

Software Requirement Specification
Калькулятор посчёта бухгалтерских услуг
L^AT_EX

Содержание

1	Introduction	3
1.1	Purpose	3
1.2	References	3
1.3	Scope	3
2	Model Requirements FURPS+ with some attributes	3
2.1	Functional requirements	3
2.2	Non functional requirements	3
2.3	Статусы требований	4
3	UML Use-Case	5
4	Методология разработки	6
4.1	Waterfall	6

1 Introduction

1.1 Purpose

Основная цель калькулятора бухгалтерских услуг - дать бизнесу и частным лицам возможность эффективно оценивать и планировать свои бухгалтерские расходы. Это проиллюстрирует цель и полное понимания построения модели. Документ также объяснит системные ограничения и интерфейс. Этот документ предназначен для в первую очередь для заказчиков проекта, которым он будет важен для составления бизнес процессов, а также для разработчиков, так как тут описаны основные цели, которые согласованы обоими сторонами.

1.2 References

<https://www.rfc-editor.org/info/rfc791>

<https://www.rfc-editor.org/info/rfc793>

<https://www.w3.org/TR/html5/>

<https://www.w3.org/TR/css3-roadmap/>

<https://www.ecma-international.org/publications-and-standards/standards/ecma-262/>

<https://www.w3.org/TR/WCAG21/>

<https://www.iso.org/standard/54534.html>

1.3 Scope

Документ охватывает функциональные и нефункциональные требования к веб-сайту, ориентирован на команду разработчиков, проектных менеджеров, и заказчика проекта. Описывает основные возможности сайта.

2 Model Requirements FURPS+ with some attributes

2.1 Functional requirements

- Should have - SEC0 - Система должна предоставлять возможность авторизации пользователей с помощью имени пользователя и пароля.
- Should have - FR0 - Система должна поддерживать редактирование констант за услуги авторизованными пользователями.
- Must have - FR1 - Система должна поддерживать возможность просмотра подробного описания услуг.
- Must have - FR2 - Система должна поддерживать возможность добавления услуг в корзину.
- Must have - FR3 - Система должна поддерживать возможность удаления услуг из корзины.
- Must have - FR4 - Система должна поддерживать возможность просмотра количества добавленных услуг.
- Must have - FR5 - Система должна показывать итоговый расчёт по выбранным услугам.
- Could have - FR6 - Система должна предоставлять возможность заморозки услуг авторизованными пользователями.
- Must have - FR7 - Система должна поддерживать возможность просмотра услуг.

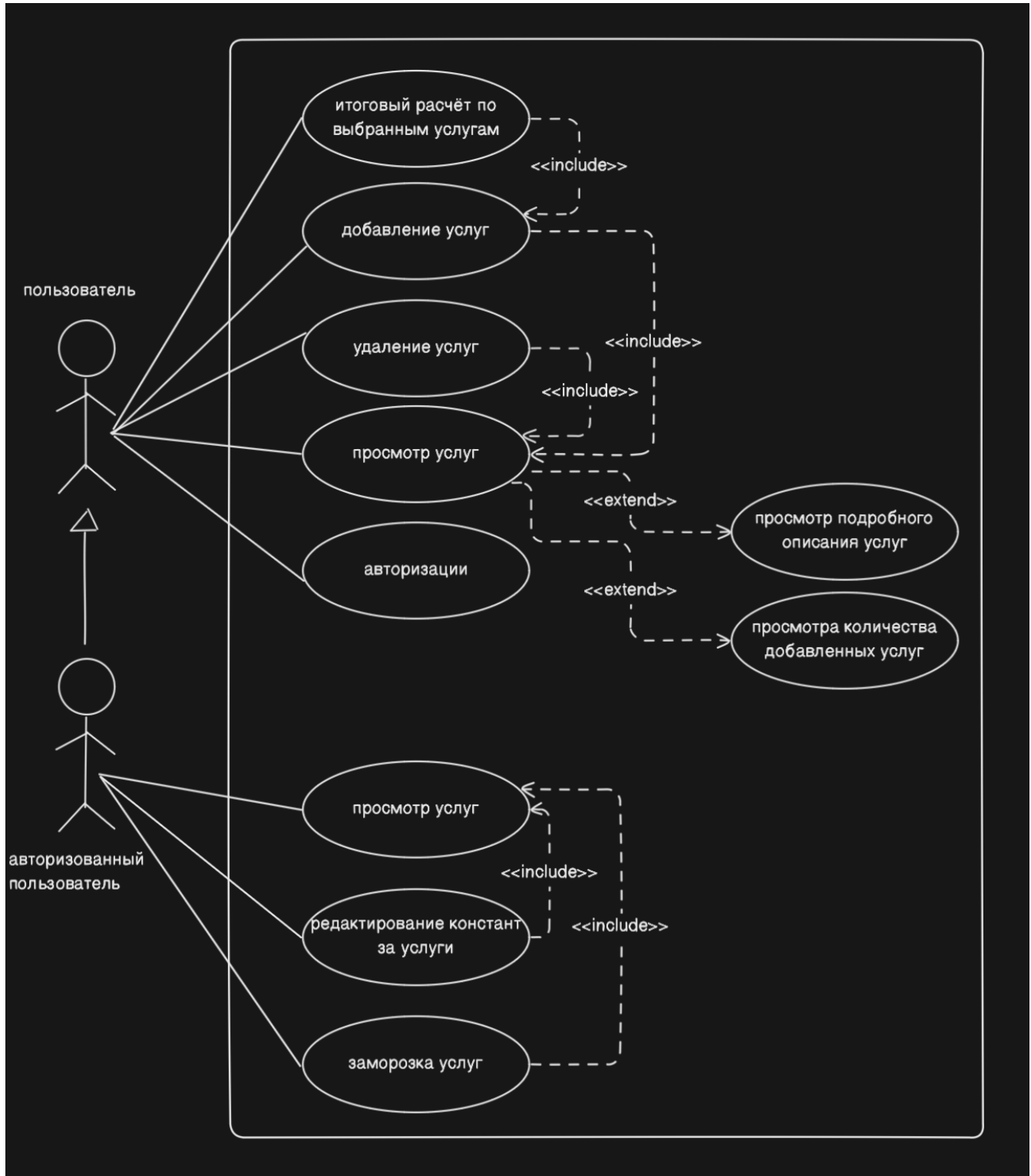
2.2 Non functional requirements

- Must have USA0 - Система должна обеспечивать адаптивный дизайн для различных устройств.
- Must have USA1 - Система должна обеспечивать ответ пользователю в промежутке от 200 ms до 1000 ms.
- Must have PERF0 - Система должна обрабатывать тысячи запросов одновременно без существенных задержек.
- Must have SUPP0 - Система должна легко масштабироваться для поддержки увеличения числа пользователей.

2.3 Статусы требований

- SEC0 - предложена
- FR0 - предложена
- FR1 - предложена
- FR2 - предложена
- FR3 - предложена
- FR4 - предложена
- FR5 - предложена
- FR6 - предложена
- FR7 - предложена
- USA0 - предложена
- USA1 - предложена
- PERF0 - предложена
- SUPP0 - предложена

3 UML Use-Case



4 Методология разработки

4.1 Waterfall

Стандартные шаги разработки:

- Определение системных требований
- Определение требований в ПО
- Анализ требований
- Проектирование программы
- Разработка кода
- Тестирование
- Введение ПО в эксплуатацию

Есть понятие итерации между фазами разработки, есть возможность отката к предыдущей фазе. На фазе тестирования можно обнаружить, что итоговые характеристики отличаются от заданных изначально, поэтому нужно менять либо требования, либо дизайн системы.