



Ассистент для команд разработки

Функциональные характеристики ПО

Всего листов: 15

Аннотация

Настоящий документ описывает функциональные характеристики программного обеспечения системы «Ассистент для команд разработки» (АКР, далее - Система). Он включает в себя описание особенностей реализации, ролей и полномочий пользователей, а также подробное описание функциональных возможностей системы, от отправки запросов Пользователя в текстовом чате и их обработки с использованием ИИ до использования плагинов для интегрированных сред разработки, формирования рекомендаций по улучшению программного кода и визуализации результатов. Также документ содержит описание технической и инфраструктурной архитектуры системы.

Содержание

Аннотация	2
1. Общие сведения	4
1.1. Наименование системы	4
1.2. Область применения системы	4
1.3. Краткое описание возможностей системы	4
1.4. Глоссарий	4
2. Цель и назначение системы	6
2.1. Цель системы	6
2.2. Назначение системы	6
2.3. Класс критичности системы	6
3. Функциональные характеристики системы	7
3.1. Особенности реализации системы	7
3.2. Роли и полномочия пользователей	7
3.3. Функциональные возможности системы	7
3.3.1. Требования к функции генерации API-ключей для аутентификации пользователей	8
3.3.2. Требования к функции разграничения прав доступа пользователей	8
3.3.3. Требования к функции поддержки текстового чата с возможностью отправки запросов к LLM и получения ответов от LLM	9
3.3.4. Требования к функции учета контекста диалога	9
3.3.5. Требования к функции функциям генерации кода, тестов и комментариев	10
3.3.6. Требования к функции уточнения и корректировки запросов к модели	10
3.3.7. Требования к функции изменения параметров генерации	10
3.3.8. Требования к функции сохранения пользовательских конфигураций параметров модели	11
3.3.9. Требования к функции предустановленных конфигураций параметров модели	11
3.3.10. Требования к функции создания, редактирования и удаления подсказок для модели (промптов)	12
3.3.11. Требования к функции сохранения и применения подсказок к диалогам	12
3.3.12. Требования к функции интеграции загруженных документов с ИИ для анализа и извлечения данных	13
3.3.13. Требования к функции визуализации результатов работы системы в графическом интерфейсе	13
4. Технологический стек	14
5. Контекстная архитектура инфраструктуры системы	15

1. Общие сведения

1.1. Наименование системы

Полное наименование системы – Ассистент для команд разработки.

Сокращенное наименование системы – АКР.

Правообладатель – ООО «ГК «ИННОТЕХ».

1.2. Область применения системы

Областью применения Системы является деятельность по написанию программного кода, тестированию ПО и написанию технической и сопровождающей документации для разработки ПО.

1.3. Краткое описание возможностей системы

Система предоставляет следующие возможности:

- Генерация текстовых ответов на вопросы Пользователя, которые содержат текст на естественном языке, код на языке программирования или файл;
- Генерация программного кода по описанию на естественном языке. Поддержка генерации на различных языках программирования;
- Написание unit-тестов к программному коду;
- Выполнение анализа исходного кода (код-ревью) и подготовка рекомендаций по его улучшению;
- Написание комментариев к pull requests;
- Написание тест-кейсов на основе требований к ПО.

1.4. Глоссарий

В тексте настоящего документа представлены следующие сокращения (см. **Таблица 1-1**).

Таблица 1-1 — Перечень сокращений

Сокращение/аббревиатура	Значение
ИИ	Искусственный интеллект
ПО	Программное обеспечение
АКР	Ассистент для команд разработки
Система	Ассистент для команд разработки
IDE	Интегрированная среда разработки
LLM	Большая языковая модель

В тексте настоящего документа представлены следующие термины и определения (см. **Таблица 1-2**).

Таблица 1-2 — Перечень терминов и определений

Термин	Определение
Аутентификация	Проверка подлинности предъявленного Пользователем идентификатора
Администратор Системы	Пользователь Системы с ролью «Администратор»
Заказчик	Физическое или юридическое лицо, которое инициирует, заказывает и финансирует разработку, модификацию или поддержку программного обеспечения Системы. После приемки услуг по разработке Системы Заказчик является Владельцем Системы
Пользователь	Работник, наделенный правами доступа к информационным ресурсам организации
Подсказка	Системный промпт для LLM - представляет собой руководящие инструкции или исходные данные, которые задаются модели, чтобы она могла генерировать ответы на запросы пользователя
Ролевая модель	Распределение прав доступа и обязанностей между ролями. Определяет, какие действия может выполнять каждая роль в рамках работы с Системой, включая создание и редактирование правил, проведение проверок и управление
Роль	Набор полномочий, который необходим Пользователю для выполнения определённых рабочих задач. Каждый сотрудник может иметь одну или несколько ролей, а каждая роль может содержать от одного до множества полномочий, которые разрешены Пользователю в рамках этой роли. Роли Пользователей в Системе: — «Пользователь»; «Администратор».
ИТ-инфраструктура	Совокупность всего программного обеспечения, оборудования, сетей и подключенных сервисов, образующих ИТ-среду организации
Компонент (программного обеспечения)	Составная часть программного обеспечения, выполняющая определенную функцию
Файл	Цифровой носитель информации, содержащий текстовые данные, с любым из перечисленных расширений: .txt, .java, .js, .ts, .html, .MD.

2. Цель и назначение системы

2.1. Цель системы

Целью создания Системы является автоматизация написания программного кода, тестов ПО, написания сценариев тестирования и текста для технической и сопровождающей документации на ПО с помощью генеративного ИИ для повышения продуктивности команд разработки ПО. Система должна обеспечивать быструю, точную и надежную генерацию кода, тестов и текста для документации, снижая риски ИБ и количество дефектов в разрабатываемом продукте.

2.2. Назначение системы

Система предназначена для ускорения процесса написания программного кода за счет ответов на вопросы, улучшения качества кода через анализ ошибок и рекомендаций по улучшению, сокращения времени разработки и вывода продуктов на рынок, сокращения материальных затрат на разработку новых сервисов.

В целях помощи командам разработки Система предназначена для удовлетворения задач использования безопасных решений, соответствующих стандартам работы в корпоративных контурах, включая поддержку замкнутых сетей и исключение утечек данных.

Система подразумевает возможность подбора размера LLM под существующую ИТ-инфраструктуру и потребности Заказчика.

Система подразумевает её использование в качестве веб-приложения или по API другими приложениями, такими как плагины для интегрированных сред разработки (IDE).

В Системе предусматривается использование инструментов для оценки эффективности решения и мониторинга активности пользователей.

Система предназначена для организаций, занимающихся разработкой и поддержкой программных продуктов с большим количеством команд разработки, и направлена на повышение безопасности и эффективности процессов разработки, анализа и тестирования ПО.

2.3. Класс критичности системы

Класс критичности Системы – Business Operational.

3. Функциональные характеристики системы

Данный раздел включает в себя описание особенностей реализации, ролей и полномочий пользователей, а также подробное описание функциональных возможностей Системы.

3.1. Особенности реализации системы

Система «Ассистент для команд разработки» реализована в виде веб-приложения, работа с которым осуществляется через веб-браузер, а также через плагины для интегрированной среды разработки (IDE) VS Code и IntelliJ IDEA.

3.2. Роли и полномочия пользователей

Система предполагает наличие следующих ролей:

- продуктовая роль: «Пользователь»;
- служебная роль: «Администратор».

В таблице ниже (см. **Таблица 3-1**) представлено описание доступных действий и полномочий каждой из ролей системы.

Таблица 3-1 — Роли пользователей

Наименование роли	Описание роли	Описание требований к квалификации
Администратор	Главная роль, предоставляющая доступ к настройкам Системы	Опыт разработки ПО, знание основ информационной безопасности, навыки администрирования серверов и вычислительных сетей. Рекомендуется наличие навыков и опыта работы с технологиями docker, kubernetes, keycloak, postgresql, MySQL.
Пользователь	Роль для лица, использующего функциональность Системы	Базовые навыки работы с компьютером и программным обеспечением, базовые навыки программирования.

3.3. Функциональные возможности системы

Функции, которые должны выполняться Системой:

- Генерация API-ключей для аутентификации пользователей;
- Разграничение прав доступа пользователей;
- Поддержка текстового чата с возможностью отправки запросов к LLM и получения ответов от LLM;
- Поддержка контекста и истории сообщений диалога текстового чата;
- Система должна предоставлять возможность генерации программного кода, кода тестов ПО и текста для технической и сопровождающей документации на разработку ПО по запросу Пользователя и автоматически - при работе с плагинами для IDE

(автодополнение кода, генерация комментариев к коду, генерация комментариев к запросу на изменение кода репозитория для последующей интеграции в основной проект);

- Обеспечение возможности уточнения и корректировки Пользователем запроса к LLM;
- Обеспечение возможности изменения параметров генерации ответов LLM в текстовом чате (температура, длина ответа, глубина контекста);
- Обеспечение возможности сохранения конфигурации параметров ответов LLM, заданных Пользователем;
- Поддержка предустановленной конфигурации параметров ответов LLM;
- Обеспечение возможности Пользователя создавать, редактировать и удалять подсказки для LLM (промпты);
- Обеспечение возможности Пользователем сохранить подсказку для LLM и возможности её применения к текстовым диалогам;
- Система должна позволять пользователям загружать документацию в качестве дополнительных знаний для LLM в различных форматах (.txt, .java, .js, .ts, .html, .MD);
- Обеспечение возможности запуска функций Системы и визуализация результатов работы функций Системы в графическом интерфейсе.

3.3.1. Требования к функции генерации API-ключей для аутентификации пользователей

Система должна обеспечить генерацию API-ключей для аутентификации пользователей и возможности предоставления дальнейшего доступа к функциональности. Первичная авторизация новых пользователей с автоматической генерацией API-ключа для каждого пользователя должна выполняться через веб-интерфейс доступа к Системе.

Аутентификация пользователя для предоставления дальнейшего доступа к функциональности Системы должна позволять однозначно идентифицировать Пользователя в Системе, назначать ему соответствующие права. Система должна обеспечивать безопасность хранения и защиту учетной записи Пользователя.

Результатом успешной работы функции должно быть сохранение информации об учетной записи пользователя, предоставление Пользователю доступа к функциональности Системы согласно назначенным правам, а также уведомление Пользователя о аутентификации.

3.3.2. Требования к функции разграничения прав доступа пользователей

Система должна обеспечить разграничение прав доступа пользователей в зависимости от их роли (Ролевая модель представлена в **Таблица 3-1** настоящего документа). Каждой роли должны соответствовать определенные права и ограничения при взаимодействии с Системой.

Предоставление разных уровней доступа позволяет управлять функциональностью Системы, предотвращая несанкционированные изменения важных настроек и данных. Администраторы должны получать расширенные возможности по управлению пользователями и конфигурацией Системы, а также доступ к интерфейсу мониторинга показателей работы Системы и активности

пользователей. Пользователи должны получать возможность взаимодействия с функциональностью Системы.

Результатом успешной работы функции должно быть обеспечение разграничения прав доступа пользователей согласно назначенной роли. В зависимости от роли пользователь должен получать доступ только к тем функциям, которые ему разрешены, а система должна корректно ограничивать доступ к административным и техническим настройкам.

3.3.3. Требования к функции поддержки текстового чата с возможностью отправки запросов к LLM и получения ответов от LLM

Система должна обеспечивать интерактивный текстовый чат, позволяющий пользователям отправлять запросы к LLM и получать ответы от LLM в режиме реального времени. Чат должен поддерживать ввод текстовых сообщений, загрузку файлов и управление контекстом диалога.

Поддержка текстового чата позволяет пользователям взаимодействовать с Системой, формулировать вопросы, уточнять детали, получать сгенерированный код, тесты, текст для документации и другую информацию, релевантную их задачам. Чат должен сохранять историю диалога, обеспечивать корректное отображение сообщений и поддерживать динамическое обновление интерфейса.

Система должна поддерживать функцию фильтрации и поиска по ключевым словам внутри диалога с LLM.

Система должна обеспечивать функцию поддержки текстового чата с LLM как в веб-интерфейсе, так и в плагинах для IDE.

Результатом успешной работы функции должно быть предоставление авторизованному Пользователю доступа к функциональности работы с текстовым чатом, такой как ввод текстовых запросов и получение текстовых ответов от LLM, просмотр истории сообщений открытого диалога и, при необходимости, корректировке запросов. Система должна обеспечивать стабильную работу чата, оперативную обработку запросов Пользователя и возможность работы с чатом через графический интерфейс.

3.3.4. Требования к функции учета контекста диалога

Система при работе Пользователя с текстовым чатом должна обеспечить способность учитывать контекст диалога и хранить историю диалога. Это включает в себя анализ предыдущих сообщений и корректировку ответов на основе ранее заданных вопросов или уточнений.

Учет контекста диалога позволяет Системе предоставлять более точные и релевантные ответы, избегая повторений и недоразумений. Система должна запоминать ключевые моменты общения, такие как запросы, темы и предпочтения пользователя, чтобы обеспечить персонализированный подход к каждому новому взаимодействию.

Результатом успешной работы функции должно быть обеспечение возможности Пользователю взаимодействовать с текстовым чатом в рамках диалога, при этом Система должна корректно интерпретировать запросы с учетом ранее обсужденных тем и вопросов. История диалога должна быть доступна для просмотра, а Система должна динамически обновлять и сохранять контекст в

процессе взаимодействия.

3.3.5. Требования к функции функциям генерации кода, тестов и комментариев

Система должна обеспечить возможность запрашивать генерацию кода, тестов и комментариев к коду через текстовый чат или в виде подсказок в открытых проектах IDE с установленным плагином Системы. Пользователь должен иметь возможность четко указать задачу (например, написать функцию, создать тест, или сгенерировать сценарии тестирования), а Система должна предоставить результат, соответствующий запросу.

Возможность запрашивать генерацию кода, тестов и текста для документации, а также генерации комментариев к коду, позволяет пользователям автоматизировать рутинные задачи, ускорить процесс разработки и обеспечения качества, а также сэкономить время на написание документации. Это важно для команд разработки, которым нужно быстро получать рабочие решения для задач в рамках проектов.

Результатом успешной работы функции должно быть выполнение генерации ответа LLM по запросам пользователей, а также автоматическая генерация подсказок и кода в IDE. Ответы должны быть отображены в графическом интерфейсе Системы и интерфейсе IDE с установленным плагином Системы и должны удовлетворять требованиям пользователей.

3.3.6. Требования к функции уточнения и корректировки запросов к модели

Система должна обеспечить возможность пользователю уточнять и корректировать свои запросы к модели в процессе взаимодействия. Пользователь должен иметь возможность изменить свой запрос, добавить дополнительные детали или задать новые вопросы на основе полученного ответа, чтобы улучшить точность и релевантность результата.

Возможность уточнения и корректировки запросов позволяет пользователям получить более точные и полезные ответы от LLM. Это особенно важно в случае, когда начальный запрос не дал нужного результата или необходимо более детализированное объяснение. Система должна легко обрабатывать уточнения и реагировать на изменения запросов, корректируя ответы модели в соответствии с новой информацией.

Результатом успешной работы функции должна быть возможность Пользователя изменять или дополнять свои запросы в ходе общения с ИИ. Система должна корректно воспринимать уточнения, адаптировать модель под новые требования и предоставлять более точные или подробные ответы в ответ на исправленные запросы.

3.3.7. Требования к функции изменения параметров генерации

Система должна обеспечить возможность пользователю изменять параметры генерации ответов, такие как температура (степень случайности генерации), длина ответа (максимальное количество символов или слов в ответе) и глубина контекста (сколько предыдущих сообщений учитывать при формировании ответа). Пользователь должен иметь простой и интуитивно понятный интерфейс для настройки этих параметров.

Возможность изменять параметры генерации позволяет пользователям адаптировать поведение модели под свои нужды, например, задавая более точные или разнообразные ответы, сокращая или увеличивая длину текста, а также регулируя уровень учитываемой в контексте диалога истории сообщений в зависимости от задачи. Это улучшает персонализацию работы с моделью и дает пользователям полный контроль над качеством и форматом ответов.

Результатом успешной работы функции должна быть возможность пользователя настроить параметры генерации для каждого запроса или сохранить настройки по умолчанию для будущих диалогов. Система должна корректно интерпретировать изменения этих параметров и предоставлять ответы, которые соответствуют заданным параметрам генерации (температуре, длине и глубине контекста).

3.3.8. Требования к функции сохранения пользовательских конфигураций параметров модели

Система должна обеспечить возможность пользователю сохранять свои настройки параметров модели (температура, длина ответа, глубина контекста) в виде пользовательских конфигураций. Эти конфигурации должны быть доступны для дальнейшего использования, с возможностью выбора сохраненных настроек при необходимости.

Возможность сохранять пользовательские конфигурации параметров модели позволяет пользователю быстро повторно использовать оптимальные настройки без необходимости вручную изменять параметры каждый раз. Это повышает удобство работы, особенно при работе с различными типами задач, где разные параметры генерации могут быть более эффективными.

Результатом успешной работы функции должна быть возможность Пользователя сохранять и загружать собственные конфигурации параметров модели, что позволит ускорить процесс настройки и повышения эффективности работы с моделью. Система должна корректно сохранять изменения и обеспечивать доступ к сохраненным конфигурациям, а также позволять пользователю редактировать или удалять их при необходимости.

3.3.9. Требования к функции предустановленных конфигураций параметров модели

Система должна обеспечить наличие предустановленных конфигураций параметров модели (температура, длина ответа, глубина контекста), которые будут предложены пользователю по умолчанию. Эти конфигурации должны быть оптимальными для наиболее распространенных задач, с возможностью выбора пользователем в зависимости от их потребностей.

Наличие предустановленных конфигураций позволяет пользователю сразу приступить к работе без необходимости настраивать параметры вручную для каждого запроса. Это упрощает взаимодействие с системой, особенно для пользователей, которые не знакомы с параметрами генерации или не хотят тратить время на их настройку. Предустановленные конфигурации также помогут достичь сбалансированных и эффективных результатов при типовых задачах.

Результатом успешной работы функции должно быть наличие в Системе нескольких предустановленных конфигураций, которые будут предлагаться пользователю при первом запуске

или в процессе работы. Пользователь должен иметь возможность выбрать одну из этих конфигураций и применить ее для своих запросов. Система должна гарантировать, что каждая предустановленная конфигурация будет оптимально настроена для широкого спектра задач, обеспечивая качество и точность ответов.

3.3.10. Требования к функции создания, редактирования и удаления подсказок для модели (пром프트)

Система должна обеспечить возможность пользователю создавать, редактировать и удалять подсказки (пром프트) для модели. Пользователь должен иметь удобный интерфейс для добавления собственных пром프트, а также возможность изменять или удалять их в любое время в процессе работы с моделью.

Возможность создавать, редактировать и удалять подсказки позволяет пользователям настроить взаимодействие с моделью в соответствии с их специфическими потребностями. Это помогает повысить точность и релевантность ответов, так как пользователь может подстраивать подсказки под определенные задачи или контексты, улучшая взаимодействие с системой и оптимизируя результаты генерации.

Результатом успешной работы функции должна быть возможность создания и управления персонализированными подсказками для модели. Пользователь должен иметь возможность добавлять новые подсказки, редактировать их для корректировки вывода модели и удалять устаревшие или ненужные подсказки. Система должна сохранять эти изменения и корректно применять подсказки при генерации ответов, повышая эффективность работы и точность взаимодействия с моделью.

3.3.11. Требования к функции сохранения и применения подсказок к диалогам

Система должна обеспечить возможность пользователю сохранять одну активную подсказку (пром프트) и применять её к текущим или будущим диалогам. Пользователь должен иметь возможность выбрать подсказку из списка сохраненных и установить её как активную для применения в ходе взаимодействия с моделью.

Возможность сохранять подсказки и применять их к диалогам позволяет пользователю персонализировать взаимодействие с моделью, улучшая качество и релевантность ответов на основе заданной подсказки. Это упрощает работу с моделью, обеспечивая более точные результаты, соответствующие специфике текущего диалога или задачи.

Результатом успешной работы функции должна быть способность пользователя сохранить одну активную подсказку и применить её к текущему диалогу. Подсказка будет влиять на ответы модели в процессе общения, при этом система должна гарантировать, что только одна подсказка будет применяться к диалогу в каждый момент времени. Пользователь сможет изменять активную подсказку, и она будет немедленно применяться к диалогу.

3.3.12. Требования к функции интеграции загруженных документов с ИИ для анализа и извлечения данных

Система должна обеспечить интеграцию загруженных документов с ИИ для анализа и извлечения данных. После загрузки файлов (.txt, .java, .js, .ts, .html, .MD), Система должна использовать модель для анализа содержимого файлов, извлекая ключевые данные, такие как текст, таблицы, графики, даты, имена и другие структурированные данные, а также предоставлять возможность их использования в дальнейших диалогах.

Интеграция загруженных документов с ИИ позволяет пользователям эффективно извлекать информацию из документов для дальнейшего использования в работе. Это упрощает процесс извлечения данных, улучшает точность анализа и позволяет оперативно использовать информацию из документов в различных задачах, таких как генерация отчетов, создание аннотаций или ответов на запросы.

Результатом успешной работы функции должна быть способность системы автоматически анализировать содержимое загруженных документов, извлекать ключевую информацию и предоставлять пользователю возможность использовать эти данные в диалогах с LLM. Пользователь сможет запросить у LLM о содержимом документа, и система ответит на основе извлеченной информации, что ускоряет обработку и анализ данных.

3.3.13. Требования к функции визуализации результатов работы системы в графическом интерфейсе

Система должна обеспечить интуитивно понятный интерфейс, который предоставляет пользователю легкий доступ ко всем ключевым функциям, включая создание и управление диалогами, загрузку документов, настройку параметров модели, поиск и фильтрацию информации, а также управление подсказками. Интерфейс должен быть простым в использовании, с логичной структурой и понятными элементами управления.

Интуитивно понятный интерфейс с доступом ко всем ключевым функциям позволяет пользователям быстро освоить систему и эффективно работать с её возможностями. Это улучшает опыт взаимодействия, минимизирует количество ошибок и повышает производительность, так как пользователи могут легко находить и использовать нужные инструменты без необходимости в дополнительном обучении.

Результатом успешной работы функции должно быть наличие удобного интерфейса, который позволит пользователю без усилий получать доступ ко всем основным функциям системы. Все элементы управления должны быть логично расположены, а взаимодействие с системой — естественным и понятным, что обеспечит удобство и эффективность работы пользователя с ИИ.

4. Технологический стек

Описание компонентов технологического стека Системы приведено в **Таблице 4-1**.

Таблица 4-1 – Компоненты технологического стека

Компонент	Описание
Базовая операционная система	Linux
Большая языковая модель	Qwen2.5-Coder-32B
Язык разработки Backend	JavaScript, TypeScript, Java, Python
Технология сборки	NPM, Gradle, PIP
Веб-сервер	Nginx
Фреймворк Frontend	React, Vite, tailwind
Реляционная СУБД	MySQL Community Edition
Резидентная СУБД	Redis Open Source
Документоориентированная БД	MongoDB
Платформа для запуска приложений в контейнерах	Docker Community Edition
Платформа оркестрации контейнеров	Kubernetes (K8s) open source
Компонент интеграции поиска и генерации ответов ИИ-моделей	LangChain, vLLM
Визуализация мониторинга	Grafana OSS

5. Контекстная архитектура инфраструктуры системы

Контекстная схема архитектуры Системы приведена на **Рис. 5-1**.

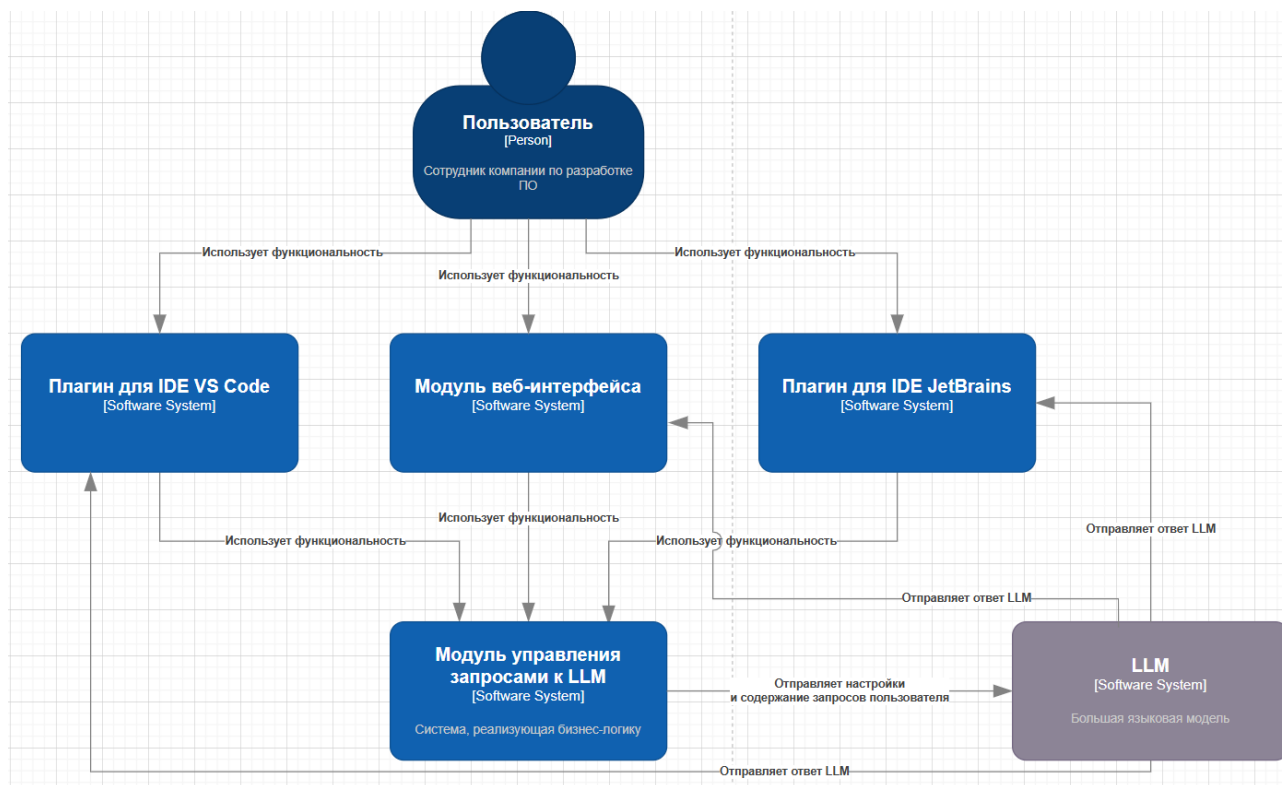


Рис. 5-1 - Концептуальная схема архитектуры Системы