



МИНОБРНАУКИ РОССИИ

Федеральное государственное бюджетное образовательное учреждение
высшего образования
«МИРЭА – Российский технологический университет»

РТУ МИРЭА

Институт информационных технологий (ИИТ)
Кафедра математического обеспечения и стандартизации информационных технологий
(МОСИТ)

ОТЧЕТ ПО ПРАКТИЧЕСКОЙ РАБОТЕ
по дисциплине «Обоснование и разработка требований к программным
системам»

Практическое занятие № 5

Вариант № 13. Обоснование и разработка требований к программной
системе управления разработками цифровой компании

Студент группы *ИКБО-65-23, Олефиров ГГ*

(подпись)

Преподаватель *Бирюкова А.А.*

(подпись)

Отчет представлен «__»____202__г.

Москва 2025 г.

1. Цель работы

Визуализировать поток деятельности с использованием UML диаграммы деятельности, детально представить внутреннюю структуру системы в виде низкоуровневых классов и связей между ними.

2. Постановка задачи

Задачами работы являются:

- визуализировать конкретный случай использования в виде деятельности;
- построить диаграмму классов и сделать вывод по представлению внутренней структуры системы.

В третьей четвертой работе была разработана следующая диаграмма классов анализа:

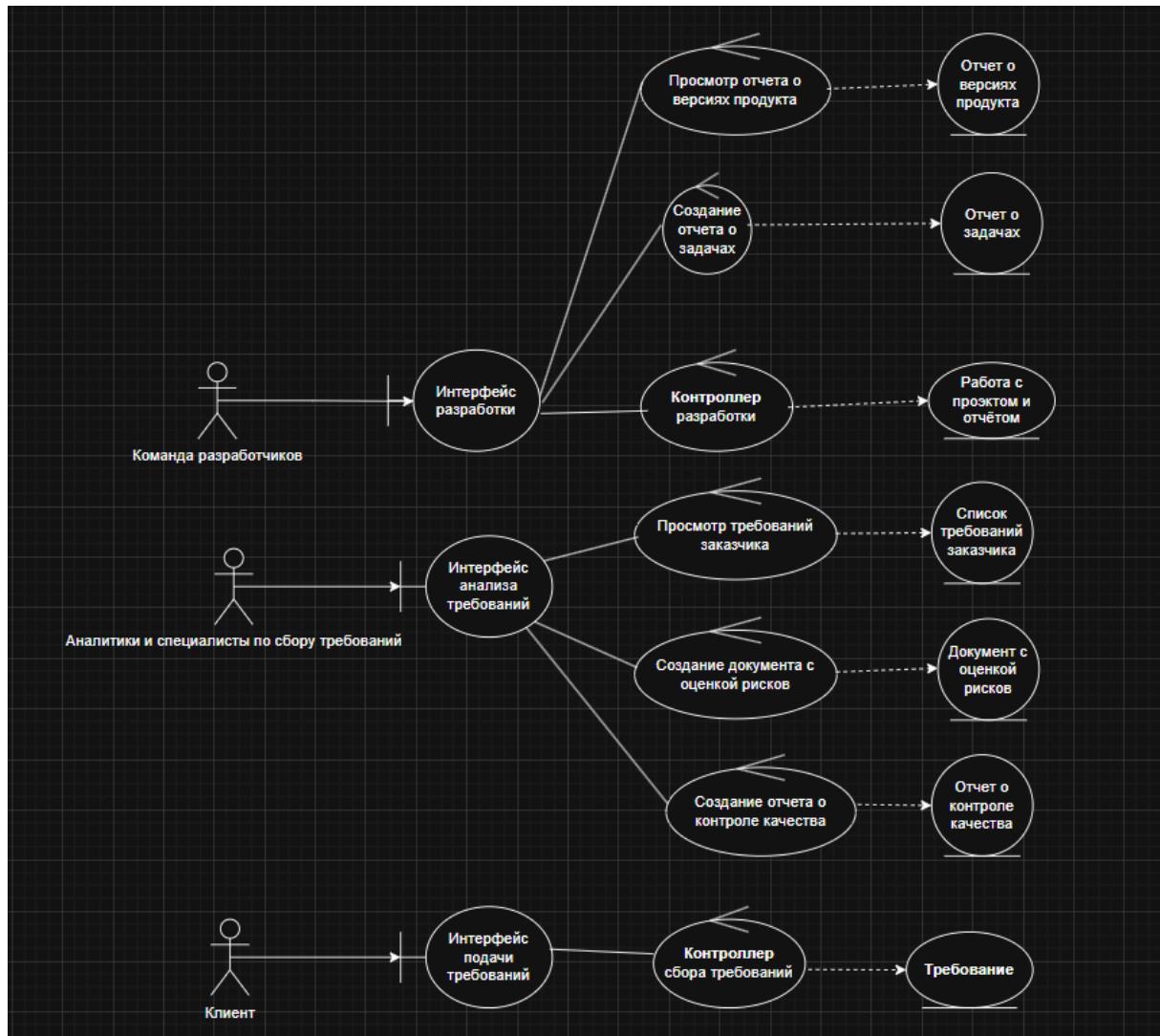


Рисунок 1 - Итоговая единная диаграмма классов анализа

3. Результат работы

3.1. Визуализация потока деятельности (Диаграмма деятельности)

Диаграммы деятельности, или активности, предназначены для уточнения вариантов использования и моделей последовательности.

Построим диаграмму деятельности для варианта использования «управления разработками цифровой компании».

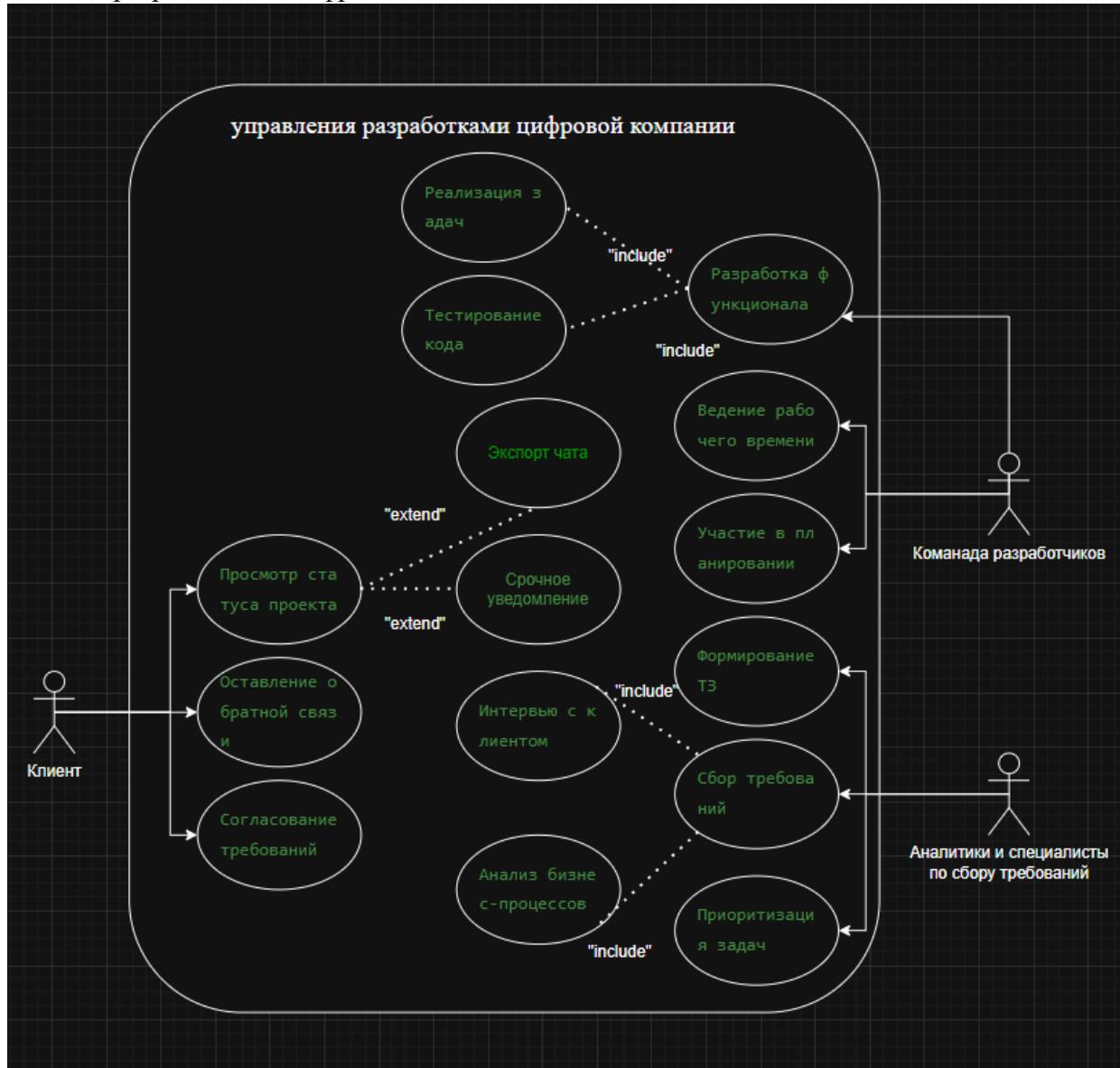


Рисунок 2 – Диаграмма вариантов использования для управления разработками цифровой компании

Первая диаграмма деятельности разработана для варианта использования "Основной процесс разработки". На диаграмме отражен полный цикл работ от получения требований до завершения разработки. Процесс включает четыре дорожки ответственности: Аналитики, Специалисты по требованиям, Команда разработчиков и Информационная система.

Поток управления начинается с анализа требований заказчика, после чего выполняется проверка их полноты через узел решения. В зависимости от результата производится либо формализация требований, либо их уточнение. Далее процесс разделяется на параллельные ветви анализа ресурсов и детализации требований с последующим соединением. Завершающие этапы включают сохранение требований в системе, получение и выполнение задач разработчиками, обновление статуса работ [2].

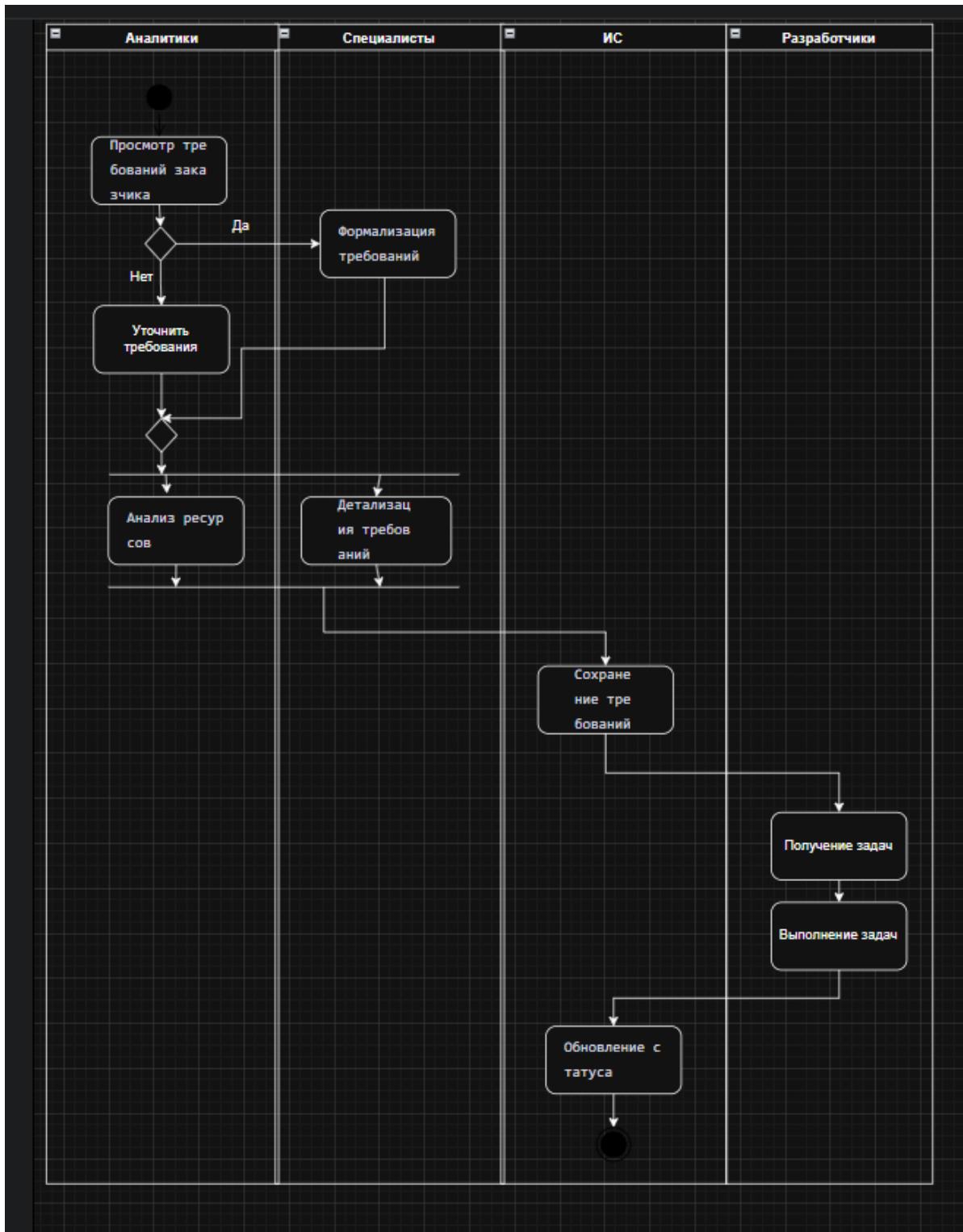


Рисунок 3 - Диаграмма основной процесс разработки

Вторая диаграмма деятельности отражает процесс "Контроля и корректировок проекта", который соответствует диаграмме последовательности из предыдущей работы. Процесс включает три дорожки: Аналитики, Команда разработчиков и Информационная система.

Деятельность начинается с запроса статуса проекта, после чего параллельно формируются отчеты о задачах и ресурсах. После анализа прогресса через узел решения определяется соответствие проекта графику. При отклонениях инициируются корректирующие действия и вносятся изменения в задачи разработки [3].

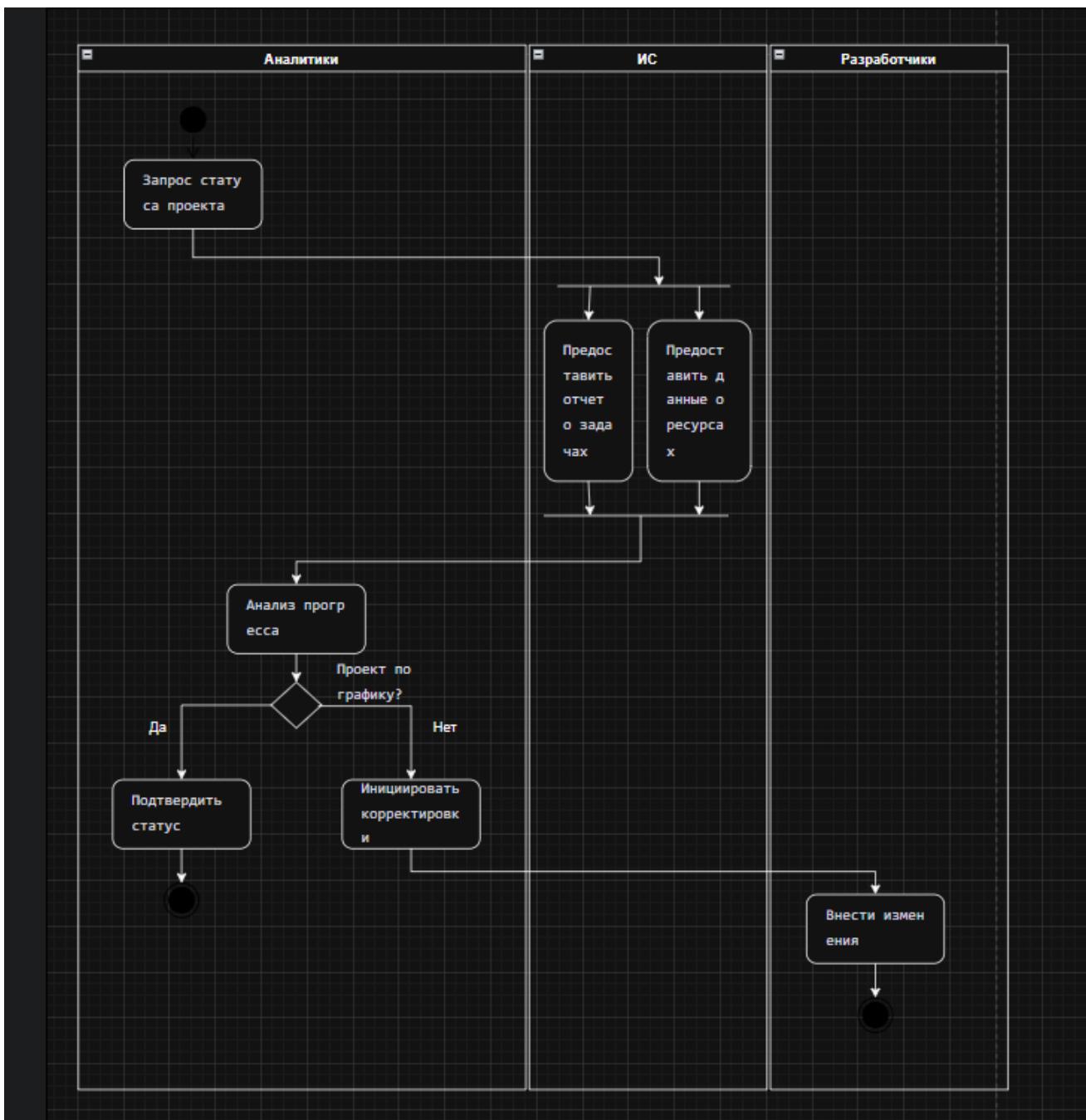


Рисунок 4 - Диаграмма процесс контролия и корректировок

3.2. Внутренняя структура системы в виде диаграммы классов

(Диаграммы классов уровня проектирования/реализации)

Построим диаграмму классов для системы управления разработками цифровой компании. Разработка выполнялась поэтапно в соответствии с методическими рекомендациями [1].

На первом этапе определены названия классов, отражающие ключевые сущности системы управления разработками. Были выделены 8 основных классов: Analyst (Аналитик), RequirementSpecialist (Специалист по требованиям), DeveloperTeam (Команда разработчиков), InformationSystem (Информационная система), Project (Проект), Requirement (Требование), Task (Задача) и ProgressReport (Отчет о прогрессе). Данный набор классов охватывает всех участников процесса разработки и основные артефакты управления проектами.

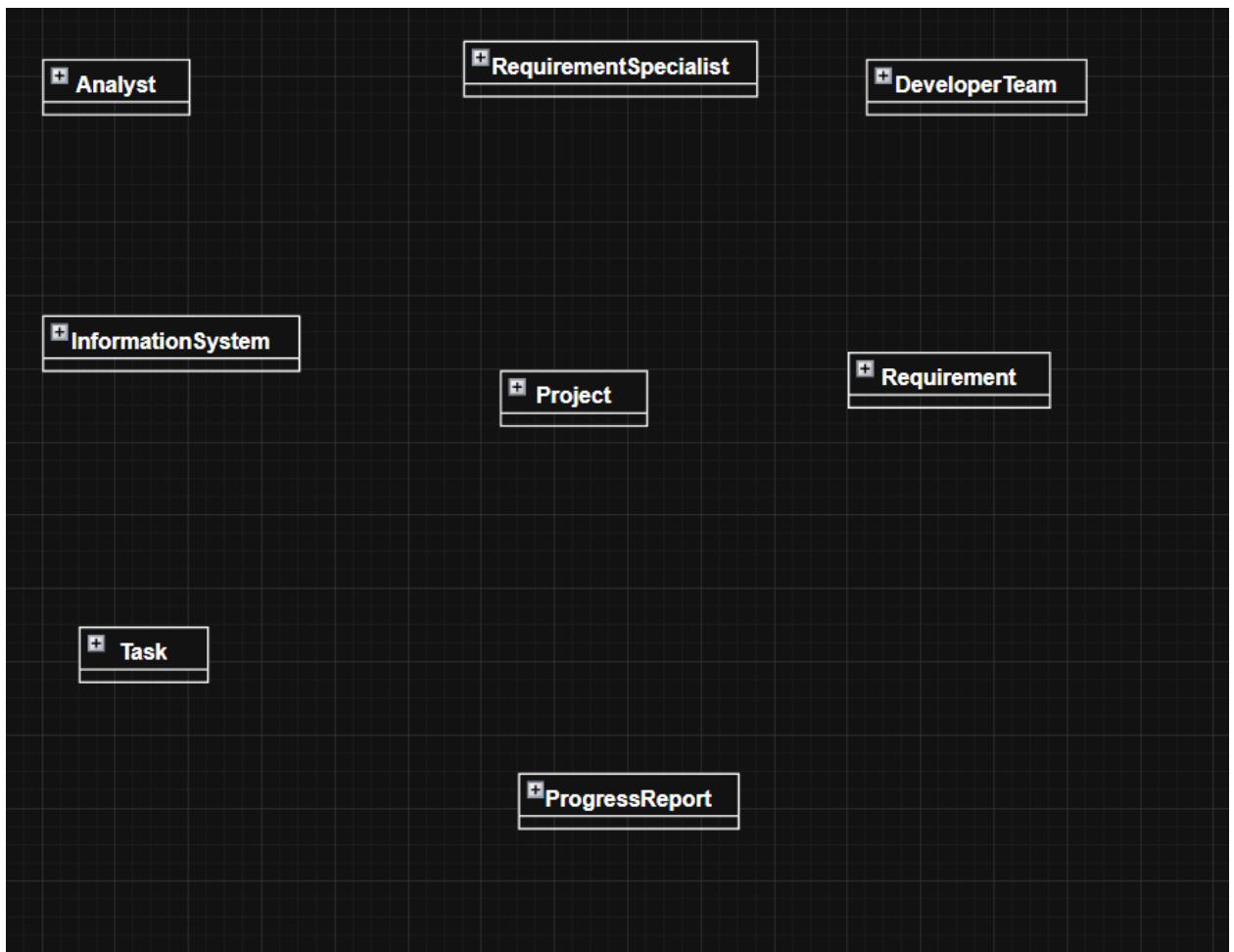


Рисунок 5 – Диаграмма классов, этап 1

На втором этапе определены атрибуты и операции классов с указанием модификаторов видимости в соответствии с нотацией UML [2]. Для каждого класса заданы уникальные идентификаторы (ID), основные характеристики и ключевые методы, обеспечивающие функциональность системы. Например, класс Analyst содержит атрибуты analystId, name, email и методы analyzeRequirements(), evaluateRisks(), requestProgressReport(), что отражает responsibilities аналитика в процессе разработки.

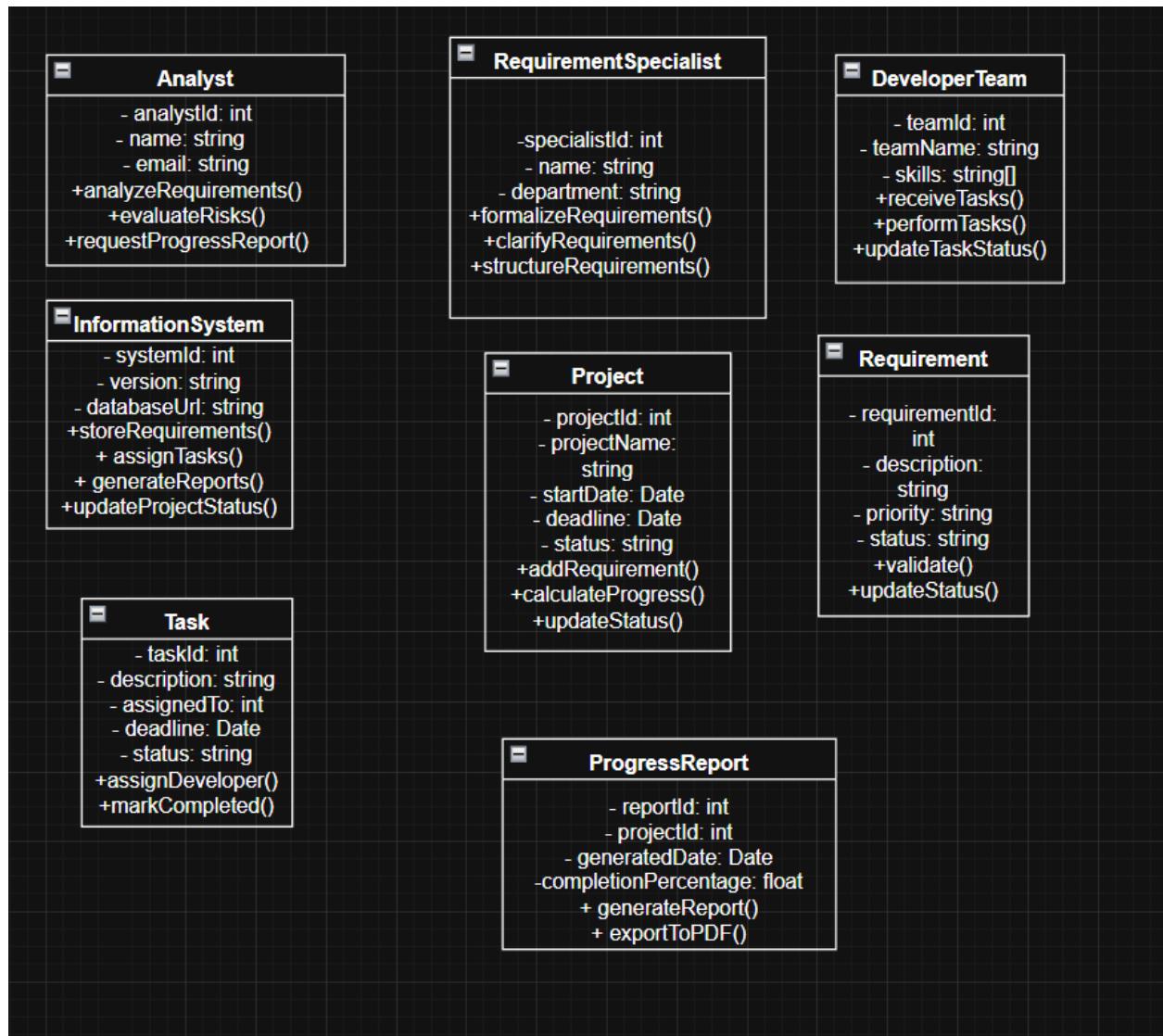


Рисунок 6 – Диаграмма классов, этап 2

На третьем этапе установлены связи между классами с использованием четырех видов отношений UML [3]. Применены:

- Ассоциация между Analyst и Project, RequirementSpecialist и Requirement, DeveloperTeam и Task
- Агрегация между Project и Requirement, Project и Task, Project и ProgressReport

- Композиция между InformationSystem и Project, InformationSystem и Requirement
- Зависимость между Requirement и Task, Analyst и ProgressReport

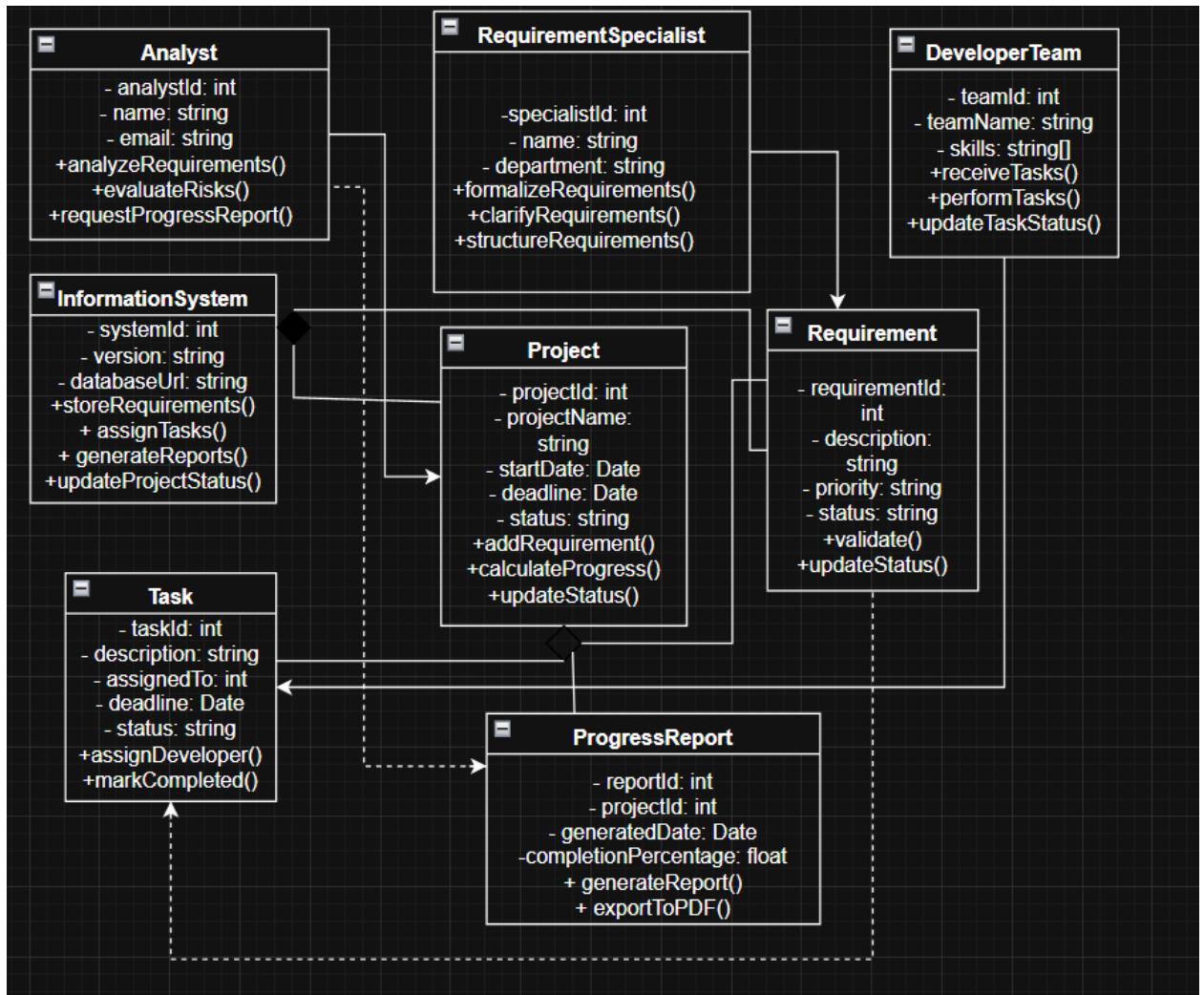


Рисунок 7 – Диаграмма классов, этап 3

Диаграмма классов тесно связана с ранее разработанной диаграммой классов анализа и полностью покрывает функционал, отраженный на диаграмме вариантов использования системы управления разработками. Все отношения между классами использованы корректно в соответствии с семантикой UML.

Разработанная диаграмма демонстрирует целостное представление внутренней структуры системы, где четко распределены responsibilities между классами и определены механизмы их взаимодействия, что обеспечивает основу для последующей реализации системы.

Список использованных источников и литературы:

1. Ахмедова, Х. Г. Обоснование и разработка требований к программным системам : учебное пособие / Х. Г. Ахмедова. — Москва : РТУ МИРЭА, 2023. — 104 с. — ISBN 978-5-7339-1934-8. — Текст : электронный // Лань : электронно-библиотечная система. — URL: <https://e.lanbook.com/book/382694> (дата обращения: 22.09.2025). — Режим доступа: для авториз. пользователей.
2. Ахмедова, Х. Г. Обоснование и разработка требований к программным системам : учебно-методическое пособие / Х. Г. Ахмедова, А. В. Овсянникова, А. А. Бирюкова. — Москва : РТУ МИРЭА, 2023. — 68 с. — ISBN 978-5-7339-2010-8. — Текст : электронный // Лань : электронно-библиотечная система. — URL: <https://e.lanbook.com/book/398132> (дата обращения: 20.10.2025).
3. Брусникин, Г. Н. Разработка UML-моделей при проектировании информационных систем : учебное пособие / Г. Н. Брусникин, Н. Ю. Соколова. — Москва : МИЭТ, 2023. — 52 с. — ISBN 978-5-7256-1016-1. — Текст : электронный // Лань : электронно-библиотечная система. — URL: <https://e.lanbook.com/book/461570> (дата обращения: 20.10.2025).