспекты менеджмента знаний) и на основе привлеченных компетенций организовать процесс создания и сбыта продукции (аспекты функционирования сети).

На основе этого можно в общем виде сформулировать основные функции управления виртуальной организацией как сетью партнеров:

1. Определение требований (задач) проекта.

2. Поиск и оценка возможных партнеров (исполнителей).

3. Выделение исполнителей, которые оптимально соответствуют задачам.

4. Привлечение и распределение исполнителей.

5. Постоянное отслеживание и перераспределение (если это необходимо) партнеров и ресурсов по задачам[[10]](http://www.vevivi.ru/best/Osobennosti-deyatelnosti-virtualnykh-organizatsii-ref158978.html" \l "_ftn14" \o ").

Вышеуказанные особенности виртуальных организаций формируют следующие требования к их сотрудникам:

1.  умение ясно формулировать задачи, выражать информацию. В международных организациях от сотрудников требуется знание иностранного языка;

2.  хорошее владение навыками письменной, электронной коммуникации;

3.  более широкий спектр знаний, «контекстное» мышление (поскольку сотрудники работают не над одним проектом, а могут быть включены одновременно в несколько проектов, решать параллельно несколько задач);

4.  умение устанавливать цели, структурировать собственное время;

5.  умение находить нестандартные решения тех или иных задач;

6.  большая персональная ответственность за результаты работы, в связи с чем сотрудники должны хорошо знать соответствующие инструкции, основы законодательства, этические нормы компании;

7.  сотрудникам необходимо представлять как организацию, частью которой они являются, так и группу, в которую они включены;

8.  способность успешно сотрудничать в виртуальной среде (что не только обусловливается психологическими особенностями индивидов, но и тесно связано с корпоративной культурой и применяемыми технологиями управления).

Управление персоналом в виртуальной организации предполагает работу в гибкой инновационной среде, что требует нестандартных подходов и решений. Изменение задач, стоящих перед организацией, требует изменения конфигурации команд, обновления и замены управленческих систем. Управление человеческими ресурсами виртуальных организаций в большей мере, чем другая управленческая деятельность в них, предъявляет повышенные требования к профессионализму лидеров, их умению работать с людьми, концептуальным способностям, умению инициировать и мотивировать инновационную активность сотрудников. Вместе с тем менеджер должен уметь передавать видение перспектив компании своим подчиненным и мотивировать их на достижение целей компании.

В виртуальной организации обретают свою специфику и технологии управления персоналом: организация найма и отбора персонала, оценка, адаптация, обучение, управление карьерой, мотивация и организация труда, управление конфликтами и стрессами и др.

Успешное управление командой сотрудников в виртуальной организации предполагает:

1) необходимость определения целевых установок для каждого сотрудника, в соответствии с которыми будет оцениваться результат его работы, вклад в реализованный проект организации;

2) создание возможности накопления специфического человеческого капитала в организации посредством дистанционного обучения сотрудников, содействия получению и сохранению новых навыков;

3) измерение конечного результата деятельности, а не процесса;

4) создание и поддержание атмосферы доверия в команде. «Работники, облеченные доверием, — мощный актив, наличие которого означает, что на всех стадиях процесса исследований, принятия решений и проведения их в жизнь вы работаете с личностями, а не с послушными роботами». Доверие становится важным ресурсом, который не кодифицируется, но выступает незримым гарантом соблюдения договора между работником и работодателем.

Таким образом, важнейшими аспектами, которые следует принимать во внимание HR-менеджеру виртуальной организации, являются следующие: инфраструктура и возможности эффективной коммуникации; психологические особенности персонала; особенности управления (методы планирования, организации, контроля производительности труда, мотивации, управление доверием и др.) и знание особенностей национальных деловых культур.[11]

Ключевым достоинством виртуальных форм организаций является: возможность выбирать и использовать наилучшие ресурсы, знания и способности с меньшими временными затратами. Из этого достоинства и самой сетевой организации вытекают такие основные конкурентные преимущества виртуальных предприятий, как:

1.  скорость выполнения рыночного заказа;

2.  возможность снижения совокупных затрат;

3.  возможность более полного удовлетворения потребностей заказчика;

4.  возможность гибкой адаптации к изменениям окружающей среды;

5. Выпуск на рынок нового продукта или услуги становится под силу гораздо меньшим по размеру организациям, располагающим лишь долей того капитала, который требуется при традиционном способе;

6.Местоположение партнеров виртуального предприятия утратит актуальность, что активизирует международное сотрудничество и приведет к более интенсивному перемещению деятельности между странами и регионами.[12]

Основными характеристиками виртуальной формы организации являются: открытая распределенная структура; гибкость; приоритет горизонтальных связей; автономность и узкая специализация членов сети; высокий статус информационных и кадровых средств интеграции.

Виртуальные организации создаются не столько людьми интеллектуального труда, сколько людьми, имеющими представление о создании структуры, которая бы юридически была не идентифицируемой и при этом работоспособной. Почти все виртуальные организации предполагают использование компьютерной коммуникации при посредстве компьютерных сетей. Проблемы, которые возникают при этом, можно кратко сформулировать так:

1. Последствия для каждого участника от принципиального повышения качества текущей интеллектуальной работы и оплаты за такую работу;

2. Условия разработки и освоения новых интеллектуальных инструментов или товаров, освоения новых сфер участия фирмы на рынке, финансирования исследовательского труда;

3. Объем полномочий и допустимость санкционированного вмешательства стратегического менеджмента со стороны проектной группы по отношению к основной структуре виртуальной организации;

4. Каким образом должна быть оформлена любая новая идея (проект, этапы, ресурсы, задействованные структуры виртуальной организации);

5. Условия субренты (отчисления, процент с прибыли, увеличение оплаты по контракту) от реализации проекта;

6.Ограниченность контроля и отсутствие четких границ;

7.Дополнительные требования к менеджерам;

8.Коммуникативные трудности, потенциаль­ное непонимание; [13]

**2.4. Принцип построения виртуальных организаций**

Прежде чем приступать к построению виртуального предприятия необходимо решить следующие вопросы:

1. Определить потребность в основных ресурсах:

-проанализировать структуру производства товаров и услуг;

-оценить роль собственных возможностей на каждом этапе производственного процесса.

2.Установить сотрудничество с партнерами в целях достижения максимального синергического эффекта:

-определения характера партнерства;

-географических границ выбора партнеров;

-требований к ним;

-перечня функций которые они должны выполнять.

Последовательность этапов создания виртуального предприятия:

1.Проектирование производственных процессов;

2.Привязка процессов к ресурсам, требующимся для их поддержки;

3.Привлечение ресурсов других компаний, когда это целесообразно;

4.Эксплуатация виртуальной организации;

5.Мониторинг производственных процессов.[14]

**3. ПРОЕКТИРОВАНИЕ СИСТЕМЫ**

**3.1. Проектирование базы данных**

**3.1.1. Анализ предметной области**

Разрабатываемое в данном дипломном проекте ПО предназначено для информационной поддержки сотрудников виртуальных предприятий при выполнении ими своих обязанностей. Программа должна помочь сотрудникам быстро и легко получать необходимую информацию от других сотрудников а также контролировать свой рабочий процесс.

Первый этап в работе системы будет регистрацией виртуального предприятия в ней. Регистрацию удобней всего сделать с помощью электронной почты. Пользователь вводит свой адрес почты и на него приходит регистрационное письмо.

Дальше пользователь, зарегистрировавший виртуальное предприятия рассылает приглашения на почту сотрудников виртуального предприятия. Сотрудники регистрируются и получают доступ к системе.

Далее рассмотрим возможности которые должна обеспечивать система.

Система должна обеспечивать обмен задачами между сотрудниками предприятия.

Задачи будут разделяться на два типа:

1. Задачи написанные сотрудником;
2. Задачи данные ему другими сотрудниками на выполнение.

Любой сотрудник может добавить новую задачу в систему, причем авторство задачи закрепляется за сотрудником. Автор задачи может изменять название задачи, описание задачи, сроки ее исполнения удалить ,назначить исполняющего.

После назначения исполняющего задачи, задача переходит в очередь выполнения и появляется в списке у сотрудника, который был указан в качестве исполняющего.

Второй важной функцией которую должна выполнять система это коммуникация сотрудников. У каждого сотрудника должен отображаться раздел сообщения в котором есть три подпункта:

* личные сообщения;
* подписки;
* отправить сообщение подписчикам.

В личных сообщениях можно выбрать любого сотрудника предприятия и отправить ему сообщение. В подписках видны сообщения от сотрудников на информацию от которых вы подписались. Также там можно подписаться на сотрудников.

Третья важная функция которую должна обеспечивать система является анализ работы виртуального предприятия. Тут предполагается два вида анализа:

1. вклад участника в проекты;
2. вклад участников в проект.

Первый вид анализа дает возможность выбрать сотрудника и диапазон дат за которые надо показать его активность. Второй вид анализа позволяет выбрать проект и диапазон дат за который нужно посмотреть кто в нем участвовал.

**3.1.2. Выделение сущностей**

Метод «сущность-связь», или ER-метод, является наиболее формализованным и простым для понимания. Суть метода состоит в построении ER-диаграмм, отображающих в графической форме основные объекты ПО и связи между ними, и в определении характеристик этих связей [15].

Основными понятиями модели «сущность-связь» являются следующие.

* ***Сущность*** – это собирательное понятие, некоторая абстракция реально существующего объекта, процесса или явления, информацию о котором необходимо хранить в базе данных. Экземпляр сущности – это информация о конкретном представителе объекта.
* ***Связь*** – это соединение между двумя или более сущностями. Экземпляр связи – это конкретная связь между конкретными представителями объектов.
* ***Атрибут*** представляет собой свойство сущности.
* ***Ключ сущности*** – атрибут или набор атрибутов, используемый для идентификации экземпляра сущности.

Необходимо заметить, что для всех сущностей необходимо добиться нужной степени детализации для возможности реализации функциональных зависимостей. Для сущностей, использующих длинные текстовые значения целесообразно вводить дополнительный атрибут, что позволит сократить затраты оперативной памяти, упростить и ускорить процесс обработки данных.

Определим предварительные сущности предметной области:

***Виртуальное предприятие*** –хранит в себе информацию о сотрудниках и проектах предприятия. Для данной сущности введем дополнительный атрибут *Код ВП.*

***Сотрудник*** - хранит информацию о сотруднике виртуального предприятия. Для данной сущности введем дополнительный атрибут *Код сотрудника.*

***Задача -*** хранит информацию о задаче созданной сотрудником. Для данной сущности введем дополнительный атрибут *Код задачи.*

***Проект*** - хранит информацию о задачах и сотруниках участвующих в проекте. Для данной сущности введем дополнительный атрибут *Код проекта.*

***Время на проект -*** хранит информацию о времени потраченном каждым сотрудником на выполнение проекта.Для данной сущности введем дополнительный атрибут *Код.*

***Сообщение*** - хранит сообщение конкретного сотрудника. Для данной сущности введем дополнительный атрибут *Код сообщения*.

***Диалог*** - хранит сообщения передаваемые между двумя сотрудниками.

Выделим связи между перечисленными сущностями. Для данной сущности введем дополнительный атрибут *Код диалога*.

***Подписка*** - хранит подписки сотрудника. Для данной сущности введем дополнительный атрибут *Код подписки*.

***Сообщение от подписки –*** хранит сообщения которые рассылаются по подписке. Для данной сущности введем дополнительный атрибут *Код сообщения подписки*.

Выделим связи между перечисленными сущностями.

**3.1.3. Выявление связей между сущностями**

***Сотрудник\_работает*** –связывает сущности *Виртуальное предприятие* и *Сотрудник.* Показывает что сотрудник работает в данном виртуальном предприятии (см. рис. 3.1). Связь – многие ко многим, так как сотрудник может работать в нескольких виртуальных предприятиях и в ВП могут работать много сотрудников. Виртуальное предприятие должно иметь хотя бы 1 сотрудника(зарегистрировавшего ВП) поэтому связь с сотрудником обязательна. Сотрудник же может не входить ни в одно ВП поэтому для него связь не обязательная.

Виртуальное предприятие

Сотрудник

N

***Сотрудник\_работает***

N

Рис. 3.1. ER-диаграмма бинарной связи **сотрудник\_работает**

***ВП\_содержит*** –связывает сущности *Проект* и *Виртуальное предприятие.* Показывает что в виртуальном предприятии есть данный проект (см. рис. 3.2).Связь –один ко многим,так как проекты создаются конкретно для каждого виртуального предприятия и в то же время в ВП может быть много проектов. ВП может не содержать проектов, поэтому связь с проектом не обязательная. Проект же должна обязательно находиться в проекте поэтому для нее связь обязательная.

Виртуальное предприятие

Проект

1

***ВП\_содержит***

N

Рис. 3.2. ER-диаграмма бинарной связи **ВП\_содержит**

***Проект\_содержит*** –связывает сущности *Проект* и *Задача.* Показывает что в проекте есть данная задача (см. рис. 3.3).Связь –один ко многим,так как задачи создаются конкретно для каждого проекта и в то же время в проекте может быть множество задач. Проект может не содержать задач, поэтому связь с задачей не обязательная. Задача же должна обязательно находиться в проекте поэтому для нее связь обязательная.

Проект

Задача

1

***Проект\_содержит***

N

Рис. 3.3. ER-диаграмма бинарной связи **Проект\_содержит**

***Диалог\_содержит*** – связывает сущности *Диалог* и *Сообщение.* Показывает что в данном диалоге есть данное сообщение (см. рис. 3.4).Связь один ко многим,так как в одном диалоге может быть много сообщений и в то же время сообщение может быть только в 1 диалоге. Диалог может не содержать сообщений, поэтому связь с сообщением не обязательная. Сообщение же должна обязательно быть привязано к конкретному диалогу поэтому для него связь обязательная.

Диалог

Сообщение

1

***Диалог\_содержит***

N

Рис. 3.4. ER-диаграмма бинарной связи **Диалог\_содержит**

***Сотрудник\_Подписан*** – связывает сущности *Сотрудник* и *Подписка.* Показывает что сотрудник подписан на данную подписку (рис. 3.5).Связь многие ко многим. У одного сотрудника может быть много подписок ,в тоже время одна и та же подписка может быть у нескольких сотрудников.

Сотрудник

Подписка

N

**Сотрудник\_Подписан**

N

Рис. 3.5. ER-диаграмма бинарной связи **Сотрудник\_Подписан**

***Диалог\_сотрудника –*** связывает сущности *Сотрудник* и *Диалог* (см. рис. 3.6*)****.*** Показывает что сотрудник участвует в данном диалоге. Связь многие ко многим.Сотрудник может не иметь сообщений поэтому связь с диалогом для него не обязательная.

Сотрудник

Диалог

N

**Диалог\_сотрудника**

N

Рис. 3.6. ER-диаграмма бинарной связи **Диалог\_сотрудника**

**Сотрудник\_потратил –** связывает сущности *Время на проект* и *Сотрудник (*см.рис. 3.7*)*. Показывает что сотрудник потратил рабочее время на проект. Связь многие ко многим так как сотрудник может потратить время много раз на проект и сотрудников в проекте также может быть много. Сотрудник мог и не потратить времени на проект поэтому связяь с временем на проект не обязательна,а вот у времени на проект должен быть сотрудник поэтому связь обязательна.

Сотрудник

Время на проект

1

**Сотрудник\_потратил**

N

Рис. 3.7. ER-диаграмма бинарной связи **Сотрудник\_потратил**

**Потрачено –** связывает сущности *Время на проект* и *Проект (см.* рис. 3.8*).* Показывает что на проект было потрачено время. Связь один ко многим.

Проект

Время на проект

1

**Потрачено**

N

Рис. 3.8. ER-диаграмма бинарной связи **Потрачено**

**Автор –** связывает сущности *Сотрудник* и *Задача*. Показывает что сотрудник автор задачи (см. рис. 3.9).Связь один ко многим так как у задачи может быть только один автор, а у сотрудника может быть много задач. Также у сотрудника может не быть задач поэтому связь не обязательна. У задачи же обязательно должен быть автор поэтому связь обязательна.

Сотрудник

Задача

1

**Автор**

N

Рис. 3.9**.** ER-диаграмма бинарной связи **Автор**

**Пришло** – связывает сущности *Подписка* и *Сообщение от подписки (*см. рис. 3.10*).* Показывает связь между сообщением от подписки и подпиской. Связь один ко многим так как от одной подписки может быть много сообщений,а у сообщения может быть только одна подписка.

Подписка

Сообщение от подписки

1

**Пришло**

N

Рис. 3.10**.** ER-диаграмма бинарной связи **Пришло**

**Участник** – связывает сущности *Сотрудник* и *Проект (*см. рис. 3.11*).* Показывает что сотрудник является участником проекта. Связь многие ко многим так как сотрудник может одновременно находиться в нескольких проектах, а в самом проекте обычно находится много сотрудников. Связь не обязательна с обеих сторон так как сотрудник может не участвовать в проекте и над проектом также может никто не работать.

Сотрудник

Проект

N

**Участник**

N

Рис. 3.11. ER-диаграмма бинарной связи **Участник**

**3.1.4 Формирование набора предварительных отношений**

Произведем формирование набора предварительных отношений с указанием предполагаемого первичного ключа для каждого отношения, используя диаграмму ER- типов.

Сформулируем правила формирования предварительных отношений.

* Если степень связи 1:N, односвязная сущность имеет любой класс принадлежности, многосвязная сущность имеет обязательный класс принадлежности, создается два предварительных отношения по одному на каждую сущность с их ключами. Затем отношения многосвязной сущности пополняется ключом односвязной сущности.
* Если степень связи N:N, то независимо от класса принадлежности сущностей требуется формирование трех отношений. Два отношения соответствуют связываемым сущностям и их ключи являются первичными ключами этих отношений. Третье отношение является связанным между первыми двумя, а его ключ объединяет ключевые атрибуты связываемых отношений.
* Если связь существует больше чем 2-мя объектами, то количество отношений обычно (n+1).

Применим правила и получим набор предварительных отношений.

1. Связь ***Сотрудник\_работает*** порождает 3 отношения:

* *Сотрудник,* первичный ключ *Код Сотрудника*;
* *Виртуальное предприятие,* первичный ключ *Код ВП;*
* *Сотрудник предприятия,* первичный ключ *Код Сотрудника, Код ВП,* внешние ключи *Код Сотрудника, Код ВП.*

1. Связь ***ВП\_содержит*** порождает 2 отношения:

* *Виртуальное предприятие*, первичный ключ *Код ВП*;
* *Проект,* первичный ключ *Код проекта*, внешний ключ *Код\_ВП.*

1. *Связь* ***Проект\_содержит*** порождает 2 отношения***:***

* *Проект,* первичный ключ *Код проекта*;
* *Задача,* первичный ключ *Код задачи*,внешний ключ *Код проекта.*

1. Связь ***Диалог\_содержит*** порождает 2 отношения:

* *Диалог,* первичный ключ *Код диалога;*
* *Сообщение*, первичный ключ *Код сообщения*, внешний ключ *Код диалога*.

1. Связь ***Сотрудник\_Подписан*** порождает 3 отношения:

* *Сотрудник*, первичный ключ *Код сотрудника*;
* *Подписка сотрудника,* первичный ключ *Код подписки ,*внешние ключи *Код подписанного сотрудника, Код сотрудника на которого подписались*;
* *Подписка,* первичный ключ *Код подписки.*

1. Связь***Диалог\_сотрудника*** порождает 3 отношения:

* *Диалог,* первичный ключ *Код диалога;*
* *Диалог сотрудника,* первичный ключ *Код диалога сотрудника*, внешние ключи *Код диалога, Код сотрудника;*
* *Сотрудник*, первичный ключ *Код сотрудника*.

1. Связь ***Сотрудник\_потратил*** порождает 2 отношения:

* *Сотрудник*, первичный ключ *Код сотрудника*;
* *Время на проект,* первичный ключ *Код.*

1. Связь***Потрачено*** порождает два отношения:

* *Время на проект,* первичный ключ *Код*;
* *Проект,* первичный ключ *Код проекта*.
  + - 1. Связь***Автор***порождает два отношения:
* *Сотрудник*, первичный ключ *Код сотрудника*;
* *Задача,* первичный ключ *Код задачи*,внешний ключ *Код автора.*
  + - 1. Связь***Пришло*** порождает два отношения:
* *Подписка,* первичный ключ *Код подписки;*
* *Сообщение от подписки,* первичный ключ *Код сообщения подписки*, внешний ключ *Код подписки.*

1. Связь ***Участник*** порождает 3 отношения:

* *Проект,* первичный ключ *Код проекта*;
* *УчастникПроекта,* первичный ключ *Код участника проекта*, внешние ключи *Код проекта*, *Код сотрудника;*
* *Сотрудник,* первичный ключ *Код сотрудника*.

После добавления неключевых атрибутов схемы отношений примут следующий вид (таблица 3.1).

Таблица 3.1. Схемы отношений после добавления неключевых атрибутов

|  |  |  |
| --- | --- | --- |
| **Отношение** | **Первичный ключ** | **Атрибуты** |
| Виртуальное предприятие | Код ВП | Название |
| Сотрудник | Код сотрудника | Имя  Фамилия  Отчество  Почта  Пароль  Должность |
| Сотрудник предприятия | Код сотрудника предприятия | Код сотрудника  Код ВП |
| Проект | Код проекта | Название  Срок сдачи  Код ВП |

Продолжение таблицы 3.1

|  |  |  |
| --- | --- | --- |
| Задача | Код задачи | Название  Код автора  Дата создания  Срок выполнения  Потраченное время  Код исполнителя  Код проекта  Задача выполнена |
| Сообшение | Код сообщения | Содержимое  Код Диалога  Код Сотрудника  Время отправки |
| Подписка | Код подписки | Код подписанного сотрудника  Код сотрудника на которого подписаны |
| Диалог | Код диалога | Код диалога сотрудника |
| Диалог сотрудника | Код диалога сотрудника | Код сотрудника |
| Время на проект | Код | Дата  Код сотрудника  Код проекта  Количество часов |
| Сообщение от подписки | Код сообщения подписки | Код подписки  Содержимое  Время отправки |
| Участник проекта | Код участника проекта | Код проекта  Код сотрудника |

**3.1.5 Разработка физической модели базы данных**

Для разработки проекта выбираем систему управления базами данных Microsoft SQL Server 2012, к достоинствам которой можно отнести:

* реляционную модель данных;
* языки конструирования запросов SQL и QBE;
* совместимость с предыдущими и последующими версиями;
* возможность удаленного доступа к данным;
* возможность экспорта/импорта данных в другие СУБД и приложения.

Разработаем схемы всех отношений, полученных на предыдущем этапе проектирования, результат представим в виде таблицы 3.2.

Таблица 3.2. Схема базы данных

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Имя отношения** | **Имя таблицы** | **Атрибут** | **Имя атрибута** | **Тип** | **Обязательный** |
| Виртуальное предприятие | VirtualEnterprise | Код ВП | idVE | Int | + |
| Название | name | NvarChar(128) | + |
| Сотрудник | Employees | Код сотрудника | employeeId | int | + |
| Имя | name | NvarChar(32) | + |
| Фамилия | family | NvarChar(64) | - |
| Отчество | patronymic | NvarChar(64) | - |
| Почта | mail | NvarChar(64) | + |
| Пароль | passwordEmployee | NvarChar(32) | + |
| Должность | position | NvarChar(64) | - |
| Сотрудник предприятия | CompanyEmployee | Код сотрудника предприятия | idCompanyEmployee | Int | + |
| Код сотрудника | idEmployee | Int | + |
| Код ВП | IdVE | int | + |
| Проект | Project | Код проекта | idProject | int | + |
| Название | name | NvarChar(64) | + |
| Срок сдачи | deadline | datetime | - |
| Код ВП | idVE | int | + |
| Задача | Tasks | Код задачи | taskId | Int | + |
| Код автора | idAuthor | int | + |
| Название | name | NvarChar(128) | + |
| Дата создания | dataCreation | datetime | + |
| Срок выполнения | deadline | datetime | - |
| Потраченное время | wastedTime | int | - |
| Исполнитель | idExecutor | int | - |
| Код проекта | idProject | int | + |
| Задача выполнена | isTaskComplete | bit | + |

Продолжение таблицы 3.2

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Сообщение | Mesage | Код сообщения | idMessage | int | + |
| содержимое | content | NvarChar(1024) | + |
| Код диалога | idDialogue | int | + |
| Время отправки | dateStart | datetime | + |
| Подписка | Subscription | Код подписки | idSubscription | int | + |
| Код подписанного  сотрудника | idSignedEmployee | int | + |
| Код сотрудника на которого подписаны | idFellowSubscribed | int | + |
| Диалог | Dialogue | Код диалога | idDialogue | int | + |
| Код диалога сотрудника | idDialogueEmployee | int | + |
| Диалог сотрудника | DialogueEmployee | Код диалога сотрудника | idDialogueEmloyee | int | + |
| Код сотрудника | idEmployee | int | + |
| Время на проект | TimeOnProject | Код | id | int | + |
| Дата | date | date | + |
| Код сотрудника | idEmployee | int | + |
| Код проекта | idProject | int | + |
| Количество часов | countTime | int | + |
| Сообщение от подписки | PostSuscription | Код сообщения подписки | IdPostSubscription | int | + |
| содержимое | content | NvarChar(1024) | + |
| Время отправки | dateStart | datetime | + |
| Код подписки | idSubscription | int | + |

Окончание таблицы 3.2

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Участник проекта | ParticipantProject | Код участника проекта | idParticipantProject | int | + |
| Код сотрудника | idEmployee | int | + |
| Код  проекта | idProject | int | + |

Схема связей базы данных будет иметь следующий вид, представленный на рис. 3.11.

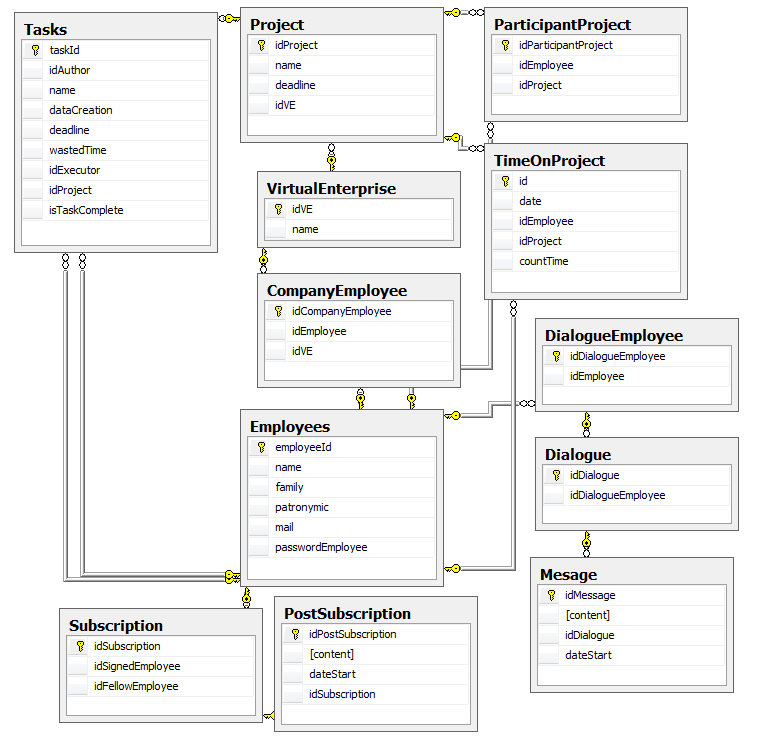


Рис. 3.11. Схема связей базы данных

**3.1.6. Разработка средств поддержания целостности данных**

Термин *целостность данных* относится к правильности и полноте информации, содержащейся в базе данных. При модификации содержимого базы данных (изменении, добавлении или удалении строк таблицы) может произойти нарушение целостности содержащихся в ней данных. Например:

* в базу могут быть внесены неправильные (или несуществующие) данные;
* в результате изменения существующих данных им могут быть присвоены некорректные значения;
* при внесении изменений в базу данных они могут быть утеряны из-за системной ошибки или сбоя в электропитании;
* при внесении изменений в базу данных они могут быть внесены лишь частично.

Поэтому для сохранения непротиворечивости и правильности хранимых данных в реляционных СУБД устанавливается одно или несколько условий целостности данных. Эти условия определяют, какие значения могут быть записаны в базу данных в результате добавления или обновления данных. Как правило, в реляционной базе данных можно использовать следующие типы условий целостности данных:

* обязательность данных. Некоторые столбцы в базе данных должны содержать значения в каждой строке; строки в таких столбцах не могут содержать значения NULL или не содержать никакого значения;
* проверка на правильность. У каждого столбца в базе данных есть свой домен, т.е. набор значений, которые допускается хранить в данном столбце;
* целостность таблицы. Первичный ключ таблицы должен в каждой строке иметь уникальное значение, отличное от значений во всех остальных строках;
* ссылочная целостность. В реляционной базе данных каждая строка таблицы-потомка с помощью внешнего ключа связана со строкой таблицы-предка, содержащей первичный ключ, значение которого равно значению внешнего ключа;
* непротиворечивость. Многие реальные операции вызывают в базе данных несколько изменений одновременно. Все они должны быть выполнены, чтобы база данных осталась в правильно, непротиворечивом состоянии;
* корректность. Обновление информации в базе данных может быть ограничено правилами, которым подчиняются данные, представленные этими обновлениями;

**Обязательность данных.** Это условие целостности данных требует, чтобы некоторые столбцы не содержали значений NULL. Такие столбцы обозначены символом «+» в колонке «Обязательный» таблиц 2.2. Данное ограничение целостности обеспечивается установкой свойства «Allow Nulls» столбца в состояние «False» при создании таблицы в Microsoft SQL Server 2012.

**Проверка на правильность.** При создании таблицы за каждым столбцом закрепляется определенный тип данных, и СУБД следит за тем, чтобы в столбец вводились данные только этого типа.

**Целостность таблицы.** Каждая строка таблицы должна иметь уникальное значение первичного ключа, иначе база данных потеряет свою целостность и перестанет быть адекватной моделью внешнего мира. Уникальность ключа обеспечивает специальный тип данных СУБД *Microsoft SQL Server 2012 – identity*, который является типом первичного ключа большинства таблиц.

**Ссылочная целостность.** Существует четыре типа изменений базы данных, которые могут нарушить ссылочную целостность отношений предок/потомок. Рассмотрим каждую из них.

*Добавление новой строки-потомка*. Когда происходит добавление новой строки в таблицу потомок, значение его внешнего ключа должно быть равно одному из значений первичного ключа в таблице-предке. Если значение внешнего ключа не равно ни одному из значений первичного ключа, то добавление такой строки разрушит базу данных, поскольку появится потомок без предка. Но добавление строки в таблицу предок не вызовет проблем; она просто станет предком без потомков. В разработанной базе данных невозможно добавление строки-потомка без внешнего ключа, имеющего место в качестве первичного ключа в таблице-предке.

*Обновление внешнего ключа в строке-потомке*. Это та же проблема, что и в предыдущей ситуации, но выраженная в иной форме. Так обновление внешнего ключа в строке-потомке возможно только на то значение, которое имеет место в качестве первичного ключа в таблице-предке.

*Удаление строки-предка*. Если из таблицы-предка будет удалена строка, у которой есть хотя бы один потомок, то строки-потомки окажутся без строк-предков. Значения внешних ключей в этих строках больше не будут равны ни одному из значений первичного ключа таблицы-предка. Но удаление строки из таблицы-потомка не вызовет проблем; просто предок этой строки после удаления будет иметь на одного потомка меньше. В разработанной базе данных запрещено удаление строк из таблиц-предков, имеющих строки-потомки.

*Обновление первичного ключа в строке предка*. Это иная форма проблемы, рассмотренной в предыдущем пункте. Если происходит изменение первичного ключа некоторой строки в таблице-предке, все существующие потомки этой строки окажутся без строки-предка, поскольку их внешние ключи не равны ни одному первичному ключу. Подобные действия не имеют места в разработанной базе данных.

**Корректность данных****.** В реальной жизни вопрос целостности данных часто бывает связан с порядками и правилами, установленными в конкретных организациях. Установление корректности проводится совместно с заказчиком, при этом уточняются действия, выполняемые при модификации базы данных. В результате были выявлены следующие ограничения:

* в поля, требующие ввода текстовых данных, не могут быть введены пустые значения.

Все эти ограничения реализованы при разработке клиентской части программы в среде *Microsoft Visual Studio 2013*.

**3.2. Основные технологии, применяемые при разработке системы**

Прежде, чем приступить к разработке схем программ, отметим, что средой разработки программного обеспечения клиентской части является *Microsoft Visual Studio 2013.* Создание программ в среде *Microsoft Visual Studio 2010* базируется на объектно-ориентированной технологии. В связи с чем, отметим ряд его достоинств.

*Microsoft Visual Studio 2013* позволяет:

* уменьшить сложность программного обеспечения;
* повысить надежность программного обеспечения;
* обеспечить возможности модификации отдельных компонентов программного обеспечения без изменения остальных его компонентов;
* обеспечить возможности повторного использования отдельных компонентов программного обеспечения.

Среда *Microsoft Visual Studio 2013* обладает мощной библиотекой визуальных и не визуальных компонентов, позволяющих облегчить разработку пользовательского интерфейса и программ в целом. Визуальные компоненты предназначены для организации пользовательского интерфейса. Не визуальные компоненты отвечают за доступ к системным ресурсам.

В качестве СУБД выбрана *Microsoft SQL Server 2008*, что позволяет использовать при разработке приложения архитектуру «клиент-сервер».

В разрабатываемой программе будет применяться трехзвенная архитектура.

Основные достоинства клиент-серверной архитектуры:

* высокая эффективность обработки данных;
* низкая нагрузка на сеть (по сравнению с файловыми системами);
* все функции по управлению базой данных (хранение, защита, резервирование) берет на себя СУБД, что повышает надежность хранения данных [16].

В качестве средства доступа к базе данных была выбрана технология *ActiveX Data Objects* (ADO). Объекты ADO позволяют писать компактные и расширяемые коды для подключения к источникам данных, например, базам данных, электронным таблицам, последовательным файлам данных или каталогам электронной почты.

Для отсылки данных на почтовый ящик будет использоваться протокол smtp.

Взаимодействие клиентского приложения с базой данных осуществляется следующим образом. Пользователь запускает клиентское приложение. Клиент устанавливает соединение с сервером. Затем клиент запрашивает данные из базы данных. Полученные данные ображаются в соответствующем компоненте представления данных. Пользователь взаимодействует с клиентским приложением, и данные изменяются. Измененные данные отправляются на сервер баз данных, где сохраняются.

Определив основные технологии, необходимые при разработке программы, перейдем к разработке её схемы.

**3.3. Разработка пользовательского интерфейса**

Пользовательский интерфейс разрабатывается при помощи визуальных средств визуального программирования входящих в состав *Microsoft Visual Studio 2013.*

На основе анализа предметной областиразработаем интерфейс приложения.Начало работы с программой начинается с регистрации компании или входа в систему если регистрация уже проведена. После входа в систему пользователь попадает на главную форму приложения. Сверху будет находиться главное меню со следующими пунктами и подпунктами:

* + 1. Список проектов
    2. Задачи
       1. Необходимо выполнить;
       2. Мои задачи;
    3. Сообщения
       1. Личные сообщения;
       2. Подписки;
       3. Отправить сообщение подписчикам;
    4. Аналитика
       1. Вклад участника в проекты;
       2. Вклад участника в проект;
    5. Компания.

Перемещение между меню будет происходить на одной форме, будут меняться только элементы управления.

Приведем список используемых для разработки пользовательского интерфейса визуальных компонент и их назначение (таблица 3.3).

Таблица 3.3. Компоненты, используемые для разработки интерфейса

| **Компонент** | **Назначение** |
| --- | --- |
| Button | Кнопка |
| ComboBox | Список |
| CheckBox | Индикатор |
| DateTimePicker | Компонент ввода даты |
| TextBox | Однострочный редактор |

Продолжение таблицы 3.3

|  |  |
| --- | --- |
| GroupBox | Поле группировки |
| Label | Надпись |
| LinkLabel | Надпись с гиперссылкой |
| MenuStrip | Меню для формы |
| Panel | Используяется для группировки элементов |
| RadioButton | Кнопка с пиктограммой |
| RichTextBox | Многострочный редактор |
| ReportViewer | Создание отчетов |
| ShowPrintButton | Кнопка печати |
| ShowZoomControl | Изменение масштаба |

С использованием приведенных компонент строится интерфейсная часть системы. Вид и структура пользовательского интерфейса будут приведены в программной документации.

Список всех созданных форм и их описание приведены в таблице 3.4.

| **Форма** | **Описание** |
| --- | --- |
| MainMenu | Главная форма программы, используется для отображения основной информации |
| Login | Форма для регистрации и входа в систему |
| ChangeTask | Форма для изменения задачи |
| ChangeEmployee | Форма для изменения данных о сотруднике |

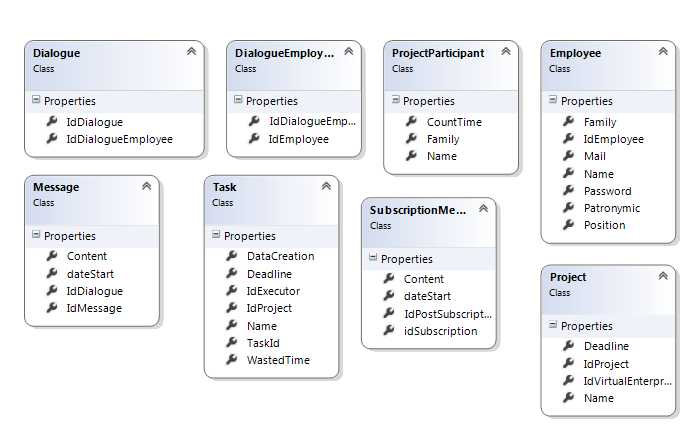
Таблица 3.4 –Формы, входящие в состав проекта

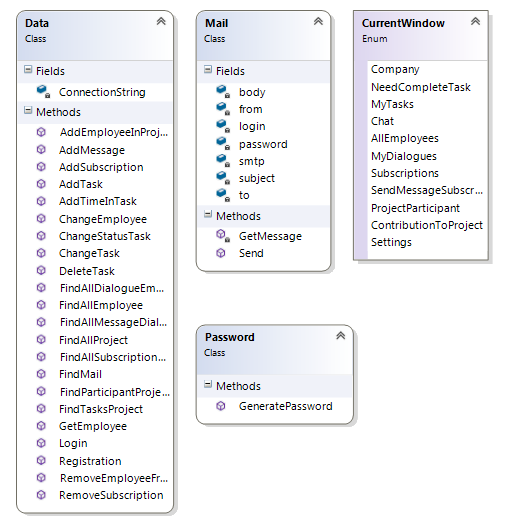
**4. ОПИСАНИЕ, ПРОГРАММНАЯ РЕАЛИЗАЦИЯ АЛГОРИТМОВ**

**4.1. Проектирование классов программы**

Диаграмма классов служит для представления статической структуры модели программного обеспечения в терминологии классов объектно-ориентированного программирования. Класс – абстрактное описание множества однородных объектов, имеющих одинаковые атрибуты, операции и отношения с объектами других классов.

На основе анализа предметной области и базы данных были спроектированы классы для хранения данных (см. рис 4.1),а также класс для обработки данных и др. (см. рис. 4.2). Далее опишем спроектированные классы.

**** Рис. 4.1. Диаграмма классов реализующих хранение данных

 Рис. 4.2. Диаграмма классов

**Классы *Dialogue, DialogueEmployee, ProjectParticipant, Employee, Message, Task, SubscriptionMessage, Project*** предназначены для хранения строк из базы данных при получении данных в программу из базы данных. Полностью соответствуют соответствующим сущностям в базе данных.

**Класс *Data***

Отвечает за запросы к базе и получении информации от базы данных.

1. FindAllEmployee-ищет всех сотрудников указанного предприятия;
2. FindParticipantProject-ищет всех участников указанного проекта;
3. AddEmployeeInProject-добавляет сотрудника в проект;
4. ChangeStatusTask-меняет статус задачи с невыполненной на выполненную и наоборот;
5. RemoveEmployeeFromProject-убирает сотрудника из проекта;
6. AddTask-добавляет задачу;
7. ChangeTask-изменяет задачу;
8. AddTimeInTask-добавляет время выполнения в задачу;
9. AddMessage-добавить новое сообщение;
10. FindAllDialogueEmployee-найти все диалоги сотрудника;
11. AddSubscription-добавить подписку;
12. RemoveSubscription-убрать подписку;
13. FindAllSubscriptionMessage-найти все сообщения от подписок;
14. GetEmployee-получить информацию о сотруднике;
15. ChangeEmployee-изменить информацию о сотруднике.

**Перечисление CurrentWindow**

Хранит все возможные переходы между окнами в нутри программы.

**4.2. Разработка алгоритмов**

После создания интерфейсной части клиентского приложения приступим к разработке алгоритмов.

Первый алгоритм реализует регистрацию пользователя в системе. Блок схема алгоритма представлена на рис. 4.3.После ввода пользователем

почты и пароля алгоритм проверяет наличие почты в базе данных и если ее там нет, то добавляет. Также уведомляет по почте об успешной регистрации.

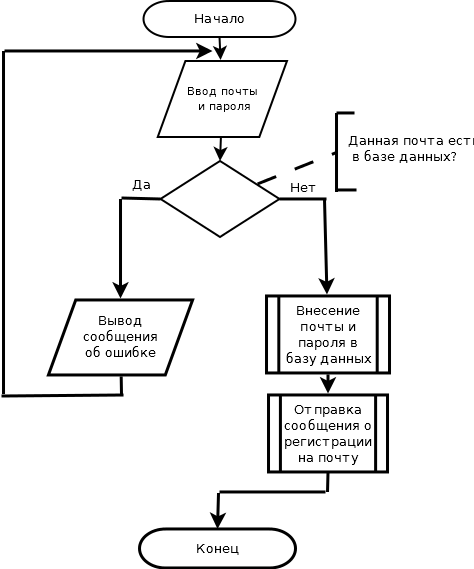


Рис. 4.3. Схема алгоритма регистрации пользователя в системе

Методы на добавление данных имеют одинаковый алгоритм. Общий алгоритм для методов на добавление данных в базу представлен на рис. 4.4.

Данный алгоритм получает на вход некоторые данные для добавления в таблицу базы данных. После происходит соединение с базой данных и происходит запрос на добавление в базу данных и вслед за ним закрытие соединения.

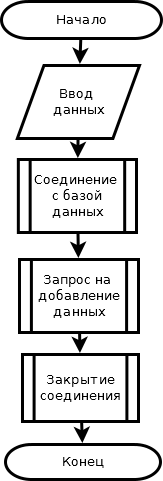


Рис. 4.4. Схема алгоритма добавления данных в базу

Методы на изменение данных также имеют один алгоритм представленный на рис 4.5.Данный алгоритм получает на вход данные о необходимых изменениях, соединяется с базой данных и делает запрос на изменение. После чего закрывает соединение.

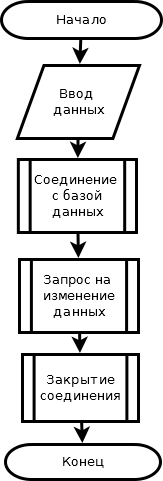
****

Рис. 4.5. Схема алгоритма изменения данных в базе

Алгоритм поиска данных в базе данных представлен на рис. 4.6.Данный алгоритм на вход получает номер строки таблицы из которой нужно получить информацию, после происходит соединение с базой данных и запрос на получение данных. После получения данных соединение закрывается и данные выводятся.

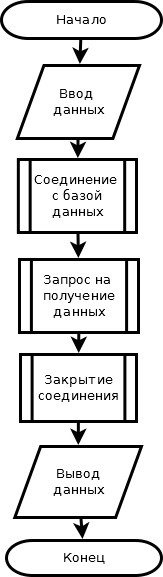
****

Рис. 4.6. Схема алгоритма поиска данных в базе

**5. ТЕХНИЧЕСКАЯ ДОКУМЕНТАЦИЯ**

**5.1. Описание применения**

Разрабатываемая программа предназначена для обеспечения функционирования виртуального предприятия.

Программа позволяет регистрировать в системе компании и набирать в них сотрудников через электронную почту.

Программа позволяет каждому сотруднику создавать новые задачи и назначать исполняющего. Программа дает возможность фиксировать в системе процесс выполнения задач.

Программа позволяет обмениваться сообщениями между сотрудниками, а также подписывать на рассылки сообщений от других сотрудников.

Программа выдает следующие отчеты:

* Вклад участника в проекты – выполненные задачи в проектах за указанный промежуток времени.
* Вклад участников в проект – показывает количество часов отработанных в проекте каждым участником за указанный промежуток времени;

Использование программы возможно не только для виртуальных предприятий, но и обычных с целью повышения производительности.

**5.2. Условия применения**

Система предназначена для работы на персональном компьютере (ПК). ПК должен удовлетворять следующим требованиям:

* на ПК должна быть установлена операционная система Windows ХР/7/8 и платформа Microsoft .NET Framework 3.5;
* ПК должен удовлетворять минимальным системным требованиям установленной операционной системы (различаются в зависимости от версии Windows);
* минимальное допустимое разрешение монитора 1024х768, видеокарта и монитор должны поддерживать этот режим работы;
* манипулятор типа «мышь»;
* клавиатура.

**5.3. Входные и выходные данные**

Входная информация представлена данными, находящимися в базе данных, а также данными полученными от пользователя.

Выходными данными являются:

* письма отправляемы на электронную почту;
* информация, хранящаяся в базе данных;
* отчеты, формируемые системой и выводимые на печать;

**5.4. Руководство программиста**

**5.4.1. Общие сведения о программе**

Представленное ПО разработано с помощью среды программирования Microsoft visual Studio 2013 c использованием СУБД Microsoft SQL Server 2008. Интерфейс и все методы обработки информации реализованы средствами Microsoft visual Studio 2013 в форме удобной для работы пользователя, а непосредственно хранимые данные содержатся в таблицах базы данных приложения Microsoft SQL Server 2008. Общая структура программы представлена модулями форм и классов, а также взаимодействием между ними.

**5.4.2. Настройка программы**

Для установки базы данных необходимо запустить скрипт создания базы данных на сервере с установленным ПО *Microsoft SQL Server 2008* и подключить к системе, используя *Microsoft SQL Manager Studio* либо средства *Transact-SQL*.

Для установки клиентской части системы необходимо скопировать папку с приложением на жесткий диск клиентского компьютера. Структура каталогов приложения:

*.\VirtualEntrerprise.exe*  – исполняемый файл программы;

При первом запуске программа автоматически подключится к серверу с базой данных. Дополнительной настройки не требуется

Для удаления программы необходимо удалить файлы программы.

**5.5. Обращение к программе**

Управление программе передается при ее запуске. Для начала работы с программой запускают исполняемый файл *VirtualEntrerprise.exe* .

**5.6. Руководство пользователя**

**5.6.1. Интерфейс программы**

Сразу после запуска программы открывается окно входа в систему (см. рис. 5.1).

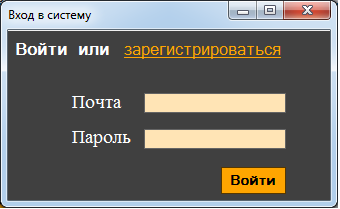


Рис. 5.1. Вход в систему

Если вы уже зарегистрированы в системе, то вам необходимо ввести вашу почту и пароль под которыми вы зарегистрированы. В противном случае необходимо нажать “зарегистрироваться”. Регистрация показана на рис. 5.2. Тут пользователю также предлагается ввести почту и пароль для регистрации в системе.

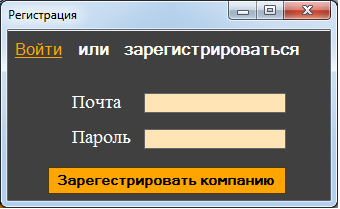


Рис. 5.2. Регистрация в системе

Стоит заметить что первая регистрация это регистрация компании, в процессе которой также создается запись под сотрудника с указанной почтой и паролем. Остальные сотрудники компании попадают в систему через приглашения. Пригласить сотрудника можно нажав “пригласить нового сотрудника” в главном меню программы (рис. 5.3). В результате нажатия появится форма для отправки приглашения (рис. 5.4).

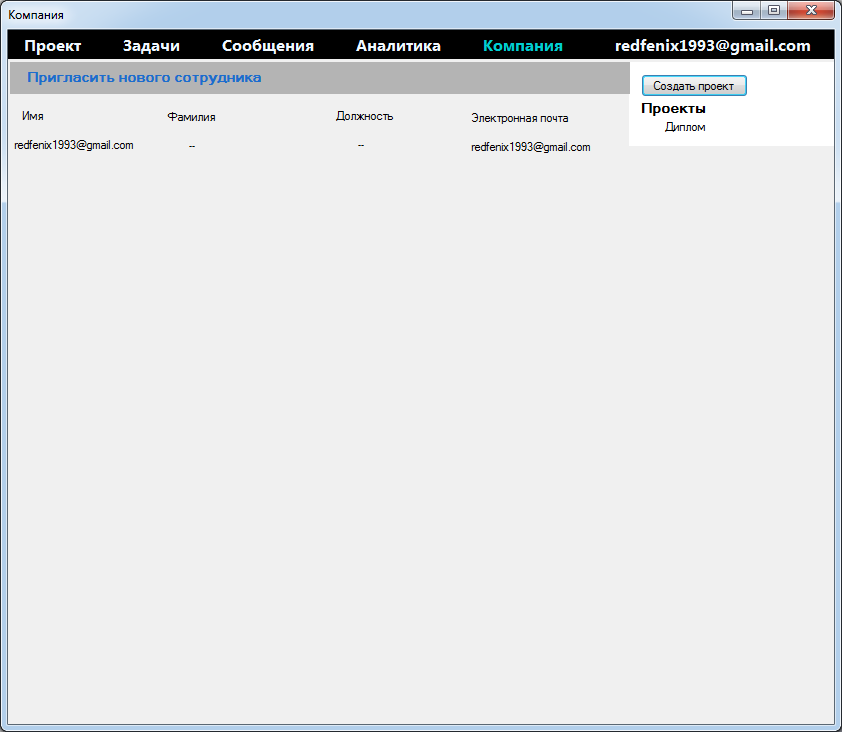


Рис. 5.3. Главное меню

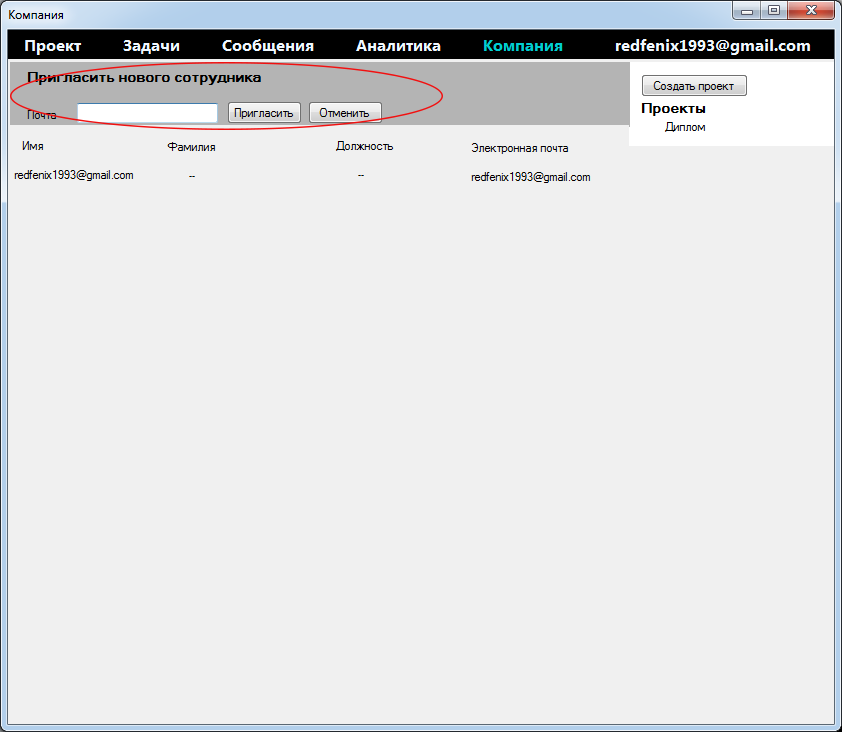


Рис. 5.4. Форма отправки приглашения

В главном меню (см. рис. 5.3) есть следующие пункты:

* Проект;
* Задачи;
* Сообщения;
* Аналитика;
* Пользователь.

Меню проект содержит список всех проектов компании (см. рис. 5.5).

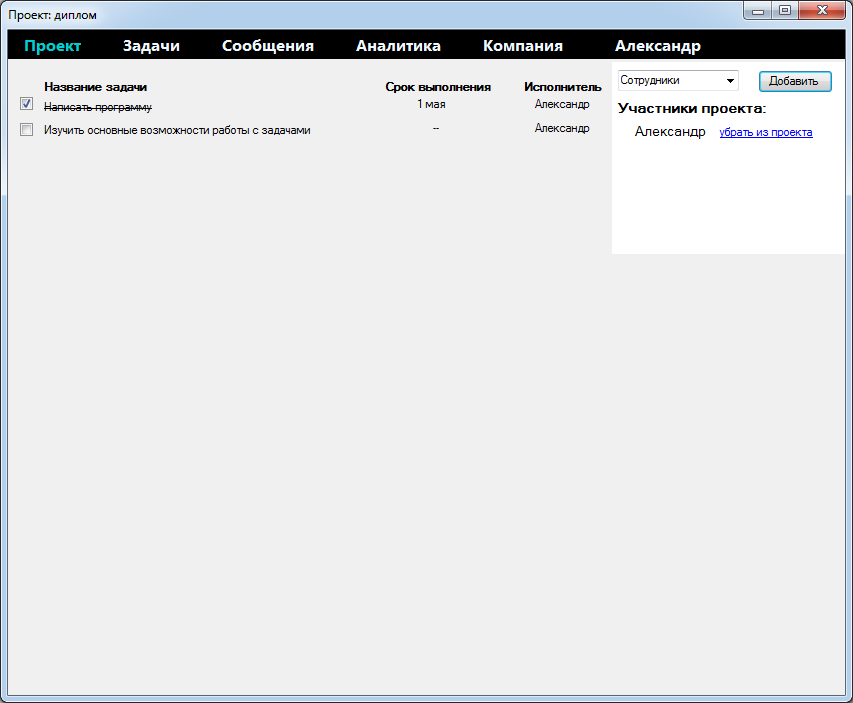


Рис. 5.5. Пункт меню “проект”

Задачи содержат два пункта:

* Необходимо выполнить;
* Мои задачи.

Пункт меню “необходимо выполнить” представлен на рис. 5.6. В этом пункте показываются задачи которые нужно выполнить (даются другими сотрудниками). В нем отображается название задачи и срок выполнения к которому надо успеть ее выполнить.Потраченное время-время которое уже потрачено на задачу, добавляется вручную нажатием на “добавить”. После выполнения задачи необходимо кликнуть на галку рядом с ней, тем самым обозначив ее завершенной.

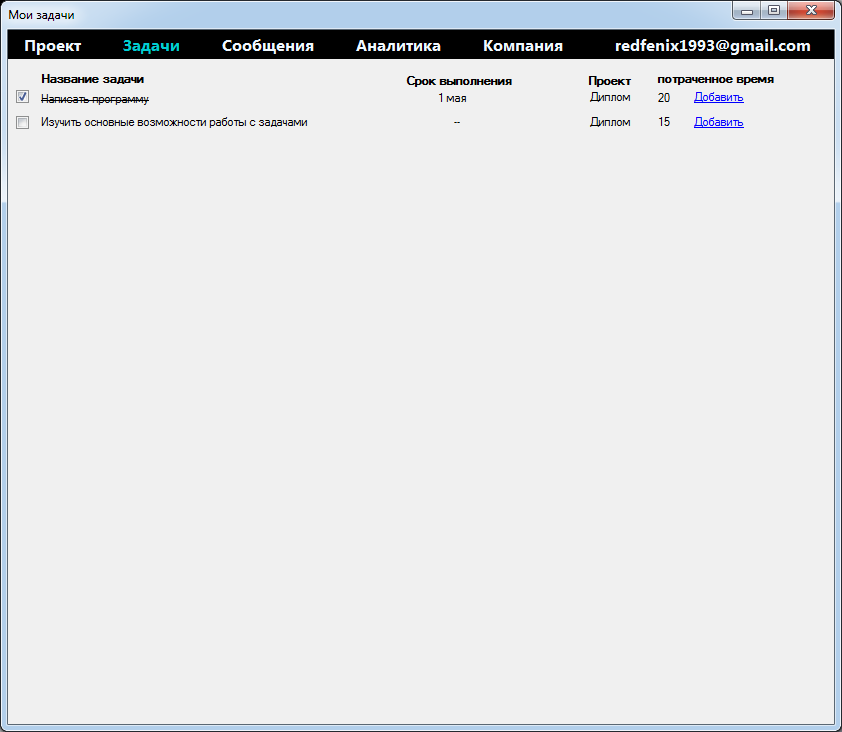


Рис. 5.6. Пункт меню “необходимо выполнить”

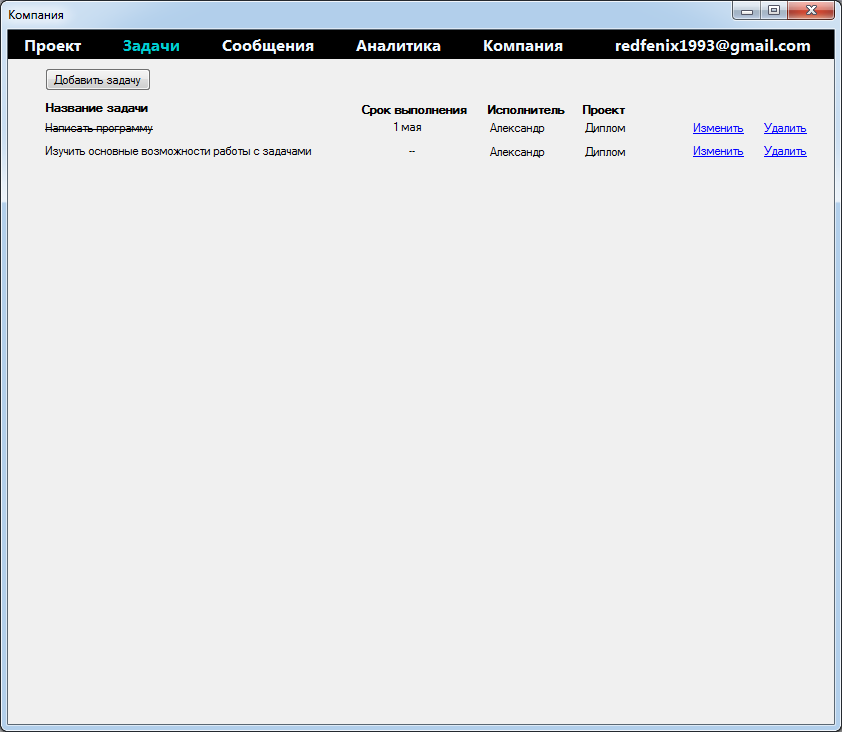
Пункт меню “мои задачи” представлен на рис. 5.7.В этом пункте мы можем добавить новые задачи и назначить исполнителя. По мере выполнения задач они будут становиться зачеркнутыми.

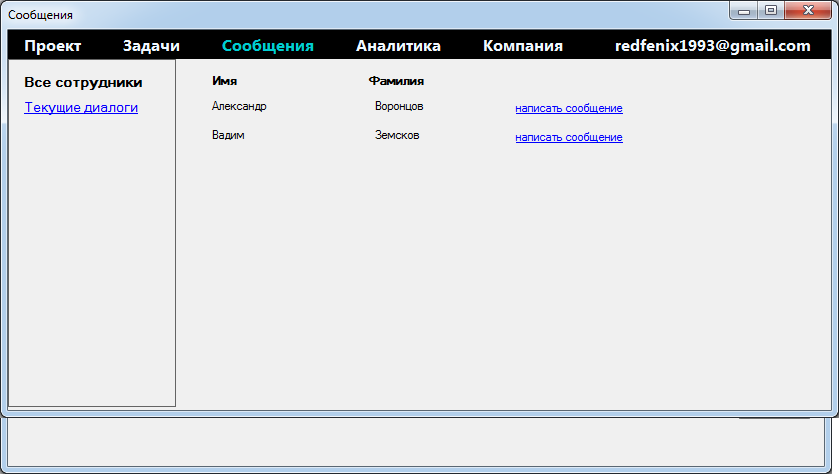
Рис. 5.7. Пункт меню “мои задачи”

Пункт меню сообщения содержит 3 подпункта:

* Личные сообщения;
* Подписки;
* Отправить сообщение подписчикам.

Меню “Личные сообщения” представлено на рис. 5.8. В этом меню есть 2 подпункта:

* Все сотрудники;
* Текущие диалоги.

**** Рис. 5.8. меню “личные сообщения”

Подпункт “Все сотрудники” содержит всех сотрудников предприятия. В текущих диалогах находятся диалоги в которые пользователь вступал с другими сотрудниками. Если нажать “написать сообщение” напротив сотрудника, то откроется окно переписки (см. рис. 5.9).В нижнем поле пользователь набирает текст и нажимает кнопку отправить.

В подпункте “Текущие диалоги” содержатся все диалоги которые начинал сотрудник.

Подпункт меню подписки представлен на рис. 5.10.После нажатия на “подписаться” вы становитесь подписанным на сотрудника и будете получать сообщения которые он рассылает по подписке.

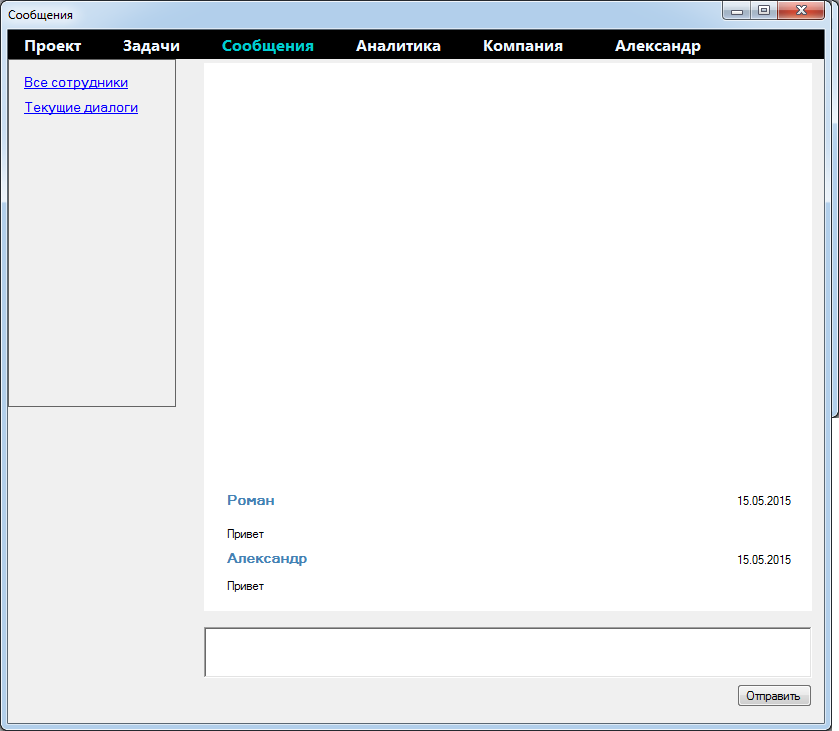


Рис. 5.9. Окно переписки

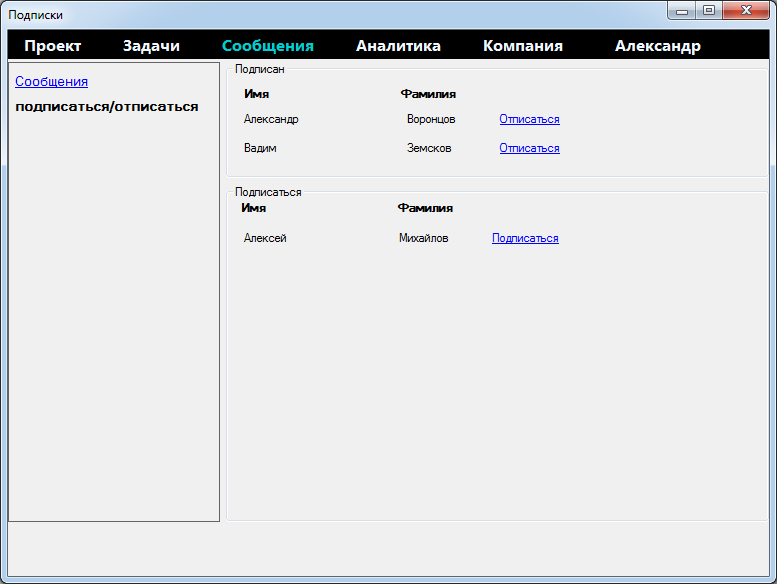


Рис. 5.10. Подписки

Пункт аналитика включает в себя 2 подпункта:

* Вклад участника в проекты
* Вклад участников в проект

В подпункте “вклад участника в проекты” (см. рис. 5.11) можно сформировать отчет по проделанной сотрудником работе. Тут пользователю предлагается выбрать временной промежуток в период которого интересна активность сотрудника. Дальше выбирается проект, который мы хотим посмотреть. В отчете показывается количество часов потраченных на проект в течении каждого дня на протяжении всего выбранного отрезка времени.

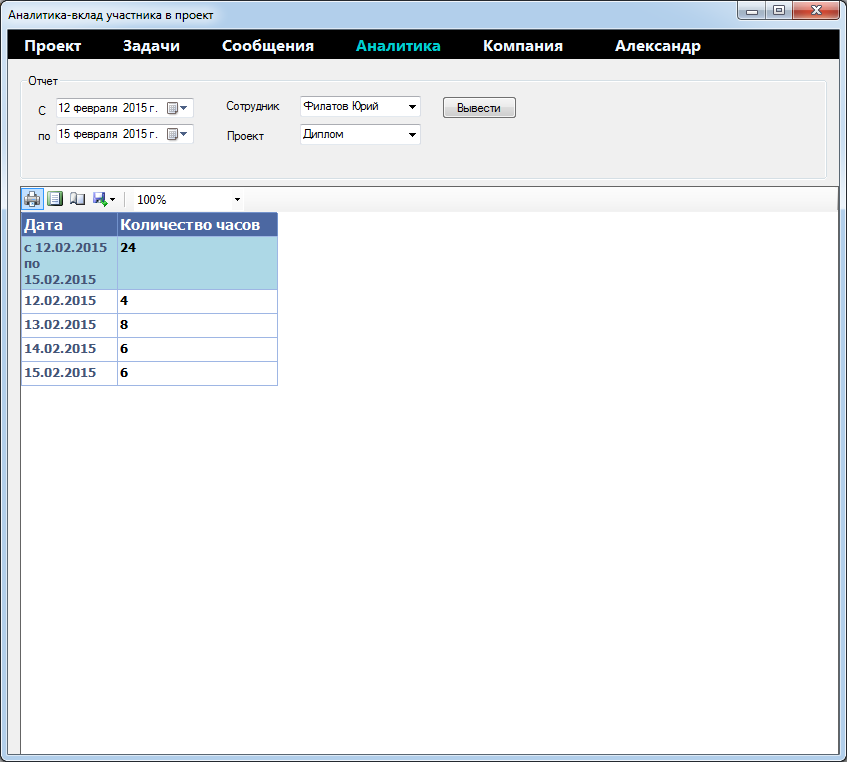


Рис. 5.11. Вклад участника в проект

В подпункте “вклад участников в проект” можно сформировать отчет по проекту (см. рис. 5.12).В нем показывается сколько часов потратил на проект каждый из участников.

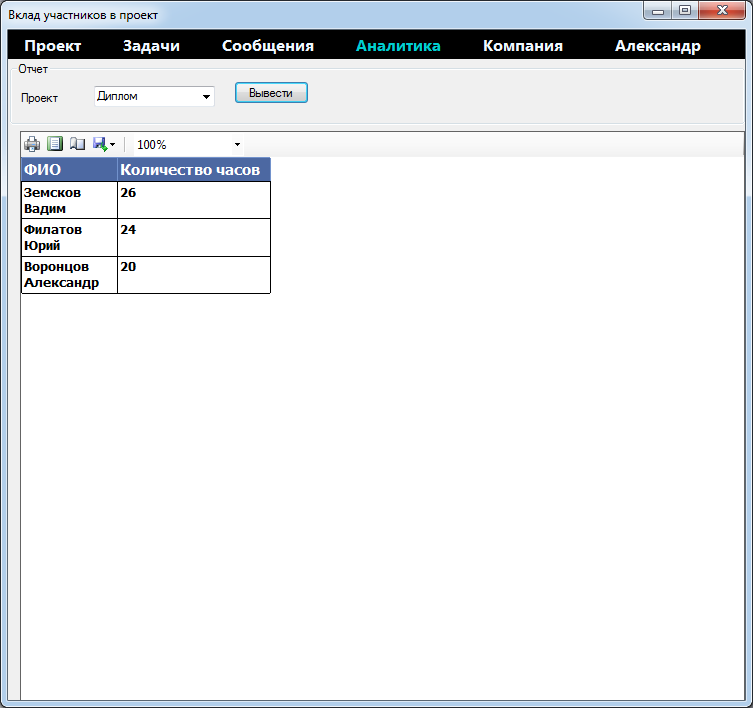


Рис. 5.12. Вклад участников в проект

**6. ТЕСТИРОВАНИЕ СИСТЕМЫ**

Процесс создания любого программного продукта всегда сопровождается проверкой корректности его выполнения – тестированием, независимо от стадии разработки. Тестирование заключается в выполнении программы с целью обнаружения ошибок.

**6.1. Общие положения**

В ходе тестирования используются данные, характерные для программы в рабочем состоянии, то есть данные для тестирования нельзя выбирать. После тестирования устанавливается соответствие ПО заданным требованиям и программным документам.

Модуль, прошедший автономное тестирование, подвергается комплексному тестированию. В процессе комплексного тестирования производится совместная проверка групп программных компонентов. В результате имеется полностью проверенная программа. На данной стадии тестирования часто обнаруживаются ошибки, пропущенные на предыдущей стадии тестирования.

Системное, или оценочное, тестирование – это завершающая стадия проверки программы, то есть испытание ПО в целом с помощью независимых тестов. Независимость тестов является существенным обстоятельством. Заказчик при приемке может настоять на проведении собственного системного тестирования. Для случая, когда сравниваются характеристики нескольких систем, такая процедура известна как сравнительное тестирование.

**6.2. Разработка методики проведения испытаний**

Разработка методики проведения испытаний предполагает проектирование системы мероприятий, обеспечивающих контроль за правильностью работы программы, выявляющей набор аномальных ситуаций, возникающих в процессе эксплуатации и подлежащих отладке.

В соответствии с поставленной перед нами задачей, тестирование программы разделим по времени на две группы тестов.

1. Промежуточные тесты (выполняются при компиляции программы или проверке корректности взаимосвязей таблиц базы в процессе их разработки). Производятся при каждой компиляции проекта, относительно проверки разрабатываемых процедур обработки событий, методов и других изменений продукта.
2. Итоговые тесты (выполняются при комплексной отладке проекта на завершающем этапе работы над ним). Такие тесты необходимо провести для всех диалоговых форм проекта, с целью выявления скрытых при компиляции дефектов целостности данных и корректности выполнения реализованных методов.

Для каждой из вышеперечисленных групп согласно разработанной методике проведения испытаний следует выполнить два теста.

1. Проверка входных данных.
2. Проверка выходных данных.

При тестировании входных данных необходимо использовать как корректные, так и некорректные наборы данных, чтобы убедиться в правильности их обработки программой.

Все тесты рекомендуется выполнять в полном объёме для второй группы. Для первой группы рекомендуется проводить их для всех данных, связанных с текущим изменением программы, во избежание появления скрытых ошибок, выявляющихся в процессе длительной эксплуатации. Такой подход к тестированию особенно актуален в условиях многоуровневой структуры вложенных форм программы и сложной системы фильтрации хранимых данных.

**6.3. Результаты проведения испытаний**

Тестирование проекта проводится в реальной вычислительной сети предприятия, что обеспечивает достоверность входных данных.

Рассмотрим примеры результатов проведения испытания для каждого из тестов второй группы.

**6.3.1. Условия тестирования**

Тестирование проводилось под управлением операционной системы Windows 7 (Service pack 1) (русифицированная).

Характеристики аппаратного обеспечения:

1. Процессор, тактовая частота которого составляет 3,0 ГГц.
2. Оперативная память 512 Мбайт.
3. Свободная емкость жесткого диска, на которой размещалась тестируемая программа 3 Мбайт.
4. Разрешение экрана 1280×1024 пикселей.
5. Мышь с программным обеспечением под Windows.

**6.3.2. Тестирование входных данных**

При тестировании программы в базу данных вносилась информация о сотрудниках, задачах, проектах. В результате в базу данных была занесена следующая информация:

* количество сотрудников - 4;
* количество проектов – 4;
* общее количество задач – 20.

Вся введенная информация корректно сохранилась в базе данных.

В целом программа функционировала без ошибок, выдавала корректные данные.

**6.3.3. Тестирование выходных данных**

Выходной информацией программы являются данные, отображаемые на экране и печатные формы отчетов.

При тестировании данных, отображаемых на экране, контролировалось соответствие информации на экране ранее введенным данным.

Было проверена выдача сообщений об ошибках в случае введения неккоректных данных. Допустим в случае если почта уже зарегистрирована в системе будет выведено сообщение об ошибке (см. рис. 6.1)

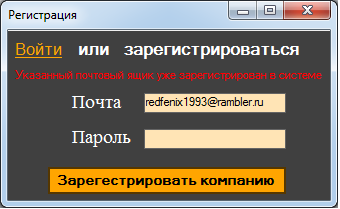


Рис. 6.1. Сообщение об ошибке при регистрации

Также было проверено что система присылает оповещения на почту сотруднику когда его приглашают в систему (см. рис. 6.2)

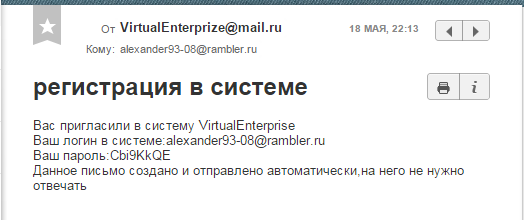


Рис. 6.2. Регистрация в системе

Аналогичным образом контролировались и прочие формы программы.

Для тестирования корректности вывода отчетов были сформированы несколько отчетов. Проверка напечатанных отчетов показала правильность их формирования, то есть информация отчетов соответствует информации базы данных.

В ходе тестирования было установлено, что разработанное программное обеспечение соответствует техническому заданию на проектирование и полностью работоспособно. Перейдем к оценке экономической эффективности от разработанного ПО, для этого рассчитаем смету затрат и дадим оценку стоимости проекта.

**7. БЕЗОПАСНОСТЬ И ЭКОЛОГИЧНОСТЬ ПРОЕКТА**

Основной целью данного дипломного проекта является разработка программного обеспечения для обеспечения работы виртуального предприятия. Разработка системы и её использование связаны с работой на ПЭВМ, следовательно разработчики и пользователи системы могут подвергаться воздействию вредных и опасных факторов, сопутствующих работе на ПЭВМ.

Частью разработки данного дипломного проекта является анализ и оценка труда на ПЭВМ, воздействие вредных и опасных факторов на оператора, а также рассмотрение мер, которые уменьшают негативное воздействие этих факторов окружающей среды, предупреждают несчастные случаи, создают высокопроизводительные, здоровые и безопасные условия труда в вычислительных центрах

Операторы ПЭВМ сталкиваются с воздействием таких производственных факторов, как недостаточное освещение рабочей зоны, влияние электромагнитного и ионизирующего излучения, повышенный уровень шума, повышенная температура окружающей среды и так далее. Большое влияние на человека оказывают также напряженность трудового процесса.

**7.1. Характеристика помещения и описание рабочего места**

Рабочее место программиста расположено на 5 этаже девятиэтажного здания и имеет площадь 12 м2(габариты: ширина ― 3м, длина ― 4 м, высота потолка― 2.5м)

Окраска стен светло-зеленого цвета, потолка – белого, пол – светло-коричневый паркет. Цветовое оформление выполнено с учетом рекомендаций СН-181-70 «Указания по проектированию цветовой отделки интерьеров и производственных зданий промышленных»: цвета стен, потолка, пола гармонируют между собой. Помещение оборудовано устройствами общего освещения, имеется 1 окно и 1 дверь.

Освещение – естественное и искусственное. Искусственное освещение в помещении осуществляется с помощью светильника над рабочим местом мощностью 40Вт подвешенным на высоте 1,6 м от пола. Вторым источником искусственного освещения является люстра в центре комнаты с 3 лампами мощностью 60Вт.

Количество рабочих мест-1.Площадь помещения 12 м2,что соответствует СанПиН 2.2.2/2.4.1340-03 «Гигиенические требования к персональным электронно-вычислительным машинам и организации работы», которые устанавливают на одно рабочее место пользователей ПЭВМ с ВДТ(видео-дисплейный терминал) на базе плоских дискретных экранов (жидкокристаллические, плазменные) площадь не менее 4,5 м2.

Корпус монитора и ПЭВМ, клавиатура и другие блоки имеют матовую поверхность одного цвета и не имеют блестящих деталей, способных создавать блики, что соответствует требованиям, содержащимся в СанПиН 2.2.2/2.4.1340-03 «Гигиенические требования к персональным электронно-вычислительным машинам и организации работы», к покрытиям и материалам, из которых изготавливаются корпуса ПК и дисплеев.

Рабочее место состоит из:

* стула;
* стола;
* ЖК-монитор Samsung 22''
* Системный блок (AMD Fx-8300,8 Gb DDR-4,500Gb HDD);
* Клавиатура;
* Манипулятор “мышь”.

В помещении имеются принтер и сканер фирмы canon.

Рабочее места оборудовано роутером обеспечивающим выход в интернет на скорости 40 Мб/c как с ПЭВМ так и с мобильных устройств с помощью WiFi.

Для электропитания используется трёхфазная сеть переменного тока частотой 50 Гц и напряжением 220 В и глухозаземленной нейтралью источника питания.

Параметры микроклимата в помещении следующие:

* температура воздуха составляет 21-24 градуса при нормативных значениях в теплое время года  23-25 градусов, в холодное  22-24 градуса;
* относительная влажность воздуха  40-50 % (норматив  40-60 %);
* скорость движения воздуха  0,1 м/с.

Пользователи-операторы ПЭВМ сталкиваются с воздействием таких опасных и вредных производственных факторов, как повышенный уровень шума, неблагоприятные температурные условия внешней среды, поражение электрическим током. Рассмотрим эти факторы более подробно.

**7.1.1. Шум**

Основным документом, регламентирующим соблюдение правил безопасности при воздействии шума, является ГОСТ 12.1.003-83 «Шум. Общие требования безопасности». Классификация основных средств защиты от шума приведена в ГОСТ 12.1.029-80 «Средства и методы защиты от шума. Классификация». При выполнении основной работы на ПК уровень шума на рабочем месте не должен превышать 50 дБА, согласно СН 2.2.4/2.1-8.562-96 «Шум на рабочих местах, в помещениях жилых, общественных зданий и на территории жилой застройки» .

Длительное воздействие интенсивного шума (выше 80 дБ) на органы слуха человека приводит к частичной или полной потере слуха. Степень потери слуха зависит от уровня звука и его продолжительности и от индивидуальной чувствительности человека. Шумящее оборудование, уровни шума которого превышают нормированные, должно находится вне помещения с ПЭВМ. Для снижения шума, создаваемого на рабочих местах внутренними источниками, а также шума, проникающего извне, следует: ослабить шум самих источников, в частности, предусмотреть применение в их конструкциях акустических экранов, звукоизолирующих кожухов и т.д.; снизить эффект суммарного воздействия на рабочие места отраженных звуковых волн за счет звукопоглощения энергии прямых звуковых волн поверхностями ограждающих конструкций; применять рациональное расположение оборудования; использовать архитектурно-планировочные и технологические решения, направленные на изоляцию источников шума.

Источниками шума в помещении являются системные блоки компьютеров (шум вентиляторов) и принтер. По результатам замеров шума уровень шума составил 35 дБА, что не превышает норму, следовательно, дополнительных мер не требуется.

**7.1.2. Микроклимат**

Большое значение имеет создание в рабочей зоне благоприятного микроклимата, который определяется температурой, влажностью, скоростью движения воздуха, атмосферным давлением, интенсивностью излучения нагретых поверхностей.

Неблагоприятные микроклиматические условия приводят к ухудшению самочувствия работника, ослаблению внимания, быстрой утомляемости, и при продолжительном воздействии могут вызвать различные заболевания.

Микроклиматические условия на рабочем месте инженера-программиста нормируются согласно СанПиН 2.2.4.548-96 «Гигиенические требования к микроклимату производственных помещений». Оптимальные микроклиматические условия установлены по критериям оптимального теплового и функционального состояния человека. Они обеспечивают общее и локальное ощущение теплового комфорта в течение 8-часовой рабочей смены при минимальном напряжении механизмов терморегуляции, не вызывают отклонений в состоянии здоровья, создают предпосылки для высокого уровня работоспособности и являются предпочтительными на рабочих местах. Оптимальные параметры микроклимата в холодный и теплый периоды года на рабочем месте инженера-программиста и оператора ПЭВМ должны соответствовать величинам, приведенным в таблице 7.1.

Таблица 7.1. Оптимальные параметры микроклимата на рабочем месте

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Период года** | **Категория работ по уровню энергозатрат, Вт** | **Температура воздуха, °С** | **Температура поверхностей, °С** | **Относи­тельная влажность**  **воздуха,**% | **Скорость движения воздуха,**  **м/с** |
| Холодный | 1а (до 139) | 22-24 | 21-25 | 60-40 | 0,1 |
| Теплый | 1а (до 139) | 23-25 | 22-26 | 60-40 | 0,1 |

Нормализация воздуха в рассматриваемом помещении достигается с помощью двух устройств кондиционирования воздуха, а также подачей чистого воздуха с помощью вентиляции. В соответствие с протоколом санитарно-гигиенической оценки условий труда, микроклимат в помещении соответствует норме.

**7.1.3. Воздействие электрического тока**

Электрические установки, к которым относится почти все оборудование ПЭВМ, представляют для человека большую потенциальную опасность, так как, проходя через тело человека, электрический ток вызывает термическое, механическое и биологическое действие. Причиной поражения электрическим током может быть несоответствие технических средств ВТ стандартам безопасности. Поэтому применяемые меры и средства электробезопасности должны соответствовать требованиям ГОСТ 12.1.038-82\* «ПДУ напряжений прикосновения и токов». Напряжения прикосновения и токи, протекающие через тело человека не должны превышать значений, указанных в таблице 6.2.

Таблица 7.2. Предельно допустимые значения напряжений прикосновения и токов

|  |  |  |
| --- | --- | --- |
| Род тока | Uпд, В | Iпд, мА |
| не более | |
| Переменный, 50 Гц | 2,0 | 0,3 |

Для защиты человека от поражения электрическим током необходимо применять технические меры, которые, для повышения уровня безопасности, целесообразно использовать в комплексе ГОСТ Р 12.1.019-2009 «Электробезопасность. Общие требования».

**7.2. Оптимизация зрительных условий труда на рабочем месте**

**7.2.1. Основные требования к освещению с учётом труда**

Свет является естественным фактором жизнедеятельности человека, играющим важную роль в сохранении здоровья и высокой работоспособности.

Действие света на организм человека чрезвычайно многообразно. Уровень освещённости оказывает действие на состояние психических функций и физиологические процессы в организме. Так, хорошее освещение действует тонизирующе, стимулирует активность деятельности человека; улучшает протекание основных нервных процессов. Рациональное освещение предупреждает развитие утомления, способствует повышению производительности труда и играет важную роль в снижении производственного травматизма. Установлено, что плохое освещение является причиной примерно 5% несчастных случаев на предприятиях.

Особое значение освещение имеет для профессий, в которых зрительная система играет главную роль в трудовой деятельности, испытывает большие нагрузки и зачастую является источником ошибок.

Состояние функции зрения, работоспособность зрительной системы человека определяются такими показателями, как острота зрения, скорость различения и устойчивость ясного видения, контрастная и цветовая чувствительность.

Основная пространственная характеристика глаза – острота зрения, определяемая величиной, обратной наименьшему расстоянию между двумя точками, при котором они видятся раздельно. Острота зрения зависит от освещённости, контраста между объектом и фоном, расстояния до наблюдаемого объекта.

Контрастом К объекта наблюдения и фона называется различие между их яркостями

К=(Во-Вф)/Вф,

где Во и Вф соответственно яркости объекта и фона, кд/м2 оптимальная величина контраста считается равной 0,6-0,9.

Скорость различения относится к временным характеристикам зрительного анализатора. Она представляет собой способность глаза быстро различать объекты наблюдения и в значительной степени определяет безопасность работы. Скорость различения мала при низкой освещённости, наличии слепящих поверхностей в поле зрения, малом контрасте объекта и фона.

При недостаточной освещённости сокращается время, в течении которого глаз человека сохраняет способность ясно различать рассматриваемый объект, - время ясного видения. На устойчивость ясного видения оказывает влияние напряжённость зрительной работы, уровень освещённости, пульсация светового потока. Как показывают физиологические исследования, время ясного видения при работе в течение 3 ч сокращается при освещённости 50 лк на 72% от исходной величины, при освещённости 75 лк – на 55%, при 100 лк – на 26%, при 200 лк – на 15%.

Снижение видимости при появлении в поле зрения блеских источников света называется ослеплённостью.

Важной характеристикой зрительного восприятия является критическая частота мельканий – минимальная частота, при которой прерывистое изображение воспринимается как непрерывное. Значение критической частоты (fкр) зависит от яркости объекта различения и его угловых размеров. Это свойство зрительного восприятия необходимо учитывать при работе на ЭВМ.

Анализ воздействия света на организм человека и основных свойств зрительного восприятия позволяет сформулировать основные требования к производственному освещению, которые заключаются в обеспечении: достаточной освещённости рабочих поверхностей, равномерности распределения яркости, отсутствия глубоких и резких теней, постоянства освещённости во времени.

**7.2.2. Обоснование организации освещения и нормативного уровня освещённости рабочего места**

Освещение рабочих мест может быть естественным и искусственным.

Естественное освещение может осуществляться через окна или световые проёмы в наружных стенах (боковое освещение), через застеклённые световые фонари и перекрытие (верхнее) или через фонари и окна одновременно (комбинированное).

Естественное освещение резко изменяется в течение дня, времени года и существенно зависит от атмосферных условий. От этих недостатков свободно искусственное освещение – освещение помещений искусственным светом с помощью электрических ламп. На некоторых предприятиях применяется совмещённое освещение, когда недостаточное естественное освещение дополняется искусственным. Искусственное освещение подразделяется на рабочее, дежурное, аварийное, эвакуационное и охранное.

Рабочее освещение предназначено для создания необходимых условий работы и нормальной эксплуатации зданий или территории. Дежурное освещение включается во вне рабочее время.

Аварийное освещение применяется в тех случаях, когда отключение рабочего освещения может привести к взрыву, пожару, длительному нарушению технологического процесса, нарушению работы таких объектов, как узлы радиопередачи и связи, электростанции и т.п. При аварийном освещении часть светильников общего освещения питаются током от автономного источника и в случае отключения основной сети продолжают работать. Согласно СНиП II–4–79 освещённость в этом случае должна составлять не менее 5% от величины , предусматриваемой нормами рабочего освещения для данного вида работ, но не менее 5 лк при газоразрядных лампах и 2 лк – при лампах накаливания.

Эвакуационное освещение включается при аварийной обстановке для эвакуации людей. Оно устанавливается в помещениях с числом работающих свыше 50 и на открытой территории. Освещённость в помещениях составляет 0,5 лк, а на открытой территории – 0,2 лк.

Охранное освещение размещается вдоль границ территорий, охраняемых в ночное время.

Искусственное освещение делится на общее, местное и комбинированное.

Общим называется освещение, при котором осветительные устройства размещаются в верхней зоне помещения и равномерно освещают всю площадь, занятую рабочими местами и оборудованием.

Если светильники концентрируют световой поток непосредственно на рабочие места, то такое освещение называется местным.

Комбинированным называется освещение, при котором наряду с общим искусственным освещением используются светильники местного освещения для создания на рабочих местах освещённости более высоких уровней.

В современных осветительных установках, предназначенных для освещения производственных помещений, в качестве источников света применяются лампы накаливания, гологенные и газоразрядные.

К основным характеристикам источников света относятся: удельная световая отдача и средний срок службы, а также мощность ламп, напряжение сети и излучаемый лампой световой поток.

При выборе источников света необходимо обращать внимание на спектральный состав света, так как он способствует не только цветоразличению в процессе выполнения трудовой задачи, но и оказывает существенное влияние на психофизиологическое состояние человека и ощущение им светового комфорта.

Желательно, чтобы спектр искусственного освещения максимально приближался к спектру естественного света.

**7.2.3. Средства и способы обеспечения требуемой освещённости и равномерности светораспределения**

Выбор параметров производственного освещения основывается на учёте требований, предъявляемых конкретным производственным процессом, в соответствии с действующими нормами и правилами.

СНиП 23-05-95 “Естественное и искусственное освещение” устанавливает минимальные уровни освещённости рабочих поверхностей в зависимости от точности зрительной работы, контраста объекта и фона, яркости фона, системы освещения и типа используемых ламп.

Точность зрительной работы характеризуется размером объекта различения. Объект различения – это элемент рассматриваемого объекта минимального размера, который нужно узнавать и различать (элемент буквы или толщина её начертания, размер отдельных деталей или расстояние между ними при пайке и монтаже и т.п.).

Поверхность, прилегающая непосредственно к объекту различения, на которой он рассматривается, называется фоном. Фон считается светлым при коэффициенте отражения поверхности более 0,4, средним – при коэффициенте отражения от 0,2 до 0,4 и тёмным – при коэффициенте отражения менее 0,2.

Гигиенические нормы для естественного освещения устанавливают требуемую величину коэффициента естественного освещения (КЕО) в зависимости от точности работ, вида освещения и географического расположения производства.

Для определения соответствия естественной освещённости в производственном помещении требуемым нормам измеряют освещённость: при верхнем и комбинированном освещении – в различных точках помещения с последующим усреднением; при боковом – на наименее освещённых рабочих местах. Одновременно измеряют наружную освещённость и определённый расчётным путём КЕО сравнивают с нормативным.

Для искусственного освещения нормируемым параметром является освещённость. В зависимости от контраста объекта с фоном и яркости фона каждый из восьми разрядов точности зрительных работ подразделяется на четыре подразряда, для каждого из которых нормируется освещённость.

Необходимый уровень освещённости тем выше, чем темнее фон, меньше объект различения и контраст объекта с фоном.

Нормы регламентируют не только количественные, но и качественные показатели освещения: показатель ослеплённости - для ограничения слепящего действия, создаваемого самосветящимися или пропускающими свет поверхностями; коэффициент пульсации (для газоразрядных ламп) – для уменьшения стробоскопического эффекта.

Для общественных и вспомогательных зданий, а также жилых помещений нормируется показатель дискомфорта с целью ограничения неравномерного распределения яркостей в поле зрения.

**7.2.4. Расчёт освещения рабочего места**

В зависимости от сложности и характера зрительных задач искусственное освещение может быть организованно в виде системы общего или комбинированного освещения. Общее освещение создается равномерно распределенными на потолке светильниками и используется, когда необходимо обеспечить одинаковую освещенность на всей рабочей площади помещения (комнаты управления, аудитории, лаборатории, коридоры и т.п.).

Проектирование системы общего искусственного освещения представляет собой последовательное решение следующих задач.

- выбор типа источников света (ламп);

- выбор типа светильников;

- размещение светильников в плане помещения и определение их количества;

- расчет светового потока ламп светильников;

- выбор стандартной лампы.

Исходными данными для расчета являются:

- гигиеничекая норма освещения согласно СНиП 23-05-95 “Естественное и искусственное освещение” Еmin(лк). Еmin = 300 лк;

- габаритные размеры производственного помещения A x B x H (м).

A = 4, B = 3,H = 2,5;

коэффициенты отражения рабочих поверхностей rр = 30% , поверхностей стен rс= 50%, поверхностей потолка rп = 70%.

Светильники выбирают с учетом характеристик рабочей среды в помещении. Так как высота помещения меньше 6 метров нам подходят люминесцентные светильники. Для освещения помещения выберем светильник БЛ-1х40. Для люминесцентных ламп  λ= 1,5.

Расстояние между центрами светильников lс= λ∙hср.

Светильники будем крепить непосредственно на потолке изз небольшой высоты помещения.

hcp=2,5-0-0,725=1,6м

Lc=1,5\*1,6\*0,3=0,72м

Индекс помещения определяется по формуле:



где a и b - длина и ширина помещения, м (a = 4, b = 3);

h - высота рабочей поверхности над полом, м (hp =0.725)



Для индекса помещения i = 2.36 и коэффициентов отражающих поверхностей : рабочих поверхностей rр = 30% , поверхностей стен rс= 50%, поверхностей потолка rп = 70% коэффициент использования светового потока согласно таблице .

Число светильников определим по следующей формуле: Nсв=12/(1,775\*1,775)=3,8≈4

Световой поток одного светильника определяется методом коэффициента использования светового потока по формуле:



S - площадь помещения, м2 ( S = 4 \* 3 = 12 );

К - коэффициент запаса, зависящий от запыленности воздуха в помещении (для воздушной среды, содержащей не более 1 мг/м3 К = 1.5 );

Z - коэффициент неравномерности освещения ( для люминесцентных ламп Z = 1.1);



Наиболее близка к такому значению лампа ЛТБ40, дающая световой поток равный 3000 лм.

Проверим фактический уровень освещенности по формуле: 



отличается от  менее чем на 10%,следовательно все посчитано верно.

Определим электрическую мощность системы освещения по формуле:





Схема размещения светильников представлено на рис. 7.1.

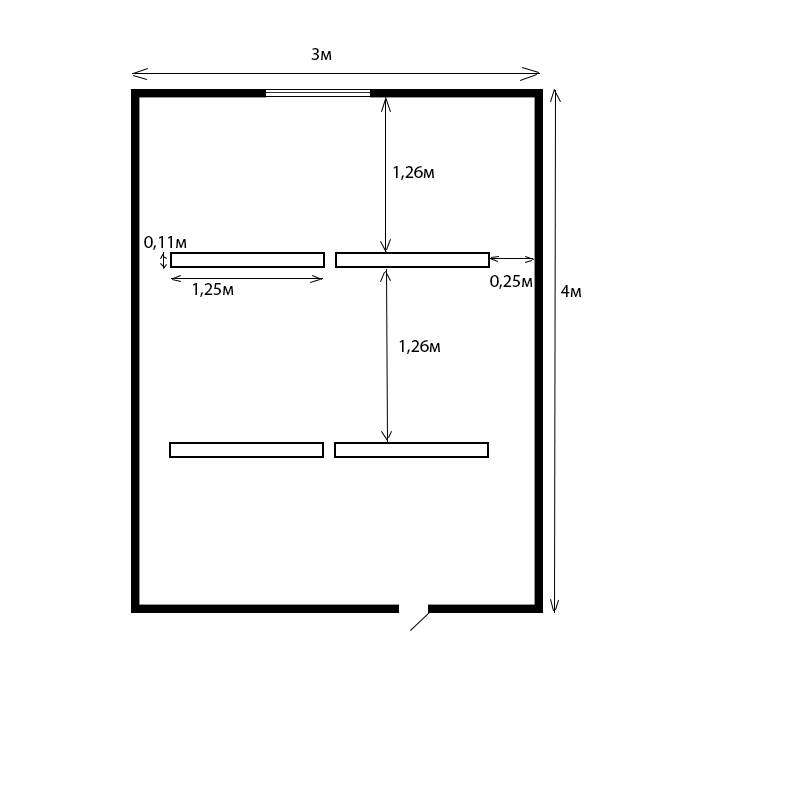


Рисунок 7.1. Схема размещения светильников в помещении.

**7.3. Обеспечение пожарной безопасности**

Пожарная безопасность – состояние объекта, при котором исключается возможность пожара, а в случае его возникновения предотвращается воздействие на людей опасных факторов пожара и обеспечивается защита материальных ценностей.

Источниками зажигания в помещении при работе на ПЭВМ могут быть электронные схемы от ПЭВМ, приборы, применяемые для технического обслуживания, устройства электропитания, кондиционирования воздуха, где в результате различных нарушений образуются перегретые элементы, электрические искры и дуги, способные вызвать загорания горючих материалов.

В анализируемом помещении возгорание может произойти по следующим причинам:

* неисправное электрооборудование, неисправности в электропроводке, электрических розетках и выключателях;
* неисправные электроприборы;
* перегрузка по току;
* короткое замыкание в электропроводке;
* несоблюдение требований пожарной безопасности, курение в помещении.

Горючими и трудногорючими веществами и материалами согласно НПБ 105-03 «Нормы пожарной безопасности «Определение категорий помещений, зданий и наружных установок по взрывопожарной и пожарной опасности» в помещении являются:

* полы (линолиум);
* столы, стулья (ДВП, ДСП, дерево, полимерные материалы);
* окна (деревянные рамы);
* изоляция электропроводки;
* компьютер и периферийное оборудование.

Определение пожароопасной категории помещения осуществляется путем сравнения максимального значения удельной временной пожарной нагрузки на любом из участков с величиной удельной пожарной нагрузки, приведенной в таблице 7.1.

Таблица 7.1. Удельная пожарная нагрузка

для категорий помещения В1-В4

|  |  |
| --- | --- |
| Категория | Удельная пожарная нагрузка g на участке, МДж×м-2 |
| В1 | Более 2200 |

Окончание таблицы 7.1

|  |  |
| --- | --- |
| В2 | 1401 – 2200 |
| В3 | 181 – 1400 |
| В4 | 1 – 180 |

Пожарная нагрузка Q, МДж, определяется по формуле (7.1)

, (7.1)

где G1 — количество пожарной нагрузки, кг,

Qph — низшая теплота сгорания материала пожарной нагрузки, МДж×кг-1.

Удельная пожарная нагрузка g, МДж×м-2, определяется из соотношения (7.2):

, (7.2)

где S — площадь размещения пожарной нагрузки, м2 (но не менее 10 м2).

Рассчитаем удельную пожарную нагрузку для помещения. Комната площадью 12 м2 в которой имеется:

1. 1 деревянный стол;
2. 2 деревянных стула;
3. 1 компьютер;
4. стены обклеены бумажными обоями;
5. на полу линолиум;
6. деревянная дверь;
7. деревянные плинтуса;
8. окно с деревянной рамой;
9. деревянная кровать;
10. железные полки с тетрадями и книгами;
11. шторы.

Затем рассчитываем удельную пожарную нагрузку и определяем категорию помещения по пожаро- и взрывоопасности.

Пожарная нагрузка будет равна:

Q=13,8\*30+13,8\*5+41,87\*3+13,4\*12+20,3\*20+13,8\*10+13,8\*3+13,8\*10+13,8\*15+13,4\*3+15,7\*1,5=1763,56 МДж/кг

Удельная пожарная нагрузка составит:

g=1763,56/12=146,96МДж\*м-2

Это значение соответствует категории В4.

Основными документами, регламентирующими соблюдение правил пожарной безопасности на рабочем месте, являются ГОСТ 12.1.004-91\* «ССБТ. Пожарная безопасность. Общие требования», а также документ «Правила пожарной безопасности в РФ» ППБ 01-03. Пределы огнестойкости строительных конструкций регламентируются СНиП 21.01.97\* «Пожарная безопасность зданий и сооружений».

Эксплуатационными мероприятиями являются своевременные профилактические осмотры, ремонты и испытания технологического оборудования и прочей техники.

К первичным средствам пожаротушения (ПСПТ) относятся устройства, инструменты и материалы, предназначенные для локализации или тушения пожара на начальной стадии его развития (огнетушители, песок, вода, несгораемые ткани, вёдра, лопаты и другие подручные средства). Наиболее широкое применение находят ***огнетушители*** - переносные и передвижные устройства для тушения очага пожара за счёт выпуска запасённого огнетушащего вещества, расположенные в известном для всех работников месте и легко доступном в случае возникновения пожара(в процессе эксплуатации необходимо выполнить требования НПБ 166-97\* «Пожарная техника. Огнетушители. Требования к эксплуатации»).

Существуют следующие виды огнетушителей:

1. углекислотные;
2. порошковые.

Для данного помещения лучше всего подходит углекислотный огнетушитель,в количестве 1 шт. Предназначены для тушения возгораний различных горючих веществ, горение которых не может происходить без доступа воздуха, на промышленных предприятиях, на транспортных средствах (железнодорожном, городском, морском транспорте), возгораний электроустановок, находящихся под напряжением не более 1000В.

Одним из основных факторов обеспечения пожарной безопасности на предприятиях является применение автоматических средств обнаружения пожаров, которые позволяют оповестить дежурный персонал о пожаре и месте его возникновения.

Пожарная сигнализация должна:

* быстро выявить место возникновения пожара;
* передать сигнал о возгорании на приёмно-контрольную станцию;
* оставаться невосприимчивой к влиянию внешних факторов, отличающихся от факторов пожара;
* передавать извещение о неисправности в самой системе оповещения.

Средствами пожарной сигнализации и оповещения оборудуются производственные здания и помещения категорий А, Б и В, помещения с вычислительной техникой и дорогостоящей аппаратурой, а также объекты государственной важности.

Любая система пожарной сигнализации состоит из пожарных извещателей и преобразователей, преобразующих факторы появления возгорания (тепло, свет, дым) в электрический сигнал, передающийся по линиям связи на приёмно-контрольную станцию, которая включает световую и звуковую сигнализацию, а также может включить автоматическую установку пожаротушения и дымоудаления.

Ручные пожарные извещатели предназначены для передачи информации по шлейфу сигнализации на приёмно-контрольную станцию. Извещатели устанавливаются в легкодоступных местах помещений, вдоль эвакуационных путей, в коридорах, на лестничных площадках, у выходов из здания. Корпус извещателя и кнопка выделяются красным цветом. От ложного срабатывания кнопка закрыта предохранительным стеклом.

Автоматические пожарные извещатели предназначены для передачи информации о возникновении загорания в автоматическом режиме. Такая система должна обеспечить своевременное обнаружение пожара и не давать ложных срабатываний при длительной эксплуатации. Рекомендуемый тип пожарного извещателя выбирается в зависимости от назначения защищаемого помещения и вида пожарной нагрузки.

В целях профилактики предлагается проводить противопожарный инструктаж, в ходе которого работники смогут ознакомиться с правилами противопожарной безопасности, а также изучить правила использования первичных средств пожаротушения. В случае возникновения пожара необходимо отключить электропитание, вызвать по телефону пожарную команду, эвакуировать людей из помещения согласно плану эвакуации и приступить к ликвидации пожара огнетушителями. При наличии небольшого очага пламени можно воспользоваться подручными средствами с целью прекращения доступа воздуха к объекту возгорания.

**7.4. Экологичность проекта**

При работе с ПЭВМ отсутствуют выбросы вредных веществ в окружающую среду, отходы только бумажные, легко утилизируемые и перерабатываемые.

Исходя из изложенного выше, видно, что рабочее место не оказывает ярко выраженного воздействия на окружающую среду, и его нельзя считать вредным источником, оно экологически безопасно. Дополнительных мер по защите окружающей среды не требуется.

**8. ЭКОНОМИЧЕСКАЯ ЧАСТЬ**

**8.1. План-график разработки**

Разработка любого программного обеспечения состоит из нескольких стадий. Чтобы этот процесс происходил организованно, без лишних затрат труда, материальных средств и в минимальные сроки, необходимо эффективное планирование.

Основной задачей планирования является распределение по срокам и исполнителям этапов разработки программного продукта, а также определение общей продолжительности их проведения.

Для наглядного отображения плана выбрана модель, известная как ленточный график.

Ленточный график процесса (подготовки производства, разработки ПО) — это его графическая модель с указанием перечня и организационно-экономических характеристик всех работ, сроков и последовательности их исполнения, отражаемых совокупностью упорядоченных во времени горизонтальных линий.

Для построения ленточного графика необходимо сформировать перечень всех основных этапов и видов работ, которые должны быть выполнены в ходе разработки, с указанием их продолжительности. Важную роль играет логическое упорядочивание последовательности выполнения отдельных видов работ. В его основе лежит анализ смыслового содержания каждого вида работ и установление взаимосвязи между видами работ. Для каждого этапа и вида работ требуется указать список исполнителей.

Продолжительность работ определяется по формуле:

,

где

* Ti – трудоемкость работ, человеко-дни;
* ni – численность исполнителей, человек.

Соответственно, трудоемкость можно рассчитать на основании продолжительности работы и количества исполнителей:

Таблица 8.1. Данные для построения ленточного графика

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| № | Наименование этапа работ | Исполнители | | Tn, дней | Ti, человеко-дней |
| Разработчик | Руководитель |
| 1 | Получение и согласование задания | **+** | **+** | 1 | 2 |
| 2 | Ознакомление с предметной областью | **+** |  | 10 | 10 |
| 3 | Анализ и разработка требований | **+** |  | 5 | 5 |
| 4 | Согласование требований | **+** | **+** | 1 | 2 |
| 5 | Проектирование структуры Базы данных | **+** |  | 2 | 2 |
| 6 | Написание скрипта создания базы данных | **+** |  | 1 | 1 |
| 7 | Проектирование архитектуры приложения | **+** |  | 3 | 3 |
| 8 | Написание программного кода | **+** |  | 30 | 30 |
| 9 | Оформление программной документации | **+** |  | 5 | 5 |
| 10 | Оформление пояснительной записки | **+** |  | 10 | 10 |
| 11 | Сдача проекта | **+** | **+** | 1 | 2 |
| Итого: | | | | 68 | 71 |

Ленточный график разработки программы, построенный по данным таблицы 8.1, приведен на рис. 8.1.



Рис. 8.1. Ленточный график

**8.2. Составление сметы затрат на разработку**

Затраты группируются в соответствии с их экономическим содержанием по следующим элементам:

* материальные затраты (за вычетом стоимости возвратных отходов);
* затраты на оплату труда;
* амортизация основных фондов;
* прочие затраты.

**8.2.1. Материальные затраты**

Материальные затраты - элемент себестоимости продукции (работ, услуг), в котором отражается стоимость приобретаемых со стороны сырья и материалов, входящих в состав вырабатываемой продукции или являющихся необходимым компонентом при изготовлении продукции (проведении работ, оказании услуг); покупных материалов, используемых в процессе производства продукции (работ, услуг) для обеспечения нормального технологического процесса и для упаковки продукции или расходуемых на другие производственные и хозяйственные нужды.

Результаты расчета затрат на материалы приведены в таблице 8.2.

Таблица 8.2. Перечень материалов

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Наименование | Единица измерения | Количество, шт. | Цена, руб. | Сумма, руб. |
| Бумага для принетера (формат А4, 500л.) | Пачка | 1 | 250 | 250 |
| Картридж для струйного принтера | Штука | 1 | 400 | 400 |
| Компакт-диск CD-RW | Штука | 1 | 40 | 40 |
| Переплет | Штука | 1 | 300 | 300 |
| USB-накопитель | Штука | 1 | 500 | 500 |
| ПО Visual studio professinaol 2013 | Штука | 1 | 17000 | 17000 |
| ИТОГО (с учетом НДС) | | | | 18490 |
| ИТОГО (без учета НДС) | | | | 15684,7 |

Змат=15684,7рублей.

Затраты на электроэнергию вычисляются как произведе­ние стоимости затраченной электроэнергии в час на затраченное количество времени. Потребляемая мощность компьютера примерно 450 Вт. За время работы над проектом (Вр) компьютер используется: Вр = 544 ч. Стоимость 1кВт/час равна 3,45 рубля. В целом затраты на электроэнергию при разработке (Зэр) составят:

Зэр = 544\* 3,45 \*0,45= 844,56 рублей.

Тогда материальные затраты составят:

Зм=Змат+ Зэр = 15684,7+ 844,56 = 16529,26 рублей.

**8.2.2. Затраты на оплату труда**

В расходы на оплату труда включаются любые начисления работникам в денежной и (или) натуральной формах, стимулирующие начисления и надбавки, компенсационные начисления, премии и единовременные поощрительные начисления, расходы, связанные с содержанием этих работников, предусмотренные нормами законодательства РФ, трудовыми или коллективными договорами.

Работу над проектом выполняют 2 человека: руководитель проекта и разработчик. Затраты на оплату труда начисляются исходя из ставки руководителя и разработчика, и времени, затраченного на выполнение работы. Заработная плата рассчитывается по следующей формуле 8.1:

*ЗП=Т\*Д*, (8.1)

где Т - тарифная ставка;

Д - количество дней работы.

Тарифная ставка вычисляется по формуле 8.2:

, (8.2)

где  - минимальная заработная плата,

 - число рабочих дней в месяце.

Оклад руководителя проекта составляет 15000 рублей. Количество рабочих дней в месяце принимается равным 22. Оклад разработчика 10000 рублей. Общее число дней работы над проектом руководителя составляет 3 дней, разработчика–68 дней, тогда заработные платы руководителя проекта и разработчика составляют соответственно:

*ЗПрук*=15000/22\*3= 2045,4 рублей.

*ЗПраз*=10000/22\*68= 30909,1 рублей.

Следовательно, фонд оплаты труда составит:

*Фзп*=2045,4+30909,1=32954,5 рублей.

**8.2.3. Амортизация основных фондов**

Затраты по данной статье представляют собой сумму амортизационных отчислений на полное восстановление основных производственных фондов. В нашем случае рассчитывается сумма затрат на амортизацию ПЭВМ за время его использования при разработки проекта.

Так как при разработке дипломного проекта использовался личный компьютер, срок использования которого составляет 5 лет, то амортизационные начисления будут равны 0 рублей за счет превышения срока полезного использования 3 года.

**8.2.4. Прочие затраты**

Прочие затраты включают в себя страховые взносы и остальные прочие расходы.

Страховые взносы берутся в размере 30,2% от фонда оплаты труда:

Величина остальных прочих расходов берется от суммы прямых общих затрат в установленном размере. Для разработки программного обеспечения они составят 10 %.

Общие прямые затраты составят следующую сумму:

руб.

Таким образом, прочие затраты составят:

Полная смета затрат на разработку данного проекта приведена в таблице 8.3.

Таблица 8.3. Смета затрат на разработку проекта

|  |  |  |
| --- | --- | --- |
| Наименование калькуляционных статей расходов | Сумма, руб. | Удельный вес, % |
| Материальные затраты, *3м* | 16529,26 | 25,7 |
| Затраты на заработную плату, *Фзп* | 32954,5 | 50,93 |
| Амортизация ПЭВМ, *Анир* | 0 | 0 |
| Прочие расходы, *Зпр* |  | 22,89 |
| **Общие затраты:** | **64299,93** | **100** |

**8.3 Расчет проектной цены подписки на ПО**

Проект будет размещен на платном хостинге. Пользователи системы будут платить ежемесячную подписку за пользование сервисом.

Стоимость подписки будет складываться из расходов на содержание сайта, прибыли которая составляет 40% от стоимости расходов и НДС, составляющего 18% от суммы расходов и прибыли.

Расходы на содержание сайта следующие:

* Аренда доменного имени 790 рублей в год
* Хостинг 247 рублей в месяц

Себестоимость C=790/12+247=312,83 рублей/месяц

Прибыль рассчитывается следующим образом:

Пр=0,4\*С=0,4\*312,83=125,13 рублей/месяц

Тогда НДС составит:

НДС=0,18\*(С+Пр)=0,18\*(312,83+125,13)=78,83 рублей/месяц

Цена подписки будет следующей:

Цп=С+Пр+НДС=516,79 рублей/месяц.

**8.4 Расчет и выводы по эффективности предложений**

После внедрения данного продукта количество сотрудников на предприятии останется прежним, экономический эффект внедрения не произведет видимых изменений на оплату труда.

Эффективность внедрения данного программного продукта заключается в объединении территориально разделенных предприятий, что существенно повысит их конкурентоспособность.

Сотрудники предприятия будут тратить меньше времени на обмен задачами и важной информацией, что увеличит скорость работы.

У руководства предприятия появятся прозрачные механизмы контроля за деятельностью предприятия что обеспечит более стабильную и надежную работу всего предприятия в целом.

**ЗАКЛЮЧЕНИЕ**

В результате работы, выполненной в рамках данного дипломного проекта, были разработаны база данных и программное обеспечение для обеспечения функционирования виртуального предприятия.

Разработанная программа имеет продуманный интерфейс в результате чего на обучение пользованием программы необходимо минимум времени. В ходе выполнения дипломного проекта было выполнено следующее:

* 1. Проведен анализ предметной области, по его результатам выполнено проектирование базы данных для хранения информации.
  2. Разработаны алгоритмы обработки информации и взаимодействия с базой данных.
  3. Разработан алгоритм пересылки сообщений другим пользователям программы.
  4. Разработан интерфейс пользователя.
  5. Проведено тестирование разработанного ПО.
  6. Выполнена оценка экономической целесообразности разработки программы.
  7. Разработанный проект рассмотрен с точки зрения безопасности и экологичности.

Применение данного программного обеспечения позволяет решать следующие задачи:

* Коммуникация сотрудников разделенных территориально;
* хранение и отображение информации о сотрудниках, задачах;
* добавление задач и распределение их между сотрудниками;
* формирование отчетов.

По результатам проектирования можно определить следующие направления развития разработанной программы:

* добавление возможностей обмена файлами;
* добавление средств голосовой коммуникации;
* создание чата с возможностью общения более двух сотрудников одновременно.

Таким образом, в настоящем дипломном проекте была выполнена поставленная в начале дипломного проектирования задача. Разработанное программное обеспечение отвечает всем заданным требованиям.

**Библиографический список**

1.http://www.genon.ru/GetAnswer.aspx?qid=5bb77ae8-13ef-43af-b442-16986ce31a19. Дата просмотра 13.03.2015

2.http://www.osp.ru/os/2000/04/177994/. Дата просмотра 03.03.2015

3.Jan Marco Leimeister, Jцrn Weigle, Helmut Krcmar. Efficiency of virtual organisations — the case of agi, JOV 3 (2001). С.2.

4. http://www.it.ru/academy/vue-form.html. Дата просмотра 04.04.2015

5.Мескон М., Альберт М., Хедоури Ф. основы менеджмента.- М., 2005 г.С.156.

6.Виртуальное предприятие как эффективная форма организации внешнеэкономической деятельности компании. А.Манюшис. Международный журнал "Проблемы теории и практики управления" №4/03. С.44.

7. Вейл П.Искусство менеджмента: Пер.с англ.-М.:Новости, 2004.С.121.

8. Паркинсон С. Искусство управления. -М., 2003 г. С.67.

9. Друкер П. Эффективный управляющий. – М., 2004 г.С.231.

10. Г.Я. Гольдштейн Стратегический инновационный менеджмент: тенденции, технологии, практика.Таганрог: Изд-во ТРТУ, 2002.

11. Л.Санкова. HR-менеджмент в виртуальных организациях. Журнал «Управление персоналом», № 5 за 2007 год.

12.Катаев А.В. Виртуальные предприятия – новая ступень в организации НИОКР // Стратегические аспекты управления НИОКР в условиях глобальной конкуренции: Отчет по НИР №01.2.00100692. Таганрог: ТРТУ, 2001.С.71.

13. Г.Я. Гольдштейн Стратегический инновационный менеджмент: тенденции, технологии, практика. Таганрог: Изд-во ТРТУ, 2002 глава 6.

14.http://www.docme.ru/doc/544958/virtual.\_nye-predpriyatiya. Дата просмотра 13.03.2015

15. Дейт К. Дж. Введение в системы баз данных: Вильямс, 1999. С. 82.

16. Нильсен П. Microsoft SQL Server 2005. Библия пользователя: Пер. с англ. – М.: ООО «И. Д. Вильямс», 2008

17. ГОСТ 12.1.004-91 «Пожарная безопасность. Общие требования»

18. СН 2.2.4/2.1-8.562-96 «Шум на рабочих местах, в помещениях жилых, общественных зданий и на территории жилой застройки»

19. СНиП 23-05-95 «Естественное и искусственное освещение»

20. ГОСТ 12.0.003-74\* «ССБТ. Опасные и вредные производственные факторы. Классификация»

21.СанПиН 2.2.2/2.4.1340-03 «Гигиенические требования к персональным электронно-вычислительным машинам и организации работы»,

22. СН-181-70 «Указания по проектированию цветовой отделки интерьеров и производственных зданий промышленных»

23.СанПиН 2.2.4.548-96 «Гигиенические требования к микроклимату производственных помещений»

**ПРИЛОЖЕНИЕ**

**Скрипт создания БД**

/\*Тема дипломного проекта: Разработка программных средств обеспечивающих функционирование виртуального предприятия

Студент: Кирюшин А.В.

Специальность 230105, группа 043.

Руководитель проекта: доктор техн. наук,

профессор кафедры ИИБМТ Антипов В.А.

Средство разработки: Microsoft Visual Studio 2013 Professional,

Microsoft SQL server 2008

Назначение: обеспечение функционирования виртуального предприятия

Дата разработки: 01.05.2015.\*/

use master;

if DB\_ID ('virtual\_enterprise') IS NOT NULL

drop database virtual\_enterprise;

create database virtual\_enterprise

go

use virtual\_enterprise

create table VirtualEnterprise

(

idVE int identity primary key not null

,name nvarchar(128) not null

)

create table Employees

(

employeeId int identity primary key not null

,name nvarchar(32) not null

,family nvarchar(64)

,patronymic nvarchar(64)

,mail nvarchar(64) not null

,passwordEmployee nvarchar(32) not null

)

create table CompanyEmployee(

idCompanyEmployee int identity primary key not null

,idEmployee int not null

,idVE int not null

)

create table Project(

idProject int identity primary key not null

,name nvarchar(64) not null

,deadline datetime

,idVE int not null

)

create table Tasks(

taskId int identity primary key not null

,idAuthor int not null

,name nvarchar(128) not null

,dataCreation datetime not null

,deadline datetime

,wastedTime int

,idExecutor int

,idProject int not null

,isTaskComplete bit

)

create table Mesage(

idMessage int identity primary key not null

,content nvarchar(1024) not null

,idDialogue int not null

,dateStart datetime not null

)

create table Subscription(

idSubscription int identity primary key not null

,idSignedEmployee int not null

,idFellowEmployee int not null

)

create table Dialogue(

idDialogue int identity primary key not null

,idDialogueEmployee int not null

)

create table DialogueEmployee(

idDialogueEmployee int identity primary key not null

,idEmployee int not null

)

create table TimeOnProject(

id int identity primary key not null

,[date] date not null

,idEmployee int not null

,idProject int not null

,countTime int not null

)

create table PostSubscription(

idPostSubscription int identity primary key not null

,[content] nvarchar(1024) not null

,dateStart datetime not null

,idSubscription int not null

)

alter table PostSubscription

add constraint FK\_come

foreign key(idSubscription)

references Subscription(idSubscription)

alter table CompanyEmployee

add constraint FK\_IdEmployee

foreign key(idEmployee)

references Employees(employeeId)

on delete cascade

on update cascade,

constraint FK\_idVE

foreign key (idVE)

references virtualEnterprise(idVE)

on delete cascade

on update cascade

alter table Project

add constraint FK\_project\_IdVE

foreign key(idVE)

references virtualEnterprise(idVE)

on delete cascade

alter table TimeOnProject

add constraint FK\_project

foreign key(idProject)

references Project(idProject)

on delete cascade

on update cascade,

constraint FK\_employee

foreign key(idEmployee)

references Employees(employeeId)

on delete cascade

on update cascade

alter table Tasks

add constraint FK\_Project\_contain

foreign key(idProject)

references Project(idProject)

on delete cascade,

constraint FK\_author\_task

foreign key(idAuthor)

references Employee(employeeId)

on delete no action,

constraint FK\_executor\_task

foreign key(idExecutor)

references Employee(employeeId)

on delete cascade

on update cascade

alter table Mesage

add constraint FK\_Dialogue\_contains

foreign key(idDialogue)

references Dialogue(idDialogue)

on delete cascade

alter table Subscription

add constraint FK\_Employee\_signed

foreign key(idSignedEmployee)

references Employees(employeeId)

on delete no action,

constraint FK\_idFellowEmployee

foreign key (idFellowEmployee)

references Employees(employeeId)

on delete cascade

alter table Dialogue

add constraint FK\_DialogueEmployee

foreign key(idDialogueEmployee)

references DialogueEmployee(idDialogueEmployee)

on delete cascade

alter table DialogueEmployee

add constraint FK\_employee\_spent

foreign key(idEmployee)

references Employees(employeeId)

on update cascade

on delete cascade

**Класс Mail**

/// <summary>

/// Отвечает за отправку писем на электронну почту

/// </summary>

class Mail

{

/// <summary>

/// smtp-сервер

/// </summary>

private string smtp = "smtp.mail.ru";

/// <summary>

/// отправитель

/// </summary>

private string from = "VirtualEnterprize@mail.ru";

/// <summary>

/// пароль почты

/// </summary>

private string password = "19930511q";

/// <summary>

/// тема письма

/// </summary>

private string subject = "";

/// <summary>

/// Адресат

/// </summary>

private string to = "";

/// <summary>

/// Содержание письма

/// </summary>

private string body = "";

/// <summary>

/// Логин

/// </summary>

private string login = "VirtualEnterprize@mail.ru";

/// <summary>

/// Отправляет письмо

/// </summary>

/// <param name="sendTo">адресат письма</param>

/// <param name="subject">тема письма</param>

/// <param name="body">содержание письма в формате html</param>

public void Send(string sendTo, string subject, string body)

{

this.from = login;

this.to = sendTo;

this.body = body;

this.subject = subject;

SmtpClient smtpServer = new SmtpClient(smtp, 25);

smtpServer.Credentials = new NetworkCredential(login, password);

smtpServer.EnableSsl = true;

smtpServer.Send(GetMessage());

}

/// <summary>

/// Получить сообщение

/// </summary>

/// <returns>возвращает обьект письмо</returns>

private MailMessage GetMessage()

{

MailMessage Message = new MailMessage();

Message.From = new MailAddress(login);

Message.To.Add(new MailAddress(to));

Message.Subject = subject;

Message.IsBodyHtml = true;

Message.Body = body;

return Message;

}

}

**Класс Password**

/// <summary>

/// Класс для генерации паролей

/// </summary>

public class Password

{

/// <summary>

/// /// <summary>

/// Сгенерировать случайный пароль

/// </summary>

/// <param name="lenght">длина пароля</param>

/// <returns>возвращает сгенирированный пароль</returns>

public string GeneratePassword(int lenght)

{

int[] arr = new int[lenght];

Random rnd = new Random();

string Password = "";

for (int i = 0; i < arr.Length; i++)

{

//генерируем случайное число и заменяем на символ согласно таблице ASC

arr[i] = rnd.Next(48, 122);

Password += (char)arr[i];

}

return Password;

}

}

}

**Класс Date**

/// <summary>

/// Отвечает за обмен данными с базой данных

/// </summary>

class Data

{

/// <summary>

/// Строка соединения с сервером

/// </summary>

private static readonly string ConnectionString = @"Data Source=USER-ПК\SQLEXPRESS;Initial Catalog=virtual\_enterprise;Integrated security=true";

/// <summary>

/// Авторизация на сервере

/// </summary>

/// <param name="mail">электронная почта</param>

/// <param name="password">пароль</param>

/// <returns>если возвращается 1 значит соединение успешно установлено</returns>

public int Login(string mail, string password)

{

int findEmployee = 0;

using (var connection = new SqlConnection(ConnectionString))

{

connection.Open();

using (var cmd = connection.CreateCommand())

{

cmd.CommandText = @"select count (employeeId)

from employees

where mail=@mail and passwordEmployee=@password";

cmd.Parameters.AddWithValue("@mail", mail);

cmd.Parameters.AddWithValue("@password", password);

findEmployee = (int)cmd.ExecuteScalar();

}

}

return findEmployee;

}

/// <summary>

/// Регистрация компании в системе

/// </summary>

/// <param name="mail">почта</param>

/// <param name="password">пароль</param>

/// <returns>возвращает true если регистрация успешна</returns>

public bool Registration(string mail, string password)

{

if (FindMail(mail) == false)

{

using (var connection = new SqlConnection(ConnectionString))

{

connection.Open();

using (var cmd = connection.CreateCommand())

{

cmd.CommandText = @"

insert into

Employees

(

mail

,passwordEmployee

,name

)

values

(

@mail

,@password

,@name

)

";

cmd.Parameters.AddWithValue("@mail", mail);

cmd.Parameters.AddWithValue("@password", password);

cmd.Parameters.AddWithValue("@name", mail);

cmd.ExecuteNonQuery();

}

return true;

}

}

else return false;

}

/// <summary>

/// Ищет конкретный адрес в базе данных

/// </summary>

/// <param name="mail">почта</param>

/// <returns>возвращает true если адрес найден</returns>

public bool FindMail(string mail)

{

bool mailFind = false;

int countMail;

using (var connection = new SqlConnection(ConnectionString))

{

connection.Open();

string findMail = @"select count (mail)

from employees

where mail=@mail";

using (var cmd = new SqlCommand(findMail, connection))

{

cmd.Parameters.AddWithValue("@mail", mail);

countMail = (int)cmd.ExecuteScalar();

}

}

if (countMail > 0)

{

mailFind = true;

}

return mailFind;

}

/// <summary>

///Найти все проекты предприятия

/// </summary>

/// <param name="idVirtualEnterprise"></param>

/// <returns>возвращает все проекты предприятия</returns>

public List<Project> FindAllProject(int idVirtualEnterprise)

{

var listProject = new List<Project>();

using (var connection = new SqlConnection(ConnectionString))

{

using (var cmd = connection.CreateCommand())

{

connection.Open();

cmd.CommandText = @"

select \*

from

Project

where

idVE=@idVirtualEnterprise

";

cmd.Parameters.AddWithValue("@idVirtualEnterprise", idVirtualEnterprise);

using (var reader = cmd.ExecuteReader())

{

while (reader.Read())

{

var project = new Project

{

IdProject = reader.GetInt32(reader.GetOrdinal("idProject")),

Name = reader.GetString(reader.GetOrdinal("name")),

Deadline = reader.GetDateTime(reader.GetOrdinal("deadline")),

IdVirtualEnterprise = reader.GetInt32(reader.GetOrdinal("idVE")),

};

listProject.Add(project);

}

}

}

}

return listProject;

}

/// <summary>

/// Найти все задачи проекта

/// </summary>

/// <param name="idProject">код проекта</param>

/// <returns>возвращает все задачи проекта</returns>

public List<Task> FindTasksProject(int idProject)

{

var listTask = new List<Task>();

using (var connection = new SqlConnection(ConnectionString))

{

using (var cmd = connection.CreateCommand())

{

connection.Open();

cmd.CommandText = @"

select \*

from

Tasks

where

idProject=@idProject

";

cmd.Parameters.AddWithValue("@idProject", idProject);

using (var reader = cmd.ExecuteReader())

{

while (reader.Read())

{

var task = new Task

{

TaskId = reader.GetInt32(reader.GetOrdinal("taskId")),

IdProject = reader.GetInt32(reader.GetOrdinal("idProject")),

Name = reader.GetString(reader.GetOrdinal("name")),

Deadline = reader.GetDateTime(reader.GetOrdinal("deadline")),

DataCreation = reader.GetDateTime(reader.GetOrdinal("dateCreation")),

WastedTime = reader.GetInt32(reader.GetOrdinal("wastedTime")),

IdExecutor = reader.GetInt32(reader.GetOrdinal("idExecutor")),

isTaskComplete = reader.GetBoolean(reader.GetOrdinal("isTaskComplete")),

Author = reader.GetInt32(reader.GetOrdinal("idAuthor")),

};

listTask.Add(task);

}

}

}

}

return listTask;

}

/// <summary>

/// Найти всех сотрудников предприятия

/// </summary>

/// <param name="idVirtualEnterprise">код виртуально предприятия</param>

/// <returns>вовращает всех сотрудников</returns>

public List<Employee> FindAllEmployee(int idVirtualEnterprise)

{

var listEmployee = new List<Employee>();

using (var connection = new SqlConnection(ConnectionString))

{

using (var cmd = connection.CreateCommand())

{

connection.Open();

cmd.CommandText = @"

select \*

from

Employees,CompanyEmployee

where

idVE=@idVirtualEnterprise and

Employees.employeeId=CompanyEmployee.idEmployee

";

cmd.Parameters.AddWithValue("@idVirtualEnterprise", idVirtualEnterprise);

listEmployee = GetEmployees(cmd);

}

}

return listEmployee;

}

/// <summary>

/// Найти участников проекта

/// </summary>

/// <param name="idproject">код проекта</param>

/// <returns>возвращает всех участников проекта</returns>

public List<Employee> FindParticipantProject(int idProject)

{

var listEmployee = new List<Employee>();

using (var connection = new SqlConnection(ConnectionString))

{

using (var cmd = connection.CreateCommand())

{

connection.Open();

cmd.CommandText = @"

select \*

from

Employees

where

idProject=@idProject

";

cmd.Parameters.AddWithValue("@idProject", idProject);

listEmployee = GetEmployees(cmd);

}

}

return listEmployee;

}

/// <summary>

/// Считывает данные сотрудников из базы данных

/// </summary>

/// <param name="cmd">sql-команда с запросом</param>

/// <returns>возвращает лист сотрудников</returns>

private List<Employee> GetEmployees(SqlCommand cmd)

{

var listEmployee = new List<Employee>();

using (var reader = cmd.ExecuteReader())

{

while (reader.Read())

{

var employee = new Employee

{

IdEmployee = reader.GetInt32(reader.GetOrdinal("idEmployee")),

Name = reader.GetString(reader.GetOrdinal("name")),

Family = reader.GetString(reader.GetOrdinal("family")),

Patronymic = reader.GetString(reader.GetOrdinal("patronymic")),

Mail = reader.GetString(reader.GetOrdinal("mail")),

Password = reader.GetString(reader.GetOrdinal("passwordEmployee")),

Position = reader.GetString(reader.GetOrdinal("position")),

};

listEmployee.Add(employee);

}

}

return listEmployee;

}

/// <summary>

/// Добавить сотрудника в проект

/// </summary>

/// <param name="idEmployee">код сотрудника</param>

/// <param name="idProject">код проекта</param>

public void AddEmployeeInProject(int idEmployee, int idProject)

{

using (var connection = new SqlConnection(ConnectionString))

{

string addEmployee = @"insert into ParticipantProject

(

idEmployee,

idProject

)

values

(

@idEmployee,

)

";

connection.Open();

using (var cmd = new SqlCommand(addEmployee, connection))

{

cmd.Parameters.AddWithValue("@idEmployee", idEmployee);

cmd.Parameters.AddWithValue("@idProject", idProject);

cmd.ExecuteNonQuery();

}

}

}

/// <summary>

/// Изменить статус выполнения задачи

/// </summary>

/// <param name="idTask">код задачи</param>

public void ChangeStatusTask(int idTask)

{

bool isTaskComplete;

using (var connection = new SqlConnection(ConnectionString))

{

using (var cmd = connection.CreateCommand())

{

connection.Open();

cmd.CommandText = @"

select isTaskComplete

from Tasks

where taskId=@idTask";

cmd.Parameters.AddWithValue("@idTask", idTask);

isTaskComplete = (bool)cmd.ExecuteScalar();

}

isTaskComplete = !isTaskComplete;

int taskComplete=0;

if (isTaskComplete)

{

taskComplete = 1;

}

using (var cmd = connection.CreateCommand())

{

cmd.CommandText = @"

update Tasks

set isTaskComplete=@taskComplete

where taskId=@idTask";

cmd.Parameters.AddWithValue("@taskComplete", taskComplete);

cmd.Parameters.AddWithValue("@idTask", idTask);

cmd.ExecuteNonQuery();

}

}

}

/// <summary>

/// Удалить сотрудника из проекта

/// </summary>

/// <param name="idEmployee">код сотрудника</param>

/// <param name="idProject">код проекта</param>

public void RemoveEmployeeFromProject(int idEmployee, int idProject)

{

using (var connection = new SqlConnection(ConnectionString))

{

using (var cmd = connection.CreateCommand())

{

connection.Open();

cmd.CommandText = @"

delete from ParticipantProject

where idEmployee=@idEmployee

and idProject=@idProject";

cmd.Parameters.AddWithValue("@idEmployee", idEmployee);

cmd.Parameters.AddWithValue("@idProject", idProject);

cmd.ExecuteNonQuery();

}

}

}

/// <summary>

/// Добавить задачу

/// </summary>

/// <param name="name">имя</param>

/// <param name="deadline">срок завершения</param>

/// <param name="idAuthor">код автора</param>

/// <param name="idExecutor">код исполнителя</param>

/// <param name="idProject">код проекта</param>

public void AddTask(string name, DateTime? deadline, int idAuthor, int idExecutor, int idProject)

{

using (var connection = new SqlConnection(ConnectionString))

{

//текущее время

DateTime? curTime = DateTime.Now;

string addTask = @"insert into Tasks

(

name,

dataCreation,

deadline,

idAuthor,

idExecutor,

idProject

)

values

(

@name,

@dataCreation,

@deadline,

@idAuthor,

@idExecutor

@idProject,

)

";

connection.Open();

using (var cmd = new SqlCommand(addTask, connection))

{

cmd.Parameters.AddWithValue("@name", name);

cmd.Parameters.AddWithValue("@dataCreation", curTime);

cmd.Parameters.AddWithValue("@deadline", deadline);

cmd.Parameters.AddWithValue("@idAuthor",idAuthor);

cmd.Parameters.AddWithValue("@idExecutor", idExecutor);

cmd.Parameters.AddWithValue("@idProject", idProject);

cmd.ExecuteNonQuery();

}

}

}

/// <summary>

/// Изменить задачу

/// </summary>

/// <param name="idTask">код задачи</param>

/// <param name="deadline">срок завершения</param>

/// <param name="idExecutor">код исполнителя</param>

/// <param name="idProject">код проекта</param>

public void ChangeTask(int idTask, DateTime? deadline, int idExecutor, int idProject)

{

using (var connection = new SqlConnection(ConnectionString))

{

using (var cmd = connection.CreateCommand())

{

cmd.CommandText = @"

update Tasks

set deadline=@deadline,

idExecutor=@idExecutor,

idProject=@idProject

where taskId=@idTask";

cmd.Parameters.AddWithValue("@deadline", deadline);

cmd.Parameters.AddWithValue("@idExecutor", idExecutor);

cmd.Parameters.AddWithValue("@idProject", idProject);

cmd.Parameters.AddWithValue("@idTask", idTask);

cmd.ExecuteNonQuery();

}

}

}

/// <summary>

/// Удалить задачу

/// </summary>

/// <param name="idTask">код задачи</param>

public void DeleteTask(int idTask)

{

using (var connection = new SqlConnection(ConnectionString))

{

using (var cmd = connection.CreateCommand())

{

connection.Open();

cmd.CommandText = @"

delete from Tasks

where idTask=@idTask";

cmd.Parameters.AddWithValue("@idEmployee", idTask);

cmd.ExecuteNonQuery();

}

}

}

/// <summary>

/// Добавить время потраченное на задачу

/// </summary>

/// <param name="idTask">код задачи</param>

/// <param name="countTime">количество времени</param>

public void AddTimeInTask(int idTask, int countTime)

{

using (var connection = new SqlConnection(ConnectionString))

{

using (var cmd = connection.CreateCommand())

{

cmd.CommandText = @"

update Tasks

set wastedTime=wastedTime+ @countTime

where taskId=@idTask";

cmd.Parameters.AddWithValue("@countTime",countTime);

cmd.Parameters.AddWithValue("@idTask", idTask);

cmd.ExecuteNonQuery();

}

}

}

public List<Message> FindAllMessageDialogue(int idDialogue)

{

var listMessage = new List<Message>();

using (var connection = new SqlConnection(ConnectionString))

{

using (var cmd = connection.CreateCommand())

{

connection.Open();

cmd.CommandText = @"

select \*

from

Message

where

idDialogue=@idDialogue

";

cmd.Parameters.AddWithValue("@idDialogue", idDialogue);

using (var reader = cmd.ExecuteReader())

{

while (reader.Read())

{

var message = new Message

{

IdMessage = reader.GetInt32(reader.GetOrdinal("idMessage")),

Content = reader.GetString(reader.GetOrdinal("content")),

IdDialogue = reader.GetInt32(reader.GetOrdinal("idDialogue")),

DateStart = reader.GetDateTime(reader.GetOrdinal("dateStart")),

};

listMessage.Add(message);

}

}

}

}

return listMessage;

}

/// <summary>

/// Добавить сообщение в диалог

/// </summary>

/// <param name="idDialogue">код диалога</param>

/// <param name="message">сообщение</param>

public void AddMessage(int idDialogue, Message message)

{

using (var connection = new SqlConnection(ConnectionString))

{

//текущее время

DateTime? curTime = DateTime.Now;

string addTask = @"insert into Tasks

(

idMessage,

content,

idDialogue,

dateStart,

)

values

(

@idMessage,

@content,

@idDialogue,

@dateStart

)

";

connection.Open();

using (var cmd = new SqlCommand(addTask, connection))

{

cmd.Parameters.AddWithValue("@idMessage", message.IdMessage);

cmd.Parameters.AddWithValue("@content",message.Content);

cmd.Parameters.AddWithValue("@idDialogue",message.IdDialogue);

cmd.Parameters.AddWithValue("@dateStart", message.DateStart);

cmd.ExecuteNonQuery();

}

}

}

/// <summary>

/// найти все диалоги сотрудника

/// </summary>

/// <param name="idEmployee">код сотрудника</param>

/// <returns>возвращает диалоги сотрудника</returns>

public List<Dialogue> FindAllDialogueEmployee(int idEmployee)

{

var listDialogue = new List<Dialogue>();

using (var connection = new SqlConnection(ConnectionString))

{

using (var cmd = connection.CreateCommand())

{

connection.Open();

cmd.CommandText = @"

select \*

from

Dialogue,DialogueEmployee

where

idEmployee=@idEmployee and

Dialogue.idDialogueEmployee=

DialogueEmployee.idDialogueEmployee

";

cmd.Parameters.AddWithValue("@idEmployee", idEmployee);

using (var reader = cmd.ExecuteReader())

{

while (reader.Read())

{

var dialogue = new Dialogue

{

IdDialogue = reader.GetInt32(reader.GetOrdinal("idDialogue")),

IdDialogueEmployee = reader.GetInt32(reader.GetOrdinal("idDialogueEmployee")),

};

listDialogue.Add(dialogue);

}

}

}

}

return listDialogue;

}

/// <summary>

/// Добавляет подписку на сотрудника

/// </summary>

/// <param name="idSignedEmployee">код подписывающегося сотрудника</param>

/// <param name="idFellowSubscription">код сотрудника на которого подписываются</param>

public void AddSubscription(int idSignedEmployee, int idFellowSubscription)

{

using (var connection = new SqlConnection(ConnectionString))

{

string addSubscription = @"insert into Subscription

(

idSignedEmployee,

idFellowSubscription,

)

values

(

@idSignedEmployee,

@idFellowSubscription,

)

";

connection.Open();

using (var cmd = new SqlCommand(addSubscription, connection))

{

cmd.Parameters.AddWithValue("@idSignedEmployee", idSignedEmployee);

cmd.Parameters.AddWithValue("@idFellowSubscription", idFellowSubscription);

cmd.ExecuteNonQuery();

}

}

}

/// <summary>

/// Отписаться от подписки

/// </summary>

/// <param name="idSignedEmployee">код подписывающегося сотрудника</param>

/// <param name="idFellowSubscription">код сотрудника на которого подписываются</param>

public void RemoveSubscription(int idSignedEmployee, int idFellowSubscription)

{

using (var connection = new SqlConnection(ConnectionString))

{

using (var cmd = connection.CreateCommand())

{

connection.Open();

cmd.CommandText = @"

delete from Subscription

where idSignedEmployee=@idSignedEmployee

and idFellowSubscription=@idFellowSubscription ";

cmd.Parameters.AddWithValue("@idFellowSubscription", idFellowSubscription);

cmd.Parameters.AddWithValue("@idSignedEmployee", idSignedEmployee);

cmd.ExecuteNonQuery();

}

}

}

/// <summary>

/// Найти все сообщения от подписок

/// </summary>

/// <param name="idSignedEmployee">код подписавшегося сотрудника</param>

/// <returns>сообщения от подписок</returns>

public List<SubscriptionMessage> FindAllSubscriptionMessage(int idSignedEmployee)

{

var listSubscriptionMessage = new List<SubscriptionMessage>();

using (var connection = new SqlConnection(ConnectionString))

{

using (var cmd = connection.CreateCommand())

{

connection.Open();

cmd.CommandText = @"

select \*

from

Subscription,PostSuscription

where

Subscription.idSubscription=

PostSuscription.PostSuscription

and idSignedEmployee=@idSignedEmployee

";

cmd.Parameters.AddWithValue("@idSignedEmployee", idSignedEmployee);

using (var reader = cmd.ExecuteReader())

{

while (reader.Read())

{

var subscriptionMessage = new SubscriptionMessage

{

IdPostSubscription= reader.GetInt32(reader.GetOrdinal("IdPostSubscription")),

Content = reader.GetString(reader.GetOrdinal("[content]")),

DateStart = reader.GetDateTime(reader.GetOrdinal("dateStart")),

IdSubscription = reader.GetInt32(reader.GetOrdinal("idSubscription")),

};

listSubscriptionMessage.Add(subscriptionMessage);

}

}

}

}

return listSubscriptionMessage;

}

/// <summary>

/// Получить иноформацию сотрудника

/// </summary>

/// <param name="idEmployee">код сотрудника</param>

/// <returns>возвращает сотрудника</returns>

public Employee GetEmployee(int idEmployee)

{

var employee=new Employee();

using (var connection = new SqlConnection(ConnectionString))

{

using (var cmd = connection.CreateCommand())

{

connection.Open();

cmd.CommandText = @"

select \*

from

Employees

where

idEmployee=@idEmployee

";

cmd.Parameters.AddWithValue("@idEmployee", idEmployee);

using (var reader = cmd.ExecuteReader())

{

while (reader.Read())

{

employee.Name= reader.GetString(reader.GetOrdinal("name"));

employee.IdEmployee=reader.GetInt32(reader.GetOrdinal("employeeId"));

employee.Family=reader.GetString(reader.GetOrdinal("family"));

employee.Patronymic=reader.GetString(reader.GetOrdinal("patronymic"));

employee.Password=reader.GetString(reader.GetOrdinal("passwordEmployee"));

employee.Mail=reader.GetString(reader.GetOrdinal("mail"));

employee.Position = reader.GetString(reader.GetOrdinal("position"));

}

}

}

}

return employee;

}

/// <summary>

/// Изменить инофрмацию сотрудника

/// </summary>

/// <param name="employee">сотрудник</param>

public void ChangeEmployee(Employee employee)

{

using (var connection = new SqlConnection(ConnectionString))

{

using (var cmd = connection.CreateCommand())

{

cmd.CommandText = @"

update Employees

set name,

family,

patronymic,

mail,

password,

position

where emloyeeId=@idEmployee";

cmd.Parameters.AddWithValue("@name",employee.Name);

cmd.Parameters.AddWithValue("@family", employee.Family);

cmd.Parameters.AddWithValue("@patronymic", employee.Patronymic );

cmd.Parameters.AddWithValue("@mail",employee.Mail);

cmd.Parameters.AddWithValue("@password", employee.Password);

cmd.Parameters.AddWithValue("@position", employee.Position);

cmd.Parameters.AddWithValue("@idEmployee", employee.IdEmployee);

cmd.ExecuteNonQuery();

}

}

}

}

}

**Класс Dialogue**

/// <summary>

/// Хранит информацию о диалоге

/// </summary>

class Dialogue

{

/// <summary>

/// Код диалога

/// </summary>

public int IdDialogue { set; get; }

/// <summary>

/// Код диалога сотрудника

/// </summary>

public int IdDialogueEmployee { set; get; }

}

**Класс employee**

/// <summary>

/// Хранит информацию о сотруднике предприятия

/// </summary>

class Employee

{

/// <summary>

/// Устанавливает и возвращает код сотрудника

/// </summary>

public int IdEmployee { set; get; }

/// <summary>

/// Устанавливает и возвращает имя

/// </summary>

public string Name { set; get; }

/// <summary>

/// Устанавливает и возвращает фамилию

/// </summary>

public string Family { set; get; }

/// <summary>

/// Устанавливает и возвращает отчество

/// </summary>

public string Patronymic { set; get; }

/// <summary>

/// Устанавливает и возвращает электронную почту

/// </summary>

public string Mail { set; get; }

/// <summary>

/// Устанавливает и возвращает пароль

/// </summary>

public string Password { set; get; }

/// <summary>

/// Устанавливает и возвращает должность

/// </summary>

public string Position { set; get; }

}

**Класс message**

/// <summary>

/// Хранит информацию о сообщении

/// </summary>

class Message

{

/// <summary>

/// Устанавливает и возвращает код сообщения

/// </summary>

public int IdMessage { set; get; }

/// <summary>

/// Устанавливает и возвращает контент

/// </summary>

public string Content { set; get; }

/// <summary>

/// Устанавливает и возвращает код диалога

/// </summary>

public int IdDialogue { set; get; }

/// <summary>

/// Устанавливает и возвращает дату отправки

/// </summary>

public DateTime dateStart { set; get; }

}

}

**Class project**

/// <summary>

/// Хранит информацию о проекте

/// </summary>

public class Project

{

/// <summary>

/// Устанавливает и возвращает код проекта

/// </summary>

public int IdProject { set; get; }

/// <summary>

/// Устанавливает и возвращает Имя проекта

/// </summary>

public int Name { set; get; }

/// <summary>

/// Устанавливает и возвращает срок завершения проекта

/// </summary>

public DateTime? Deadline { set; get; }

/// <summary>

/// Устанавливает и возвращает код виртуального предприятия

/// </summary>

public int IdVirtualEnterprise { set; get; }

}

**Class projectParticipant**

/// <summary>

/// Хранит информацию о участнике проекта

/// </summary>

class ProjectParticipant

{

/// <summary>

/// Устанавливает и возвращает количество потраченного времени на проект

/// </summary>

public int CountTime{ set; get; }

/// <summary>

///Устанавливает и возвращает имя сотрудника

/// </summary>

public int Name { set; get; }

/// <summary>

/// Устанавливает и возвращает фамилию сотрудника

/// </summary>

public int Family { set; get; }

}

**Класс SubscriptionMessage**

/// <summary>

/// Хранит сообщения от подписок

/// </summary>

class SubscriptionMessage

{

/// <summary>

/// Код сообщения подписки

/// </summary>

public int IdPostSubscription{ set; get; }

/// <summary>

/// Содержимое сообщения

/// </summary>

public string Content { set; get; }

/// <summary>

/// Дата отправки

/// </summary>

public DateTime dateStart { set; get; }

/// <summary>

/// Код подписки

/// </summary>

public int idSubscription { set; get; }

}

**Класс Task**

/// <summary>

/// хранит информацию о задаче

/// </summary>

public class Task

{

/// <summary>

/// Устанавливает и возвращает код задачи

/// </summary>

public int TaskId{set;get;}

/// <summary>

/// Устанавливает и возвращает имя задачи

/// </summary>

public string Name { set; get; }

/// <summary>

/// Устанавливает и возвращает дата создания

/// </summary>

public string DataCreation { set; get; }

/// <summary>

/// Устанавливает и возвращает срок завершения

/// </summary>

public DateTime? Deadline {set;get;}

/// <summary>

/// Устанавливает и возвращает потраченное время

/// </summary>

public int WastedTime { set; get; }

/// <summary>

/// Устанавливает и возвращает код исполнителя

/// </summary>

public int IdExecutor{set;get;}

/// <summary>

/// Устанавливает и возвращает код проекта

/// </summary>

public int IdProject { set; get; }

}