

Оглавление

1	Аналитический раздел	4
1.1	Введение	4
1.2	Возможные прецеденты	4
1.3	Осуществляемая деятельность	6
1.4	Вывод	8
2	Конструкторский раздел	9
2.1	Введение	9
2.2	Общая структура системы	9
2.3	Система мониторинга	9
2.4	Фронтэнд пользователей	10
2.5	Фронтэнд вычислительных узлов	10
2.6	Система управления сессией	10
2.7	Система управления	10
2.8	Система хранения данных	10
2.9	Система балансировки нагрузки	10
2.10	Система вычисления	10
2.11	Вывод	10
3	Технологический раздел	11
3.1	Введение	11
3.2	Выбор языка программирования	11
3.3	Выбор программных средств	11
3.4	?необязательно? программная реализация отдельных ком- понентов	11
3.5	Тестирование	11
3.6	Вывод	11
4	Заключение	12

Глоссарий

- КОМПЛЕКС
- сус
- сбн
- су
-
- задача
- ПОЛЬЗОВАТЕЛЬ
- ВЫЧИСЛИТЕЛЬНЫЙ узел
-
- $x?$
-
-

Введение

1. Аналитический раздел

1.1. Введение

В данном разделе выполняется анализ предметной области. Результаты анализа представляются в виде диаграм прецедентов и деятельности.

1.2. Возможные прецеденты

Комплекс при его работе предоставляет пользователю следующие варианты использования:

- регистрация пользователя;
- авторизация пользователя;
- постановка задачи на исполнение;
- просмотр статуса задачи;
- отмена задачи.

Диаграмма этих и дополнительных служебных прецедентов приведена на рис. 1.

С учётом требований к разделению внутреннего функционала комплекса, диаграмма прецедентов на рис. 1 расщепляется на набор диаграмм, соответствующих каждой из выделенных подсистем. Соответствующие диаграммы приведены на рисунках 2,3,4.

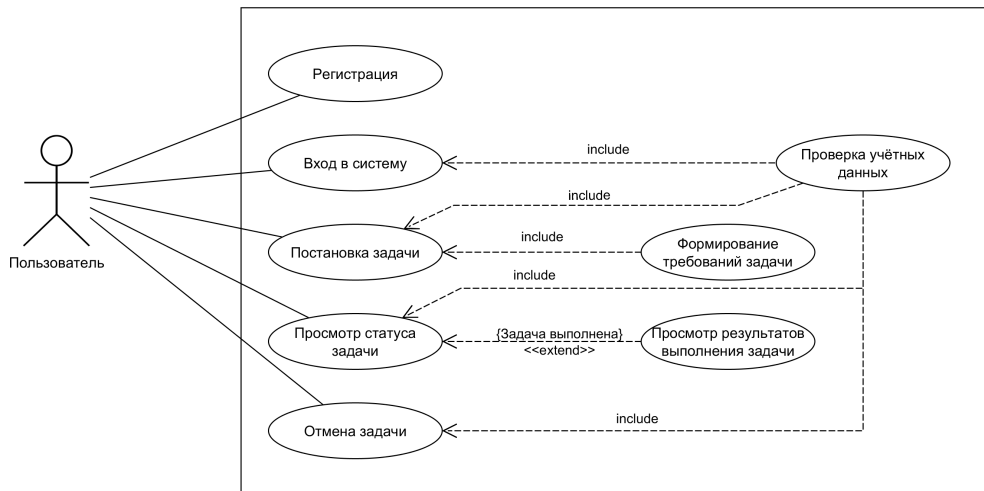


Рис. 1: Диаграмма прецедентов всеего комплекса в целом

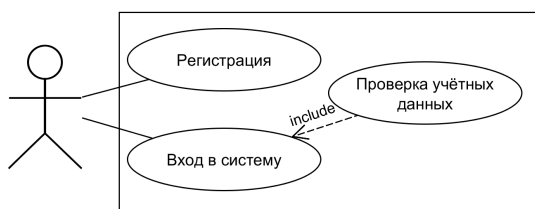


Рис. 2: Диаграмма прецедентов СУС

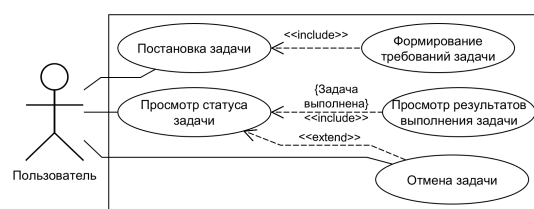


Рис. 3: Диаграмма прецедентов СУ

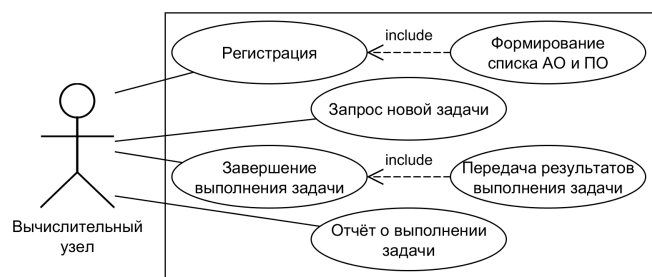


Рис. 4: Диаграмма прецедентов СБН

1.3. Осуществляемая деятельность

Прецеденты, описанные в предыдущем пункте, отвечают определённой деятельности. Диаграмма деятельности на рис. 5 описывает полный процесс взаимодействия пользователя с комплексом.

С учётом требований к разделению внутреннего функционала комплекса, диаграмма деятельности на рис. 5 расщепляется на набор диаграмм, соответствующих определённым подсистемам из выделенных.

Диаграммы действий прецедентов подсистемы управления сессией “регистрация” и “вход в систему” приведены на рисунках 6 и 7 соответственно.

Диаграммы действий прецедентов системы балансировки нагрузки “регистрация”, “запрос новой задачи” и “завершение выполнения задачи” приведены на рисунках 8, 9 и 10 соответственно.

Диаграммы действий прецедентов системы управления “постановка задачи” и “просмотр статуса задачи” приведены на рисунках 11 и 12 соответственно.

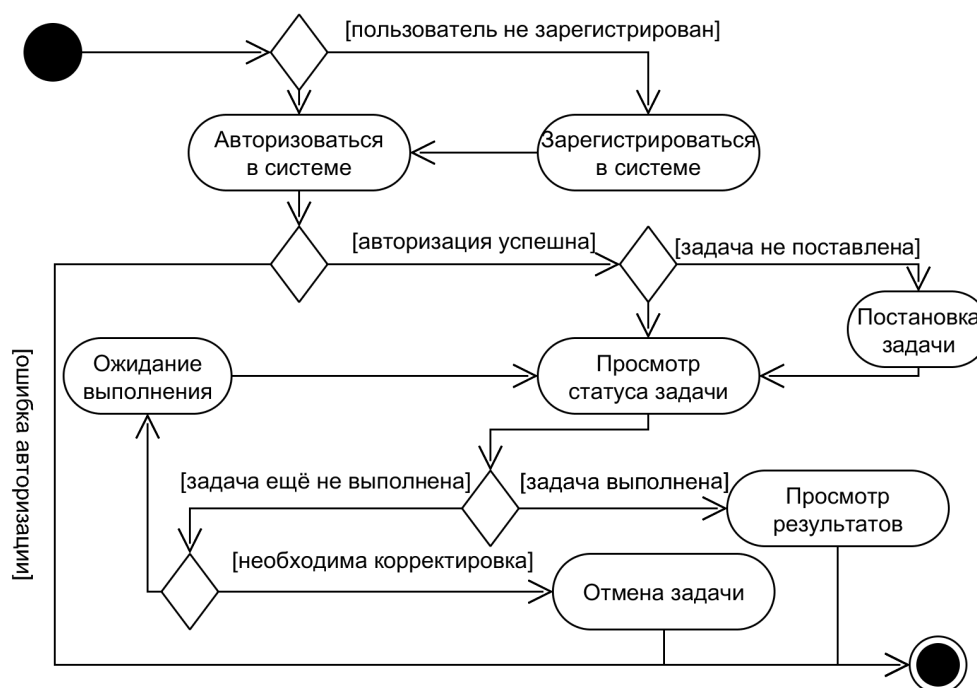


Рис. 5: Диаграмма действий прецедента “общая деятельность” для системы в целом

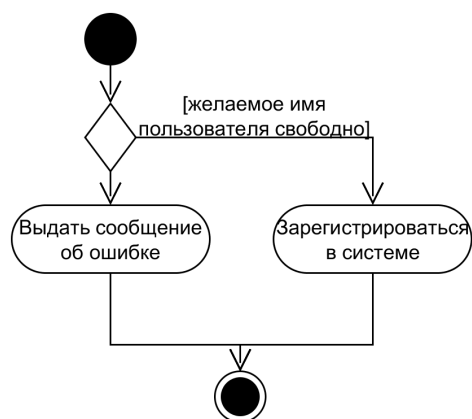


Рис. 6: Диаграмма действий прецедента “регистрация” СУС

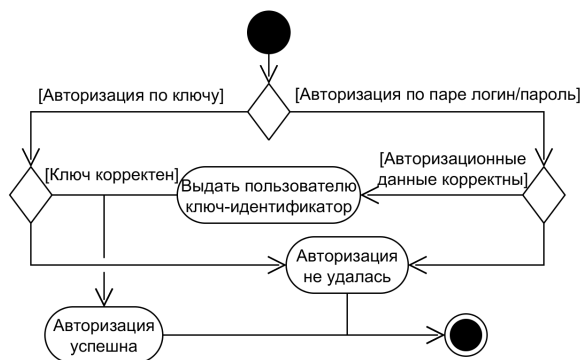


Рис. 7: Диаграмма действий прецедента “вход в систему” СУС

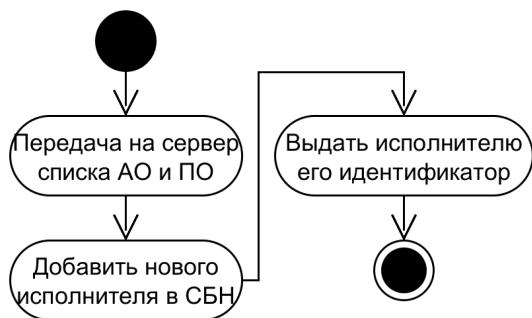


Рис. 8: Диаграмма действий прецедента “регистрация” СБН

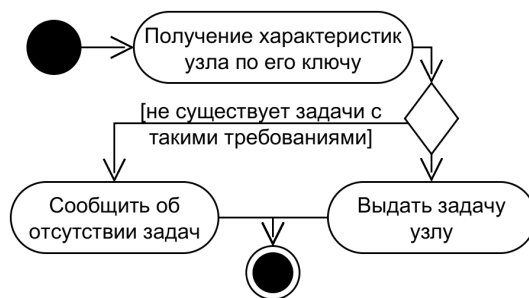


Рис. 9: Диаграмма действий прецедента “запрос новой задачи” СБН

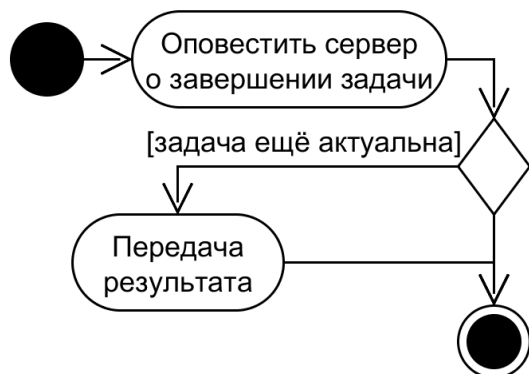


Рис. 10: Диаграмма действий прецедента “завершение выполнения задачи” СБН

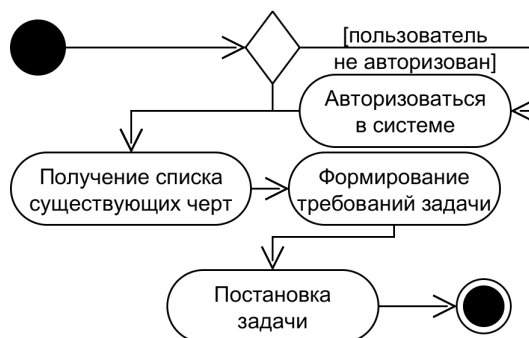


Рис. 11: Диаграмма действий прецедента “постановка задачи” СУ

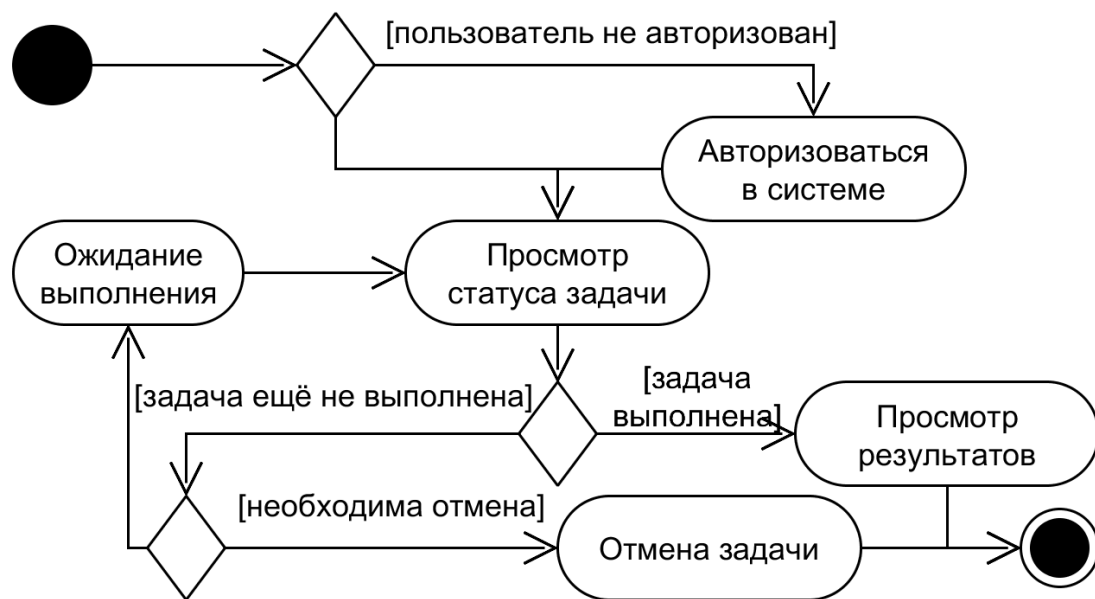


Рис. 12: Диаграмма действий прецедента “просмотр статуса задачи” СУ

1.4. Вывод

В данном разделе были приведены диаграммы, описывающие функционал основных узлов системы. Данный анализ в дальнейшем используется для более строгой формализации функционала подсистем.

2. Конструкторский раздел

2.1. Введение

В данном разделе приводятся результаты проектирования системы. С применением UML-диаграмм описывается общая структура комплекса и требуемый функционал отдельных узлов системы.

2.2. Общая структура системы

Для того, чтобы удовлетворить требованиям по предоставлению механизма деградации функциональности, а также для упрощения процесса разработки, комплекс должна быть разделена на отдельные слабосвязанные элементы.

2.3. Система мониторинга

Задача данной подсистемы – отслеживание топологии сети. Все узлы комплекса должны оповещать СМ о своём статусе работы, и любой узел может получить от комплекса список активных в данный момент узлов.

Невозможность любой другой подсистемы связаться с системой мониторинга рассматривается как ошибка сети, нарушающая нормальное функционирование комплекса.

Исходя из требований к СМ и с учётом REST-методик, она должна предоставлять следующее API:

- Ресурс: `/services`
Метод: GET
Возвращаемое значение: список активных сервисов
- Ресурс: `/services/type`
Метод: GET
Возвращаемое значение: список активных сервисов такого типа
- Ресурс: `/services/type`
Метод: POST
Параметры: `port`, `state`?
Возвращаемое значение: сообщение об успешной регистрации сервиса и распознанный адрес сервиса
Ошибки: отсутствует параметр `'port'`: HTTP 422

- Ресурс: `/services/type/address`

Метод: GET

Возвращаемое значение: статусное сообщение выбранного сервиса, аннотированное временем создания

Ошибки: сервис не найден: HTTP 404

- Ресурс: `/services/type/address`

Метод: PUT

Параметры: `state?`

Возвращаемое значение: сообщение об успешном обновлении статусного сообщения

2.4. Фронтэнд пользователей

— проверка безопасности + редирект на логику, возможно логика=апи, а здесь будет ещё и вебморда

2.5. Фронтэнд вычислительных узлов

Задача данной подсистемы – перенаправление запросов от вычислительных узлов на балансировщик нагрузки.

Исходя из требований к СМ и с учётом REST-методик, она должна предоставлять следующее API:

- Ресурс: `/nodes`

Метод: POST

Параметры: список черт вычислительного узла

Возвращаемое значение: сообщение об успешной регистрации узла и назначенный идентификатор

- Ресурс: `/nodes/identifier`

Метод: PUT

Параметры: состояние расчёта

Возвращаемое значение: сообщение об успешном обновлении статуса

Ошибки: узел разрегистрован за неактивностью: HTTP 404

- Ресурс: `/tasks/newtask`

Метод: GET

Параметры: идентификатор вычислительного узла

Возвращаемое значение: пакет данных, описывающих задачу

Ошибки: подходящих задач нет: HTTP 404

- **Ресурс:** /tasks/identifier

Метод: PUT

Параметры: результат выполнения задачи

Возвращаемое значение: сообщение об успешном приёме результата

2.6. Система управления сессией

Задача данной подсистемы – регистрация, авторизация и аутентификация пользователей в сети.

Исходя из требований к СМ и с учётом REST-методик, она должна предоставлять следующее API:

- **Ресурс:** /users

Метод: POST

Параметры: желаемая пара логин / пароль (возможно, хешированный)

Возвращаемое значение: сообщение об успешной регистрации пользователя

Ошибки: пользователь с таким именем уже зарегистрирован: HTTP 403

- **Ресурс:** /users/username

Метод: GET

Параметры: пароль (возможно, хешированный)

Возвращаемое значение: сгенерированный ключ доступа

Ошибки: некорректная пара логин / пароль: HTTP 403

- **Ресурс:** /validate

Метод: GET

Параметры: ключ доступа

Возвращаемое значение: сообщение об успешной проверке ключа

Ошибки: некорректный ключ: HTTP 401

2.7. Система управления

— интерфейсная часть, для людей. апи + вебморда; возможно вебморда на фронтенде пользователей.

2.8. Система хранения данных

— сюда пойдёт как минимум ER, так же будет описание апишки. реализация и д.классов и иже с ними – в технологическом

2.9. Система балансировки нагрузки

— собственно отвечает за координацию задач. имеет всё апи фронтенда вычислительных узлов (который просто редиректит запросы к ней), плюс некоторое апи по которому её опрашивают другие узлы комплекса.

2.10. Система вычисления

Данная система является активной, и своего API не имеет. Представлена данная система набором вычислительных узлов с установленным на них специальным ПО, осуществляющем взаимодействие с остальными сервисами системы и

2.11. Вывод

3. Технологический раздел

3.1. Введение

В данном разделе производится выбор языка программирования и сопутствующих программных средств. Описываются основные моменты программной реализации и описывается методика тестирования.

3.2. Выбор языка программирования

3.3. Выбор программных средств

3.4. ?необязательно? программная реализация отдельных компонентов

3.5. Тестирование

3.6. Вывод

4. Заключение