ФГБОУ ВО «Ивановский государственный

энергетический университет имени В.И. Ленина»

Факультет информатики и вычислительной техники

Кафедра программного обеспечения компьютерных систем

Система управления задачами

Пояснительная записка к курсовому проекту по дисциплинам «Проектирование и архитектура ПС», «Конструирование»

Выполнил:

Ст. гр. 4-42 Потонов Г.Е. \_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_

(подпись) (дата)

Руководители:

Доцент кафедры ПОКС Игнатьев Е. Б. \_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_

(подпись) (дата)

Старш. преподаватель каф. ПОКС Фомина О.В. \_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_

(подпись) (дата)

Иваново, 2022

# Аннотация

Аннотация пояснительной записки к курсовому проекту «Система управления задачами».

Исполнитель: Потонов Г.Е.

Руководители: Игнатьев Е. Б., Фомина О.В.

Документ содержит: страниц, 3 рисунков, 1 таблиц,

Ключевые слова: ЗАДАЧА, ИНФОРМАЦИОННАЯ СИСТЕМА, БАЗА ДАННЫХ.

Учебный курсовой проект выполняется по дисциплинам «Проектирование и архитектура ПС», «Конструирование» и посвящен разработке проекта системы управления задачами. Пояснительная записка содержит следующие документы:

1. Задание.
2. Система управления задачами. Техническое задание.
3. Система управления задачами. Технорабочий проект.

# Оглавление

[**Аннотация**](#_quo921jewhxg) **2**

[**Оглавление**](#_due3ixco1wn1) **2**

[**Термины, определения и сокращения**](#_vfbm3j51wzww) **6**

[**Задание**](#_tp0dsscdoqb3) **7**

[**Введение**](#_34kc6jc0qzu) **8**

[**Система управления задачами. Техническое задание**](#_l326ak42287w) **9**

[1 Общие сведения](#_jn0o7hjj7jnr) 9

[1.1 Наименование системы](#_so3y0bjg2dzg) 9

[1.2 Заказчик и Разработчик системы](#_qp48966ymv4v) 9

[1.3 Основание для разработки](#_djh5eotupu0z) 9

[1.4 Плановые сроки начала и окончания работы по разработке проекта](#_q57wpecrh5o0) 9

[1.5 Сведения об источниках и порядке финансирования работ](#_4o3flaob2ros) 9

[1.6 Порядок оформления и предъявления Заказчику результатов работ](#_1bp1a6an16kb) 9

[2 Назначение и цели создания системы](#_gtxy76fyvrmv) 9

[2.1 Назначение системы](#_7z1kj5wvlm2n) 9

[2.2 Цели создания системы](#_u1ie63junqi4) 9

[3 Характеристика объектов автоматизации](#_agvx50zawfws) 10

[3.1 Объекты автоматизации](#_qv3mbabcfsc0) 10

[3.2 Концептуальная модель предметной области](#_muwjlfrf8mo3) 10

[4 Требования к системе](#_6aw1mh6e8q72) 11

[4.1 Требования к системе в целом](#_9hxb40gn2dqg) 11

[4.1.1 Требования к структуре и функционированию системы](#_193ykhkz8sn9) 11

[Архитектура системы](#_pvhf0ezdkh6d) 11

[Информационный обмен между компонентами системы](#_7bx1t06zskmk) 11

[Взаимосвязи со смежными системами](#_g16ns7muy40v) 11

[Перспективы развития, модернизации системы](#_3o6qu44p64vr) 11

[4.1.2 Требования к численности и квалификации персонала системы](#_t6kho284mt0f) 11

[4.1.3 Показатели назначения](#_9pd4iy69v287) 11

[4.1.4 Требования к надежности](#_hup1gm13stsj) 12

[4.1.5 Требования безопасности](#_kz1euze4bjcx) 12

[4.1.6 Требования к эргономике и технической эстетике](#_vu0cksnovfq9) 12

[4.1.7 Требования к эксплуатации, техническому обслуживанию, ремонту и хранению компонентов системы](#_je2dmo29dr54) 12

[4.1.8 Требования к защите информации от несанкционированного доступа](#_mgbidpychaiv) 13

[4.1.9 Требования по сохранности информации при авариях](#_ltxvcqwi9ss9) 13

[4.1.10 Требования к защите от влияния внешних воздействий](#_76s98x25mdvc) 13

[4.1.11 Требования к патентной чистоте](#_i2lqpfn4lgmp) 13

[4.1.12 Требования по стандартизации и унификации](#_r7bj7pubz042) 13

[4.1.13 Дополнительные требования](#_q06gn8doueiw) 13

[4.2 Требования к функциям, выполняемым системой](#_uorcstpk478r) 13

[4.2.2 Спецификация вариантов использования](#_1gnje5mk7xne) 15

[4.3 Требования к видам обеспечения](#_glyp36qb7o6t) 18

[4.3.1 Требования к математическому обеспечению](#_cqql9wf3377i) 18

[4.3.2 Требования к информационному обеспечению](#_1xkpcy91cqli) 18

[4.3.3 Требования к лингвистическому обеспечению](#_hx2dmsi2ykky) 18

[4.3.4 Требования к программному обеспечению](#_i1emjkboj639) 18

[4.3.5 Требования к техническому обеспечению](#_loiixeltqeec) 19

[4.3.6 Требования к метрологическому обеспечению](#_pgktcq83f2m0) 19

[4.3.7 Требования к организационному обеспечению](#_7xtzotu5nkdx) 19

[4.3.8 Требования к методическому обеспечению](#_3wm8s7dhj2rt) 20

[5 Состав и содержание работ по созданию системы](#_eqsfbpml0x6) 20

[6 Порядок контроля и приемки системы](#_wsam66oyjjyi) 20

[7 Требования к составу и содержанию работ по подготовке объекта автоматизации к вводу системы в действие](#_uigkgp52hdu4) 21

[8 Требования к документированию](#_3541ad73uj03) 21

[8.1 Требования к составу документов](#_xwwmh2na5k1h) 21

[8.2 Требования к оформлению документов](#_bzjvb0a0qxw) 21

[8.2.1 Технорабочий проект](#_sj8awggp23ke) 21

[9 Источники разработки](#_ukm45jwgh4ej) 22

[**Система управления автовокзалом. Технорабочий проект**](#_byqoj4g1tx78) **23**

[1 Общие положения](#_25b2l0r) 23

[1.1 Наименование систем](#_34oev4dzll89) 23

[1.2 Основание для разработки](#_7o444exnedd4) 23

[1.3 Перечень организаций, участвующих в разработке системы, сроки выполнения этапов](#_1jlao46) 23

[1.4 Цели, назначение и области использования](#_43ky6rz) 23

[1.5 Подтверждение соответствия проектных решений действующим нормам и правилам техники безопасности, пожаро- и взрывобезопасности](#_9vjvoywp8har) 24

[1.6 Нормативно-технические документы, использованные при проектировании](#_xvir7l) 24

[2 Описание процесса деятельности](#_3hv69ve) 24

[3 Основные технические решения](#_1x0gk37) 24

[3.1 Архитектура системы](#_4h042r0) 24

[Информационный обмен между компонентами системы](#_5vhy1mfdjw6j) 26

[3.2 Решения по взаимосвязям со смежными системами](#_3vac5uf) 26

[3.3 Решения по режимам функционирования, диагностированию работы системы](#_2afmg28) 26

[3.4 Решения по численности, квалификации и функциям персонала, режимам его работы, порядку взаимодействия](#_mwqqa1hograz) 26

[3.5 Сведения об обеспечении заданных в ТЗ потребительских характеристик системы, определяющих ее качество](#_39kk8xu) 26

[3.5.1 Надёжность](#_rf1tq4n1hybp) 26

[3.5.2 Эргономика и техническая эстетика](#_2nusc19) 26

[3.5.3 Защищенность информации](#_1302m92) 26

[3.6 Состав функций, реализуемых системой](#_3mzq4wv) 27

[3.7 Решения по комплексу технических средств](#_haapch) 28

[3.8 Решения по информационному обеспечению](#_319y80a) 28

[3.8.1 Физическая модель БД](#_1gf8i83) 28

[3.8.2 Скрипт создания БД](#_noj3oi4o9m2t) 29

[3.9 Решения по программному обеспечению](#_3ep43zb) 34

[3.9.1 Классы](#_ks0u644klfkq) 34

[3.9.2 Взаимодействия](#_f3to9pcapsoa) 40

[3.9.2.1 Диаграмма последовательности](#_ygraorxn3dgp) 40

[4 Мероприятия по подготовке к вводу системы в действие](#_14ykbeg) 41

[4.1 Мероприятия по приведению информации к виду, пригодному для обработки на ЭВМ](#_3oy7u29) 41

[4.2 Мероприятия по обучению и проверке квалификации персонала](#_243i4a2) 41

[4.3 Мероприятия по созданию необходимых подразделений и рабочих мест](#_j8sehv) 41

[4.4 Мероприятия по изменению объекта автоматизации](#_338fx5o) 41

[5 Источники разработки](#_1idq7dh) 41

# Термины, определения и сокращения

В настоящем документе использованы термины и определения, предусмотренные ГОСТ 34.003-90 [1]. Кроме того, использован ряд терминов и определений, не предусмотренных ГОСТ:

| **Термин** | **Определение** |
| --- | --- |
| Задача | Четко сформулированная цель, которой хочет добиться пользователь |

Ниже приводятся сокращения, использованные в документе:

| **Сокращение** | **Определение** |
| --- | --- |
| ИГЭУ | ФГБОУ ВПО «Ивановский государственный энергетический университет имени В.И. Ленина» |
| БД | База данных |
| ОС | Операционная система |
| ПК | Персональный компьютер |
| ПО | Программное обеспечение |
| ПОКС | Кафедра программного обеспечения компьютерных систем ИГЭУ |
| СУБД | Система управления базами данных |
| ТЗ | Техническое задание |
| MS | Корпорация Microsoft |
| UI | Пользовательский интерфейс (User Interface) |
| UML | Унифицированный язык моделирования (Unified Modeling Language) |

# Задание

Задание по курсовому проекту студента Потонова Георгия Евгеньевича

1. Тема: Система управления задачами.
2. Срок сдачи студентом работы: 25 декабря 2022 г.
3. Исходные данные:

Необходимо разработать проект системы управления задачами.

Для работы с системой авторизация не нужна. Пользователь может добавлять задачи в список задач, изменять задачу (название, описание, вид, статус выполнения, даты и время начала и окончания, интервалы повторения, изменить задачу на повторяющуюся и наоборот, если задача повторяющаяся). Можно вывести отчет по виду, статусу или повторяемости задач.

1. Дата выдачи задания: 23 сентября 2021 г.

Задание выдал:

доц. каф. ПОКС \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Игнатьев Е.Б.

ст. пр. ПОКС \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Фомина О.В.

Задание принял к исполнению:

студент гр. 3-42 \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Потонов Г.Е.

# Введение

Жизнь в современном мире течет гораздо быстрее, чем в даже в прошлом веке, не говоря уже о более ранних временах. Современный человек постоянно занят делами, и порой ему становится сложно удержать в голове все дела, которые ему нужно выполнить. Чтобы решить эту проблему, многие люди ведут ежедневник.

Система, разработанная в ходе данной работы является логическим продолжением развития идеи ежедневника. Только с использованием цифровых возможностей 21-ого века. Перенос ежедневника в компьютер позволит проще управлять задачами, сортировать их, делать выборки из больших списков.

# Система управления задачами. Техническое задание

## 1 Общие сведения

### 1.1 Наименование системы

Полное наименование системы - “Система управления задачами”.

Сокращенное наименование системы - “ToDoApp”.

### 1.2 Заказчик и Разработчик системы

Заказчик системы: Федеральное государственное бюджетное образовательное учреждение высшего образования «Ивановский государственный энергетический университет имени В.И. Ленина» (ИГЭУ); 153003, г. Иваново, ул. Рабфаковская, д. 34.

Разработчик системы: Потонов Георгий Евгеньевич, ст. гр. 4-42.

### 1.3 Основание для разработки

Разработка ведется на основании задания на курсовой проект по дисциплинам: «Проектирование и архитектура ПС», «Конструирование».

Задание утверждено на заседании кафедры ПОКС 01.09.2013 и выдано преподавателем кафедры Игнатьевым Е.Б.

### 1.4 Плановые сроки начала и окончания работы по разработке проекта

Начало: 23 сентября 2021 г.

Окончание: 25 декабря 2022 г.

### 1.5 Сведения об источниках и порядке финансирования работ

Финансирование работ отсутствует.

### 1.6 Порядок оформления и предъявления Заказчику результатов работ

Разработчик оформляет результаты работ над проектом, в виде технорабочего проекта и передает их Заказчику.

## 2 Назначение и цели создания системы

### 2.1 Назначение системы

Система “ToDoApp” предназначена для управления задачами пользователя.

### 2.2 Цели создания системы

Основными целями создания системы являются:

* повышение удобства управления задачами по сравнению с ежедневником
* добавление возможности легко производить выборку задач по параметрам

## 3 Характеристика объектов автоматизации

## 3.1 Объекты автоматизации

Объектами автоматизации являются процессы управления задачами, выборки задач по параметрам.

## 3.2 Концептуальная модель предметной области

В результате обследования предметной области была разработана модель объектов предметной области (Domain Object Model – DOM), описывающая классы предметной области и связи между ними.

Рис. 3.1 представляет диаграмму классов для концептуальных классов.

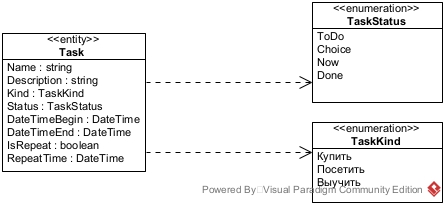


Рис. 3.1 Концептуальная модель предметной области

## 4 Требования к системе

### 4.1 Требования к системе в целом

#### 4.1.1 Требования к структуре и функционированию системы

##### Архитектура системы

Система “ToDoApp” должна иметь трехуровневую архитектуру:

* Прикладной уровень
* Уровень бизнес-логики
* Уровень доступа к данным в базе данных
* База данных

##### Информационный обмен между компонентами системы

Прикладной уровень реализуется в виде приложения WinForms. А уровень бизнес-логики и уровень доступа к данным в базе данных реализуются в виде библиотек.

Доступ к базе данных осуществляется посредством инструмента Entity Framework.

##### Взаимосвязи со смежными системами

Обеспечение взаимосвязи со смежными системами не требуется.

##### Перспективы развития, модернизации системы

* Можно добавить возможность редактирования видов и статусов задач, а также добавления кастомных видов и статусов задач.
* Было бы плюсом реализация приложения в виде проекта WPF, использующего шаблон проектирования MVVM.
* Также можно реализовать проект с использованием языка Java (библиотека GUI-компонентов Swing) или использовать QT — фреймворк для разработки кроссплатформенного программного обеспечения на языке программирования C++.
* Можно добавить реализацию защиты от несанкционированного доступа.
* Можно реализовать канбан-доску, где с областями, соответствующими статусу задач.
* Можно реализовать возможность вывода уведомлений.
* Можно объединить отчеты по виду, статусу, повторяемости в один. Таким образом получится один отчет с тремя фильтрами.

#### 4.1.2 Требования к численности и квалификации персонала системы

Пользователями системы являются обычные люди.

Пользователи системы должны иметь опыт работы с персональным компьютером на базе операционных систем Microsoft Windows, macOS или Linux на уровне обычного пользователя.

#### 4.1.3 Показатели назначения

Масштабирование по производительности и объему обрабатываемой информации не требуется.

#### 4.1.4 Требования к надежности

Система должна быть доступна 24 часа каждый день недели. Не должно быть никакого связанного с БД времени простоя.

Отказ компонентов программного обеспечения не должен ставить под угрозу корректность и целостность БД. Пользователь должен иметь возможность повторно начать программу после отказа и найти информацию БД непротиворечивой и не повреждённой в результате отказа.

Система должна сохранять работоспособность и обеспечивать восстановление своих функций при возникновении следующих внештатных ситуаций:

– при сбоях в системе электроснабжения аппаратной части, приводящих к перезагрузке ОС, восстановление программы должно происходить после перезагрузки ОС и запуска системы;

– при ошибках в работе аппаратных средств (кроме носителей данных и программ) восстановление функций системы возлагается на ОС;

– при ошибках, связанных с программным обеспечением (ОС и драйверы устройств), восстановление работоспособности возлагается на ОС.

Система должна обеспечивать корректную обработку ситуаций, вызванных недопустимыми и несогласованными значениями входных данных. В указанных случаях пользователю должны выдаваться соответствующие уведомления, после чего система должна возвращаться в рабочее состояние.

Для обеспечения устойчивости к отказам электроснабжения все устройства хранения и обработки информации должны быть подключены к электросети через источники бесперебойного питания.

#### 4.1.5 Требования безопасности

Все технические решения, использованные при создании системы, а также при определении требований к аппаратному обеспечению, должны соответствовать действующим нормам и правилам техники безопасности, пожаробезопасности и взрывобезопасности, а также охраны окружающей среды при эксплуатации.

#### 4.1.6 Требования к эргономике и технической эстетике

Взаимодействие пользователей с прикладным программным обеспечением, входящим в состав системы должно осуществляться посредством визуального графического интерфейса (GUI). Интерфейс системы должен быть понятным и удобным, не должен быть перегружен графическими элементами и должен обеспечивать быстрое отображение экранных форм. Навигационные элементы должны быть выполнены в удобной для пользователя форме. Средства редактирования информации должны удовлетворять принятым соглашениям в части использования функциональных клавиш, режимов работы, поиска, использования оконной системы.

#### 4.1.7 Требования к эксплуатации, техническому обслуживанию, ремонту и хранению компонентов системы

Система должна быть рассчитана на эксплуатацию в составе программно-технического комплекса Заказчика и учитывать разделение ИТ инфраструктуры Заказчика на внутреннюю и внешнюю. Техническая и физическая защита аппаратных компонентов системы, носителей данных, бесперебойное энергоснабжение, резервирование ресурсов, текущее обслуживание реализуется техническими и организационными средствами, предусмотренными в ИТ инфраструктуре Заказчика.

Для нормальной эксплуатации разрабатываемой системы должно быть обеспечено бесперебойное питание персональных компьютеров (ПК). При эксплуатации системы должна быть обеспечена соответствующая стандартам хранения носителей и эксплуатации ПК температура и влажность воздуха.

#### 4.1.8 Требования к защите информации от несанкционированного доступа

ИС не должна обеспечивать защиту от несанкционированного доступа (НСД).

#### 4.1.9 Требования по сохранности информации при авариях

Программное обеспечение системы должно восстанавливать свое функционирование при корректном перезапуске аппаратных средств. Должна быть предусмотрена возможность организации автоматического и (или) ручного резервного копирования данных системы средствами системного и базового программного обеспечения (ОС, СУБД), входящего в состав программно-технического комплекса Заказчика.

Приведенные выше требования не распространяются на компоненты системы, разработанные третьими сторонами и действительны только при соблюдении правил эксплуатации этих компонентов, включая своевременную установку обновлений, рекомендованных производителями покупного программного обеспечения.

#### 4.1.10 Требования к защите от влияния внешних воздействий

Требования к защите от влияния внешних воздействий не предъявляются.

#### 4.1.11 Требования к патентной чистоте

Установка системы в целом, как и установка отдельных частей системы не должна предъявлять дополнительных требований к покупке лицензий на программное обеспечение сторонних производителей, кроме программного обеспечения, указанного в разделе 4.3.4.

#### 4.1.12 Требования по стандартизации и унификации

Требования по стандартизации и унификации не предъявляются.

#### 4.1.13 Дополнительные требования

Специальные требования не предъявляются.

### 4.2 Требования к функциям, выполняемым системой

Функции, подлежащие автоматизации перечислены в модели вариантов использования (рисунок 4.2.1).

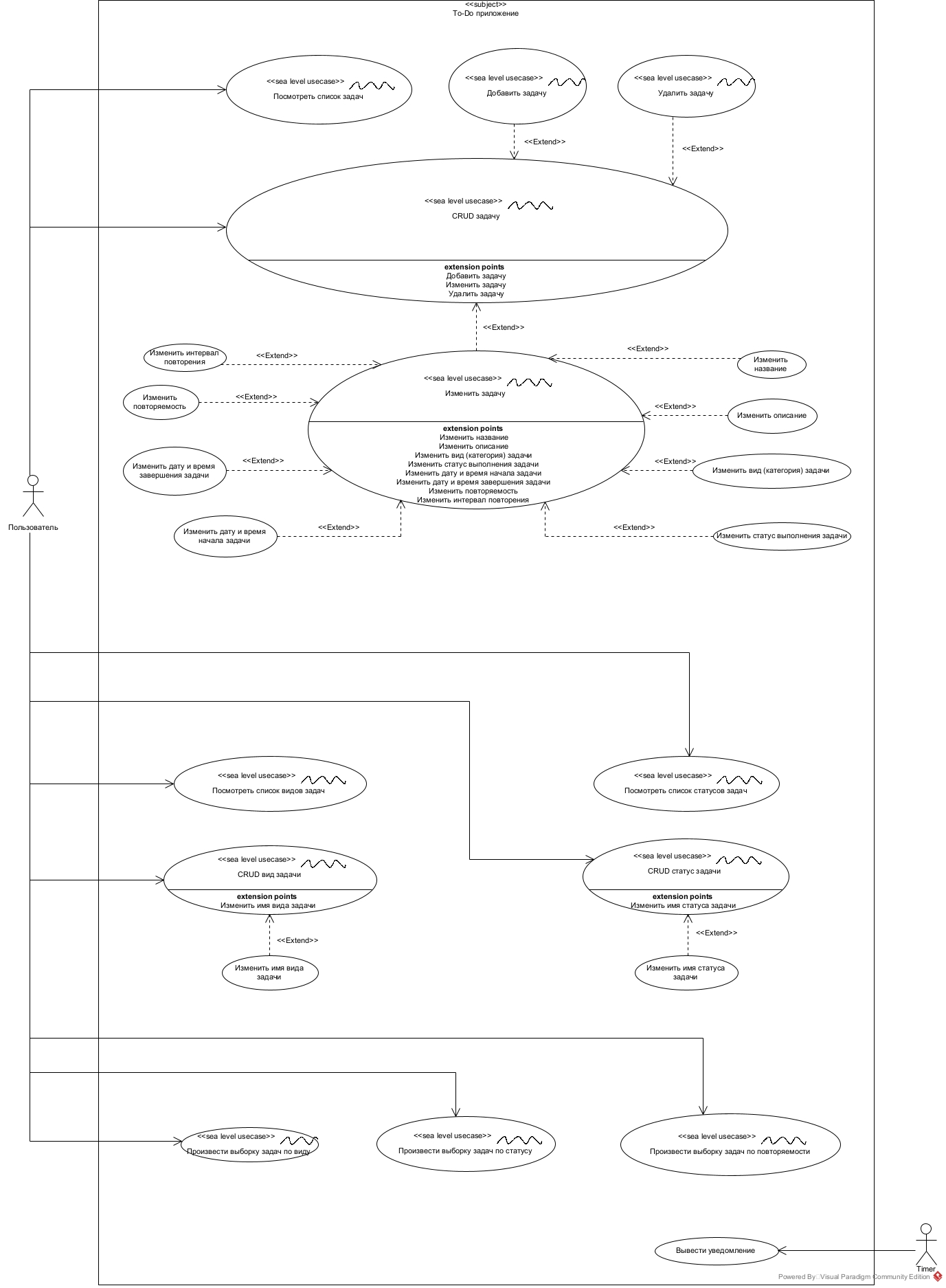


Рис. 4.2.1 Диаграмма вариантов использования

#### 4.2.2 Спецификация вариантов использования

* Вариант использования Добавить задачу

Данный вариант использования описывает добавление задачи в базу данных пользователем

Основной поток:

1. Пользователь нажимает кнопку «Добавить задачу».
2. Система открывает окно добавления задачи.
3. Пользователь вводит данные о задаче.
4. Пользователь нажимает кнопку «ОК».
5. Система добавляет запись в базу данных.

Альтернативный поток:

3а. Пользователь нажимает кнопку «Отмена».

3а1. Система закрывает окно добавления задачи.

4а. Пользователь нажимает кнопку «Отмена».

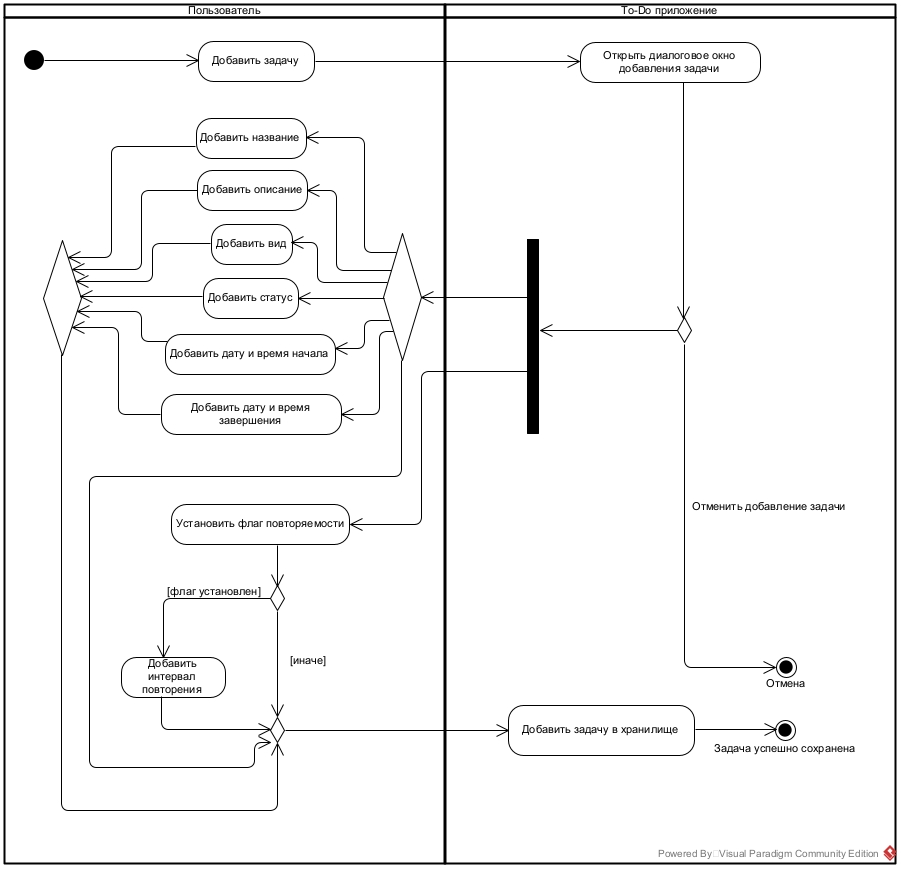
4а1. Система закрывает окно добавления задачи.

Предусловия:

Пользователь находится на вкладке «Задачи».

Постусловия:

Пользователь возвращается на вкладку «Задачи».



4.2.2.1 Диаграмма деятельности

* Вариант использования Изменить задачу

Данный вариант использования описывает изменение задачи в базу данных пользователем

Основной поток:

1. Пользователь нажимает кнопку «Изменить задачу»
2. Система открывает окно изменения задачи.
3. Пользователь вводу новые данные о задаче.
4. Пользователь нажимает кнопку «ОК».
5. Система изменяет запись в базе данных.

Альтернативный поток:

3а. Пользователь нажимает кнопку «Отмена».

3а1. Система закрывает окно изменения задачи.

4а. Пользователь нажимает кнопку «Отмена».

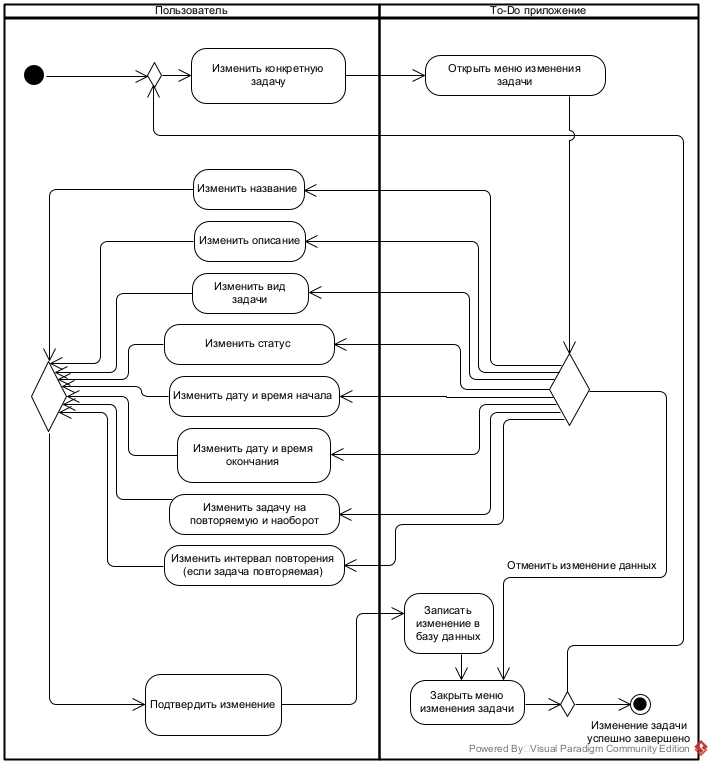
4а1. Система закрывает окно изменения задачи.

Предусловия:

Пользователь находится на вкладке «Задачи».

Постусловия:

Пользователь возвращается на вкладку «Задачи».



4.2.2.2 Диаграмма деятельности

* Вариант использования Посмотреть список задач

Данный вариант использования описывает просмотр списка задач.

Основной поток:

1. Пользователь просматривает список задач во вкладке «Задачи»

Предусловия:

Пользователь находится на вкладке «Задачи».

Постусловия:

Пользователь возвращается на вкладку «Задачи».

* Вариант использования Удалить задачу

Данный вариант использования описывает удаление задачи.

Основной поток:

1. Пользователь выделяет в таблице со списком задач на вкладке «Задачи» задачу.
2. Система удаляет задачу из базы данных.
3. Система обновляет таблицу со списком задач

Предусловия:

Пользователь находится на вкладке «Задачи».

Постусловия:

Пользователь возвращается на вкладку «Задачи».

* Вариант использования Произвести выборку задач по виду

Данный вариант использования описывает выборку задач по виду.

Основной поток:

1. Пользователь выбирает в чекбоксе вид задач.
2. Пользователь нажимает кнопку «Искать».
3. Система выводит список задач, вид которых соответствует выбранному пользователем.

Предусловия:

Пользователь находится на вкладке «Выборка задач по виду».

Постусловия:

Пользователь возвращается на вкладку «Выборка задач по виду».

* Вариант использования Произвести выборку задач по статусу

Данный вариант использования описывает выборку задач по статусу.

Основной поток:

1. Пользователь выбирает в чекбоксе статус задач.
2. Пользователь нажимает кнопку «Искать».
3. Система выводит список задач, статус которых соответствует выбранному пользователем.

Предусловия:

Пользователь находится на вкладке «Выборка задач по статусу».

Постусловия:

Пользователь возвращается на вкладку «Выборка задач по статусу».

* Вариант использования Произвести выборку задач по повторяемости

Данный вариант использования описывает выборку задач по повторяемости.

Основной поток:

1. Пользователь отмечает флажком чекбокс.
2. Пользователь нажимает кнопку «Искать»
3. Система выводит список повторяемых задач.

Альтернативный поток:

1а. Пользователь отмечает флажком чекбокс.

1а1. Пользователь нажимает кнопку «Искать».

1а2. Система выводит список не повторяемых задач.

Предусловия:

Пользователь находится на вкладке «Выборка задач по повторяемости».

Постусловия:

Пользователь возвращается на вкладку «Выборка задач по повторяемости».

### 4.3 Требования к видам обеспечения

#### 4.3.1 Требования к математическому обеспечению

Требования к математическому обеспечению не предъявляются.

#### 4.3.2 Требования к информационному обеспечению

Уровень хранения данных в системе должен быть построен на платформе реляционной СУБД. Для обеспечения целостности данных должны использоваться встроенные механизмы СУБД.

База данных предназначена для хранения:

* сведений о задачах,
* сведений о видах задач,
* сведений о статусах задач.

#### 4.3.3 Требования к лингвистическому обеспечению

Программное обеспечение системы должно быть разработано на языке программирования C#.

Все прикладное программное обеспечение системы для организации взаимодействия с пользователем должно использовать русский и английский языки.

#### 4.3.4 Требования к программному обеспечению

Проект должен использовать СУБД Microsoft SQL Server, но он должен быть легко перестраиваемым для других реляционных БД.

1-я версия должна использовать C# и Entity Framework для доступа из программы к БД.

Для создания кода должна использоваться управляемая тестированием разработка. Для проверки кода - приемочные испытания. Тестируемые единицы, полученные в результате управляемой тестированием разработки и приемочных испытаний, используются для регрессионного тестирования, когда код будет заменен на 2-ю версию.

Разрабатываемая система должна быть рассчитана на функционирование в следующей программной среде:

**Рабочая станция**

Типовое программное обеспечение, устанавливаемое на рабочую станцию:

1. Базовая операционная система: Windows 7 Home, Windows 10 Home.
2. .NET Framework 4.8.1 (входит в состав базовой операционной системы)

**Сервер**

ПО, устанавливаемое на компьютер сервера:

1. Базовая ОС – Microsoft Windows 10.
2. Средство для web-публикации локальных информационных ресурсов – Internet Information Server (Входит в состав базовой операционной системы).
3. Система управления базами данных – SQL Server Management Studio.
4. Firewall для защиты внутренних ресурсов системы, при наличии подключения к транзитным провайдерам услуг передачи данных – Microsoft ISA Server.

#### 4.3.5 Требования к техническому обеспечению

Техническое обеспечение системы должно максимально и наиболее эффективным образом использовать существующие технические средства.

В состав комплекса должен входить персональный компьютер (ПК) пользователя.

Минимальные требования к характеристикам компонентов технического обеспечения, при которых значения временных параметров системы должны соответствовать предъявленным в ТЗ требованиям (п. 4.2.3):

1. для ПК пользователя:

* процессор – Intel Pentium 1.5 ГГц;
* объем оперативной памяти – 1024 Мб;
* дисковая память – 1 Гб;
* сетевой адаптер – 100 Мбит/с.

#### 4.3.6 Требования к метрологическому обеспечению

Требования к метрологическому обеспечению не предъявляются.

#### 4.3.7 Требования к организационному обеспечению

Организационное обеспечение системы должно быть достаточным для эффективного выполнения персоналом возложенных на него обязанностей при осуществлении автоматизированных и связанных с ними неавтоматизированных функций системы.

Заказчиком должны быть подготовлены изменения в положения о структурных подразделениях, в которых будет эксплуатироваться система “ToDoApp”.

Заказчиком должны быть подготовлены изменения к действующим должностным инструкциям, для персонала, который будет участвовать в эксплуатации Системы.

Заказчиком должен быть подготовлен приказ о приёмке системы “ToDoApp” и вводе её в эксплуатацию с указанием ответственных за эксплуатацию системы.

#### 4.3.8 Требования к методическому обеспечению

Требования к методическому обеспечению не предъявляются.

## 5 Состав и содержание работ по созданию системы

Разработка и сдача проекта должна вестись по этапам (Таблица 5.1).

Таблица 5.1. Этапы работ над проектом

| **Наименование этапа** | **Результаты этапа** | **Дата начала этапа** | **Дата завершения этапа** |
| --- | --- | --- | --- |
| 1. Разработка, согласование и утверждение технического задания | Техническое задание | 23.09.2021 | 5.10.2021 |
| 2. Разработка, согласование и утверждение технорабочего проекта | Технорабочий проект | 28.10.2021 | 13.12.2021 |
| 3. Подготовка презентации и защита проекта | Пояснительная записка и презентация | 16.12.2021 | 25.12.2022 |

## 6 Порядок контроля и приемки системы

По окончании работ проект принимается Приёмной комиссией.

Заседание комиссии проводится в конце 5-го семестра перед зачётной неделей.

Проект подлежит защите. Защита проекта проводится Разработчиком перед членами Приёмной комиссии.

Приёмная комиссия назначается из числа преподавателей кафедры ПОКС.

Комиссии предоставляется полностью оформленная и подписанная пояснительная записка и презентация.

Проект принимается, если он удовлетворяет всем пунктам данного технического задания. По результатам защиты проекта Приёмная комиссия выставляет оценку.

## 7 Требования к составу и содержанию работ по подготовке объекта автоматизации к вводу системы в действие

Для ввода Системы в действие не требуются подготовительные работы.

## 8 Требования к документированию

### 8.1 Требования к составу документов

По окончании работ над проектом все разработанные документы объединяются в Пояснительную записку. Она должна содержать:

1. титульный лист,
2. аннотацию,
3. оглавление,
4. термины, определения и сокращения, использованные в Пояснительной записке,
5. Задание,
6. введение,
7. Техническое задание,
8. Технорабочий проект;
9. список литературы.

Заказчику предоставляются:

1. Пояснительная записка в формате MS Word 2010;
2. Пояснительная записка, распечатанная на бумаге формата А4 – 1 экземпляр;
3. Презентация (в формате MS PowerPoint), демонстрирующая основные проектные решения.

### 8.2 Требования к оформлению документов

Техническое задание и Технорабочий проект оформляются в соответствии с ГОСТ 34.201-89 [2], ГОСТ 34.602-89 [4] и РД 50-34.698.90 [5].

#### 8.2.1 Технорабочий проект

Технорабочий проект должен содержать следующие разделы:

1. общие положения;
2. описание процесса деятельности;
3. основные технические решения;
4. мероприятия по подготовке объекта автоматизации к вводу системы в действие;
5. схема функциональной структуры;
6. описание автоматизируемых функций;
7. описание комплекса технических средств;
8. описание программного обеспечения;
9. описание информационного обеспечения;
10. описание организационной структуры;
11. руководство пользователя.

## 9 Источники разработки

При разработке ТЗ использовались следующие источники:

1. ГОСТ 34.003-90. Информационные технологии. Комплекс стандартов на автоматизированные системы. Автоматизированные системы. Термины и определения.
2. ГОСТ 34.201-89. Информационная технология. Комплекс стандартов на автоматизированные системы. Виды, комплектность и обозначение документов при создании автоматизированных систем.
3. ГОСТ 34.601-90 Информационная технология. Комплекс стандартов на автоматизированные системы. Автоматизированные системы. Стадии создания.
4. ГОСТ 34.602-89. Информационная технология. Комплекс стандартов на автоматизированные системы. Техническое задание на создание автоматизированной системы.
5. РД 50-34.698.90. Методические указания. Информационная технология. Комплекс стандартов на автоматизированные системы. Автоматизированные системы. Требования к содержанию документов.
6. РД Гостехкомиссии России. Безопасность информационных технологий. Критерии оценки безопасности информационных технологий.- 2002 г..
7. Практическая программная инженерия на основе учебного примера / Л.А. Мацяшек, Б.Л. Лионг. – М.: БИНОМ. Лаборатория знаний, 2009. – 956 с.

# 

# 

# Система управления автовокзалом. Технорабочий проект

## 1 Общие положения

### 1.1 Наименование систем

Полное наименование системы - “Система управления задачами”

Сокращенное наименование системы - “ToDoApp”.

### 1.2 Основание для разработки

Разработка ведется на основании задания на курсовой проект по дисциплинам: «Проектирование и архитектура ПС», «Проектирование человеко-машинного интерфейса», «Конструирование».

Задание утверждено на заседании кафедры ПОКС 01.09.2013 и выдано преподавателем кафедры Игнатьевым Е.Б.

### 1.3 Перечень организаций, участвующих в разработке системы, сроки выполнения этапов

Заказчик системы: Федеральное государственное бюджетное образовательное учреждение высшего образования «Ивановский государственный энергетический университет имени В.И. Ленина» (ИГЭУ); 153003, г. Иваново, ул. Рабфаковская, д. 34.

Разработчик системы: Потонов Георгий Евгеньевич, ст. гр. 4-42.

Сроки выполнения этапов работы над проектом показаны в таблице 1.1.

Таблица 1.1. Этапы работ над проектом

| **Наименование этапа** | **Результаты этапа** | **Дата начала этапа** | **Дата завершения этапа** |
| --- | --- | --- | --- |
| 1. Разработка, согласование и утверждение технорабочего проекта | Технорабочий проект | 28.10.2021 | 13.12.2021 |
| 2. Подготовка презентации и защита проекта | Пояснительная записка и презентация | 12.12.2021 | 25.12.2022 |

### 1.4 Цели, назначение и области использования

Основными целями создания системы являются:

* повышение удобства управления задачами по сравнению с ежедневником
* добавление возможности легко производить выборку задач по параметрам

### 1.5 Подтверждение соответствия проектных решений действующим нормам и правилам техники безопасности, пожаро- и взрывобезопасности

Все технические решения, использованные при разработке проекта, соответствуют действующим нормам и правилам техники безопасности, пожаробезопасности и взрывобезопасности, а также охраны окружающей среды при эксплуатации.

### 1.6 Нормативно-технические документы, использованные при проектировании

При проектировании использовалась спецификация текущей версии UML - Documents associated with. Unified Modeling Language (UML), v2.4.1. Release date: August 2011.- http://www.omg.org/spec/UML/2.4.1/ [8].

### 2 Описание процесса деятельности

Система ToDoApp автоматизирует процесс управления задачами и процесс выборки задач по параметрам. Пользователь системы управляет клиентским приложением.

## 3 Основные технические решения

### 3.1 Архитектура системы

Система “ToDoApp” должна иметь трехуровневую архитектуру:

* Прикладной уровень
* Уровень бизнес-логики
* Уровень доступа к данным в базе данных

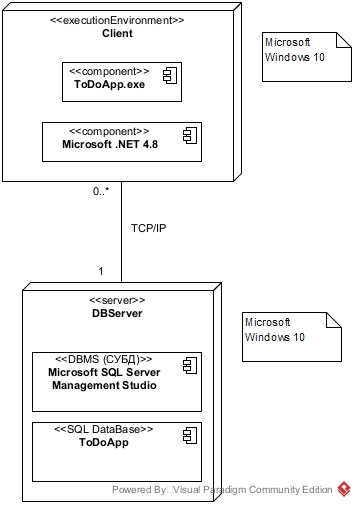


Рис. 3.1.1. Диаграмма развёртывания

## 

Рис. 3.1.2. Структурный шаблон MVVM

Модель (Model) предоставляет данные и реагирует на команды контроллера, изменяя своё состояние.

Представление (View) отвечает за отображение данных модели пользователю, реагируя на изменения модели.

Контроллер (Controller) интерпретирует действия пользователя, оповещая модель о необходимости изменений.

#### Информационный обмен между компонентами системы

Форматы данных будут разработаны и утверждены на этапе технического проектирования.

### 3.2 Решения по взаимосвязям со смежными системами

Система не взаимодействует с другими системами.

### 3.3 Решения по режимам функционирования, диагностированию работы системы

В работе системы можно выделить следующие режимы работы:

1. Работа пользователя

Специальных процедур диагностирования состояния компонентов системы не предусмотрено.

### 3.4 Решения по численности, квалификации и функциям персонала, режимам его работы, порядку взаимодействия

ToDoApp — однопользовательское приложение. Не требуется никакого обучения для пользователя, знакомого с компьютером

### 3.5 Сведения об обеспечении заданных в ТЗ потребительских характеристик системы, определяющих ее качество

#### 3.5.1 Надёжность

Для обеспечения устойчивости к отказам электроснабжения все устройства хранения и обработки информации подключаются к электросети через источники бесперебойного питания соответствующей мощности.

#### 3.5.2 Эргономика и техническая эстетика

Требования эргономичности и технической эстетики будут обеспечены использованием набора компонентов Windows Forms.

#### 3.5.3 Защищенность информации

Защита от несанкционированного доступа обеспечивается через средства парольной аутентификации ОС и БД.

Для восстановления работоспособности системы после сбоев предусмотрено резервное копирование БД средствами СУБД MS SQL Server Management Studio.

### 3.6 Состав функций, реализуемых системой

На диаграмме вариантов использования (Рис. 3.6.1) приведены функции системы.

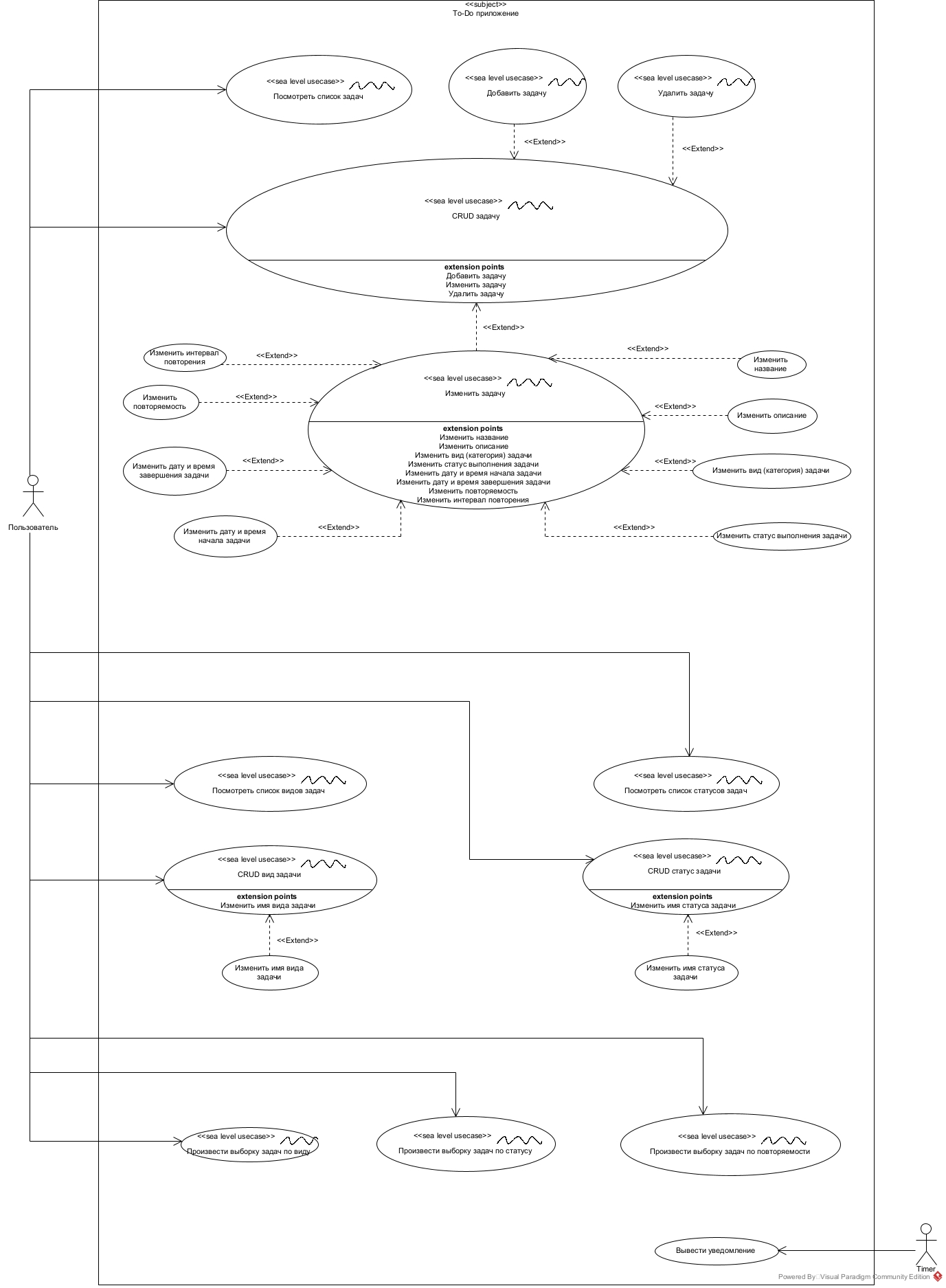


Рис. 3.6.1. Диаграмма вариантов использования

### 3.7 Решения по комплексу технических средств

В соответствии с ТЗ:

1. в качестве сервера БД будет использован компьютер, имеющийся у AEM-организации, и установленный в специальном серверном помещении;
2. в качестве ПК пользователя будет использован компьютер установленный в настоящее время на рабочем месте пользователя;
3. компьютеры объединены локальной вычислительной сетью с пропускной способностью 100 Мбит/с.

**Характеристики сервера**

– процессор – 2 х Intel Xeon 3 ГГц;

– объем оперативной памяти – 16 Гб;

– дисковая подсистема – 4 х 146 Гб;

– устройство чтения компакт-дисков (DVD-ROM);

– сетевой адаптер – 100 Мбит/с.

**Минимальные характеристики для ПК пользователя**

– процессор – Intel Pentium 1.5 ГГц;

– объем оперативной памяти – 256 Мб;

– дисковая память – 40 Гб;

– сетевой адаптер – 100 Мбит/с.

### 3.8 Решения по информационному обеспечению

Компонентом информационного обеспечения является системная БД «ToDoApp».

БД существует заранее и загружена данными относительно задач, видов задач, статусов задач. Прикладная программа ToDoApp извлекает эту информацию из БД, выводит списки, готовит отчеты.

#### 3.8.1 Физическая модель БД

На основе Концептуальной модели предметной области, приведённой в ТЗ (п. 3.2) была получена логическая модель реляционной БД (Рис. 3.4).

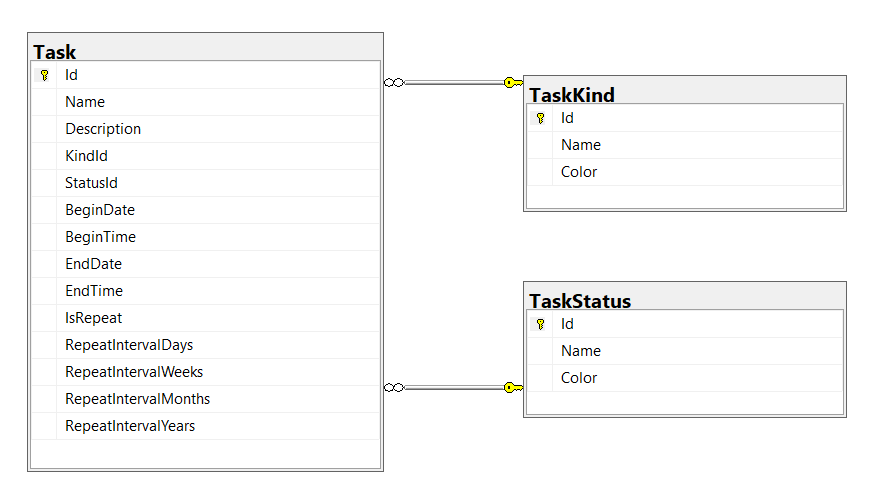


Рис. 3.8.1.1. Физическая модель БД

Каждому концептуальному классу поставлена в соответствие таблица:

1. таблица Task – предназначена для хранения данных о задачах;
2. таблица TaskKind – справочник видов задач;
3. таблица TaskStatus – справочник статусов задач.

#### 3.8.2 Скрипт создания БД

USE [master]

GO

/\*\*\*\*\*\* Object: Database [ToDoApp] Script Date: Вс, 11.12.2022 20:08:16 \*\*\*\*\*\*/

CREATE DATABASE [ToDoApp]

CONTAINMENT = NONE

ON PRIMARY

( NAME = N'to-do app', FILENAME = N'C:\Program Files\Microsoft SQL Server\MSSQL15.SQLEXPRESS\MSSQL\DATA\to-do app.mdf' , SIZE = 8192KB , MAXSIZE = UNLIMITED, FILEGROWTH = 65536KB )

LOG ON

( NAME = N'to-do app\_log', FILENAME = N'C:\Program Files\Microsoft SQL Server\MSSQL15.SQLEXPRESS\MSSQL\DATA\to-do app\_log.ldf' , SIZE = 8192KB , MAXSIZE = 2048GB , FILEGROWTH = 65536KB )

WITH CATALOG\_COLLATION = DATABASE\_DEFAULT

GO

ALTER DATABASE [ToDoApp] SET COMPATIBILITY\_LEVEL = 150

GO

IF (1 = FULLTEXTSERVICEPROPERTY('IsFullTextInstalled'))

begin

EXEC [ToDoApp].[dbo].[sp\_fulltext\_database] @action = 'enable'

end

GO

ALTER DATABASE [ToDoApp] SET ANSI\_NULL\_DEFAULT OFF

GO

ALTER DATABASE [ToDoApp] SET ANSI\_NULLS OFF

GO

ALTER DATABASE [ToDoApp] SET ANSI\_PADDING OFF

GO

ALTER DATABASE [ToDoApp] SET ANSI\_WARNINGS OFF

GO

ALTER DATABASE [ToDoApp] SET ARITHABORT OFF

GO

ALTER DATABASE [ToDoApp] SET AUTO\_CLOSE OFF

GO

ALTER DATABASE [ToDoApp] SET AUTO\_SHRINK OFF

GO

ALTER DATABASE [ToDoApp] SET AUTO\_UPDATE\_STATISTICS ON

GO

ALTER DATABASE [ToDoApp] SET CURSOR\_CLOSE\_ON\_COMMIT OFF

GO

ALTER DATABASE [ToDoApp] SET CURSOR\_DEFAULT GLOBAL

GO

ALTER DATABASE [ToDoApp] SET CONCAT\_NULL\_YIELDS\_NULL OFF

GO

ALTER DATABASE [ToDoApp] SET NUMERIC\_ROUNDABORT OFF

GO

ALTER DATABASE [ToDoApp] SET QUOTED\_IDENTIFIER OFF

GO

ALTER DATABASE [ToDoApp] SET RECURSIVE\_TRIGGERS OFF

GO

ALTER DATABASE [ToDoApp] SET DISABLE\_BROKER

GO

ALTER DATABASE [ToDoApp] SET AUTO\_UPDATE\_STATISTICS\_ASYNC OFF

GO

ALTER DATABASE [ToDoApp] SET DATE\_CORRELATION\_OPTIMIZATION OFF

GO

ALTER DATABASE [ToDoApp] SET TRUSTWORTHY OFF

GO

ALTER DATABASE [ToDoApp] SET ALLOW\_SNAPSHOT\_ISOLATION OFF

GO

ALTER DATABASE [ToDoApp] SET PARAMETERIZATION SIMPLE

GO

ALTER DATABASE [ToDoApp] SET READ\_COMMITTED\_SNAPSHOT OFF

GO

ALTER DATABASE [ToDoApp] SET HONOR\_BROKER\_PRIORITY OFF

GO

ALTER DATABASE [ToDoApp] SET RECOVERY SIMPLE

GO

ALTER DATABASE [ToDoApp] SET MULTI\_USER

GO

ALTER DATABASE [ToDoApp] SET PAGE\_VERIFY CHECKSUM

GO

ALTER DATABASE [ToDoApp] SET DB\_CHAINING OFF

GO

ALTER DATABASE [ToDoApp] SET FILESTREAM( NON\_TRANSACTED\_ACCESS = OFF )

GO

ALTER DATABASE [ToDoApp] SET TARGET\_RECOVERY\_TIME = 60 SECONDS

GO

ALTER DATABASE [ToDoApp] SET DELAYED\_DURABILITY = DISABLED

GO

ALTER DATABASE [ToDoApp] SET ACCELERATED\_DATABASE\_RECOVERY = OFF

GO

ALTER DATABASE [ToDoApp] SET QUERY\_STORE = OFF

GO

USE [ToDoApp]

GO

/\*\*\*\*\*\* Object: Table [dbo].[Task] Script Date: Вс, 11.12.2022 20:08:16 \*\*\*\*\*\*/

SET ANSI\_NULLS ON

GO

SET QUOTED\_IDENTIFIER ON

GO

CREATE TABLE [dbo].[Task](

[Id] [int] IDENTITY(1,1) NOT NULL,

[Name] [nvarchar](max) NULL,

[Description] [nvarchar](max) NULL,

[KindId] [int] NULL,

[StatusId] [int] NULL,

[BeginDate] [date] NULL,

[BeginTime] [time](7) NULL,

[EndDate] [date] NULL,

[EndTime] [time](7) NULL,

[IsRepeat] [bit] NOT NULL,

[RepeatIntervalDays] [int] NULL,

[RepeatIntervalWeeks] [int] NULL,

[RepeatIntervalMonths] [int] NULL,

[RepeatIntervalYears] [int] NULL,

CONSTRAINT [PK\_Task] PRIMARY KEY CLUSTERED

(

[Id] ASC

)WITH (PAD\_INDEX = OFF, STATISTICS\_NORECOMPUTE = OFF, IGNORE\_DUP\_KEY = OFF, ALLOW\_ROW\_LOCKS = ON, ALLOW\_PAGE\_LOCKS = ON, OPTIMIZE\_FOR\_SEQUENTIAL\_KEY = OFF) ON [PRIMARY]

) ON [PRIMARY] TEXTIMAGE\_ON [PRIMARY]

GO

/\*\*\*\*\*\* Object: Table [dbo].[TaskKind] Script Date: Вс, 11.12.2022 20:08:16 \*\*\*\*\*\*/

SET ANSI\_NULLS ON

GO

SET QUOTED\_IDENTIFIER ON

GO

CREATE TABLE [dbo].[TaskKind](

[Id] [int] IDENTITY(1,1) NOT NULL,

[Name] [nvarchar](max) NULL,

[Color] [int] NULL,

CONSTRAINT [PK\_TaskType] PRIMARY KEY CLUSTERED

(

[Id] ASC

)WITH (PAD\_INDEX = OFF, STATISTICS\_NORECOMPUTE = OFF, IGNORE\_DUP\_KEY = OFF, ALLOW\_ROW\_LOCKS = ON, ALLOW\_PAGE\_LOCKS = ON, OPTIMIZE\_FOR\_SEQUENTIAL\_KEY = OFF) ON [PRIMARY]

) ON [PRIMARY] TEXTIMAGE\_ON [PRIMARY]

GO

/\*\*\*\*\*\* Object: Table [dbo].[TaskStatus] Script Date: Вс, 11.12.2022 20:08:16 \*\*\*\*\*\*/

SET ANSI\_NULLS ON

GO

SET QUOTED\_IDENTIFIER ON

GO

CREATE TABLE [dbo].[TaskStatus](

[Id] [int] IDENTITY(1,1) NOT NULL,

[Name] [nvarchar](max) NULL,

[Color] [int] NULL,

CONSTRAINT [PK\_BoardStatus] PRIMARY KEY CLUSTERED

(

[Id] ASC

)WITH (PAD\_INDEX = OFF, STATISTICS\_NORECOMPUTE = OFF, IGNORE\_DUP\_KEY = OFF, ALLOW\_ROW\_LOCKS = ON, ALLOW\_PAGE\_LOCKS = ON, OPTIMIZE\_FOR\_SEQUENTIAL\_KEY = OFF) ON [PRIMARY]

) ON [PRIMARY] TEXTIMAGE\_ON [PRIMARY]

GO

ALTER TABLE [dbo].[Task] ADD CONSTRAINT [DF\_Task\_TypeId] DEFAULT ((1)) FOR [KindId]

GO

ALTER TABLE [dbo].[Task] ADD CONSTRAINT [DF\_Task\_IsRepeat] DEFAULT ((0)) FOR [IsRepeat]

GO

ALTER TABLE [dbo].[Task] ADD CONSTRAINT [DF\_Task\_RepeatIntervalDays] DEFAULT ((0)) FOR [RepeatIntervalDays]

GO

ALTER TABLE [dbo].[Task] ADD CONSTRAINT [DF\_Task\_RepeatIntervalWeeks] DEFAULT ((0)) FOR [RepeatIntervalWeeks]

GO

ALTER TABLE [dbo].[Task] ADD CONSTRAINT [DF\_Task\_RepeatIntervalMonths] DEFAULT ((0)) FOR [RepeatIntervalMonths]

GO

ALTER TABLE [dbo].[Task] ADD CONSTRAINT [DF\_Task\_RepeatIntervalYears] DEFAULT ((0)) FOR [RepeatIntervalYears]

GO

ALTER TABLE [dbo].[Task] WITH CHECK ADD CONSTRAINT [FK\_Task\_TaskStatus] FOREIGN KEY([StatusId])

REFERENCES [dbo].[TaskStatus] ([Id])

GO

ALTER TABLE [dbo].[Task] CHECK CONSTRAINT [FK\_Task\_TaskStatus]

GO

ALTER TABLE [dbo].[Task] WITH CHECK ADD CONSTRAINT [FK\_Task\_TaskType] FOREIGN KEY([KindId])

REFERENCES [dbo].[TaskKind] ([Id])

GO

ALTER TABLE [dbo].[Task] CHECK CONSTRAINT [FK\_Task\_TaskType]

GO

/\*\*\*\*\*\* Object: StoredProcedure [dbo].[IsRepeatReportProcedure] Script Date: Вс, 11.12.2022 20:08:16 \*\*\*\*\*\*/

SET ANSI\_NULLS ON

GO

SET QUOTED\_IDENTIFIER ON

GO

-- =============================================

-- Author: <Author,,Name>

-- Create date: <Create Date,,>

-- Description: <Description,,>

-- =============================================

CREATE PROCEDURE [dbo].[IsRepeatReportProcedure]

-- Add the parameters for the stored procedure here

/\*<@Param1, sysname, @p1> <Datatype\_For\_Param1, , int> = <Default\_Value\_For\_Param1, , 0>, \*/

@X int

AS

BEGIN

-- SET NOCOUNT ON added to prevent extra result sets from

-- interfering with SELECT statements.

SET NOCOUNT ON;

-- Insert statements for procedure here

SELECT

arr.Name,

arr.Description,

arr.KindId,

arr.StatusId,

arr.BeginDate,

arr.BeginTime,

arr.EndDate,

arr.EndTime,

arr.RepeatIntervalDays,

arr.RepeatIntervalWeeks,

arr.RepeatIntervalMonths,

arr.RepeatIntervalYears

FROM Task arr

WHERE arr.IsRepeat = @X

END

GO

USE [master]

GO

ALTER DATABASE [ToDoApp] SET READ\_WRITE

GO

### 3.9 Решения по программному обеспечению

Решения проектирования по программному обеспечению представлены проектом классов (диаграмма классов и описание полей и методов классов) и моделью взаимодействия (диаграмма последовательности, коммуникации, состояния, деятельности).

#### 3.9.1 Классы

* Библиотека Data Access Level
  + Класс TaskDBMsSQL- контекст базы данных
  + Классы моделей
    - Reports
      * Подклассы
        + ReportSelectionByKind\_Entity - модель для отчета выборки задач по виду

Name

Description

StatusId

BeginDate

BeginTime

EndDate

EndTime

IsRepeat

RepeatIntervalDays

RepeatIntervalWeeks

RepeatIntervalMonths

RepeatIntervalYears

* + - * + ReportSelectionByStatus\_Entity - модель для отчета выборки задач по статусу

Name

Description

KindId

BeginDate

BeginTime

EndDate

EndTime

IsRepeat

RepeatIntervalDays

RepeatIntervalWeeks

RepeatIntervalMonths

RepeatIntervalYears

* + - * + ReportSelectionByRepeatability\_Entity - модель для отчета выборки задач по повторяемости

Name

Description

KindId

StatusId

BeginDate

BeginTime

EndDate

EndTime

RepeatIntervalDays

RepeatIntervalWeeks

RepeatIntervalMonths

RepeatIntervalYears

* + - Task - задача
      * Name
      * Description
      * KindId
      * StatusId
      * BeginDate
      * BeginTime
      * EndDate
      * EndTime
      * IsRepeat
      * RepeatIntervalDays
      * RepeatIntervalWeeks
      * RepeatIntervalMonths
      * RepeatIntervalYears
    - TaskKind - вид задачи
      * Id
      * Name
      * Color
    - TaskStatus - статус задачи
      * Id
      * Name
      * Color
  + Классы интерфейсов
    - IDbRepository - интерфейс для классов паттерна «Unit of work» паттерна «Репозиторий»
      * Tasks
      * TaskKinds
      * TaskStatuses
      * Reports
    - IReportsRepository - интерфейс для классов отчетов паттерна «Репозиторий»
      * ReportSelectionByKind
      * ReportSelectionByStatus
      * ReportSelectionByRepeatability
    - IRepository - интерфейс для классов сущностей паттерна «Репозиторий»
      * GetList
      * GetItem
      * Create
      * Update
      * Delete
  + Классы шаблона «Репозиторий»
    - DbRepositorySQL - класс, реализующий паттерн «Unit of work»
      * Tasks
      * TaskKinds
      * TaskStatuses
      * Reports
      * Save()
    - ReportRepositorySQL - классов отчетов паттерна «Репозиторий»
      * ReportSelectionByKind()
      * ReportSelectionByStatus()
      * ReportSelectionByRepeatability()
    - TaskKindRepositorySQL - класс видов задач паттерна «Репозиторий»
      * GetList()
      * GetItem()
      * Create()
      * Update()
      * Delete()
    - TaskRepositorySQL - класс задач паттерна «Репозиторий»
      * GetList()
      * GetItem()
      * Create()
      * Update()
      * Delete()
    - TaskStatusRepositorySQL - класс статусов задач паттерна «Репозиторий»
      * GetList()
      * GetItem()
      * Create()
      * Update()
      * Delete()
* Библиотека Business Logic Level
  + DbDataOperation - класс CRUD для сущностей в базе данных
    - CreateTask()
    - CreateKind()
    - CreateStatus()
    - GetTask()
    - GetAllTasks()
    - GetTaskKind()
    - GetAllTaskKinds()
    - GetAllTaskKindsForCombobox()
    - GetTaskStatus()
    - GetAllTaskStatuses()
    - GetAllTaskStatusesForCombobox()
    - UpdateTask()
    - UpdateTaskKind()
    - UpdateTaskStatus()
    - DeleteTask()
    - DeleteTaskKind()
    - DeleteTaskStatus()
    - Save()
  + Классы Data Transfer Object: предназначены для передачи данных между слоями BLL и PL
    - ReportDTOs - модели отчетов
      * Подклассы
        + SelectionByKindDTO - модель для отчета выборки задач по виду

Name

Description

StatusId

BeginDate

BeginTime

EndDate

EndTime

IsRepeat

RepeatIntervalDays

RepeatIntervalWeeks

RepeatIntervalMonths

RepeatIntervalYears

* + - * + SelectionByStatusDTO - модель для отчета выборки задач по статусу

Name

Description

KindId

BeginDate

BeginTime

EndDate

EndTime

IsRepeat

RepeatIntervalDays

RepeatIntervalWeeks

RepeatIntervalMonths

RepeatIntervalYears

* + - * + SelectionByRepeatabilityDTO - модель для отчета выборки задач по повторяемости

Name

Description

KindId

StatusId

BeginDate

BeginTime

EndDate

EndTime

RepeatIntervalDays

RepeatIntervalWeeks

RepeatIntervalMonths

RepeatIntervalYears

* + - TaskDTO - модель задачи
      * Name
      * Description
      * KindId
      * StatusId
      * BeginDate
      * BeginTime
      * EndDate
      * EndTime
      * IsRepeat
      * RepeatIntervalDays
      * RepeatIntervalWeeks
      * RepeatIntervalMonths
      * RepeatIntervalYears
    - TaskKindDTO - модель вида задачи
      * Id
      * Name
      * Color
    - TaskStatusDTO - модель статуса задачи
      * Id
      * Name
      * Color
  + Классы интерфейсов
    - IDbCrud - интерфейс для классов создания, чтения, обновления, удаления сущностей
      * CreateTask()
      * CreateKind()
      * CreateStatus()
      * GetTask()
      * GetAllTasks()
      * GetTaskKind()
      * GetAllTaskKinds()
      * GetAllTaskKindsForCombobox()
      * GetTaskStatus()
      * GetAllTaskStatuses()
      * GetAllTaskStatusesForCombobox()
      * UpdateTask()
      * UpdateTaskKind()
      * UpdateTaskStatus()
      * DeleteTask()
      * DeleteTaskKind()
      * DeleteTaskStatus()
    - IReportsService - интерфейс для классов создания отчетов
      * ReportSelectionByKind
      * ReportSelectionByStatus
      * ReportSelectionByRepeatability
    - ITaskService - интерфейс для классов, особым образом связанных с логикой работы с классами
  + Классы сервисов
    - ReportsService - класс создания отчетов
      * ReportSelectionByKind
      * ReportSelectionByStatus
      * ReportSelectionByRepeatability
    - TaskService - класс, особым образом связанный с логикой работы с классами
  + ServiceModule - класс для работы сервисов Ninject
* Windows Forms - приложение Presentation Level
  + Program - главный класс программы, с его метода Main() начинается выполнение программы
    - Main()
  + MainForm - главное окно приложения
    - dbCrud
    - reportsService
    - taskService
    - LoadData()
    - buttonAddTask\_Click()
    - buttonTaskEdit\_Click()
    - buttonTaskDelete\_Click()
    - buttonTaskUpdate\_Click()
    - buttonTaskSelectionByKind\_Click()
    - buttonTaskSelectionByStatus\_Click()
    - buttonIsRepeatReport\_Click()
    - GetSelectedRow()
  + DialogAddTask - окно добавления задачи
    - task
    - buttonAddTaskOk\_Click()
  + DialogEditTask - окно изменения задачи
    - task
    - buttonEditTaskOk\_Click
  + DialogAddKind - окно изменения вида задачи
  + DialogEditKind - окно изменения вида задачи
  + DialogAddStatus - окно изменения вида задачи
  + DialogEditStatus - окно изменения вида задачи

### 3.9.2 Взаимодействия

#### 3.9.2.1 Диаграмма последовательности

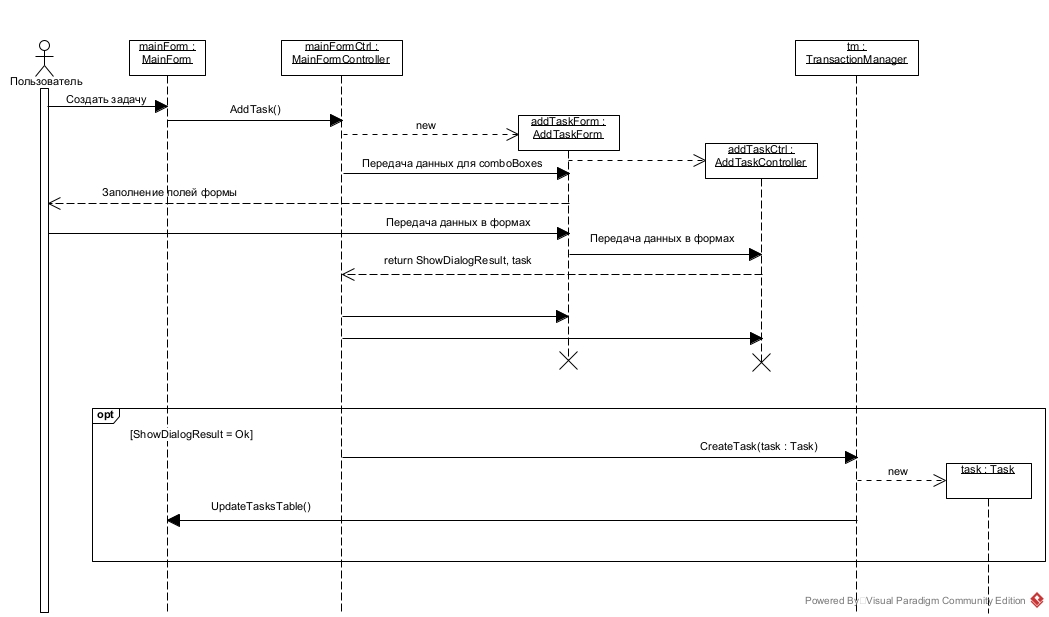


Рис. 3.9.2.1.1. Диаграмма последовательности ВИ «Добавить задачу»

#### 3.9.2.2 Диаграмма классов

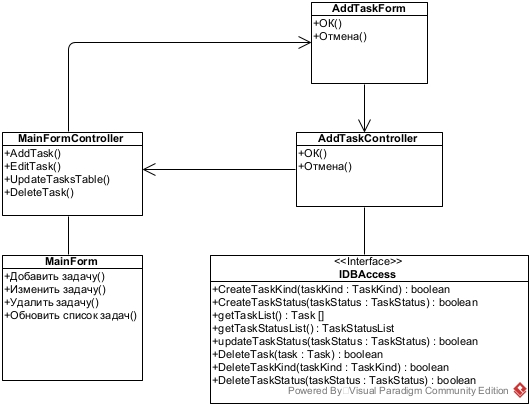


Рис. 3.9.2.1.2. Диаграмма классов варианта «Добавить задачу»

#### 3.9.2.3 Диаграмма коммуникации

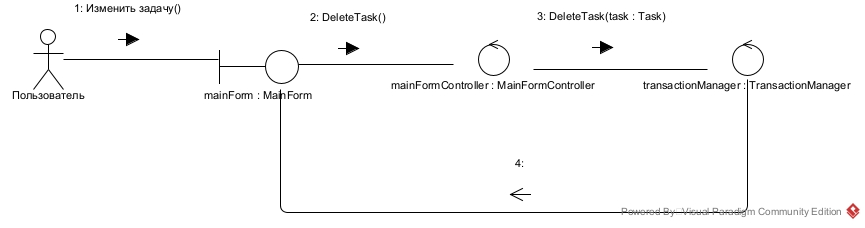


Рис. 3.9.2.2.1. Диаграмма коммуникации варианта использования «Изменить задачу»

#### 3.9.2.4 Диаграмма деятельности

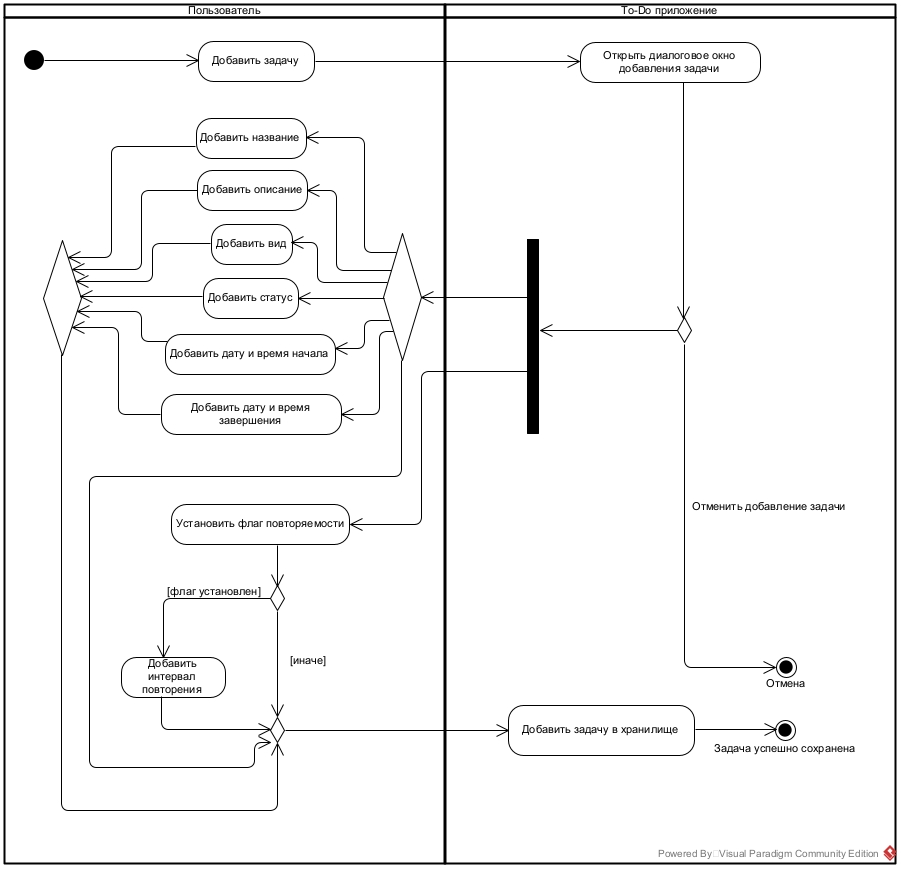


Рис. 3.9.2.4.1. Диаграмма деятельности для варианта использования «Добавить задачу»

## 4 Мероприятия по подготовке к вводу системы в действие

Для ввода Системы в действие необходимо:

1. подготовить у Заказчика всё необходимое техническое обеспечение;
2. установить на сервер и клиентские ПК системное, базовое и прикладное ПО;
3. установить на сервер БД;
4. провести испытания системы.

### 4.1 Мероприятия по приведению информации к виду, пригодному для обработки на ЭВМ

Для ввода системы в эксплуатацию необходимо выполнить первичное наполнение БД:

1. ввести в БД сведения о Задачах.

### 4.2 Мероприятия по обучению и проверке квалификации персонала

Специального обучения Пользователей работе с Системой не требуется.

### 4.3 Мероприятия по созданию необходимых подразделений и рабочих мест

Новые подразделения не создаются.

### 4.4 Мероприятия по изменению объекта автоматизации

Никаких изменений не требуется.

## 5 Источники разработки

При разработке технорабочего проекта использовались следующие источники:

1. ГОСТ 34.003-90. Информационные технологии. Комплекс стандартов на автоматизированные системы. Автоматизированные системы. Термины и определения.
2. ГОСТ 34.201-89. Информационная технология. Комплекс стандартов на автоматизированные системы. Виды, комплектность и обозначение документов при создании автоматизированных систем.
3. ГОСТ 34.601-90 Информационная технология. Комплекс стандартов на автоматизированные системы. Автоматизированные системы. Стадии создания.
4. ГОСТ 34.602-89. Информационная технология. Комплекс стандартов на автоматизированные системы. Техническое задание на создание автоматизированной системы.
5. РД 50-34.698.90. Методические указания. Информационная технология. Комплекс стандартов на автоматизированные системы. Автоматизированные системы. Требования к содержанию документов.
6. РД Гостехкомиссии России. Безопасность информационных технологий. Критерии оценки безопасности информационных технологий.- 2002 г.
7. Практическая программная инженерия на основе учебного примера / Л.А. Мацяшек, Б.Л. Лионг. – М.: БИНОМ. Лаборатория знаний, 2009. – 956 с.
8. Documents associated with. Unified Modeling Language (UML), v2.4.1. Release date: August 2011.- http://www.omg.org/spec/UML/2.4.1.