Федеральное государственное автономное

образовательное учреждение

Высшего образования

«СИБИРСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»

Институт космических и информационных технологий

институт

Программная инженерия

кафедра

**ОТЧЕТ О ПРАКТИЧЕСКОЙ РАБОТЕ №1**

Сбор и тестирование требований

тема

|  |
| --- |
| Преподаватель |
| подпись, дата                    инициалы, фамилия |
| Студент     КИ23-17/1Б, 32321533                                         Р. А. Карандаев |
| номер группы, зачетной книжки           подпись, дата                    инициалы, фамилия |
| Студент     КИ23-17/1Б, 32324850                                      М. В. Музалевский |
| номер группы, зачетной книжки           подпись, дата                    инициалы, фамилия |
|  |

Красноярск 2025

# Цель

Данная работа направлена на развитие комплексных навыков анализа, формализации и верификации требований, а также на отработку эффективного взаимодействия в команде при решении реальных задач проектирования. Студенты учатся выявлять существенные аспекты бизнес-потребностей, корректно структурировать и оформлять требования с применением стандартных моделей, что способствует более глубокому пониманию процессов коммуникации с заказчиком и повышения качества проектной документации.

Задачи

– Выбрать один вариант из списка предполагаемых бизнес-требований;

– Составить перечень вопросов для уточнения бизнес-требований и на их основе сформировать пользовательские требования (преподаватель выступает в роли представителя заказчика и отвечает на вопросы из списка);

– Формализовать пользовательские требования с применением UML-диаграммы прецедентов;

– Предоставить набор пользовательских требований своему коллеге для анализа и формирования замечаний;

– Скорректировать пользовательские требования с учетом замечаний, уточнив при необходимости у преподавателя недостающую информацию;

– Сформировать проектные требования и повторно предоставить своему коллеге для анализа и формирования замечаний;

– Устранить замечания и оформить отчет, содержащий информацию со всех этапов формирования требований.

# Ход выполнения

Был выбран второй вариант работы. Ниже скопирован его текст.

Приложение для планирования выполнения домашних заданий (в школе/университете и тд). Учет заданного, дата наиболее позднего возможного момента для начала выполнения, учет времени просрочки “дедлайна”. Каждому заданию выдавать приоритет.

Ниже представлено интервью с заказчиком.

1. Вопрос: "Какую основную проблему должно решать приложение?" Ответ: "Распределение времени на выполнение заданий".

2. Вопрос: "Какими основными функциями должно обладать приложение (минимально необходимый функционал)?" Ответ: "Создать задачу, назначить ей длительность, показ, когда человек должен начать выполнять задачу, чтобы успеть к указанному пользователем времени".

3. Вопрос: "Требуется ли регистрация и авторизация пользователей?" Ответ: "Да".

4. Вопрос: "Какого уровня защиты данных ожидается? (Нужны ли шифрование, соответствие GDPR и т.п.)" Ответ: "Нет".

5. Вопрос: "Будет ли приложение бесплатным, платным или условно-бесплатным (Freemium)?" Ответ: "Бесплатным".

6. Вопрос: "Рассматривается ли размещение рекламы?" Ответ: "Нет".

7. Вопрос: "На какую возрастную категорию (школьники, студенты вуза, аспиранты и т.д.) ориентировано приложение?" Ответ: "Пользоваться может любой".

8. Вопрос: "Насколько технически подкованы предполагаемые пользователи?" Ответ: "В состоянии заполнить форму, пользоваться Goggle календарем".

9. Вопрос: "На каких платформах (iOS, Android, Web) планируется выпуск приложения?" Ответ: "Web".

10. Вопрос: "Как должен определяться приоритет: вручную пользователем или автоматически по каким-то критериям (сложность, срочность и т.д.)?" Ответ: "Вручную".

11. Вопрос: "Сколько уровней приоритетов планируется: низкий/средний/высокий или более гибкая градация?" Ответ: "Пять уровней: очень высокий, высокий, нормальный, низкий, очень низкий".

12. Вопрос: "Какие статусы заданий необходимы? (Например, «Назначено», «В процессе», «Выполнено», «Просрочено» и т.д.)" Ответ: "К выполнению, выполняется, на паузе, отменено, выполнено, удалено".

13. Вопрос: "Хотите ли вы, чтобы приложение само рассчитывало «наиболее поздний момент начала выполнения» задачи на основе заданного дедлайна и оценочного времени выполнения?" Ответ: "Оно и должно этим заниматься".

14. Вопрос: "Нужно ли учитывать регулярные (повторяющиеся) задания?" Ответ: "Да, мы можем создавать регулярные, повторяющиеся задания".

15. Вопрос: "Будет ли функциональность календаря (с визуализацией дат и дедлайнов) встроенной в приложение?" Ответ: "Да".

16. Вопрос: "Нужно ли автоматизировать смену статуса при наступлении дедлайна или окончании времени выполнения?" Ответ: "Нет, статус меняется вручную пользователем. Если дедлайн наступил, то система должна уведомить пользователя и предложить выбрать выполнено или невыполнено задание".

17. Вопрос: "Как вы планируете использовать информацию о времени просрочки: для уведомлений, рейтингов, статистики?" Ответ: "Для уведомлений".

18. Вопрос: "Требуется ли какая-то специальная логика обработки просроченных заданий? (Например, отдельный список или выделение цветом)" Ответ: "Просроченные задания могут быть выделены цветом".

19. Вопрос: "Нужно ли учитывать время, выделяемое на внезапные обстоятельства (например, дополнительный буфер перед дедлайном)? Может ли пользователь самостоятельно регулировать этот буфер?" Ответ: "В таком случае мы ставим задачу на паузу".

20. Вопрос: "Требуется ли учитывать тип задания (реферат, лабораторная работа, проект, тест и т.д.)? Планируется ли хранить разную дополнительную информацию в зависимости от типа?" Ответ: "Нет".

21. Вопрос: "Есть ли необходимость указывать, что выполнение одного задания зависит от завершения другого? (Например, «Часть 2» задачи может быть начата только после «Части 1».)" Ответ: "Да".

22. Вопрос: "Нужно ли позволять нескольким пользователям совместно работать над одним заданием (делить прогресс, видеть общие дедлайны)? Требуются ли уведомления для каждого участника группы об изменениях статуса?" Ответ: "Нет. У нас нет параллельности задач, нет параллельности пользователей".

23. Вопрос: "Нужно ли собирать статистику о том, сколько времени уходит на задания разных типов, сложностей, дисциплин и т.д.?" Ответ: "Можно сделать".

24. Вопрос: "Планируется ли система поощрений (очков, бейджей, рейтингов) за своевременное выполнение заданий?" Ответ: "Нет".

25. Вопрос: "Нужно ли импортировать задания из внешних источников (PDF-документов, LMS, электронных журналов)? Требуется ли возможность экспорта расписания или списка заданий в Excel, PDF или другие форматы?" Ответ: "Импорт задач из google календаря".

26. Вопрос: "Нужно ли настраивать язык интерфейса, поддержку нескольких языков?" Ответ: "Поддержка мультиязычности".

27. Вопрос: "Нужно ли интегрировать приложение с другими календарями (Google Calendar, Outlook и т.д.)?" Ответ: "Google, Yandex календарь".

28. Вопрос: "Требуются ли интеграции с системами обучения (например, Moodle, Google Classroom)?" Ответ: "Нет".

29. Вопрос: "Какие роли пользователей планируются? (Студент, Преподаватель, Администратор)" Ответ: "Ролей нет. Но есть отдельно админстратор".

30. Вопрос: "Нужно ли хранить вложения (файлы, ссылки, изображения), связанные с заданием?" Ответ: "Нужно".

31. Вопрос: "Нужно ли учитывать каникулы, праздничные и выходные дни, чтобы не ставить дедлайны на такие даты?" Ответ: "Нет. Это можно подсвечивать, но дедлайн можно поставить в такие дни".

Предварительный список пользовательских требований написан ниже.

1. Приложение должно позволять создавать задачи с названием, дедлайном и оценочным временем.

2. Приложение должно позволять редактировать задачи (название, дедлайн, время, приоритет, статус).

3. Приложение должно позволять удалять задачи вручную; для этого предусмотрен статус «Удалено».

4. Должны быть статусы: «К выполнению», «Выполняется», «На паузе», «Отменено», «Выполнено», «Удалено»; переключаются только вручную.

5. Должно быть 5 уровней приоритета: «Очень высокий», «Высокий», «Нормальный», «Низкий», «Очень низкий» (устанавливаются вручную).

6. Приложение должно поддерживать повторяющиеся (регулярные) задания (ежедневные, еженедельные и др.).

7. Приложение должно позволять указывать зависимости между заданиями (одно не начинается до завершения другого).

8. Приложение должно позволять прикреплять файлы, ссылки, изображения к заданию.

9. Приложение должно автоматически рассчитывать «наиболее поздний момент начала» (по дедлайну и оценочному времени).

10. Приложение должно иметь встроенный календарь с визуализацией заданий и дедлайнов.

11. Просроченные задания должны выделяться цветом и требовать ручного выбора статуса («Выполнено» или «Не выполнено»).

12. Приложение может подсвечивать выходные и праздничные дни, но не блокирует постановку дедлайна на них.

13. Пользователь может ставить задачу «На паузу» при внезапных обстоятельствах.

14. Приложение должно уведомлять о приближении дедлайна (тип и момент уведомления можно настраивать).

15. При наступлении дедлайна и невыполненной задаче приложение должно предложить выбор: «Выполнено» или «Не выполнено».

16. Приложение должно позволять выбирать интервал уведомлений (за день, за час и т.п.).

17. Приложение должно уметь импортировать события из Google и Яндекс Календарей в виде задач.

18. Приложение должно поддерживать несколько языков интерфейса (мультиязычность).

19. Приложение должно обеспечивать регистрацию и авторизацию (логин/пароль).

20. Нет требований о повышенной защите (нет шифрования или GDPR).

21. В системе есть две роли: обычный пользователь и администратор (управление пользователями и настройками).

22. Нет совместного редактирования задач несколькими пользователями.

23. Приложение может формировать статистику по выполненным и просроченным задачам.

24. Приложение не содержит системы поощрений (нет очков, рейтингов, бейджей).

25. Приложение должно быть бесплатным и без рекламы.

26. Пользователь может назначать дедлайн на любую дату, включая выходные и праздники.

27. Интерфейс должен быть понятным для пользователей с базовыми навыками (заполнение форм, использование календаря).

28. Приложение должно позволять сортировать задачи (по дате, приоритету, статусу) и фильтровать (только «К выполнению», только «Просроченные» и т.д.).

29. Приложение должно иметь поиск по названию или описанию задач.

30. Приложение должно отображать задачи в списке и в календарном виде.

31. Приложение не меняет статус задачи автоматически при наступлении дедлайна (только предлагает выбрать).

32. При желании можно добавить расширенную статистику (фактическое время vs. оценочное, количество просрочек).

33. Желательно иметь дашборд или главный экран с ближайшими дедлайнами и общим состоянием задач.

34. Приложение не поддерживает параллельное выполнение нескольких задач.

35. Повторяющиеся задания должны поддерживать разные варианты периодичности (день, неделя, месяц и т.д.).

36. Платформа, на которой будет работать приложение, – web.  
 В ходе анализа предварительных требований выявлены следующие недостатки.

1. Расчёт «наиболее позднего момента» и зависимости. Необходимо уточнить, как происходит автоматический пересчёт последнего возможного времени начала выполнения задачи при изменении дедлайна или оценочного времени, а также каким образом система обрабатывает зависимости между задачами, особенно если сроки или оценки меняются, и что делать в случае пересечения дедлайнов для зависимых задач.

2. Повторяющиеся (регулярные) задания. Следует детально описать, как генерируются повторяющиеся задачи – создаётся ли серия автоматически или генерируется следующая итерация по наступлении нужного интервала, как задаётся дата окончания повторений и как система обрабатывает просроченные или пропущенные итерации.

3. Статусы. «Не выполнено», «Отменено» и «Удалено»: Требуется чётко разграничить понятия и сценарии использования статусов, объяснить, в каких случаях применяется «Не выполнено», чем отличается «Отменено» от «Удалено», и можно ли восстановить задачу из этих состояний.

4. Приоритеты. Необходимо зафиксировать, что приоритеты устанавливаются и изменяются пользователем вручную, а также уточнить, что система не меняет их автоматически при изменении дедлайна или просрочке, что важно для понимания логики планирования.

5. Уведомления и их логика. Требуется описать механизм уведомлений – как именно происходит оповещение о приближении дедлайна, какие каналы используются (например, email, браузерные уведомления), как система действует при паузе задачи и повторно ли оповещает, если пользователь не отвечает.

6. Обработка «пауз» и незавершённых задач. Нужно прояснить, как постановка задачи на «паузу» влияет на расчёт «наиболее позднего момента начала» и дедлайны, а также описать возможность возобновления задачи без потери предыдущего прогресса.

7. Отсутствие параллельного выполнения. Требуется уточнить, как система обрабатывает ситуации с пересекающимися дедлайнами, если пользователь планирует выполнять задачи последовательно, а также как приложение предупреждает о возможных временных конфликтах.

8. Импорт из Google и Яндекс Календарей. Необходимо описать, осуществляется ли импорт в виде однократной синхронизации или поддерживается двусторонняя синхронизация, а также каким образом система предотвращает дублирование задач при импорте.

9. Учет реального времени и тайм-зон. Важно зафиксировать, учитываются ли часовые пояса при планировании, особенно если пользователь находится в разных регионах, и насколько точно (по минутам или часам) определяется время выполнения задачи.

10. Регистрация, авторизация и роли. Следует уточнить функционал администрирования – какие именно возможности предоставляет администратор, имеет ли он доступ к задачам пользователей и какие базовые механизмы безопасности (например, восстановление пароля) предусмотрены, даже если повышенная защита не требуется.

11. Статистика. Необходимо детализировать, какие метрики будут собираться и отображаться (например, количество выполненных, просроченных задач, сравнение оценочного и фактического времени), а также описать механизм ввода фактического времени выполнения, если оно будет использоваться для расширенной статистики.

12. Файлы и вложения. Требуется указать допустимые типы файлов, ограничения по размеру и количеству вложений, а также определить, что происходит с файлами при изменении статуса задачи, например, при её удалении или отмене.

13. Гибкость в настройке календаря. Следует уточнить, как происходит подсветка выходных и праздничных дней, какие праздники включаются по умолчанию и возможно ли задавать собственные нерабочие дни, а также как это влияет на установку дедлайнов.

14. Отсутствующие уточнения о времени выполнения. Важно описать формат задания оценочного времени (часы, дни, минуты), а также как приложение рассчитывает время начала задачи, если требование к дедлайну связано с точным поминутным планированием.

15. Редактирование и удаление задач. Необходимо зафиксировать, можно ли редактировать задачи, уже переведённые в статусы «Выполнено» или «Отменено», и какие ограничения накладываются на изменение параметров задачи после определённых действий.

16. Сортировка, фильтры и поиск. Требуется детально описать логику сортировки задач по разным критериям (дата, приоритет, статус), механизм фильтрации и расширенный поиск, включая возможность поиска по дополнительным параметрам, таким как статус или дата дедлайна.

17. Мультиюзерная часть. Следует уточнить, как реализована работа с пользователями – может ли администратор видеть задачи всех пользователей, и какие групповые настройки или функции управления (кроме стандартной регистрации) предусмотрены, учитывая отсутствие совместного редактирования.

18. Локализация (мультиязычность). Нужно зафиксировать перечень поддерживаемых языков, а также описать, как переключение языка влияет на отображаемые данные и сохраняется ли введённая пользователем информация на исходном языке.

19. Логика при просрочке. Требуется уточнить, как система обрабатывает просроченные задачи – изменяется ли их статус автоматически до выбора пользователем, как отображается состояние задачи, если пользователь не реагирует на уведомление, и какие действия предлагаются для исправления ситуации.

20. Дальнейшие сценарии развития. Следует обозначить опциональность расширенной статистики и возможных будущих модулей, таких как детальный анализ фактического времени выполнения, чтобы в дальнейшем можно было легко расширять функциональность без нарушения базовой логики работы приложения.

Исправленные и уточненные требования представлены ниже.

1. Приложение должно позволять создавать задачи с названием, дедлайном и оценочным временем. В процессе создания учитывается часовой пояс пользователя (если он задаётся в профиле), чтобы корректно рассчитывать дату и время дедлайна. Формат длительности может задаваться в часах и минутах, а при необходимости — в днях.

2. Приложение должно позволять редактировать задачи (название, дедлайн, время, приоритет, статус). При изменении дедлайна или оценочного времени система пересчитывает «наиболее поздний момент начала», учитывая часовой пояс и возможные зависимости от других задач. В случае, если изменение затрагивает зависимые задачи, приложение предлагает обновить их параметры (дедлайн, длительность и т.д.).

3. Приложение должно позволять удалять задачи вручную; для этого предусмотрен статус «Удалено». Статус «Удалено» означает, что пользователь не планирует возвращаться к задаче. Администратор или сам пользователь могут при необходимости вернуть удаленную задачу в активное состояние.

4. Должны быть статусы: «К выполнению», «Выполняется», «На паузе», «Отменено», «Выполнено», «Удалено»; переключаются только вручную. «Отменено» указывает на то, что задача больше не актуальна и не будет выполняться; «Удалено» — пользователь сознательно удалил задачу из общего списка. При наступлении дедлайна невыполненную задачу можно отметить как «Не выполнено» (что логически приравнивается к отменённой или проваленной задаче) или «Выполнено», но без автоматической смены статуса системой.

5. Должно быть 5 уровней приоритета: «Очень высокий», «Высокий», «Нормальный», «Низкий», «Очень низкий» (устанавливаются вручную). Приложение не изменяет приоритет автоматически при наступлении дедлайна, просрочке или иных событиях: изменение приоритета полностью контролирует пользователь.

6. Приложение должно поддерживать повторяющиеся (регулярные) задания (ежедневные, еженедельные и др.). Необходимо уметь задавать периодичность (день, неделя, месяц и т.д.) и при желании ограничивать серию датой окончания повторов. Если одна итерация задачи просрочена, отменена или не выполнена, последующие итерации тем не менее могут формироваться по заданному расписанию, а пользователь сам решает, переносить ли эти задания или пропускать их.

7. Приложение должно позволять указывать зависимости между заданиями (одно не начинается до завершения другого). При изменении статуса (например, завершении) или параметров предшествующей задачи приложение предлагает пересчитать «наиболее поздний момент начала» для всех зависимых задач, чтобы избежать рассинхронизации в сроках.

8. Приложение должно позволять прикреплять файлы, ссылки, изображения к заданию. Допустимый тип и объём вложений может быть настроен администратором. При переводе задачи в «Удалено» или «Отменено» файлы будут удаляться.

9. Приложение должно автоматически рассчитывать «наиболее поздний момент начала» (по дедлайну и оценочному времени). Пересчёт происходит всякий раз, когда меняются дедлайн, длительность или взаимосвязанные задачи, с учётом того, что пользователь не выполняет задачи параллельно и может ставить какую-то задачу на паузу.

10. Приложение должно иметь встроенный календарь с визуализацией заданий и дедлайнов. Он должен корректно обрабатывать часовой пояс пользователя, а также показывать подсветку выходных и праздничных дней (если включена соответствующая настройка) без запрета на постановку дедлайна в эти даты.

11. Просроченные задания должны выделяться цветом и требовать ручного выбора статуса («Выполнено» или «Не выполнено»). Пока пользователь не примет решение, такая задача остаётся в состоянии «просрочено» с визуальным выделением; при необходимости приложение может продолжать уведомлять об этом.

12. Приложение может подсвечивать выходные и праздничные дни, но не блокирует постановку дедлайна на них. Список праздничных дней либо задаётся администратором, либо определяется в зависимости от региона пользователя. Эти дни носят лишь рекомендательный характер, а установленные задачи можно выполнять в любые даты.

13. Пользователь может ставить задачу «На паузу» при внезапных обстоятельствах. На паузе задача не считается просроченной, а уведомления можно временно приостановить по настройкам. При снятии задачи с паузы приложение пересматривает «наиболее поздний момент начала» с учётом фактического смещения во времени.

14. Приложение должно уведомлять о приближении дедлайна (тип и момент уведомления можно настраивать). Уведомления могут приходить за день, за час или в другой выбранный пользователем интервал; при желании можно настроить повторные напоминания до тех пор, пока задача не получит окончательный статус.

15. При наступлении дедлайна и невыполненной задаче приложение должно предложить выбор: «Выполнено» или «Не выполнено». Если пользователь не реагирует, задача остаётся выделенной как просроченная, и система может периодически повторять запрос на определение статуса, чтобы не оставлять задачу «подвешенной».

16. Приложение должно позволять выбирать интервал уведомлений (за день, за час и т.п.). Механизм уведомлений должен учитывать часовой пояс пользователя и подстраивать время оповещения, чтобы оно было актуальным для конкретного региона.

17. Приложение должно уметь импортировать события из Google и Яндекс Календарей в виде задач. Необходимо предусмотреть логику обработки дубликатов при повторном импорте, а также решить, нужна ли двусторонняя синхронизация (обратная выгрузка изменений в сторонний календарь) или достаточно одностороннего импорта.

18. Приложение должно поддерживать несколько языков интерфейса (мультиязычность). Переключение языка переводит все системные тексты и уведомления, но не изменяет пользовательские названия, описания и вложенные файлы; задачу, введённую на одном языке, приложение не переименует автоматически.

19. Приложение должно обеспечивать регистрацию и авторизацию (логин/пароль). Несмотря на отсутствие строгих требований по защите данных, необходимо иметь базовую безопасность (например, HTTPS, механизм восстановления пароля, ограничение попыток входа) для защиты учётных записей.

20. Нет требований о повышенной защите (нет шифрования или GDPR). При этом рекомендуется использовать стандартные инструменты (TLS/HTTPS) для шифрования соединения, чтобы минимизировать риски утечек.

21. В системе есть две роли: обычный пользователь и администратор (управление пользователями и настройками). Администратор определяет глобальные настройки (например, лимиты вложений, праздничные дни, языки) и может модерировать учётные записи.

22. Нет совместного редактирования задач несколькими пользователями. Каждая задача принадлежит одному конкретному пользователю, не предполагается общий доступ или разделение прогресса; любые групповые сценарии выполняются вне данного приложения.

23. Приложение может формировать статистику по выполненным и просроченным задачам. Пользователь может видеть сводку по количеству завершённых или отменённых заданий, а также общий процент просрочек за определённый период, что способствует анализу загрузки.

24. Приложение не содержит системы поощрений (нет очков, рейтингов, бейджей). Основной акцент делается на функциональности планирования, а не на геймификации.

25. Приложение должно быть бесплатным и без рекламы. Пользователь не должен сталкиваться с принудительными платными функциями, баннерами или прочими маркетинговыми элементами, отвлекающими от задач.

26. Пользователь может назначать дедлайн на любую дату, включая выходные и праздники. Приложение только подсвечивает такие дни, но полностью оставляет свободу планирования на усмотрение пользователя.

27. Интерфейс должен быть понятным для пользователей с базовыми навыками (заполнение форм, использование календаря). Никакие сложные действия (вроде параллельного редактирования или масштабной интеграции) не предусмотрены, всё сводится к формам и календарному виду.

28. Приложение должно позволять сортировать задачи (по дате, приоритету, статусу) и фильтровать (только «К выполнению», только «Просроченные» и т.д.). Если пользователь выбирает несколько критериев сортировки, нужно определить их приоритет (например, сначала приоритет, потом дата).

29. Приложение должно иметь поиск по названию или описанию задач. Поиск осуществляется по любым ключевым словам, введённым пользователем, и выводит соответствующие результаты в удобном виде (список или календарь).

30. Приложение должно отображать задачи в списке и в календарном виде. Каждый пользователь волен переключаться между этими представлениями, выбирая наиболее удобный формат планирования и контроля за дедлайнами.

31. Приложение не меняет статус задачи автоматически при наступлении дедлайна (только предлагает выбрать). Если пользователь не выбрал, задача отображается как просроченная, но формально её статус остаётся прежним (например, «К выполнению»), пока пользователь не зафиксирует решение.

32. При желании можно добавить расширенную статистику (фактическое время vs. оценочное, количество просрочек). Если пользователь укажет фактическое время выполнения, приложение может выводить сравнительный анализ для выявления расхождений между планом и реальностью, а также сохранять историю задержек.

33. Желательно иметь дашборд или главный экран с ближайшими дедлайнами и общим состоянием задач. Такой обзор позволяет быстро увидеть, какие задачи требуют внимания, какие задачи на паузе или просрочены, и какие приоритеты выставлены.

34. Приложение не поддерживает параллельное выполнение нескольких задач. Если у разных задач совпадают дедлайны, приложение может лишь предупреждать о возможном временном конфликте, но не предлагает механизмов для одновременного трекинга. Считается, что пользователь планирует задачи последовательно.

35. Повторяющиеся задания должны поддерживать разные варианты периодичности (день, неделя, месяц и т.д.). Приложение должно корректно обрабатывать разрыв серии (пропуск или отмену отдельного экземпляра) без нарушения расписания последующих повторений и при необходимости позволять пользователю вручную корректировать каждую итерацию.

36. Платформа – web. Приложение должно работать с большинством популярных браузеров, таких как Google Chrome, Microsoft edge, Mozilla Firefox, Apple Safari, Opera, Brave и другими. Причем интерфейс должен быть адаптирован под разрешение мониторов пк, экранов телефонов и планшетов.

На рисунке 1 изображена диаграмма прецедентов.

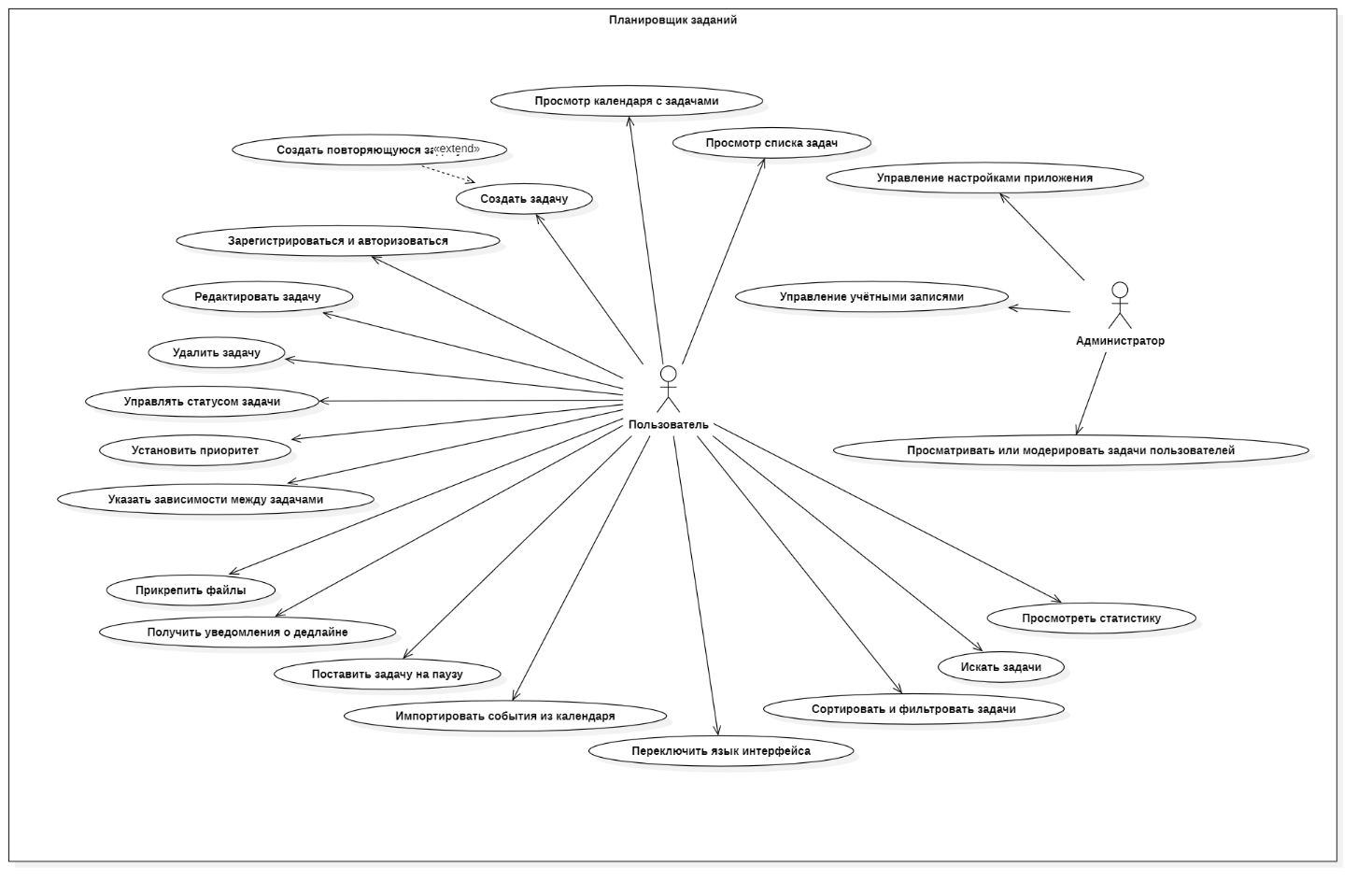


Рисунок 1 – Диаграмма прецедентов

Ниже перечислены основные варианты использования.

Use Case 1. Регистрация и авторизация. Актор: Пользователь. Предусловия: приложение доступно, пользователь не авторизован (для регистрации). Основной поток: пользователь открывает приложение, регистрируется (вводит логин, пароль, e-mail), система проверяет и создаёт учётную запись, затем пользователь авторизуется (вводит логин/пароль), система даёт доступ к личному кабинету. Альтернативно (логин занят, неверный пароль) система сообщает об ошибке. Результат: пользователь авторизован или получает уведомление о невозможности входа.

Use Case 2. Создать задачу. Актор: Пользователь. Предусловия: пользователь авторизован. Основной поток: пользователь нажимает «Создать задачу», вводит название, дедлайн, оценочное время, приоритет, при необходимости указывает повторяемость и зависимости, система рассчитывает «наиболее поздний момент начала» и создаёт задачу, которая появляется в списке/календаре. Альтернативно (неверно введённые данные) система сообщает об ошибке. Результат: новая задача добавлена в приложение.

Use Case 3. Редактировать задачу. Актор: Пользователь. Предусловия: пользователь авторизован, задача уже существует. Основной поток: пользователь выбирает задачу, меняет нужные параметры (название, дедлайн, оценочное время, приоритет, статус и т.д.), система пересчитывает «наиболее поздний момент начала» и предлагает обновить связанные задачи; пользователь подтверждает и изменения сохраняются. Альтернативно (некорректные данные) система просит исправить. Результат: задача обновлена согласно новым параметрам.

Use Case 4. Удалить задачу. Актор: Пользователь. Предусловия: пользователь авторизован, задача не в статусе «Удалено». Основной поток: пользователь выбирает задачу, нажимает «Удалить» или переключает статус на «Удалено», система запрашивает подтверждение, после чего переводит задачу в «Удалено» (скрывает из общего списка). Альтернативно (отмена действия) задача остаётся доступной. Результат: задача скрыта или окончательно удалена (в зависимости от настроек).

Use Case 5: Управлять статусами задачи. Актор: Пользователь. Предусловия: пользователь авторизован, задача доступна для изменения. Основной поток: пользователь вручную переключает статус («К выполнению», «Выполняется», «На паузе», «Отменено», «Выполнено»), система обновляет отображение; при наступлении дедлайна система предлагает «Выполнено» или «Не выполнено». Альтернативно (отказ или выбор «На паузе») система приостанавливает уведомления, а при возобновлении пересчитывает сроки. Результат: задача имеет актуальный статус, учитываемый в списке и календаре.

Use Case 6: Импортировать события из календаря. Актор: Пользователь. Предусловия: пользователь авторизован, есть доступ к Google/Яндекс Календарю. Основной поток: пользователь заходит в раздел «Импорт», авторизуется в сервисе (OAuth), система загружает список событий, пользователь выбирает, что импортировать, система создаёт соответствующие задачи (название, дата, при необходимости дедлайн). Альтернативно (ошибка авторизации или дубликаты) система запрашивает повтор, пропуск или отмену. Результат: выбранные события добавлены как задачи в приложении.

Use Case 7: Сортировать, фильтровать и искать задачи. Актор: Пользователь. Предусловия: пользователь авторизован, есть задачи в системе. Основной поток: пользователь открывает список/календарь, выбирает параметры сортировки (по дате, приоритету, статусу), фильтрует (например, только просроченные) и при необходимости вводит текст для поиска, система отображает задачи, соответствующие критериям. Альтернативно (сложная мультисортировка) система уточняет порядок сортировки. Результат: пользователь видит нужный поднабор задач удобным образом.

Use Case 8: Управление учётными записями. Актор: Администратор. Предусловия: администратор авторизован с правами админа. Основной поток: администратор заходит в «Управление пользователями», видит список, может создавать новые учётные записи, блокировать, удалять или сбрасывать пароли; система подтверждает действия. Альтернативно (ошибка при вводе данных) система сообщает об ошибке. Результат: список пользователей и их статусы обновлены, доступ контролируется.

Use Case 9: Управление настройками приложения. Актор: Администратор. Предусловия: администратор авторизован с правами админа. Основной поток: администратор переходит в «Настройки приложения», меняет лимиты на вложения, праздничные дни, языки, политику восстановления удалённых задач и т.д., система применяет изменения. Альтернативно (ошибка формата данных) система просит исправить. Результат: новые настройки вступают в силу, влияют на всех пользователей.

Use Case 10: Просмотреть статистику. Актор: Пользователь (или Администратор, если требуется глобальная статистика). Предусловия: актор авторизован, есть выполненные, просроченные или отменённые задачи. Основной поток: пользователь открывает «Статистику», система показывает количество выполненных/просроченных/отменённых задач, при наличии фактического времени сравнивает с оценочным. Альтернативно (нет внесённого фактического времени) сравнения не отображаются. Результат: пользователь видит обобщённую информацию, администратор может смотреть агрегированные данные по всем пользователям.

Пользовательские истории представлены ниже.

«Как пользователь, я хочу зарегистрироваться в системе (указать логин/пароль), чтобы иметь возможность создавать и управлять своими задачами.»

«Как администратор, я хочу иметь возможность добавлять новых пользователей, чтобы упростить процесс регистрации для определённой группы.»

«Как пользователь, я хочу создавать новые задачи с указанием дедлайна и оценки времени, чтобы планировать их выполнение и знать, когда нужно начать.»

«Как пользователь, я хочу изменять параметры существующей задачи (название, дедлайн, приоритет), чтобы оперативно реагировать на изменения и поддерживать актуальный список дел.»

«Как пользователь, я хочу удалять ненужные задачи или переводить их в статус «Удалено», чтобы не засорять основной список задач.»

«Как пользователь, я хочу переключать статус задачи (например, «На паузе» или «Выполнено»), чтобы адекватно отражать текущий прогресс и приостанавливать работу при необходимости.»

«Как пользователь, я хочу иметь возможность отмечать задачу как «Не выполнено», если я пропустил дедлайн, чтобы аналитика по просроченным задачам была корректной.»

«Как пользователь, я хочу вручную выставлять приоритет (от «Очень высокий» до «Очень низкий»), чтобы понимать, какие задачи самые важные и требуют срочного внимания.»

«Как пользователь, я хочу создавать повторяющиеся задачи (ежедневные, еженедельные и т.п.), чтобы не вводить одни и те же задания вручную каждый раз.»

«Как пользователь, я хочу указывать, что одна задача не может начаться, пока не будет выполнена другая, чтобы правильно планировать порядок работы над связанными задачами.»

«Как пользователь, я хочу прикреплять файлы и ссылки к задаче, чтобы хранить в одном месте все материалы, необходимые для её выполнения.»

«Как пользователь, я хочу, чтобы система высчитывала, когда мне нужно начать задачу, исходя из дедлайна и оценочного времени, чтобы я мог более эффективно распределять своё время.»

«Как пользователь, я хочу видеть задачи в календаре и в виде списка, чтобы мне было удобно планировать и быстро находить нужную информацию.»

«Как пользователь, я хочу получать уведомления за определённое время до дедлайна (за час, за день и т.д.), чтобы не забыть вовремя начать и закончить задачу.»

«Как пользователь, я хочу иметь возможность приостанавливать выполнение задачи («На паузе») в случае непредвиденных обстоятельств, чтобы перераспределить своё время.»

«Как пользователь, я хочу импортировать события из моего Google или Яндекс Календаря, чтобы все задачи и события хранились в одном приложении.»

«Как пользователь, я хочу переключать язык интерфейса на удобный мне (например, русский/английский), чтобы использовать приложение на том языке, который я лучше понимаю.»

«Как пользователь, я хочу сортировать задачи по приоритету, дате или статусу, а также фильтровать и искать по названию, чтобы быстро находить нужные задачи.»

«Как пользователь, я хочу смотреть статистику по выполненным и просроченным задачам, чтобы понимать, насколько я укладываюсь в сроки и где нужно улучшить планирование.»

«Как администратор, я хочу управлять учётными записями пользователей (создавать, удалять, блокировать), чтобы поддерживать актуальную базу пользователей и их прав.»

«Как администратор, я хочу менять глобальные настройки приложения (праздничные дни, лимиты на вложения, языки), чтобы обеспечить корректную работу для всех пользователей.»

«Как пользователь, я хочу при наступлении дедлайна выбирать «Выполнено» или «Не выполнено» и видеть выделение просроченных задач, чтобы не терять контроль над графиком и своевременно реагировать.»

Списком ниже перечислены функциональные требования.

1. Регистрация и авторизация. Пользователь должен иметь возможность зарегистрироваться, указав логин/пароль, и авторизоваться в системе для доступа к личному кабинету.

2. Управление учётными записями (для администратора). Администратор может создавать, удалять, блокировать учётные записи пользователей, а также восстанавливать им пароли и управлять общими настройками приложения (праздничные дни, лимиты на вложения, языки и т. п.).

3. Создание задач. Пользователь может создавать новые задачи с указанием названия, дедлайна, оценочного времени и (опционально) приоритета, регулярности (повторений), а также прикреплять файлы.

4. Редактирование задач. Пользователь может изменять параметры существующих задач (название, дедлайн, длительность, приоритет, повторяемость, вложения и т. д.). При изменении дедлайна или длительности пересчитывается «наиболее поздний момент начала».

5. Удаление задач (статус «Удалено»). Пользователь может переводить задачу в статус «Удалено» для скрытия из активного списка; при необходимости допускается восстановление (если политика приложения это поддерживает).

6. Управление статусами. Пользователь может вручную переключать статусы задач («К выполнению», «Выполняется», «На паузе», «Отменено», «Выполнено», «Удалено»). При наступлении дедлайна без результата система просит выбрать «Выполнено» или «Не выполнено».

7. Приоритеты задач. Пользователь назначает приоритет (пять уровней: от «Очень высокого» до «Очень низкого») вручную, система не изменяет приоритет автоматически.

8. Повторяющиеся задачи. Поддержка регулярных заданий с разными вариантами периодичности (день, неделя, месяц и т. д.). Можно указать дату завершения повторений или оставить их бессрочными.

9. Зависимости между задачами. Возможность указать, что задача B не может начаться до выполнения задачи A. При изменении дедлайна или статуса задачи A система предлагает пересчитать время для задачи B.

10. Прикрепление файлов, ссылок, изображений. Пользователь может загружать вложения к задаче; администратор задаёт правила (форматы, объём, максимальное количество).

11. Рассчёт «наиболее позднего момента начала». Система автоматически рассчитывает, когда нужно начать задачу, учитывая дедлайн и оценочное время. При обновлении параметров (например, при постановке задачи «На паузу») расчёт пересматривается.

12. Уведомления о приближении дедлайна. Пользователь может настраивать момент (за час, за день и т. п.) и способ получения уведомлений (e-mail, push-уведомления). При наступлении дедлайна задача помечается как просроченная (если не выполнена).

13. Импорт событий из Google/Яндекс Календарей. Пользователь может импортировать внешние события в виде задач, при этом система проверяет дубликаты и создаёт новые записи в списке/календаре приложения.

14. Отображение задач (список и календарь). Приложение предоставляет календарный вид с подсветкой дедлайнов и выделением выходных/праздничных дней, а также список задач, где пользователь может сортировать и фильтровать по статусу, приоритету, дате и т. д.

15. Сортировка, фильтрация и поиск. Пользователь может сортировать задачи (по дате, приоритету, статусу), фильтровать (только «К выполнению», только просроченные и т. д.) и осуществлять поиск по названию или описанию.

16. Локализация интерфейса (мультиязычность). Приложение должно поддерживать переключение языка для системных сообщений, меню и уведомлений. Введённые пользователем данные (названия задач) не переводятся автоматически.

17. Статистика выполнения. Пользователь может просматривать статистику по выполненным и просроченным задачам (количество, сроки). При желании можно учитывать фактическое время выполнения, чтобы анализировать расхождения с оценкой.

18. Настройки приложения (администратор). Администратор определяет лимиты на вложения, список праздничных дней, политику хранения «удалённых» задач, а также включает/отключает функции (например, расширенную статистику).

19. Постановка задачи «На паузу». Пользователь может временно приостановить выполнение задачи, при этом система пересчитывает сроки при повторном запуске. Уведомления об этой задаче можно отключить или приостановить на время паузы.

Нефункциональные требования сформулированы ниже.

1. Производительность и масштабируемость. Приложение должно обрабатывать запросы от нескольких десятков (или сотен) одновременно активных пользователей без существенного снижения скорости отклика. При увеличении числа пользователей архитектура должна позволять горизонтальное или вертикальное масштабирование (добавление ресурсов сервера, распределённые инстансы).

2. Доступность и надёжность. Система должна быть доступна для пользователей большую часть времени (например, 99.9% в рабочие часы), чтобы обеспечить непрерывную работу с задачами. В случае сбоя или перезагрузки сервера восстановление должно происходить быстро (в пределах нескольких минут).

3. Безопасность на базовом уровне. Хотя нет требований о повышенной защите (GDPR и сложном шифровании), система должна обеспечивать безопасное хранение паролей (например, хеширование) и защищённую передачу данных (HTTPS). Администраторские функции должны быть доступны только пользователям с ролью «Администратор», и неавторизованные пользователи не должны иметь доступ к чужим задачам.

4. Удобство (usability) для базовых пользователей. Приложение рассчитано на людей, умеющих заполнять формы и пользоваться календарными сервисами, поэтому все ключевые функции (создание задач, смена статуса, настройка уведомлений) должны быть доступны в несколько кликов, без сложных дополнительных шагов. В документации/подсказках желательно указывать краткие инструкции по каждому действию.

5. Кросс-браузерная совместимость. Приложение ориентировано на веб, поэтому должно корректно работать в современных версиях основных браузеров (Chrome, Firefox, Safari, Edge). Интерфейс должен адекватно отображаться на различных размерах экрана, если предполагается использование на планшетах или небольших ноутбуках.

6. Локализация. Система должна поддерживать мультиязычность без необходимости глобальной перестройки; при переключении языка меняются все системные сообщения, уведомления и элементы меню. При этом пользовательские данные (названия задач, описания) не переводятся автоматически.

7. Расширяемость и поддерживаемость. Кодовая база должна быть организована таким образом, чтобы добавление новых функций (например, интеграция с другими календарями или аналитическими модулями) не требовало радикального переписывания приложения. Любые обновления не должны нарушать существующий функционал и логику задач.

8. Время отклика. При стандартном уровне загрузки (среднее количество пользователей онлайн) приложение должно отвечать на базовые действия (открытие списка задач, календаря, редактирование) в пределах нескольких секунд. В идеале действия, связанные с редактированием или сменой статусов, выполняются практически мгновенно для пользователя.

9. Стабильность при сбоях и неустойчивом соединении. В случае кратковременных проблем с сетью или серверами приложение при повторном подключении должно корректно восстанавливать сессию пользователя (если сессия не истекла) и не приводить к потере уже введённых данных (например, при сохранении формы).

10. Лёгкость развертывания и администрирования. Приложение должно устанавливаться и настраиваться без сложных зависимостей и скриптов; базовые параметры (например, список праздников, лимиты на вложения) меняются из административной панели. При необходимости обновления (патчи, новые версии) администратор выполняет минимальный набор действий, не влияя на основную логику приложения.

# Вывод

В результате выполнения практической работы студенты получили практический опыт системного подхода к сбору и обработке требований, научились задавать целенаправленные вопросы для уточнения потребностей и формировать корректные пользовательские и проектные спецификации. Отчет отражает важность итеративного взаимодействия и обмена обратной связью в процессе работы, демонстрируя, что качественная формализация требований является ключевым элементом успешной реализации проекта. Полученные навыки и умения являются важным шагом в профессиональном развитии и могут быть успешно применены в реальных условиях разработки программного обеспечения.