**Федеральное агентство связи Федеральное государственное образовательное бюджетное учреждение высшего профессионального образования «Санкт-Петербургский государственный университет телекоммуникаций им. проф. М.А. Бонч-Бруевича»**

Факультет «Инфокоммуникационных сетей и систем»

**КУРСОВАЯ РАБОТА**

**по дисциплине**: Проектирование и архитектура программных систем

**по теме**: «Разработка САА для микроконтроллера Mik32 Amur»

доцент каф. ПИиВТ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ к.т.н., Смирнов К.А.

Санкт-Петербург

2024 г

**Оглавление**

1. Актуальность……………………………………………………………….....4

2. Общие сведения………………………………………………………………5

2.1. Полное наименование системы и ее условное обозначение………… 5

2.2. Плановые сроки начала и окончания работы по созданию системы...5

3. Назначение и цели создания системы……………..……………….……….5

3.1. Назначение системы……………………………………………………..5

3.2. Цели создания системы………………………………………………….6

4. Характеристики объекта информатизации…………………………………7

4.1. Краткие сведения об объекте информатизации или ссылки на документы, содержащие такую информацию………………………………...7

4.2. Сведения об условиях эксплуатации объекта информатизации……...8

5. Требования к структуре и функционированию системы…………………..9

5.1. Перечень подсистем, их назначение и основные характеристики……9

5.2. Требования к функциям (задачам), выполняемым системой………...11

5.3 Требования к способам и средства связи для информационного обмена между компонентами системы………………………………………………...21

5.4. Требования к эргономике и технической эстетике…………………...22

5.5 Требования к защите информации от несанкционированного доступа

…………………………………………………………………………….22

5.6. Перспективы развития, модернизации системы……………………....23

6. Требования к видам обеспечения…………………………………………...23

6.1. Требования к лингвистическому обеспечению………………………..23

6.2. Требования к программному обеспечению…………………………….23

**Словарь с терминологией**

Библиотека — сборник подпрограмм или объектов, используемых для разработки программного обеспечения.

Встраиваемая система (встроенная система, англ. embedded system) — специализированная микропроцессорная система управления, контроля и мониторинга, концепция разработки которой заключается в том, что такая система будет работать, будучи встроенной непосредственно в устройство, которым она управляет.

Интернет вещей (англ. Internet of Things, IoT) — концепция сети передачи данных между физическими объектами («вещами»), оснащёнными встроенными средствами и технологиями для взаимодействия друг с другом или с внешней средой.

Интерфейс (англ. interface) — это устройство или система для взаимодействия между не связанными друг с другом объектами.

Исполнительный механизм — устройство в системе автоматического регулирования, дистанционного управления и других систем управления, осуществляющее механическое перемещение регулирующего органа.

САА (слой аппаратных абстракций, англ. Hardware Abstraction Layer, HAL) — слой абстрагирования, реализованный в программном обеспечении, находящийся между физическим уровнем аппаратного обеспечения и программным обеспечением, запускаемом на этом компьютере.

Фреймворк — программная платформа, определяющая структуру программной системы; программное обеспечение, облегчающее разработку и объединение разных компонентов большого программного проекта.

1. **Актуальность**

Современные тенденции в области разработки встроенных систем и интернета вещей требуют высокой производительности, надежности и безопасности программного обеспечения. Микроконтроллеры, такие как Mik32 Amur, становятся все более популярными благодаря своей способности эффективно обрабатывать данные и поддерживать различные протоколы связи. Однако для реализации полного потенциала этих устройств необходимы надежные программные решения, которые обеспечивают взаимодействие между аппаратным обеспечением и приложениями.

Разработка САА для микроконтроллера Mik32 Amur актуальна по нескольким причинам:

* **Безопасность**: САА минимизирует риски, связанные с управлением памятью, что критично для встроенных систем, где сбои могут привести к серьезным последствиям. Это особенно важно для приложений, работающих в условиях реального времени, где стабильность и надежность являются приоритетами;
* **Производительность**: Высокая производительность низкоуровневых языков позволяет эффективно использовать ресурсы микроконтроллера Mik32 Amur, что особенно важно для задач, требующих быстрой обработки данных и минимальной задержки;
* **Совместимость** **и** **гибкость**: Mik32 Amur поддерживает множество интерфейсов и протоколов, что делает САА легко интегрируемым с существующими решениями и позволяющим разработчикам быстро адаптировать свои приложения под новые требования;
* **Поддержка** **сообщества** **и** **экосистемы**: Активное сообщество и растущая библиотека готовых решений создают благоприятные условия для разработки новых инструментов и библиотек, что способствует ускорению процесса разработки и повышению качества программных продуктов.

**2. Общие сведения**

**2.1. Полное наименование системы и ее условное обозначение**

**Полное наименование:** “Слой аппаратных абстракций для микроконтроллера Mik32 Amur” – библиотека для написания пользовательских программ. Она позволяет разработчикам программного обеспечения получать доступ к внутренним ресурсам и интерфейсам коммуникации микроконтроллера. Также, при надобности, с помощью функционала библиотеки возможна огранизация работы с подключенными к микроконтроллеру устройствами периферии.

**Условное обозначение:** САА Mik32 Amur.

**2.2. Плановые сроки начала и окончания работы по созданию системы**

**Дата начала:** 11.01.2024 г.

**Дата окончания:** 01.05.2025 г.

**3. Назначение и цели создания системы**

**3.1. Назначение системы**

САА для микроконтроллера Mik32 Amur представляет собой комплексный инструмент, предназначенный для упрощения разработки и взаимодействия с аппаратными компонентами. Она ориентирована как на индивидуальных разработчиков, так и на команды, работающие над проектами в области встроенных систем.

САА обеспечивает разработчиков высокоуровневым интерфейсом, который позволяет легко и эффективно взаимодействовать с различными аппаратными модулями, такими как датчики, исполнительные механизмы и интерфейсы связи, что позволяет разработчикам сосредоточиться на логике приложения, а не на низкоуровневых деталях работы оборудования.

Кроме того, САА предоставляет инструменты для анализа и оптимизации работы микроконтроллера, позволяя разработчикам адаптировать свои решения с учетом целевых требований производительности и энергопотребления.

САА включает в себя широкий спектр инструментов для мониторинга и отладки, таких как вывод информации в консоль разработчика, средства тестирования и профилирования, что позволяет разработчикам проводить глубокий анализ производительности и выявлять потенциальные проблемы на ранних стадиях разработки.

Суммируя вышеизложенное, система САА для микроконтроллера Mik32 Amur стремится предоставить высокотехнологичное и интегрированное решение для разработки встроенных приложений, учитывая сложные аспекты работы с аппаратными компонентами и предоставляя разработчикам инструменты для достижения оптимальной производительности и надежности своих решений. Аналогами системы САА являются различные библиотеки и фреймворки, такие как ESP IDF и Arduino Core.

**3.2. Цели создания системы**

Целями создания САА Mik32 Amur являются:

* Создание высокоуровневого интерфейса:

Разработка удобного и интуитивно понятного интерфейса для взаимодействия с аппаратными компонентами микроконтроллера.

* Повышение безопасности программного обеспечения:

Использование языка Rust в разработке САА для снижения рисков, связанных с ошибками, такими как переполнение буфера и некорректное управление памятью.

* Оптимизация производительности:

Обеспечение эффективного использования ресурсов микроконтроллера, включая поддержку многопоточности и асинхронного программирования.

* Обеспечение совместимости и расширяемости:

Поддержка множества интерфейсов и протоколов, что обеспечит совместимость с различными устройствами и системами.

* Упрощение процесса тестирования и отладки:

Разработка инструментов и методологий, облегчающих тестирование и отладку программного обеспечения.

Таким образом, цели создания системы САА для микроконтроллера Mik32 Amur направлены на разработку надёжной, безопасной и эффективной платформы, позволяющей упростить процесс разработки встроенных приложений.

**4. Характеристика объекта информатизации**

**4.1. Краткие сведения об объекте информатизации или ссылки на документы, содержащие такую информацию**

**Назначение системы:**

* САА предназначен для упрощения управления аппаратными ресурсами микроконтроллера Mik32 Amur, предоставляя разработчикам инструменты для эффективного взаимодействия с компонентами и принятия обоснованных решений в процессе разработки.

**Основные функции:**

* Управление внутренними ресурсами микроконтроллера;
* Инструменты для оценки эффективности работы микроконтроллера;
* Интеграция с различными внешними модулями и протоколами для расширения функциональности.

**Технические характеристики:**

* Обеспечивает высокую производительность и надежность при работе с аппаратными ресурсами;
* Поддерживает различные интерфейсы и протоколы для интеграции с внешними устройствами, что позволяет легко расширять функциональность системы.

**Условия эксплуатации:**

* САА может быть интегрирован в различные программные среды, что позволяет разработчикам легко адаптировать его к специфическим требованиям проектов.
* Система предназначена для работы в условиях, обеспечивающих стабильную и надежную эксплуатацию микроконтроллера.

**Преимущества:**

* Обеспечение эффективного управления ресурсами и оптимизация работы микроконтроллера;
* Интеграция с различными внешними модулями для расширения возможностей системы;
* САА обеспечивает надежное управление памятью, предотвращая переполнение буферов и другие ошибки, что способствует стабильной и безопасной работе приложений на микроконтроллере.

**4.2. Сведения об условиях эксплуатации объекта информатизации**

**Температурные условия:**

Микроконтроллер Mik32 Amur должен функционировать в диапазоне температур, соответствующем спецификациям производителя.

**Электропитание:**

САА должна быть спроектирована с учетом стабильного и надежного питания. Микроконтроллер должен работать в условиях, когда напряжение питания находится в пределах допустимых значений, что обеспечивает защиту от перепадов напряжения и минимизирует риск повреждения оборудования.

**Электромагнитные помехи:**

Должна быть обеспечена защита от электромагнитных помех и других источников интерференции, которые могут повлиять на работу микроконтроллера и его взаимодействие с внешними устройствами.

**5. Характеристика объекта информатизации**

**5.1. Перечень подсистем, их назначение и основные характеристики**

* Подсистема Авторизации и Регистрации пользователя
* Подсистема Портфолио
* Подсистема Блог

На рисунке 1 представлена схема взаимодействия подсистем.

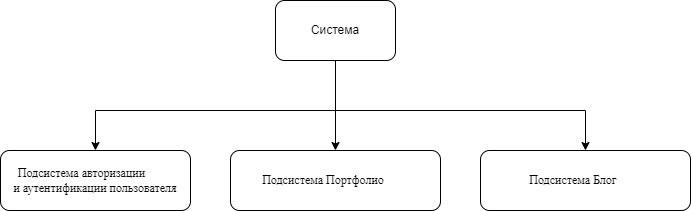


Рисунок 1. Древовидная структура подсистем

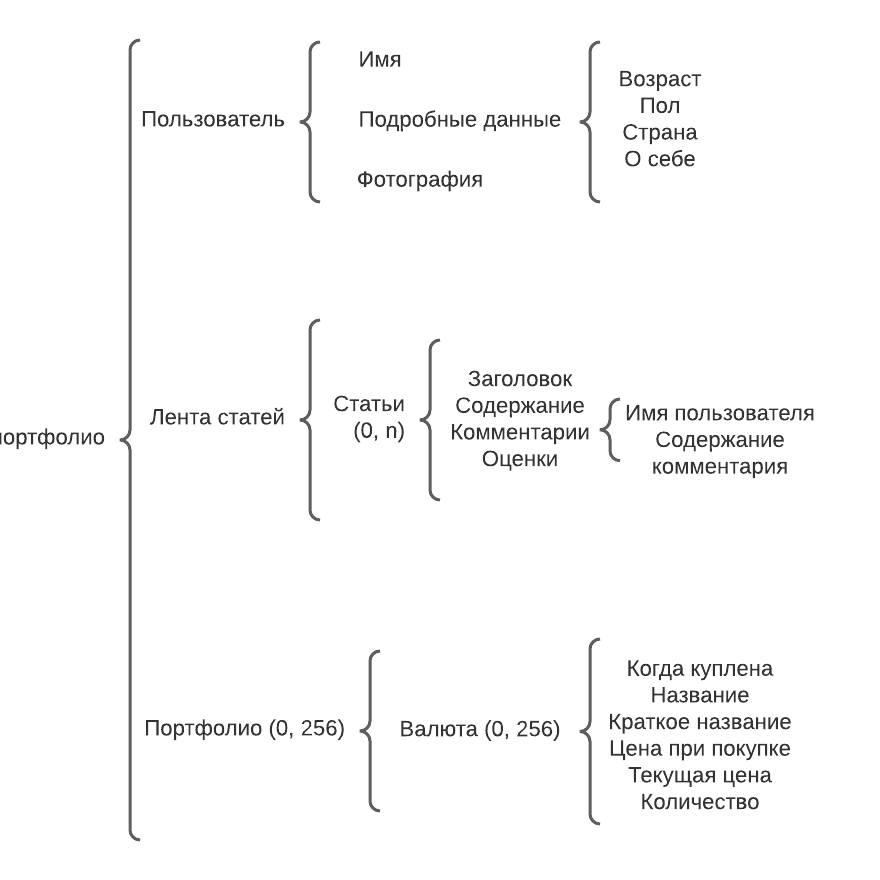


Рисунок 2. Диаграмма описания организации данных в системе Криптопортфолио

**5.2. Требования к функциям (задачам), выполняемым системой;**

**Подсистема авторизации и регистрации пользователя**

Подсистема необходима для создания профиля пользователя, авторизации и аутентификации. Она позволяет создавать профили (регистрироваться) для новых пользователей, а также совершать вход пользователям, у которых уже имеются профили в системе Крипто-портфолио.

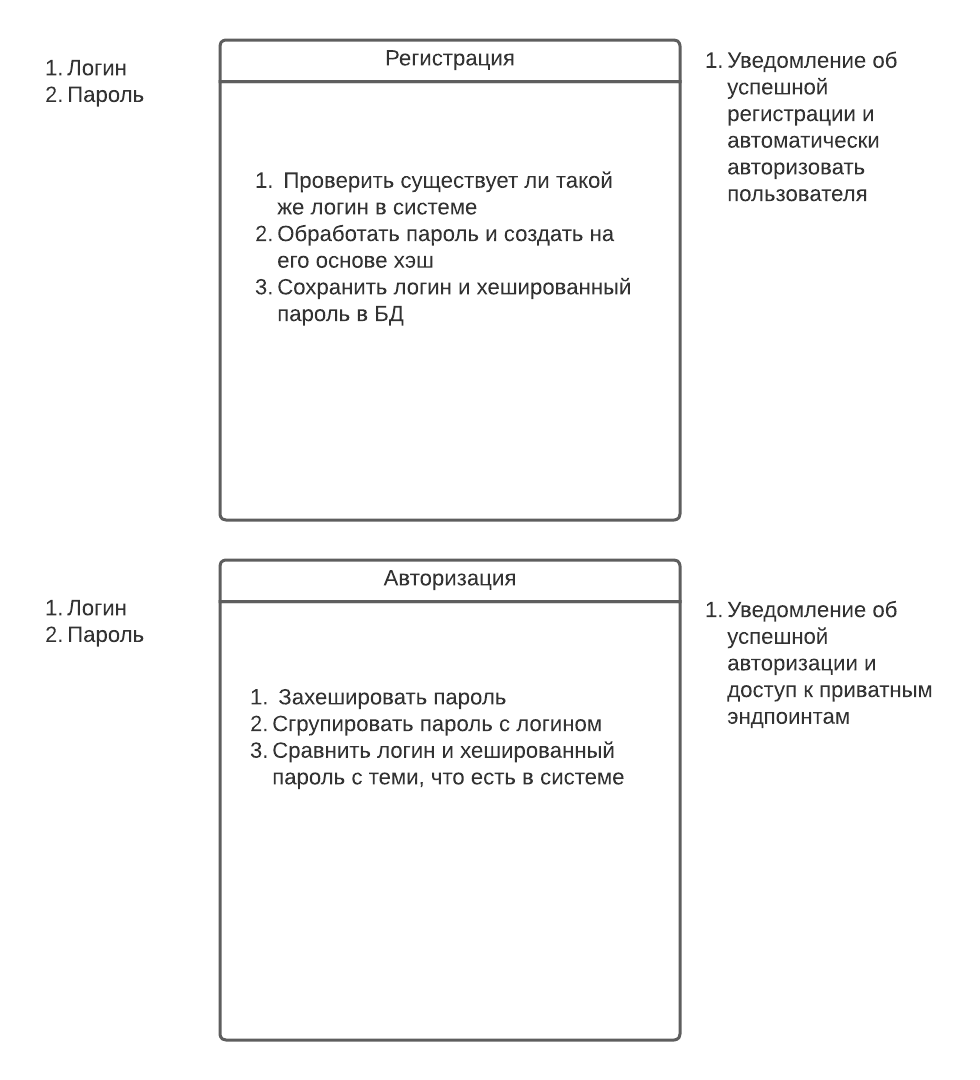


Рисунок 3. Схема HIPO для авторизации и регистрации пользователей

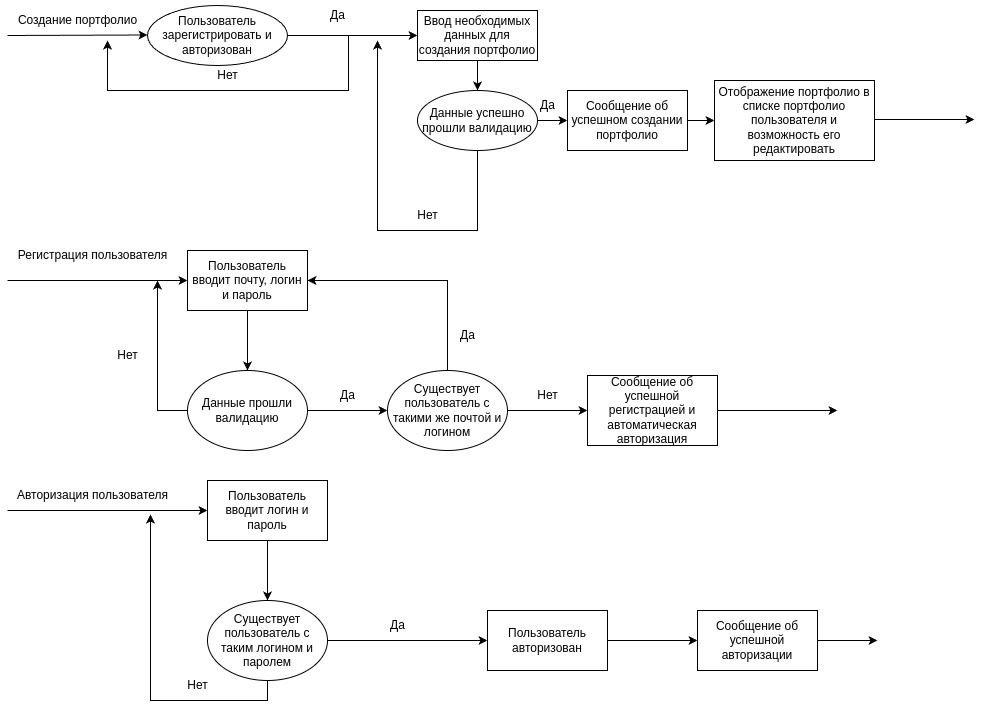


Рисунок 4. Блок-схема авторизации и регистрации пользовател

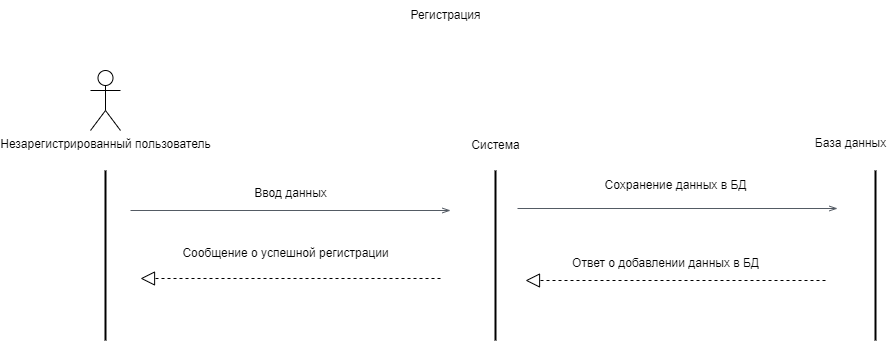


Рисунок 5. Регистрация пользователя

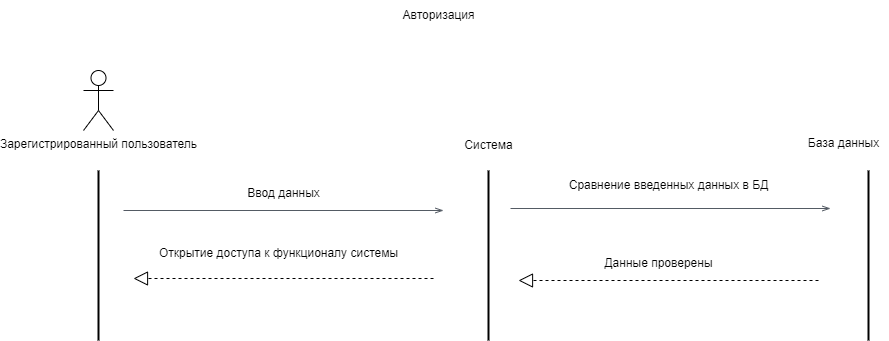


Рисунок 6. UML диаграмма авторизации

**Подсистема Портфолио**

Подсистема необходима для создания и управления портфолио пользователям. Она позволяет добавлять в нее валюты, анализировать их и управлять ими.

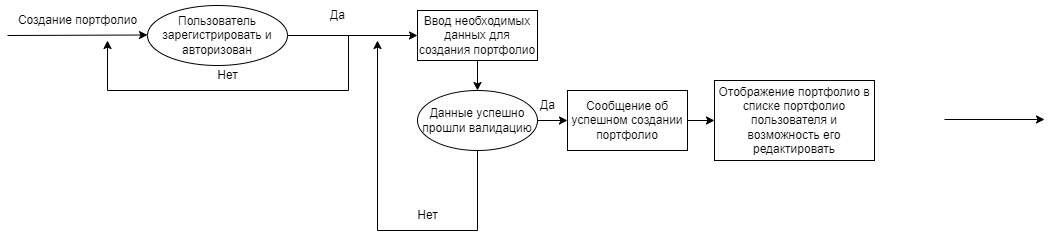


Рисунок 7. Блок схема создания портофолио

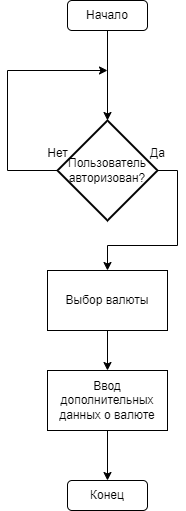


Рисунок 8. Добавление валюты в портфолио

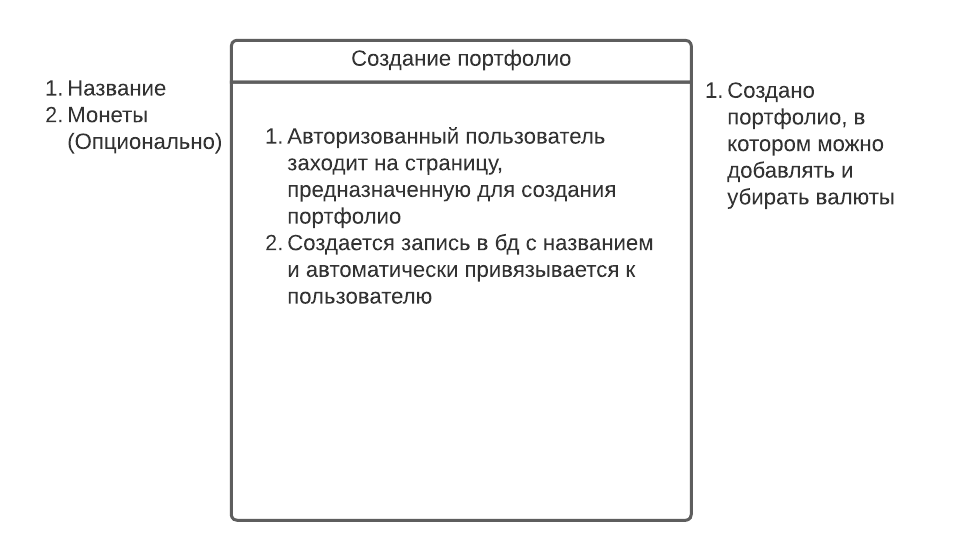
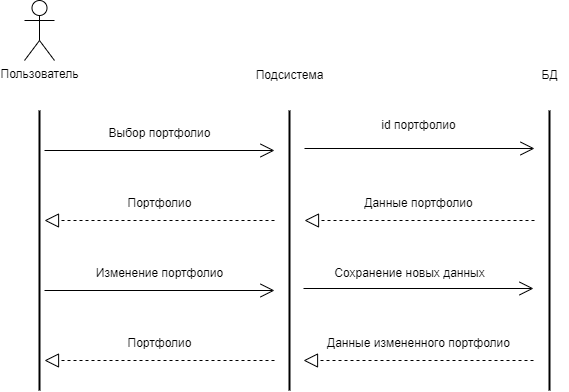


Рисунок 9. Схема HIPO создание портфолио



**Подсистема Блог**

Подсистема Блог необходима для добавления и просмотра пользовательских статей на тему крипто торговли. С помощью данной системы авторизованные пользователи могут создавать и публиковать свои статьи, также каждую опубликованную статью можно комментировать и оставлять оценку для нее.

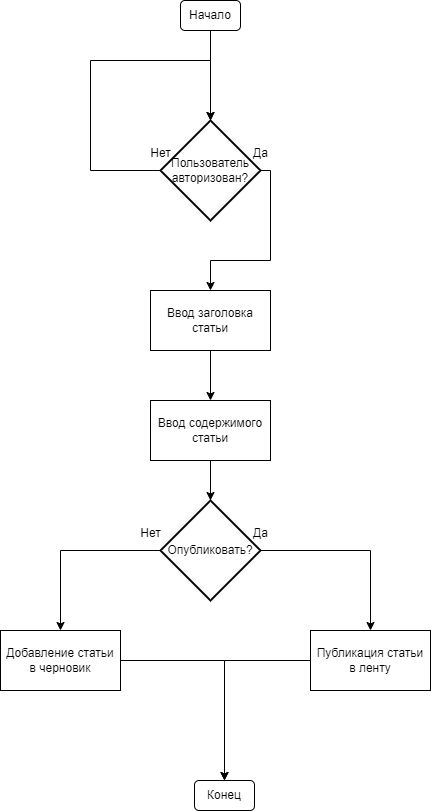


Рисунок 10. Блок-схема добавления статьи

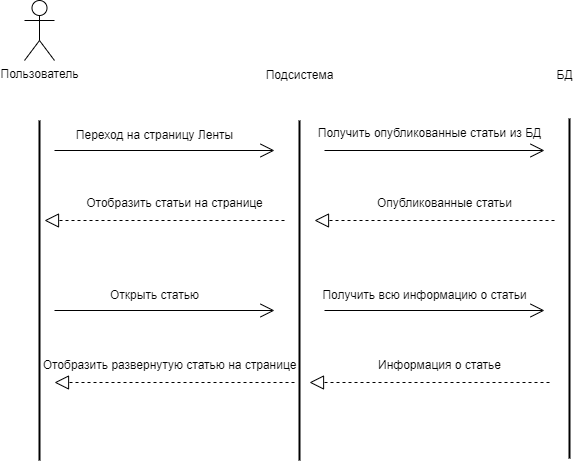


Рисунок 11. Отображение ленты статей и переход на страницу с выбранной статьей

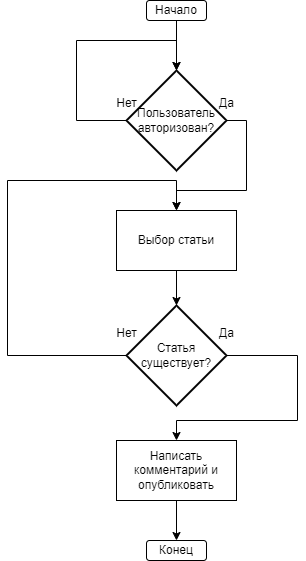


Рисунок 12. Блок-схема опубликования комментария для статьи

**5.3. Требования к способам и средствам связи для информационного обмена между компонентами системы**

Система должна поддерживать автоматический обмен данными между различными компонентами.

Использование стандартных протоколов связи, таких как HTTPS для безопасной передачи данных по сети, может обеспечить шифрование информации и защиту от несанкционированного доступа.

Реализация RESTful API (Representational State Transfer) может обеспечить удобство интеграции с различными компонентами системы, позволяя передавать данные в формате JSON или XML.

Использование асинхронных механизмов обмена сообщениями, таких как очереди сообщений (например, RabbitMQ или Apache Kafka), может улучшить производительность и отзывчивость системы.

Разделение системы на небольшие, автономные микросервисы может облегчить коммуникацию между компонентами и обеспечить гибкость в разработке и масштабировании.

Обеспечение протоколов шифрования для защиты передаваемой информации и механизмов аутентификации для проверки подлинности и авторизации компонентов.

Внедрение механизмов мониторинга и журналирования обмена данными для отслеживания производительности, выявления проблем и обеспечения безопасности.

Способы связи должны быть гибкими и легко адаптируемыми для совместимости с различными технологиями и изменениями в инфраструктуре.

Реализация механизмов обработки ошибок и транзакций для обеспечения надежности и целостности обмена информацией.

Данные требования смогут обеспечить эффективную, безопасную и надежную связь между компонентами системы.

**5.4. Требования к эргономике и технической эстетике**

Система должна удовлетворять следующим требованиям:

* взаимодействие системы и пользователя осуществляется на русском языке, за исключением системных сообщений.
* использование стандарта API.
* Интерфейс должен быть выполнен по стандартам:
  1. ISO 9241-12-1998
  2. ISO/IES 10741-1995
  3. ISO 9241-10-1996
  4. ISO/IES 13407-1999
  5. ГОСТ Р ИСО/МЭК 12119-2000
  6. ГОСТ Р ИСО/МЭК 9126 – 93
* дизайн интерфейса должен соответствовать корпоративному стилю.

**5.5. Требования к защите информации от несанкционированного доступа**

Защита информации от несанкционированного доступа в системе Криптопортфолио является критическим аспектом, учитывая чувствительность финансовой информации и персональных данных пользователей.

* Реализация надежных механизмов аутентификации, таких как двухфакторная аутентификация, для обеспечения подлинности пользователей.
* Система должна строго управлять доступом, обеспечивая правильные уровни авторизации для каждого пользователя и компонента.
* Применение шифрования для защиты данных, особенно при передаче чувствительной информации по сети.
* Реализация механизмов защиты сессий для предотвращения атак, связанных с угоном сеанса.
* Эффективное управление шифровальными ключами для предотвращения их утечки и обеспечения безопасного хранения.
* Внедрение системы мониторинга безопасности для обнаружения необычной активности, подозрительных попыток входа и других потенциальных угроз.
* Регистрация действий пользователей и системных событий для возможности проведения аудита и выявления возможных нарушений.
* Обеспечение физической безопасности серверов и инфраструктуры, где развернута система, чтобы предотвратить физический доступ к оборудованию.
* Регулярное обновление программного обеспечения и патчи для закрытия уязвимостей и предотвращения эксплуатации известных угроз.
* Применение мер безопасности на уровне сети, таких как брандмауэры и виртуальные частные сети (VPN), для защиты передаваемых данных.
* Разработка механизмов обработки ошибок с уведомлением администраторов системы о возможных проблемах безопасности.

**5.6. Перспективы развития, модернизации системы**

* Интеграция системы с крипто биржами для автоматического переноса данных о имеющихся валют пользователей и возможности совершать транзакции с помощью нашей системы
* Добавления раздела с образовательными курсами связанными с крипто торговлей и экономики.

**6. Требования к видам обеспечения**

**6.1. Требования к лингвистическому обеспечению**

Разработка программного обеспечения должна вестись на высокоуровневом языке программирования.

Движок должен быть написан на языке Python. Использование другого языка снизит производительность системы.

Программный интерфейс приложения (API) может быть реализован средствами иных языков программирования, но Python является рекомендацией, поскольку благодаря ему есть возможность писать асинхронный код, который может значительно снизить требования к ресурсам сервера.

**6.2. Требования к программному обеспечению**

**Язык программирования и фреймворки:**

* Python 3.x: Использование последней стабильной версии языка Python.
* Django или Flask: Выбор фреймворка для разработки веб-приложения в зависимости от требований проекта.

**СУБД и Работа с Данными**:

* PostgreSQL или MySQL: Выбор между реляционными СУБД в зависимости от требований к структуре данных.
* SQLAlchemy: Использование библиотеки SQLAlchemy для работы с базой данных из Python.

**Коммуникация и API:**

* RESTful API: Реализация API для обмена данными между компонентами системы.
* Django REST framework (если используется Django): Расширение для Django для удобной работы с API.

**Веб-разработка:**

* HTML, CSS, JavaScript: Основные технологии для веб-разработки.
* Frontend-фреймворк (например, React, Vue.js): В случае необходимости интерактивного пользовательского интерфейса.

**Аутентификация и Авторизация:**

* Django Authentication System (если используется Django): Встроенная система аутентификации.
* OAuth: Поддержка стандарта для внешней аутентификации, если это требуется.

**Шифрование и Безопасность:**

* bcrypt: Использование алгоритма хеширования для защиты паролей.
* Django Security Middleware (если используется Django): Дополнительные меры безопасности для веб-приложения.

**Мониторинг и Журналирование:**

* Django Debug Toolbar (если используется Django): Инструмент для отладки и мониторинга.
* Логирование: Использование библиотек, таких как Python's logging, для записи событий и ошибок.

**Тестирование:**

* pytest: Фреймворк для написания и запуска тестов.
* Django Testing Framework (если используется Django): Инструменты для тестирования Django-приложений.

**Управление Зависимостями и Виртуальные Окружения:**

* pipenv: Инструмент для управления зависимостями и создания виртуальных окружений.

**Контейнеризация и Оркестрация:**

* Docker: для контейнеризации приложения.
* Docker Compose: для управления множеством контейнеров и их конфигурацией.
* Kubernetes (при необходимости): для автоматизации развертывания и управления контейнеризированными приложениями.

**Интеграция с Сторонними Сервисами:**

* Библиотеки для работы с API криптовалютных бирж (например, ccxt).
* Интеграция с внешними платежными системами, если предусмотрено выполнение финансовых транзакций.

**Заключение**

Проектирование системы Криптопортфолио представляет собой важный этап в создании инновационного и актуального инструмента для управления криптовалютными активами. Обладая высоким потенциалом для интеграции с современными технологиями и отвечая на растущий спрос на криптовалютные решения, система Криптопортфолио является перспективным проектом.

Разработанные требования к программному обеспечению учитывают современные стандарты безопасности, эффективность взаимодействия компонентов системы, а также гибкость в использовании. Выбор языка программирования Python и соответствующих инструментов обеспечивает удобство разработки, а также обеспечивает высокий уровень защиты информации и простоту поддержки.

Особое внимание уделено аспектам безопасности, аутентификации и шифрованию данных, что является критически важным в контексте работы с финансовой информацией. Также предусмотрены средства мониторинга и журналирования для оперативного обнаружения и реагирования на потенциальные угрозы.

Система Криптопортфолио призвана обеспечить пользователям удобство и безопасность при управлении и мониторинге их криптовалютных активов. Её разработка открывает перспективы для создания современного, инновационного и востребованного инструмента в динамичной среде криптовалютных рынков.