

ТРЕК: «ПРИКЛАДНОЙ АРХИТЕКТУРНЫЙ СЕРВИС»

КОМАНДА «АВАНГАРД»



Виктор Гоголев

Архитектор ИС



Якубов Юсуф

Аналитик данных



Чудопалов Богдан

Backend-разработчик



Жулянов Владислав

Архитектор ИС

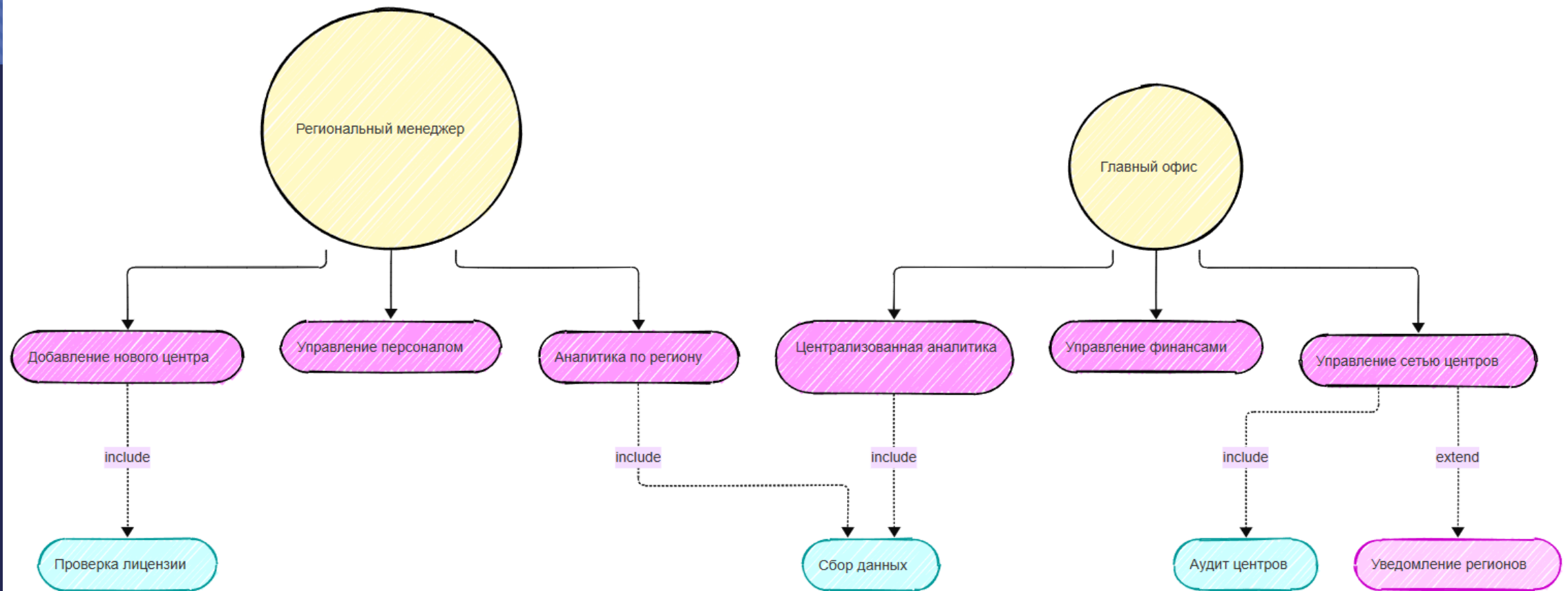


Сокол Игорь

Аналитик данных

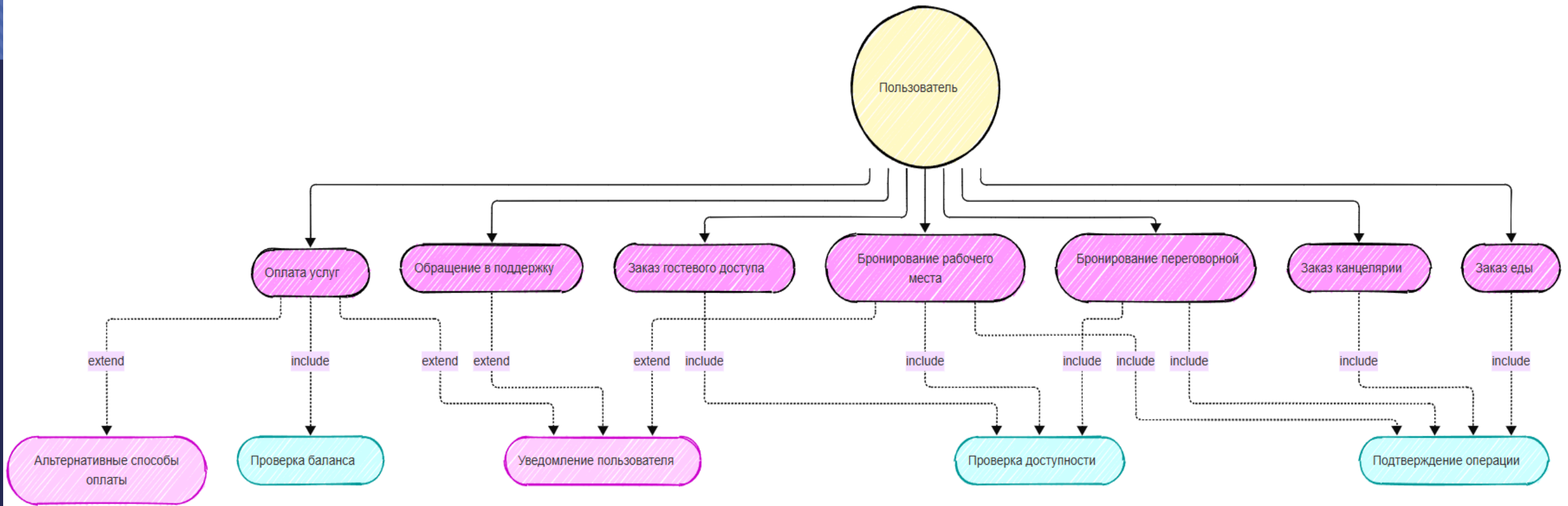


функциональная структура системы — схема и описание функций системы и их зависимостей



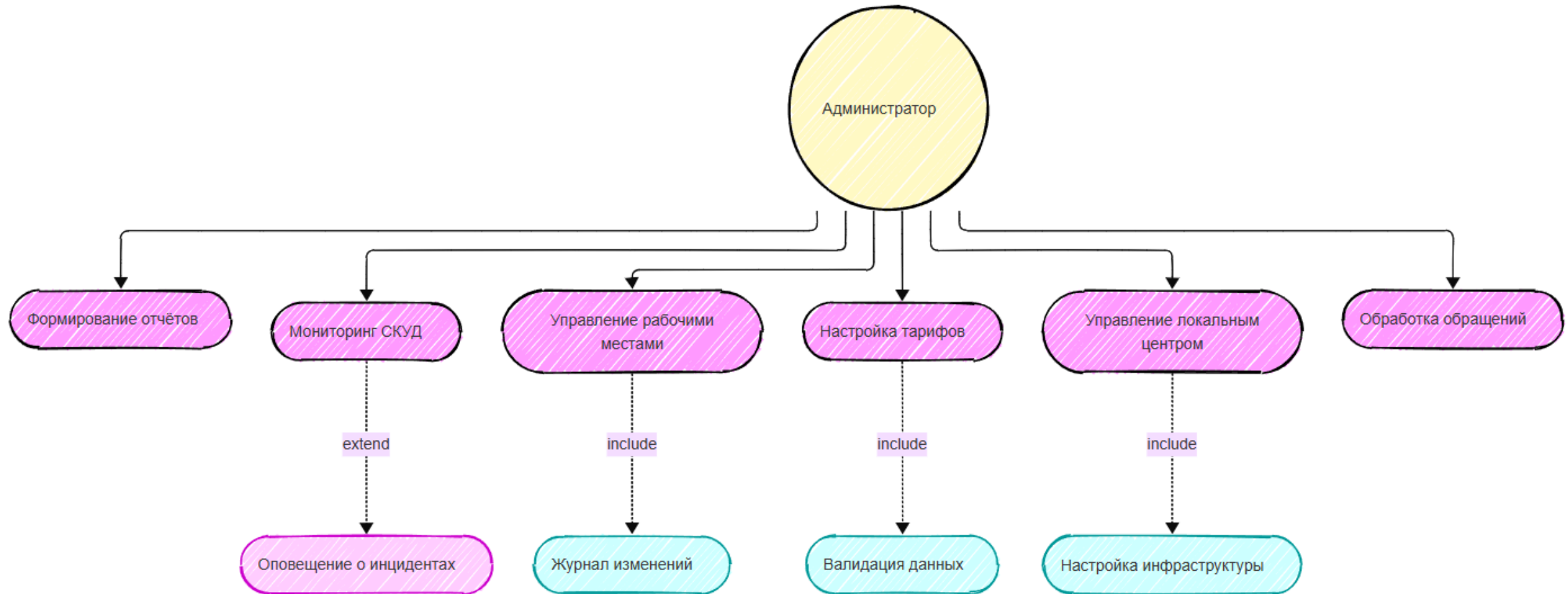
Use-case диаграмма (сущности Региональный менеджер и Главный офис)

функциональная структура системы — схема и описание функций системы и их зависимостей



Use-case диаграмма (сущность - пользователь)

функциональная структура системы — схема и описание функций системы и их зависимостей



Use-case диаграмма (сущность - администратор)

диаграмма потоков (функциональная структура сервиса) — все взаимодействия внутри сервиса и «снаружи» (DFD-формат)

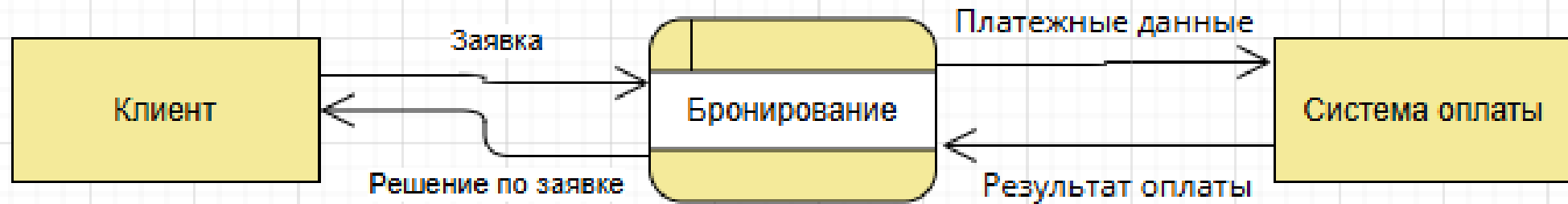


диаграмма потоков данных (верхний уровень абстракции)

диаграмма потоков (функциональная структура сервиса) — все взаимодействия внутри сервиса и «снаружи» (DFD-формат)

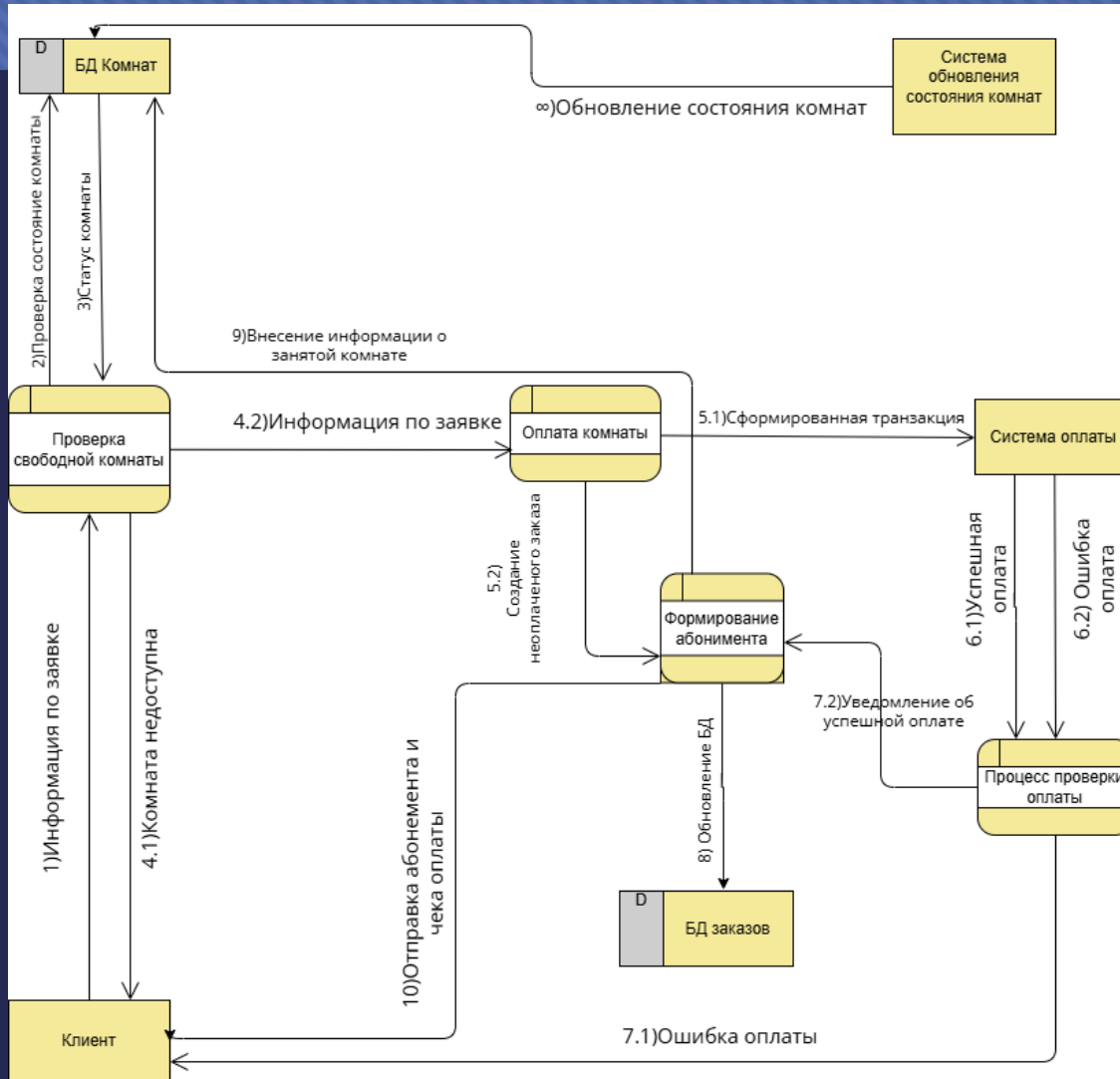
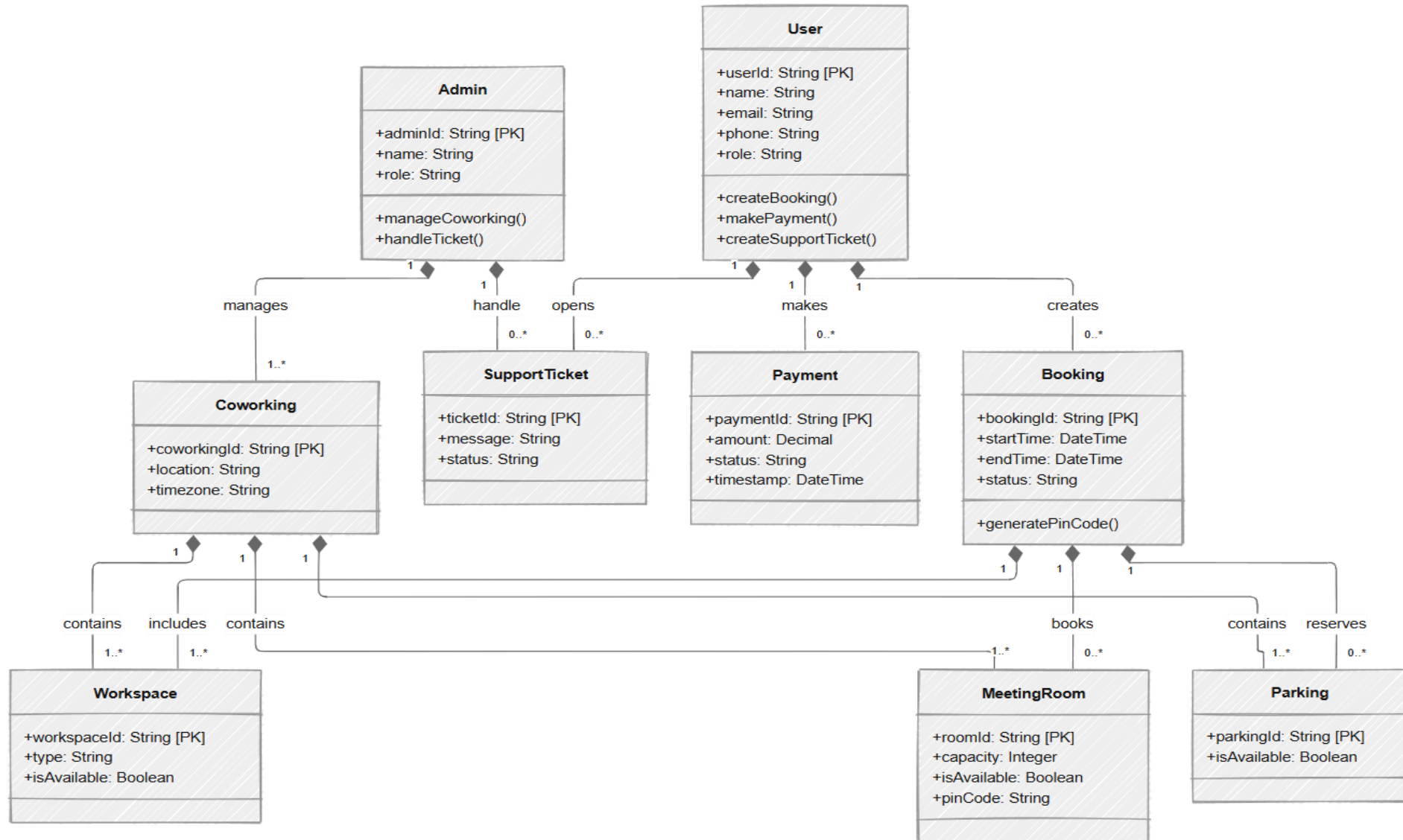


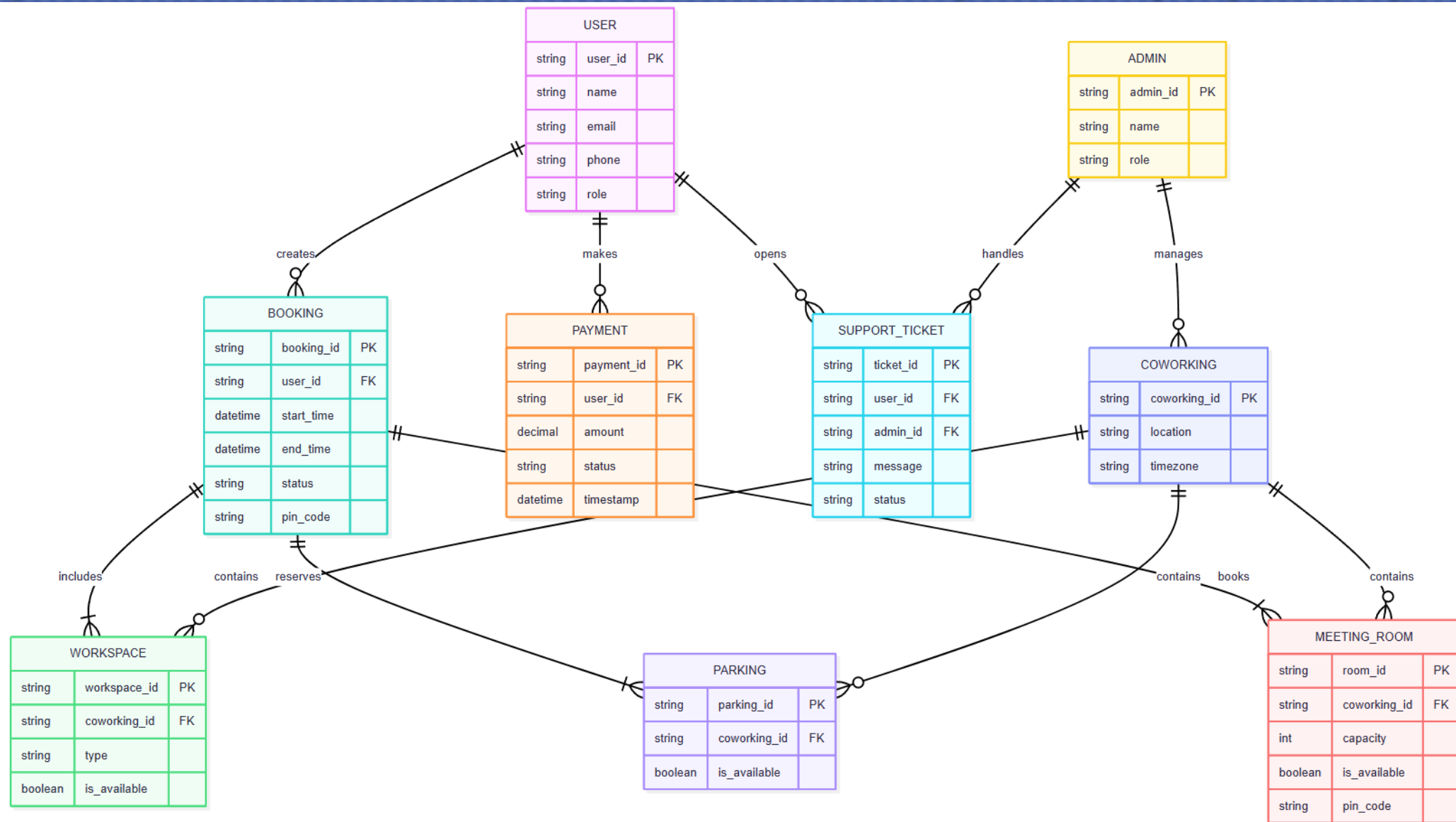
диаграмма потоков данных
(декомпозированная)

архитектура данных — логическая и физическая модель данных (UML-формат)



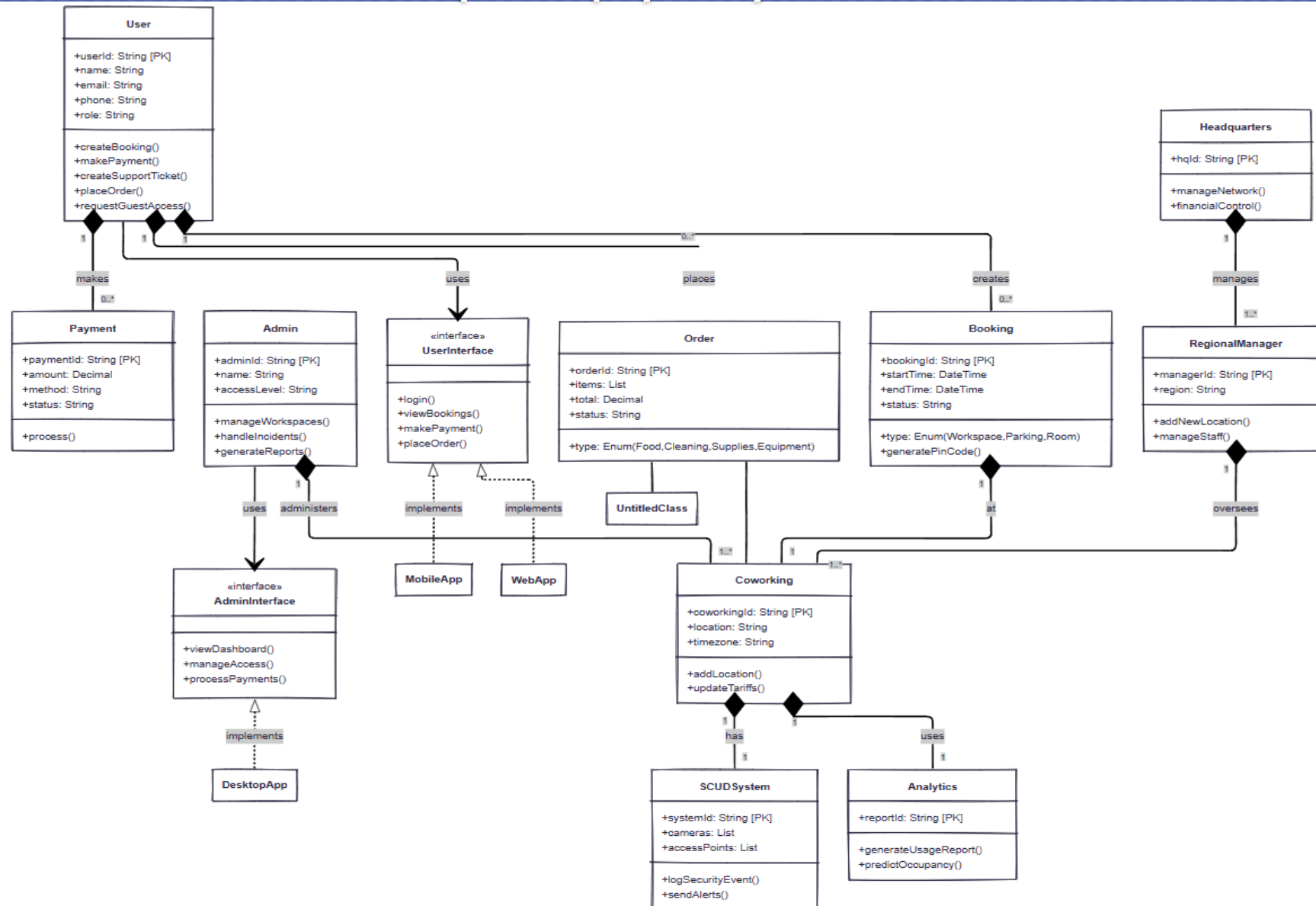
Базовый вариант диаграммы классов в качестве логической и концептуальной модели данных

архитектура данных — логическая и физическая модель данных (UML-формат)



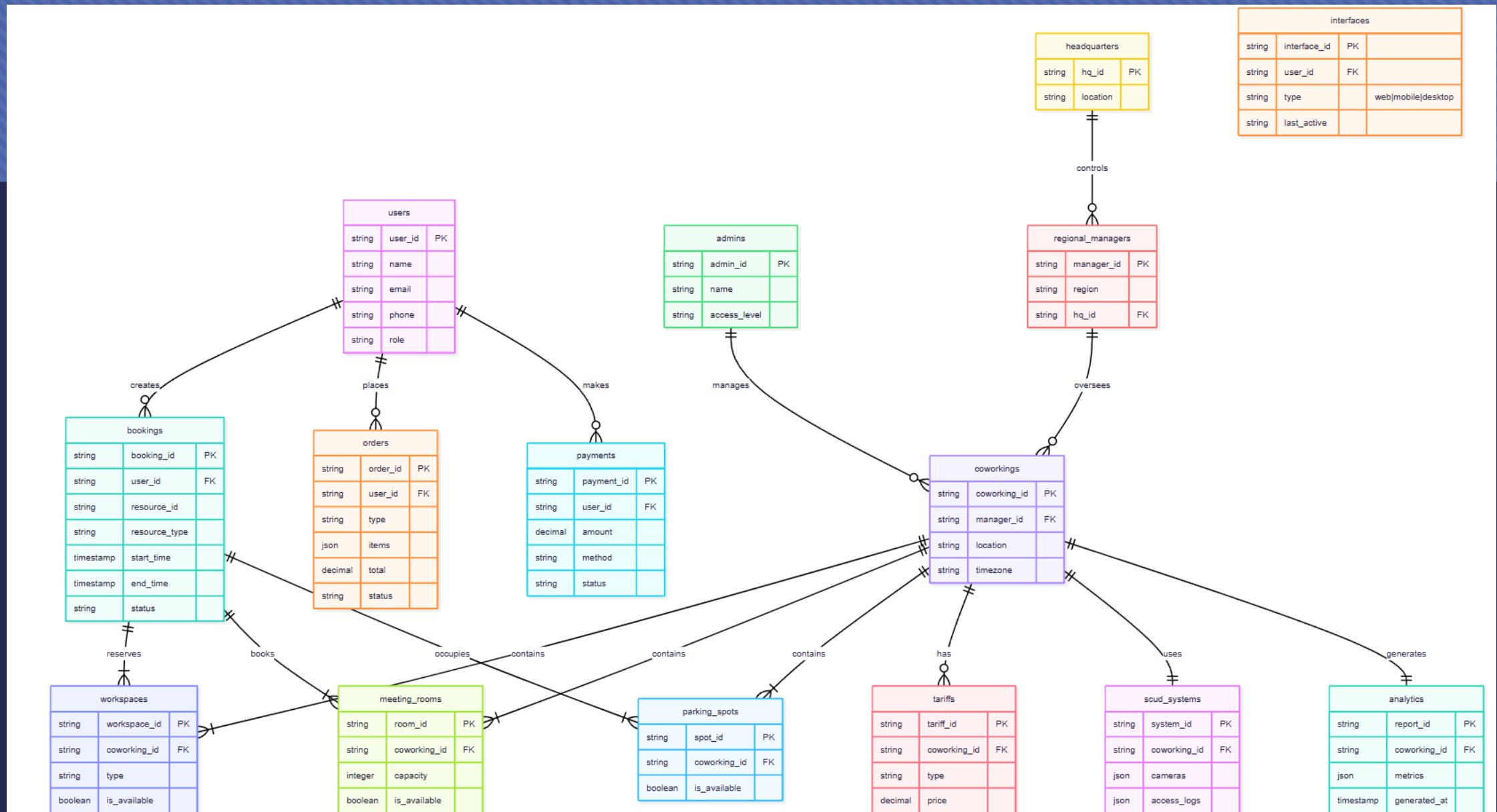
Базовый вариант ER-диаграмма в качестве физической модели данных

архитектура данных — логическая и физическая модель данных (UML-формат)



Расширенный вариант диаграммы классов в качестве логической и концептуальной модели данных

архитектура данных — логическая и физическая модель данных (UML-формат)



Расширенный вариант ER-диаграмма в качестве физической модели данных

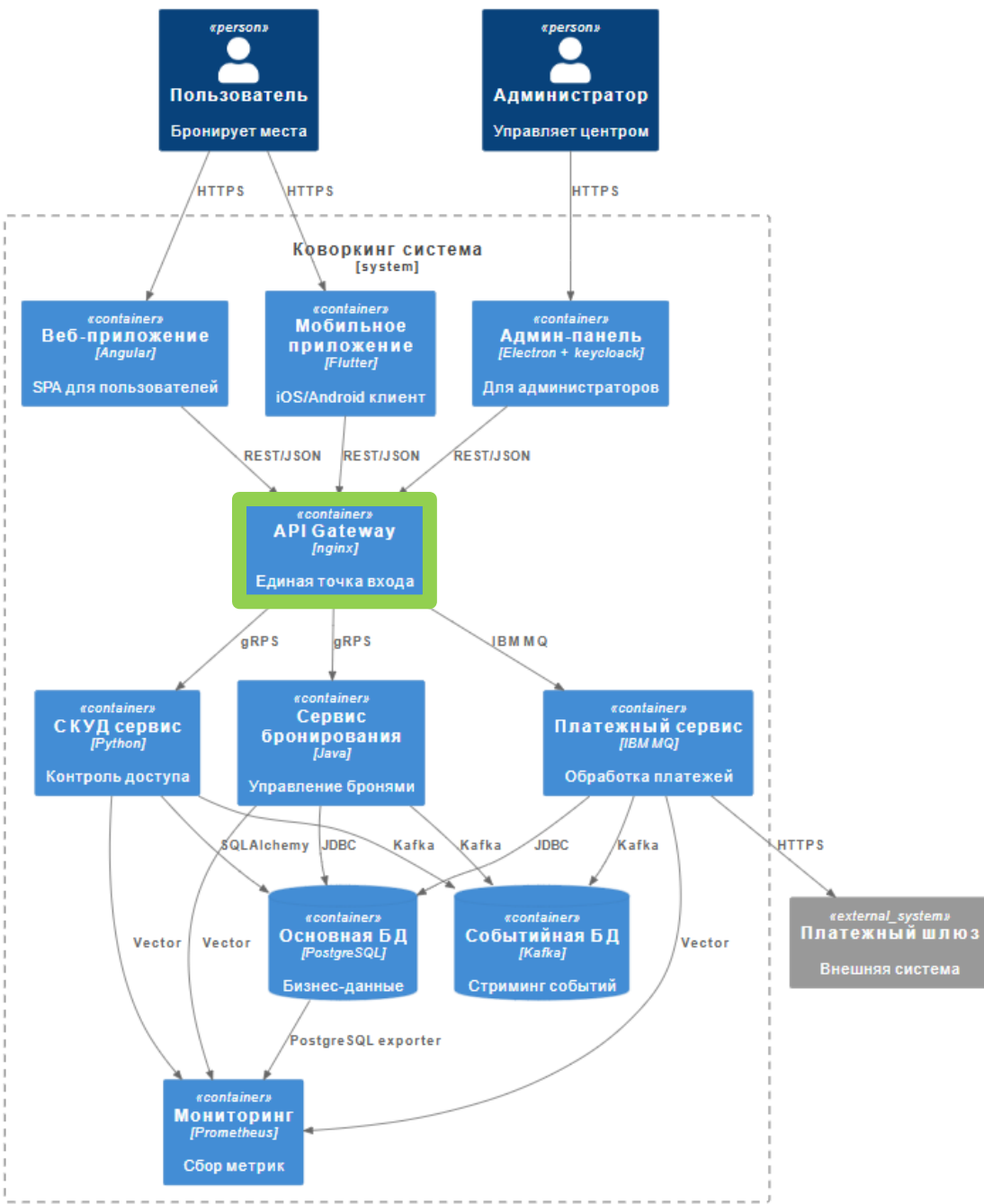
архитектура системы — внутреннее устройство проектируемой системы с указанием технологий реализации и взаимодействий с внешними системами

Диаграмма C4 – level 1 [Context]

Система управления коворкингами - Контекст



Система управления коворкингами - Контейнеры



архитектура системы — внутреннее устройство проектируемой системы с указанием технологий реализации и взаимодействий с внешними системами

Диаграмма C4 – level 2 [Container]

архитектура системы — внутреннее устройство проектируемой системы с указанием технологий реализации и взаимодействий с внешними системами

Диаграмма C4 – level 3 [Component]



API модулей системы — описание функций API, способов их вызова, возвращаемый результат (протоколы, методы и т.д.) (Swagger-формат)

Coworking Management API

1.0.0 OAS 3.0

API для управления коворкингами федеральной сети с JWT аутентификацией для защищённых операций

Servers

https://api.coworking.ru/v1

Authorize

Coworkings

POST	/coworkings	Создать новый коворкинг	🔒	▼
GET	/coworkings	Получить список всех коворкингов		▼
GET	/coworkings/{id}	Получить коворкинг по ID		▼
PATCH	/coworkings/{id}	Частично обновить коворкинг	🔒	▼
PUT	/coworkings/{id}	Полностью обновить коворкинг	🔒	▼
DELETE	/coworkings/{id}	Удалить коворкинг по ID	🔒	▼

Список endpoint для управления Coworking

POST /coworkings Создать новый коворкинг



Parameters

Try it out

No parameters

Request body required

application/json



Example Value | Schema

```
{
  "name": "string",
  "location": "string",
  "timezone": "string",
  "capacity": 0,
  "description": "string"
}
```

Responses

Code | Description

Links

201 Коворкинг успешно создан

No links

Media type

application/json

Controls Accept header.

Example Value | Schema

```
{
  "id": 1,
  "name": "Coworking Moscow Central",
  "location": "Москва, ул. Тверская, 10",
  "timezone": "Europe/Moscow",
  "capacity": 2000,
  "description": "Современный коворкинг в центре Москвы"
}
```

400 Ошибка в данных запроса

No links

Media type

application/json

Example Value | Schema

```
{
  "error": "message"
}
```

***POST /coworkings –
тело запроса,
параметры, статус
КОДЫ ОТВЕТОВ***

GET

/coworkings

Получить список всех коворкингов

^

Parameters

Try it out

No parameters

Responses

Code	Description	Links
200	<div>Список коворкингов</div> <div>Media type</div> <div>application/json</div> <div>Controls Accept header.</div> <div>Example Value Schema</div> <div><pre>[{ "id": 1, "name": "Coworking Moscow Central", "location": "Москва, ул. Тверская, 10", "timezone": "Europe/Moscow", "capacity": 2000, "description": "Современный коворкинг в центре Москвы" }]</pre></div>	

***GET /coworkings – тело запроса,
параметры, статус коды ответов***

GET

/coworkings/{id} Получить коворкинг по ID

^

Parameters

Try it out

Name	Description
id * required	Идентификатор коворкинга
integer (path)	<input type="text" value="id"/>

Responses

Code	Description	Links
200	<div>Данные коворкинга</div> <div>Media type</div> <div>application/json</div> <div>Content Accept header</div> <div>Example Value Schema</div> <div><pre>{ "id": 1, "name": "Coworking Moscow Central", "location": "Москва, ул. Тверская, 18", "timezone": "Europe/Moscow", "capacity": 2000, "description": "Современный коворкинг в центре Москвы"}</pre></div>	No links
404	<div>Коворкинг не найден</div> <div>Media type</div> <div>application/json</div> <div>Example Value Schema</div> <div><pre>{ "error": "message"}</pre></div>	No links

***GET /coworkings/{id} – тело запроса,
параметры, статус коды ответов***

PATCH

/coworkings/{id} Частично обновить коворкинг

🔒 ⬆

Parameters

Try it out

Name	Description
id <small>★ required</small>	<div>id</div>
integer	
(path)	

Request body required

application/json

Example Value | Schema

```
{  "name": "string",  "location": "string",  "timezone": "string",  "capacity": 0,  "description": "string"}
```

Responses

Code	Description	Links
200	Коворкинг обновлен <div><div>Media type<div>application/json</div></div><div>Control: Accept header.</div><div>Example Value Schema</div><div><pre>{ "id": 1, "name": "Coworking Moscow Central", "location": "Москва, ул. Тверская, 10", "timezone": "Europe/Moscow", "capacity": 2000, "description": "Современный коворкинг в центре Москвы"}</pre></div></div>	No links
404	Коворкинг не найден <div><div>Media type<div>application/json</div></div><div>Example Value Schema</div><div><pre>{ "error": "message"}</pre></div></div>	No links

PATH
/coworkings/{id} –
тело запроса,
параметры, статус
КОДЫ ОТВЕТОВ

PUT /coworkings/{id} – тело запроса, параметры, статус КОДЫ ОТВЕТОВ

PUT

/coworkings/{id} Полностью обновить коворкинг

Parameters

Try it out

Name	Description
id <small>★ required</small>	<input type="text" value="id"/>
integer	
(path)	

Request body required

application/json

Example Value | Schema



```
{
  "name": "string",
  "location": "string",
  "timezone": "string",
  "capacity": 0,
  "description": "string"
}
```

Responses

Code	Description	Links
200	Коворкинг полностью обновлен	No links
	<div>Media type</div> <div>application/json</div> <div>Content Accept header</div> <div>Example Value Schema</div> <div><pre>{ "id": 1, "name": "Coworking Moscow Central", "location": "Москва, ул. Тверская, 18", "timezone": "Europe/Moscow", "capacity": 2000, "description": "Современный коворкинг в центре Москвы" }</pre></div>	
404	Коворкинг не найден	No links
	<div>Media type</div> <div>application/json</div> <div>Example Value Schema</div> <div><pre>{ "error": "message" }</pre></div>	

DELETE

/coworkings/{id} Удалить коворкинг по ID



Parameters

Try it out

Name	Description
<div><div>id * required</div><div>integer</div><div>(path)</div></div>	<div>id</div>

Responses

Code	Description	Links
204	Коворкинг успешно удален	No links
404	Коворкинг не найден	No links

Media type

application/json

Example Value

Schema

```
{
  "error": "message"
}
```

***DELETE /coworkings/{id} – тело запроса,
параметры, статус коды ответов***


```

Coworking {
  id          integer
              example: 1
              Уникальный идентификатор коворкинга

  name*       string
              example: Coworking Moscow Central
              Название коворкинга

  location*    string
              example: Москва, ул. Тверская, 10
              Адрес коворкинга

  timezone     string
              example: Europe/Moscow
              Часовой пояс коворкинга

  capacity     integer
              example: 2000
              Количество рабочих мест

  description  string
              example: Современный коворкинг в центре Москвы
              Описание коворкинга
}

```

```

CoworkingCreateRequest {
  name*       string
  location*    string
  timezone*    string
  capacity*    integer
  description  string
}

```

```

CoworkingUpdateRequest {
  name        string
  location     string
  timezone     string
  capacity     integer
  description  string
}

```

```

ErrorResponse {
  error  string
              example: message
}

```

Сущности, их поля и типы данных для управления Coworking

API модулей системы — описание функций API, способов их вызова, возвращаемый результат (протоколы, методы и т.д.) (Swagger-формат)

Authentication Service API 1.0.0 OAS 3.0

API сервиса аутентификации и авторизации пользователей федеральной сети коворкингов

Servers

<https://auth.coworking.ru/api/v1>

Authorize

Authentication

POST `/auth/login` Вход пользователя/администратора

POST `/auth/logout` Выход пользователя (аннулирование refresh токена)

POST `/auth/refresh` Обновление access токена по refresh токenu

POST `/auth/register` Регистрация нового пользователя

Список endpoint для авторизации/аутентификации пользователей

POST

/auth/login

Вход пользователя/администратора

⌵

Parameters

Try it out

No parameters

Request body

required

application/json

⌵

Example Value

Schema

```
{
  "email": "user@example.com",
  "password": "P@ssw0rd!"
}
```

Responses

Code	Description	Links
200	<div>Успешный вход, возвращает токены доступа и обновления</div> <div><div>Media type</div><div>application/json</div><div>⌵</div></div> <div>Controls Accept header.</div> <div><div>Example Value</div><div>Schema</div></div> <div><pre>{ "access_token": "string", "refresh_token": "string" }</pre></div>	No links
401	<div>Неверные учетные данные</div> <div><div>Media type</div><div>application/json</div><div>⌵</div></div> <div><div>Example Value</div><div>Schema</div></div> <div><pre>{ "error": "Invalid credentials" }</pre></div>	No links

POST /auth/login – тело запроса, параметры, статус коды ответов

POST

/auth/logout

Выход пользователя (аннулирование refresh токена)

^

Parameters

Try it out

No parameters

Request body required

application/json

Example Value | Schema

```
{  "refresh_token": "string"}
```

Responses

Code	Description	Links
200	Успешный выход <div><div>Media type<div>application/json</div></div><div>Content Accept header.</div><div>Example Value Schema<div><pre>{ "message": "Logout successful"}</pre></div></div></div> <td>No links</td>	No links
400	Ошибка при выходе (например, неверный токен) <div><div>Media type<div>application/json</div></div><div>Example Value Schema<div><pre>{ "error": "Invalid credentials"}</pre></div></div></div> <td>No links</td>	No links

POST /auth/logout – тело запроса, параметры, статус коды ответов

POST

/auth/refresh

Обновление access токена по refresh токenu

^

Parameters

Try it out

No parameters

Request body

required

application/json

▼

Example Value

Schema

```
{
  "refresh_token": "string"
}
```

Responses

Code	Description	Links
200	Новый access токен <div><div>Media type</div><div>application/json</div><div>▼</div><div>Controls Accept header.</div><div>Example Value</div><div>Schema</div><div><pre>{ "access_token": "string" }</pre></div></div>	No links
401	Неверный или просроченный refresh токен <div><div>Media type</div><div>application/json</div><div>▼</div><div>Example Value</div><div>Schema</div><div><pre>{ "error": "Invalid credentials" }</pre></div></div>	No links

POST /auth/refresh – тело запроса, параметры, статус коды ответов

POST

/auth/register

Регистрация нового пользователя

⌵

Parameters

Try it out

No parameters

Request body

required

application/json

⌵

Example Value

Schema

```
{
  "email": "newuser@example.com",
  "password": "NewUserPass123!",
  "full_name": "Иван Иванов"
}
```

Responses

Code	Description	Links
201	<div>Пользователь успешно зарегистрирован</div> <div>Media type</div> <div>application/json</div> <div>⌵</div> <div>Controls Accept header.</div> <div>Example Value</div> <div>Schema</div> <div><pre>{ "user_id": 101, "message": "Регистрация прошла успешно" }</pre></div>	No links
400	<div>Ошибка в данных регистрации (например, email уже занят)</div> <div>Media type</div> <div>application/json</div> <div>⌵</div> <div>Example Value</div> <div>Schema</div> <div><pre>{ "error": "Invalid credentials" }</pre></div>	No links

POST /auth/register – тело запроса, параметры, статус коды ответов

Schemas	
LoginRequest ▾ { <div> <div>email*</div> <div> ▾ string(\$email) <i>example: user@example.com</i> Электронная почта пользователя </div> </div> <div> <div>password*</div> <div> ▾ string(\$password) <i>example: P@ssw0rd!</i> Пароль пользователя </div> </div>	↕
LoginResponse ▾ { <div> <div>access_token</div> <div> ▾ string JWT токен доступа </div> </div> <div> <div>refresh_token</div> <div> ▾ string Токен обновления для получения нового access_token </div> </div>	↕
LogoutRequest ▾ { <div> <div>refresh_token*</div> <div> ▾ string Токен обновления, который нужно аннулировать </div> </div>	↕
RefreshTokenRequest ▾ { <div> <div>refresh_token*</div> <div> ▾ string Токен обновления для получения нового access_token </div> </div>	↕

Сущности, их поля и типы данных для авторизации/аутентификации пользователей

RefreshTokenResponse ▾ {

access_token ▾ string
Новый JWT token доступа

}

RegisterRequest ▾ {

email* ▾ string(\$email)
example: newuser@example.com
Электронная почта нового пользователя

password* ▾ string(\$password)
example: NewUserPass123!
Пароль нового пользователя

full_name* ▾ string
example: Иван Иванов
Полное имя пользователя

}

RegisterResponse ▾ {

user_id ▾ integer
example: 101

message ▾ string
example: Регистрация прошла успешно

}

ErrorResponse ▾ {

error ☐ string
example: Invalid credentials

}

Сущности, их поля и типы данных для авторизации/аутентификации пользователей

API модулей системы — описание функций API, способов их вызова, возвращаемый результат (протоколы, методы и т.д.) (Swagger-формат)

Booking Service API 1.0.0 OAS 3.0

API сервиса бронирования помещений и рабочих мест федеральной сети коворкингов

Servers

<https://booking.coworking.ru/api/v1>

Authorize

Rooms

GET

/rooms

Получить список помещений для бронирования

🔒

▼

POST

/rooms/{id}/book

Забронировать помещение

🔒

▼

GET

/users/{id}/rooms

Получить список бронирований помещений пользователя

🔒

▼



Список endpoint для бронирования помещений и рабочих мест

Rooms

GET

/rooms

Получить список помещений для бронирования



Получение списка внутренних помещений коворкинга с возможностью фильтрации

Parameters

Try it out

Name	Description
location	Фильтр по локации
string (query)	<input type="text" value="location"/>
type	Фильтр по типу помещения
string (query)	<input type="text" value="type"/>

Responses

Code	Description	Links
200	Список помещений	No links

Media type

Controls Accept header.

Example Value | Schema

```
[
  {
    "id": 456,
    "name": "Переговорная №1",
    "location": "Москва, ул. Тверская, 10",
    "type": "room",
    "status": "available"
  }
]
```

GET /rooms – тело запроса, параметры, статус коды ответов

POST

/rooms/{id}/book

Забронировать помещение

Создание бронирования на помещение по ID

Parameters

Name	Description
id <small>★ required</small>	Идентификатор помещения
integer	
(path)	id

Request body required

Example Value | Schema

```
{  "start_date": "2025-05-21T09:00:00+03:00",  "end_date": "2025-05-21T18:00:00+03:00"}
```

Responses	
Code	Description
200	Успешное бронирование помещения <div>Media type application/json</div> <div>Controls Accept Header.</div> <div>Example Value Schema</div> <pre>{ "status": "success", "booking_id": 789, "resource_id": 456, "resource_type": "room", "start_date": "2025-05-21T09:00:00+03:00", "end_date": "2025-05-21T18:00:00+03:00"}</pre>
400	Ошибка в данных бронирования <div>Media type application/json</div> <div>Example Value Schema</div> <pre>{ "error": "Resource not found"}</pre>
404	Помещение не найдено <div>Media type application/json</div> <div>Example Value Schema</div> <pre>{ "error": "Resource not found"}</pre>

***POST /rooms/{id}/book – тело запроса, параметры, статус коды
ОТВЕТОВ***

GET

/users/{id}/rooms

Получить список бронирований помещений пользователя

🔒 ⬆

Получение всех бронирований помещений, сделанных пользователем

Parameters

Try it out

Name	Description
id <small>* required</small>	Идентификатор пользователя
integer (path)	<input type="text" value="id"/>

Responses

Code	Description	Links
200	Список бронирований помещений пользователя <div>Media type <div>application/json</div><div>Controls Accept header</div><div>Example Value Schema</div><div><pre>[{ "status": "success", "booking_id": 789, "resource_id": 456, "resource_type": "room", "start_date": "2025-05-21T09:00:00+03:00", "end_date": "2025-05-21T18:00:00+03:00" }]</pre></div></div>	No links
404	Пользователь не найден <div>Media type <div>application/json</div><div>Example Value Schema</div><div><pre>{ "error": "Resource not found" }</pre></div></div>	No links

**GET /users/{id}/rooms– тело запроса, параметры, статус коды
ОТВЕТОВ**

BookingRequest ▾ {

start_date* ▾ string(\$date-time)
example: 2025-05-21T09:00:00+03:00
Дата и время начала бронирования

end_date* ▾ string(\$date-time)
example: 2025-05-21T18:00:00+03:00
Дата и время окончания бронирования

}

BookingResponse ▾ {

status ▾ string
example: success

booking_id ▾ integer
example: 789

resource_id ▾ integer
example: 456

resource_type ▾ string
example: room

start_date ▾ string(\$date-time)
example: 2025-05-21T09:00:00+03:00

end_date ▾ string(\$date-time)
example: 2025-05-21T18:00:00+03:00

}

Resource ▾ {

id ▾ integer
example: 456

name ▾ string
example: Переговорная №1

location ▾ string
example: Москва, ул. Тверская, 10

type ▾ string
example: room

status ▾ string
example: available

}

ErrorResponse ▾ {

error ▾ string
example: Resource not found

}

Сущности, их поля и типы данных для бронирования помещений и рабочих мест

API модулей системы — описание функций API, способов их вызова, возвращаемый результат (протоколы, методы и т.д.) (Swagger-формат)

Workspace Service API

1.0.0 OAS 3.0

API сервиса управления рабочими местами федеральной сети коворкингов

Servers

https://workspace.coworking.ru/api/v1

Authorize

Workspaces

GET

/workspaces

Получить список рабочих мест

GET

/workspaces/{id}

Получить детали рабочего места

POST

/workspaces/{id}/book

Забронировать рабочее место

POST

/workspaces/{id}/release

Освободить рабочее место

GET

/users/{id}/workspaces



Получить список бронирований пользователя

Список endpoint для управления рабочими местами

GET

/workspaces

Получить список рабочих мест



Получение списка рабочих мест с возможностью фильтрации по локации и статусу

Parameters

Try it out

Name	Description
location	Фильтр по локации
string (query)	<input type="text" value="location"/>
status	Фильтр по статусу рабочего места
string (query)	<input type="text" value="status"/>

Responses

Code	Description	Links
200	<div>Список рабочих мест</div> <div>Media type</div> <div><div>application/json</div></div> <div>Controls Accept header.</div> <div>Example Value Schema</div> <div><pre>[{ "id": 123, "location": "Москва, ул. Тверская, 10", "status": "available", "type": "open_space" }]</pre></div>	

No links

 |

GET /workspaces – тело запроса, параметры, статус коды ответов

GET

/workspaces/{id} Получить детали рабочего места

🔒 ⬆

Получение полной информации о рабочем месте по ID

Parameters

Try it out

Name	Description
id * required	Идентификатор рабочего места
integer (path)	<input type="text" value="id"/>

Responses

Code	Description	Links
200	<div>Детали рабочего места</div> <div>Media type</div> <div>application/json</div> <div>Controls Accept header.</div> <div>Example Value Schema</div> <div><pre>{ "id": 123, "location": "Москва, ул. Тверская, 10", "status": "available", "type": "open_space"}</pre></div>	No links
404	<div>Рабочее место не найдено</div> <div>Media type</div> <div>application/json</div> <div>Example Value Schema</div> <div><pre>{ "error": "Workspace not found"}</pre></div>	No links

GET /workspaces/{id} – тело запроса, параметры, статус коды ответов

POST

/workspaces/{id}/book

Забронировать рабочее место

Создание бронирования рабочего места на указанный период

Parameters

Name	Description
id * required	Идентификатор рабочего места для бронирования
integer	id
(path)	

Request body required

Example Value

Schema

```
{
  "start_date": "2025-05-21T09:00:00+03:00",
  "end_date": "2025-05-21T18:00:00+03:00"
}
```



Responses	
Code	Description
200	Успешное бронирование рабочего места <div>Media type application/json Controls Accept header.</div> <div>Example Value Schema</div> <pre>{ "status": "success", "booking_id": 456, "workspace_id": 123, "start_date": "2025-05-21T09:00:00+03:00", "end_date": "2025-05-21T18:00:00+03:00" }</pre>
400	Ошибка в данных бронирования <div>Media type application/json</div> <div>Example Value Schema</div> <pre>{ "error": "Workspace not found" }</pre>
404	Рабочее место не найдено <div>Media type application/json</div> <div>Example Value Schema</div> <pre>{ "error": "Workspace not found" }</pre>

POST /workspaces/{id}/book– тело запроса, параметры, статус коды
ОТВЕТОВ

POST

/workspaces/{id}/release

Освободить рабочее место



Освобождение ранее забронированного рабочего места

Parameters

Try it out

Name	Description
id * required	Идентификатор рабочего места для освобождения
integer (path)	<input type="text" value="id"/>

Responses

Code	Description	Links
200	<div>Успешное освобождение рабочего места</div> <div>Media type</div> <div><div>application/json</div></div> <div>Controls Accept header.</div> <div>Example Value Schema</div> <div><pre>{ "status": "success" }</pre></div>	No links
404	<div>Рабочее место не найдено</div> <div>Media type</div> <div><div>application/json</div></div> <div>Example Value Schema</div> <div><pre>{ "error": "Workspace not found" }</pre></div>	No links

POST /workspaces/{id}/release– тело запроса, параметры, статус коды
ОТВЕТОВ

GET

/users/{id}/workspaces

Получить список бронирований пользователя

Получение всех бронирований рабочих мест, сделанных пользователем

Parameters

Name	Description
id * required	Идентификатор пользователя
integer (path)	<input type="text" value="id"/>

Responses

Responses

Code	Description
200	<div>Список бронирований пользователя</div> <div>Media type</div> <div>application/json</div> <div>Controls Accept header.</div> <div>Example Value Schema</div> <div><pre>[{ "status": "success", "booking_id": 456, "workspace_id": 123, "start_date": "2025-05-21T09:00:00+03:00", "end_date": "2025-05-21T18:00:00+03:00" }]</pre></div>
404	<div>Пользователь не найден</div> <div>Media type</div> <div>application/json</div> <div>Example Value Schema</div> <div><pre>{ "error": "Workspace not found" }</pre></div>

***GET /users/{id}/workspace– тело запроса, параметры, статус коды
ОТВЕТОВ***

Schemas

Workspace ▾ {

id ▾ integer
 example: 123

location ▾ string
 example: Москва, ул. Тверская, 10

status ▾ string
 example: available
 Статус рабочего места (доступно, занято, забронировано)

type ▾ string
 example: open_space
 Тип рабочего места (открытое, кабинет, переговорная)

}

BookingRequest ▾ {

start_date* ▾ string(\$date-time)
 example: 2025-05-21T09:00:00+03:00

end_date* ▾ string(\$date-time)
 example: 2025-05-21T18:00:00+03:00

}

**Сущности, их поля и типы
данных для управления
рабочими местами**

Сущности, их поля и типы данных для управления рабочими местами

BookingResponse ▾ {

```
  status          ▾ string
                  example: success

  booking_id      ▾ integer
                  example: 456

  workspace_id    ▾ integer
                  example: 123

  start_date      ▾ string($date-time)
                  example: 2025-05-21T09:00:00+03:00

  end_date        ▾ string($date-time)
                  example: 2025-05-21T18:00:00+03:00
}
```

ErrorResponse ▾ {

```
  error          ▾ string
                  example: Workspace not found
}
```

API модулей системы — описание функций API, способов их вызова, возвращаемый результат (протоколы, методы и т.д.) (Swagger-формат)

Support Service API 1.0.0 OAS 3.0

API службы поддержки федеральной сети коворкингов

Servers

<https://support.coworking.ru/api/v1> ▾

Authorize



Support



POST

/support/tickets Создать новое обращение в службу поддержки



GET

/support/tickets Получить список обращений пользователя или администратора



GET

/support/tickets/{ticket_id} Получить детали конкретного обращения



POST

/support/tickets/{ticket_id}/reply Ответить на обращение в службу поддержки



Список endpoint для службы поддержки

POST

/support/tickets Создать новое обращение в службу поддержки

Parameters

No parameters

Request body required

Example Value | Schema

```
{
  "subject": "Проблема с бронированием",
  "description": "Не удаётся забронировать рабочее место на 21 мая",
  "attachments": [
    "file1.png",
    "log.txt"
  ]
}
```



Responses	
Code	Description
201	Обращение успешно создано
	Media type
	<div>application/json</div>
	Controls Accept header.
	Example Value Schema
	<pre>{ "ticket_id": 12345, "status": "open" }</pre>
400	Ошибка в данных запроса
	Media type
	<div>application/json</div>
	Example Value Schema
	<pre>{ "error": "Ticket not found" }</pre>

POST /support/tickets – тело запроса, параметры, статус коды ответов

GET

/support/tickets

Получить список обращений пользователя или администратора



Parameters

Try it out

Name	Description
status	Фильтр по статусу обращения (например, open, closed)
string (query)	<input type="text" value="status"/>
date	Фильтр по дате создания обращения
string(\$date) (query)	<input type="text" value="date"/>

Responses

Code	Description	Links
200	<div>Список обращений</div> <div>Media type</div> <div><div>application/json</div><div>▼</div></div> <div>Controls Accept header.</div> <div>Example Value Schema</div> <div><pre>[{ "ticket_id": 12345, "subject": "Проблема с бронированием", "status": "open", "created_at": "2025-05-20T10:00:00Z" }]</pre></div>	No links

GET /support/tickets – тело запроса, параметры, статус коды ответов

GET

/support/tickets/{ticket_id} Получить детали конкретного обращения

Parameters

Name	Description
ticket_id * required	Идентификатор обращения
integer (path)	<div>ticket_id</div>

***GET /support/tickets/{ticket_id} –
тело запроса, параметры,
статус коды ответов***

Responses

Code	Description
200	<div>Детали обращения</div> <div>Media type<div>application/json</div></div> <div>Controls Accept header.</div> <div>Example Value Schema</div> <div><pre>{ "ticket_id": 12345, "subject": "Проблема с бронированием", "status": "open", "created_at": "2025-05-20T10:00:00Z", "description": "Не удаётся забронировать рабочее место на 21 мая", "attachments": ["file1.png", "log.txt"], "messages": [{ "sender": "support_agent", "message": "Пожалуйста, уточните, какую ошибку вы видите?", "sent_at": "2025-05-20T11:00:00Z" }] }</pre></div>
404	<div>Обращение не найдено</div> <div>Media type<div>application/json</div></div> <div>Example Value Schema</div> <div><pre>{ "error": "Ticket not found" }</pre></div>

POST**/support/tickets/{ticket_id}/reply** Ответить на обращение в службу поддержки**Parameters**

Name	Description
------	-------------

ticket_id * required	Идентификатор обращения
-----------------------------	-------------------------

integer (path)	<input type="text" value="ticket_id"/>
-------------------	--

Request body required

Example Value | Schema

```
{  
  "message": "Проблема устранена, попробуйте снова."  
}
```

Responses

Code	Description
------	-------------

200	Ответ успешно добавлен
-----	------------------------

Media type

Controls Accept header.

Example Value | Schema

```
{  
  "status": "success"  
}
```

404	Обращение не найдено
-----	----------------------

Media type

Example Value | Schema

```
{  
  "error": "Ticket not found"  
}
```

GET /support/tickets/{ticket_id}/reply – тело запроса, параметры, статус коды ответов

SupportTicketCreateRequest ▾ {

subject* ▾ *string*
example: Проблема с бронированием
Тема обращения

description* ▾ *string*
example: Не удаётся забронировать рабочее место на 21 мая
Подробное описание проблемы

attachments ▾ [
example: List ["file1.png", "log.txt"]
Список URL или идентификаторов вложений

> [...]]

}

SupportTicketResponse ▾ {

ticket_id ▾ *integer*
example: 12345

status ▾ *string*
example: open

}

**Сущности, их поля и типы
данных для службы
поддержки**

Сущности, их поля и типы данных для службы поддержки

SupportTicketSummary ▾ {

```
  ticket_id      ▾ integer
                  example: 12345

  subject        ▾ string
                  example: Проблема с бронированием

  status         ▾ string
                  example: open

  created_at     ▾ string($date-time)
                  example: 2025-05-20T10:00:00Z

}
```

SupportTicketDetail ▾ {

```
  ticket_id      ▾ integer
                  example: 12345

  subject        ▾ string
                  example: Проблема с бронированием

  status         ▾ string
                  example: open

  created_at     ▾ string($date-time)
                  example: 2025-05-20T10:00:00Z

  description    ▾ string
                  example: Не удаётся забронировать рабочее место на 21 мая

  attachments    ▾ [
                  example: List [ "file1.png", "log.txt" ]

  ▾ string]

  messages       ▾ [ ▾ {
                      sender      ▾ string
                                  example: support_agent

                      message     ▾ string
                                  example: Пожалуйста, уточните, какую ошибку вы видите?

                      sent_at     ▾ string($date-time)
                                  example: 2025-05-20T11:00:00Z

                    }]

}
```

SupportTicketReplyRequest ▾ {

message*

▾ string

example: Проблема устранена, попробуйте снова.

Текст ответа

}

ErrorResponse ▾ {

error

▾ string

example: Ticket not found

}

Сущности, их поля и типы данных для службы поддержки

Если тут плохо, файлы .yaml со swagger доступны в папке «/ API модулей системы»
Рядом с презентацией"

Некоторые endpoint не успели оформить в презентации, они так же находятся в папке
«папке / API модулей системы»