

Федеральное государственное автономное
образовательное учреждение
высшего образования
«СИБИРСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»
Институт космических и информационных технологий
Кафедра вычислительной техники

УТВЕРЖДАЮ
Заведующий кафедрой
_____ О.В. Непомнящий
(подпись)
«___» _____ 2019

БАКАЛАВРСКАЯ РАБОТА

09.03.01 Информатика и вычислительная техника
(код и наименование направления)

Автоматизированная система учёта рабочего времени.

Подсистема управления пользователями и управления корпоративным
календарем
тема

Руководитель	_____	<u>доцент, к.т.н.</u>	<u>Н.Ю. Сиротина</u>
	подпись, дата	должность, ученая степень	инициалы, фамилия
Выпускник	_____		<u>В.О. Мартыненко</u>
	подпись, дата		
Нормоконтролер	_____	<u>доцент, к.т.н.</u>	<u>В.И. Иванов</u>
	подпись, дата	должность, ученая степень	инициалы, фамилия

Красноярск 2019

СОДЕРЖАНИЕ

ВВЕДЕНИЕ.....	4
1 Анализ требований и аналогичных подсистем	6
1.1 Требования к подсистемам.....	6
1.2 Анализ аналогичных подсистем	7
1.2.1 Анализ аналогичных подсистем управления пользователей	8
1.2.1.1 ИАС «Градоустройство». Подсистема «Администрирование» ..	8
1.2.1.2 1С: Предприятие 8. Администрирование пользователей	10
1.2.1.3 UMI.CMS.....	11
1.2.2 Анализ подсистем управления корпоративным календарем.....	12
1.2.2.1 1С: Предприятие 8. Производственный календарь	12
1.2.2.2 Производственный календарь 2019	13
1.3 Вывод.....	14
2 Проектирование.....	16
2.1 Архитектура.....	16
2.2 Структура данных	17
2.3 Подсистема управления пользователями	22
2.3.1 Аутентификация, авторизация и регистрация пользователей.....	22
2.3.2 Прецеденты.....	24
2.3.3 Диаграмма классов.....	27
2.4 Подсистема управления корпоративным календарем.....	28
2.4.1 Прецеденты.....	28

2.4.2 Диаграмма классов.....	32
2.5 Вывод.....	34
3 Практическая реализация.....	35
3.1 Серверная часть.....	35
3.2 Клиентская часть.....	36
3.3 Тестирование.....	38
3.4 Развертывание ПО.....	42
3.5 Вывод.....	44
ЗАКЛЮЧЕНИЕ.....	45
СПИСОК СОКРАЩЕНИЙ.....	46
СПИСОК ИСПОЛЬЗУЕМЫХ ИСТОЧНИКОВ.....	47
ПРИЛОЖЕНИЕ А.....	49
ПРИЛОЖЕНИЕ Б.....	50
ПРИЛОЖЕНИЕ В.....	56

ВВЕДЕНИЕ

В современном мире существует высокая потребность в применении систем, призванных уменьшить количество бумажной отчетности, увеличить эффективность распределения рабочего времени и скорость взаимодействия между сотрудниками. Как правило, для выполнения вышеперечисленных задач требуется разработка специализированного программного обеспечения, наиболее соответствующего потребностям компании-заказчика. Большая часть программного обеспечения, разрабатываемого в наши дни, требует подключения к сети Интернет, поэтому наиболее часто делается выбор в пользу создания веб-приложения, которое не требует от пользователя установки каких-либо дополнительных программ и компонентов и предоставляет доступ к своему функционалу из любой точки мира.

Несмотря на то, что на данный момент существует множество программ, реализующих учет рабочего времени, каждая из них не полностью удовлетворяет требованиям компании-заказчика, в следствии чего было принято решение о разработке собственной системы.

Так как система учета рабочего времени реализует большой функционал, она была разбита на следующие подсистемы, разрабатываемые параллельно:

- Подсистема управления корпо календарем;
- Подсистема управления пользователями;
- Подсистема учета отгулов и отпусков;
- Подсистема управления отчетностью.

Целью выпускной квалификационной работы является разработка подсистем управления пользователями и управления корпоративным календарем.

В ходе работы были решаются следующие задачи:

- Изучить предметную область;
- Выполнить сравнительный анализ существующих аналогов;
- Спроектировать и разработать требуемые подсистемы;

- Протестировать разработанные подсистемы;
- Интегрировать подсистемы в основную систему;
- Составить документацию;
- Ввести разработанное программное обеспечение в эксплуатацию.

В соответствии с требованиями заказчика выбираются способы хранения и управления данными, методы аутентификации и авторизации, а также решается будет ли приложение в открытом доступе, или оно будет расположено только в корпоративном домене.

В результате выполненной работы реализовано веб-приложение на платформе *ASP .NET Core 2.0*. Подсистема управления пользователями предназначена для корпоративной сети. В ней реализован доступ по правам пользователя, а также администрирование пользователей. Подсистема управления календарем предоставляет возможность ведение корпоративного календаря, его просмотр и редактирование. Интерфейс и функционал подсистем разработаны с учетом потребностей компании-заказчика.

В разработанных подсистемах использованы следующие технологии:

- *ASP .Net Core 2.0*;
- Сетевые протоколы аутентификации *Kerberos* и *NTLM*;
- Служба каталогов *Active Directory*;
- *UI*-фреймворк *Telerik Kendo UI*;
- Инструментарий с открытым исходным кодом *Bootstrap*;
- *Entity Framework Core* для работы с базой данных;
- Набор серверов *IIS*;
- Шаблон проектирования *MVC*;
- *SQL Server*;
- Механизм *Dependency Injection*.

1 Анализ требований и аналогичных подсистем

1.1 Требования к подсистемам

Поставлена задача разработки двух подсистем: управления пользователями и управления корпоративным календарем. Обе подсистемы должны соответствовать следующим критериям.

Подсистемы должны быть реализованы на платформе *ASP .NET Core 2.0*, которая является открытым ресурсом, поддерживающим кроссплатформенность, модульный конвейер *HTTP*-запросов, создание и использование пакетов *NuGet*, а также возможность развертывать приложение как с помощью специализированного программного обеспечения, так и в рамках собственного процесса. Данная платформа включает в себя *ASP.NET MVC Framework*, который объединяет функциональность *MVC*, *Web API* и *Web Pages* [1].

Требования к подсистеме управления пользователями:

- Обеспечивать работу нескольких категорий (ролей) пользователей: сотрудник, подтверждающий, согласующий, менеджер системы, администратор;
- Аутентификация с помощью сетевых протоколов *Kerberos* и *NTLM*;
- Регистрация и авторизация пользователей через корпоративный домен посредством *Active Directory*;
- Множественные роли;
- Пользователь должен иметь доступ к функциям системы в соответствии со своими ролями;
- Подсистема должна сохранять данные аутентифицированного пользователя в рамках сессии с помощью *Cookies* и перенаправлять его на стартовую страницу (наполнение страницы различается для каждой роли).

Функционал подсистемы управления пользователями:

- Редактирование учетных записей пользователя;
- Назначение согласующих и подтверждающих;
- При назначении согласующего сотруднику уведомлять согласующего в случае если у сотрудника обнаружены нерассмотренные заявки.

Требования к подсистеме управления корпоративным календарем:

- Выделение выходных и рабочих дней;
- Добавление описания к выделенным дням;
- Загрузка дат из *Excel* файла.

Функционал подсистемы управления корпоративным календарем:

- Просмотр корпоративного календаря всеми пользователями;
- Редактирование календаря менеджером системы.

1.2 Анализ аналогичных подсистем

На сегодняшний день существует множество подсистем управления пользователями. Реализаций подсистемы управления корпоративным календарем гораздо меньше. Все найденные аналоги платные и не распространяются в свободном доступе. Критерии сравнения для подсистем управления пользователями:

- Поддержка нескольких ролей;
- Доступ к системе по правам пользователей;
- Авторизация пользователей через корпоративный домен;
- Регистрация пользователя через Active Directory;
- Возможность редактирования учётных записей;
- Возможность назначения зависимости одних пользователей от других;
- Возможность наличия нескольких ролей у одного пользователя;
- Доступ к функциям по ролям.

Критерии сравнения для подсистем управления корпоративным календарем:

- Просмотр корпоративного календаря;
- Возможность отметить дни на календаре;
- Возможность отмечать выходные дни ставшие рабочими и наоборот;
- Изменение отмеченных дней в календаре;
- Добавление описания к выделенному дню.

1.2.1 Анализ аналогичных подсистем управления пользователей

1.2.1.1 ИАС «Градоустройство». Подсистема «Администрирование»

ИАС «Градоустройство» - информационно-аналитическая система поддержки принятия управленческих решений. **Подсистема «Администрирование»** обеспечивает конфигурирование политик разграничения прав доступа пользователей к объектам и функциям ИАС «Градоустройство» [2].

Функциональность системы:

- Создание и изменение пользователей и групп пользователей – исполнителей и групп исполнителей административных процедур предоставления муниципальных услуг и исполнения муниципальных функций в области градостроительства в электронном виде на основании организационной структуры учреждения-пользователя ИАС «Градоустройство» (с учетом отделов и должностей), иных пользователей системы и сервисов межведомственного электронного взаимодействия;
- Создание и изменение учетных записей;
- Разграничение прав доступа для каждой учетной записи к каталогам в структуре объектов БД ИАС «Градоустройство», типам

объектов и их отдельным атрибутам на просмотр, создание, удаление и изменение элементов БД [2].

В системе реализована модель управления правами на основе ролей (групп). Ролевой метод управления доступом контролирует доступ пользователей к информации на основе типов их активностей в системе. Применение данного метода подразумевает определение ролей в системе. Роль определяет совокупность действий и обязанностей, связанных с определенным видом деятельности. Таким образом, вместо того, чтобы указывать все типы доступа для каждого пользователя к каждому объекту, достаточно указать тип доступа к объектам для роли. А пользователям, в свою очередь, указать их роли. Предусмотрена возможность участия пользователя в нескольких группах [2].

- Аутентификация пользователей при входе в систему;
- Экспорт списка пользователей с установленными правами (матрица доступа) в *MS Excel*;
- Восстановление данных.

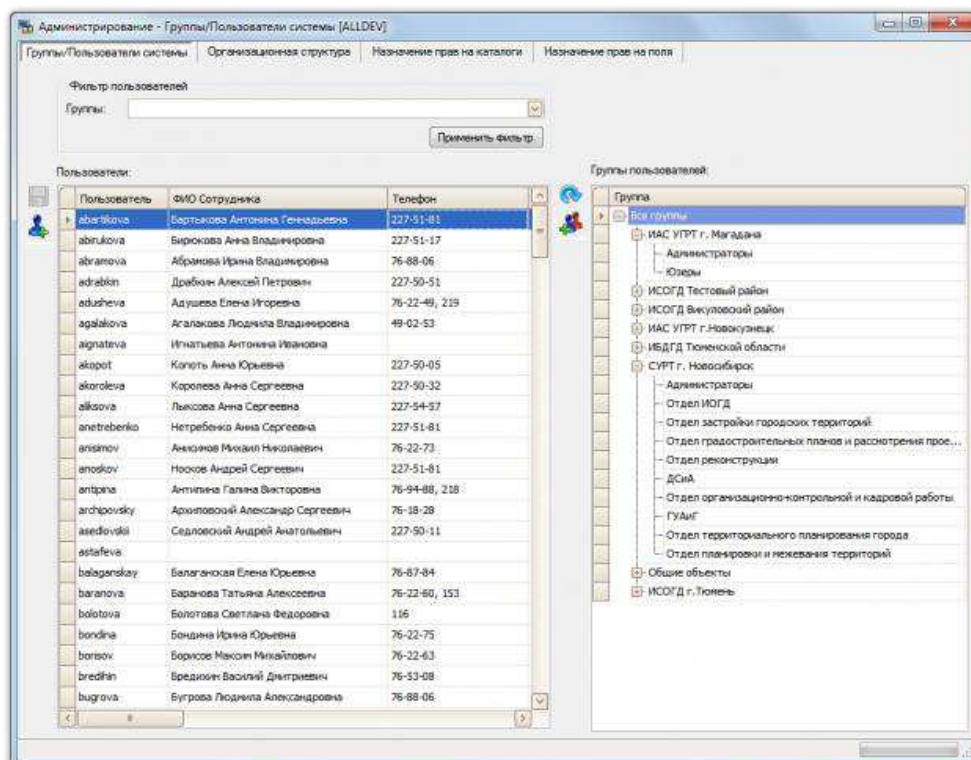


Рисунок 1 – Интерфейс программы ИАС «Градоустройство»

1.2.1.2 1С: Предприятие 8. Администрирование пользователей

1С: Предприятие 8 – система программ, предназначенная для автоматизации деятельности на предприятии. В программе существует подсистема администрирования пользователей [3].

Функциональность подсистемы:

- Ведение списка пользователей, работающих в программе;
- Ведение списка внешних пользователей, имеющих ограниченный доступ к специализированным рабочим местам, предусмотренным в программе (например, Мои заказы, Анкеты респондента, Оформление заявок и т.п.);
- Настройка прав доступа пользователей и внешних пользователей (при внедрении совместно с подсистемой Управление доступом осуществляется средствами подсистемы Управление доступом);
- Группировка списка пользователей (и внешних пользователей);
- Очистка и копирование настроек отчетов, форм, рабочего стола, разделов командного интерфейса, "Избранного", печати табличных документов и других персональных настроек пользователей (и внешних пользователей);
- Индивидуальная и групповая настройка прав доступа пользователей с помощью профилей и групп доступа;
- Настройка ограничений прав доступа на уровне записей – для отдельных элементов данных информационной базы (элементов справочников, документов, записей регистров и т.д.);
- Отчет по правам интересующего пользователя или группы пользователей;
- Предусмотрены два варианта внедрения в прикладное решение: обычный и упрощенный. Обычный режим настройки прав доступа рассчитан на многопользовательские прикладные решения, в которых, как правило, выполняется групповая настройка прав на

базе групп доступа. В упрощенном режиме настройка прав выполняется индивидуально для каждого пользователя. Второй режим предназначен для конфигураций с небольшим числом пользователей, каждый из которых обладает своим собственным уникальным набором прав [3].

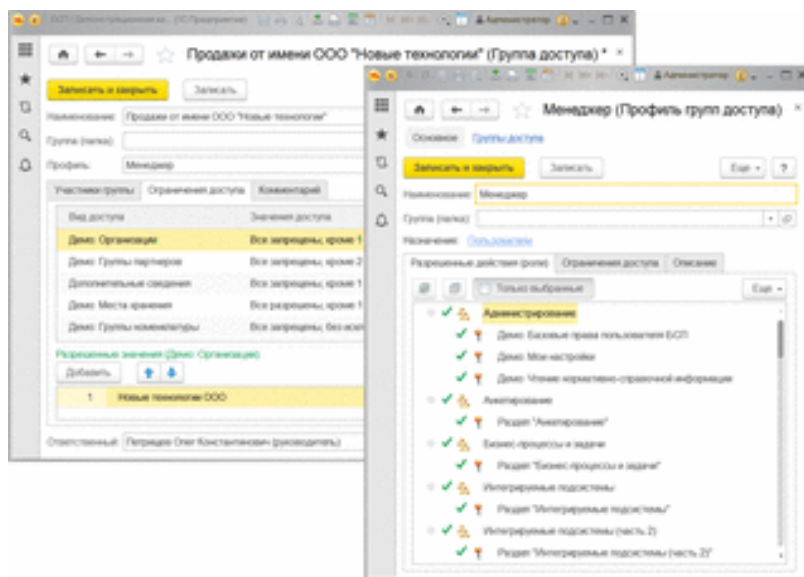


Рисунок 2 – Интерфейс администрирования в 1С: Предприятие 8

1.2.1.3 UMI.CMS

UMI.CMS – система управления сайтами. Содержит в себе подсистему администрирования пользователей [4].

Функциональность:

- Добавлять неограниченное число пользователей и групп пользователей. Назначать группу пользователей по умолчанию;
- Регистрировать пользователей с активацией аккаунтов;
- Назначить пользователя, чьи права будут использоваться для ограничения доступа другим незарегистрированным пользователям;
- Определять права на просмотр страниц и на управление контентом;
- Определять права на доступ к доменам;

- Определять права на удаление и восстановление объектов из корзины;
- Определять права доступа к модулям на просмотр и редактирование данных, а также к отдельным объектам сайта в рамках того или иного модуля;
- Определять права на управление шаблонами данных и справочниками [4].

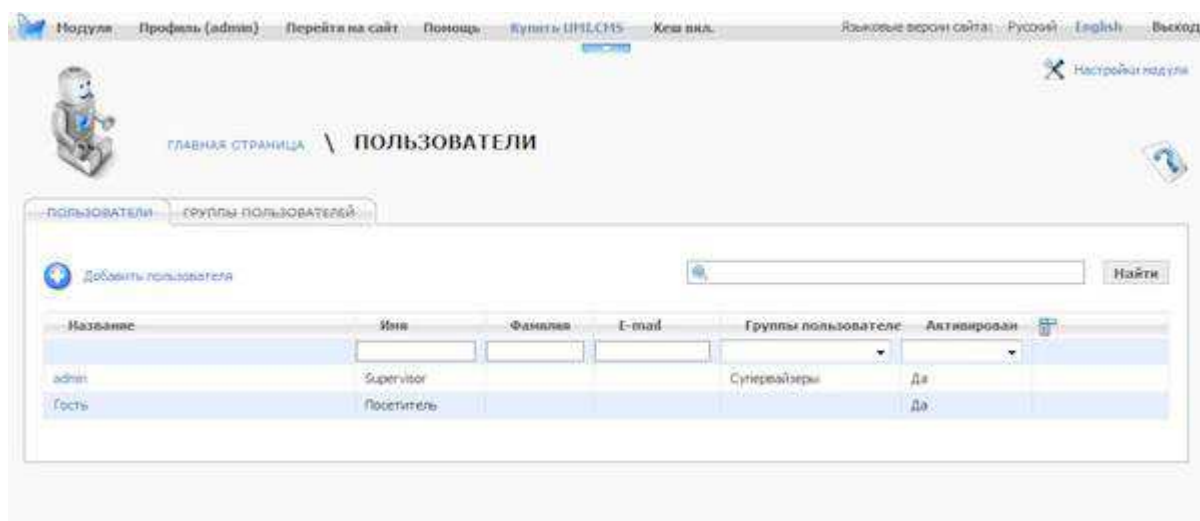


Рисунок 3 – Интерфейс программы *UMI.CMS*

1.2.2 Анализ подсистем управления корпоративным календарем

1.2.2.1 3 IC: Предприятие 8. Производственный календарь

В программе существует подсистема управления производственным календарем [5].

Функциональность подсистемы:

- Возможность создать несколько календарей;
- Возможность заполнить календарь по умолчанию выбранными данными;
- Возможность перенести или изменить день.

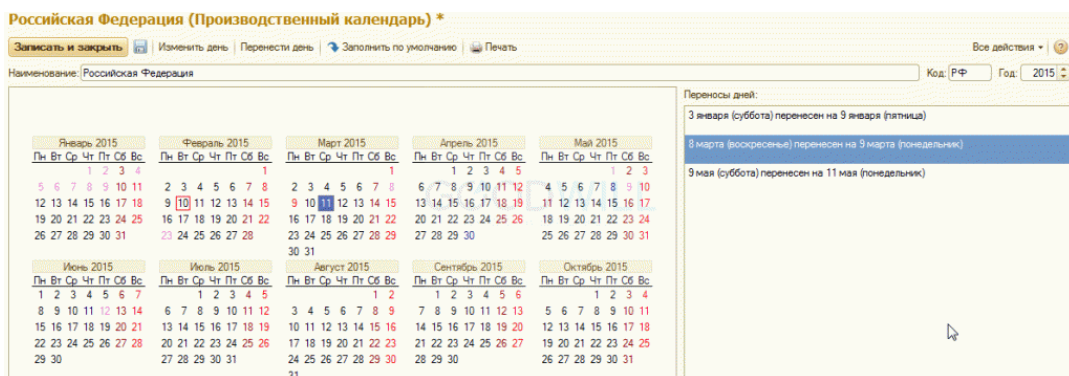


Рисунок 4 – Производственный календарь в 1С: Предприятие 8

1.2.2.2 Производственный календарь 2019

Программа предназначена для удобного просмотра производственных календарей и расчета даты окончания отпуска с учетом праздничных дней. Программа полностью бесплатная [6].

Функции программы:

- Вывод производственного календаря за 2006 - 2019 годы с учетом законодательства РФ (2008-2019 для Украины);
- Расчет даты окончания отпуска с учетом праздничных дней.

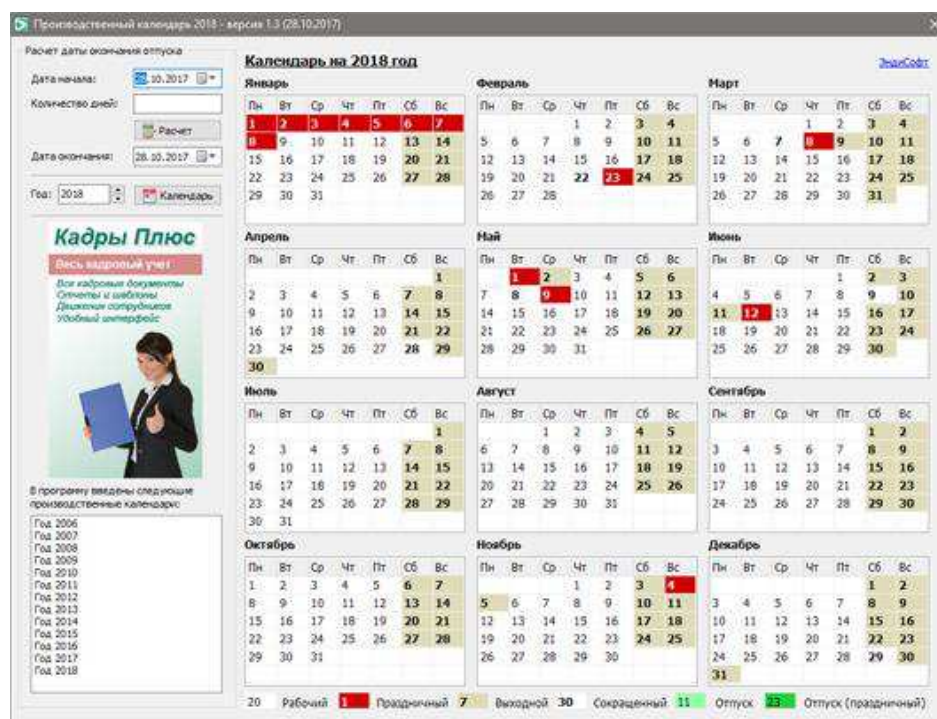


Рисунок 5 – Интерфейс программы Производственный календарь 2019

1.3 Выводы

В результате обзора существующих аналогов подсистемы администрирования были выявлены следующие недостатки:

- Не реализованы необходимые роли;
- Нет функционала назначения одним сотрудникам других сотрудников по какому-либо признаку;
- У подсистем, нет возможности синхронизации с корпоративным доменом.

Таблица 1 – Таблица сравнения аналогичных подсистем управлением пользователями

Критерии	ИАС «Градоустройство»	1С: Предприятие 8	UMI.CMS
Поддержка нескольких ролей	+	+	+
Доступ к системе по правам пользователей	+	+	+
Авторизация пользователей через корпоративный домен	-	+	-
Регистрация пользователя через Active Directory	-	+	-
Возможность редактирования учётных записей	+	+	+
Возможность назначения зависимости одних пользователей от других	-	-	-
Возможность наличия нескольких ролей у одного пользователя	+	+	+
Доступ к функциям по ролям	+	+	+

Среди аналогов подсистемы управления календарем отсутствуют следующие функции:

- Возможность отмечать выходные дни, ставшие рабочими;
- Возможность отметить дни на календаре;
- Наличие описания к измененному дню с подробностями события.

Таблица 2 – Таблица сравнения аналогичных подсистем управлением корпоративным календарем

Критерии	Производственный календарь 2019	1С: Предприятие 8
Просмотр календаря	+	+
Возможность отметить дни на календаре	-	+
Возможность отмечать выходные дни, ставшие рабочими и наоборот	-	-
Изменение отмеченных дней в календаре	-	+
Описания к измененному дню	-	-

Рассмотренные аналоги не реализуют функционал, представленный в требованиях заказчика, а также являются частью более больших систем не содержащих открытого кода, что делает их интеграцию в систему учета рабочего времени невозможной. В связи с этим были разработаны собственные подсистемы.

2 Проектирование

2.1 Архитектура

В качестве архитектуры разрабатываемых подсистем используется паттерн *MVC*, реализованный с помощью *ASP.NET MVC Framework*. Фреймворк предоставляет классическую реализацию паттерна *MVC* с разделением приложения на три компонента *Модель – Представление – Контроллер*. Однако, из-за большого объема бизнес-логики в данной реализации добавляется четвертый компонент, называемый *Сервис*. Этот компонент хранит в себе всю бизнес-логику подсистем, что повышает структурированность системы и уменьшает связность кода, давая возможность его дальнейшего использования в других приложениях. На рисунке 6 представлены все компоненты и их взаимодействие между собой [1].

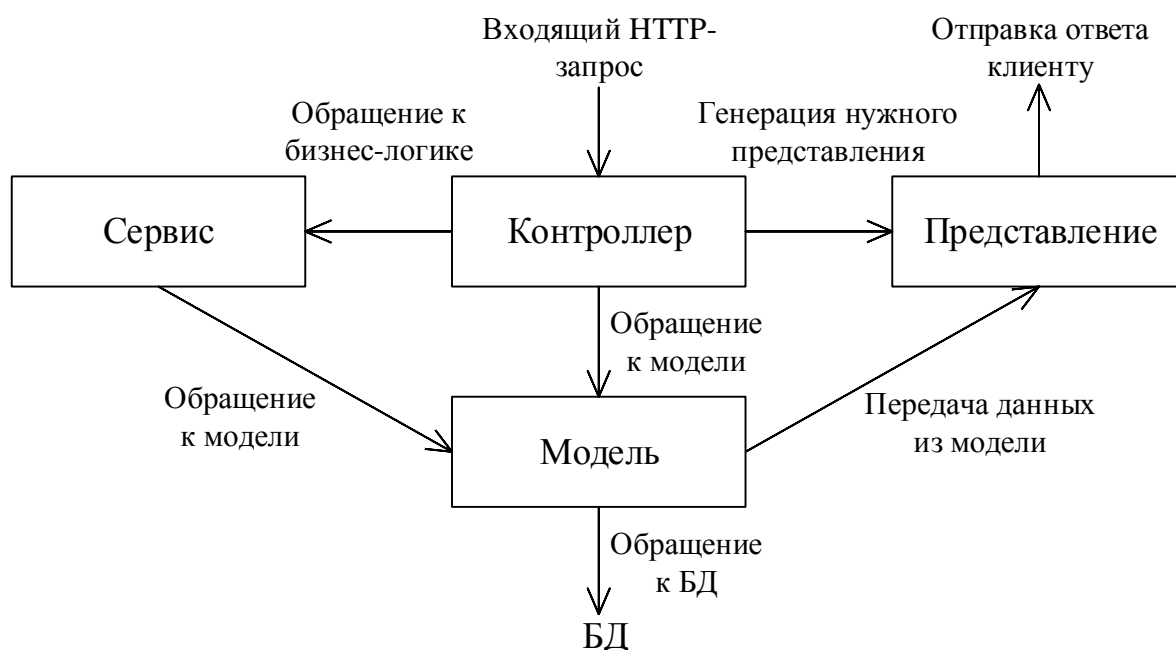


Рисунок 6 – Архитектура системы

Благодаря такому разграничению компонентов системы возможно реализовать концепцию разделения ответственности, при которой каждый компонент отвечает за свою строго очерченную схему, что упрощает разработку,

поддержку и тестирование отдельных компонентов.

Модель описывает используемые в приложении данные, а также логику, связанную с ними, например, валидацию данных. В *MVC*, как правило, модели содержат два типа объектов: объекты, хранящиеся в БД и объекты, используемые представлениями для отображения и передачи данных.

Представление отвечает за пользовательский интерфейс системы, нередко является *html*-страницей, через которую пользователь взаимодействует с приложением. Представление может содержать логику отображения данных, однако оно не содержит логику управления данными или обработки запроса.

Контроллер — центральный компонент архитектуры, связывающий пользователя и приложение, представление и хранилище данных. Контроллер получает пользовательские данные и передает их в *сервис* для последующей обработки и в зависимости от результатов обработки генерирует ответ, например, в виде *html*-страницы, наполненной определенными данными.

Сервис содержит бизнес-логику системы и занимается обработкой данных, полученных от *контроллера*, после чего возвращает их обратно. Также именно в сервисе реализована логика управления данными, добавление, удаление и редактирование.

2.2 Структура данных

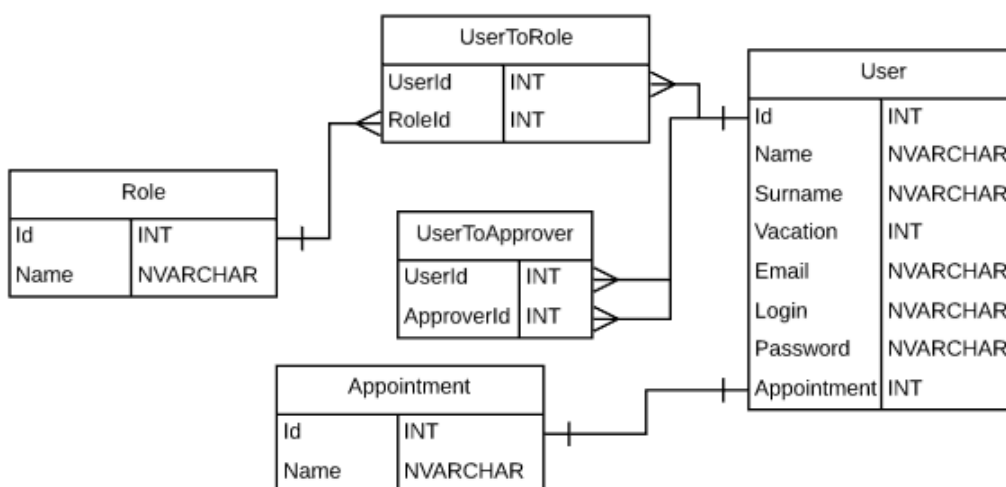


Рисунок 7 – Структура данных для подсистемы управления пользователями

На рисунке 7 представлена *ER*-диаграмма, отображающая структуру данных, используемых подсистемой управления пользователями. Ключевой таблицей является таблица «*User*», хранящая данные о пользователях системы, она связана сама с собой через таблицу «*UserToApprover*», которая демонстрирует отношение между сотрудниками и согласующими, а также с таблицами «*Role*» через таблицу «*UserToRole*» и «*Appointment*», которые хранят информацию о доступных ролях и должностях пользователей.

Calendar	
Id	INT
Type	BIT
Date	DATE
Description	NVARCHAR

Рисунок 8 – Структура данных для подсистемы управления корпоративным календарем

Подсистема управления корпоративным календарем использует лишь одну таблицу «*Calendar*», рисунок 8, которая представляет информацию о датах, отмеченных на календаре. В таблице 3 представлено подробное описание диаграмм.

Таблица 3 – Описание *ER*-диаграммы

Название поля	Описание
<i>User</i>	
<i>Id</i>	Уникальный идентификатор пользователя (Первичный ключ таблицы).
<i>Name</i>	Имя.
<i>Surname</i>	Фамилия.
<i>Vacation</i>	Количество отпускных дней.
<i>Email</i>	Адрес электронной почты.
<i>Login</i>	Имя учетной записи.
<i>Password</i>	Пароль от учетной записи.

Продолжение таблицы 3

<i>AppointmentId</i>	Идентификатор должности (Внешний ключ, служит для связи с таблицей должностей «Appointment»).
<i>UserToApprover</i>	
<i>Id</i>	Уникальный идентификатор (Первичный ключ таблицы).
<i>UserId</i>	Идентификатор сотрудника. Внешний ключ для связи с таблицей пользователей.
<i>ApproverId</i>	Идентификатор согласующего. Внешний ключ для связи с таблицей пользователей.
<i>Role</i>	
<i>Id</i>	Уникальный идентификатор роли (Первичный ключ таблицы).
<i>Name</i>	Наименование роли.
<i>UserToRole</i>	
<i>Id</i>	Уникальный идентификатор (Первичный ключ таблицы).
<i>UserId</i>	Внешний ключ для связи с таблицей пользователей.
<i>RoleId</i>	Внешний ключ для связи с таблицей ролей.
<i>Appointment</i>	
<i>Id</i>	Уникальный идентификатор должности (Первичный ключ таблицы).
<i>Name</i>	Наименование должности.
<i>Calendar</i>	
<i>Id</i>	Уникальный идентификатор даты, отмеченной на календаре (Первичный ключ таблицы).
<i>Type</i>	Тип отмеченной даты.
<i>Date</i>	Дата, отмеченная на календаре.
<i>Description</i>	Описание к отмеченной дате.

В реализуемых подсистемах используется механизм разделения ответственности ролей, так что никакие две роли не имеют одинаковый функционал. В таблице 4 приведена матрица ролей, которая наглядно показывает функционал каждой роли. По умолчанию все пользователи системы имеют роль сотрудник, и как видно из матрицы ролей, чтобы расширить возможности пользователя в системе следует добавить ему дополнительную роль, таким образом пользователи могут иметь несколько ролей одновременно.

Таблица 4 – Матрица ролей

Действие \ Роль	Сотрудник	Согласующий	Подтверждающий	Менеджер	Администратор
Подача заявки через приложение	+				
Подача заявки письмом	+				
Удаление своей заявки на рассмотрении	+				
Просмотр своих заявок	+				
Изменение даты в заявке, если предложена новая дата	+				
Просмотр календаря выходных и праздников	+				
Получение почтовых уведомлений о изменении своей заявки	+				
Подача согласованной заявки		+			
Отклонение заявки на рассмотрении		+			
Просмотр заявок своих подчинённых		+			
Согласование заявки		+			
Получение почтовых уведомлений о изменении заявок подчиненных		+			

Продолжение таблицы 4

Предложение новой даты к заявке на рассмотрении		+			
Подача подтвержденной заявки			+		
Подтверждение заявки			+		
Получение почтовых уведомлений о изменении заявок всех сотрудников			+		
Просмотр заявок всех пользователей			+		
Отклонение заявки без учета согласования			+		
Загрузка выходных и праздников из файла				+	
Редактирование календаря выходных и праздников				+	
Подтверждение выполнения отпуска сотрудника				+	
Возвращение сотрудника из отпуска раньше срока				+	
Просмотр подтвержденных и выполненных заявок				+	
Просмотр списка всех пользователей					+
Редактирование информации о пользователе					+

2.3 Подсистема управления пользователями

2.3.1 Аутентификация, авторизация и регистрация пользователей

В разрабатываемой подсистеме процессы аутентификации, авторизации и регистрации полностью автоматизированы и не нуждаются в контроле со стороны оператора системы. На рисунке 9 изображен алгоритм работы данных механизмов.

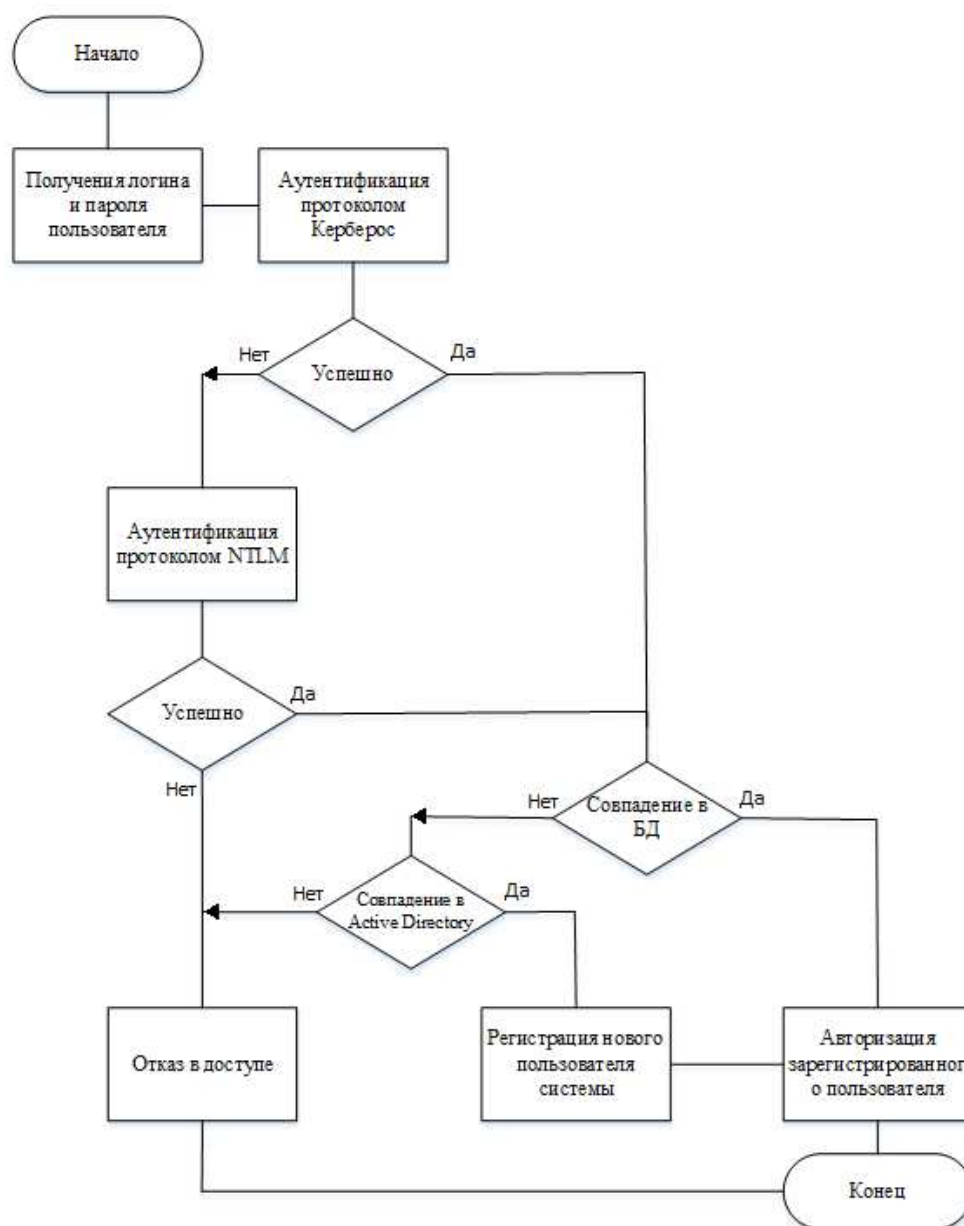


Рисунок 9 – Алгоритм работы механизмов аутентификации, авторизации и регистрации

Аутентификация – проверка подлинности идентификатора, предъявленного пользователем. Положительным результатом аутентификации является установка доверительных отношений, выдача сессионного ключа, а также авторизация пользователя в системе [7]. В данной реализации аутентификации используются сетевые протоколы *Kerberos* и *NTLM*.

Kerberos – сетевой протокол аутентификации, предлагающий механизм взаимной аутентификации клиента и сервера перед установлением связи между ними, в протоколе учтён тот факт, что начальный обмен информацией между клиентом и сервером происходит в незащищенной среде, а передаваемые пакеты могут быть перехвачены и модифицированы. На рисунке 10 представлен алгоритм работы протокола [9].

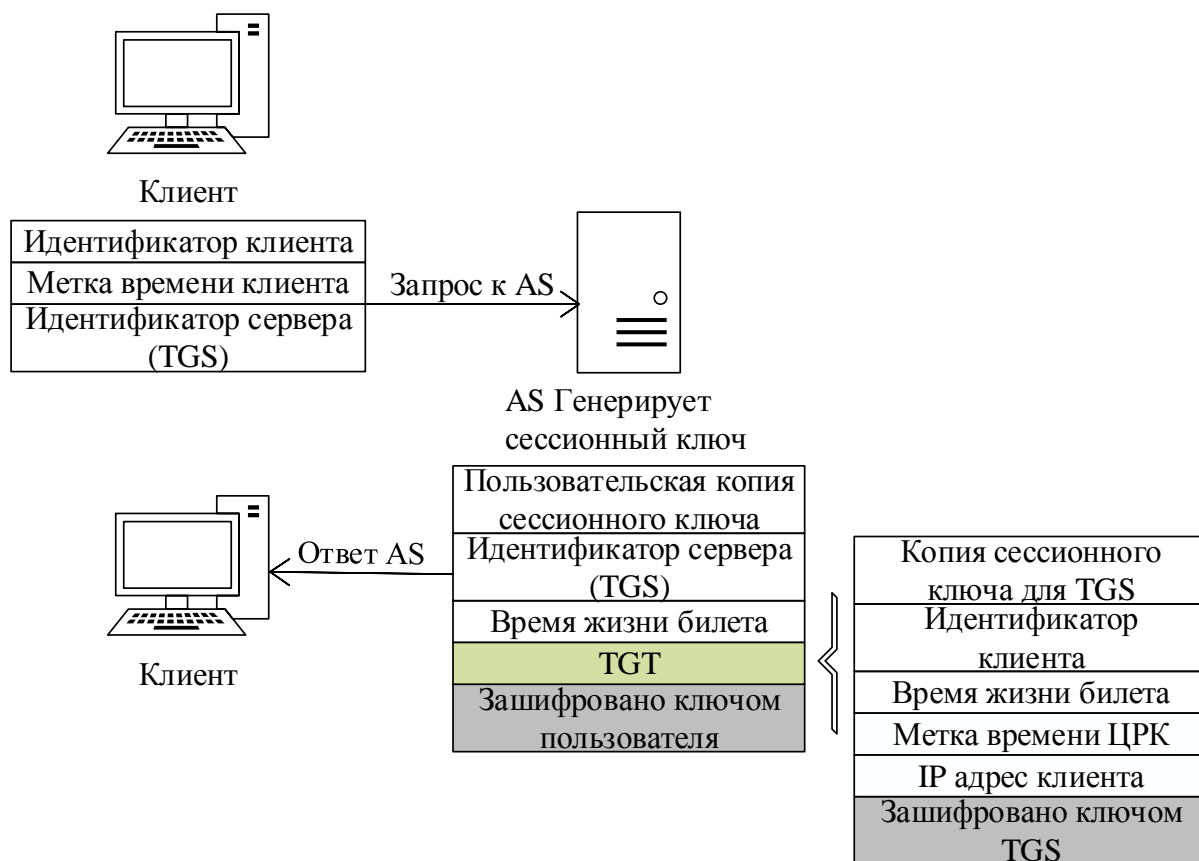


Рисунок 10 – Этапы аутентификации клиента средствами протокола Kerberos 4

TGT (Ticket Granting Ticket) – механизм позволяющий пользователю аутентифицироваться в нескольких сервисах, используя свои доверительные данные только один раз.

AS (Authentication Server) – сервер аутентификации.

TGS (Ticket Granting Server) – сервер выдачи мандатов и разрешений.

NTLM – протокол проверки подлинности запроса и ответа, который использует три сообщения для аутентификации клиента в среде, ориентированной на соединение и четвертое дополнительное сообщение, если требуется проверка целостности. На рисунке 11 представлен алгоритм работы протокола [10].

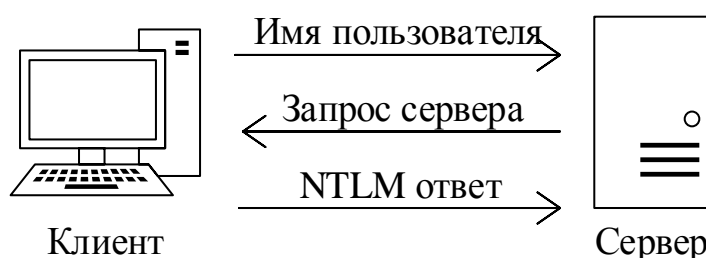


Рисунок 11 – Этапы аутентификации клиента средствами протокола *NTLM*

Авторизация – процесс предоставления определенному пользователю или группе пользователей прав на выполнение определенных действий и подтверждение данных прав при попытке выполнения этих действий [7]. Для авторизации используется технология *Active Directory*.

Active Directory – это иерархически организованное хранилище данных об объектах сети, обеспечивающее удобные средства для поиска и использования этих данных. Компьютер, на котором работает *Active Directory*, называется контроллером домена. Технология *Active Directory* основана на стандартных интернет-протоколах и помогает четко определять структуру сети [8].

Регистрация – процесс создания учетной записи, которая будет храниться на сервере и при помощи, которой пользователь сможет получить доступ к системе.

2.3.2 Прецеденты

На рисунке 12 изображены прецеденты для подсистемы управления пользователями.

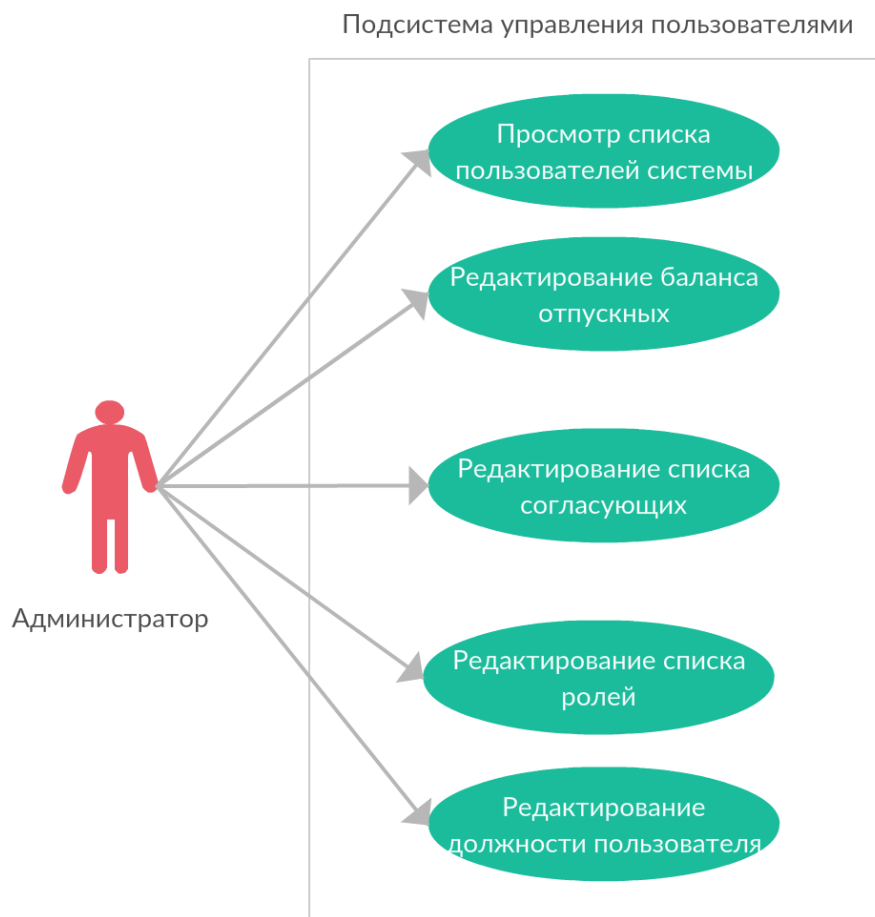


Рисунок 12 – Диаграмма прецедентов для подсистемы управления пользователями

1. Просмотр списка пользователей системы

Предусловие: администратор авторизован.

№	Действие пользователя	Действие системы
1	Переходит на страницу «Список пользователей».	Открывает страницу «Список пользователей», подгружает всех пользователей из БД и отображает их на странице.

Постусловие: загружена страница «Пользователи».

2. Редактирование баланса отпускных

Предусловие: администратор авторизован. Загружена страница «Пользователи».

Есть хотя бы один зарегистрированный пользователь.

№	Действие пользователя	Действие системы
1	Нажимает на поле отвечающее за количество отпускных конкретного пользователя.	Отображает данное поле в режиме редактирования.
2	Изменяет количество отпускных.	Запоминает отредактированные данные.
3	После окончания редактирования нажимает кнопку «Сохранить изменения».	Система сохраняет все сделанные изменения и добавляет их в БД.

Постусловие: изменен баланс отпускных конкретного пользователя.

3. Редактирование списка согласующих

Предусловие: администратор авторизован. Загружена страница «Пользователи».

Есть хотя бы один зарегистрированный пользователь.

№	Действие пользователя	Действие системы
1	Нажимает на поле отвечающее за согласующего конкретного пользователя.	Отображает выпадающее меню содержащее список всех согласующих.
2	Назначает либо удаляет согласующих конкретного пользователя.	Запоминает отредактированные данные.
3	После окончания редактирования нажимает кнопку «Сохранить изменения».	Система сохраняет все сделанные изменения и добавляет их в БД.

Постусловие: отредактирован список согласующих конкретного пользователя.

4. Редактирование списка ролей

Предусловие: администратор авторизован. Загружена страница «Пользователи».

Есть хотя бы один зарегистрированный пользователь.

№	Действие пользователя	Действие системы
1	Нажимает на поле отвечающее за роли конкретного пользователя.	Отображает выпадающее меню содержащее список всех доступных ролей.
2	Назначает либо удаляет роли конкретному пользователю.	Запоминает отредактированные данные.
3	После окончания редактирования нажимает кнопку «Сохранить изменения».	Система сохраняет все сделанные изменения и добавляет их в БД.

Постусловие: отредактирован список ролей конкретного пользователя.

5. Редактирование должности пользователя

Предусловие: администратор авторизован. Загружена страница «Пользователи».

Есть хотя бы один зарегистрированный пользователь.

№	Действие пользователя	Действие системы
1	Нажимает на поле отвечающее за должность конкретного пользователя.	Отображает выпадающее меню содержащее список всех доступных должностей.
2	Назначает либо удаляет должность конкретному пользователю.	Запоминает отредактированные данные.
3	После окончания редактирования нажимает кнопку «Сохранить изменения».	Система сохраняет все сделанные изменения и добавляет их в БД.

Постусловие: отредактирована должность конкретного пользователя.

2.3.3 Диаграмма классов

UML диаграмма классов для подсистемы управления пользователями представлена на рисунке А.1, а ее описание в таблице Б.1. На рисунке 13 представлена укрупненная диаграмма классов.

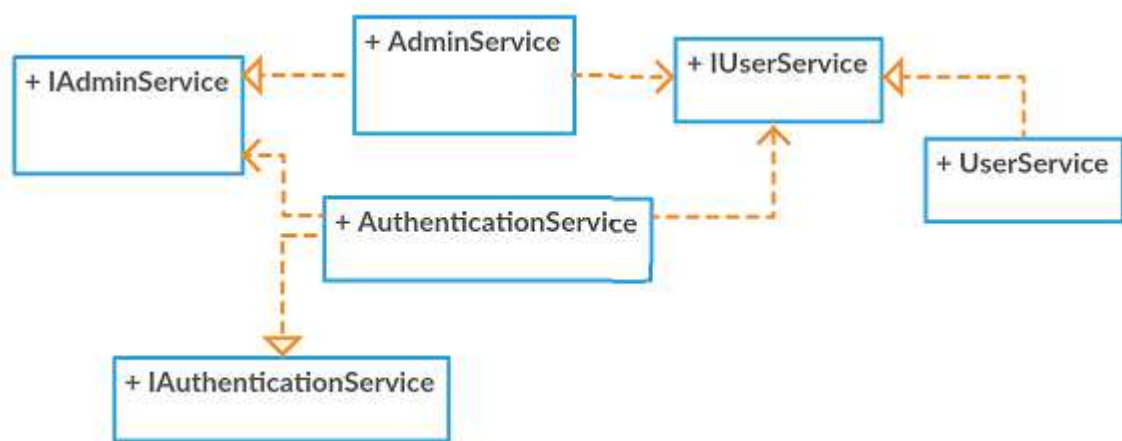


Рисунок 13 – Укрупненная диаграмма классов для подсистемы управления пользователями

2.4 Подсистема управления корпоративным календарем

2.4.1 Прецеденты

На рисунке 14 изображены прецеденты для подсистемы управления корпоративным календарем.

1. Просмотр календаря

Предусловие: менеджер или сотрудник авторизован.

№	Действие пользователя	Действие системы
1	Переходит на страницу «Календарь».	Открывает страницу «Календарь», подгружает все отмеченные даты из БД и отображает полученный календарь на странице.

Постусловие: загружена страница «Календарь».

2. Выделение даты на календаре

Предусловие: менеджер авторизован. Загружена страница «Календарь».

№	Действие пользователя	Действие системы
1	Нажимает в календаре дату, не отмеченную ранее.	Открывает диалоговое окно с формами выбора типа даты «Выходной» или «Рабочий» и ее описания.

2	По желанию заполняет описание события.	
3	Нажимает кнопку «Сохранить»	Выделяет день синим цветом если он «Рабочий» и красным, если «Выходной» и заносит данные о новой дате в БД. При наведении мышкой на отмеченную дату показывает краткое описание даты во всплывающей подсказке.

Постусловие: отмечена новая дата.

3. Загрузка выходных дней из файла

Предусловие: менеджер авторизован. Загружена страница «Календарь». Файл соответствует необходимому формату.

№	Действие пользователя	Действие системы
1	Выбирает год и нажимает на кнопку «Загрузить файл».	Открывает диалоговое окно для выбора загружаемого файла.
2	Выбирает файл.	Загружает файл на сервер. Полученные данные добавляются в БД. Все даты из файла считаются выходными, им устанавливается тип даты «Выходной» и они отмечаются на календаре красным цветом, описание остается пустым. При наведении мышкой на отмеченную дату показывает краткое описание даты во всплывающей подсказке.

Постусловие: отмечены новые даты.

4. Просмотр информации о дате

Предусловие: менеджер или сотрудник авторизован. Загружена страница «Календарь». Есть хотя бы одна отмеченная дата.

№	Действие пользователя	Действие системы
1	Нажимает в календаре отмеченную дату.	Открывает диалоговое окно, в котором указан тип даты («Выходной» или «Рабочий» день) и ее описание.

Постусловие: –.

5. Редактирование информации о дате

Предусловие: менеджер авторизован. Загружена страница «Календарь». Есть хотя бы одна отмеченная дата.

№	Действие пользователя	Действие системы
1	Нажимает правой кнопкой мыши отмеченную дату в календаре.	Открывает контекстное меню с опциями «Изменить» или «Удалить».
2	Выбирает опцию «Изменить»	Открывает диалоговое окно с формами выбора типа даты «Выходной» или «Рабочий» и ее описания.
3	Изменяет описание и тип по своему усмотрению.	
4	Нажимает кнопку «Сохранить»	Изменяет выбранную дату и заносит изменения в БД. Выделяет день синим цветом если он «Рабочий» и красным, если «Выходной». При наведении мышкой на отмеченную дату показывает краткое описание даты во всплывающей подсказке.

Постусловие: отредактирована информация о выбранной дате.

6. Удаление выделения с даты

Предусловие: менеджер авторизован. Загружена страница «корпоративный календарь». Есть хотя бы одна отмеченная дата.

№	Действие пользователя	Действие системы
1	Нажимает правой кнопкой мыши день в календаре, отмеченный выходным.	Открывает контекстное меню с опциями «Изменить» или «Удалить».
2	Выбирает опцию «Удалить»	Убирает выделение с выбранной даты и удаляет ее из БД. По умолчанию делает цвет белым. Удаляет всплывающее окно-подсказку.

Постусловие: удаляет выделение с даты.

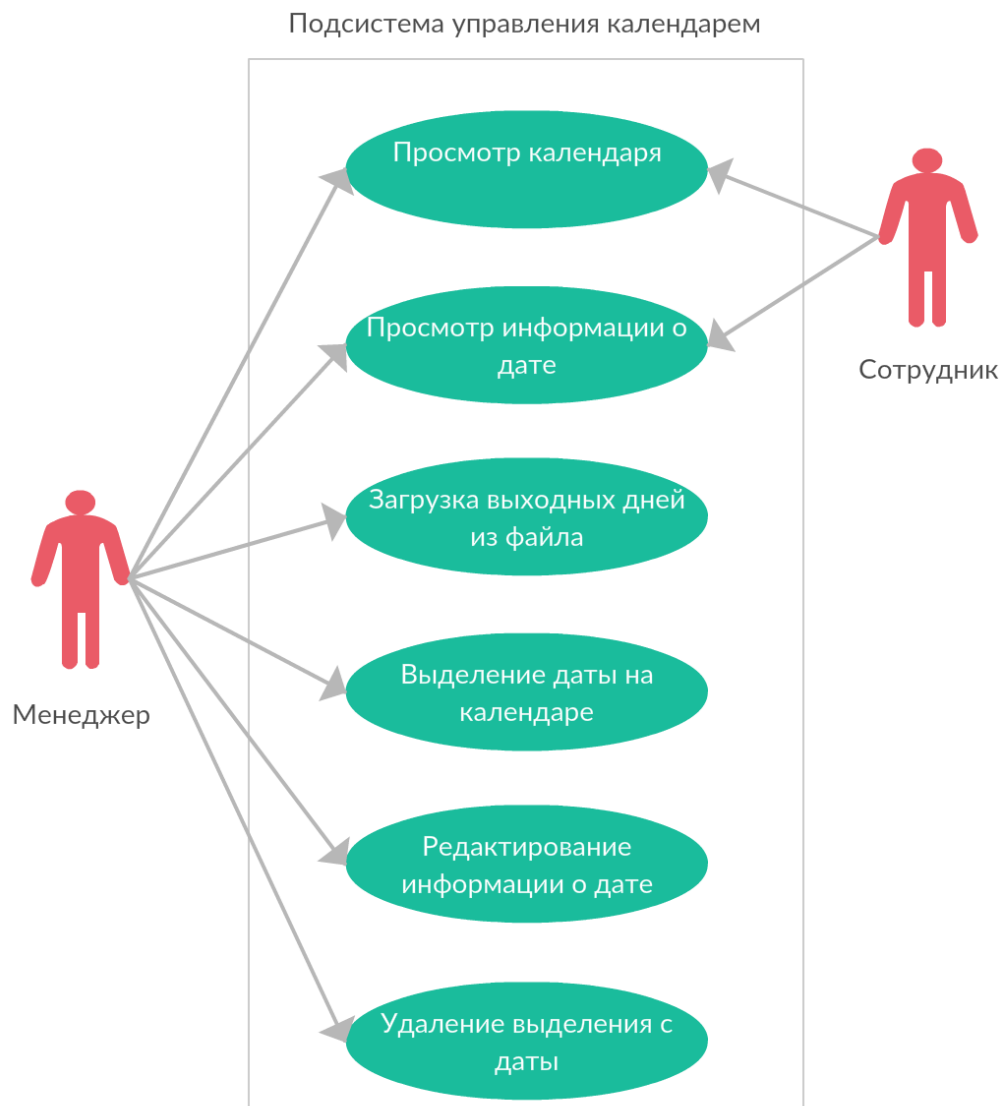


Рисунок 14 – Диаграмма прецедентов для подсистемы управления корпоративным календарем

2.4.2 Диаграмма классов

В таблице 5 реализовано описание классов, изображенных на рисунке 15.

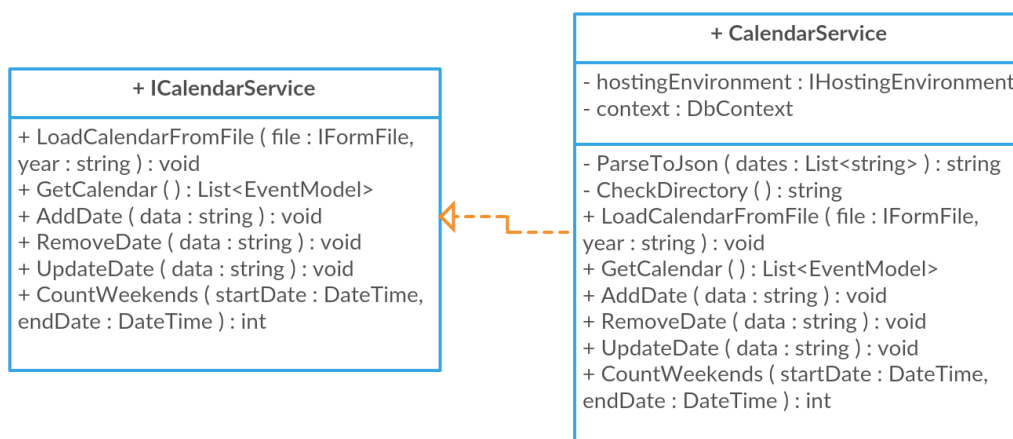


Рисунок 15 – UML диаграмма классов для подсистемы управления корпоративным календарем

Таблица 5 – Описание диаграммы классов для системы управления корпоративным календарем

Метод	Параметры	Описание
<i>CalendarService</i>		
<i>ParseToJson (dates)</i>	Входные: <i>dates</i> – список дат из <i>Excel</i> файла « <i>string</i> ». Выходные: выходным параметром функции является строка в формате <i>JSON</i>	Метод преобразует данные о датах, полученные из файла <i>Excel</i> в строку в формате <i>JSON</i> для последующего отображения этих дат на календаре.
<i>GetCalendar ()</i>	Входные: отсутствуют. Выходные: список объектов, использующийся для дальнейшей работы с датами « <i>List<EventModel></i> ».	Метод получает данные о датах из БД и преобразует их в удобный формат, для их последующей обработки другими методами системы.
<i>UpdateDate (data)</i>	Входные: <i>data</i> – строка в формате <i>JSON</i> хранящая в себе информацию о дате. Выходные: отсутствуют.	Метод обрабатывает входную <i>JSON</i> строку и обновляет дату в БД.

Продолжение таблицы 5

<i>CheckDirectory ()</i>	<p>Входные: отсутствуют.</p> <p>Выходные: путь к директории с загруженными файлами «string».</p>	<p>Метод используется при загрузке <i>Excel</i> файла с датами на сервер, он проверяет существует ли директория загрузки для файлов, если да, то возвращает путь к ней, если нет, то создает ее и возвращает путь к ней. Путь к директории занесен в файл конфигурации приложения.</p>
<i>LoadClendarFromFile (file, year)</i>	<p>Входные: <i>file</i> – файл, принятый от клиента «<i>IFormFile</i>»; <i>year</i> – год для которого нужно взять данные, параметр опциональный, и в случае если он не будет указан, то данные берутся для всех годов, что есть в файле «string».</p> <p>Выходные: отсутствуют.</p>	<p>Метод используется для разбора файла, пришедшего от клиента, преобразования данных в нужный формат формат и занесение их в БД. Использует выше описанные методы: «<i>ParseToJson</i>» и «<i>CheckDirectory</i>».</p>
<i>AddDate (data)</i>	<p>Входные: <i>data</i> – строка в формате <i>JSON</i> хранящая в себе информацию о дате.</p> <p>Выходные: отсутствуют.</p>	<p>Метод обрабатывает входную <i>JSON</i> строку и добавляет новую дату в БД.</p>
<i>RemoveDate (data)</i>	<p>Входные: <i>data</i> – строка в формате <i>JSON</i> хранящая в себе информацию о дате.</p> <p>Выходные: отсутствуют.</p>	<p>Метод обрабатывает входную <i>JSON</i> строку и удаляет дату из БД.</p>

Окончание таблицы 5

<i>CountWeekends</i> (<i>startDate</i> , <i>endDate</i>)	Входные: <i>startDate</i> – дата начала отсчета « <i>DateTime</i> »; <i>endDate</i> – дата окончания отсчета « <i>DateTime</i> ». Выходные: количество рабочих дней между двумя датами « <i>int</i> ».	Метод возвращает количество рабочих дней между двумя датами без учета выходных дней (Сб, Вс), если они не были помечены в календаре, как рабочие и дней, помеченных в календаре, как выходной день.
---	--	---

2.5 Вывод

В данной главе был рассмотрен процесс проектирования подсистем управления пользователями и управления корпоративным календарём.

Проектирование проходило в несколько этапов. Первые два этапа являются общими для обеих подсистем, на них были спроектированы архитектура и структура данных. На третьем и четвертом этапах были спроектированы сами подсистемы, а именно алгоритм аутентификации, авторизации и регистрации пользователей, диаграммы классов, а также диаграммы прецедентов с подробным описанием.

В результате сформировалась четкое представление будущих подсистем и теперь можно приступать к их реализации.

3 Практическая реализация

Практическая реализацию можно поделить на два этапа: реализация серверной части и клиентской.

3.1 Серверная часть

Серверная часть реализована на платформе *ASP.NET Core 2.0* при помощи объектно-ориентированного языка программирования *C#*. Для обращения к БД используется *ADO.NET Entity Framework (EF)*.

EF – представляет специальную объектно-ориентированную технологию на базе Фреймворка *.NET* предназначенную для работы с данными. Средства *ADO.NET* позволяют создавать подключения, команды и прочие объекты для взаимодействия с базами данных, а *Entity Framework* представляет собой более высокий уровень абстракции, который позволяет отстраниться от самой БД и работать с данными независимо от типа хранилища. Если на физическом уровне мы оперируем таблицами, индексами, первичными и внешними ключами, то на уровне, который нам предлагает *Entity Framework*, мы уже работаем с объектами. Для более простого и быстрого построения запросов к БД компания *Microsoft* создала так называемый язык интегрированных запросов или *LINQ* [11].

LINQ – простой и удобный язык запросов к источнику данных. В качестве источника данных может выступать объект, реализующий интерфейс *IEnumerable* (например, стандартные коллекции, массивы), набор данных *DataSet*, документ *XML*. Но вне зависимости от типа источника *LINQ* позволяет применить ко всем один и тот же подход для выборки данных [12].

В качестве СУБД используется *Microsoft SQL Server*. *MS SQL Server* – это платформа для решения критически важных задач в области хранения данных, обладающая высокой доступностью, повышенной производительностью и безопасностью. Решение представляет собой хорошо масштабируемый, полностью реляционный, быстродействующий сервер, способный обрабатывать большие объемы данных для клиент-серверных приложений. Большая

производительность *MS SQL Server* обеспечивается новыми технологиями работы с памятью, что помогает ускорять существующие и создавать новые сценарии работы с данными. Кроме того, *SQL Server* позволяет использовать новые гибридные облачные решения и пользоваться новыми преимуществами облачных вычислений. Расширенные функции безопасности, в сочетании со встроенными, удобными для использования инструментами и управляемым доступом к данным, позволяют решать множество задач, связанных с хранением данных. основной используемый язык запросов — *Transact-SQL* [13].

3.2 Клиентская часть

Клиентская часть подсистем разработана с использованием *HTML*, *CSS*, *JavaScript* и движка представлений *Razor Pages*, который предоставляет разработчику возможность писать логику отображения контента на языке *C#* внутри *HTML* страницы, что повышает гибкость и быстроту разработки пользовательского интерфейса. В качестве фреймворка для разработки графического интерфейса используется *Telerik Kendo UI*.

Telerik Kendo UI – это *JavaScript*-фреймворк, который может быть использован для создания веб и мобильных приложений с использованием *HTML5* и *JavaScript*. Он обеспечивает набор мощных виджетов, которые могут быть связаны с источниками данных. В разрабатываемых подсистемах данный фреймворк используется для реализации всплывающих окон, списка пользователей, а также уведомлений [14].

Результаты разработки клиентской части для подсистемы управления пользователями и подсистемы управления корпоративным календарем представлены на рисунках 16 и 17.

Список пользователей системы

<input checked="" type="checkbox"/> SAVE CHANGES <input type="checkbox"/> CANCEL CHANGES						
Имя	Фамилия	Почта	Должность	Баланс	Роль	Согласующие
Anna	Vulk	sav_em1@dmdevelopment.ru	Нет должности	28	Сотрудник	
Natalia	Corfield	sav_conf@dmdevelopment.ru	Нет должности	28	Подтверждающий, Сотрудник	
Viktoria	Bistroko	sav_ag2@dmdevelopment.ru	Нет должности	28	Согласующий, Сотрудник	
Ivan	Alexandrov	sav_ag1@dmdevelopment.ru	Нет должности	28	Согласующий, Сотрудник	
Vladimir	Vladimirov	sav_admin@dmdevelopment.ru	Нет должности	28	Администратор	

Отображены записи 1 - 5 из 6

© 2018 - Система учета отпусков и отгулов

Рисунок 16 – Скриншот клиентской части для подсистемы управления пользователями

Производственный календарь

Выберите год: 2019 Загрузить файл с датами: Загрузить файл

<	2017	2018	2019	2020	2021	>
Январь						
Пн	Вт	Ср	Чт	Пт	Сб	Вс
	1	2	3	4	5	6
7	8	9	10	11	12	13
14	15	16	17	18	19	20
21	22	23	24	25	26	27
28	29	30	31			
Февраль						
Пн	Вт	Ср	Чт	Пт	Сб	Вс
				1	2	3
4	5	6	7	8	9	10
11	12	13	14	15	16	17
18	19	20	21	22	23	24
25	26	27	28			
Март						
Пн	Вт	Ср	Чт	Пт	Сб	Вс
				1	2	3
4	5	6	7	8	9	10
11	12	13	14	15	16	17
18	19	20	21	22	23	24
25	26	27	28	29	30	31
Апрель						
Пн	Вт	Ср	Чт	Пт	Сб	Вс
1	2	3	4	5	6	7
8	9	10	11	12	13	14
15	16	17	18	19	20	21
22	23	24	25	26	27	28
29	30					
Май						
Пн	Вт	Ср	Чт	Пт	Сб	Вс
	1	2	3	4	5	
6	7	8	9	10	11	12
13	14	15	16	17	18	19
20	21	22	23	24	25	26
27	28	29	30	31		
Июнь						
Пн	Вт	Ср	Чт	Пт	Сб	Вс
					1	2
3	4	5	6	7	8	9
10	11	12	13	14	15	16
17	18	19	20	21	22	23
24	25	26	27	28	29	30
Июль						
Пн	Вт	Ср	Чт	Пт	Сб	Вс
1	2	3	4	5	6	7
8	9	10	11	12	13	14
15	16	17	18	19	20	21
22	23	24	25	26	27	28
29	30	31				
Август						
Пн	Вт	Ср	Чт	Пт	Сб	Вс
						1
2	3	4	5	6	7	8
9	10	11	12	13	14	15
16	17	18	19	20	21	22
23	24	25	26	27	28	29
30	31					
Сентябрь						
Пн	Вт	Ср	Чт	Пт	Сб	Вс
						1
2	3	4	5	6	7	8
9	10	11	12	13	14	15
16	17	18	19	20	21	22
23	24	25	26	27	28	29
30						
Октябрь						
Пн	Вт	Ср	Чт	Пт	Сб	Вс
1	2	3	4	5	6	
7	8	9	10	11	12	13
14	15	16	17	18	19	20
21	22	23	24	25	26	27
28	29	30	31			
Ноябрь						
Пн	Вт	Ср	Чт	Пт	Сб	Вс
				1	2	3
4	5	6	7	8	9	10
11	12	13	14	15	16	17
18	19	20	21	22	23	24
25	26	27	28	29	30	
Декабрь						
Пн	Вт	Ср	Чт	Пт	Сб	Вс
						1
2	3	4	5	6	7	8
9	10	11	12	13	14	15
16	17	18	19	20	21	22
23	24	25	26	27	28	29
30	31					

© 2018 - Система учета отпусков и отгулов

Рисунок 17 – Скриншот клиентской части для подсистемы управления корпоративным календарем

3.3 Тестирование

Тестирование – этап разработки программного обеспечения, реализующий проверку соответствия между реальным и ожидаемым поведением продукта на конечном наборе тестов. Также тестирование является одной из техник контроля качества программного обеспечения и ставит перед собой следующие цели:

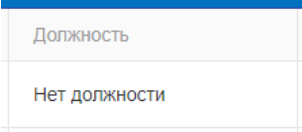
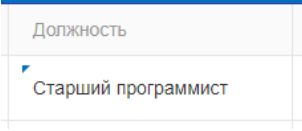
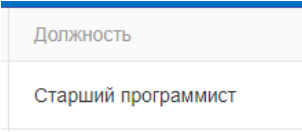
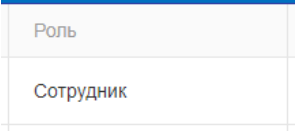
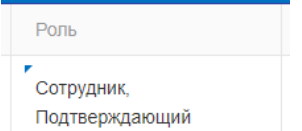
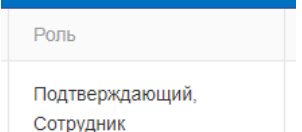
- Повысить вероятность того, что приложение, предназначенное для тестирования, будет работать правильно при любых обстоятельствах.
- Повысить вероятность того, что приложение, предназначенное для тестирования, будет соответствовать всем описанным требованиям.
- Предоставить актуальную информацию о состоянии продукта на данный момент.

Тестирование может быть автоматическим и ручным, рассмотрим ручное тестирование на примере тест-кейсов для подсистемы управления пользователями, представленных в таблице 6.

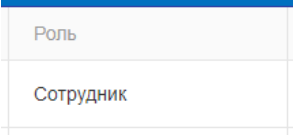
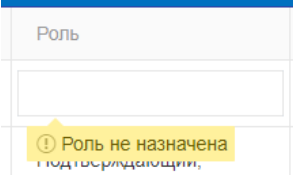

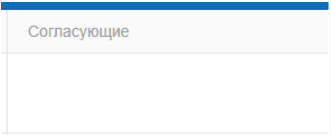
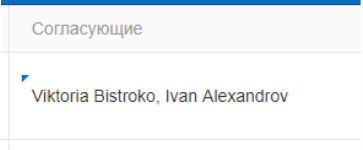
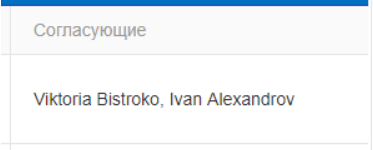
Таблица 6 – Тест-кейсы для подсистемы управления пользователями

Действие	Ожидаемый результат	Результат теста
1. Проверка отображения страницы		
- Открыть страницу «Пользователи».	- Страница успешно загружена. - Отображается таблица содержащая информацию о пользователях.	- Система корректно реагирует на действие пользователя. - Реальный результат соответствует ожидаемому, рисунок 16. - Тест пройден.
2. Проверка системы оповещений		
- Сохраняем данные.	- Выдает оповещение о сохранении изменений.	- Сохраняем данные, выдается всплывающее окно-оповещение: 
- Отменяем изменения.	- Выдает оповещение об отмене изменений.	- Отменяем изменения, выдается всплывающее окно-оповещение: 

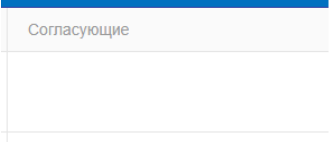
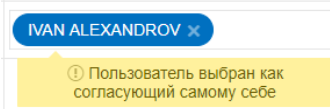

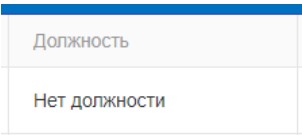
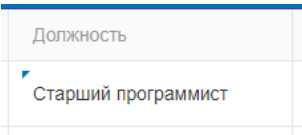
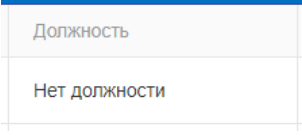
Продолжение таблицы 6

3. Проверка изменения должности пользователя		
<ul style="list-style-type: none"> - Изменить должность пользователя. - Сохранить изменения. 	<ul style="list-style-type: none"> - При нажатии на поле, содержащее должность, открывается выпадающее меню с возможными должностями. - Выбранная должность заменяет текущую должность пользователя. - Поле помечается как измененное. - После сохранения пометка об изменении удаляется. - Должность изменена. 	<ul style="list-style-type: none"> - Исходное состояние:  - Выбираем должность, поле помечается, как измененное:  - Сохраняем изменение, пометка с поля снимется, должность изменена:  - Тест пройден.
4. Проверка назначения ролей		
<ul style="list-style-type: none"> - Изменить роли пользователя. - Сохранить изменения. 	<ul style="list-style-type: none"> - При нажатии на поле, содержащее должность, открывается выпадающее меню с возможными ролями. - Выбранная роль добавляется к текущей роли пользователя. - Поле помечается как измененное. - После сохранения пометка об изменении удаляется. - Роль добавлена. 	<ul style="list-style-type: none"> - Исходное состояние:  - Выбираем роль, поле помечается, как измененное:  - Сохраняем изменение, пометка с поля снимется, новая роль успешно добавлена:  - Тест пройден.

Продолжение таблицы 6

<p>- Удалить все роли пользователя.</p> <p>- Пытаемся сохранить изменения.</p>	<p>- Выдается предупреждающее сообщение.</p> <p>- Запрещается всяческое изменение или сохранение данных, пока у пользователя отсутствует роль.</p>	<p>- Исходное состояние:</p>  <p>- Очищаем список ролей, выдается предупреждение:</p>  <p>- Заблокировано изменение и сохранение данных. Активна только кнопка отмены изменений и поле, в котором вызвана ошибка:</p>  <p>- Тест пройден.</p>
<p>5. Проверка назначения согласующих</p>		
<p>- Назначить согласующих.</p> <p>- Сохранить изменения.</p>	<p>- При нажатии на поле, содержащее согласующих, открывается выпадающее меню с возможными согласующими.</p> <p>- Выбранный согласующий добавляется к текущим согласующим пользователя.</p> <p>- Поле помечается как измененное.</p> <p>- После сохранения пометка об изменении удаляется.</p> <p>- Согласующий назначен.</p>	<p>- Исходное состояние:</p>  <p>- Выбираем согласующего, поле помечается, как измененное:</p>  <p>- Сохраняем изменение, пометка с поля снимется, новый согласующий успешно добавлен:</p>  <p>- Тест пройден.</p>

Окончание таблицы 6

<ul style="list-style-type: none"> - Удалить всех согласующих пользователя. - Пытаемся сохранить изменения. 	<ul style="list-style-type: none"> - Ожидаемая реакция идентична предыдущему случаю. 	<ul style="list-style-type: none"> - Результат теста идентичен предыдущему. - Тест пройден
<ul style="list-style-type: none"> - Назначаем согласующего самого себе. - Пытаемся сохранить изменения. 	<ul style="list-style-type: none"> - Выдается предупреждающее сообщение. - Запрещается всяческое изменение или сохранение данных, пока проблема не будет устранена. 	<ul style="list-style-type: none"> - Исходное состояние:  - Выбираем согласующего, самого себе:  - Заблокировано изменение и сохранение данных. Активна только кнопка отмены изменений и поле, в котором вызвана ошибка:  - Тест пройден.
6. Проверка реакции таблицы на кнопку отмену изменений		
<ul style="list-style-type: none"> - Изменяем какой-нибудь параметр. - Отменяем изменение. 	<ul style="list-style-type: none"> - Поле помечается как измененное. - После отмены изменений вся информация возвращается в первоначальное состояние. 	<ul style="list-style-type: none"> - Исходное состояние:  - Выбираем должность, поле помечается, как измененное:  - Отменяем изменение, пометка с поля снимется, данные вернулись в первоначальное состояние:  - Тест пройден.

По итогам тестирования разработанные подсистемы имеют хороший показатель надежности и полностью соответствуют своему назначению.

3.4 Развертывание ПО

Развертывание программного обеспечения – это ряд действий, направленных на приведение программной системы в рабочее состояние. Данный процесс является частью жизненного цикла программного обеспечения. В целом процесс развертывания состоит из взаимосвязанных действий с возможностью перехода между ними. Поскольку каждая программная система является уникальной, трудно предсказать все процессы и процедуры во время развертывания, однако есть ряд основных операций, которые используются повсеместно:

- Компиляцию кода – преобразование программы, написанной на высокоуровневом языке в эквивалентную программу на низкоуровневом языке, близком к машинному коду;
- Подтягивание зависимостей проекта;
- Выгрузка кода на сервер;
- Выполнение настроечных операций на сервере, например, настройка аутентификации и т.д.

В случае с разработанной системой использовались автоматические сборка и развертывание, реализованные средствами продукта компании *Microsoft, Team Foundation Server (TFS)*.

TFS - представляет собой комплексное решение, объединяющее в себе систему управления версиями, сбор данных, построение отчетов, отслеживание статусов и изменений по проекту, предназначенное для совместной разработки программного обеспечения. В качестве сервера для разрабатываемого продукта был использован *Windows Server 2012 R2* с установленным на него *IIS* [15].

IIS (Internet Information Services) — набор сервисов от компании *Microsoft* для работы веб-сервера и других интернет служб. *IIS* устанавливается на сервер и работает с протоколами *HTTP/HTTPS, POP3, SMTP, FTP, NNTP* [16].

На рисунках 15 и 16 представлены отчеты о сборке и развертывании проекта средствами *TFS*.

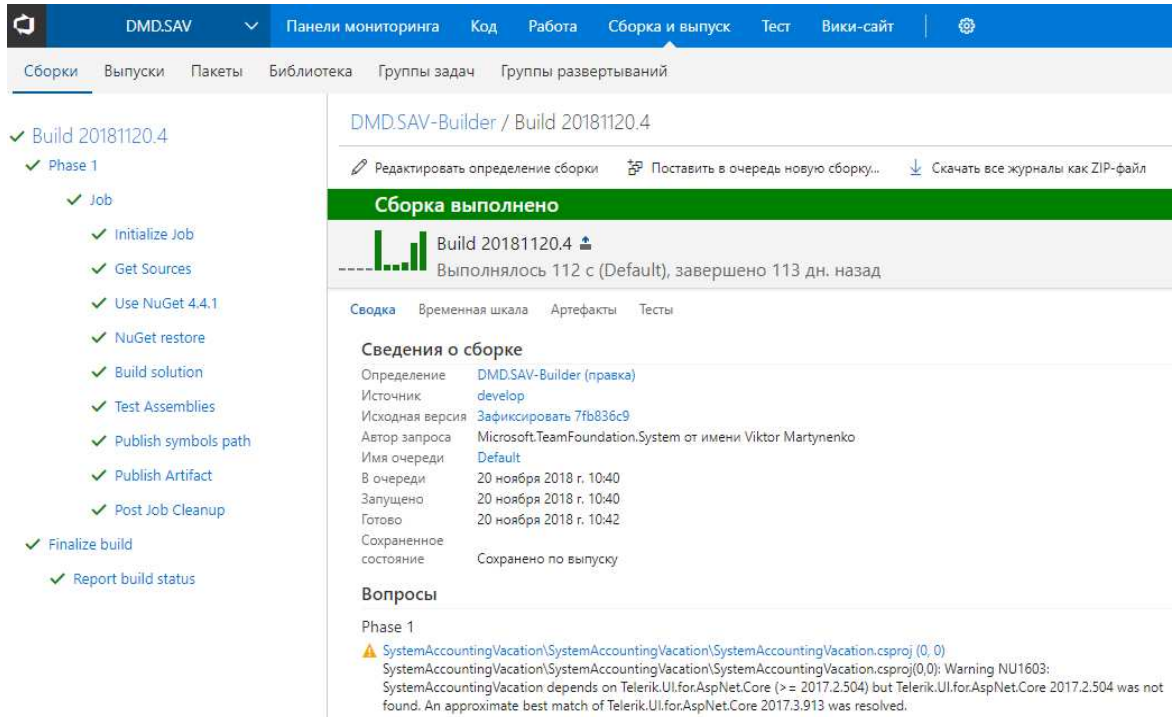


Рисунок 18 – Отчет о сборке проекта

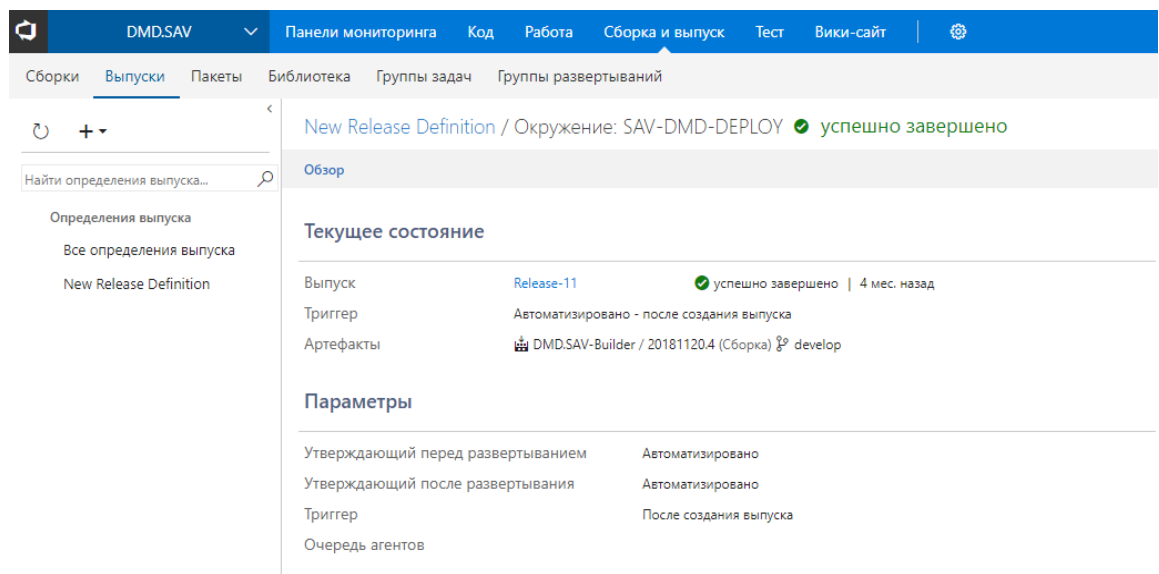


Рисунок 19 – Отчет о развертывании проекта на сервере

3.5 Вывод

По итогам проделанной работы были разработаны подсистема управления пользователями, и подсистема управления корпоративным календарем которые полностью соответствуют своему назначению. Обе подсистемы были протестированы на конечном наборе тестов и развернуты на реальном сервере.

ЗАКЛЮЧЕНИЕ

Разработанные в ходе работы подсистемы стали частью проекта «Система учета рабочего времени», который был одобрен заказчиком и успешно внедрен на производстве, это подтверждает акт о внедрении, представленный в приложении В. В настоящее время продолжаются работы по расширению функционала системы.

Полученная в итоге система призвана обеспечить уменьшение количества бумажной отчетности, увеличение эффективности распределения рабочего времени и скорости взаимодействия между сотрудниками.

В ходе разработки подсистем были улучшены навыки по следующим областям:

- Проектирование
 - *UML* - диаграммы
 - *ER* – диаграммы
 - Диаграммы прецедентов
 - Алгоритмические схемы
- Практическая реализация
 - *C#*
 - *JavaScript*
 - *HTML*
 - *CSS*
 - *SQL*
 - *LINQ*
 - Тестирование ПО
 - Развертывание ПО

СПИСОК СОКРАЩЕНИЙ

В данной выпускной квалификационной работе используются следующие сокращения:

ASP – Active Server Pages.

IIS – Internet Information Services.

TFS – Team Foundation Server.

MVC – Model-View-Controller.

SQL – Structured Query Language.

HTTP – HyperText Transfer Protocol.

API – Application Programming Interface.

ER – Entity-Relationship.

EF – Entity-Framework

HTML – HyperText Markup Language.

LINQ – Language Integrated Query.

CSS – Cascading Style Sheets.

БД – База Данных.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. ASP.NET Core - новая эпоха в развитии ASP.NET [Электронный ресурс]: Metanit // Информационный портал о различных языках и технологиях программирования. – Режим доступа: <https://metanit.com/sharp/aspnet5/1.1.php>.
2. Подсистема «Администрирование» [Электронный ресурс]: ИТП «Град» // Технологические решения, разработанные ООО «ИТП «Град». – Режим доступа: <https://itpgrad.ru/node/682>.
3. 1С: Предприятие 8 [Электронный ресурс]: 1С // Система программ 1С. – Режим доступа: <http://v8.1c.ru/>.
4. Документация UMI.CMS [Электронный ресурс]: UMI.CMS // Система управления веб-сайтами. – Режим доступа: <https://docs.umi-cms.ru/>.
5. Производственный календарь в 1С 8.3 Бухгалтерия [Электронный ресурс]: GoodWill // Внедрение и продажа 1С. – Режим доступа: <https://programmist1s.ru/proizvodstvennyiy-kalendar-1s-na-god/>.
6. Производственный календарь 2019 [Электронный ресурс]: AndySoftware // Софт для отдела кадров и системы автоматизации. – Режим доступа: <https://andysoftware.com/ru/pc/>.
7. Аутентификация пользователя [Электронный ресурс]: Life-prog // Сайт о различных языках и технологиях программирования. – Режим доступа: https://life-prog.ru/view_programmer.php?id=158&page=16.
8. Обзор доменных служб Active Directory [Электронный ресурс]: Microsoft Docs // Хранилище документации Майкрософт для пользователей, разработчиков и ИТ-специалистов. – Режим доступа: <https://docs.microsoft.com/ru-ru/windows-server/identity/ad-ds/get-started/virtual-dc/active-directory-domain-services-overview>.

9. Kerberos [Электронный ресурс]: BMSTU // Портал национальной электронной библиотеки им. Н. Э. Баумана. – Режим доступа: <https://ru.bmstu.wiki/Kerberos>.

10. NTLM [Электронный ресурс]: BMSTU // Портал национальной электронной библиотеки им. Н. Э. Баумана. – Режим доступа: [https://ru.bmstu.wiki/NTLM_\(NT_LAN_Manager\)](https://ru.bmstu.wiki/NTLM_(NT_LAN_Manager)).

11. Введение в Entity Framework [Электронный ресурс]: Metanit // Информационный портал о различных языках и технологиях программирования. – Режим доступа: <https://metanit.com/sharp/entityframework/1.1.php>.

12. LINQ [Электронный ресурс]: Metanit // Информационный портал о различных языках и технологиях программирования. – Режим доступа: <https://metanit.com/sharp/tutorial/15.1.php>.

13. Microsoft SQL Server [Электронный ресурс]: Navicon // Аналитические решения в сфере IT. – Режим доступа: <https://navicongroup.ru/platforms/4025/>.

14. Telerik UI [Электронный ресурс]: Telerik // Telerik UI official webpage. – Режим доступа: <https://docs.telerik.com>.

15. Team Foundation Server [Электронный ресурс]: Visual Studio // Лучшие в своем классе средства для разработчиков. – Режим доступа: <https://visualstudio.microsoft.com/ru/tfs/?rr=https%3A%2F%2Fru.wikipedia.org%2>.

16. IIS [Электронный ресурс]: IIS // A flexible & easy-to-manage web server. – Режим доступа: <https://www.iis.net/>.

ПРИЛОЖЕНИЕ А

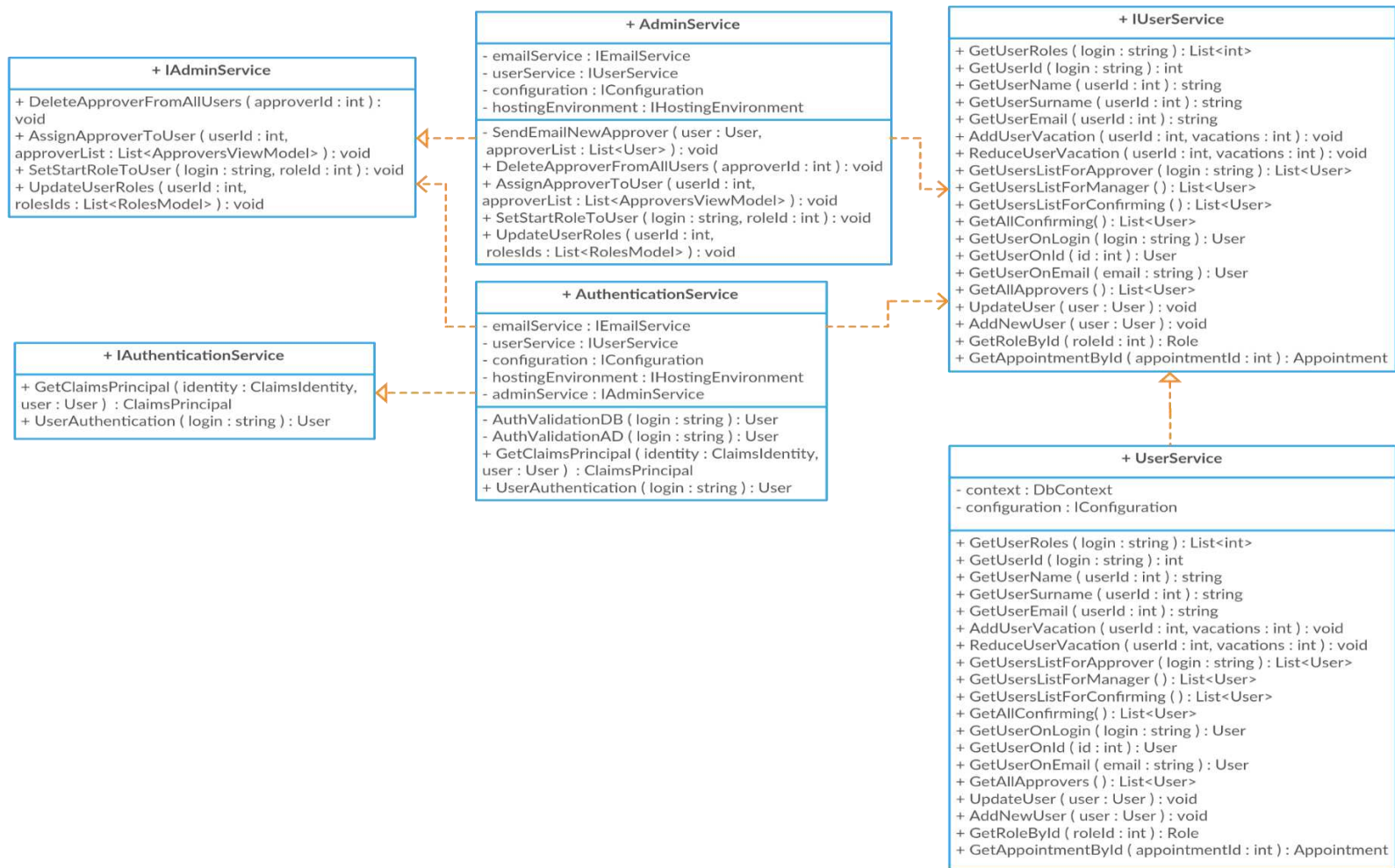


Рисунок А.1 – Диаграмма классов для подсистемы управления пользователями

ПРИЛОЖЕНИЕ Б

Таблица Б.1 – Описание диаграммы классов для подсистемы управления пользователями

Метод	Параметры	Описание
<i>AdminService</i>		
<i>SendEmailNewApprover</i> (<i>user, approverList</i>)	Входные: <i>user</i> – объект « <i>User</i> », содержащий информацию о пользователе; <i>approverList</i> – список объектов « <i>User</i> », содержащих информации о пользователях. Выходные: отсутствуют.	Метод генерирует оповещение для согласующих из списка, что к ним прикреплен новый сотрудник. Для отправки оповещения используются методы сервиса электронной почты.
<i>DeleteApproverFromAllUsers</i> (<i>approverId</i>)	Входные: <i>approverId</i> – идентификатор согласующего « <i>int</i> ». Выходные: отсутствуют.	Метод реализует удаление согласующего у всех сотрудников, которые к нему прикреплены. Используется в случае, если пользователю системы убрали роль согласующего, а у него остались прикрепленные сотрудники.
<i>AssignApproverToUser</i> (<i>userId, approverList</i>)	Входные: <i>user</i> – объект « <i>User</i> », содержащий информацию о пользователе; <i>approverList</i> – список объектов « <i>User</i> », содержащих информации о пользователях. Выходные: отсутствуют	Метод используется для прикрепления согласующих из списка к определенному сотруднику.

Продолжение таблицы Б1

<i>SetStartRoleToUser</i> (<i>login</i> , <i>roleId</i>)	Входные: <i>login</i> – имя учетной записи пользователя « <i>string</i> »; <i>roleId</i> – идентификатор роли « <i>int</i> ». Выходные: отсутствуют	Метод устанавливает начальную роль при регистрации нового пользователя.
<i>UpdateUserRoles</i> (<i>userId</i> , <i>rolesIds</i>)	Входные: <i>userId</i> – идентификатор пользователя « <i>int</i> »; <i>rolesIds</i> – список идентификаторов ролей « <i>List<int></i> ». Выходные: отсутствуют	Метод используется для обновления списка ролей пользователя системы.
<i>AuthenticationService</i>		
<i>AuthValidationDB</i> (<i>login</i>)	Входные: <i>login</i> – имя учетной записи пользователя « <i>string</i> ». Выходные: объект « <i>User</i> », содержащий информацию о пользователе.	Метод реализует поиск пользователя в БД по логину. Если пользователь найден, то возвращается объект, содержащий информацию о пользователе, если нет, то возвращается пустой объект.
<i>AuthValidationAD</i> (<i>login</i>)	Входные: <i>login</i> – имя учетной записи пользователя « <i>string</i> ». Выходные: объект « <i>User</i> », содержащий информацию о пользователе.	Метод реализует поиск пользователя в <i>Active Directory</i> по логину. Если пользователь найден, то возвращается объект, содержащий информацию о пользователе, если нет, то возвращается пустой объект.
<i>GetRoleById</i> (<i>roleId</i>)	Входные: <i>roleId</i> – идентификатор роли « <i>int</i> ». Выходные: объект типа « <i>Role</i> », содержащий информацию об определенной роли пользователя.	Метод выполняет поиск роли в БД по ее идентификатору и в случае успеха возвращает объект типа « <i>Role</i> », иначе возвращается пустой объект.

Продолжение таблицы Б1

<p><i>UserAuthentication (login)</i></p>	<p>Входные: <i>login</i> – имя учетной записи пользователя «string». Выходные: объект «User», содержащий информацию о пользователе.</p>	<p>Метод реализующий аутентификацию пользователя в системе использует методы «AuthValidationAD» и «AuthValidationDB». В случае если метод «AuthValidationDB» вернул не пустой объект работа функции завершается, и она возвращает этот объект в контроллер. В противном случае вызывается метод «AuthValidationAD», если он возвращает не пустой объект, то происходит регистрация пользователя в системе и добавление его в БД, после чего информация о пользователе передается в контроллер, иначе пользователю с таким именем учетной записи будет отказано в доступе.</p>
<p><i>GetClaimsPrincipal(identity, user)</i></p>	<p>Входные: <i>identity</i> – экземпляр «удостоверения» пользователя «ClaimsIdentity»; <i>user</i> – пользователь для которого вызван данный метод «User». Выходные: объект, представляющий из себя реализацию «удостоверения» пользователя в системе «ClaimsIdentity».</p>	<p>Метод создающий объект, который является «удостоверением» пользователя в системе, в нем хранятся все роли пользователя, а также его имя учетной записи. Полученное «удостоверение» используется системой, для выдачи разрешений пользователю на использование ресурсов.</p>

Продолжение таблицы Б1

<i>UserService</i>		
<i>GetUserRoles (login)</i>	Входные: <i>login</i> – имя учетной записи пользователя «string». Выходные: список идентификаторов ролей «List<int>».	Метод, который в случае успеха возвращает идентификаторы всех ролей пользователя с заданным именем учетной записи иначе возвращается пустой объект.
<i>GetUserId (login)</i>	Входные: <i>login</i> – имя учетной записи пользователя «string». Выходные: идентификатор пользователя «int».	Метод, который в случае успеха возвращает идентификатор пользователя с заданным именем учетной записи, иначе возвращается пустой объект.
<i>GetUserName (userId)</i>	Входные: <i>userId</i> – идентификатор пользователя «int». Выходные: имя пользователя «string».	Метод возвращает имя пользователя с заданным идентификатором, в случае успеха, иначе возвращается пустой объект.
<i>GetUserSurname (userId)</i>	Входные: <i>userId</i> – идентификатор пользователя «int». Выходные: фамилия пользователя «string».	Метод возвращает фамилию пользователя с заданным идентификатором, в случае успеха, иначе возвращается пустой объект.
<i>GetUserEmail (userId)</i>	Входные: <i>userId</i> – идентификатор пользователя «int». Выходные: адрес электронной почты пользователя «string».	Метод возвращает адрес электронной почты пользователя с заданным идентификатором, в случае успеха, иначе возвращается пустой объект.
<i>AddUserVacation (userId, vacations)</i>	Входные: <i>userId</i> – идентификатор пользователя «int»; <i>vacations</i> – количество отпускных «int». Выходные: отсутствуют.	Метод увеличивающий количество отпускных пользователя.
<i>ReduceUserVacation (userId, vacations)</i>	Входные: <i>userId</i> – идентификатор пользователя «int»; <i>vacations</i> – количество отпускных «int». Выходные: отсутствуют.	Метод уменьшающий количество отпускных пользователя.

Продолжение таблицы Б1

<p><i>GetUsersListForApprover</i> (<i>login</i>)</p>	<p>Входные: <i>login</i> – имя учетной записи согласующего «<i>string</i>». Выходные: список сотрудников «<i>List<User></i>».</p>	<p>Метод формирует список сотрудников с несогласованными заявками, которые прикреплены к согласующему с данным именем учетной записи и возвращает его в случае успеха, иначе возвращается пустой объект.</p>
<p><i>GetUsersListForManager</i> ()</p>	<p>Входные: отсутствуют. Выходные: список сотрудников «<i>List<User></i>».</p>	<p>Метод формирует список сотрудников для менеджера, у которых есть хотя бы одна подтвержденная или выполненная заявка и возвращает его в случае успеха, иначе возвращается пустой объект.</p>
<p><i>GetUsersListForConfirming</i> ()</p>	<p>Входные: отсутствуют. Выходные: список сотрудников «<i>List<User></i>».</p>	<p>Метод формирует список сотрудников для подтверждающего, у которых есть хотя бы одна согласованная заявка и возвращает его в случае успеха, иначе возвращается пустой объект.</p>
<p><i>GetAllConfirming</i> ()</p>	<p>Входные: отсутствуют. Выходные: список подтверждающих «<i>List<User></i>».</p>	<p>Метод формирует список пользователей системы, которые имеют роль «подтверждающий» и возвращает его в случае успеха, иначе возвращается пустой объект.</p>
<p><i>GetUserOnLogin</i> (<i>login</i>)</p>	<p>Входные: <i>login</i> – имя учетной записи пользователя «<i>string</i>». Выходные: пользователь «<i>User</i>».</p>	<p>Метод выполняет поиск в БД пользователя по имени его учетной записи и в случае успеха возвращает объект, содержащий необходимую информацию, иначе возвращается пустой объект.</p>

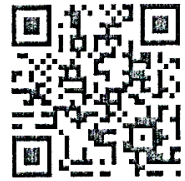
Окончание таблицы Б1

<i>GetUserOnId (userId)</i>	Входные: <i>userId</i> – идентификатор пользователя « <i>int</i> ». Выходные: пользователь « <i>User</i> ».	Метод выполняет поиск в БД пользователя по его идентификатору и в случае успеха возвращает объект, содержащий необходимую информацию, иначе возвращается пустой объект.
<i>GetUserOnEmail (email)</i>	Входные: <i>email</i> – электронная почта пользователя « <i>string</i> ». Выходные: пользователь « <i>User</i> ».	Метод выполняет поиск в БД пользователя по его электронной почте и в случае успеха возвращает объект, содержащий необходимую информацию, иначе возвращается пустой объект.
<i>GetAllApprovers ()</i>	Входные: отсутствуют. Выходные: список подтверждающих « <i>List<User></i> ».	Метод формирует список пользователей системы, которые имеют роль «согласующий» и возвращает его в случае успеха, иначе возвращается пустой объект.
<i>UpdateUser (user)</i>	Входные: <i>user</i> – объект типа « <i>User</i> », содержащий информацию о пользователе. Выходные: отсутствуют.	Метод выполняет обновление данных о пользователе в БД.
<i>AddNewUser (user)</i>	Входные: <i>user</i> – объект типа « <i>User</i> », содержащий информацию о пользователе. Выходные: отсутствуют.	Метод выполняет добавление в БД записи о новом пользователе.
<i>GetAppointmentById (appointmentId)</i>	Входные: <i>appointmentId</i> – идентификатор должности « <i>int</i> ». Выходные: объект типа « <i>Appointment</i> », содержащий информацию об определенной должности.	Метод выполняет поиск должности в БД по ее идентификатору и в случае успеха возвращает объект типа « <i>Appointment</i> », иначе возвращается пустой объект.

ПРИЛОЖЕНИЕ В



Общество с ограниченной ответственностью
«Диджитал Майнд Девелопмент» (ООО «ДМД»)
ОГРН 1052465119167 ИНН 1052465119167
Юридический адрес: 660062, г. Красноярск, ул. Высотная д.2, стр.8,
пом.6, офис 317
Почтовый адрес: 660062, г. Красноярск, ул. Высотная д.2, стр.8,
пом.6, офис 317
тел. +7 (391)29 63 778, +7 (391)20 50 778
e-mail: dmd@dmdevelopment.ru
<http://www.dmdevelopment.ru>



Исх. № 190517-1
От 17.05.2019 г.

Акт внедрения

Настоящим актом подтверждаем, что результаты выпускной квалификационной работы Мартыненко Виктора Олеговича на тему: «Автоматизированная система учета рабочего времени. Подсистемы управления пользователями и управления производственным календарем» внедрены в производственный процесс в ООО «Диджитал Майнд Девелопмент».

Назначении системы:

- автоматизация работы по организации рабочего времени;
- сокращение времени на подготовку отчетности;
- обеспечение возможности электронного учета отгулов и отпусков;
- обеспечение возможности электронного ведения производственного календаря;

Показатели эффективности достигаемые за счет внедрения системы:

- сокращение сроков подготовки отчетов на 20%;
- уменьшение трудозатрат при учете рабочего времени на 30%;
- общее снижение количества ошибок при учете рабочего времени на 30%.

Директор ООО «ДМД»
А.М.Епихин



17.05.2019 г.

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

Институт космических и информационных технологий
Кафедра вычислительной техники

УТВЕРЖДАЮ

Заведующий кафедрой

О. В. Непомнящий

подпись инициалы, фамилия

«27» 06 2019 г.

БАКАЛАВРСКАЯ РАБОТА

09.03.01 Информатика и вычислительная техника

Автоматизированная система учёта рабочего времени.
Подсистема управления пользователями и управления корпоративным
календарем

Руководитель	<u>Н. Ю. Сиротина</u> подпись, дата	<u>доцент, канд. техн. наук</u> должность, учёная степень	<u>Н. Ю. Сиротина</u> инициалы, фамилия
Выпускник	<u>В. О. Мартыненко</u> подпись, дата		<u>В. О. Мартыненко</u> инициалы, фамилия
Нормоконтролер	<u>В. И. Иванов</u> подпись, дата <u>26.06.19</u>	<u>доцент, канд. техн. наук</u> должность, учёная степень	<u>В. И. Иванов</u> инициалы, фамилия

Красноярск 2019