**EXERCISE – 1 Actors, Individuals, and the Job Search Domain**

Акторы, связанные с системой поиска работы:

* Соискатель (JobSeeker)
* Работодатель (Employer)
* Администратор системы (Platform Admin)

Для каждой из этих ролей рассмотрим следующие взаимодействия с системой поиска работы:

* Соискатель работы ищет вакансии.
* Соискатель работы подает заявление на работу.
* Работодатель размещает вакансии.
* Работодатель просматривает заявления на работу.
* Администратор платформы управляет учетными записями.
* Администратор платформы контролирует безопасность системы.

Потенциальные конфликты и их отражение в требованиях к ПО:

1. Пол как работодатель и соискатель:

* **Конфликт:** Пол может получить доступ к конфиденциальной информации о других соискателях или неправомерно повлиять на процесс рассмотрения заявлений.
* **Решение**: Ввести разделение ролей в системе с четкими правилами доступа и действий, доступных для каждой роли. Также необходим механизм аудита действий пользователей для обеспечения прозрачности процессов.

1. Работодатель использует систему для поиска информации о конкурентах:

* **Конфликт**: Работодатель может использовать JSE не только для поиска кандидатов на вакансии, но и для сбора информации о вакансиях конкурентов, что может привести к неэтичному использованию платформы.
* **Решение:** В требованиях к системе необходимо установить политику использования, которая запрещает использование информации о вакансиях для не предназначенных целей. Можно также ограничить видимость некоторой информации о вакансиях для работодателей, не давая доступ к деталям вакансий конкурентов.

1. Соискатель использует систему для массовой рассылки нецелевых заявлений:

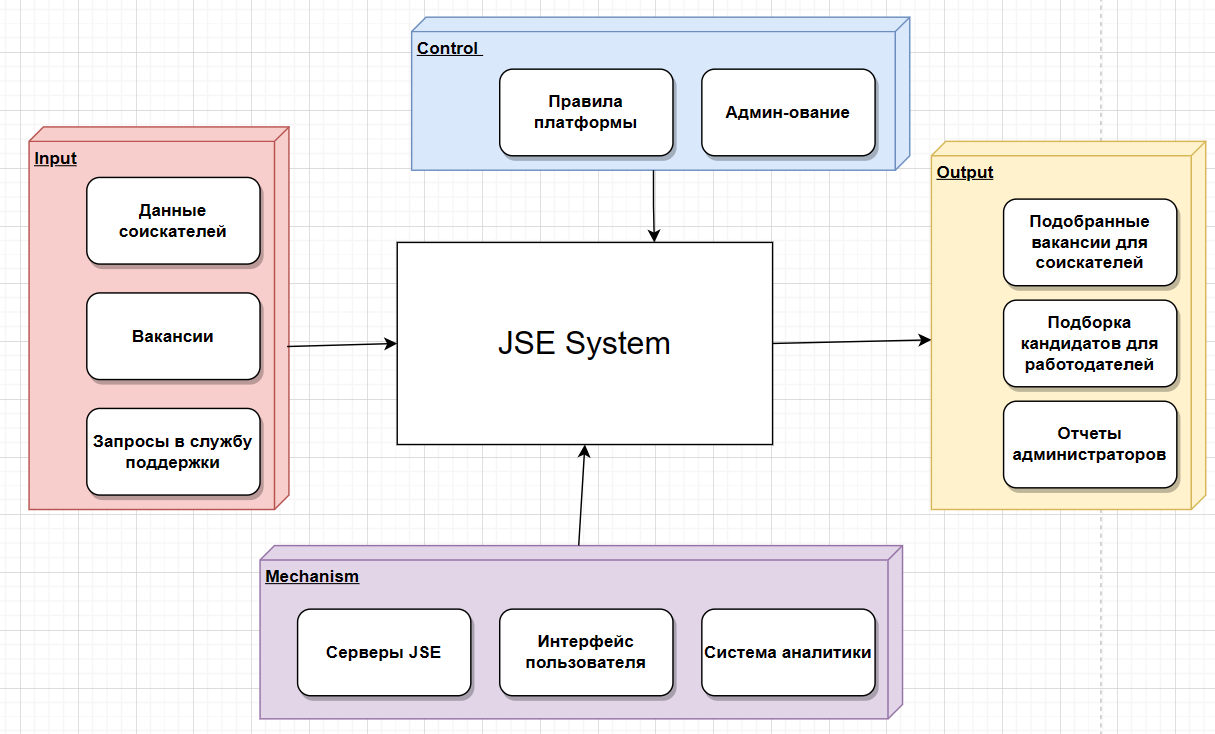
* **Конфликт:** Соискатель может использовать JSE для массовой отправки заявлений на вакансии, не соответствуя критериям этих вакансий, что приводит к перегрузке работодателей неактуальными кандидатурами.
* **Решение:** Разработать механизм фильтрации и верификации заявлений перед их отправкой работодателю, чтобы убедиться, что соискатели соответствуют ключевым требованиям вакансии. Ввести систему предупреждений для соискателей, злоупотребляющих функционалом массовой рассылки, и, в крайних случаях, ограничить им возможность подачи заявлений.

1. Соискатель получает предложения от нескольких работодателей:

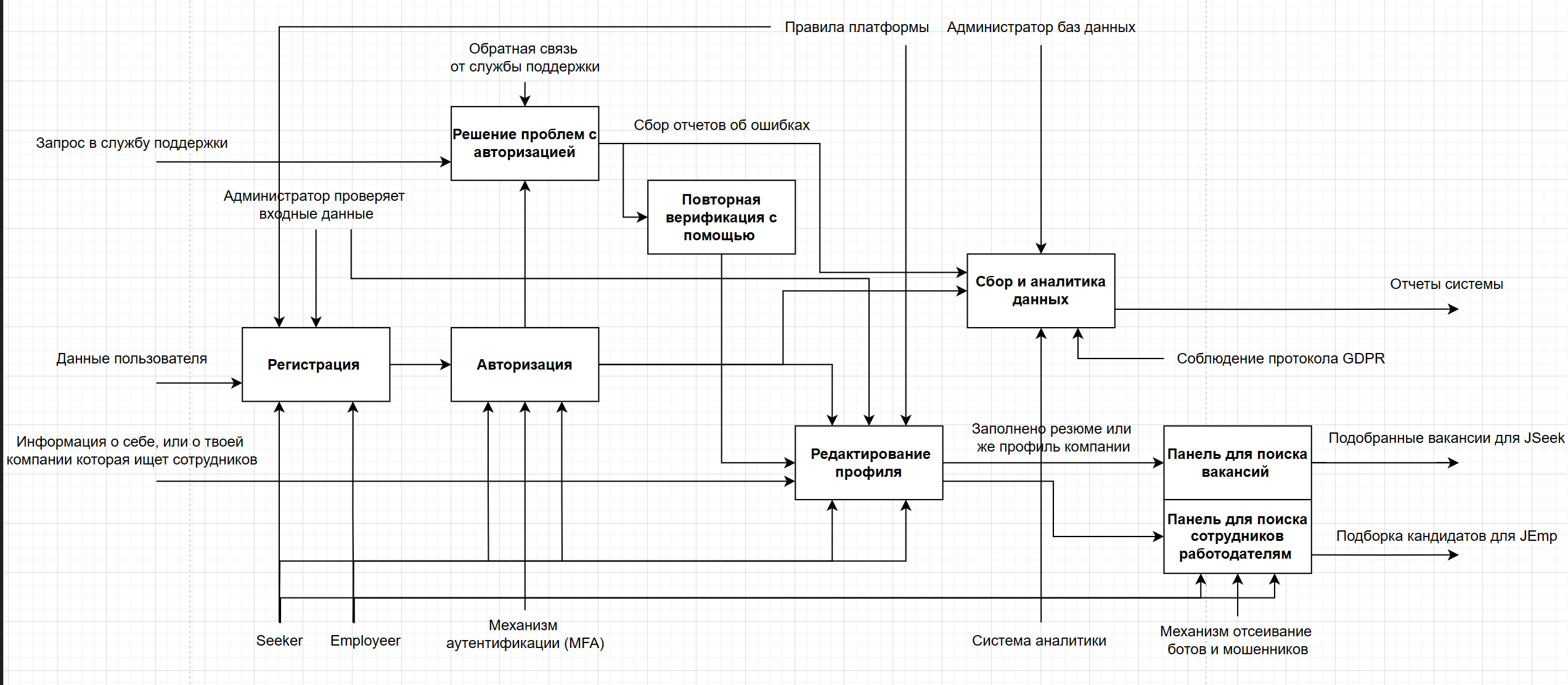
* **Конфликт:** Соискатель может чувствовать себя перегруженным при получении множества предложений от работодателей, особенно если эти предложения не полностью соответствуют его интересам или квалификации.
* **Решение:** Разработать систему рекомендаций, которая помогает соискателям фильтровать предложения на основе их предпочтений, квалификации и карьерных целей. Предоставить инструменты для управления предложениями, позволяющие легко отклонять неинтересующие вакансии.

**EXERCISE – 2 Context Diagram**

In short



Decomposition (подробнее о процессах)



**EXERCISE – 3 Major JSE decisions**

1. Пересмотрите свои высокоприоритетные риски

* Недостаточная защита данных пользователей.
* Недостаточная масштабируемость системы при увеличении числа пользователей.
* Сложность интеграции с внешними сервисами для проверки резюме и профилей.

1. Архитектурные решения
2. Решение: Внедрение слоев безопасности и шифрования.

* **Риск:** Несанкционированный доступ к данным пользователей и их утечка.
* **Отражение в архитектуре:** Реализация многоуровневой архитектуры с защищенными каналами связи, включая использование HTTPS и шифрование баз данных.
* **Отклоненные альтернативы:** Использование только базового парольного доступа без двухфакторной аутентификации.
* **Причина отклонения:** Недостаточное уровень безопасности и уязвимость для атак.

1. Решение: Применение микро сервисной архитектуры.

* **Риск:** Проблемы с масштабируемостью при увеличение числа пользователей.
* **Отражение в архитектуре:** Разработка системы с использованием набора микросервисов, каждый из которых отвечает за определенную функцию, обеспечивая тем самым легкость масштабирования и обслуживания.
* **Отклоненные альтернативы:** Монолитная архитектура.
* **Причина отклонения:** Монолитные системы сложнее масштабировать и обновлять, особенно при быстром росте пользовательской базы.

1. Решение: API-центрическая интеграция с внешними сервисами.

* **Риск:** Ограниченная функциональность и проблемы интеграции с внешними платформами.
* **Отражение в архитектуре:** Создание мощного API для интеграции с внешними сервисами, что позволяет легко добавлять новые сервисы и функции.
* **Отклоненные альтернативы:** Разовые, жестко закодированные интеграции.
* **Причина отклонения:** Негибкий подход, который ограничивает будущее расширение и обновление системы.

1. Требования качества:

* **Безопасность:** Реализация комплексных мер безопасности, включая регулярные аудиты безопасности и обновления системы.
* **Надежность:** Внедрение отказоустойчивых механизмов и резервного копирования данных.
* **Производительность:** Оптимизация процессов и инфраструктуры для обеспечения высокой производительности при большом числе запросов.

1. Документирование решений:
2. **Решение:** Внедрение механизмов обеспечения конфиденциальности и безопасности персональных данных.
3. **Адресованные качественные аспекты:** Защита персональных данных пользователей, соответствие требованиям GDPR и обеспечение высокого уровня доверия со стороны пользователей.
4. **Как решение отражено в архитектуре:** Внедрены методы шифрования данных, усилены механизмы аутентификации и авторизации, введены процессы для управления доступом и контроля за использованием персональных данных.
5. **Отклоненные альтернативы:** Не обеспечивать соответствие GDPR для обеспечения конфиденциальности и безопасности персональных данных.
6. **Причина отклонения:** Несоблюдение GDPR могло бы создать риск нарушения законодательства и ущерба для пользователей и системы JSE.