

Национальный исследовательский университет “Высшая школа экономики”.

Факультет компьютерных наук. Программная инженерия.

Проектирование архитектуры программных систем.

Домашнее задание №2 студента группы БПИ213 Абрамова Александра Сергеевича.

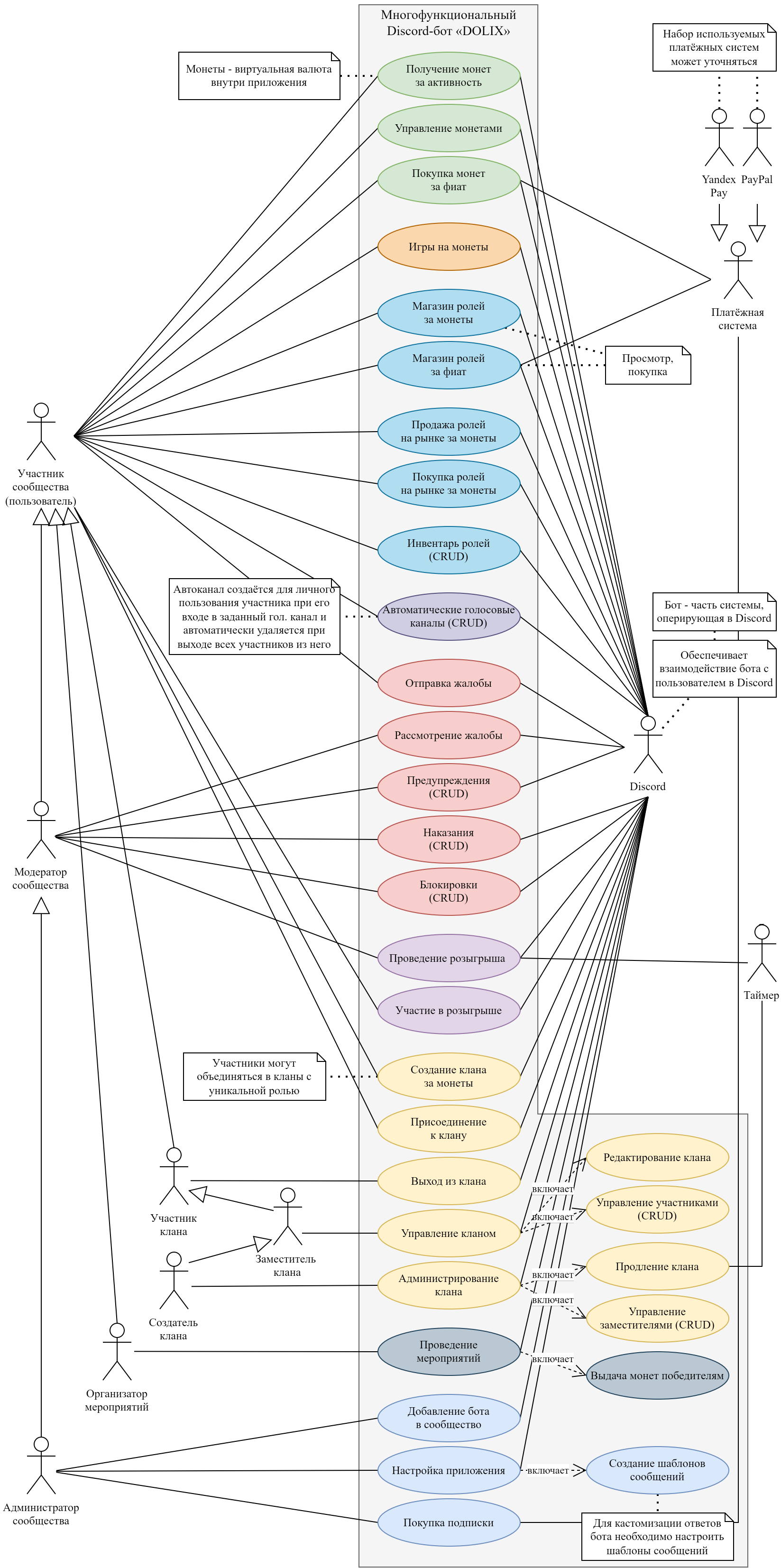
Задание выполнено по личному проекту - многофункциональный бот для платформы Discord.

СОДЕРЖАНИЕ

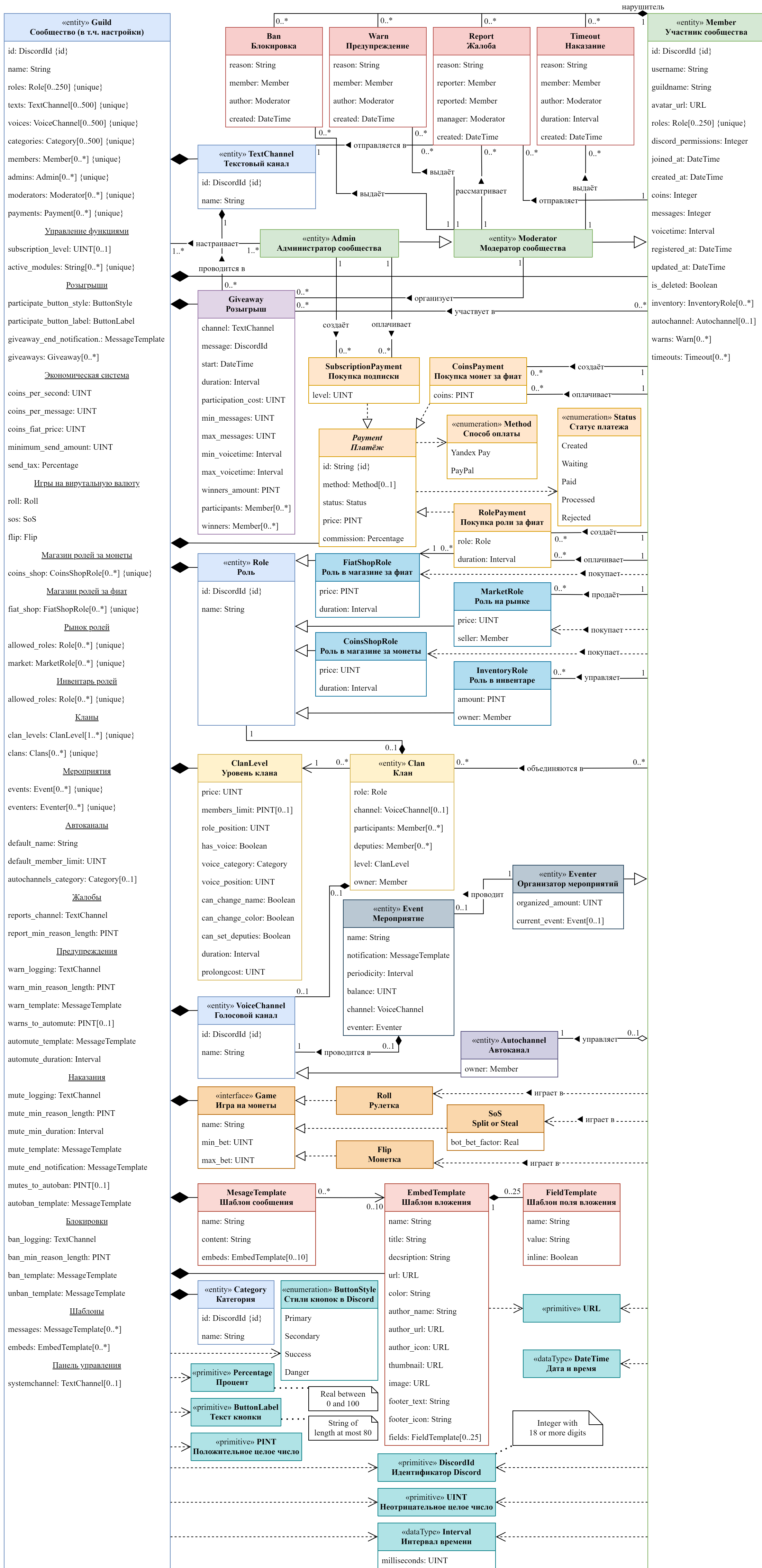
МОДЕЛЬ ПРЕЦЕДЕНТОВ.....	3
МОДЕЛЬ ПРЕДМЕТНОЙ ОБЛАСТИ.....	4
СПЕЦИФИКАЦИЯ ПРЕЦЕДЕНТА 1.....	5
СПЕЦИФИКАЦИЯ ПРЕЦЕДЕНТА 2.....	10
ДРУГИЕ ТРЕБОВАНИЯ.....	14
ПОДСИСТЕМА “САЙТ – АВТОРИЗАЦИЯ”.....	16
Диаграмма классов.....	16
Диаграмма последовательности – выбор языка.....	17
Диаграмма последовательности – выбор темы.....	18
Диаграмма последовательности – авторизация.....	19
Диаграмма последовательности – вход в аккаунт.....	20
ПОДСИСТЕМА “СЕРВИС ОПЛАТЫ”.....	21
Диаграмма классов.....	21
Диаграмма последовательности – создание платежа.....	22
Диаграмма последовательности – создание формы оплаты.....	23
Диаграмма последовательности – обработка платежа.....	24
ПОДСИСТЕМА “DISCORD – КЛИЕНТ”.....	25
Диаграмма классов.....	25
Диаграмма последовательности – приём взаимодействия.....	26
Диаграмма последовательности – отправка сообщения.....	27
ПОДСИСТЕМА “ПАНЕЛЬ УПРАВЛЕНИЯ” – ПРЕЦЕДЕНТ “НАСТРОЙКА	
ПРИЛОЖЕНИЯ”.....	28
Диаграмма классов.....	28
Диаграмма последовательности.....	29
ПАКЕТ “ПОКУПКА МОНЕТ” ПОДСИСТЕМЫ “ЭКОНОМИКА” – ПРЕЦЕДЕНТ	
“ПОКУПКА МОНЕТ ЗА ФИАТ”.....	30

Диаграмма классов.....	30
Диаграмма последовательности – обработка взаимодействия.....	31
Диаграмма последовательности – обработка платежа.....	32
ОБЩАЯ ДИАГРАММА КЛАССОВ.....	33
ДИАГРАММА ПАКЕТОВ.....	34

МОДЕЛЬ ПРЕЦЕДЕНТОВ



МОДЕЛЬ ПРЕДМЕТНОЙ ОБЛАСТИ



СПЕЦИФИКАЦИЯ ПРЕЦЕДЕНТА 1

Название ◇	Настройка приложения
Аннотация ◇	<p>Одной из ключевых конкурентных особенностей продукта является возможность гибкой настройки всех функций под потребности конкретного сообщества, в связи с чем важно уделить особенное внимание разработке удобной для пользователей панели управления, которая бы предоставила возможность детальной настройки приложения при простоте и понятности своего интерфейса и принципов работы. Хотя настройку всех существующих параметров можно реализовать и на поздних этапах проекта, панель управления для настройки минимального набора основных параметров должна присутствовать уже на самых ранних этапах жизненного цикла.</p>
Автор ◇	Абрамов Александр Сергеевич, программный инженер продукта, студент группы БПИ213
Рамки применения	Ко всей системе
Значимость	Ключевая задача
Приоритет	Высокий
Статус реализации	Проработан, не реализован
Первичный актёр ◇	Администратор сообщества
Вторичные актёры и их требования	<p>Discord:</p> <ol style="list-style-type: none"> 1. Авторизация пользователей - https://discord.com/developers/docs/topics/oauth2 2. Отправка лога изменений - https://discord.com/developers/docs/resources/channel#create-message
Базовый поток ◇	<p>А. Пользователь заходит на сайт.</p> <p>В. Система формирует пользовательский интерфейс с общей информацией о продукте и кнопкой авторизации через Discord.</p> <p>С. Пользователь авторизуется через Discord по протоколу OAuth2.</p> <p>Д. Система формирует пользовательский интерфейс с меню выбора сообщества для настройки.</p> <p>Е. Пользователь выбирает сообщество, на котором хочет произвести</p>

	<p>настройку бота.</p> <p>F. Система убеждается, что пользователь действительно является администратором выбранного сообщества.</p> <p>G. Система формирует пользовательский интерфейс с описанием общих принципов использования панели управления и с меню навигации по ней.</p> <p>H. Пользователь переходит на страницу с настройками, которые хочет отредактировать.</p> <p>I. Система генерирует пользовательский интерфейс с полями ввода, необходимыми для настройки параметров соответствующих функций, и заполняет их текущими значениями.</p> <p>J. Пользователь редактирует значения всех или некоторых элементов.</p> <p>K. Пользователь нажимает кнопку сохранения изменений.</p> <p>L. Система запрашивает подтверждение, что пользователь действительно хочет сохранить новые значения.</p> <p>M. Пользователь подтверждает намерение сохранить изменения нажатием соответствующей кнопки.</p> <p>N. Система проверяет корректность введенных значений. Значения корректны.</p> <p>O. Система производит сохранение новых значений.</p> <p>P. Система отправляет лог изменений в системный канал сообщества, если таковой задан, посредством запроса к API Discord.</p> <p>Q. Система сообщает пользователю об успешном выполнении операции.</p> <p>R. Пользователь завершает настройку приложения.</p>
Альтернативные потоки ◇	<p>Альтернативный поток AT1. Условие начала: пользователь хочет прекратить работу.</p> <p>A. Пользователь покидает страницы панели управления.</p> <p>Альтернативный поток AT2. Условие начала: пользователь нажимает кнопку изменения стиля интерфейса.</p> <p>A. Система формирует меню выбора стиля интерфейса.</p> <p>B. Пользователь выбирает желаемый стиль.</p> <p>C. Система вносит соответствующие корректировки в интерфейс.</p> <p>D. Продолжение основного потока.</p> <p>Альтернативный поток AT3. Условие начала: пользователь нажимает кнопку изменения языка интерфейса.</p>

- A. Система формирует меню выбора языка интерфейса.
- B. Пользователь выбирает желаемый язык.
- C. Система формирует интерфейс на выбранном языке.
- D. Продолжение основного потока.

Альтернативный поток C1. Условие начала: на шаге C нет связи с Discord.

- A. Система переходит в аварийный режим и формирует соответствующий пользовательский интерфейс с сообщением об ошибке.

Альтернативный поток F1. Условие начала: пользователь не является администратором выбранного сообщества.

- A. Система формирует пользовательский интерфейс с сообщением об ошибке.
- B. Переход на шаг D основного потока.

Альтернативный поток G1. Условие начала: пользователь хочет настроить шаблоны сообщений, доступные для использования в настройках.

- A. Пользователь переходит на страницу управления шаблонами сообщений.
- B. Переход к прецеденту “Создание шаблонов сообщений”.

Альтернативный поток G2. Условие начала: пользователь хочет изменить сообщество, на котором производит настройку бота.

- A. Пользователь нажимает кнопку изменения сообщества.
- B. Переход на шаг D основного потока.

Альтернативный поток I1. Условие начала: пользователь хочет перейти на другую страницу панели управления.

- A. Переход на шаг H основного потока.

Альтернативный поток I2. Условие начала: пользователь хочет изменить сообщество, на котором он производит настройку бота.

- A. Пользователь нажимает кнопку изменения сообщества.
- B. Переход на шаг D основного потока.

Альтернативный поток J1. Условие начала: на шаге J пользователь редактирует поле ввода строки добавления элементов в таблицу.

- A. Система создаёт кнопку удаления отредактированной строки.
- B. Система создаёт новую строку с пустыми полями ввода для дальнейшего добавления элементов в таблицу.

	<p>С. Переход на шаг J основного потока.</p> <p>Альтернативный поток J2. Условие начала: на шаге J пользователь нажимает кнопку удаления строки таблицы.</p> <p>А. Система удаляет все элементы, связанные с соответствующей строкой.</p> <p>В. Переход на шаг J основного потока.</p> <p>Альтернативный поток L1. Условие начала: пользователь отменяет сохранение изменений нажатием соответствующей кнопки.</p> <p>А. Переход на шаг J основного потока.</p> <p>Альтернативный поток N1. Условие начала: на шаге N введённые значения некорректны.</p> <p>А. Система формирует пользовательский интерфейс с сообщением об ошибке.</p> <p>В. Переход на шаг J основного потока.</p> <p>Альтернативный поток Q1. Условие начала: пользователь хочет продолжить работу.</p> <p>А. Переход на шаг I основного потока.</p>
Предусловие ◇	Пользователь является администратором сообщества, на котором хочет произвести настройку приложения, и бот добавлен на соответствующий сервер Discord.
Постусловия ◇	<ol style="list-style-type: none"> 1. В сообществе установлены новые настройки приложения, удовлетворяющие всем требованиям и ограничениям. 2. Лог изменений отправлен в системный канал сообщества, если он задан.
Специальные бизнес-требования	<ol style="list-style-type: none"> 1. На шаге С основного потока система не должна запрашивать больше доступа к аккаунту пользователя, чем ей необходимо для работы. 2. Система должна обеспечивать высокий уровень надёжности. 3. Интерфейс должен быть прост, удобен и понятен. 4. Интерфейс должен быть реализован в двух стилях (светлый, тёмный) на двух языках (русский, английский). Добавление стилей и языков должно быть возможно на любом этапе жизненного цикла проекта без изменения архитектуры системы. 5. Внесённые пользователем на шаге J основного потока изменения не

	должны “пропадать” при переходе к альтернативным потокам L1 и N1.
Связь с атрибутами качества	<ol style="list-style-type: none"> 1. Система не должна допускать несанкционированный доступ к панели управления. 2. Интерфейс должен быть понятен конечному пользователю, обладающему базовыми навыками обращения с компьютером. 3. Интерфейс должен быть снабжен достаточным для понимания типов и ограничений ожидаемых значений, а также для интерпретации возникающих ошибок количеством документации и подсказок.
Проектные ограничения	Панель управления для настройки минимального набора основных параметров должна быть реализована на самых ранних этапах жизненного цикла. Поддержка других существующих параметров может быть реализована позже.
Список технологий реализации	<p>Серверная часть:</p> <ol style="list-style-type: none"> 1. Язык программирования Node.JS; 2. Фреймворк Fastify; 3. База данных PostgreSQL, библиотека pg; 4. Хранилище данных Redis, библиотека ioredis; 5. Шаблонизатор веб-страниц EJS; <p>Клиентская часть:</p> <ol style="list-style-type: none"> 1. Язык разметки HTML; 2. Язык стилей CSS; 3. Язык программирования JavaScript; <p>Клиент-серверное и межсервисное взаимодействие:</p> <ol style="list-style-type: none"> 1. Протокол HTTPS; 2. Протокол OAuth2;
Открытые проблемы	Отсутствуют

СПЕЦИФИКАЦИЯ ПРЕЦЕДЕНТА 2

Название ◇	Покупка монет за фиат
Аннотация ◇	Одним из основных способов монетизации приложения будет взимание комиссии с продажи виртуальной валюты сообществами. Для этого необходимо реализовать удобную, “прозрачную” автоматизированную систему, в рамках которой пользователи смогут покупать монеты за фиатную валюту. При этом большая часть средств перечисляется на баланс сообщества, которым затем можно оплатить подписку на премиальные функции.
Автор ◇	Абрамов Александр Сергеевич, программный инженер продукта, студент группы БПИ213
Рамки применения	Ко всей системе
Значимость	Ключевая задача
Приоритет	Высокий
Статус реализации	Проработан, не реализован
Первичный актёр ◇	Участник сообщества
Вторичные актёры и их требования	<ol style="list-style-type: none"> 1. Discord - средство взаимодействия с пользователем, https://discord.com/developers/docs 2. Платёжные системы - обработка платежей. <ol style="list-style-type: none"> a. Yandex Pay - https://pay.yandex.ru/business b. PayPal - https://www.paypal.com/ru/webapps/mpp/partners-and-developers
Базовый поток ◇	<ol style="list-style-type: none"> A. Пользователь использует команду покупки монет в любом текстовом канале сервера, указывая количество монет к покупке. B. Бот вычисляет стоимость указанного количества монет с учётом настроек сообщества. C. Бот создаёт платёж на необходимую сумму и получает его уникальный номер. D. Бот отвечает на команду сообщением, в котором подтверждается информация о платеже, с кнопкой, которая направляет пользователя на страницу платежа на сайте.

	<p>Е. Пользователь переходит по кнопке.</p> <p>Г. Система формирует пользовательский интерфейс страницы с данными платежа.</p> <p>Г. Пользователь проверяет, что вся информация верна.</p> <p>Н. Пользователь выбирает способ оплаты.</p> <p>И. Система создаёт платёжную форму с помощью API выбранного способа оплаты и перенаправляет пользователя на неё.</p> <p>Ж. Пользователь переходит на страницу оплаты.</p> <p>К. Пользователь осуществляет оплату в стороннем сервисе.</p> <p>Л. Система проверяет статус платежа с помощью API выбранного способа оплаты. Оплата произведена успешно.</p> <p>М. Система начисляет монеты на баланс пользователя.</p> <p>Н. Система увеличивает баланс сообщества на долю размера платежа, задаваемую в файле конфигурации.</p> <p>О. Система устанавливает платежу статус “Processed”.</p> <p>Р. Система отправляет уведомление о покупке в системный канал сообщества, если таковой задан, посредством запроса к API Discord.</p>
Альтернативные потоки ◇	<p>Альтернативный поток G1. Условие начала: на шаге Г пользователь обнаружил ошибку в информации о платеже.</p> <p>А. Переход на шаг А основного потока.</p> <p>Альтернативный поток И1. Условие начала: на шаге И нет связи с платёжной системой.</p> <p>А. Система переходит в аварийный режим и формирует соответствующий пользовательский интерфейс с сообщением об ошибке.</p> <p>Альтернативный поток Л1. Условие начала: на шаге Л получен статус, указывающий, что оплата еще не была произведена, или нет связи с платёжной системой.</p> <p>А. Перейти на шаг Л.</p> <p>Альтернативный поток Л2. Условие начала: на шаге Л получен статус, указывающий, что оплата не была произведена успешно (платёж отклонён).</p> <p>А. Система устанавливает платежу статус “Rejected”.</p>
Предусловие ◇	Бот добавлен в сообщество, функция экономики активна, и пользователь

	может использовать команду покупки монет на необходимом сервере.
Постусловия ◇	<ol style="list-style-type: none"> 1. На виртуальный счёт пользователя начислено приобретённое количество монет. 2. На баланс сообщества начислено соответствующее количество средств. 3. Уведомление о покупке отправлено в системный канал сообщества, если он задан.
Специальные бизнес-требования	<ol style="list-style-type: none"> 1. Система не должна сохранять информацию, которая не является необходимой для её непосредственной работы, включая персональные данные пользователей и реквизиты использованных способов оплаты. 2. Система должна обеспечивать высокий уровень надёжности. 3. Интерфейс страницы платежа должен быть прост, удобен и понятен. 4. Для повышения доверия пользователей к системе обработка платежа после его оплаты должна производиться быстро.
Связь с атрибутами качества	<ol style="list-style-type: none"> 1. Система не должна допускать “накрутку” монет в обход оплаты. 2. Интерфейс страницы платежа должен быть понятен конечному пользователю, обладающему базовыми навыками обращения с компьютером. 3. Обработка платежа - начисление монет на баланс пользователя - при отсутствии технических неполадок сервера или платёжной системы не должно занимать более 5 минут после оплаты счёта пользователем на шаге К основного потока.
Проектные ограничения	На начальном этапе проекта должна быть реализована поддержка только платёжных систем Yandex Pay и PayPal. В будущем список может расширяться; добавление новых сервисов приёма платежей не должно требовать изменения архитектуры системы.
Список технологий реализации	<p>Бот (Discord-клиент):</p> <ol style="list-style-type: none"> 1. Язык программирования Node.JS; 2. Библиотека Discord.JS; <p>Серверная часть:</p> <ol style="list-style-type: none"> 1. Язык программирования Node.JS; 2. Фреймворк Fastify; 3. База данных PostgreSQL, библиотека pg;

	<p>4. Шаблонизатор веб-страниц EJS;</p> <p>Клиентская часть:</p> <ol style="list-style-type: none"> 1. Язык разметки HTML; 2. Язык стилей CSS; 3. Язык программирования JavaScript; <p>Клиент-серверное и межсервисное взаимодействие:</p> <ol style="list-style-type: none"> 1. Протокол HTTPS;
Открытые проблемы	Отсутствуют

ДРУГИЕ ТРЕБОВАНИЯ

1. Система должна поддерживать добавление новых модулей (настроек и логики в Discord-клиенте с, возможно, необходимостью хранения и обработки каких-либо данных) без значительных изменений архитектуры в целом.
2. Все параметры конфигурации приложения должны быть размещены в одном или нескольких файлах, явно отделённых от исходного кода.
3. Интерфейс должен корректно работать во всех современных браузерах:
 - a. Google Chrome версии 119 или выше;
 - b. Microsoft Edge версии 119 или выше;
 - c. Mozilla Firefox версии 121 или выше;
 - d. Opera версии 106 или выше;
 - e. Safari версии 16.4 или выше;
4. Требования к надёжности:
 - a. Приложение не должно аварийно завершаться при любом наборе входных данных. В случае отказа программы, не связанного с техническими неполадками сервера, время восстановления системы не должно превышать 30 минут.
 - b. Программа должна обеспечивать проверку корректности полученных от пользователей данных.
 - c. Система не должна допускать несанкционированный доступ к функциям и командам.
 - d. Система должна корректно обрабатывать лимиты запросов к API Discord: задерживать или отклонять запросы, которые заведомо превысят установленные ограничения.
5. Минимальные требования к серверному оборудованию для корректной работы приложения при штатной нагрузке в 1000 активных пользователей:
 - a. 2 виртуальных ядра процессора Intel Xeon E5 2620 или AMD EPYC 7402P;
 - b. 4 ГБ оперативной памяти;
 - c. 8 ГБ SSD хранилища;
 - d. Скорость интернет-соединения 500 Мбит/с;
 - e. Операционная система Linux Debian 12 с возможностью запуска Node.JS приложений на портах 80 и 443;
 - f. Наличие безопасного доступа к серверу и возможности загрузки на него необходимых файлов;
6. Общие требования к клиентскому оборудованию для корректной работы приложения:
 - a. Устройство, поддерживающее последнюю версию установленной операционной системы.
 - b. Устройство, соответствующее требованиям к клиентскому оборудованию приложения Discord.
 - c. Возможность использования одного из поддерживаемых браузеров (см. п. 3).
 - d. Свободный доступ в интернет со скоростью соединения не менее 5 Мбит/с.

7. Требования к временным характеристикам:

- а. При штатной нагрузке на устройстве, соответствующем требованиям к серверному оборудованию (см. п. 5), система должна отвечать на любой запрос не более, чем за 0.5 секунды без учёта затрат на взаимодействие со сторонними сервисами.
- б. В штатном режиме обработка ответа на любой запрос и его отображение на экране не должны занимать более 3-х секунд на устройстве, отвечающем требованиям к клиентскому оборудованию (см. п. 6).

ПОДСИСТЕМА “САЙТ – АВТОРИЗАЦИЯ”

Диаграмма классов

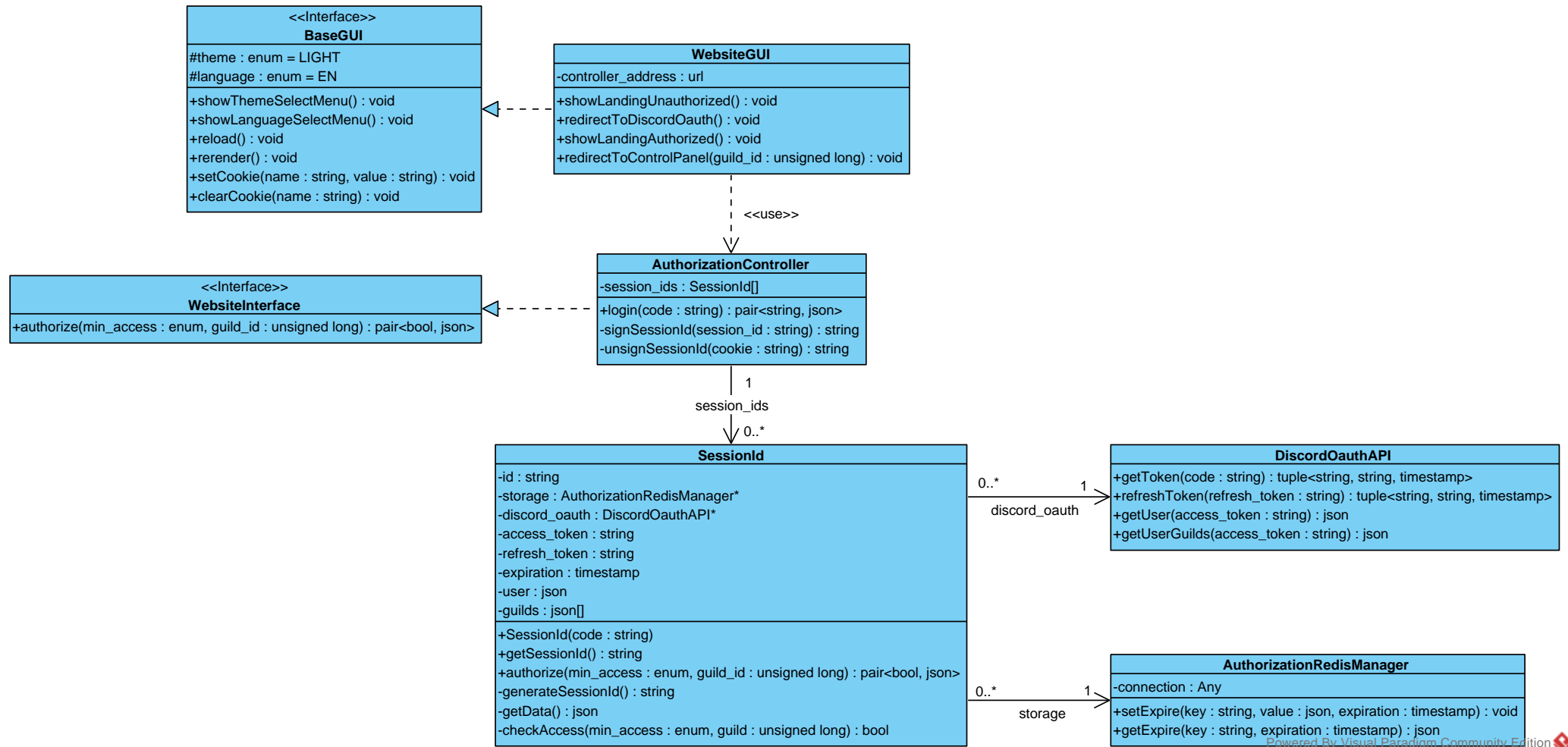


Диаграмма последовательности – выбор языка

sd [Choose language]

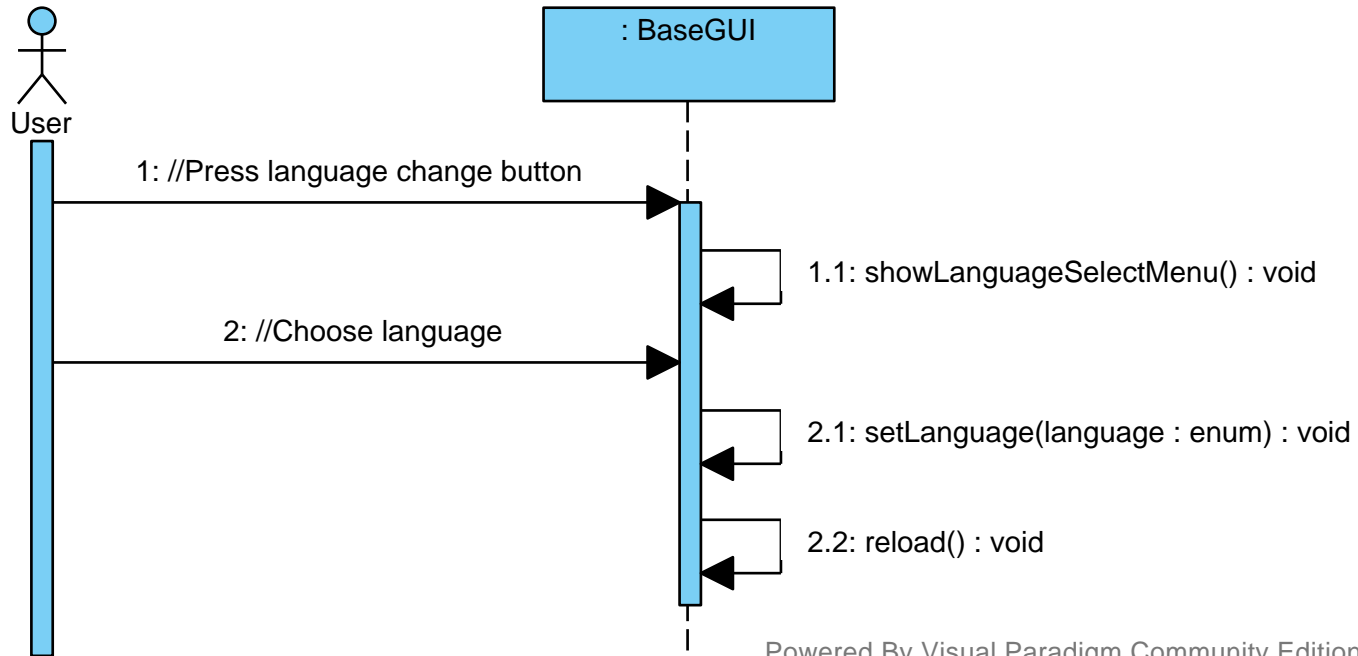


Диаграмма последовательности – выбор темы

sd [Choose theme]

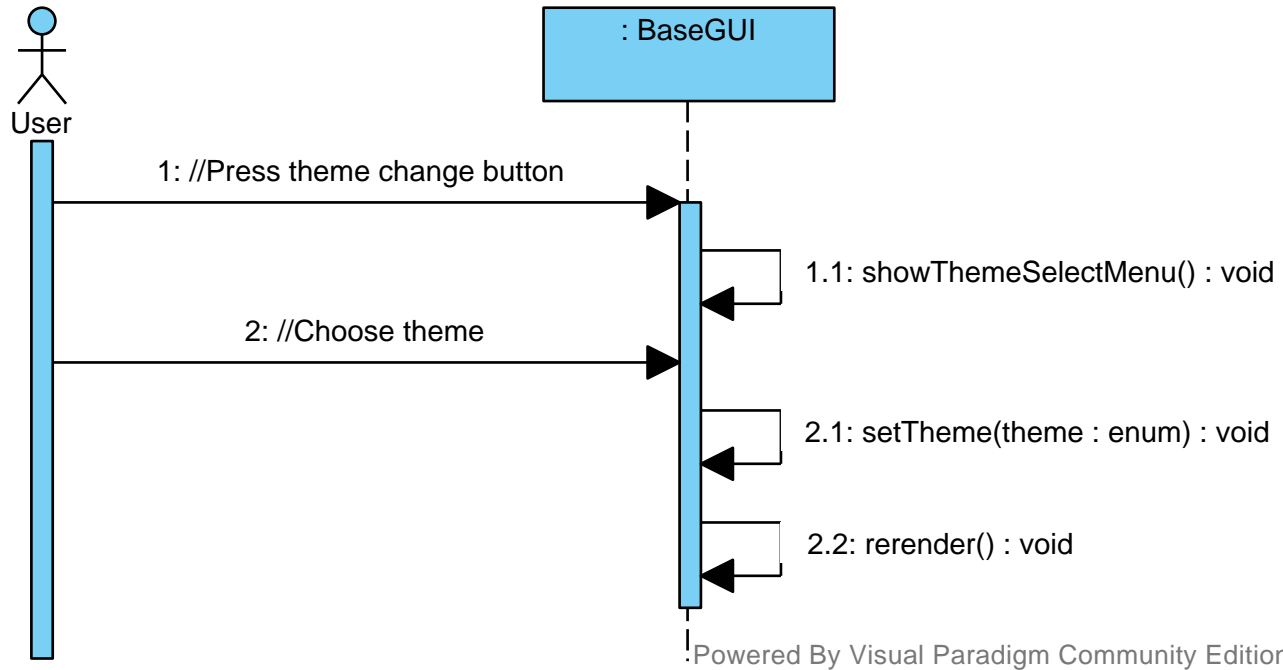


Диаграмма последовательности – авторизация

sd [Authorization]

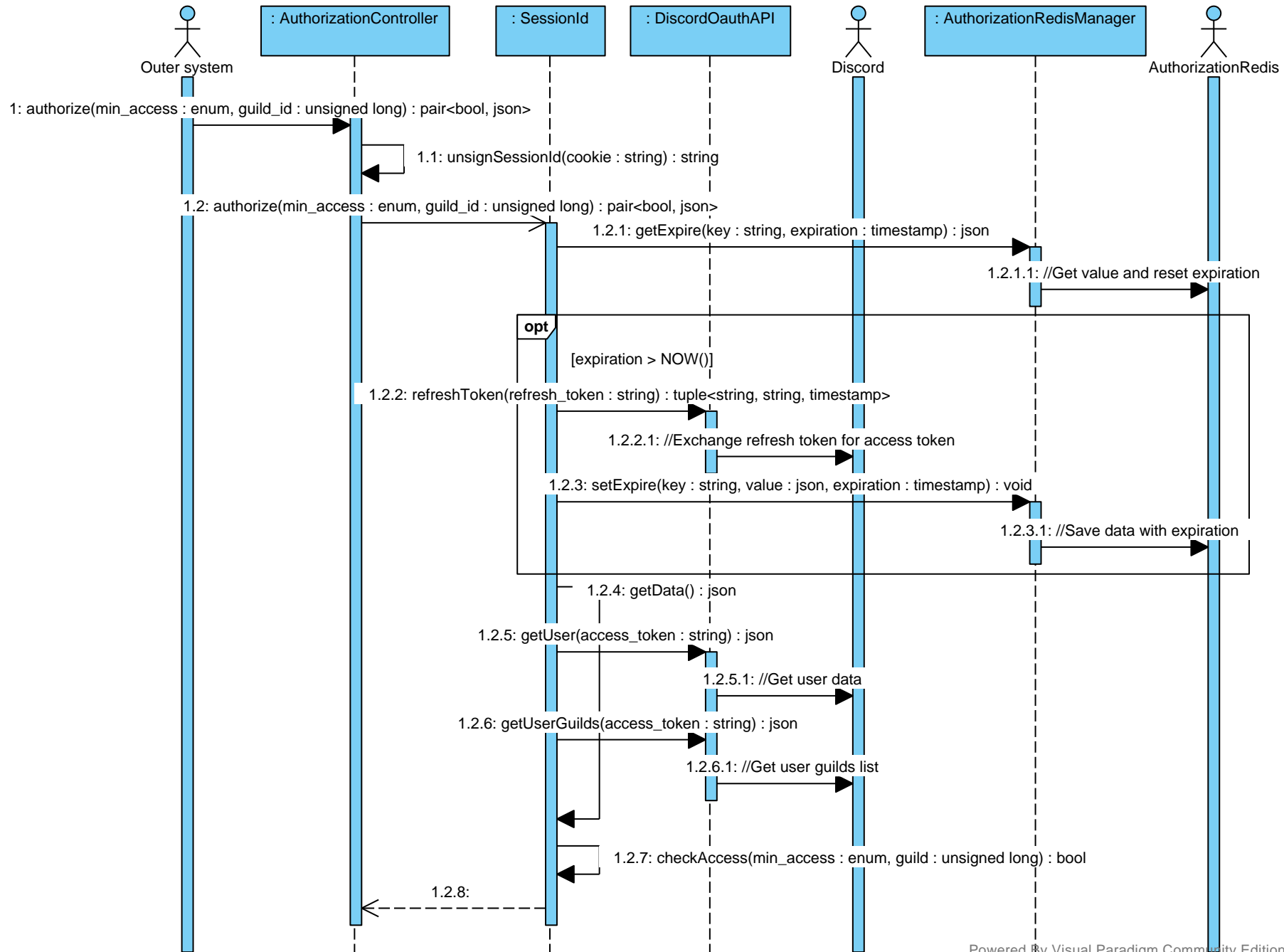
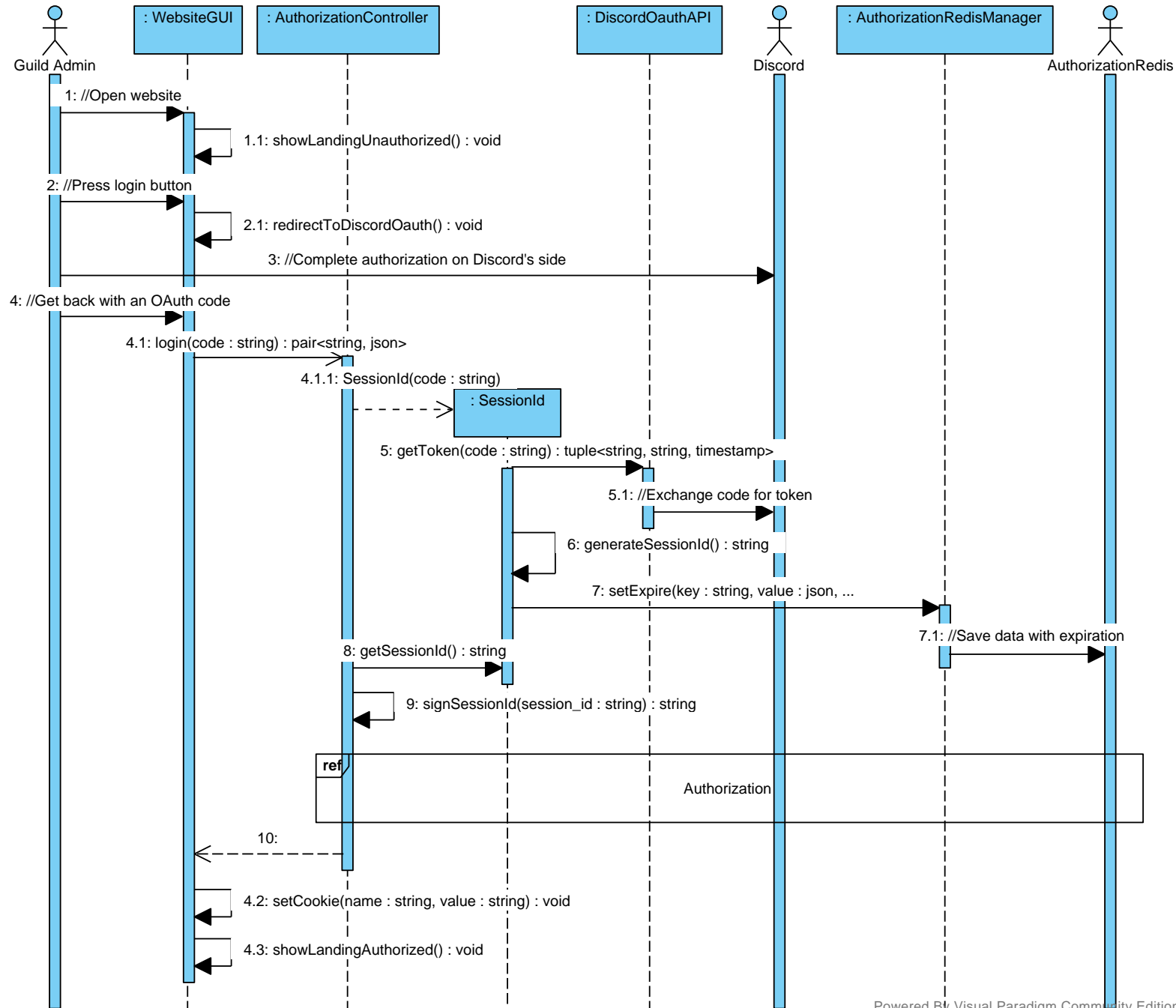


Диаграмма последовательности – вход в аккаунт

sd [Login]



ПОДСИСТЕМА “СЕРВИС ОПЛАТЫ”

Диаграмма классов

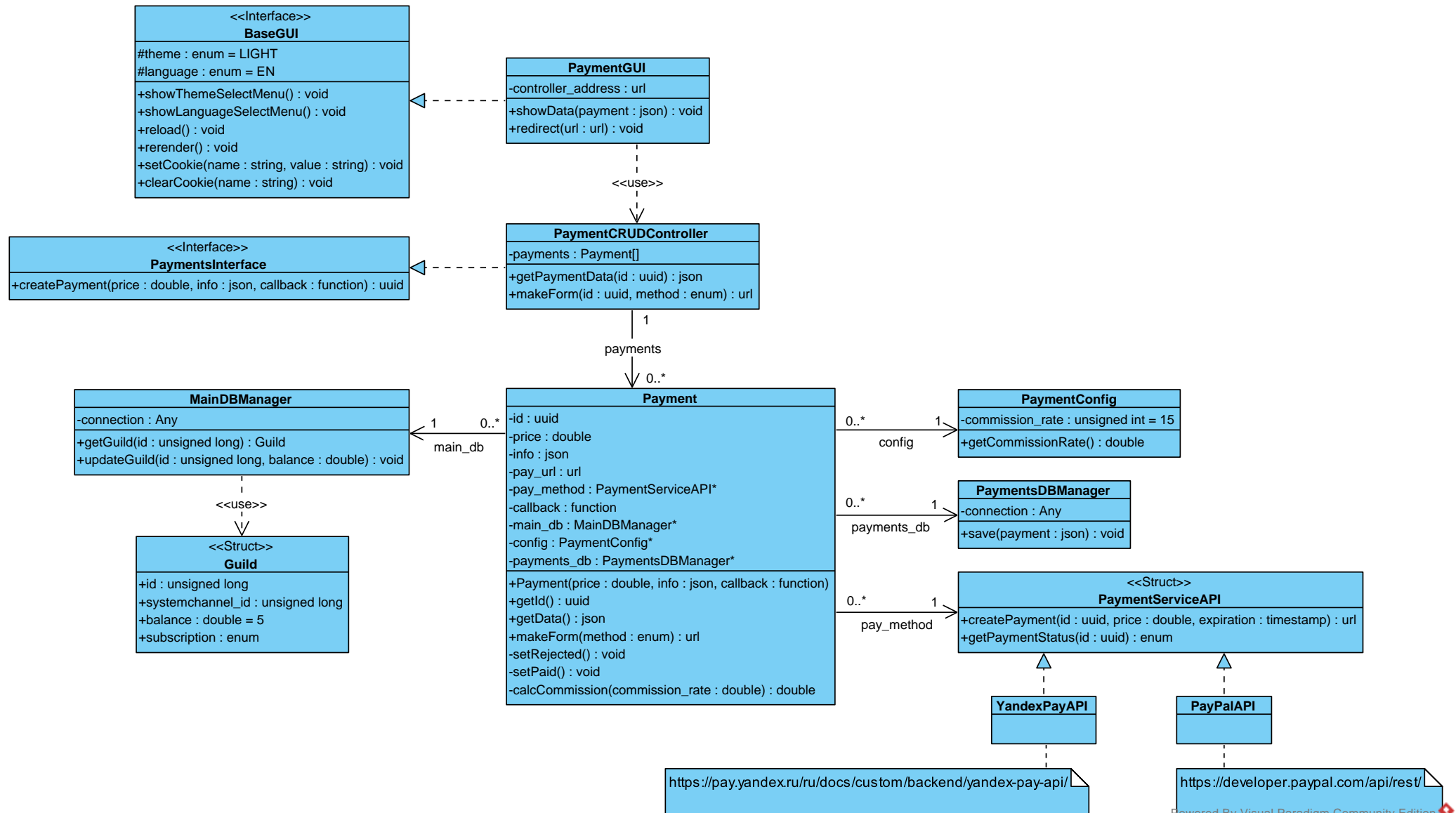


Диаграмма последовательности – создание платежа

sd [CreatePayment]

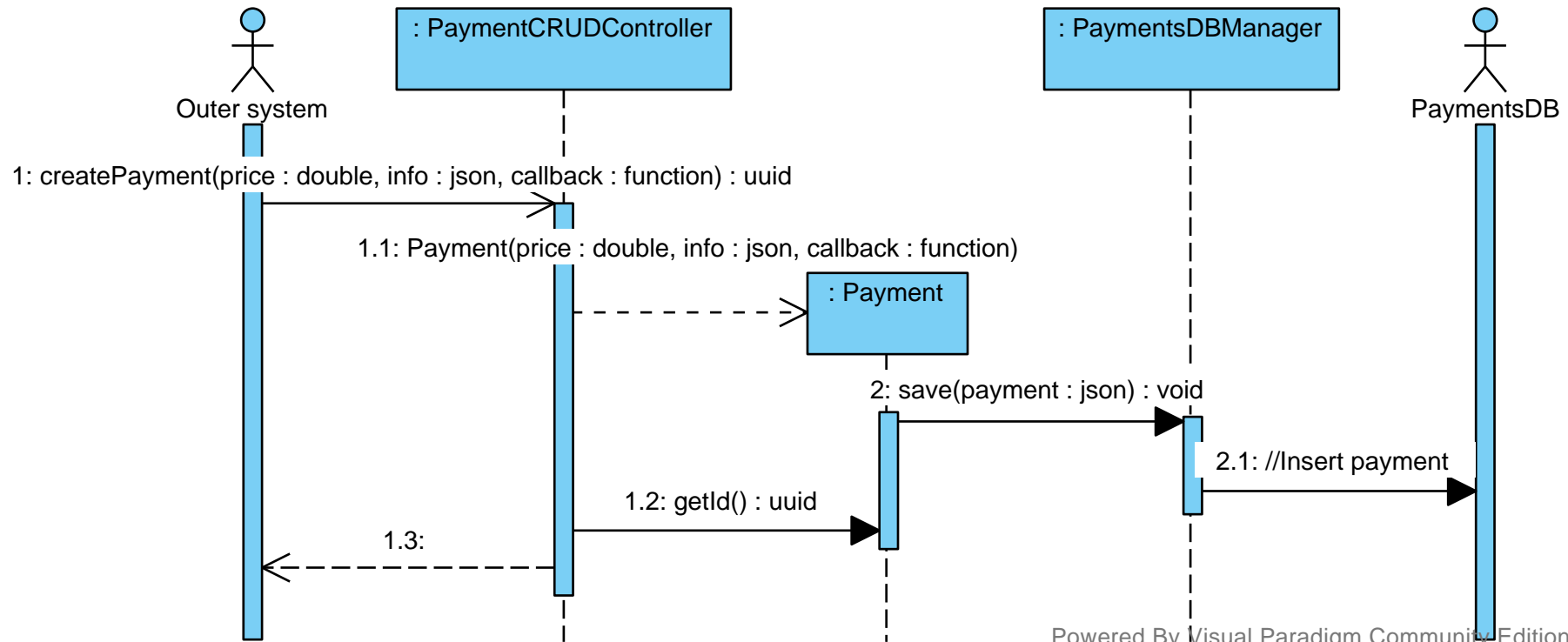


Диаграмма последовательности – создание формы оплаты

sd [MakePaymentForm]

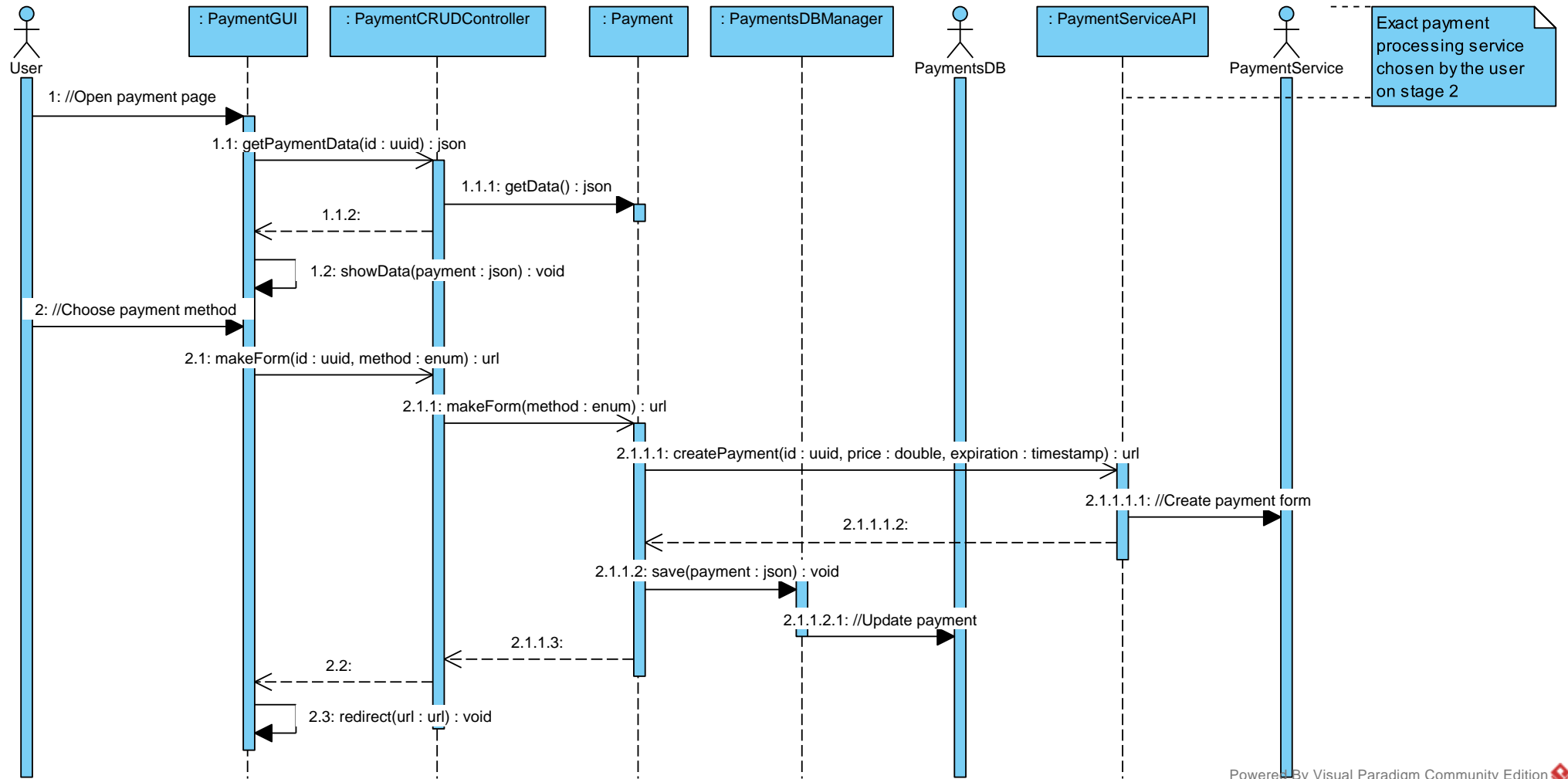
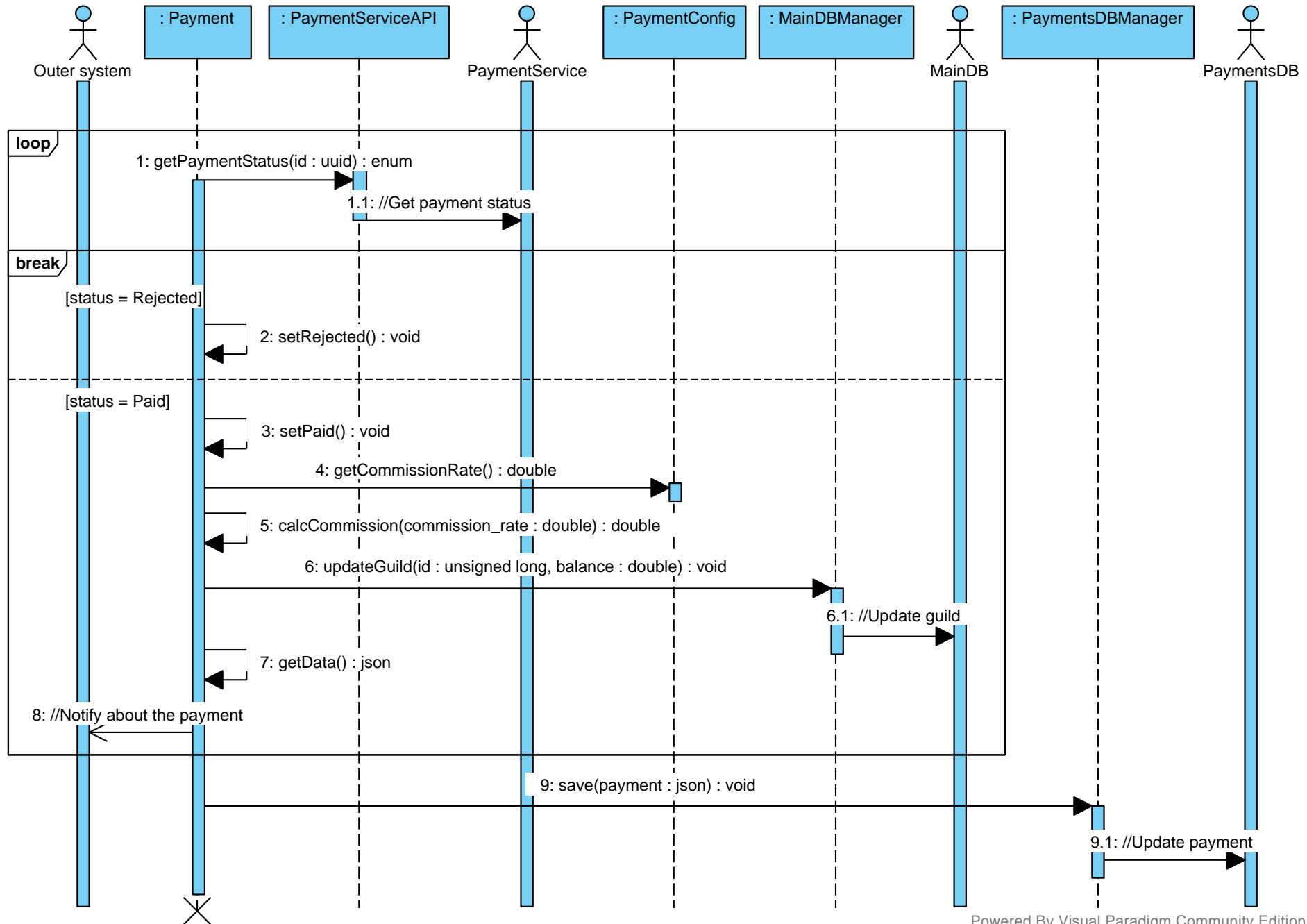


Диаграмма последовательности – обработка платежа

sd [ProcessPayment]



ПОДСИСТЕМА “DISCORD – КЛИЕНТ”

Диаграмма классов

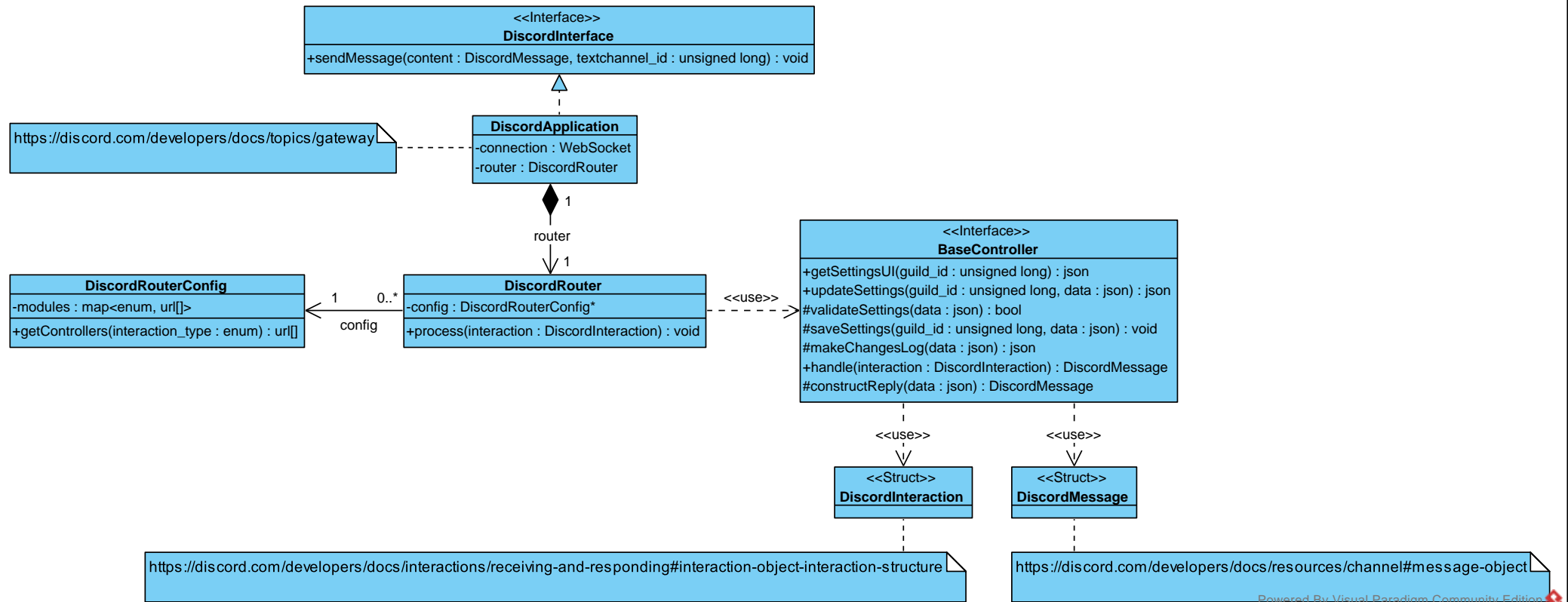


Диаграмма последовательности – приём взаимодействия

sd [ReceiveInteraction]

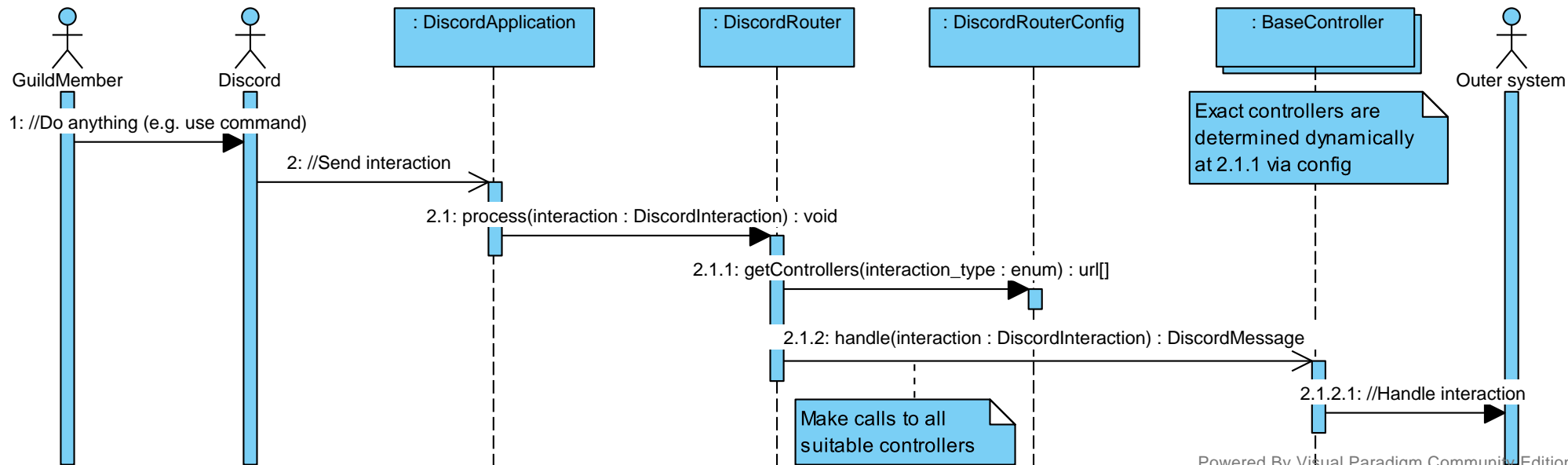
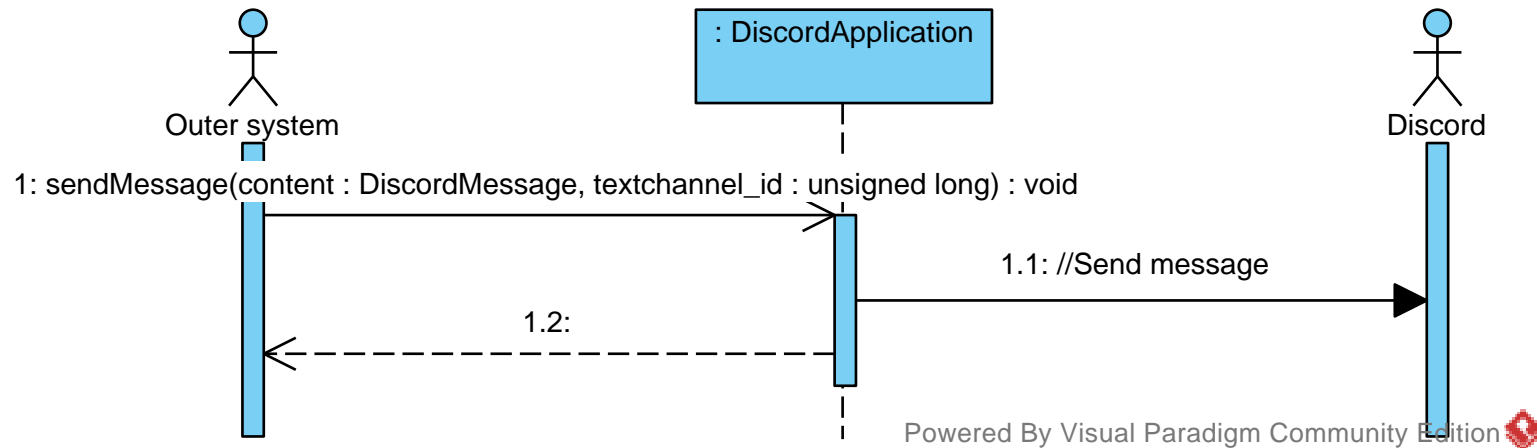


Диаграмма последовательности – отправка сообщения

sd [SendMessage]



**ПОДСИСТЕМА “ПАНЕЛЬ УПРАВЛЕНИЯ” – ПРЕЦЕДЕНТ “НАСТРОЙКА
ПРИЛОЖЕНИЯ”**

Диаграмма классов

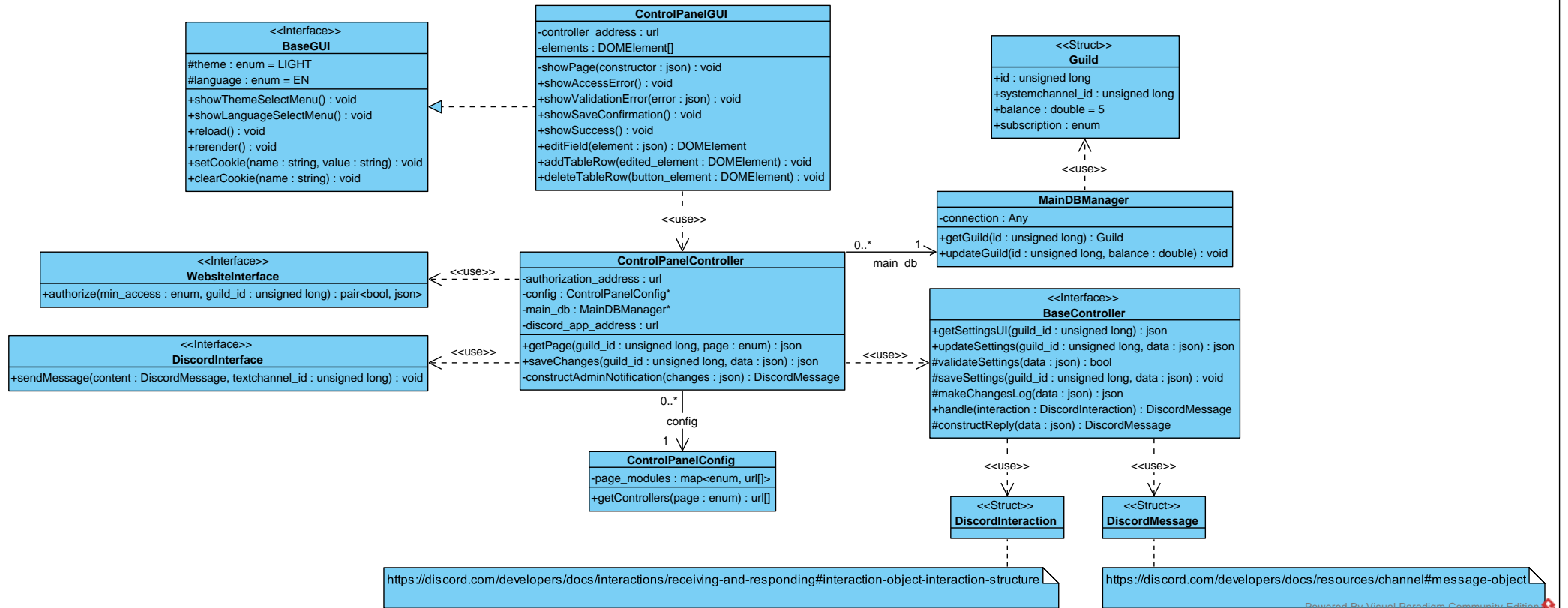
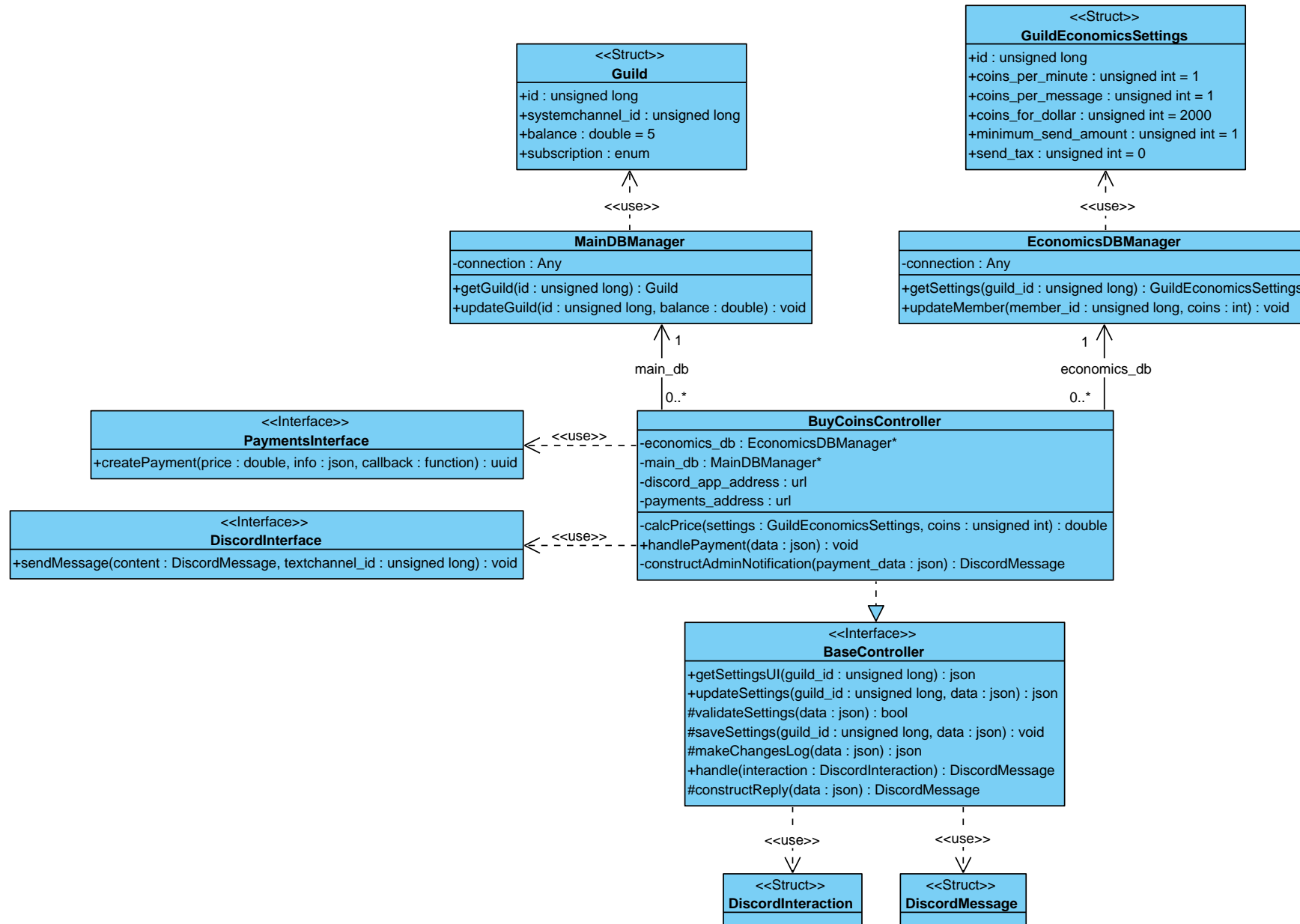


Диаграмма последовательности

**ПАКЕТ “ПОКУПКА МОНЕТ” ПОДСИСТЕМЫ “ЭКОНОМИКА” – ПРЕЦЕДЕНТ
“ПОКУПКА МОНЕТ ЗА ФИАТ”**

Диаграмма классов



<https://discord.com/developers/docs/interactions/receiving-and-responding#interaction-object-interaction-structure>

<https://discord.com/developers/docs/resources/channel#message-object>

Диаграмма последовательности – обработка взаимодействия

sd [HandleInteraction]

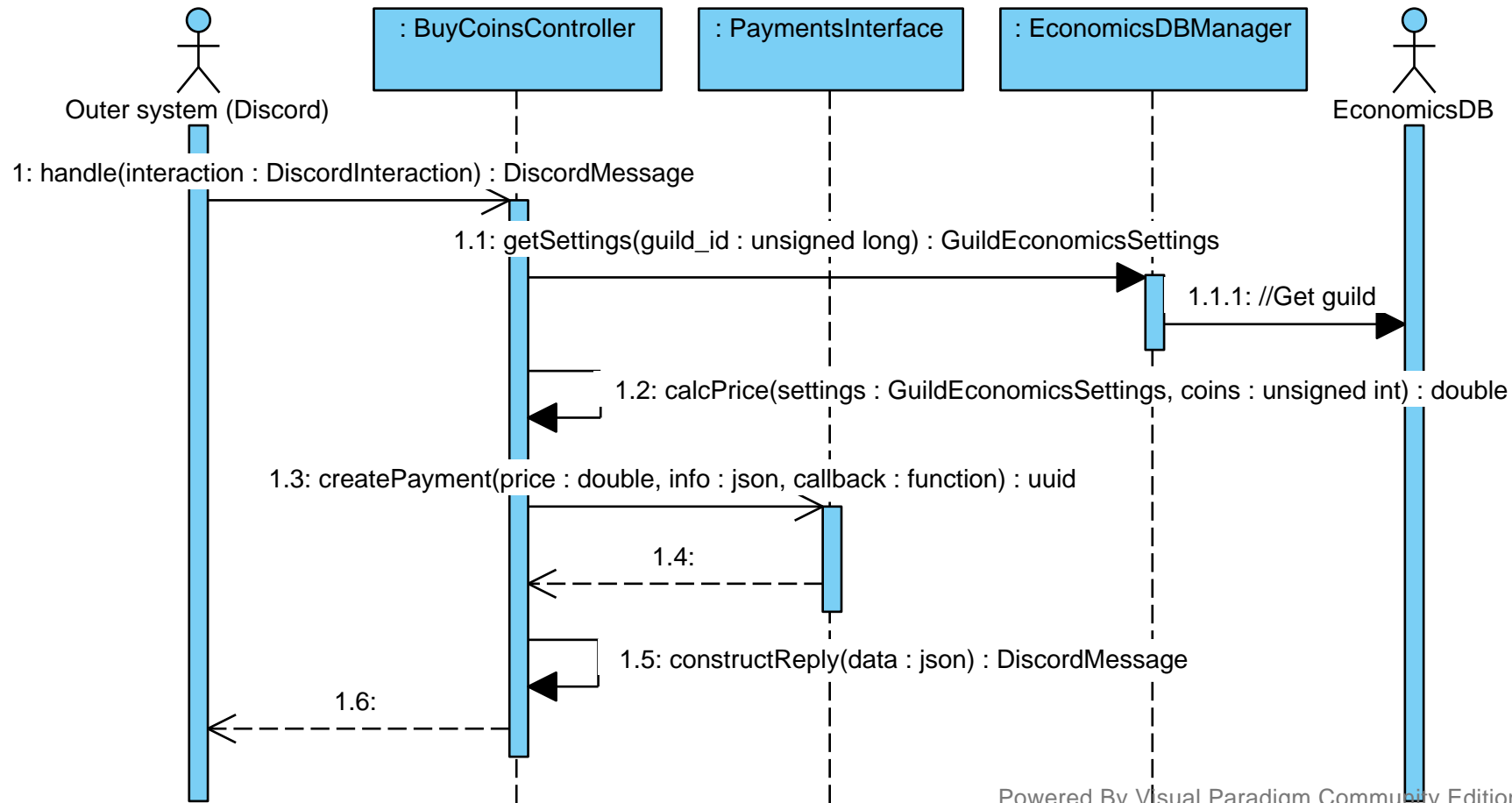
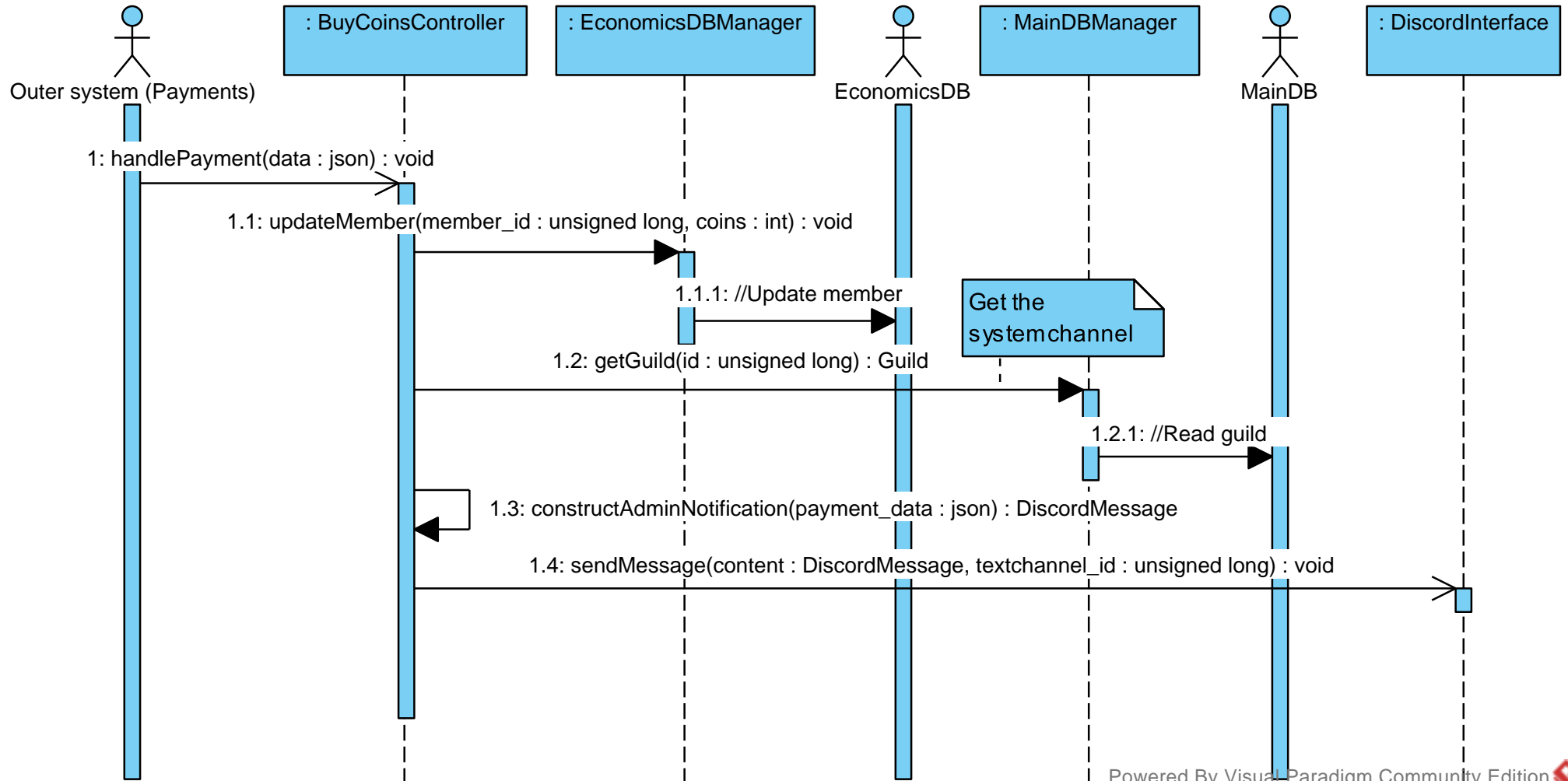


Диаграмма последовательности – обработка платежа

sd [HandlePayment]



ОБЩАЯ ДИАГРАММА КЛАССОВ

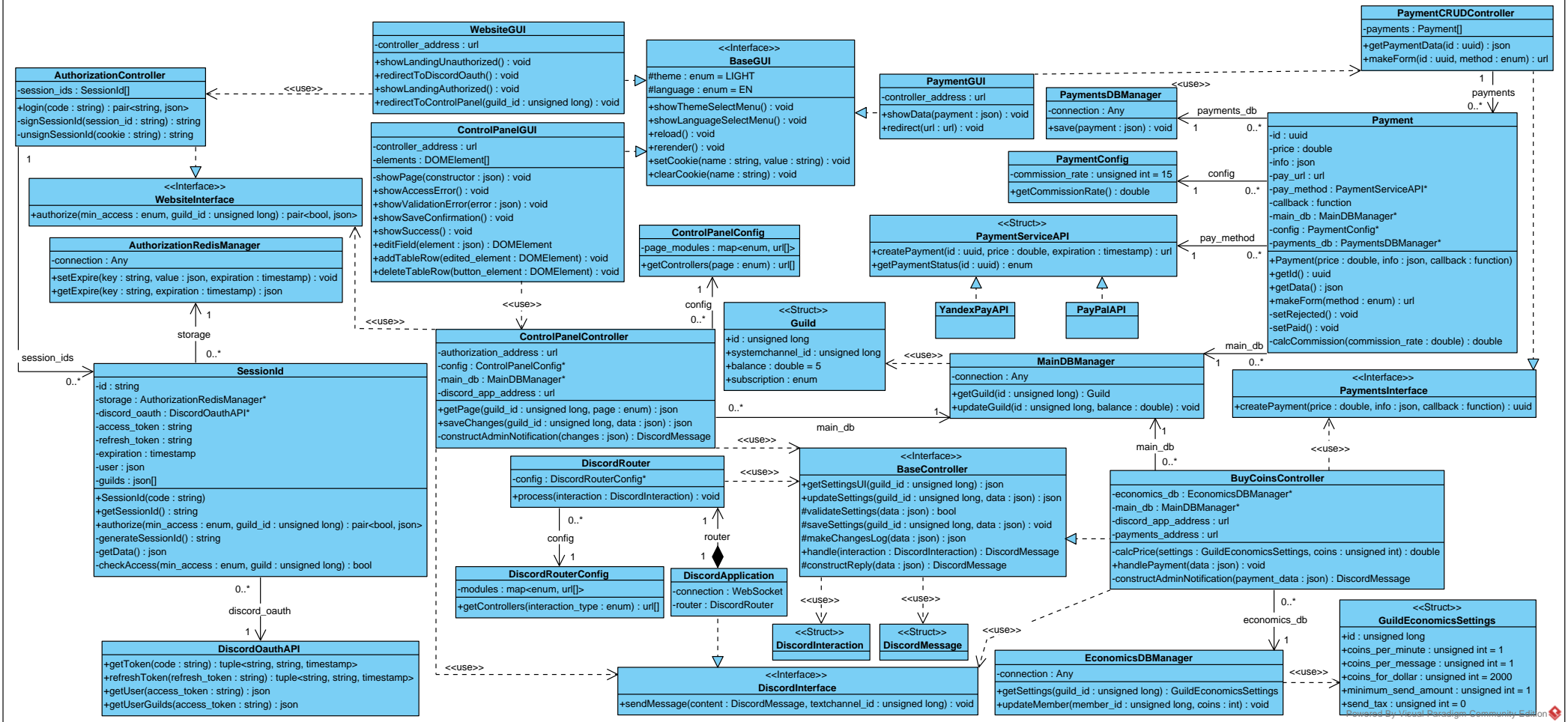


ДИАГРАММА ПАКЕТОВ

