

Федеральное государственное автономное
образовательное учреждение
высшего образования
«СИБИРСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»
Институт космических и информационных технологий
институт
Кафедра «Информатика»
кафедра

УТВЕРЖДАЮ
Заведующий кафедрой
_____ А.С. Кузнецов
подпись инициалы, фамилия
« ____ » _____ 2019г.

БАКАЛАВРСКАЯ РАБОТА

09.03.04 Программная инженерия

код и наименование специальности

Автоматизация управления задачами в компании «3Д Коре»

тема

Руководитель	доцент кафедры «Информатика», канд. тех. н.		В.В. Тынченко
	_____	_____	_____
	должность, ученая степень	подпись, дата	фамилия, инициалы
Выпускник			А.С. Поплюйков
		_____	_____
		подпись, дата	фамилия, инициалы
Нормоконтролер			О.А. Антамошкин
		_____	_____
		подпись, дата	фамилия, инициалы

Красноярск 2019

Реферат

Выпускная квалификационная работа по теме «Автоматизация управления задачами в компании «3Д Коре»», содержит 55 страниц текстового материала, 22 рисунка, 16 использованных источников.

Цель выпускной квалификационной работы заключается в автоматизации управления задачами в компании, путем разработки веб-приложения.

Выпускная квалификационная работа содержит введение, 2 главы, заключение, список использованных источников.

Первая глава содержит анализ предметной области, обосновании внедрения программного средства, а также анализ похожих программных средств.

Вторая глава включает проектирование базы данных, программного обеспечения, руководство программиста и руководство пользователя.

Заключение содержит анализ полученных результатов.

Содержание

Введение	4
1. Аналитическая часть.	6
1.1 Характеристика объекта исследования	6
1.2 Обоснование необходимости создания автоматизированной информационной системы	9
1.3 Требования к разрабатываемой автоматизированной системе	13
1.4 Анализ программных продуктов аналогичного назначения	16
1.5 Выбор средств разработки	20
1.5.1 Выбор системы управления базами данных	20
1.5.2 Выбор среды программирования	23
2 Разработка автоматизированной системы	28
2.1 Концептуальное проектирование базы данных	28
2.2 Логическое проектирование базы данных	30
2.3 Проектирование физической модели базы данных.....	33
2.4 Проектирование программного обеспечения	34
2.4.1 Диаграмма прецедентов	34
2.4.2 Структура автоматизированной системы.....	36
2.4.3 UML диаграмма активности	38
2.5 Руководство программиста.....	40
2.6 Руководство пользователя	41
Заключение	49
Список используемой литературы:	51

Введение

Ни для кого не секрет, что сегодня каждое предприятие, от небольшой компании до огромной корпорации, стремится занять на рынке лидирующее положение, а для этого необходимо правильно и рационально управлять своими бизнес задачами. В современных условиях эффективное управление представляет собой ценный ресурс организации. Следовательно, повышение эффективности управленческой деятельности становится основным направлением совершенствования деятельности предприятия в целом. Наиболее очевидным способом повышения эффективности протекания трудового процесса является его автоматизация.

Но самое трудное при решении этой задачи – построить единую систему, которая будет отвечать запросам сотрудников и руководства всех подразделений. Одним из последних достижений в эволюции автоматизированных интегрированных систем управления предприятием является разработка системы планирования бизнес-задач предприятия с помощью визуального инструмента канбан-доски.

Канбан-доска – это таблица с несколькими столбцами, внутри столбцов находятся стикеры с задачами. Все задачи, выполняемые в их рамках, можно вручную распределять по стадиям, которые может самостоятельно определить руководитель предприятия. Перемещать задачи смогут сами исполнители. Очень удобно отслеживать, на каком этапе работы находится та или иная задача. Легко и удобно поддается контролю остаток времени до завершения выполнения бизнес-задачи, что позволяет эффективно распоряжаться трудовыми и материальными ресурсами компании.

Цель выпускной квалификационной работы состоит в повышении эффективности системы управления бизнес-задачами на предприятии «ЗДКоре» посредством автоматизации ее функций.

Задачи работы:

- провести анализ деятельности предприятия, обосновать необходимость разработки автоматизированной системы;
- выявить основные требования к разрабатываемой автоматизированной системе;
- исследовать рынок существующих программных продуктов аналогичного назначения;
- обосновать выбор инструментов разработки;
- спроектировать и разработать автоматизированную систему управления задачами.

1. Аналитическая часть

1.1 Характеристика объекта исследования

Компания «3Д Коре» осуществляет проектирование и разработку 3Д принтеров. Также у компании есть структурные подразделения, интернет-магазин форм для печенья и пряников «3Д Китчен» и подразделение по оказанию услуг 3Д печати, «3Д Центр». Компания имеет 2 офиса в городе Красноярск. Штат сотрудников компании насчитывает 10 человек. На рисунке 1.1 приведена организационная диаграмма компании.

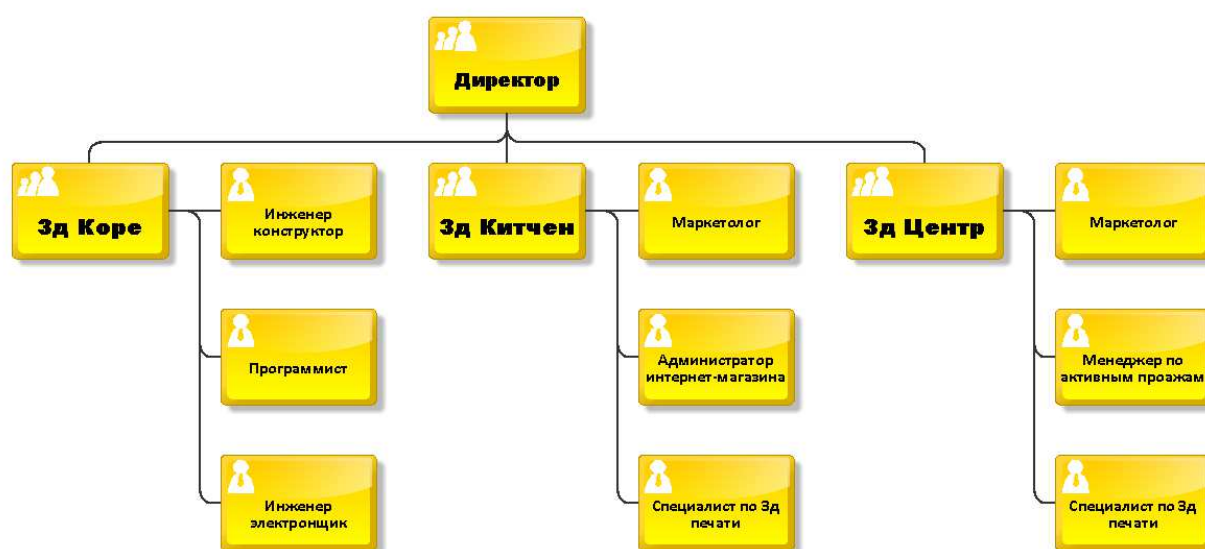


Рисунок 1.1 – Организационная диаграмма компании

Для того чтобы понимать структуру постановки задач для структурных подразделений, была составлена диаграмма цепочек добавленной стоимости, VAD-диаграмма (рисунок 1.2).

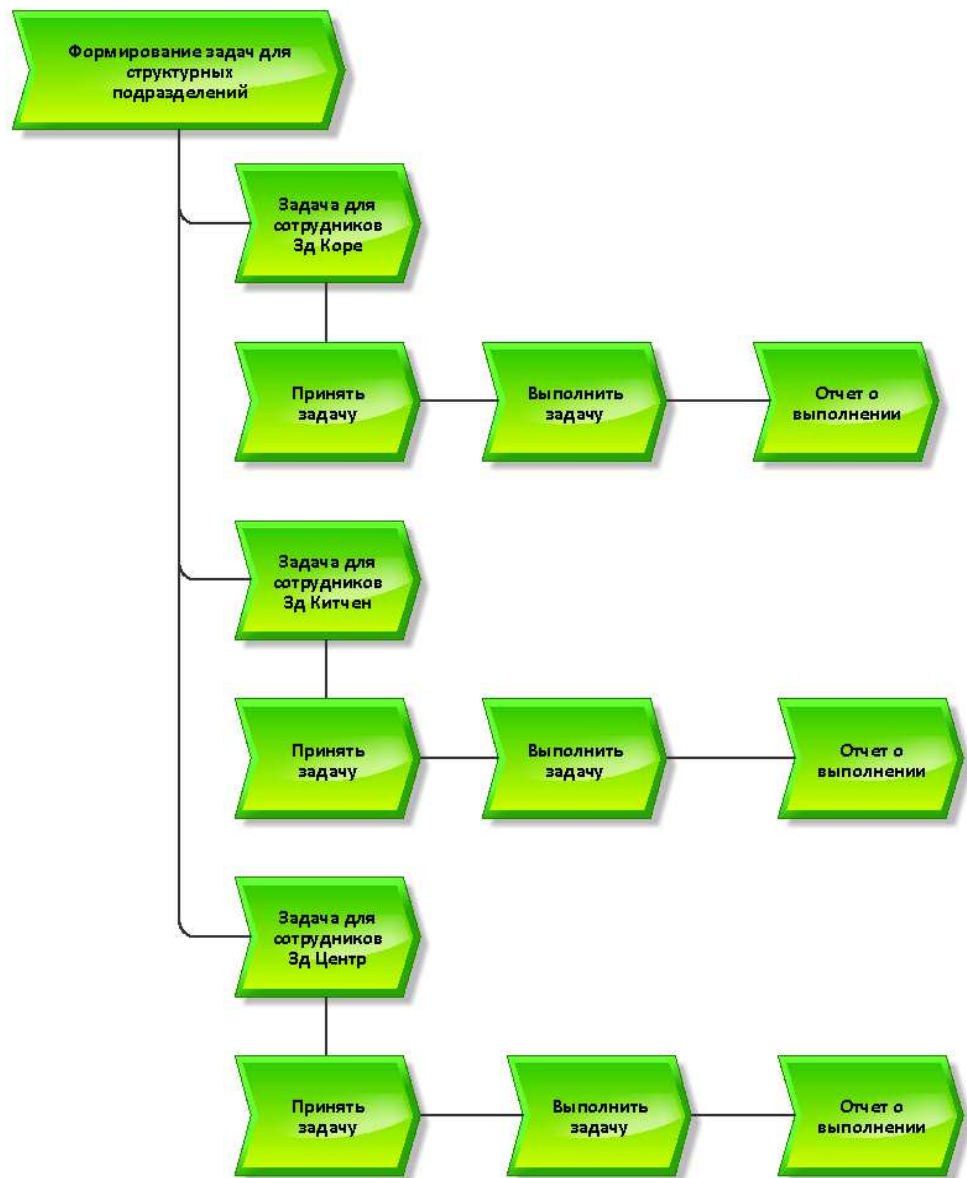


Рисунок 1.2 – VAD диаграмма

Так как формирование задач в компании для всех структурных подразделений и сотрудников похоже, смоделируем диаграмму бизнес-процесса eEPC только для формирования задачи на 3д печать eEPC (рисунок 1.3).

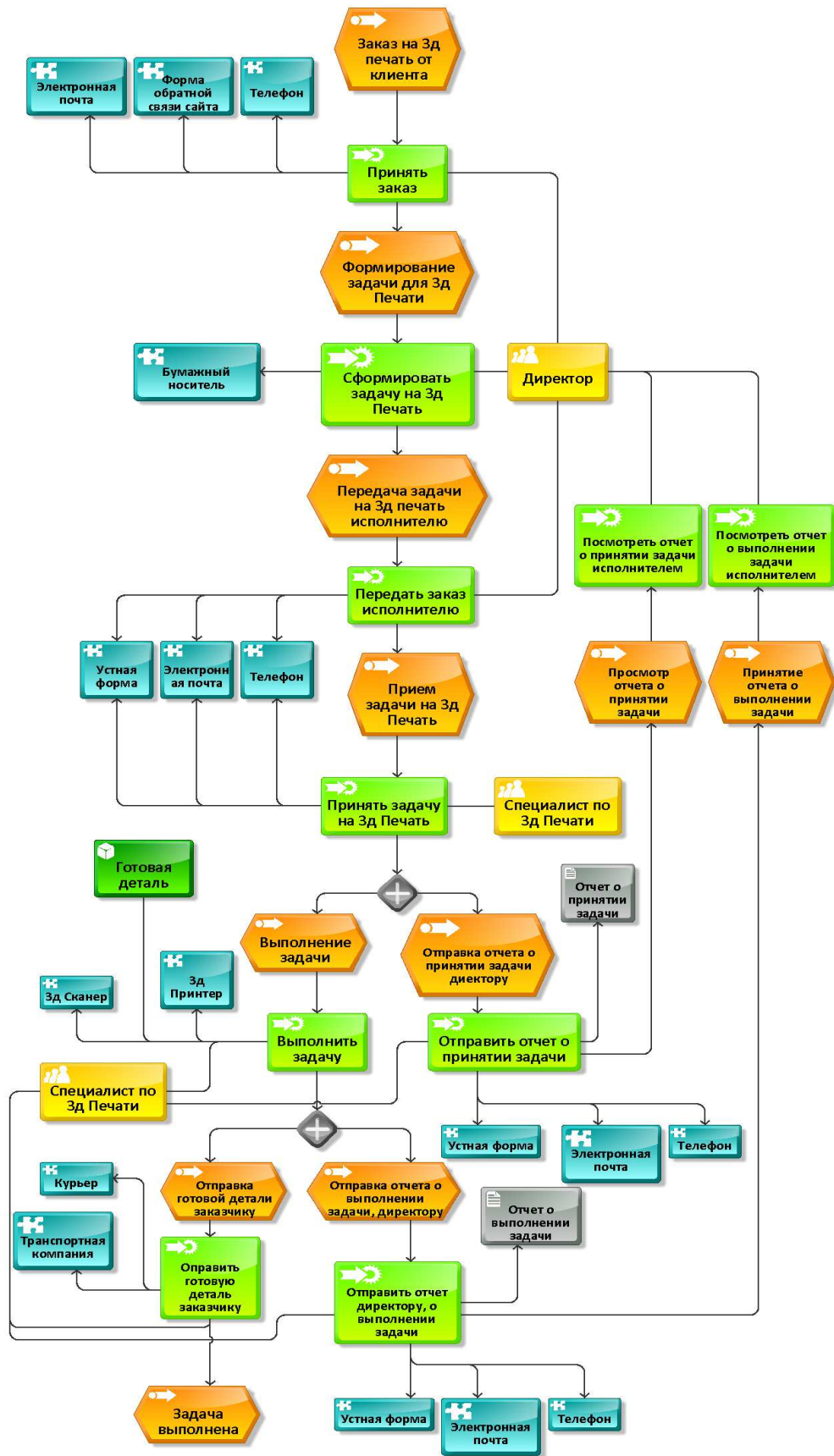


Рисунок 1.3 – eEPC диаграмма

На данный момент в компании отсутствует программное обеспечение автоматизации бизнес-задач. Из диаграммы eEPC видно, что директор получает от клиента заказ, на основе него формирует задачу на бумажном носителе, передает ее исполнителю (специалист по 3д печати) по следующим средствам связи – телефон, электронная почта, – или ставит задачи перед персоналом в устной форме. В свою очередь, исполнители формируют отчеты о статусах выполнения задачи и передают его директору, опять же с помощью телефона, электронной почты или в устной форме. Это приводит к долгому формированию задач, поэтому в процессе деятельности предприятия отмечены «потери» заданий на выполнение задач. Более 50 % задач не выполнялись в срок.

1.2 Обоснование необходимости создания автоматизированной информационной системы

На основании пункта 1.1 можно сделать вывод, что при возросших объёмах поступающих заказов и управленческих задач увеличивается время на их заполнение и формирование.

Директору приходится формировать задачи для исполнителя на бумажном носителе, что делает невозможным долгое хранение в архиве подобных документов. Постановка задачи для персонала производится с помощью телефона, электронной почты или же в устной форме, что приводит к неточному исполнению задачи. Отсутствие контроля за ходом выполнения задач может привести к полному непониманию, чем заняты сотрудники на текущий момент, что, в свою очередь, может привести к неверному планированию рабочего процесса, а это повлечет за собой большое количество провальных проектов, за которые берется компания 3д Коре.

Решить данную проблему можно двумя способами:

- привлечение новых сотрудников;

– автоматизация функций управления и контроля бизнес-задач предприятия.

Первый способ значительно дороже, что делает его непривлекательным в долгосрочной перспективе. К тому же массовые повторяющиеся операции по формированию задач и ведению учета выполнения поставленных задач, легко поддаются формализации и, следовательно, автоматизации.

Таким образом, возникает необходимость разработать АИС, автоматизирующую функции управления задачами данного предприятия.

На рисунке 1.4 можно увидеть диаграмму eEPC после внедрения АИС на предприятии.

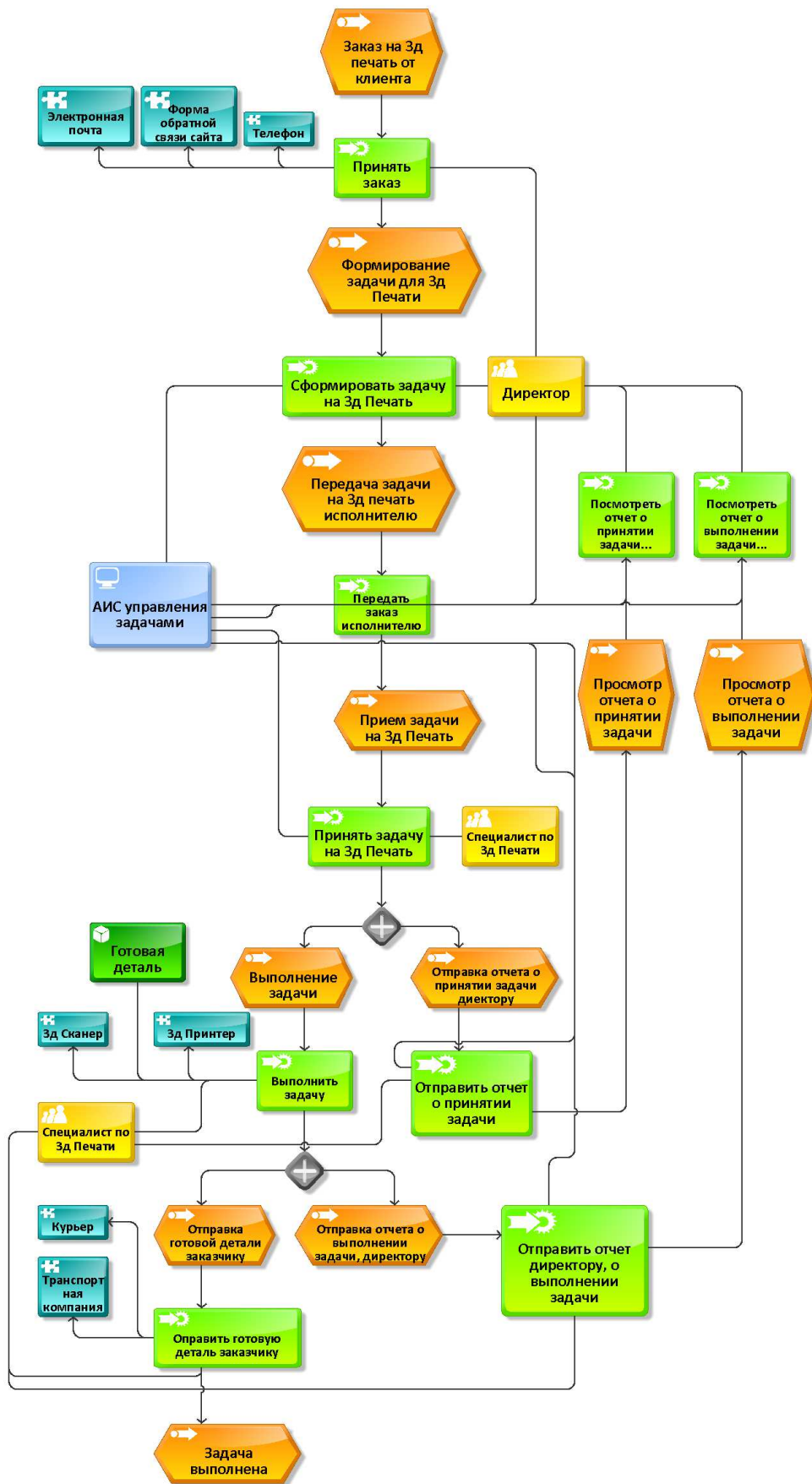


Рисунок 1.4 – eEPC диаграмма после внедрения АИС

Из диаграммы видно, что АИС используется на всем жизненном цикле задачи. С помощью автоматизированной системы директор сможет контролировать весь процесс выполнения задачи. Появится возможность отслеживать, на каком этапе выполнения сейчас находится задача.

Для управления бизнес-задачами было принято решение разработать интернет-приложение на основе канбан-доски. Система канбан была впервые в мире реализована фирмой «Toyota». В 1959 году эта фирма начала эксперименты с системой канбан, и в 1962 году запустила процесс перевода всего производства на этот принцип.

Появление термина канбан связано с перечислением стандартных операций: мастера участков перечисляли выполняемые работы на бумаге и вывешивали их на видном месте рядом с такими же списками мастеров других участков. То есть это была публичная, общедоступная доска, на которую прикрепляли цветные стикеры (карточки) с задачами.

По мере прохождения задач от одного этапа к другому их переносили в соответствующий столбец.

Канбан-доска позволяет визуализировать рабочий процесс, своевременно выявлять проблемы в процессе работы, осуществлять контроль и самоконтроль, вести учет выполненных задач, улучшить взаимодействие в коллективе, вести учет клиентов и др.

Пример простейшей доски включает столбцы: «Сделать», «В работе» и «Сделано» (см рисунок 1.5). В столбцах наклеены стикеры (карточки) с задачами.

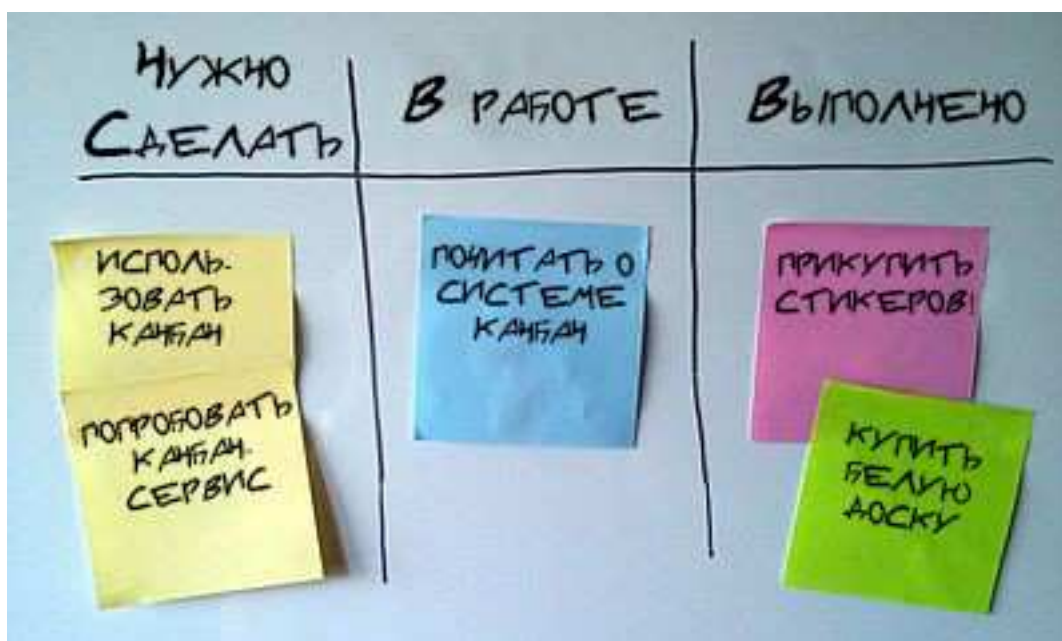


Рисунок 1.5 – Простейший пример канбан-доски

1.3 Требования к разрабатываемой автоматизированной системе

На основании вышеизложенного, к автоматизированной информационной системе были сформированы следующие требования. Система должна обладать такими функциями:

- авторизовать пользователей;
- каждый пользователь должен иметь личный кабинет;
- у каждого пользователя должны быть реализованы уровни доступа к модулям АИС;
- хранить данные всех пользователей;
- отображать канбан-доску с текущими задачами;
- канбан-доска должна состоять из 3ех столбцов «сделать», «в работе», «сделано»;
- у каждой задачи должен быть исполнитель;
- каждая задача должна обладать описанием и возможностью установления срока выполнения;
- все выполненные задачи хранятся в архиве задач;

- реализовать возможность файлового обмена файлами до 300 мб;
- стикер с задачами должен содержать ссылки на комментарии, календарь, профиль ответственного сотрудника и прикрепленный файл;
- система должна формировать отчеты для руководителя по пользователям и задачам, по заданным параметрам.

Требования к надежности системы:

В целях надежности программного обеспечения она должна удовлетворять следующим требованиям:

- разработанная программа должна обладать средствами защиты от ошибочных действий пользователей;
- все ошибки должны отображаться с комментариями или подсказками по их устранению;
- исключить возможность доступа к файлам конфигурации пользователям.

Для повышения надежности необходимо принять следующие меры:

- периодически осуществлять резервное копирование информации;
- регулярно проверять целостность базы данных.

Требования к интерфейсу:

- дизайн пользовательского интерфейса должен адаптироваться под основные разрешения экранов настольных (десктоп), портативных (ноутбук) и планшетных устройств. Максимальное разрешение по горизонтали - 1920px, минимальное - 1024px. В диапазоне указанных разрешений горизонтальная прокрутка не должна появляться;
- интерфейс системы, включая графики и диаграммы, должен адаптироваться под разрешение экрана;
- элементы интерфейса (пункты меню, кнопки, поля ввода в формах, раскрывающиеся списки, и т. д.) должны адаптироваться по размеру под устройство, на котором просматривается система;

– дизайн пользовательского интерфейса должен быть разработан с учетом принципа сохранения работоспособности при потере части функциональности системы;

– дизайн пользовательского интерфейса должен корректно отображаться и работать на настольных компьютерах, ноутбуках, планшетных устройствах;

– интерфейс системы должен поддерживать русский язык;

– интерфейс системы должен обеспечивать наглядное, интуитивно понятное представление структуры размещенной информации, быстрый и логичный переход к соответствующим разделам системы;

– навигационные элементы интерфейса системы должны обеспечивать однозначное понимание пользователем их смысла и обеспечивать навигацию по всем доступным пользователю разделам системы и отображать соответствующую информацию;

– интерфейс системы должен позволять решать задачи пользователя наиболее быстрым, простым и удобным из возможных способов;

– интерфейс системы должен быть рассчитан на пользователей, не имеющих специальных технических знаний и навыков в области компьютерной техники, и быть легко осваиваем ими, желательно без необходимости обращения к руководству пользователя;

– интерфейс должен быть адаптивным, т. е. обеспечивать высокую степень удобства использования не только на широких настольных экранах, но и на портативных и планшетных устройствах;

– в целом дизайн и удобство интерфейса должны быть на уровне ожиданий современного пользователя, имеющего опыт работы с лучшими образцами подобных сервисов, и восприниматься им как комфортная, удобная и приятная рабочая среда.

Требования к аппаратной и программной платформе функционирования системы.

Требования к серверной части.

Для обеспечения работы автоматизированной системы управления задачами требуется следующая минимальная конфигурация:

- система на основе процессора Intel или AMD с частотой 2 GHz и выше;
- не менее 2 Гб оперативной памяти;
- жесткий диск, содержащих не менее 100 Гб свободного пространства;
- сетевой адаптер;
- клавиатура и мышь.

Требования к клиентской части.

Минимальная конфигурация персонального компьютера пользователя для работы в АИС в Интернете такова:

- система на основе процессора Intel или AMD с тактовой частотой 1.8 GHz или выше;
- 1 Гб оперативной памяти;
- свободное пространство на жестком диске не менее 1 Гб;
- сетевой адаптер;
- монитор;
- клавиатура и мышь.

Также в обоих случаях необходимо наличие исправно функционирующей сети Интернет.

1.4 Анализ программных продуктов аналогичного назначения

На сегодняшний день автоматизированные системы с канбан-досками набирают популярность во многих отраслях бизнеса. В связи с этим, на рынке представлено множество решений для организации. Рассмотрим несколько из них:

- Trello созданная американской компанией Fog Creek Software;
- Битрикс24 разработчик российская компания ООО «1с – Битрекс»;
- Taskify разработчик чешская компания Ongoing.

Трелло один из самых популярных в России сервисов канбан-досок. В нем все предельно просто и понятно (рисунок 1.6).

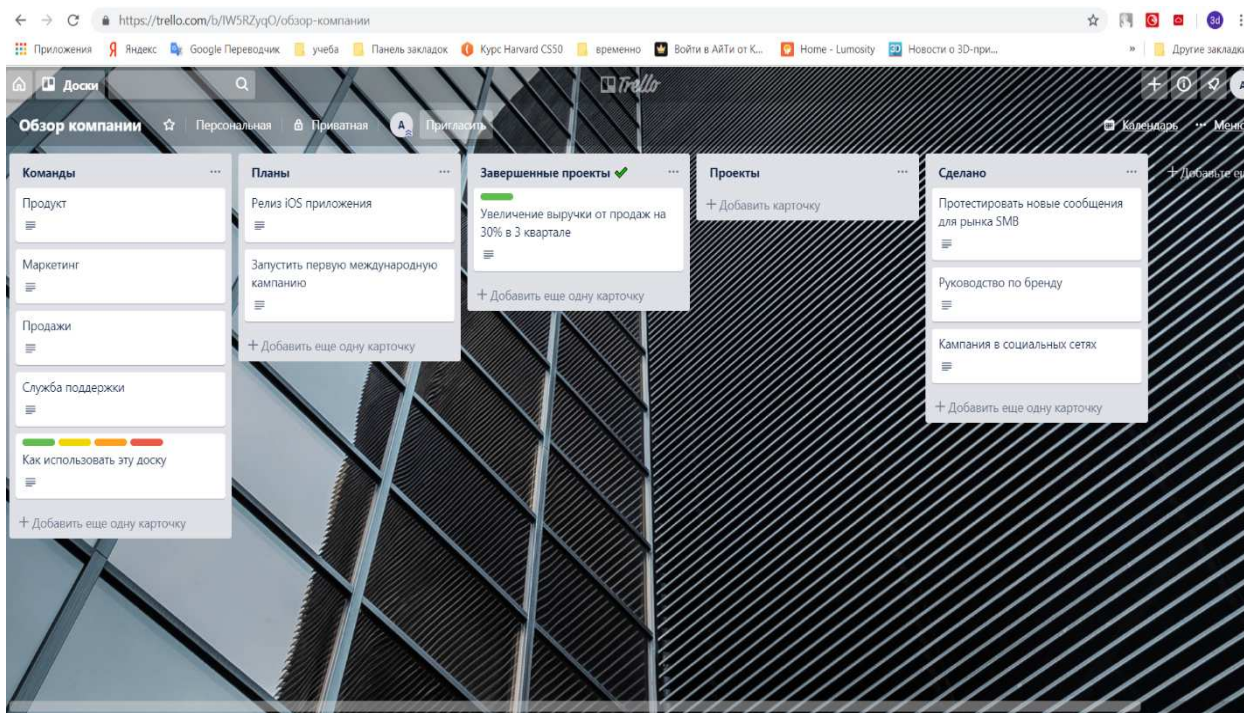


Рисунок 1.6 – Интерфейс сервиса Trello

Можно создать сколько угодно списков (проектов) с карточками. В карточках есть сроки, метки, возможность прикреплять файлы, писать комментарии и прочее. Ограничений на количество пользователей для доски нет. Весь основной канбан-функционал предоставляется бесплатно. Особенностью Trello является красивый интерфейс и множество интеграций, например Slack для корпоративного общения. К сожалению, данная система не позволяет формировать отчеты по выполненным задачам и пользователям. Отсутствует возможность делегировать права на просмотр задач, внутри группы. За дополнительные функции, например, такие как цветные стикеры задач, размер файла вложения больше 10 мб, придётся заплатить.

Битрикс24. Возможностей и инструментов для организации работы в Битрикс 24 огромное количество (рисунок 1.7).

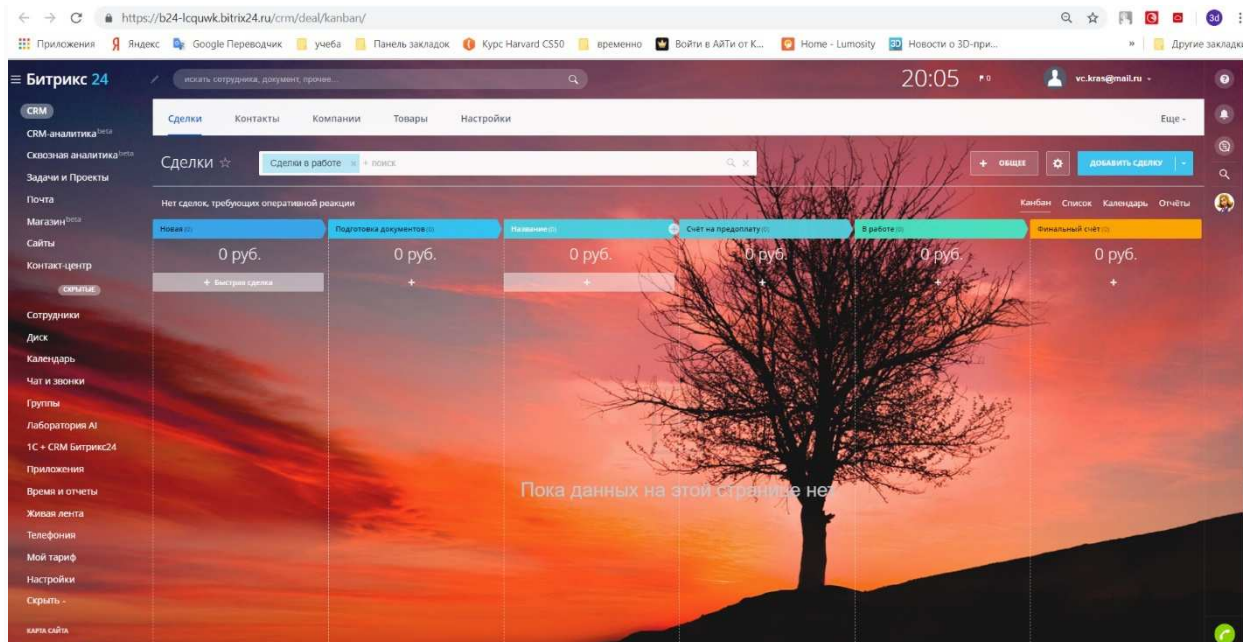


Рисунок 1.7 – Интерфейс сервиса Битрикс24

Канбан это только один вариант представления задач. Другие варианты: традиционные списки, диаграмма Ганта. У канбан-доски есть по умолчанию три колонки: «новые», «выполняются» и «сделаны», между которыми можно произвольно перемещать задачи. Доступно создание множества проектов с разделением сотрудников. Кнопка «Мой План» позволяет сотруднику видеть только свои задачи. Кроме этого, есть много фильтров для удобной навигации по карточкам. Внутри карточки можно:

- Назначить ответственного, наблюдателя, соисполнителей;
- Определить крайний срок, время выполнения, отслеживать сроки по диаграмме Ганта, получать отчеты по работе;
- Приоритезировать задачи, выставляя важность;
- Добавить напоминания, уведомления;
- Прикрепить файлы;
- Писать комментарии;
- Добавлять подзадачи и прочее.

Битрикс24 – это не только задачи, это CRM-система, корпоративный чат, огромный каталог интеграций, API и так далее. Если говорить исключительно о канбане то у него одна из самых функциональных досок. 12 сотрудников могут использовать ее, в рамках бесплатного тарифа. Самый главным минусом данной системы является ее многофункциональность, в системе слишком много излишнего функционала, который не требуется для разрабатываемой АИС. Также к недостаткам можно отнести платные функции по формированию отчетов для руководителя, распределению прав доступа к доскам и задачам.

Taskify – полностью бесплатный сервис, со скромным набором функций и встроенной рекламой (рисунок 1.8).

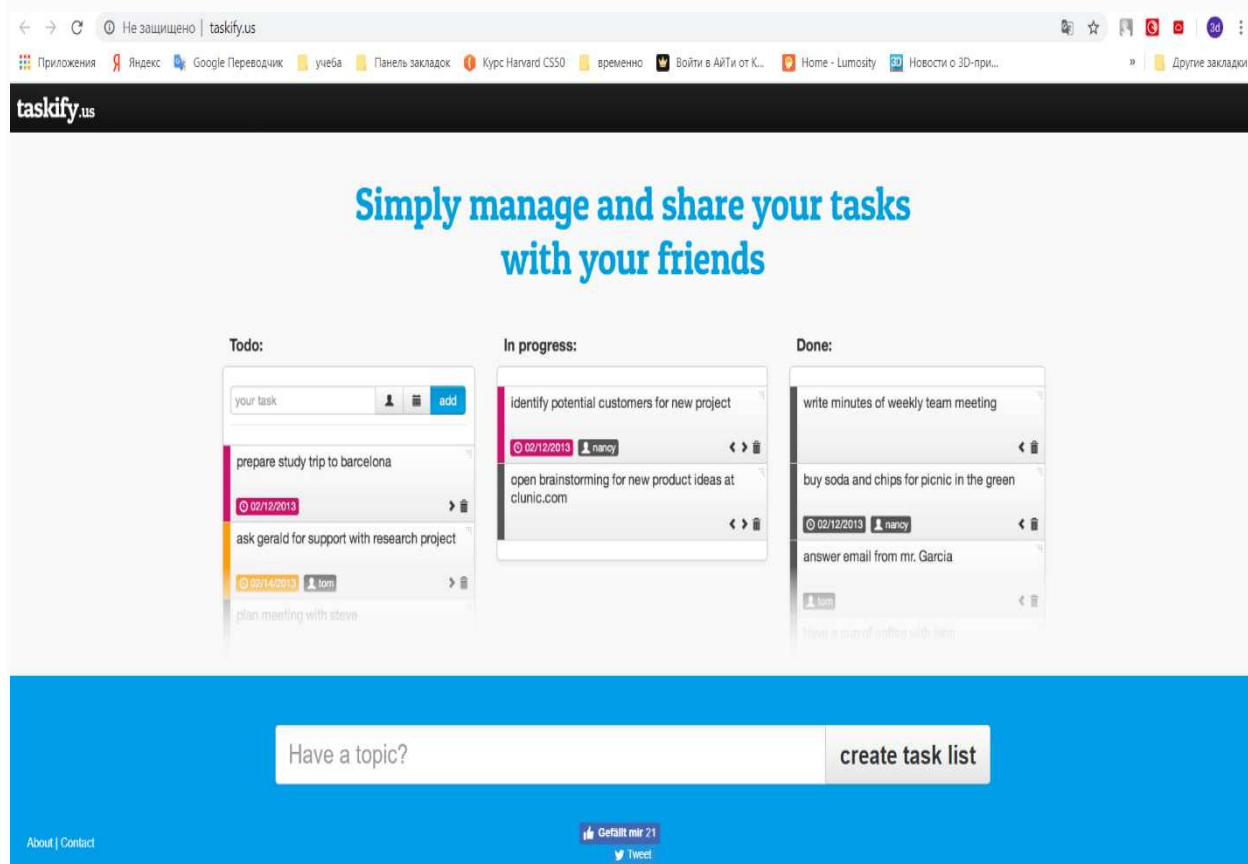


Рисунок 1.8 – Интерфейс сервиса Taskify

Есть стандартные три колонки с перетаскиванием карточек. В карточках указывается только исполнитель и крайний срок выполнения задачи. Цветом

определяется срочность задач. Можно подключить пользователей с помощью ссылки.

Сервис доступен только на английском языке, задачи можно писать на любом языке. Очень ограниченный сервис, не хватает множество функций.

Проанализировав вышеперечисленные приложения, заказчик пришел к выводу, что ни одно из них не подходит для решения задач компании «Зд Коре», поэтому необходимо разрабатывать собственную автоматизированную систему управления задачами.

1.5 Выбор средств разработки

1.5.1 Выбор системы управления базами данных

База данных – это некоторый набор данных, который хранятся по определенным правилам. На сегодняшний день чаще всего применяют реляционную модель данных. Данная модель предусматривает хранение данных в таблицах, которые соединяются между собой некоторыми связями.

Для работы с базами данных используются системы управления базами данных. Система управления базами данных (СУБД) – это комплекс программ, который позволяет выполнять различные операции с базой данных, например создание, редактирование, удаление таблиц.

Рынок СУБД довольно обширный. Присутствуют как платные, так и бесплатные продукты. Каждая из этих систем реализует основные механизмы управления базами данных, а также предлагает свои уникальные функции для работы с данными.

Для сравнения я выбрал несколько популярных СУБД. Обоснование выбора для использования конкретного программного продукта будет основано на анализе функций предоставляемого функционала. Также немаловажным фактором будет являться стоимость СУБД.

Рассмотрим следующие продукты.

MySQL.

На сегодняшний день является одной из самых распространённых СУБД. Как правило, используется при разработке веб-сайтов и других веб-приложений. Система является бесплатной и открытой, что является одним из значимых факторов ее популярности. Активное сообщество пользователей постоянно улучшает ее, разрабатывая собственные плагины, добавляя новый функционал.

Как правило большинство современных СУБД для своей работы использует язык SQL - structured query language. Но MySQL использует его не в чистом виде, то есть некоторый функционал языка отсутствует.

К преимуществам данной системы можно отнести:

- СУБД легко установить. Как правило, она идет в комплекте с веб-сервером, например как LAMP(Linux,Apache,MySQL,PHP). Программный продукт может управлять через консоль или графический интерфейс;
- СУБД практически полностью поддерживает язык SQL;
- MySQL может достаточно просто работать с большими объемами данных и при необходимости легко масштабироваться;
- Некоторые упрощения некоторых стандартов языка SQL позволяют ускорить работу СУБД;

К недостаткам использования MySQL можно отнести:

- Неполное соответствие стандартам языка SQL;
- Проблемы с надежностью. MySQL из-за некоторых методов обработки информации в некоторых ситуациях уступает другим СУБД. Например: при кластерной обработке MySQL может организовать DDos атаку на собственную базу.

Postgre SQL.

Из числа всех бесплатных СУБД Postgre SQL является самым профессиональным решением. Postgre SQL – бесплатная и открытая система.

Но в отличие от MySQL данная СУБД старается полностью соответствовать стандартам языка SQL.

К достоинствам данной системы можно отнести:

- Postgre бесплатная система с открытым исходным кодом, которая соответствует стандартам SQL;
- Достаточно большое количество дополнений, которые позволяют расширять функционал системы;
- В отличие от других СУБД она обеспечивает более высокий уровень целостности данных и транзакций;
- Поддерживает формат json.

К недостаткам можно отнести:

- При выполнении достаточно простых операций эта СУБД использует значительно больше ресурсов, чем ее конкуренты;
- Несмотря на достаточно большую базу пользователей, трудно найти хостинг, который будет поддерживать Postgre.

Microsoft SQL server.

Microsoft SQL server – СУБД, разработанная компанией Microsoft. Так же как MySQL и Postgre эта СУБД предназначена для управления реляционными базами данных. Microsoft SQL server использует язык Transact-SQL. Данный язык соответствует стандартам языка SQL. Данная СУБД может использоваться как с маленькими базами предприятия, так и с базами огромных компаний.

К преимуществам СУБД можно отнести:

- Шифрование всей хранимой информации в БД. Шифрование и расшифровка информации происходит непосредственно перед чтением или записью информации;
- Интуитивно понятный интерфейс;
- Высокая производительность СУБД;

- Легкая масштабируемость системы.

К недостаткам можно отнести:

- Ограниченный выбор платформ. СУБД будет работать только под управлением операционных систем семейства windows;

- Ограниченное число средств для разработки клиентских приложений.

По умолчанию количество платформ для разработки клиентской части ограничено. Другие СУБД поддерживают большее количество платформ;

- Microsoft SQL server является не бесплатным ПО. SQL server имеет несколько редакций, которые отличаются предоставляемым функционалом и стоимостью.

Выше рассмотрены одни из самых популярных СУБД. Рынок данных систем достаточно обширен и не ограничивается рассматриваемыми СУБД. Главным вопросом является: какую СУБД использовать?

При использовании PostgreSQL СУБД в перспективе значительно проще перейти на другое решение. Также идеально подходит для организаций с ограниченным бюджетом. Поэтому для разработки приложения я выбрал PostgreSQL СУБД.

1.5.2 Выбор среды программирования

Для разработки АИС было принято решение использовать язык программирования Java.

Он обладает своими особенностями и преимуществами, среди которых выделяют такие:

- Возможность более легкого взаимодействия с памятью устройств;

- Способность решать нестандартные ситуации;

- Хорошие способности к фильтрации событий и информации;

- Большой набор стандартных возможностей;

- С помощью Java можно создавать функциональные сетевые приложения.

В целом Java отлично подходит для веб-разработки, так как хорошо справляется не только с указанными задачами, но и легко взаимодействует с базами данных, имеет отличные возможности функционального программирования, предоставляет возможность работать в многопоточном режиме и многое другое.

Каждому программисту нужна IDE (integrated development environment), которая будет делать за него всю грязную работу во время написания кода. IDE должна обладать разнообразными средствами интеграции с библиотеками классов и фреймворками.

Решение в пользу той или другой IDE должно основываться на том, что мы разрабатываем, какую роль играем в команде разработчиков, какую методологию разработки применяем и каков наш уровень, как программиста. Кроме того, нельзя исключать такие факторы, как стандарты, принятые в команде разработчиков и наши персональные предпочтения.

Есть 3 наиболее популярных IDE, которые чаще всего применяются для разработки серверных приложений на Java (так называемый, бэкенд) – это IntelliJ IDEA, Eclipse и NetBeans.

Рассмотрим по порядку каждую из них.

IntelliJ IDEA. Быстрый и надежный инструмент для веб-разработки на Java. Эту среду разработки для Java под Windows развивает компания JetBrains. Она постоянно обновляет и дополняет ее, чтобы она соответствовал потребностям рынка.

IntelliJ IDEA поставляется в двух редакциях. Бесплатная версия предназначена для студентов, преподавателей и разработчиков плагинов. Платная - для организаций, разрабатывающих коммерческие приложения.

IntelliJ IDEA является одной из самых функциональных сред для java разработки и оснащена системой интеллектуальной помощи в написании кода. Исходя из контекста, IDEA настраивает работу автодополнения и доступность инструментов. Обилие инструментов позволяет ускорить

разработку, например, с помощью шаблонов и повторений, а также увеличить производительность конечной программы. Огромное количество плагинов и надстроек под любую задачу делают среду java разработки IDEA почти идеальным инструментом.

Eclipse – это современная, кросс-платформенная и свободно распространяемая IDE с открытым исходным кодом, доступная для корпоративной веб-разработки. В 1998 году IBM Software Group решила создать IDE, которая смогла бы закрепиться на рынке и стать одним из его лидеров. Это привело к созданию Eclipse.

Интересно, что Eclipse сама по себе является примером лучших Java-приложений, поскольку написана на Java. Она всегда была стабильной полнофункциональной платформой высочайшего качества для разработки современных веб-приложений.

Eclipse обладает всеми необходимыми функциями, которые должна иметь идеальная Java IDE:

- Поддержка Java 8.0 и 9.0;
- Рефакторинг кода, редактирование кода с проверкой запроса, инкрементная компиляция, перекрестные ссылки, автоматическое предложение вариантов кода;
- Интегрированный статический анализ кода;
- Интеллектуальное завершение кода и быстрое исправление;
- Удобство и производительность;
- Поддержка Windows / Linux / Mac OS X.

NetBeans созданная разработчиками для разработчиков с целью сделать рабочий процесс максимально простым и эффективным.

С помощью NetBeans можно создавать не только корпоративные веб-приложения, но и мобильное программное обеспечение, приложения для ПК. Кроме этого, NetBeans позволяет повторно использовать рабочее

пространство проекта на нескольких платформах, таких как Windows, Linux, Mac OS X и Solaris от Oracle.

NetBeans решает большинство проблем, с которыми разработчик может столкнуться в процессе работы. Эта IDE позволяет управлять всеми задачами: анализом, проектированием, отладкой, модульным тестированием, управлением исходным кодом и развертыванием.

В разрабатываемом коде могут быть не выявленные ошибки, которые трудно найти. Но встроенные инструменты NetBeans, такие как анализ статического кода, интеграция плагина FindBug помогут выявить и устранить сложные проблемы в коде. Кроме этого, отладчик NetBeans предоставляет возможность быстро перемещаться по коду, устанавливать контрольные точки, добавлять заметки и отслеживать исполнение кода.

Также IDE поставляется с мощным профайлером кода, который помогает оптимизировать производительность приложения (скорость работы и потребление памяти). Кроме этого, она включает в себя визуальный отладчик, позволяющий корректировать создаваемые пользовательские интерфейсы, без необходимости правки кода. Стоит уделить внимание инструменту Maven, поддерживаемому NetBeans. С его помощью добавление зависимостей производится всего в несколько кликов. Это позволяет избежать лишней загрузки или перестроения индексов.

Давайте посмотрим, за что еще NetBeans считается одной из лучших Java IDE.

Она поддерживает HTML5, CSS3, JavaScript (ES 5/6) и Angular JS. NetBeans также поддерживает автоматическое завершение для этих языков, а также директивы Angular.

NetBeans поддерживает все популярные веб-приложения (Spring / Struts / Wicket) и библиотеки, такие как PrimeFaces, RichFaces, ICEfaces.

Из рассмотренных выше IDE для разработки, я выбрал IntelliJ IDEA.

По причине того, что среда разработки Eclipse очень популярна среди разработчиков Java, в ней легко запутаться из-за множества ракурсов и вкладок, переключение между которыми может быть не всегда удобно и очевидно. Кроме того, Eclipse обладает самым богатым набором плагинов среди всех IDE, что может привести к проблемам их несовместимости между собой.

По сравнению с NetBeans, IntelliJ IDEA – это то же самое, только намного лучше. Он работает быстро, никогда не тормозит и активно развивается. Благодаря тому, что компания JetBrains (производители ПО) подходят к своей IDE, как к товару, они пытаются максимально его улучшить и оптимизировать.

На высоком уровне реализовано и форматирование кода для лучшей читаемости другими разработчиками. IDEA использует автоматическое форматирование, среда разработки позволяет сконцентрироваться на действительно важных вещах. Еще одним плюсом для меня стала возможность интеграции с Git и другими системами контроля версий, встроенный мощнейший отладчик, встроенное определение степени покрытия кода тестами, построение диаграмм классов и переходы по иерархиям вверх и вниз, встроенный декомпилятор байткода Java и др.

2 Разработка автоматизированной системы.

2.1 Концептуальное проектирование базы данных

Важнейшим компонентом любой информационной системы является база данных. База данных (Data Base) – структурированный, организованный набор данных, объединенный в соответствии с некоторой выбранной моделью и описывающий характеристики какой-либо физической или виртуальной системы.

При проектировании базы данных любая рассматриваемая предметная область имеет свои особенности. Но в тоже время обладает и общими для всех предметных областей элементами. Так, в основной массе решаемых задач автоматизации обязательно фигурируют такие классы объектов, как предприятие (организация), структурная единица предприятия (цех, отдел, факультет, отделение и т. п.), люди, или физические лица, разного рода материальные объекты. Все процессы, происходящие в предметной области и которые необходимо учитывать в базе данных, осуществляются на основе документов, которые в свою очередь фиксируют сбор, перемещение, расход каких-либо данных. Таким образом, многие ситуации можно смоделировать, применяя существующие шаблоны.

При разработке стандартной схемы организации был определен следующий персонал, который включает директора и сотрудников подразделений. Выделим следующие этапы взаимодействия сотрудников и директора: регистрация сотрудника в системе, постановка задачи, назначение ответственного за выполнение задачи, отчет о выполнении задачи, так как отчетом о выполнении задачи в разрабатываемой системе является перемещение задачи между столбцами канбан-доски, будем рассматривать отчет о выполнении задачи как канбан-доску.

После изучения предметной области и анализа структуры системы были определены объекты. Список сущностей и связей представлен в таблице 1.

Таблица 1 – Сущности предметной области

Название	Ключ сущности	Атрибут
Руководитель	Id_Руководителя	ФИО Телефон Логин Пароль e-mail Фото
Сотрудник	Id_Сотрудника	ФИО Телефон e-mail логин пароль дата рождения должность структурное подразделение задача Фото
Задача	Id_Задачи	Дата создания Дата завершения Сотрудник Вложения Описание Комментарии Структурное подразделение
Канбан-доска	Id_доски	Статус выполнения задачи Задача
Структурное подразделение	Id_структурного подразделения	Сотрудники Задачи

Исходя из имеющихся данных, построим концептуальную модель базы данных (рисунок 2.1), необходимую для дальнейшего проектирования приложения.

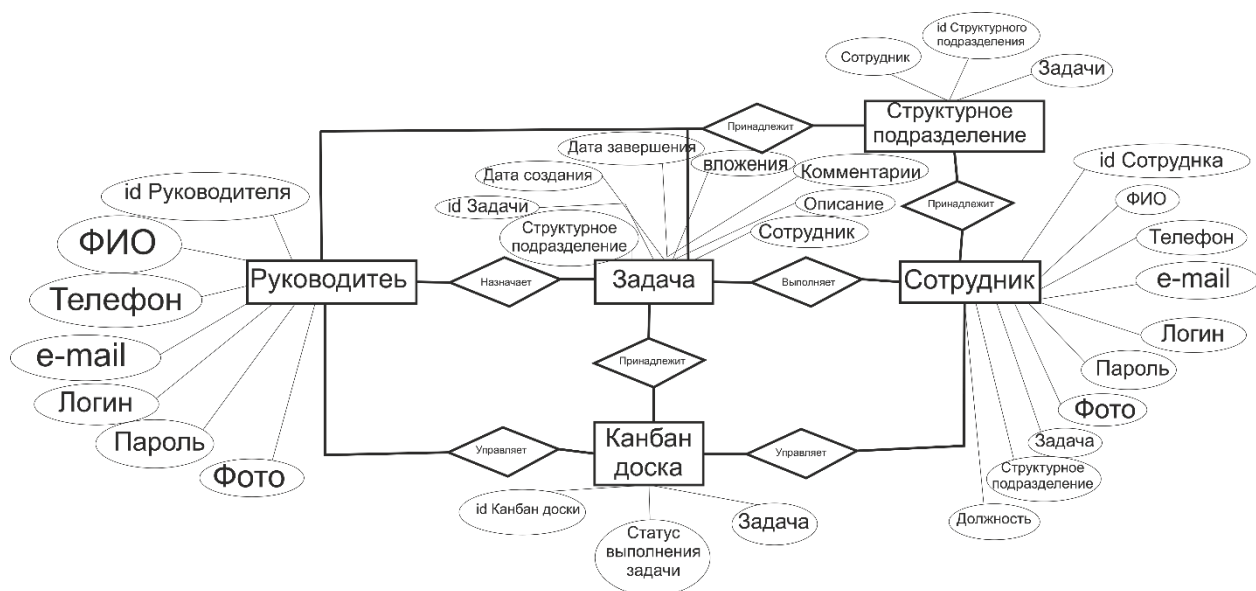


Рисунок 2.1 – Концептуальная модель базы данных

Следующим шагом проектирования является создание логической структуры реляционной базы данных.

2.2 Логическое проектирование базы данных

Каждый информационный объект модели данных отображается с соответствующей реляционной таблицей. Структура реляционной таблицы определяется требуемым составом соответствующего информационного объекта, где каждый столбец (поле) соответствует одному из реквизитов объекта. Ключевые реквизиты объекта образуют уникальный ключ реляционной таблицы. Для каждого столбца требуется указать формат и размер данных. Строки (записи) таблицы соответствуют экземплярам объекта и генерируются при загрузке таблицы. На рисунке 2.2 приведена логическая структура базы данных.

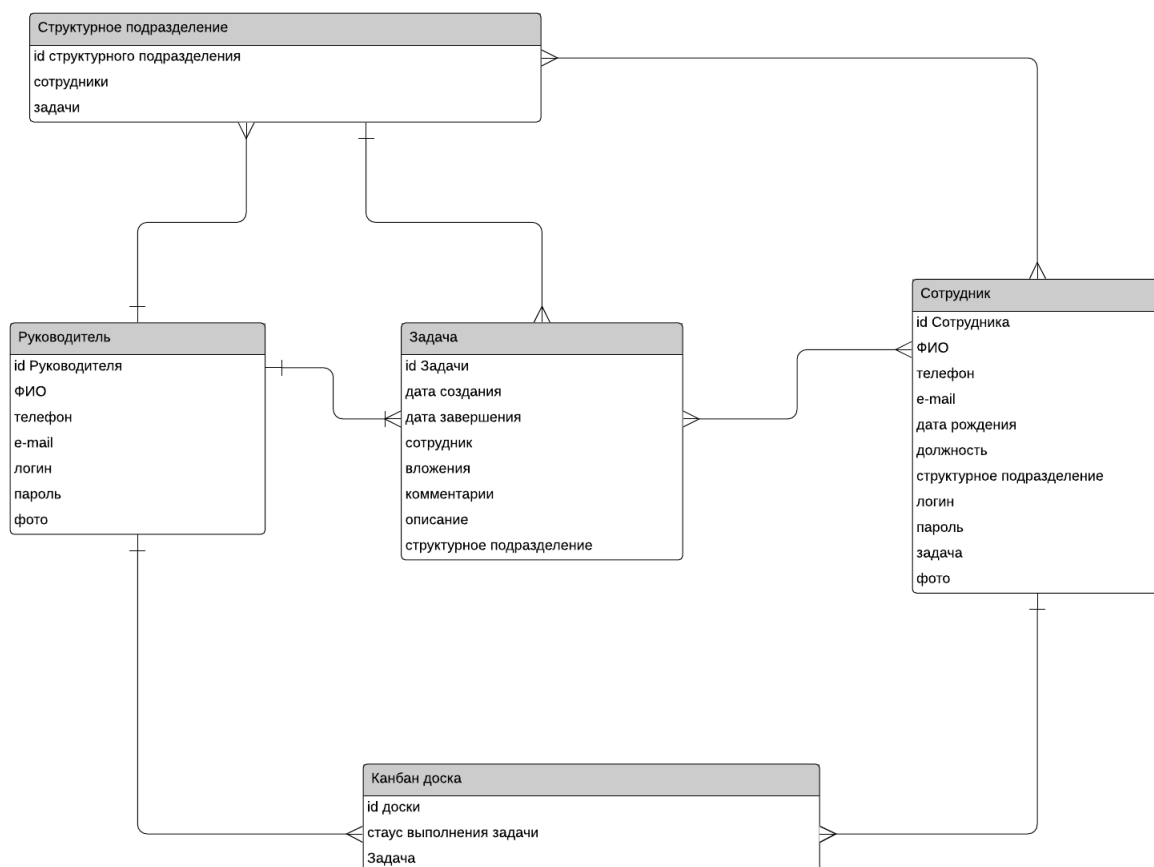


Рисунок 2.2 – Логическая структура базы данных

Связи между объектами модели данных реализуются теми же реквизитами, ключами связи в соответствующих таблицах. Ключом соединения всегда является уникальный ключ главной таблицы. Ключ в подчиненной таблице это либо часть уникального ключа в нем, либо поле, которое не является частью первичного ключа. Ключ связи в подчиненной таблице называется внешним ключом. Определение отношений «один ко многим» в этой схеме должно соответствовать построенной модели данных. Появление схемы данных практически совпадает с графическим представлением информационно-логической модели. В таблицах 2, 3, 4, 5 и 6 показаны структуры объектов «Руководитель», «Задача», «Сотрудник», «Структурное подразделение», «Канбан-доска», соответственно.

Таблица 2 – Структура таблицы «Руководитель»

Содержание поля	Название поля	Тип поля	Примечание
Идентификатор	Id_director	Serial	Первичный ключ
Фамилия	SecondName	text	Обязательное поле
Имя	FirstName	text	Обязательное поле
Отчество	Patronymic	text	Обязательное поле
Телефон	Phone	integer	–
e-mail	Mail	Variable charter	Обязательное поле
Логин	Login	Variable charter	Обязательное поле
Пароль	Pass	Variable charter	Обязательное поле
Фото	Photo	Variable charter	–

Таблица 3 – Структура таблицы «Задача»

Содержание поля	Название поля	Тип поля	Примечание
Идентификатор	Id_task	Serial	Первичный ключ
Дата создания	Start_date	Date	–
Дата завершения	Finish_date	Date	–
Сотрудник	Id_member	integer	Внешний ключ к «сотрудник»
Вложения	faile	Variable charter	–
Комментарии	comment	Text	–
Описание	Description	Text	Обязательное поле
Структурное подразделение	Id_group	integer	Внешний ключ к «структурное подразделение»

Таблица 4 – Структура таблицы «Сотрудник»

Содержание поля	Название поля	Тип поля	Примечание
Идентификатор	Id_member	Serial	Первичный ключ
Фамилия	SecondName	text	Обязательное поле
Содержание поля	Название поля	Тип поля	Примечание
Имя	FirstName	text	Обязательное поле

Отчество	Patronymic	text	Обязательное поле
Телефон	Phone	integer	–
e-mail	Mail	Variable charter	Обязательное поле
Логин	Login	Variable charter	Обязательное поле
Пароль	Pass	Variable charter	Обязательное поле
Фото	Photo	Variable charter	–
Дата рождения	Birthday	Integer	–
Структурное подразделение	Id_group	Variable charter	Внешний ключ к «структурное подразделение»
Должность	Position	Text	Обязательное поле
Задача	Id_task	Variable charter	Внешний ключ к «задача»

Таблица 5 – Структура таблицы «Структурное подразделение»

Ключевое поле	Название поля	Тип поля	Примечание
Индентификатор	Id_group	Serial	Первичный ключ
Сотрудник	Id_member	Integer	Внешний ключ к «Сотрудник»
Задача	Id_task	Integer	Внешний ключ к «задача»

Таблица 6 – Структура таблицы «Канбан-доска»

Ключевое поле	Название поля	Тип поля	Примечание
Идентификатор	Id_board	Serial	Первичный ключ
Статус выполнения задачи	Task_status	Integer	Обязательное поле
Задача	Id_task	Числовой	Внешний ключ к «задача»

2.3 Проектирование физической модели базы данных

Физическая модель базы данных содержит все детали, необходимые для создания базы данных: наименования таблиц и столбцов, типы данных, определения первичных и внешних ключей и т. п. Данные для проектирования физической модели возьмем из таблиц 2, 3, 4, 5, 6.

Физическая модель строится на основе логической. Имена сущностей и полей (таблиц и столбцов) должны быть записаны латинскими буквами, без пробелов, начиная с буквы.

Для каждого атрибута необходимо определить тип данных его значений.

Реляционная модель данных требует преобразования связей «многие ко многим» и иерархии наследования и допускает их только на уровне логической модели базы данных. На основе вышеизложенного построена физическая модель базы данных (рисунок 2.3).

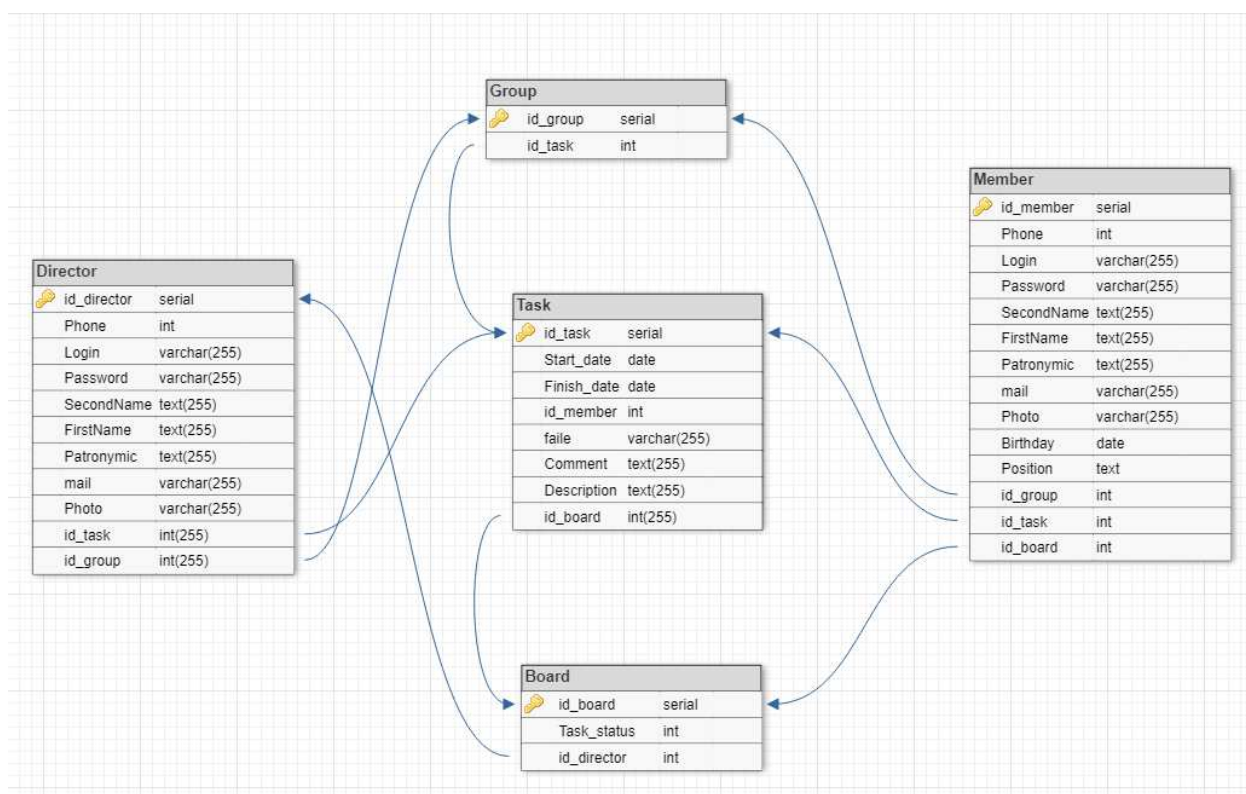


Рисунок 2.3 – Физическая модель базы данных

2.4 Проектирование программного обеспечения

2.4.1 Диаграмма прецедентов

При проектировании автоматизированной системы, для формализации функциональных требований использована UML-диаграмма прецедентов.

Суть данной диаграммы состоит в следующем: проектируемая система представляется в виде множества актеров, взаимодействующих с системой с помощью так называемых вариантов использования. При этом актером называется любой объект, субъект или система, взаимодействующая с проектируемой системой извне. В свою очередь вариант использования – это спецификация сервисов (функций), которые система предоставляет актеру. Другими словами, каждый вариант использования определяет некоторый набор действий, совершаемых системой при взаимодействии с актером. При этом в модели никак не отражается то, каким образом будет реализован этот набор действий. UML диаграмма прецедентов представлена на рисунке 2.4.

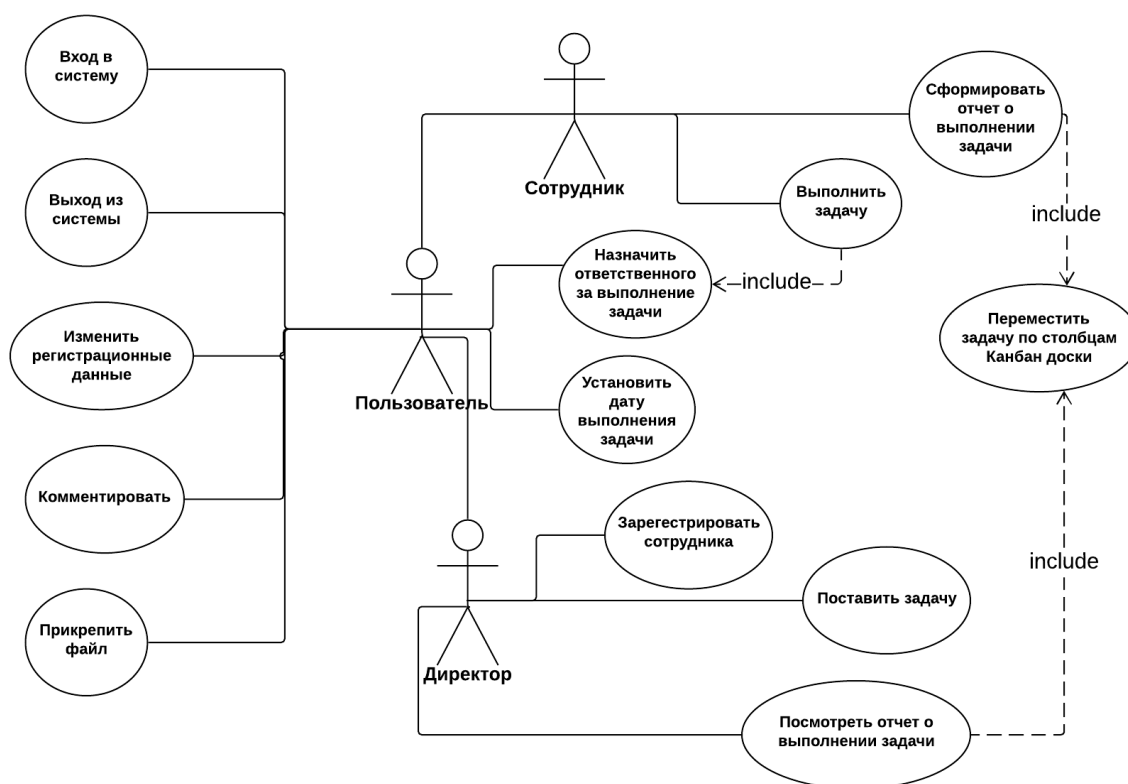


Рисунок 2.4 – Диаграмма прецедентов

Основываясь на диаграмме прецедентов, представленной выше, выделим основные варианты использования:

- Вход, выход из системы;
- Регистрация сотрудников в автоматизированной системе;
- Изменение регистрационных данных. Возможность изменить телефон, e-mail, должность, принадлежность сотрудника к структурному подразделению;
- Добавление комментариев. Сотрудник и директор имеет возможность оставлять комментарии к текущим и выполненным задачам;
- Прикрепить файл, если задача требует наличия дополнительной информации, например, макетом или результатом выполненной задачи является информация, представленная в файле (документ, макет, исполняемая программа и так далее);
- Постановка задачи. Директор формирует задачу и оформляет ее в системе;
- Назначение ответственного за выполнения задачи. Если сотрудники самостоятельно не выбрали задачу для исполнения, директор сам назначает ответственного сотрудника;
- Установка даты выполнения задачи. Любой пользователь системы (директор или сотрудник), может назначить дату завершения выполнения задачи;
- Формирование и просмотр отчета о выполнении задачи. Данная функция имеет визуальный характер. Отчет представлен в виде миграции задачи по столбцам канбан-доски.

2.4.2 Структура автоматизированной системы

Следующим шагом было спроектирована структура веб-приложения автоматизированной системы управления задачами, рисунок 2.5.



Рисунок 2.5 – Структура веб-приложения

Структура сайта – это схема размещения его основных разделов и страниц относительно друг друга. Это план-схема, которая показывает, каким образом строится сайт, логическая связка его страниц.

На структуре можно выделить следующие страницы.

«Главная страница». Страница, которую видит авторизованный пользователь, содержит информацию о последних действиях пользователя, а также ссылки и меню на другие страницы.

Страница «группа», на этой странице приставлен список всех созданных групп (структурных подразделений) на сайте.

«Создать группу», используя форму, на данной странице директор создает новую группу (структурное подразделение).

При переходе на страницу «пользователи», отображается список всех зарегистрированных пользователей, а также с этой страницы можно перейти на страницу «создание пользователя».

На главной странице располагается ссылка на страницу с «профилем пользователя», где можно редактировать свои данные.

Страница «добавить задачу», на этой странице директор создает задачу.

«Календарь», на этой странице представлен календарь. На календаре отображаются задачи, у которых есть срок выполнения.

При помощи страницы «поиск», пользователь может выполнить поиск, по заданным параметрам, задач или пользователей.

На странице Страница «доска с задачами», размещается канбан-доска.

2.4.3 UML диаграмма активности

В качестве графического представления для выделения основных функций системы была применена диаграмма вариантов использования (use case). Диаграмма вариантов использования дает представление что должна делать система. На вопрос «как система должна работать?» отвечает диаграмма активностей

То есть если варианты использования ставят перед системой цель, то диаграмма активности показывает последовательность действий, необходимых для ее достижения. Действия (action) – это элементарные шаги, которые не предполагают дальнейшую декомпозицию.

Деятельность может содержать входящие и/или исходящие дуги деятельности, показывающие потоки управления и потоки данных. Если поток соединяет две деятельности, он является потоком управления. Если поток заканчивается объектом, он является потоком данных.

Деятельность выполняется, только тогда, когда готовы все «входы», после выполнения, деятельность передает управление и(или) данные на свои «выходы». Саму диаграмму деятельности принято располагать таким образом, чтобы действия следовали слева направо или сверху вниз. На рисунке 2.6 представлена UML диаграмма активностей.

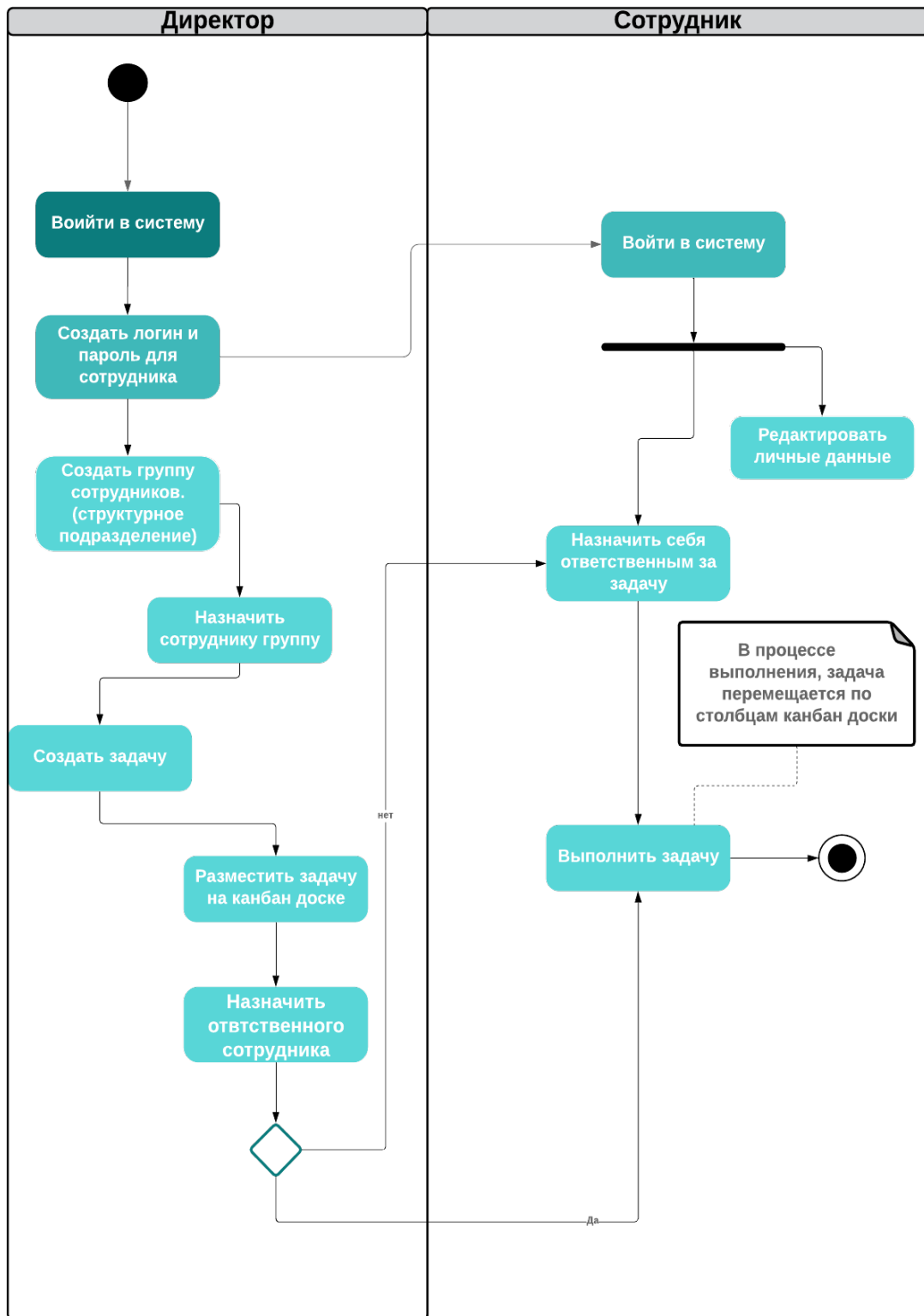


Рисунок 2.6 – UML диаграмма активностей

2.5 Руководство программиста

Назначение системы.

Веб-приложение «Автоматизированная система управления задачами» разработано для компании «3Д коре». Разработанная система позволяет управлять задачами в компании. Данной системой могут воспользоваться только зарегистрированные пользователи. Регистрацию осуществляет директор компании.

Аппаратные и программные требования к платформе функционирования системы.

Минимальные требования к серверной части: процессор (Intel/AMD совместимый x32 или x64) с тактовой частотой 2ГГц и выше, ОЗУ 4 Гб. 5 Гб доступного пространства на жестком диске.

Требования к каналам связи:

Канал связи должен поддерживать протоколы передачи данных: HTTP / HTTPS. Необходимая ширина канала сервера веб-приложения: минимальная: 3 Мбит/сек; рекомендуемая: 8 Мбит/сек.

Операционная система.

Поддерживаются следующие версии ОС семейства Windows:

- Windows 7 Service Pack 1;
- Windows 8;
- Windows 8.1;
- Windows 10;
- Windows Server 2008 R2 Service Pack 1;
- Windows Server 2008 Service Pack 2;
- Windows Server 2012;
- Windows Server 2012 R2;

А также версии ОС семейства Linux:

- Debian;

- Ubuntu;
- Red Hat Enterprise Linux;
- Astra Linux.

Процесс установки системы.

Для создания серверной части использовался язык Java v 1.8. с фреймворком Spring, Java веб-сервер Apache Tomcat 8.5. Для создания клиентской части CSS и JavaScript и jQuery. База данных PostgreSQL 9.5.2.

На локальном или выделенном сервере установить операционную систему. На операционную систему необходимо установить следующее программное обеспечение:

- Apache Tomcat 8.5;
- Установка Java 1.8;
- Создание выделенного пользователя tomcat;
- Загрузка дистрибутива Apache Tomcat 8.5;
- Установка прав доступа на ресурсы веб-сервера;
- Настройка параметров автозапуска веб-сервера;
- Запуск и тестирование Apache Tomcat;
- Настройка портов для взаимодействия с веб-сервером.

2.6 Руководство пользователя

Данное руководство предназначено для директора и сотрудников компании «3Д Коре».

Регистрация и авторизация в системе. Директор компании является зарегистрированным пользователем системы по умолчанию. Он же является администратором сайта. Вход любого пользователя осуществляется с использованием личного логина и пароля. Регистрацию сотрудников выполняет директор. Для появления формы «вход», необходимо перейти на главную страницу (рисунок 2.7).

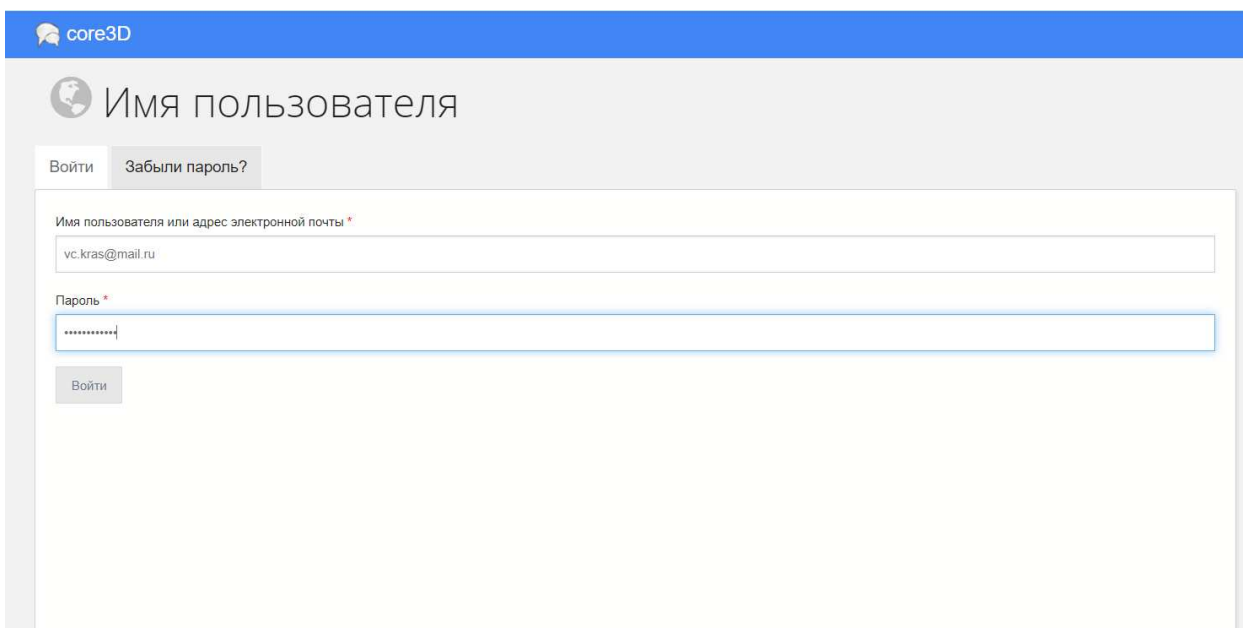


Рисунок 2.7 – Скриншот главной страницы

Выполняя вход как директор, пользователь попадает на страницу «административное меню директора», (рисунок 2.8).

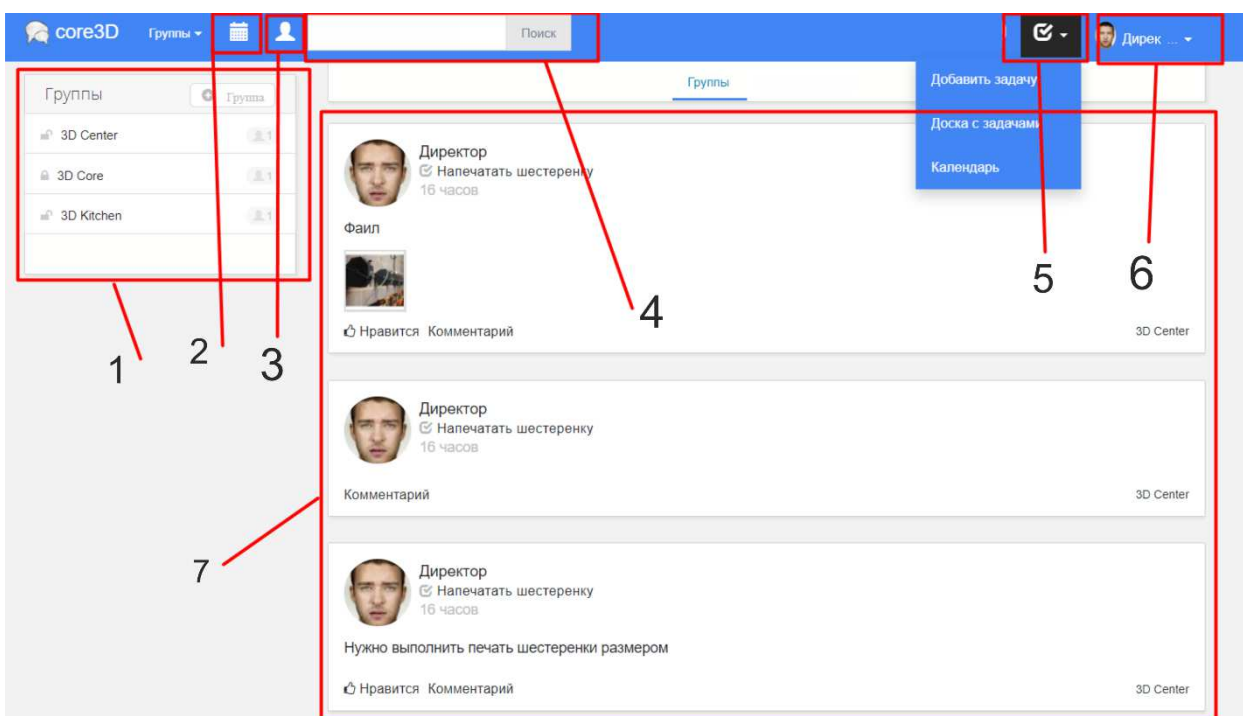


Рисунок 2.8 – Административное меню директора

На данной странице для пользователя доступны следующие возможности:

1. Окно со списком существующих на данный момент групп (структурных подразделений). Как видно на скриншоте, создано 3 структурных подразделения «3Д Центр», «3Д Коре» и «3Д кинтчен». В этом же окне, в правом верхнем углу расположена кнопка для добавления новой группы. Страница сайта с возможностью добавления новой группы представлена на рисунке 2.9.

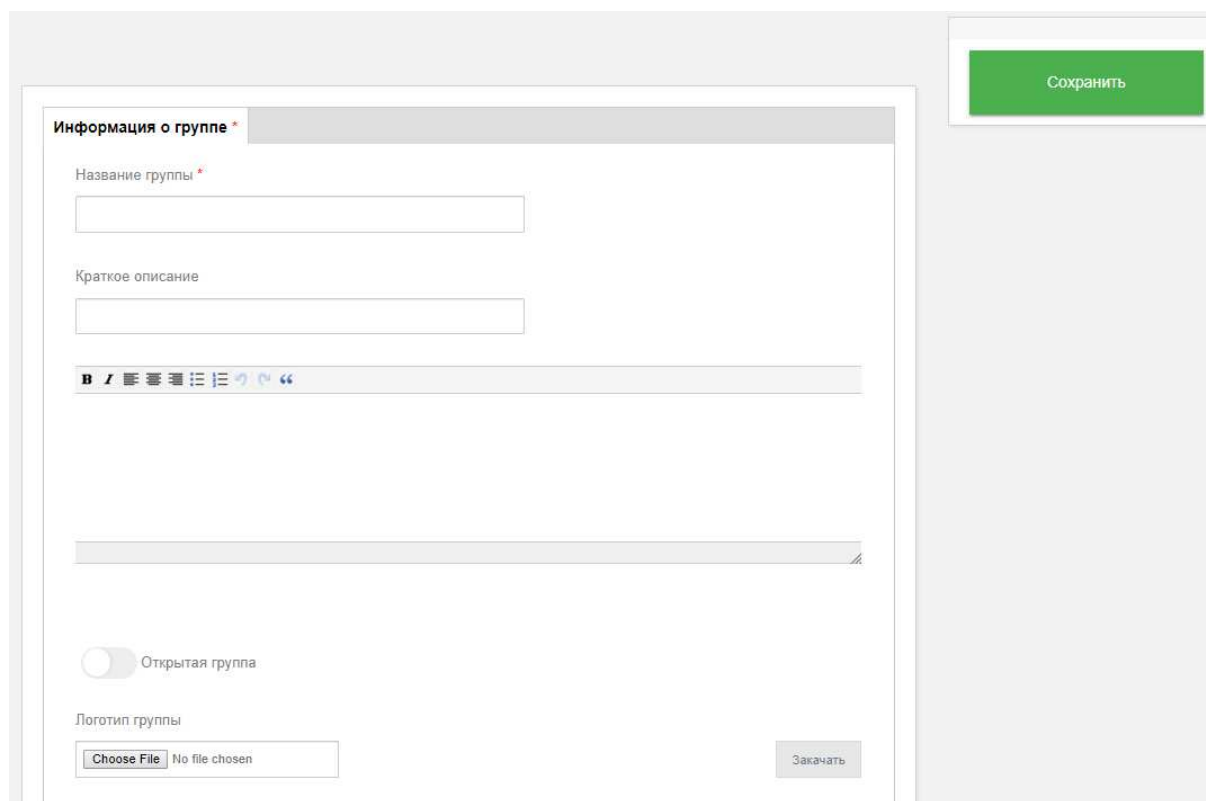


Рисунок 2.9 – Скриншот страницы создание группы

2. Ссылка на «календарь». При нажатии на данную ссылку пользователь попадает на страницу с календарем. На календаре отмечены все задачи со сроком выполнения (рисунок 2.10).

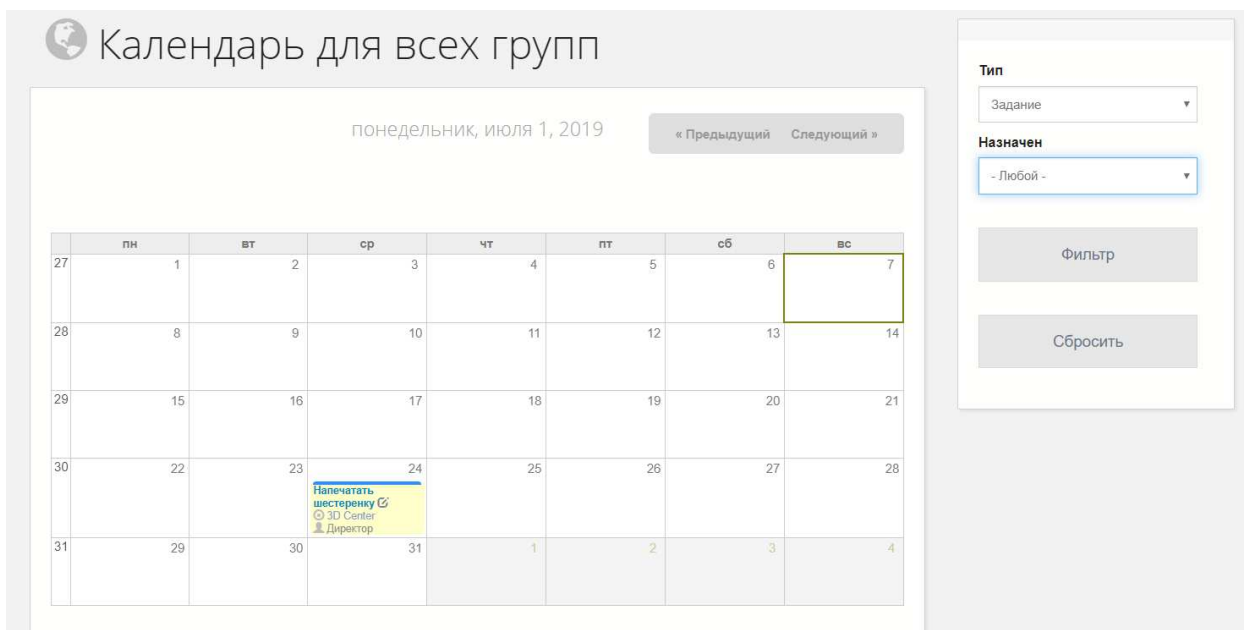


Рисунок 2.10 – Скриншот страницы с календарем

3. Ссылка на страницу «с текущими пользователями» (сотрудниками).

Скриншот этой страницы изображен на рисунке 2.11.

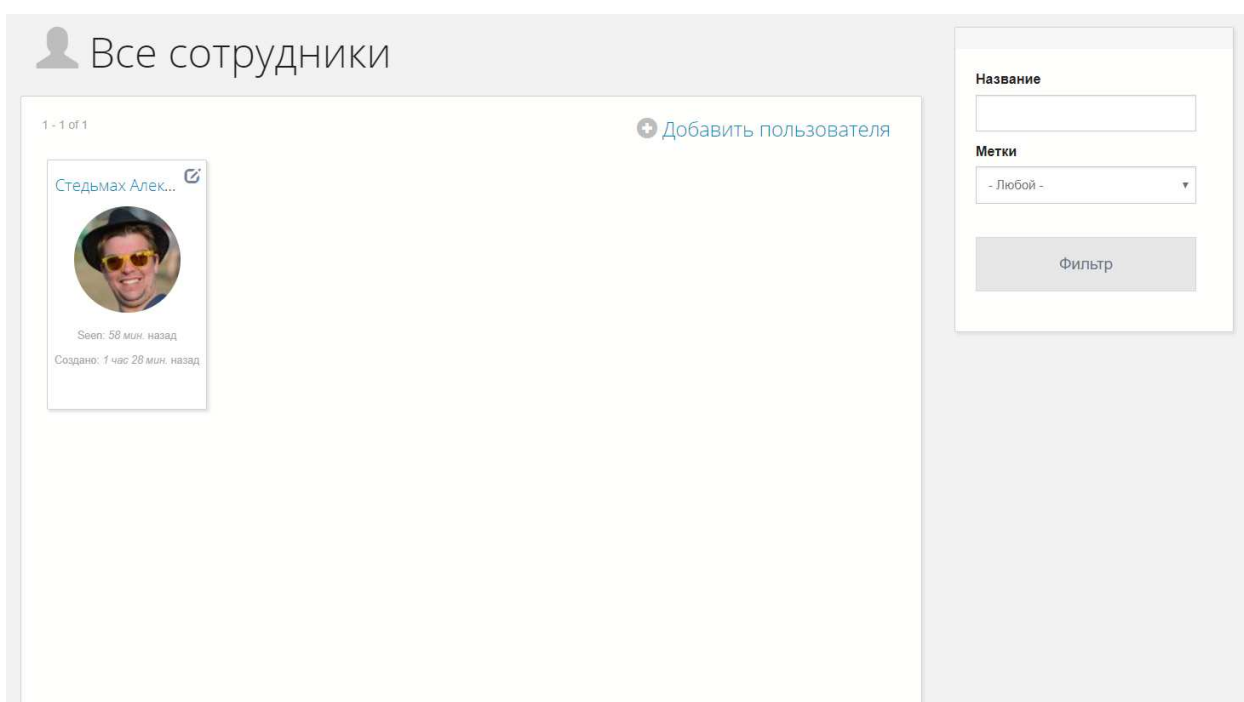


Рисунок 2.11 – Скриншот страницы с текущими пользователями (сотрудниками)

На данной странице отображаются все зарегистрированные сотрудники, кнопка добавления нового сотрудника. В верхнем правом углу аватара

пользователя расположена кнопка редактирования профиля сотрудника. В правой части страницы реализована возможность применить фильтр к отображаемым на этой странице сотрудникам. На рисунке 2.12 представлен скриншот страницы «добавление нового пользователя».

Рисунок 2.12 – Скриншот страницы добавления нового пользователя (сотрудника)

На этой странице руководитель задает имя сотруднику, указывает e-mail сотрудника, в дальнейшем сотруднику придёт приглашение на почту. Еще руководитель присваивает роль, доступно две роли «сотрудник» и «директор». В закладке «группы» директор может назначить сотрудника в одну или несколько существующих групп (структурные подразделения). Пароль для пользователя директор создает на странице редактирования профиля сотрудника.

4. Строка поиска по сайту. Реализована возможность поиска по названию задачи или по сотруднику.

5. «Раскрывающиеся меню» работы с задачами. В данном меню доступен пункт меню с ссылкой на страницу «создания задачи» (рисунок 2.13).

Добавить задачу

Название:

Группа:

Для:

Статус:

Дата готовности:

[Добавить файл](#)

Рисунок 2.13 – Скриншот страницы создать задачу

На данной странице представлены следующие поля. «Название», пользователь (директор) указывает названия создаваемой задачи. «Группа», в данном поле можно присвоить задаче структурное подразделение, в котором эта задача будет выполняться. Поле «для» служит для назначения ответственного сотрудника за создаваемую задачу. В поле «статус» директор назначает статус для задачи, в соответствии со столбцами канбан-доски. Для создаваемой задачи можно указать дату ее выполнения в поле «дата выполнения». Также реализована возможность прикрепить необходимый для выполнения задачи файл. Все поля, кроме «названия» и «статус», являются не обязательными.

Следующий пункт в раскрывающемся меню – это «доска с задачами», при переходе по данной ссылке пользователь попадает на страницу с «канбан-доской» (рисунок 2.14).

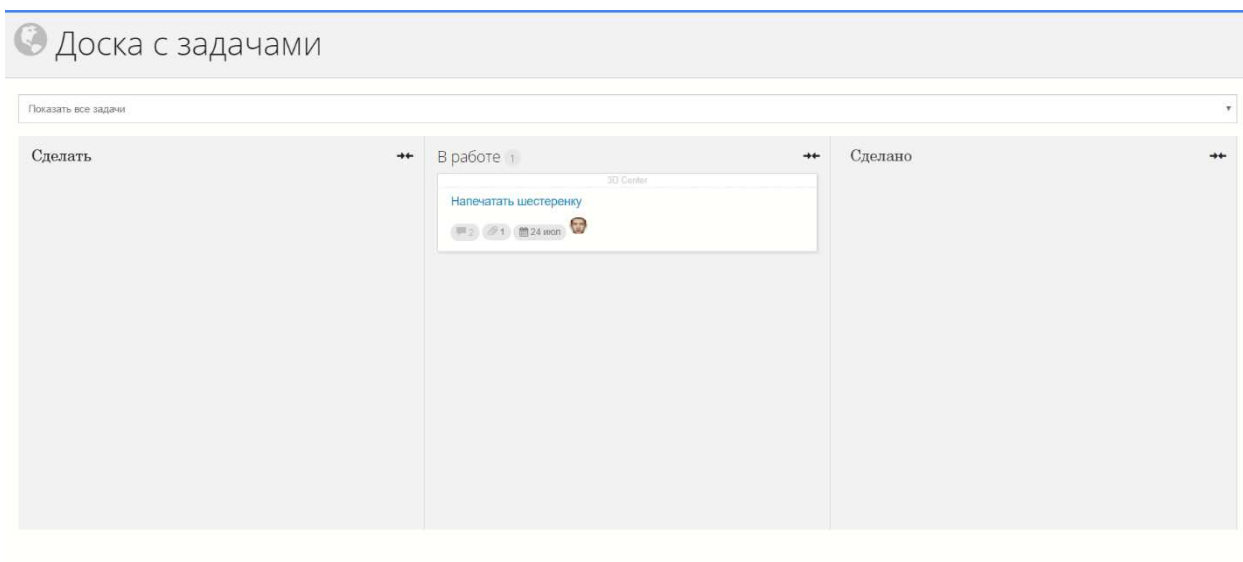


Рисунок 2.14 – Скриншот страницы с канбан-доской

На этой странице у пользователя есть возможность работы с «канбан-доской». «Канбан-доска» состоит из трех столбцов: «Сделать», «В работе», «Сделано». В каждом из столбцов может находиться неограниченное число задач. Перемещение окна с задачей по столбцам выполняется следующим образом: навести курсор на задачу, нажать левую кнопку мыши, перетащить задачу в требуемый столбец. У «задачи», отображаемой на доске, есть название, являющейся ссылкой, на «страницу задачи». В нижней части окна с задачей находятся три ссылки, слева направо, ссылка на комментарий к задаче, ссылка на прикрепленные файлы, ссылка на календарь (если установлена дата выполнения задачи), а также отображается фотография ответственного за задачу сотрудника.

6. «Дополнительное раскрывающиеся меню». В данном меню находится ссылка на редактирование профили и выхода из учетной записи.

7. В данной области на странице административного меню директора располагается информация о последних действиях пользователей на сайте.

Если вход в систему выполнен «сотрудником», то у данного пользователя остается тот же интерфейс системы, что и при входе в систему «директором», за исключением, у сотрудника отсутствуют возможности

создавать группы (структурные подразделения) и создавать новых пользователей.

Заключение

В соответствии с поставленной целью, в рамках выпускной квалификационной работы было разработано веб-приложение для автоматизации управления задачами в компании 3Д Коре.

В процессе выполнения работы были достигнуты следующие результаты:

1. Проведен анализ предметной области. Составлена организационная диаграмма предприятия. Для того чтобы понимать структуру постановки задач, для структурных подразделений, была составлена диаграмма цепочки добавленной стоимости, VAD диаграмма. Основываясь на результатах анализа предприятия, принято решение, что необходимо разработать автоматизированную систему управления задачами;

2. Сформированы первоначальные требования к разрабатываемой системе;

3. Проведен анализ существующих аналогов на рынке. Были выделены три наиболее подходящих решения, которые были представлены на выбор директору компании «3Д Коре». Взяв во внимание комментарии относительно готовых решений от директора, принято решение о целесообразности проектирования собственной автоматизированной системы;

4. На первоначальном этапе проектирования была разработана модель базы данных. Построена концептуальную модель, на основе ее сформирована таблицы для моделирования логической и физической модели базы данных;

5. На следующем этапе работы, было спроектировано программное обеспечение. Для формализации функциональных требования использовалась UML диаграмму прецедентов. Сформирована структура сайта. Диаграмма вариантов использования дала представление что должна делать система. На вопрос «как система должна работать?» получен ответ с помощью диаграммы активностей;

6. Составлено руководства программиста и пользователя.

Результатом работы является, веб-приложение, для автоматизации управления задачами на предприятии 3Д Коре. Данное приложение успешно внедрено на предприятие и в данный момент осуществляется поддержка веб-приложения, а также ведутся работы над следующими расширениями возможностей:

1. Формирование отчета о выполненных задачах конкретным сотрудником;

2. Проектирование и разработка модуля по автоматическому начислению премии для сотрудника, выполнившего заказ вовремя или ранее установленного срока;

3. Интеграция модуля обработки клиентских заказов с интернет-магазина «3Д Китчен», для автоматического формирования задачи;

4. Автоматическая рассылка всем пользователям, на электронную почту, сообщения о предстоящем дне рождения сотрудника;

5. Проектирование и разработка модуля синхронизации сообщения с социальных сетей и корпоративной электронной почты.

По отзывам руководителя предприятия, замечены следующие улучшения:

1. Сократилось время на постановку задачи для сотрудников;

2. На 65% увеличилось количество выполненных заказов в срок;

3. Увеличился объем работ, выполняемый сотрудниками;

4. У руководителя увеличилось количество свободного времени.

Список используемой литературы

1. Буч, Гради Введение в UML от создателей языка / Гради Буч, Джеймс Рамбо, Ивар Якобсон. - М.: ДМК Пресс, 2015. - 496 с.
2. Буч, Грейди Язык UML. Руководство пользователя / Грейди Буч, Джеймс Рамбо, Айвар Джекобсон. - М.: ДМК, 2015. - 432 с.
3. Документация к PostgreSQL 9.6.2. [Электронный ресурс] – Режим доступа: <http://repo.postgrespro.ru/doc/pgsql/9.6.2/ru/postgres-A4-fop.pdf>
4. Справочное руководство по MySQL 4.0 [Электронный ресурс] – Режим доступа: <http://www.mysql.ru/docs/mysql-man-4.0-ru/introduction.html#features>
5. Вахрушев В. И., Гусева Т. Ф., Седнева Д.А., Климов П.А. Разработка диаграммы VAD на примере моделирования бизнес-процесса «Транспортировка готовой продукции» // Современные научные исследования и инновации. 2016. № 1 [Электронный ресурс]. URL: <http://web.snauka.ru/issues/2016/01/61616>
6. Шеер, А.-В. ARIS - моделирование бизнес-процессов / А.-В. Шеер. - М.: Вильямс, 2016. - 224 с.
7. Беляцкая, Т. Н. Анализ интеллектуальных информационных систем на примере сCRM и ERP / Т. Н. Беляцкая. - М.: Синергия, 2015. - 106 с.
8. Ильин, В. В. Внедрение ERP-систем: управление экономической эффективностью / В. В. Ильин. - М.: Теревинф, 2015. - 833 с.
9. Шнайдер Microsoft SQL Server 6.5. Проектирование высокопроизводительных баз данных / Шнайдер, Роберт. - М.: ЛОРИ, 2016. - 361 с.
10. Редько, В.Н. Базы данных и информационные системы / В. Н. Редько, И.А. Басараб. - М.: Знание, 2015. - 516 с
11. Мюллер, Р.Д. Проектирование баз данных и UML / Р. Д. Мюллер; Пер. с англ. Е.Н. Молодцова. – М.: Лори, 2014. – 420 с.

12. Стружкин, Н.П. Базы данных: проектирование, практикум: Учебное пособие для академического бакалавриата / Н.П. Стружкин, В.В. Годин. – Лберцы: Юрайт, 2016. – 291 с.
13. Вуд, Кит Расширение библиотеки jQuery / Кит Вуд. - М.: ДМК Пресс, 2018. - 184 с.
14. Панченко И. PostgreSQL: вчера, сегодня, завтра // Открытые системы. СУБД. 2015. № 3
15. Eclipse, NetBeans или IntelliJ IDEA? Выбираем IDE для Java-разработки. [Электронный ресурс] – Режим доступа: <https://javarush.ru/groups/posts/1642-eclipse-netbeans-ili-intellij-idea-vihbiraem-ide-dlja-java-razrabotki>
16. Руководство разработчика библиотеки Spring [Электронный ресурс] <https://spring.io/guides>

Федеральное государственное автономное
образовательное учреждение
высшего образования
«СИБИРСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»
Институт космических и информационных технологий
институт
Кафедра «Информатика»
кафедра

УТВЕРЖДАЮ

Заведующий кафедрой

А.С. Кузнецов

подпись

инициалы, фамилия

«11»

07

2019г.

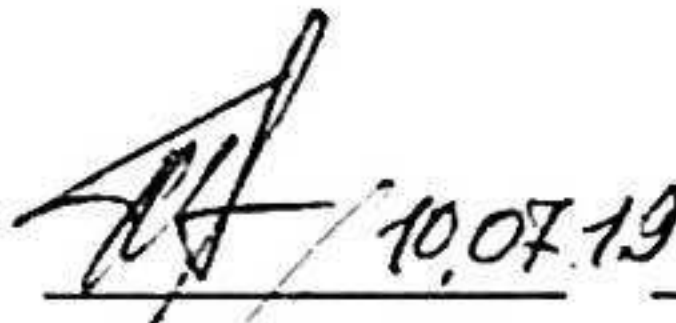

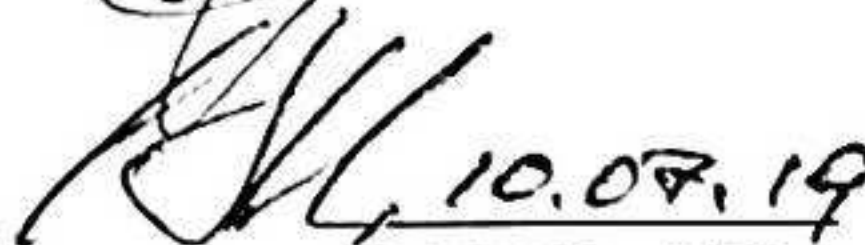
БАКАЛАВРСКАЯ РАБОТА

09.03.04 Программная инженерия

код и наименование специальности

Автоматизация управления задачами в компании «3Д Коре»

тема

Руководитель	доцент кафедры «Информатика», канд. тех. н.	 10.07.19	В.В. Тынченко
	должность, ученая степень	подпись, дата	фамилия, инициалы
Выпускник		 10.07.19	А.С. Поплюйков
		подпись, дата	фамилия, инициалы
Нормоконтролер		 10.07.19	О.А. Антамошкин
		подпись, дата	фамилия, инициалы

Красноярск 2019

Федеральное государственное автономное
образовательное учреждение
высшего образования
«СИБИРСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»

Институт космических и информационных технологий
институт
Кафедра «Информатика»
кафедра

УТВЕРЖДАЮ

~~Заведующий кафедрой~~

А. С. Кузнецов

подпись

инициалы, фамилия

«21»

05

2019 г.

ЗАДАНИЕ
НА ВЫПУСКНУЮ КВАЛИФИКАЦИОННУЮ РАБОТУ
в форме бакалаврской работы

Студенту Поплюйкову Александру Сергеевичу
фамилия, имя, отчество

Группа ВКИ15-16БВВ Направление (специальность) 09.03.04
подпись, дата код

Программная инженерия
наименование

Тема выпускной квалификационной работы:
Автоматизация управления задачами в компании «ЗД Коре»

Утверждена приказом по университету: № 6826/с от 2019-05-21

Руководитель ВКР: В.В. Тынченко, доцент кафедры «Информатика», канд. тех. н.
инициалы, фамилия, должность, ученая степень

Исходные данные для ВКР: информация о бизнес-процессах управления задачами компании в «ЗД Коре»

Перечень разделов ВКР:
введение, аналитическая часть, разработка автоматизированной системы, заключение, список использованных источников

Перечень графического материала: презентационные слайды PowerPoint.

Руководитель ВКР 
подпись В. В. Тынченко
фамилия, инициалы

Задание принял к исполнению 
подпись А. С. Поплюйков
фамилия, инициалы студента

«21» мая 2019 г.