МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ

РОССИЙСКОЙ ФЕДЕРАЦИИ

федеральное государственное бюджетное образовательное учреждение

высшего образования

«Кемеровский государственный университет» (КемГУ)

**Институт Цифры**

**Кафедра цифровых технологий**

**ОТЧЕТ**

**О ВЫПОЛНЕНИИ ЛАБОРАТОРНОЙ РАБОТЫ №4(1)**

**ПО ПРОГРАММНОЙ ИНЖЕНЕРИИ**

Студента 3 курса, группы ПИ-202

Балахонцева Владимира Игоревича

Направление подготовки

*09.03.03 Прикладная информатика*

Руководитель:

ассистент

В.В. Илькевич

Кемерово 2023

**Цели работы:** ознакомиться с предложенным материалом и создать требования к сайту.

**Задачи:**

1. Документ Software Requirements Specification, содержащий список требований к сайту.
2. UseCase-диаграммы прецедентов использования, реализующих функциональные требования.
3. Выводы по работе.

**1.UseCase – диаграмма прецедентов.**

1.2 Operating environment

* 1.2.1 Сайт поддерживается браузерами Firefox, Opera, Сhrome, Яндекс и другие виды.
* 1.2.2 Front – реализован при помощи HTML, CSS, JS
* 1.2.3 Back – реализован на PHP

1.3 User documentation

Документация не требуется

Для создания диаграмм прецедентов был выбран сайт по чтению манги. (Remanga.org: <https://remanga.org/>). Рассмотрим диаграмму прецедентов на рисунке 1.

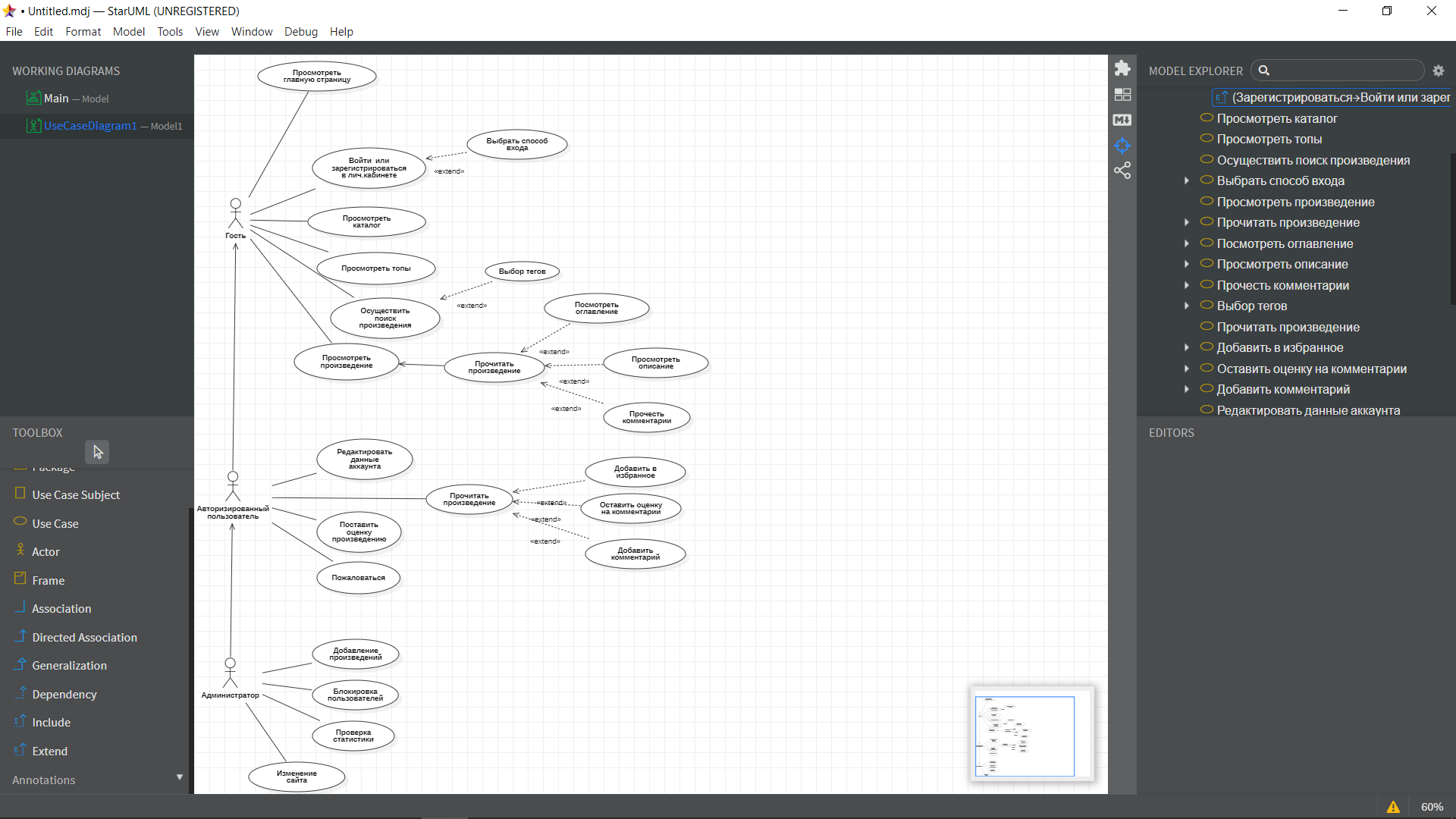


Рисунок 1 UseCase диаграмма сайта

**3. Specific requirements**

**3.1 Functional requirements**

**Требования владельца сайта:**

FR1. Система должна предоставлять возможность добавления новых произведений.

FR2. Система должна предоставлять возможность авторизации в роли администратора.

FR3. Система должна предоставлять функционал просмотра и учета статистики.

FR4. Система должна предоставлять возможности блокировки пользователей.

FR5. Система должна предоставлять возможность изменения сайта.

**Требования пользователей сайта:**

FR6. Система должна предоставлять функционал регистрации на сайте.

FR7. Система должна предоставлять функционал авторизации пользователя.

FR8. Система должна предоставлять функционал редактирования данных аккаунта пользователя.

SEC1. Система должна предоставлять возможность при утрате пароля его восстановить, а также возможность выбора входа.

FR9. Система должна предоставлять возможность поиска произведения по названию, автору, тегу.

FR10. Система должна предоставлять возможность просмотра Топов произведений по рейтингу сайта и перехода на последующий сайт.

FR11. Система должна предоставлять возможность просмотреть список недавно добавленных произведений.

FR12. Система должна предоставлять возможность прочитать выбранное произведение на русском языке.

FR13. Система должна предоставлять возможность просмотреть оглавление произведения и перейти на соответствующую главу.

FR14. Система должна предоставлять возможность регулировать тематику сайта.

FR15. Система должна предоставлять возможность отметить произведение как прочитанное.

FR16. Система должна предоставлять возможность запомнить страницу и добавить ее в закладки.

FR17. Система должна предоставлять возможность поставить оценку произведению.

FR18. Система должна предоставлять возможность перехода на страницу выбранного произведения.

FR19. Система должна предоставлять возможность поделиться ссылкой произведения на сайте.

FR20. Система должна предоставлять возможность просмотра последних отзывов и перехода на соответствующие произведение.

FR21. Система должна предоставлять возможность оставить отзыв о произведение.

* 1. **Non-functional requirements**

*NFUR – Non-functional usability requirements*

*NFRR – Non-functional reliability requirements*

*NFPR – Non-functional performance requirements*

*NFSR – Non-functional supportability requirements*

NFRR1. Сайт должен корректно работать в Firefox, Opera, Сhrome, Яндекс и других видах.

NFPR1. Сайт должен корректно работать при пиковых нагрузках (до 40 запросов/сек)

NFSR1. Сайт должен автоматически возобновлять свою работу при непредвиденном отключении в течение часа

NFUR1. Время отклика сайта должно не превышать 3 секунд

NFUR2. URL-адрес сайта – <https://remanga.org/>

NFUR3. Доступный просмотр списка выбранных произведений

NFUR4. В навигационной панели есть ссылки на все категории страниц сайта.

NFUR5. В навигационной панели есть разные режимы подсветки сайта и настройки

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| ID | Требование | Полезность | Трудоемкость, человеко-час | Стабильность |
| Functional requirements (владельц сайта): | | | | |
| FR1 | Система должна предоставлять возможность добавления новых произведений. | Важная | 5 | Средняя |
| FR2 | Система должна предоставлять возможность авторизации в роли администратора. | Полезная | 6 | Низкая |
| FR3 | Система должна предоставлять функционал просмотра и учета статистики. | Критическая | 7 | Высокая |
| FR4 | Система должна предоставлять возможности блокировки пользователей. | Важная | 4 | Высокая |
| FR5 | Система должна предоставлять возможность изменения сайта. | Критическая | 6 | Высокая |
| Functional requirements (пользователей сайта): | | | | |
| FR6 | Система должна предоставлять функционал регистрации на сайте. | Полезная | 2 | Низкая |
| FR7 | Система должна предоставлять функционал авторизации пользователя. | Важная | 4 | Средняя |
| FR8 | Система должна предоставлять функционал редактирования данных аккаунта пользователя. | Полезная | 2 | Средняя |
| FR9 | Система должна предоставлять возможность поиска произведения по названию, автору, тегу. | Критическая | 7 | Средняя |
| FR10 | Система должна предоставлять возможность просмотра Топов произведений по рейтингу сайта и перехода на последующий сайт. | Полезная | 3 | Средняя |
| FR11 | Система должна предоставлять возможность просмотреть список недавно добавленных произведений. | Полезная | 3 | Низкая |
| FR12 | Система должна предоставлять возможность прочитать выбранное произведение на русском языке. | Критическая | 6 | Высокая |
| FR13 | Система должна предоставлять возможность просмотреть оглавление произведения и перейти на соответствующую главу. | Важная | 4 | Низкая |
| FR14 | Система должна предоставлять возможность регулировать тематику сайта. | Критическая | 12 | Высокая |
| FR15 | Система должна предоставлять возможность отметить произведение как прочитанное. | Полезная | 2 | Низкая |
| FR16 | Система должна предоставлять возможность запомнить страницу и добавить ее в закладки. | Полезная | 4 | Средняя |
| FR17 | Система должна предоставлять возможность поставить оценку произведению. | Полезная | 2 | Средняя |
| FR18 | Система должна предоставлять возможность перехода на страницу выбранного произведения. | Полезная | 7 | Средняя |
| FR19 | Система должна предоставлять возможность поделиться ссылкой произведения на сайте. | Полезная | 3 | Средняя |
| FR20 | Система должна предоставлять возможность просмотра последних отзывов и перехода на соответствующие произведение. | Критическая | 7 | Низкая |
| FR21 | Система должна предоставлять возможность оставить отзыв о произведение. | Полезная | 3 | Высокая |
| Non-functional requirements: | | | | |  |  |  | Низкая |
| NFRR1 | Сайт должен корректно работать при пиковых нагрузках (до 40 запросов/сек) | Критическая | 20 | Низкая |
| NFPR1 | Сайт должен корректно работать в Firefox, Opera, Сhrome, Яндекс и других видах. | Критическая | 20 | Средняя |
| NFSR1 | Сайт должен автоматически возобновлять свою работу при непредвиденном отключении в течение часа | Критическая | 15 | Низкая |
| NFUR1 | Время отклика сайта должно не превышать 3 секунд | Важная | 20 | Средняя |
| NFUR2 | URL-адрес сайта – <https://remanga.org/> | Важная | 3 | Низкая |
| NFUR3 | Доступный просмотр списка выбранных произведений | Важная | 3 | Высокая |
| NFUR4 | В навигационной панели есть ссылки на все категории страниц сайта. | Важная | 2 | Средняя |
| NFUR5 | В навигационной панели есть разные режимы подсветки сайта и настройки | Важная | 2 | Средняя |

**РИСКИ**

|  |  |  |  |
| --- | --- | --- | --- |
| Риск | Вероятность | Последствия | Список требований, подверженных влиянию |
| Недостаток квалификации  сотрудников | Низкая | -Увеличение количества багов  -Увеличение времени реализации проекта:  -Увеличение затрат на исправление допущенных ошибок  -Необходимость тратить время на повышение квалификации сотрудников | FR 1-21, NFRR1, NFPR1, NFSR1, NFUR 1-5 |
| Недостаток  финансирования проекта | Средний | - Распад команды  -Недостаток оборудования  -Недостаток функционала  -Недостатки при создании и обслуживании сайта | NFRR1, NFPR1, NFSR1, NFUR1 |
| Недостаток времени на  реализацию проекта | Низкая | -Незаконченность проекта  - Возникновение ошибок  - Потеря потока клиентов | NFUR 2-4, |
| Смена внешних API | Средняя | - Трудны поиск и потеря прошлого места в списки поисков  - Потеря связей между различными модулями сайта | FR5, FR19 |
| Смена концепции предоставления услуг | Низкая | - Изменение  Всего функционала сайта | Все |
| Нехватка клиентов для тестирования | Низкая | -Увеличение времени реализации проекта  -Падение производительности  - Увеличение риска возникновение последующих ошибок | NFPR1, NFSR1, NFUR 1 |
| Потеря смысловой взаимосвязи интерфейса | Средняя | - Дезориентация пользователя вовремя использования сайта  - Увеличение срока выполнения поиска произведения  - Отток клиентов | FR3 |

* 1. **Use-cases (Прецеденты)**

|  |  |
| --- | --- |
| Имя прецедента | Просмотр главной страницы |
| ID | 1 |
| Краткое описание | Просмотр главной страницы сайта с различными представленными рекомендациями |
| Актёры, вовлеченные в прецедент | Пользователь, Гость |
| Основной поток | 1. Прецедент начинается, когда Пользователь выбирает значок главного сайта 2. Система переводит пользователя или гостя на главную страницу сайта |
| Постусловия | Пользователь или гость видит представленные произведения или предложения |
| Альтернативные потоки | – |

|  |  |
| --- | --- |
| Имя прецедента | Войти или зарегистрироваться в личный кабинет |
| ID | 2 |
| Краткое описание | Регистрация нового пользователя или вход уже заведенного пользователя в базу данных, по номеру телефона |
| Актёры, вовлеченные в прецедент | Пользователь |
| Основной поток | 1. Прецедент начинается, когда Пользователь выбирает опцию «Войти». 2. Система запрашивает у Пользователя номер его телефона. 3. Пользователь вводит в форму номер своего телефона. 4. Если данные Пользователя удовлетворяют, то    1. Пользователь заходит в систему 5. Иначе    1. Данные Пользователя проходят регистрацию, и он подтверждает свой номер телефона    2. Данные Пользователя попадают в базу данных |
| Постусловия | Пользователь получает доступ к своему личному кабинету и изменению различных настроек |
| Альтернативные потоки | Выбор способа входа |

|  |  |
| --- | --- |
| Имя прецедента | Альтернативный поток: выбор способа входа |
| ID | 2.1 |
| Краткое описание | Пользователь может выбирать способ входа по предложенным вариантам |
| Актёры, вовлеченные в прецедент | Пользователь |
| Основной поток | 1. Прецедент начинается, когда Пользователь выбирает опцию «Войти». 2. Система запрашивает способ входа в систему 3. Пользователь выбирает один из предложенных вариантов   3.1 Пользователь вбивает пароль и заходит  3.2 Пользователь вбивает неверные данные и остается на поле ввода  3.3 После получения доступа пользователь проходит в личный кабинет   1. Пользователь возвращается на главную страницу |
| Постусловия | Пользователь получает доступ к своему личному кабинету и изменению различных настроек |
| Альтернативные потоки | - |

|  |  |
| --- | --- |
| Имя прецедента | Просмотреть каталог |
| ID | 3 |
| Краткое описание | Выбор каталога на основании предпочтений пользователя |
| Актёры, вовлеченные в прецедент | Пользователь, гость |
| Основной поток | 1. Прецедент начинается, когда Пользователь выбирает вкладку каталог. 2. Пользователь может дальше просмотреть каталог или выбрать понравившейся произведение |
| Постусловия | Пользователь выбирает произведения находящиеся в каталоги |
| Альтернативные потоки | - |

|  |  |
| --- | --- |
| Имя прецедента | Посмотреть топы |
| ID | 4 |
| Краткое описание | Пользователь может просмотреть представленный топовые произведения с оценкой на сайте |
| Актёры, вовлеченные в прецедент | Пользователь, гость |
| Основной поток | 1. Прецедент начинается, когда Пользователь выбирает вкладку топов 2. Система показывает имеющиеся топовые произведения, из которых пользователь может выбрать понравившиеся |
| Постусловия | Пользователь получает возможность влиять на подборку произведений в топе. |
| Альтернативные потоки | - |

|  |  |
| --- | --- |
| Имя прецедента | Осуществить поиск произведения |
| ID | 5 |
| Краткое описание | Поиск произведения по названию, автору тегу и т.п. |
| Актёры, вовлеченные в прецедент | Пользователь, гость |
| Основной поток | 1. Прецедент начинается, когда Пользователь вводит необходимые параметры поиска. 2. Если выбранные им параметры есть в системе, то    1. Пользователь переходит на необходимое произведение 3. Иначе    1. Пользователь видит, что система не смогла найти необходимое произведение |
| Постусловия | Пользователь осуществляет поиск произведения по известным ему данным |
| Альтернативные потоки | Выбор тегов |

|  |  |
| --- | --- |
| Имя прецедента | Альтернативный поток: Выбор тегов |
| ID | 5.1 |
| Краткое описание | Выбор критериев поиска по определенным тегам. |
| Актёры, вовлеченные в прецедент | Пользователь, гость |
| Основной поток | 1. Прецедент начинается, когда Пользователь выбирает поиск произведения по тегам 2. Если выбранные им параметры есть в системе, то    1. Пользователь переходит на необходимое произведение 3. Иначе    1. Пользователь видит, что система не смогла найти необходимое произведение |
| Постусловия | Пользователь ищет необходимое произведение |
| Альтернативные потоки | - |

|  |  |
| --- | --- |
| Имя прецедента | Просмотреть произведение |
| ID | 6 |
| Краткое описание | Выбрать произведение и просмотреть основные сведения |
| Актёры, вовлеченные в прецедент | Пользователь, гость |
| Основной поток | 1. Прецедент начинается, когда Пользователь выбрал произведение 2. Далее пользователей нажимает на данное произведение и получает краткую информацию о нем. |
| Постусловия | - |
| Альтернативные потоки | - |

|  |  |
| --- | --- |
| Имя прецедента | Прочитать произведение |
| ID | 6.1 |
| Краткое описание | Пользователь выбирает произведение и может прочитать основные сведения о данном произведение. |
| Актёры, вовлеченные в прецедент | Пользователь, гость |
| Основной поток | 1. Прецедент начинается, когда Пользователь выбрал нужное произведение 2. Далее пользователь выбирает вкладку прочитать и выбирает главу, с которой хочет начать чтение. |
| Постусловия | Просмотр оглавлений произведений или работа с самим произведением перед его прочтением |
| Альтернативные потоки | Просмотр оглавления |

|  |  |
| --- | --- |
| Имя прецедента | Альтернативный поток: Просмотр оглавления |
| ID | 6.2 |
| Краткое описание | Пользователь может просматривать оглавление читаемого произведения |
| Актёры, вовлеченные в прецедент | Пользователь, гость |
| Основной поток | 1. Прецедент начинается, когда Пользователь читает интересующую его произведение 2. Система позволяет просматривать оглавление, не отрываясь от чтения и не переходя на главный экран. |
| Постусловия | Чтение и выбор необходимого пользователю произведения |
| Альтернативные потоки | Просмотреть описание |

|  |  |
| --- | --- |
| Имя прецедента | Альтернативный поток: Просмотреть описание |
| ID | 6.3 |
| Краткое описание | Пользователь может просматривать описание читаемого произведения |
| Актёры, вовлеченные в прецедент | Пользователь, гость |
| Основной поток | 1. Прецедент начинается, когда Пользователь читает интересующую его произведение 2. Система позволяет просматривать описание, не отрываясь от чтения и не переходя на главный экран. |
| Постусловия | Чтение и выбор необходимого пользователю произведения |
| Альтернативные потоки | Просмотреть комментарии |

|  |  |
| --- | --- |
| Имя прецедента | Альтернативный поток: Просмотреть комментарии |
| ID | 6.4 |
| Краткое описание | Пользователь может просматривать комментарии читаемого произведения |
| Актёры, вовлеченные в прецедент | Пользователь, гость |
| Основной поток | 1. Прецедент начинается, когда Пользователь читает интересующую его произведение 2. Система позволяет просматривать комментарии, не отрываясь от чтения и не переходя на главный экран. |
| Постусловия | Чтение и выбор необходимого пользователю произведения |
| Альтернативные потоки | - |

|  |  |
| --- | --- |
| Имя прецедента | Редактировать данные аккаунта |
| ID | 7 |
| Краткое описание | Редактирование профиля Пользователя |
| Актёры, вовлеченные в прецедент | Пользователь |
| Основной поток | 1. Прецедент начинается, когда Пользователь выбирает опцию «Редактирование профиля» в своем профиле. 2. Система запрашивает у Пользователя его имя, дата рождения, пол, E-mail. 3. Пользователь вводит в форму имя, дата рождения, пол, E-mail. 4. Если данные Пользователя удовлетворяют, то    1. Данные сохраняются 5. Иначе    1. Выдается ошибка о некорректности данных. |
| Постусловия | Пользователь получает различные возможности, предоставляемые сайтом |
| Альтернативные потоки | - |

|  |  |
| --- | --- |
| Имя прецедента | Поставить оценку произведению |
| ID | 8 |
| Краткое описание | Поставить оценку произведению на выбор |
| Актёры, вовлеченные в прецедент | Пользователь |
| Основной поток | 1. Прецедент начинается, когда Пользователь выбирает опцию поставить оценку произведению. 2. Система запрашивает выбор оценки и сохраняет выбор пользователя 3. Система предлагает оставить комментарий или вернуться на главный экран или к произведению |
| Постусловия | Пользователь получает различные возможности, предоставляемые сайтом |
| Альтернативные потоки | - |

|  |  |
| --- | --- |
| Имя прецедента | Прочитать произведение |
| ID | 9 |
| Краткое описание | Пользователь может прочитать произведение, выбранное ранее |
| Актёры, вовлеченные в прецедент | Пользователь |
| Основной поток | 1. Прецедент начинается, когда Пользователь выбирает опцию прочитать произведение. 2. Система переносит пользователя на первую страницу произведения или к оглавлению, в зависимости от выбора пользователя. |
| Постусловия | Пользователь получает различные возможности, предоставляемые сайтом |
| Альтернативные потоки | Добавить в избранное |

|  |  |
| --- | --- |
| Имя прецедента | Альтернативный поток: Добавить в избранное |
| ID | 9.1 |
| Краткое описание | Пользователь может добавить в избранное читаемое или интересуемое произведение |
| Актёры, вовлеченные в прецедент | Пользователь |
| Основной поток | 1. Прецедент начинается, когда Пользователь выбирает перейти к произведению. 2. Система на главном экране предоставляет информацию о произведение и дает возможность добавления в избранное |
| Постусловия | Пользователь получает различные возможности, предоставляемые сайтом |
| Альтернативные потоки | Оставить оценку на комментарии |

|  |  |
| --- | --- |
| Имя прецедента | Альтернативный поток: Оставить оценку на комментарии |
| ID | 9.2 |
| Краткое описание | Пользователь может оставлять оценку на комментарии к произведению или главе |
| Актёры, вовлеченные в прецедент | Пользователь |
| Основной поток | 1. Прецедент начинается, когда Пользователь выбирает перейти к вкладке комментарии 2. Система на главном экране предоставляет различные комментарии других пользователей и самого пользователя (если они имеются) 3. Пользователь получает возможность оценки комментариев |
| Постусловия | Пользователь получает различные возможности, предоставляемые сайтом |
| Альтернативные потоки | Добавить комментарий |

|  |  |
| --- | --- |
| Имя прецедента | Альтернативный поток: Добавить комментарий |
| ID | 9.3 |
| Краткое описание | Пользователь может оставлять комментарии к произведению или главе |
| Актёры, вовлеченные в прецедент | Пользователь |
| Основной поток | 1. Прецедент начинается, когда Пользователь выбирает перейти к вкладке комментарии 2. Система на главном экране предоставляет различные комментарии других пользователей и самого пользователя (если они имеются) 3. Пользователь может добавлять комментарии или редактировать старые |
| Постусловия | Пользователь получает различные возможности, предоставляемые сайтом |
| Альтернативные потоки | - |

|  |  |
| --- | --- |
| Имя прецедента | Пожаловаться |
| ID | 10 |
| Краткое описание | Пользователь может пожаловаться на выбранное произведение или комментарий |
| Актёры, вовлеченные в прецедент | Пользователь |
| Основной поток | 1. Прецедент начинается, когда Пользователь выбирает перейти к вкладке произведения 2. Система на главном экране предоставляет кнопку пожаловаться 3. Пользователь может выбрать определенный критерий или написать свою жалобу |
| Постусловия | Пользователь получает различные возможности, предоставляемые сайтом |
| Альтернативные потоки | - |

|  |  |
| --- | --- |
| Имя прецедента | Добавление произведений |
| ID | 11 |
| Краткое описание | Администратор сайта может добавлять различные произведения |
| Актёры, вовлеченные в прецедент | Пользователь, Администратор |
| Основной поток | 1. Прецедент начинается, когда Администратор собирается добавить новое произведение 2. Администратор добавляет новое произведение в определенный раздел сайта и привязывает различными ссылками |
| Постусловия | - |
| Альтернативные потоки | - |

|  |  |
| --- | --- |
| Имя прецедента | Блокировка пользователя |
| ID | 12 |
| Краткое описание | Администратор сайта может блокировать различных пользователей сайта |
| Актёры, вовлеченные в прецедент | Пользователь, Администратор |
| Основной поток | 1. Прецедент начинается, когда Администратор собирается заблокировать пользователя 2. Администратор блокирует пользователя и выставляет причину блокировки |
| Постусловия | - |
| Альтернативные потоки | - |

|  |  |
| --- | --- |
| Имя прецедента | Проверка статистики |
| ID | 13 |
| Краткое описание | Администратор сайта может проверять статистку интересующих его параметров сайта |
| Актёры, вовлеченные в прецедент | Администратор |
| Основной поток | 1. Прецедент начинается, когда Администратор собирается проверить статистику 2. Администратор проверяет статистику выставляя нужные критерии и получает основную или подробную информацию о выбранных параметрах |
| Постусловия | - |
| Альтернативные потоки | - |

|  |  |
| --- | --- |
| Имя прецедента | Изменение сайта |
| ID | 14 |
| Краткое описание | Администратор сайта может изменять сайт и дорабатывать изменения |
| Актёры, вовлеченные в прецедент | Администратор |
| Основной поток | 1. Прецедент начинается, когда Администратор собирается изменить сайт 2. Администратор подбирает основные изменения, а далее внедряет их в существующую базу страницы сайта |
| Постусловия | - |
| Альтернативные потоки | - |

**Выводы:** в ходе выполнения лабораторной работы мы ознакомились с методологией RUP и структурой SRS документа, была создана Use-Case диаграмма и составлен список требований к сайту, оформленный в виде документа SRS.

Вопросы:

1. Методологии разработки ПО. Унифицированный процесс.

**Основные модели разработки ПО:**

Code and fix — модель кодирования и устранения ошибок;

Waterfall Model — каскадная модель, или «водопад». В этой модели разработка осуществляется поэтапно: каждая следующая стадия начинается только после того, как заканчивается предыдущая. «Водопад» подходит для разработки проектов в *медицинской и космической отрасли, где уже сформирована обширная база документов*(СНиПов и спецификаций), на основе которых можно написать требования к новому ПО;

V-model — V-образная модель, разработка через тестирование. Это усовершенствованная каскадная модель, в которой заказчик с командой программистов одновременно составляют требования к системе и описывают, как будут тестировать её на каждом этапе. V-модель подходит для проектов, в которых важна надёжность и цена ошибки очень высока. Например, при разработке подушек безопасности для автомобилей или систем наблюдения за пациентами в клиниках.;

Incremental Model — инкрементная модель. Это модель разработки по частям. Инкрементная модель подходит для проектов, в которых точное техзадание прописано уже на старте, а продукт должен быстро выйти на рынок;

Iterative Model — итеративная (или итерационная) модель. Это модель, при которой заказчик не обязан понимать, какой продукт хочет получить в итоге, и может не прописывать сразу подробное техзадание. Итеративная модель подходит для работы над большими проектами с неопределёнными требованиями, либо для задач с инновационным подходом, когда заказчик не уверен в результате.;

Spiral Model — спиральная модель. Используя эту модель, заказчик и команда разработчиков серьёзно анализируют риски проекта и выполняют его итерациями. Последующая стадия основывается на предыдущей, а в конце каждого витка — цикла итераций — принимается решение, продолжать ли проект.;

Chaos model — модель хаоса;

Prototype Model — прототипная модель.

Из этих моделей наиболее популярны пять основных: каскадная, V-образная, инкрементная, итерационная и спиральная.

Rational Unified Process (RUP) — методология разработки программного обеспечения, созданная компанией Rational Software. RUP использует итеративную модель разработки. В конце каждой итерации (в идеале продолжающейся от 2 до 6 недель) проектная команда должна достичь запланированных на данную итерацию целей, создать или доработать проектные артефакты и получить промежуточную, но функциональную версию конечного продукта. Итеративная разработка позволяет быстро реагировать на меняющиеся требования, обнаруживать и устранять риски на ранних стадиях проекта, а также эффективно контролировать качество создаваемого продукта. Первые идеи итеративной модели разработки были заложены в «спиральной модели».

Полный жизненный цикл разработки продукта состоит из четырёх фаз, каждая из которых включает в себя одну или несколько итераций:

1. Начальная стадия (Inception)

В фазе начальной стадии:

* Формируются видение и границы проекта.
* Создается экономическое обоснование (business case).
* Определяются основные требования, ограничения и ключевая функциональность продукта.
* Создается базовая версия модели прецедентов.
* Оцениваются риски.

При завершении начальной фазы оценивается достижение этапа жизненного цикла цели (англ. Lifecycle Objective Milestone), которое предполагает соглашение заинтересованных сторон о продолжении проекта.

2. Уточнение (Elaboration)

В фазе «Уточнение» производится анализ предметной области и построение исполняемой архитектуры. Это включает в себя:

* Документирование требований (включая детальное описание для большинства прецедентов).
* Спроектированную, реализованную и оттестированную исполняемую архитектуру.
* Обновленное экономическое обоснование и более точные оценки сроков и стоимости.
* Сниженные основные риски.

Успешное выполнение фазы уточнения означает достижение этапа жизненного цикла архитектуры (англ. Lifecycle Architecture Milestone).

3. Построение (Construction)

В фазе «Построение» происходит реализация большей части функциональности продукта. Фаза Построение завершается первым внешним релизом системы и вехой начальной функциональной готовности (Initial Operational Capability).

4. Внедрение (Transition)

В фазе «Внедрение» создается финальная версия продукта и передается от разработчика к заказчику. Это включает в себя программу бета-тестирования, обучение пользователей, а также определение качества продукта. В случае, если качество не соответствует ожиданиям пользователей или критериям, установленным в фазе Начало, фаза Внедрение повторяется снова. Выполнение всех целей означает достижение вехи готового продукта (Product Release) и завершение полного цикла разработки.

1. Требования и их категоризация. Атрибуты требований.

Требование — потребность или ожидание, которое установлено, обычно предполагается или является обязательным.

Требования к ПО могут быть:

1. Функциональными и нефункциональными.

* Функциональные требования описывают функции, которые должно выполнять ПО. Например, предоставлять канал коммуникации для пользователя или переводить данные из одного формата в другой. То есть, речь идет о функционале продукта.
* Нефункциональные требования касаются таких вещей, как доступность, надежность, способность к восстановлению, поддерживаемость, масштабируемость, производительность, безопасность и прочие «…ость».

2. Производными и навязанными.

* Вытекает ли это требование из других требований?
* Это требование было явно и недвусмысленно высказано стейкхолдерами?

3. Ориентированными на продукт или на процесс его разработки.

* «Программа должна проверять права пользователя» — это требование к продукту.
* «Программа должна разрабатываться инкрементально. В ходе разработки должна использоваться непрерывная интеграция» — это требование к процессу.

4. Разной приоритетности. Для назначения приоритета может использоваться шкала с фиксированными значениями «обязательно», «крайне желательно», «желательно» и «необязательно».

5. Разного масштаба. Одни требования касаются проектирования отдельных компонентов, другие — архитектуры всего ПО. Нефункциональные требования чаще всего касаются всей программы в целом.

6. Изменчивыми или стабильными. Например, изначально может быть ясно, что требования будут меняться в ходе жизненного цикла продукта. В таком случае реализация программы должна быть толерантной к внесению изменений.

Атрибуты требований – требование в виде объекта, который обладает свойствами, отличающими его от других требований. В дополнение к тестовому описанию, каждое функциональное требование должно иметь несколько поддерживающих его фрагментов информации, или атрибутов (attributes), связанных с ним. Эти атрибуты представляют контекст и основу каждого требования, они располагаются за описанием предполагаемой функциональности. Вы можете сохранить атрибуты в таблице, базе данных или, что наиболее эффективно, в средстве управления требованиями. Сложно управлять более десятком атрибутов, когда они хранятся только в документах. Средства управления требованиями предоставляют несколько сгенерированных системой атрибутов, а также предоставляют возможность определить другие атрибуты, часть из которых может создаваться автоматически. Эти средства позволяют фильтровать, сортировать выбранные подмножества требований на основании значений их атрибутов и запрашивать базу данных для их просмотра. Например, можно вызвать список всех высокоприоритетных требований, которые Шери должна реализовать в версии 2.3 и которые имеют статус Approved (Одобрено). Далее приведены возможные атрибуты требований:

• дата создания требования;

• номер текущей версии требования;

• автор, создавший требование;

• приоритет;

• состояние требования;

• происхождение или источник требования;

• логическое обоснование требования;

• номер выпуска или итерации, на которую назначено требование;

• контактное лицо или ответственный за принятие решений по внесению изменений в требование;

• метод проверки или критерий приемки.

1. Язык UML.

UML— язык графического описания для объектного моделирования в области разработки программного обеспечения, для моделирования бизнес-процессов, системного проектирования и отображения организационных структур.

UML является языком широкого профиля, это — открытый стандарт, использующий графические обозначения для создания абстрактной модели системы, называемой UML-моделью. UML был создан для определения, визуализации, проектирования и документирования, в основном, программных систем. UML не является языком программирования, но на основании UML-моделей возможна генерация кода.

1. Прецеденты использования. UseCase-диаграммы - состав, виды связей.

Диаграмма прецедентов или диаграмма вариантов использования (англ. use case diagram) в UML — диаграмма, отражающая отношения между акторами и прецедентами и являющаяся составной частью модели прецедентов, позволяющей описать систему на концептуальном уровне.

Прецедент — возможность моделируемой системы (часть её функциональности), благодаря которой пользователь может получить конкретный, измеримый и нужный ему результат. Прецедент соответствует отдельному сервису системы, определяет один из вариантов её использования и описывает типичный способ взаимодействия пользователя с системой. Варианты использования обычно применяются для спецификации внешних требований к системе.

Основное назначение диаграммы — описание функциональности и поведения, позволяющее заказчику, конечному пользователю и разработчику совместно обсуждать проектируемую или существующую систему.

При моделировании прецедентов системный аналитик стремится:

* чётко отделить систему от её окружения;
* определить язык или система программирования) при составлении модели прецедентов опускаются (для них составляется другой документ).

Актёр обозначает любые сущности, использующие систему. Этими сущностями могут быть люди, технические устройства или даже другие системы.

Вариант использования (Эллипс) – набор действий, который может быть использован актёром для взаимодействия с системой.

Отношение ассоциации – отображают на диаграмме информацию о том, какие варианты использования могут быть использованы каждым актёром.

Отношение обобщения означает, что некоторый актёр (вариант использования) может быть обобщён до другого актёра (варианта использования).

Когда мы используем отношение включения, мы подразумеваем, что составные варианты использования обязательно входят в состав общего варианта использования.

Отношение расширения — это выборочное отношение включения. Если отношение включения обозначает, что элемент обязательно включается в состав другого элемента, то в случае отношения расширения это включение необязательно.