МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ  
РОССИЙСКОЙ ФЕДЕРАЦИИ

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ  
«ВЯТСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ»

Институт математики и информационных систем

Факультет автоматики и вычислительной техники

Кафедра систем автоматизации управления

Дата сдачи на проверку:

«\_\_\_» \_\_\_\_\_\_\_\_\_\_ 2024 г.

Проверено:

«\_\_\_» \_\_\_\_\_\_\_\_\_\_ 2024 г.

**Разработка информационной системы задачника**

Отчет по практическим работам  
по дисциплине

«Проектирование информационных систем»

Разработали студенты гр. ИТб-3301-01-00 \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ /Олюнин В. К./

(Подпись)

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ /Кузнецов Д. Д./

(Подпись)

Руководитель ст. преподаватель \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ /Кузьминых М. М./

(Подпись)

Работа защищена «\_\_\_\_\_\_\_\_\_\_\_» «\_\_\_» \_\_\_\_\_\_\_\_\_\_ 2024 г.

Киров 2024

Цель работы: закрепление изученного теоретического материала и практических навыков, полученных в ходе освоения курса «Проектирование информационных систем». Для достижения указанной цели, в ходе выполнения работы необходимо пройти все этапы одного из процессов проектирования и решить задачи, стоящие на каждом из этапов.

1. **Бизнес-моделирование**

Целью разработки задачника, является создание системы управления задачами, которая позволит пользователям организовывать и контролировать рабочие процессы. Данная система включает в себя следующие ключевые элементы:

* доски: основные контейнеры для проектов, где организуются задачи;
* типы задач: категории задач, которые можно настраивать под конкретные нужды проекта;
* задачи: конкретные задания, которые помещаются в соответствующие типы задач.
  1. **Роли и пользователи**

Система задачника предусматривает несколько уровней доступа и ролей пользователей, каждая из которых обладает определенным набором прав и обязанностей. Создатель доски является основной ролью, имеющей полный доступ ко всем функциям системы. Этот пользователь может редактировать типы задач и сами задачи, изменять параметры доски, закрывать доски, перемещать задачи между различными типами задач, назначать задачи другим пользователям, а также управлять ролями участников.

Редактор представляет собой роль, предназначенную для пользователей, которым необходимо вносить изменения в задачи, но не требуется полный доступ ко всем функциям системы. Редактор может редактировать задачи и типы задач, назначать задачи другим пользователям, а также перемещать задачи между типами задач.

Исполнитель – это роль для пользователей, которые выполняют задачи, назначенные им редакторами или создателями досок. Исполнители могут перемещать только те задачи, которые были им назначены, между различными типами задач.

* 1. **Процессы и взаимодействия**

Основные процессы, происходящие в системе задачника, включают:

* создание и настройка доски: создатель доски организует новый проект, создавая доску и определяя основные типы задач;
* создание и редактирование типов задач: пользователи с соответствующими правами могут добавлять новые типы задач и изменять существующие;
* создание и управление задачами: в рамках каждой доски редактор или создатель создают задачи, назначают их исполнителям и следят за их выполнением;
* перенос задач: задачи могут перемещаться между типами в зависимости от их статуса и прогресса;
* назначение задач исполнителям: задачи распределяются между участниками проекта, что позволяет четко определить ответственность за выполнение каждой задачи;
* управление ролями и правами пользователей: создатель доски управляет ролями участников, назначая редакторов и исполнителей.
  1. **Нормативные документы и стандарты**

Проектирование и разработка задачника ведется в соответствии с современными стандартами разработки программного обеспечения, такими как:

* Unified Modeling Language (UML): используется для построения объектно-ориентированных моделей системы;
* Rational Unified Process (RUP): итеративный процесс разработки, который обеспечивает четкую структуризацию проекта и этапов разработки.

В данном разделе были разобраны ключевые элементы системы задачника, такие как доски, типы задач и сами задачи. Также были описаны роли пользователей и основные процессы, обеспечивающие организацию и контроль рабочих процессов. Проектирование системы осуществляется в соответствии с современными стандартами UML и RUP.

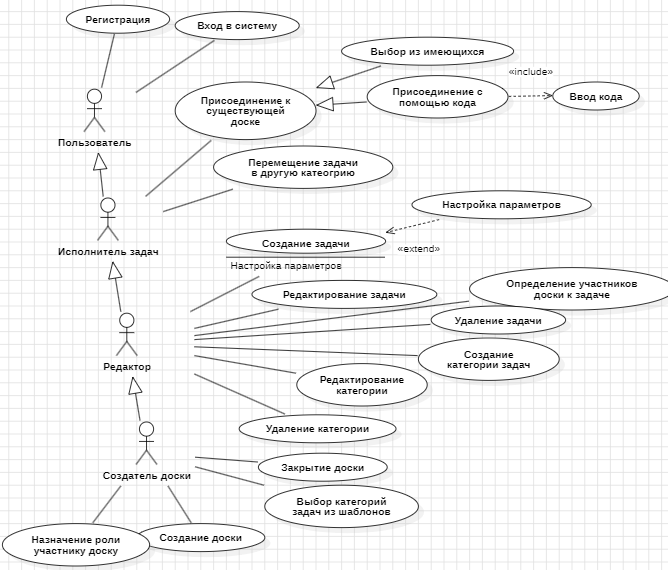
1. **Управление требованиями**

В этом разделе будут выявлены как функциональные, так и нефункциональные требования к проектируемой системе задачника.

* 1. **Функциональные требования**

Поскольку основой является RUP и UML, функциональные требования будут оформляться в виде диаграммы вариантов использования (Use-Case Diagram), а также диаграммы деятельности (Activity Diagram).

На рисунке 1 представлена диаграмма вариантов использования.

  
Рисунок 1 – Диаграмма вариантов использования (Use-Case Diagram)

Данная диаграмма иллюстрирует взаимодействие пользователей с задачником. Диаграмма включает несколько основных ролей пользователей: пользователь, исполнитель задач, редактор и создатель доски. Для каждой роли определены действия (прецеденты), которые они могут выполнять в системе.

Пользователь может зарегистрироваться в системе и выполнить авторизацию для доступа к функционалу задачника.

Исполнитель задач, наследуя функции пользователя, может уже присоединиться к существующей доске, выбрав необходимую доску из списка имеющихся или присоединиться к другой доске с помощью ввода кода. В доске исполнитель может перемещать назначенные ему задачи между различными категориями.

Редактор, помимо всех возможностей исполнителя задач, может создавать и редактировать задачи и категории, а также настраивать параметры задач. Редактор также может назначать участников доски на выполнение конкретных задач.

Создатель доски, обладая всеми функциями редактора, может создавать новые доски для организации проектов, выбирая готовые категории задач из существующих шаблонов. Также создатель управляет ролями участников доски.

Диаграмма вариантов использования показывает основные действия, которые могут выполнять пользователи системы в зависимости от их роли.

Диаграмма деятельности (Activity Diagram) иллюстрирует последовательность действий пользователей в системе задачника. Данная диаграмма представлена на рисунке 2.

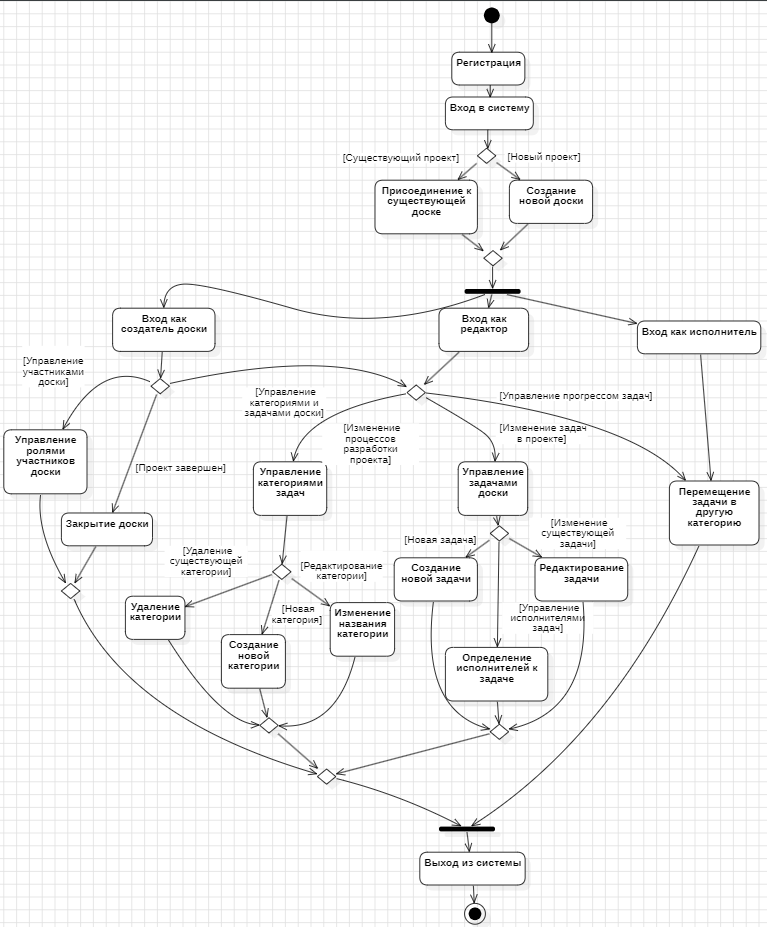
  
Рисунок 2 – Диаграмма деятельности (Activity Diagram)

Диаграмма деятельности описывает поток выполнения различных задач в зависимости от ролей пользователей: пользователь, исполнитель задач, редактор и создатель доски. Диаграмма включает основные действия и решения, с которыми сталкиваются пользователи при работе с задачником.

Процесс начинается с регистрации нового пользователя, после чего пользователь входит в систему. Далее пользователь может выбрать между присоединением к существующей доске или созданием новой доски. Если выбран существующий проект, пользователь присоединяется к существующей доске. Если создается новый проект, происходит создание новой доски.

После этого пользователь может войти в систему в роли исполнителя задач, управляющего задачами или создателя доски. Если пользователь входит как исполнитель задач, он может перемещать задачи в другую категорию в зависимости от их прогресса.

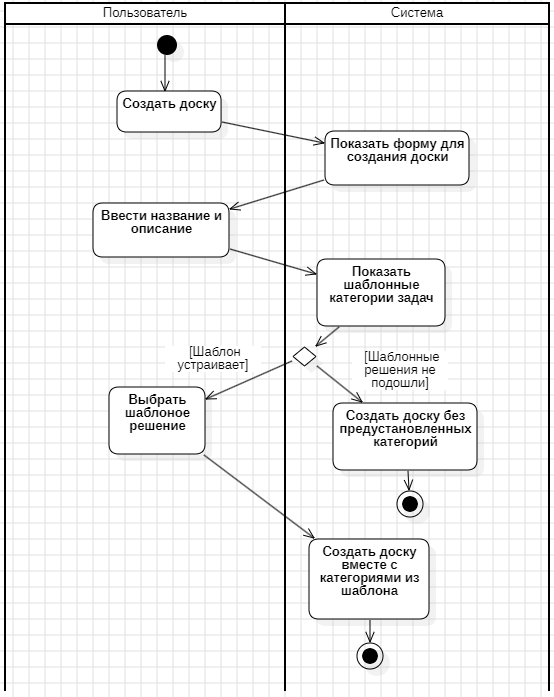
Если пользователь входит как редактор, он может управлять категориями и задачами доски, что включает изменение и создание задач и категорий в проекте, управление исполнителями задач.

Создатель доски, помимо всех функций редактора, может управлять ролями участников доски. Также в случае, если проект завершен, он может закрыть доску.

Основной поток действий завершается выходом пользователя из системы.

Диаграмма активности наглядно показывает основные действия и решения, которые пользователи могут выполнять в системе задачника в зависимости от их роли.

Диаграмма деятельности, представленная на рисунке 3, иллюстрирует процесс создания доски пользователем в системе задачника. Она включает шаги, которые выполняются как пользователем, так и системой.

  
Рисунок 3 – Диаграмма деятельности, рассматривающая процесс создания доски (Activity Diagram)

Процесс начинается с того, что пользователь инициирует создание доски. Система в ответ на это показывает форму для создания доски, где пользователь вводит название и описание будущей доски.

После этого система предлагает пользователю выбрать категории из шаблонных решений. Пользователь может принять одно из предложенных шаблонных решений или отказаться от них.

Если пользователь выбирает шаблонное решение, система создает доску вместе с категориями задач, указанными в выбранном шаблоне. В случае, если шаблонные решения не подошли, система создает доску без предустановленных категорий, предоставляя пользователю возможность добавить категории самостоятельно позже.

Текстовое описание одного из вариантов использования (создание задачи) приведено в Приложении А в формате, предложенном Алистером Коберном. Этот вариант использования описывает процесс, в котором пользователь создает новую задачу в уже существующей доске.

* 1. **Нефункциональные требования**

Исходя из общих соображений о функционировании предметной области, выделяются следующие нефункциональные требования:

1. объемы хранимых и обрабатываемых данных:
   * система должна поддерживать хранение данных о задачах, досках и пользователях, а также историю изменений;
   * объем данных может включать тысячи задач, сотни досок и десятки тысяч пользователей;
2. аппаратные ресурсы:
   * постоянная память: не менее 100 ГБ для хранения данных;
   * оперативная память: не менее 16 ГБ для обеспечения быстрого доступа к данным и выполнения операций;
3. быстродействие:
   * система должна обеспечивать минимальное время отклика при выполнении основных операций;
   * интерфейс пользователя должен загружаться быстро и без задержек, обеспечивая комфортную работу;
4. надежность:
   * система должна быть устойчивой и надежной, обеспечивая высокую доступность и минимальное время простоя;
   * данные должны быть защищены от потери и повреждения.

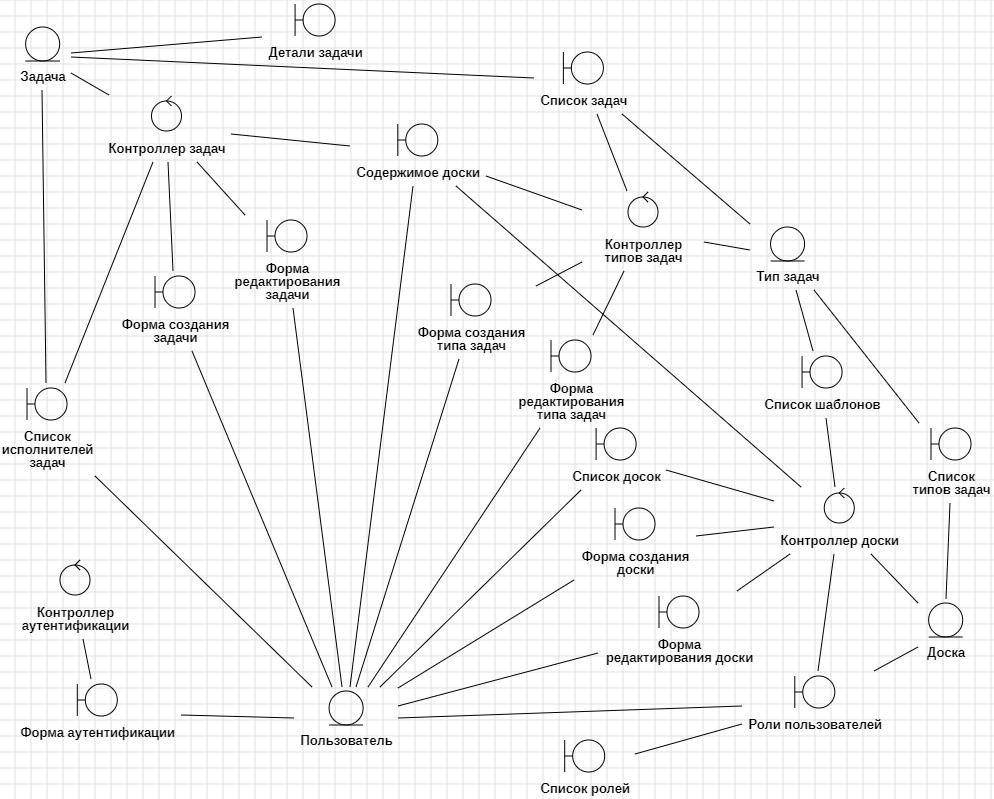
Таким образом, функциональные и нефункциональные требования, выделенные для проектируемой системы задачника, обеспечивают реализацию всех необходимых функций для управления задачами.

1. **Анализ предметной области**

В данном разделе будут проанализированы классы, существующие в предметной области, а также описано их взаимодействие в рамках реализации вариантов использования. Также будет рассмотрена архитектура будущей системы.

* 1. **Классы и их взаимодействие**

На рисунке 4 представлена диаграмма классов этапа анализа

Рисунок 4 – Диаграмма классов этапа анализа

В системе выделены четыре ключевые сущности: пользователь, доска, тип задач и задача. Для управления этими сущностями предусмотрены четыре контроллера: контроллер доски, контроллер типов задач, контроллер задач и контроллер аутентификации.

Пользователь взаимодействует с системой через различные формы для создания и редактирования задач, досок и типов задач. Для аутентификации пользователя используется контроллер аутентификации, связанный с формой аутентификации.

Доска содержит различные типы задач. Для хранения информации о типах задач, содержащихся на доске, используется граничный класс «Список типов задач». Тип задач включает определенный набор задач. Для каждого типа задач может существовать определенный набор задач, для этого предусмотрен граничный класс «Список задач». Задача имеет связанные подробности, которые хранятся в граничном классе «Детали задачи».

Пользователь может взаимодействовать с досками, типами задач и задачами с помощью соответствующих форм создания и редактирования. Для определения функционала пользователя относительно конкретной доски существует связь между «Роли пользователей» и «Контроллер доски». Каждый пользователь может иметь разные роли на разных досках. Для этого предусмотрены граничные классы «Список ролей» и «Роли пользователей». «Роли пользователей» содержат информацию о том, какие роли занимают пользователи относительно различных досок.

Для отображения досок используется граничный класс «Список досок», а для определения содержимого доски используется граничный класс «Содержимое доски», который связан с контроллерами доски, типов задач и задач одновременно.

При создании доски пользователь может выбрать шаблонный набор типов задач. Для этого используется граничный класс «Список шаблонов», который содержит эти наборы типов задач.

Для задач могут быть назначены конкретные исполнители. Для этого используется граничный класс «Список исполнителей задач».

Таким образом, диаграмма классов наглядно иллюстрирует структуру и взаимодействие ключевых сущностей системы задачника, а также способы управления этими сущностями через различные формы и контроллеры.

Для упорядочения выявленных классов был применен анализ по категориям, результат которого представлен в виде таблицы 1 в приложении Б.

* 1. **Описание архитектуры системы**

**Приложение А**  
(обязательное)  
**Текстовое описания варианта использования**

Создать задачу

**Контекст использования:** пользователь создает новую задачу в уже существующей доске.

**Область действия:** система управления задачами.

**Уровень:** подфункция.

**Основное действующее лицо:** редактор или создатель доски.

**Участники и интересы:**

1. **пользователь**: хочет создать задачу для отслеживания рабочего процесса;
2. с**истема:** должна предоставить интерфейс для создания задачи и сохранить данные в базе данных.

**Предусловие:** пользователь прошел аутентификацию; создана доска; имеется хотя бы один тип задач; пользователь имеет роль редактора или создателя доски.

**Минимальные гарантии:** система должна обеспечить сохранение данных о созданной задаче, даже в случае частичных ошибок.

**Гарантии успеха:** новая задача создана и видна пользователю в соответствующем типе задач.

**Триггер:** пользователь инициирует создание новой задачи.

**Основной сценарий:**

1. **пользователь открывает доску;**
2. **пользователь выбирает тип задач, в который нужно добавить новую задачу;**
3. **пользователь нажимает на кнопку "Создать задачу";**
4. **система отображает форму для создания задачи;**
5. **пользователь заполняет поля: название, описание, дата дедлайна (выбирает исполнителей по желанию);**
6. п**ользователь нажимает кнопку "Сохранить";**
7. **система сохраняет задачу в базе данных;**
8. **система подтверждает создание задачи и отображает её в списке задач выбранного типа.**

**Расширения:**

1. пользователь не заполнил обязательные поля (описание): система отображает сообщение об ошибке и просит заполнить обязательные поля.
2. ошибка при сохранении данных: система показывает сообщение об ошибке и предлагает повторить попытку.

**Список изменений в технологии и данных:**

1. создание нового объекта задачи в базе данных: таблица задач (добавляется новая запись).

**Вспомогательная информация:** необходимо наличие подключенной базы данных для хранения задач.

**Приложение Б**  
(обязательное)  
**Анализ по категориям**

Таблица 1 – упорядочение выявленных классов по категориям

|  |  |  |
| --- | --- | --- |
| № | Категория | Классы |
| 1 | Физические или материальные объекты |  |
| 2 | Спецификации, описания объектов |  |
| 3 | Места |  |
| 4 | Транзакции |  |
| 5 | Элементы транзакций |  |
| 6 | Роли людей |  |
| 7 | Контейнеры других объектов |  |
| 8 | Содержимое контейнеров |  |
| 9 | Другие компьютеры и прочие системы, внешние по отношению к создаваемой |  |
| 10 | Организации |  |
| 11 | События |  |
| 12 | Процессы |  |
| 13 | Каталоги |  |
| 14 | Записи о деятельности |  |