

# Сервис решений для бизнеса

---

## Верхнеуровневое описание

Задача проекта - предоставлять набор сервисов (единоразово и по подписке), помогающих бизнесу: юридическое сопровождение, ведение бухгалтерии, организация доставки. Клиент - представитель определенной компании - получает доступ к сервисам. С другой стороны, компании, сотрудничающие с этим проектом, обслуживают определенные типы заявок и могут управлять полученными заявками.

Заинтересованным лицом, кроме клиента, являются компании, предоставляющие определенный набор услуг, т.е. такой сервис является по своей сути агрегатором, но от клиента (по изначальной идее) не должен требоваться выбор конкретной компании. Для клиента доступен выбор услуги и заказ либо единоразово, либо постоянно (по подписке). Для компании доступен прием и отклонение заявок, просмотр принятых заявок. Администраторам доступно управление заявками, просмотр статусов и доп. средства контроля, необходимые для того, чтобы сервис мог предоставлять гарантии, что заказанная услуга будет оказана.

## Аналоги

profi.ru - Посредник между профессионалами и клиентами. Отличие от текущей идеи в ориентировании на разовые задачи и на частных клиентов.

## Действующие лица

1. Клиент - представитель бизнеса
2. Поставщик - компания, предоставляющая определенный набор услуг
3. Администратор сервиса

## Функциональные требования

1. Управление пользователями
  1. Регистрация клиента
  2. Регистрация поставщика
  3. Авторизация клиента
  4. Авторизация поставщика
2. Управление финансами
  1. Пользователь может пополнять баланс кошелька
  2. Пользователь может выводить средства на свой банковский счет
3. Управление подписками со стороны клиента
  1. Клиент может оформлять подписку на услуги
  2. Клиент может отменять подписку
  3. Клиент может просматривать оформленные заявки
  4. Система уведомляет клиента в том случае, если ни один поставщик не принял заявку в течение определенного времени
4. Управление подписками со стороны поставщика
  1. Поставщик может просматривать заявки тех типов, которые может обрабатывать

2. Поставщик может принимать заявки
5. Взаимодействие между клиентом и поставщиком
  1. Клиент может создавать заказы в рамках своей подписки
  2. Поставщик получает уведомления о новых заказах и помечать их как принятые в обработку
  3. Клиент и поставщик могут вести диалог
  4. Поставщик и клиент могут помечать их как выполненные
  5. Клиент и поставщик могут пожаловаться на заказ
6. Действия администратора
  1. Администратор может просматривать жалобы
  2. Администратор может связываться с клиентом и поставщиком
  3. Администратор может возвращать часть денег поставщику и клиенту

## Развертывание

Проект будет состоять из 6 микросервисов:

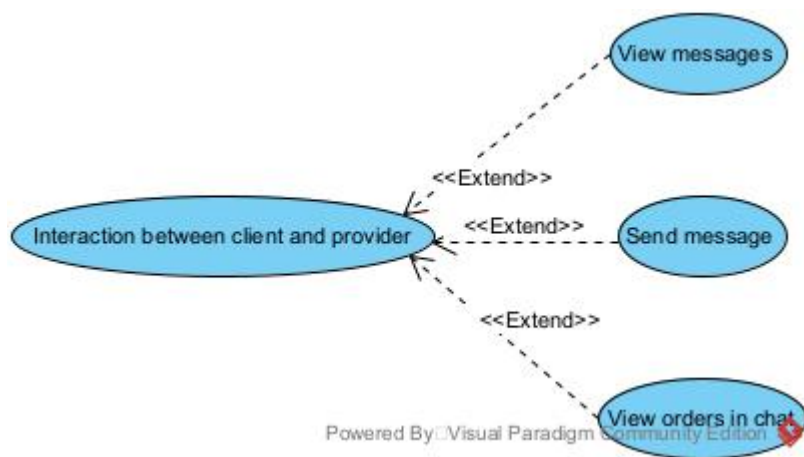
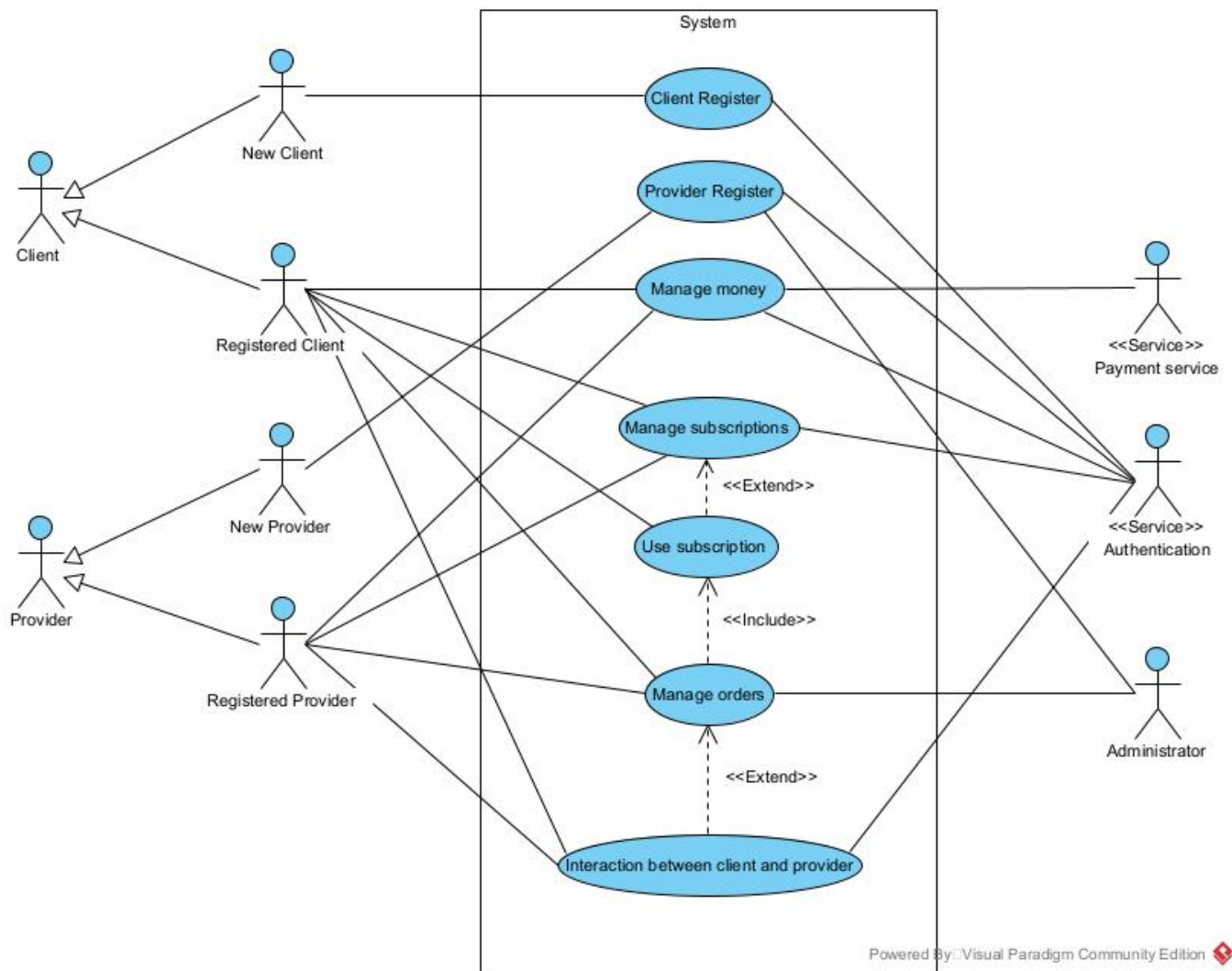
- Auth Service - управление пользователями
- Business Service - управление сервисами, провайдерами и подписками
- Chat Service - обеспечение возможности общения между клиентом и провайдером
- Notification Service - взаимодействие с внешним e-mail сервером
- Payment Service - обеспечение финансовых транзакций
- Frontend Service - сервер для SPA

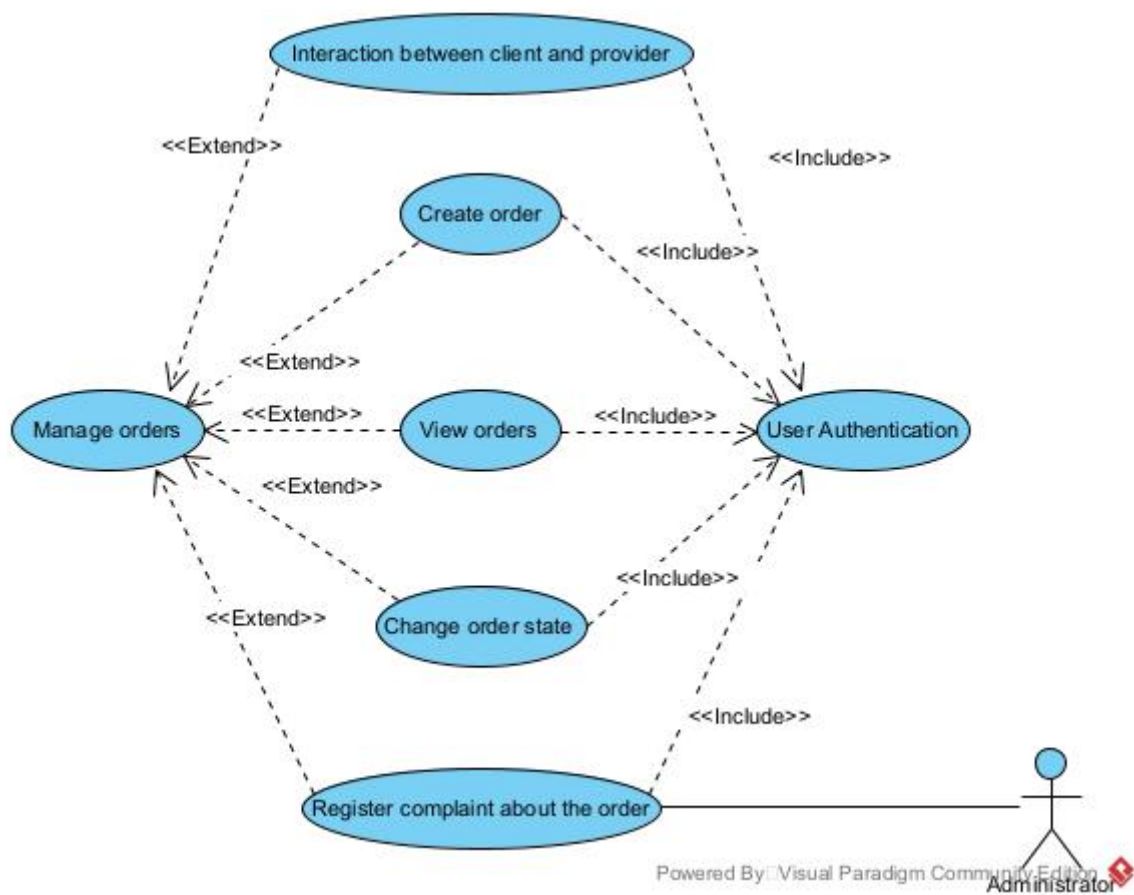
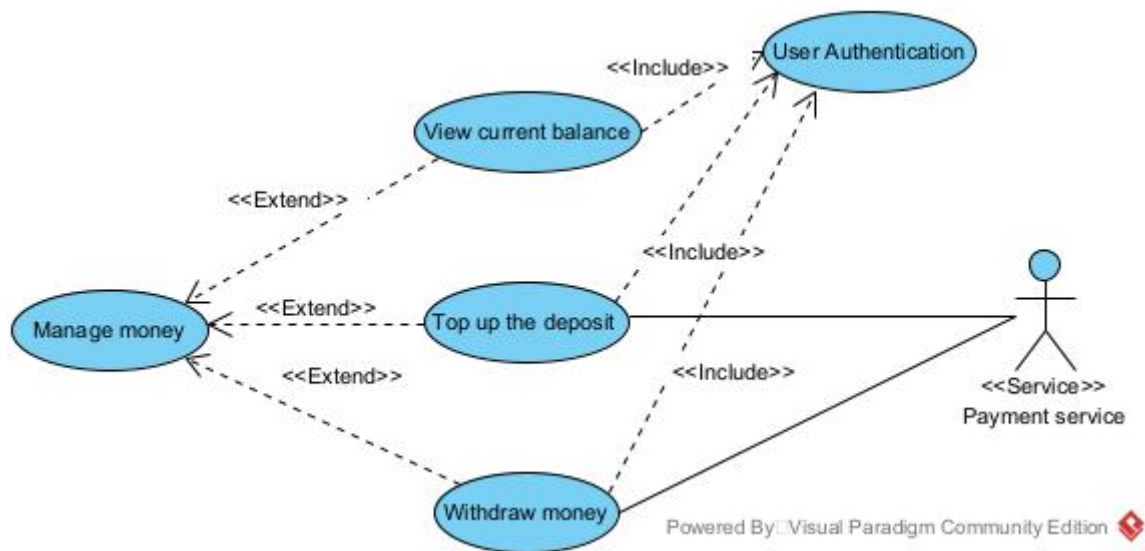
Для развертывания проекта будет использован небольшой Kubernetes кластер, состоящий из 3 машин. Для stateful приложений, таких как PostgreSQL, на четвертой машине будет развернут контейнеризованный NFS-сервер. Для асинхронного взаимодействия будет развернут RabbitMQ внутри кластера. В качестве gateway будет использован NGINX Ingress Controller.

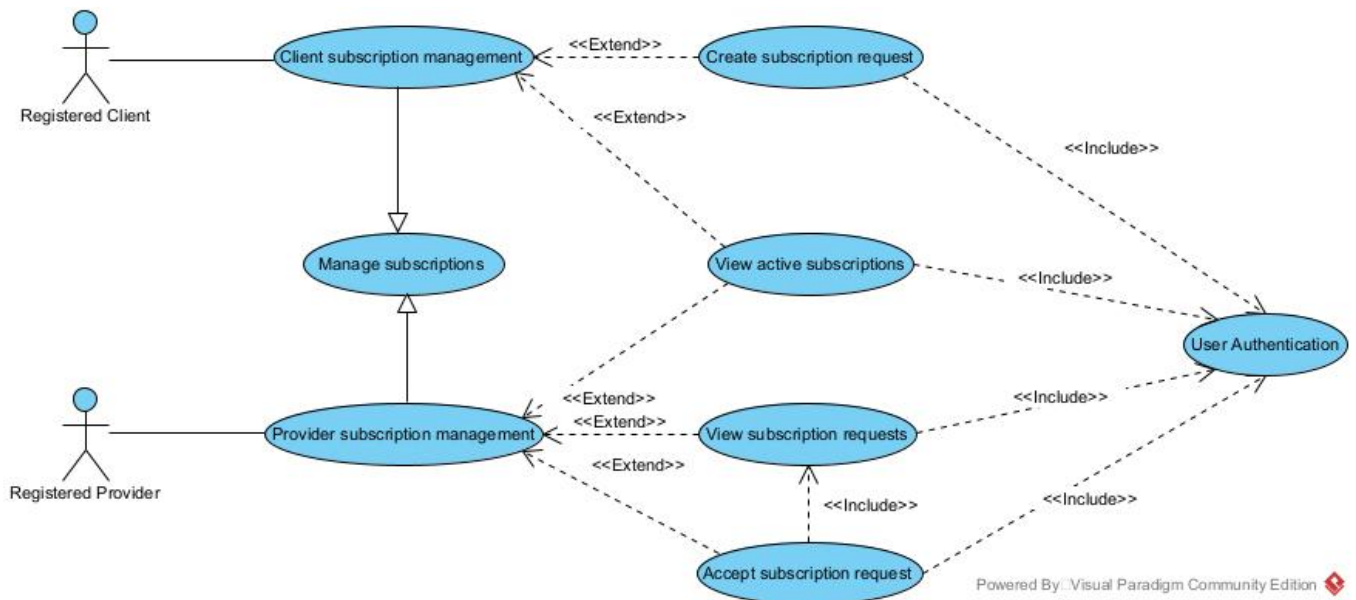
Также будет настроен CI/CD с использованием GitHub Actions & Github Container Registry.

## Diagrams

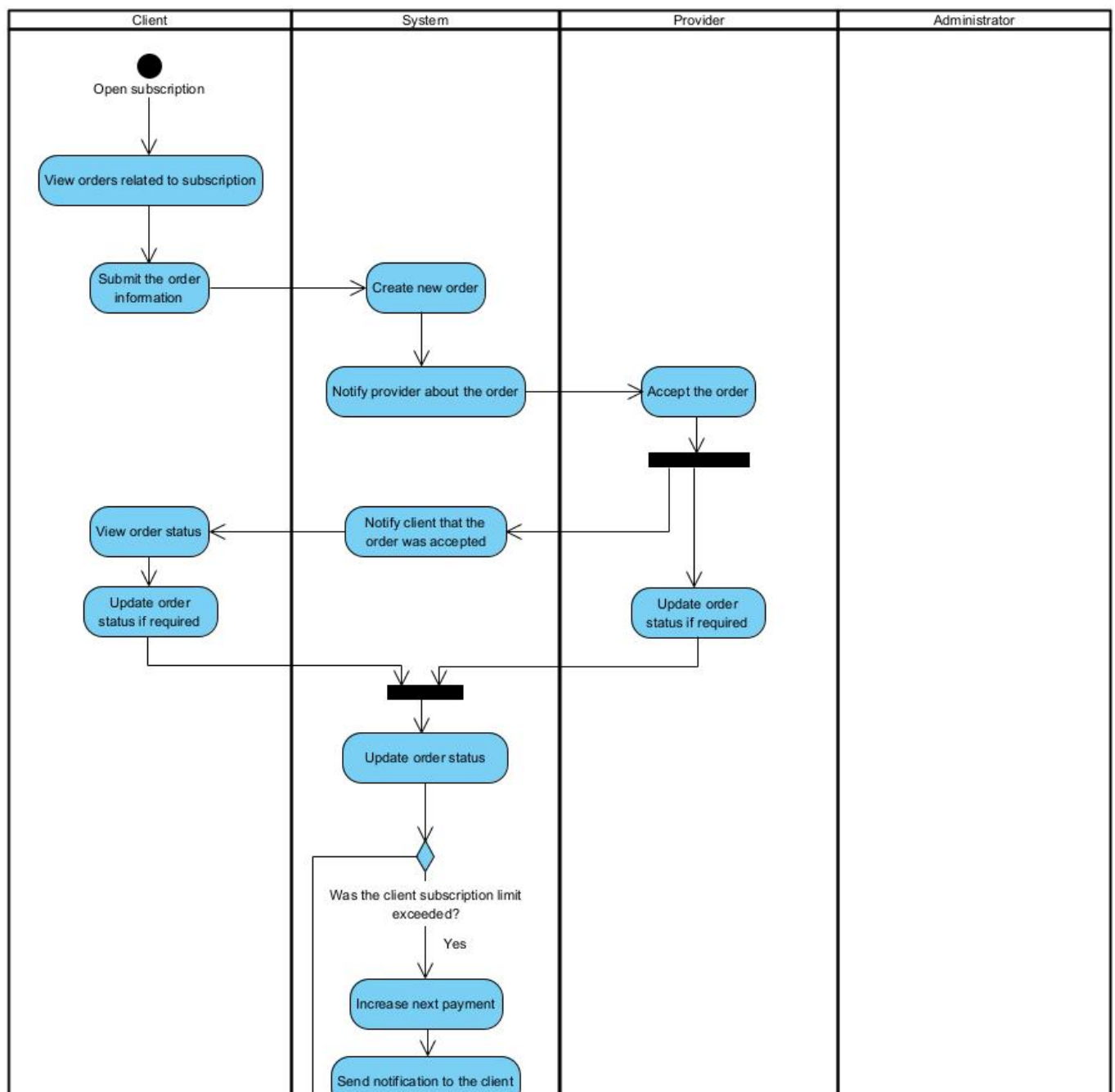
Use case diagrams

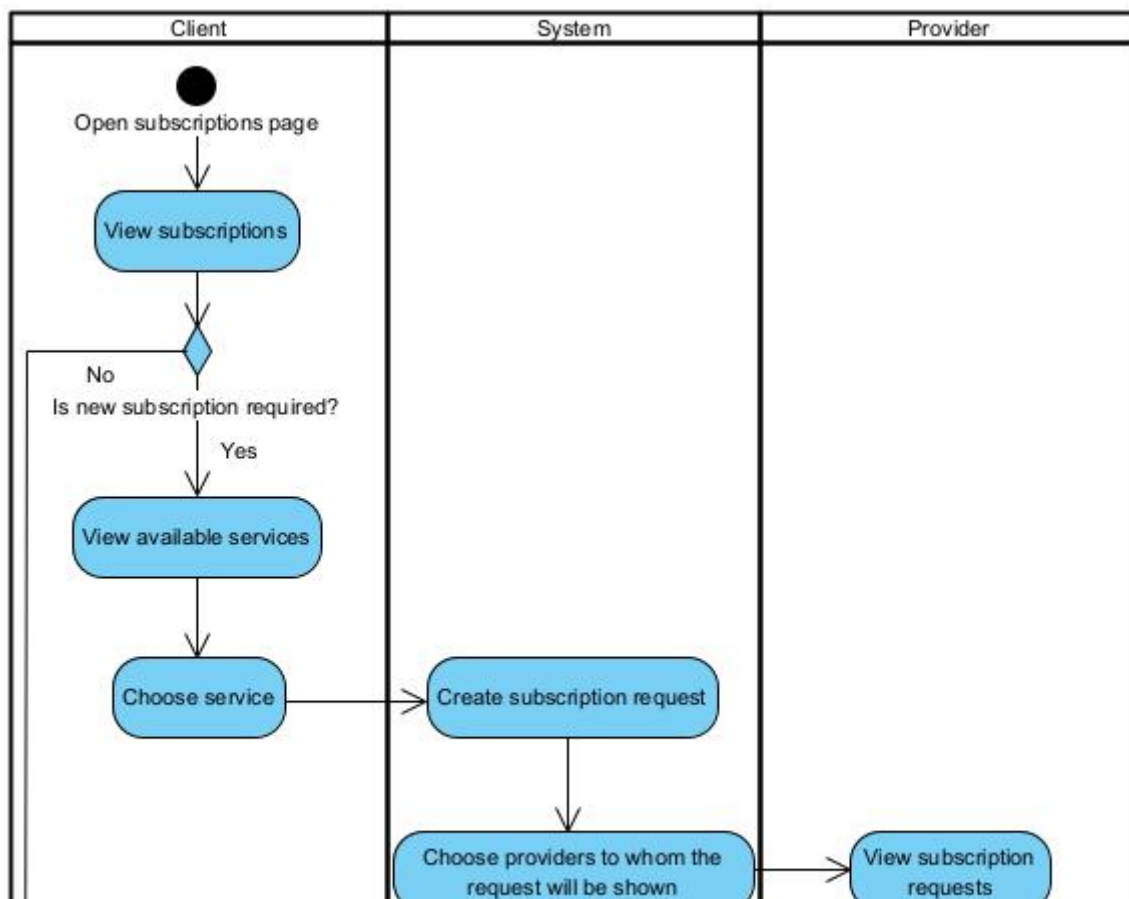
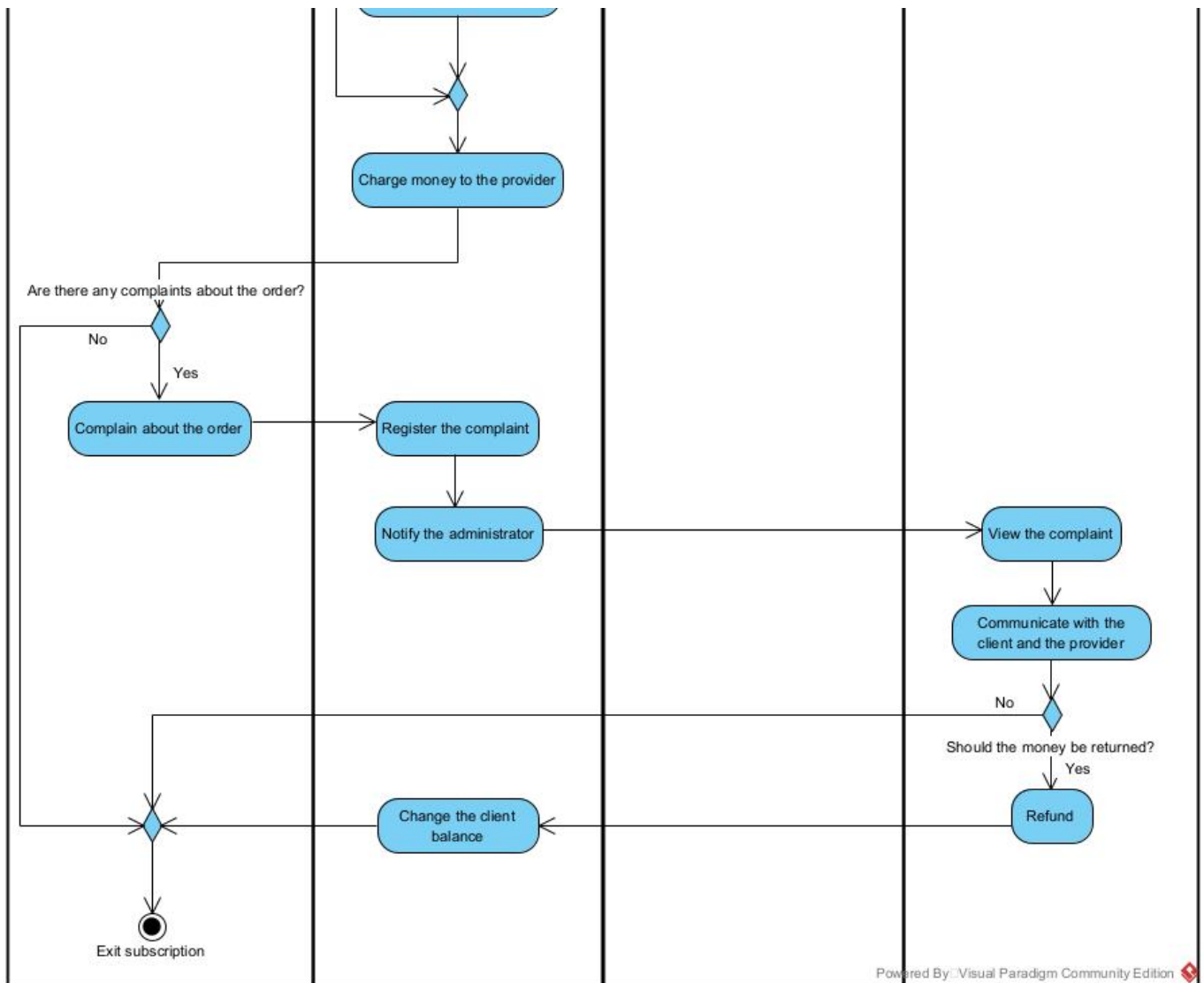


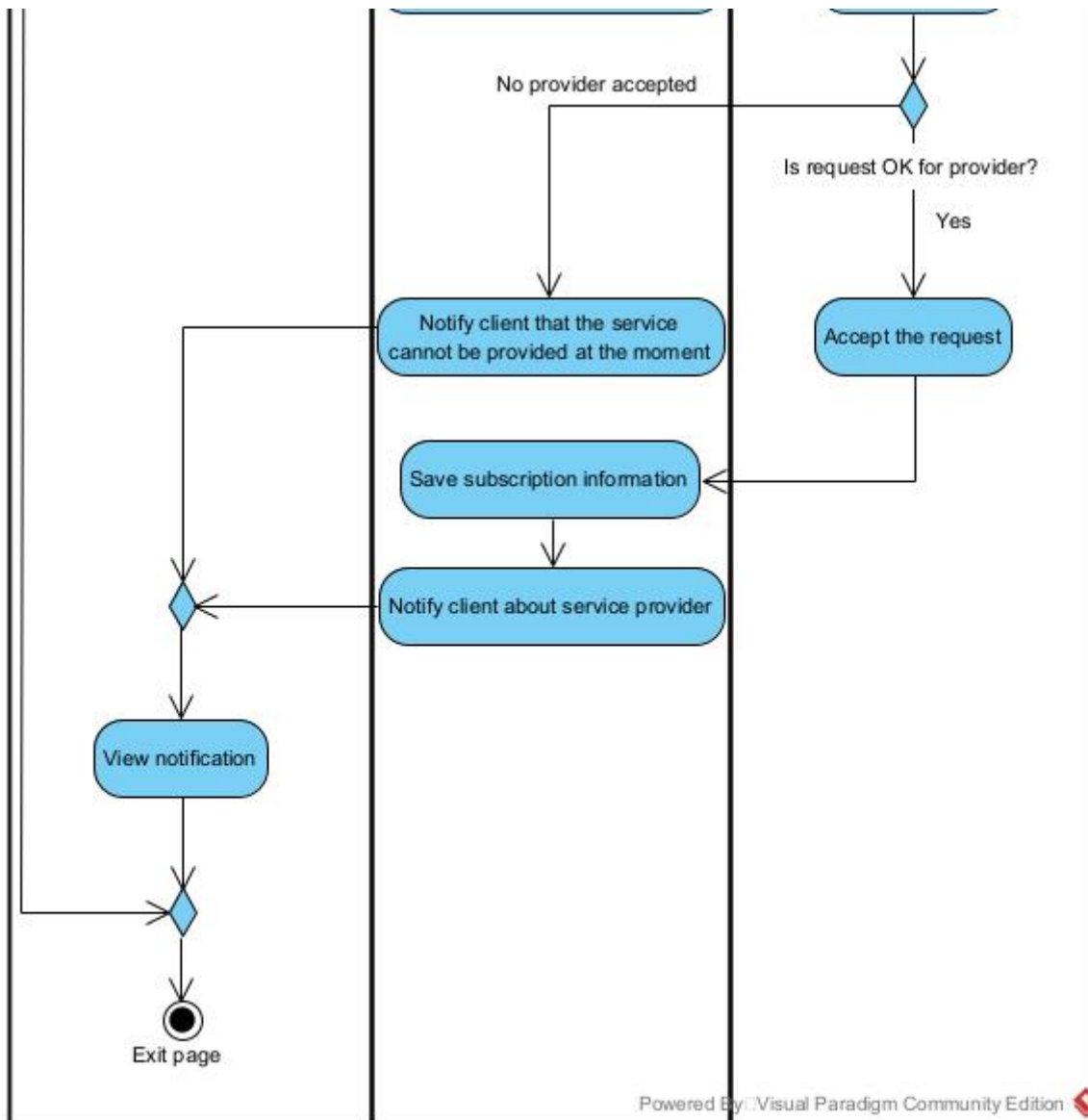


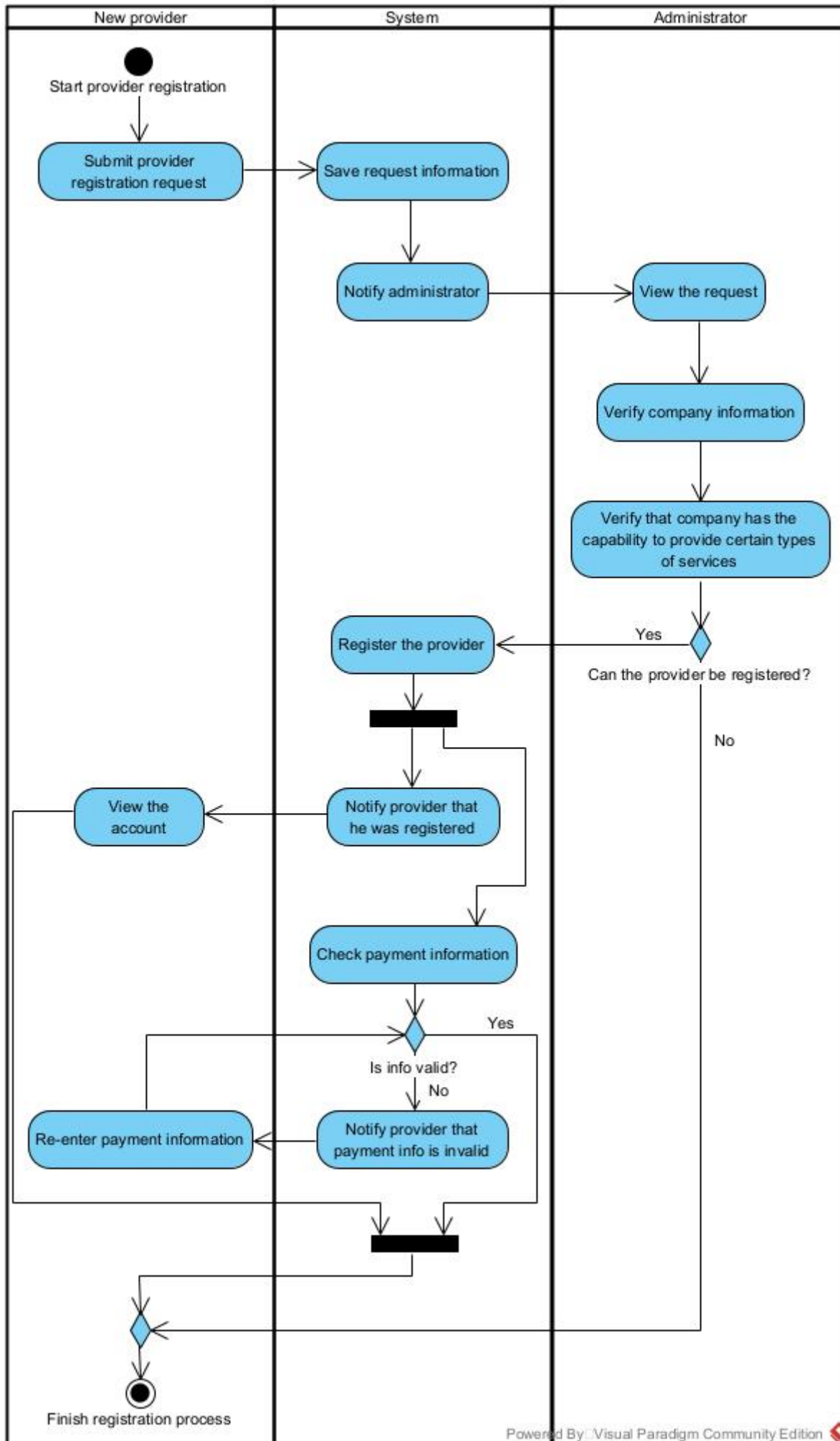


## Activity diagrams



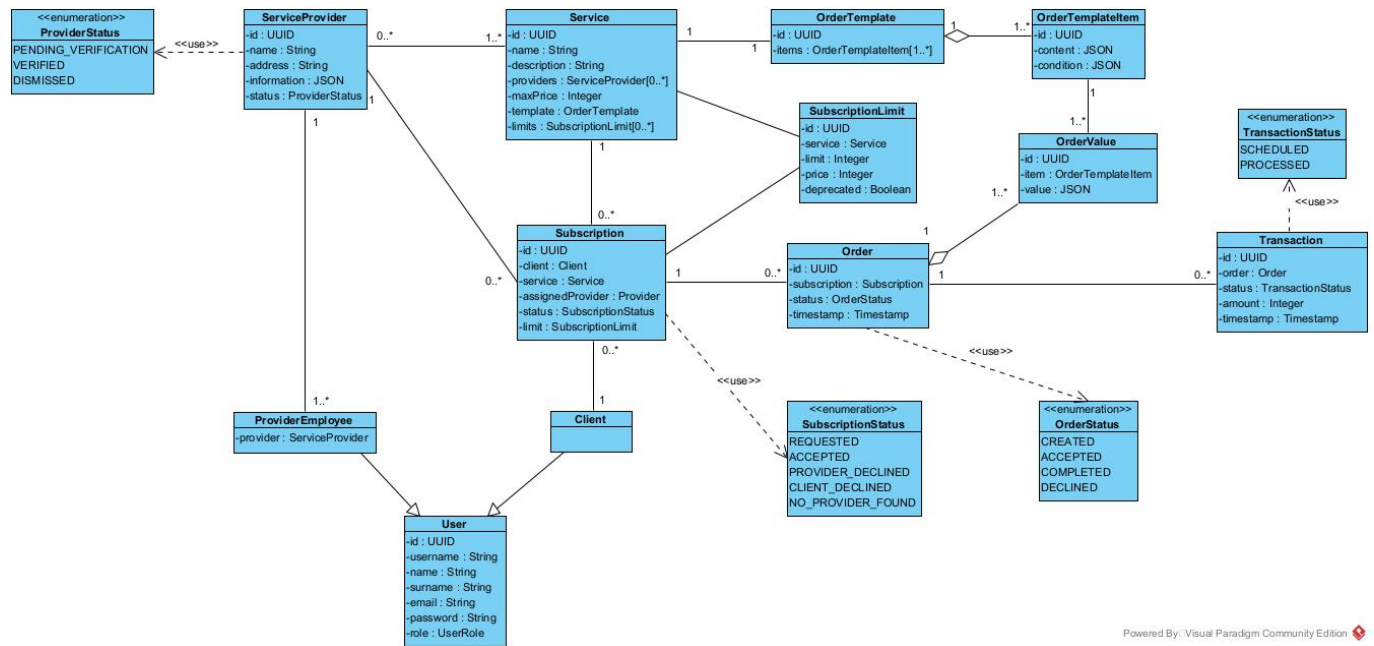






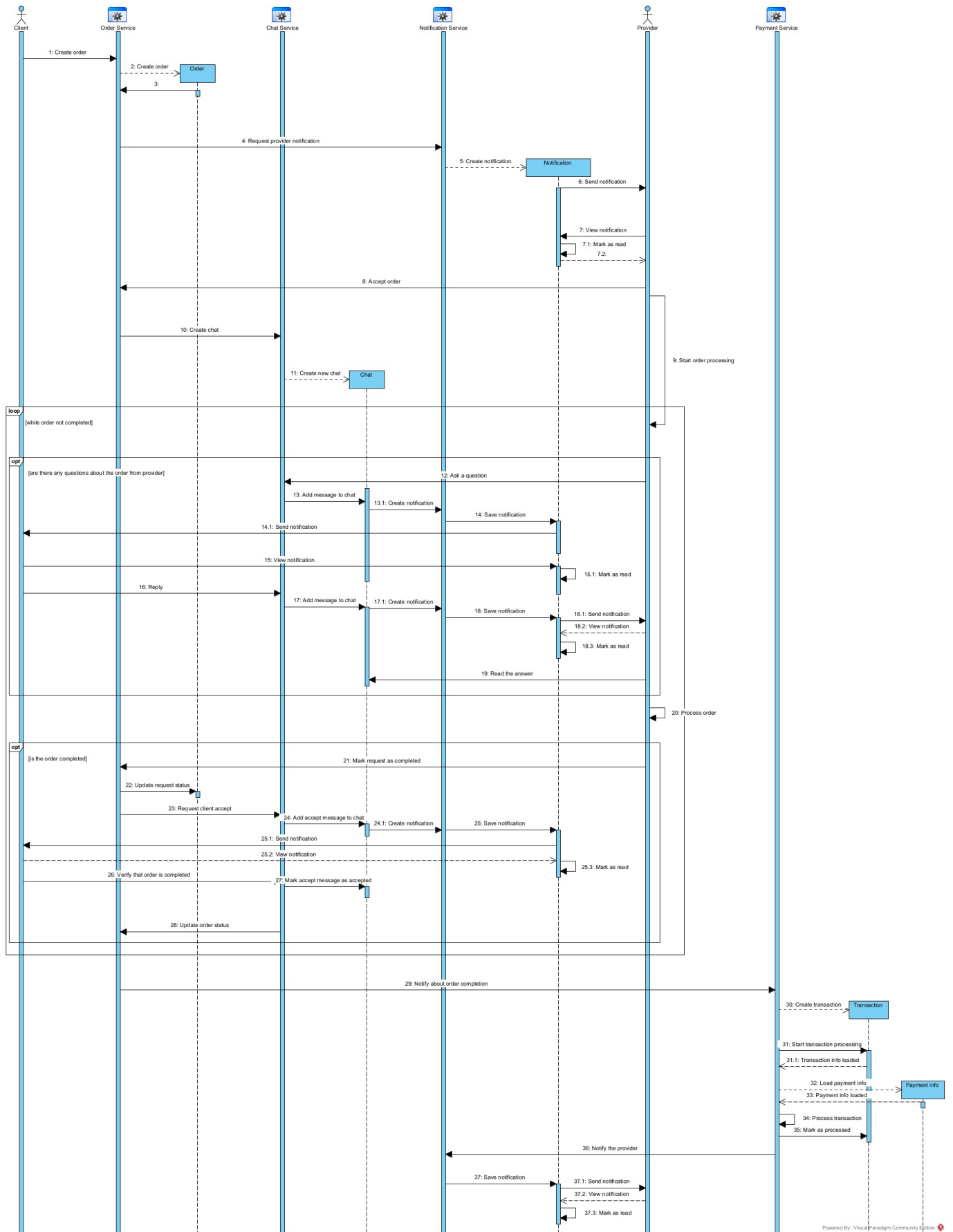


## Class diagrams



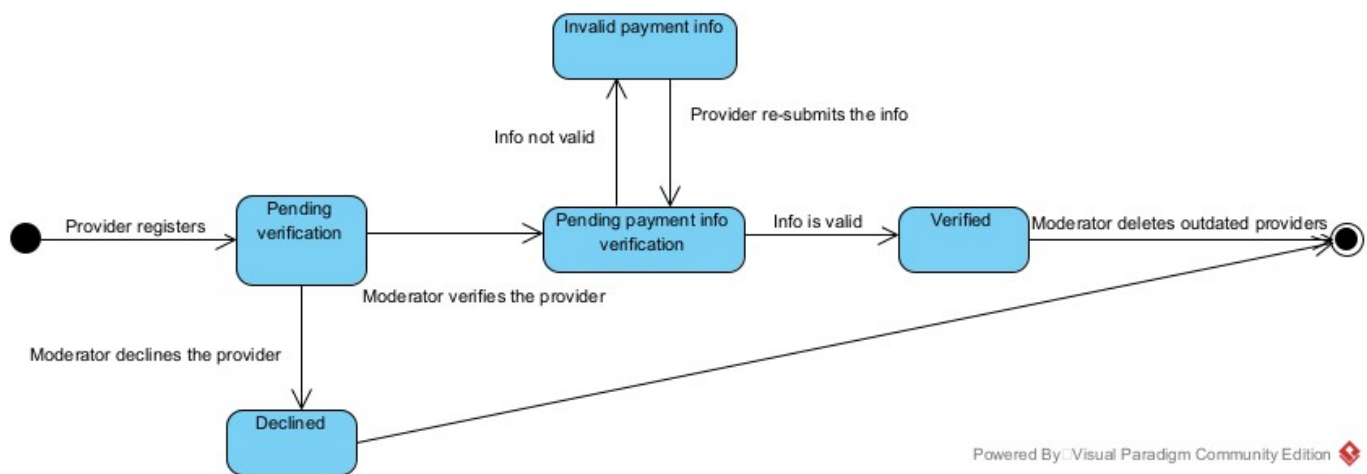
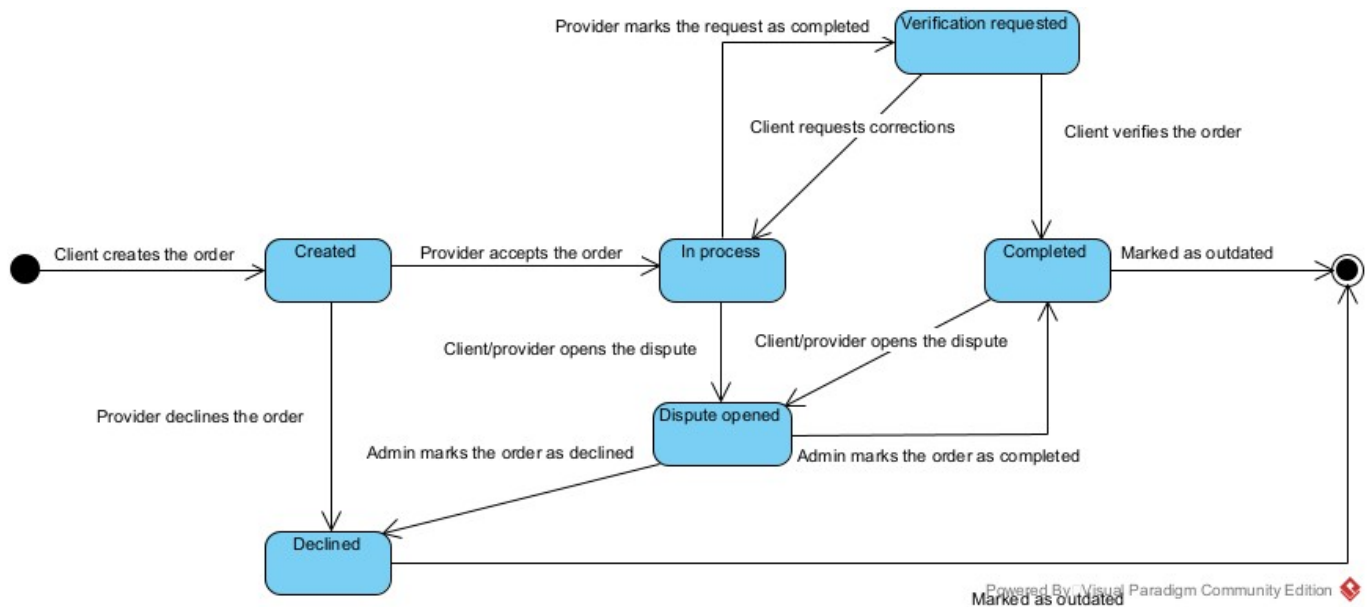
## Deployment diagrams



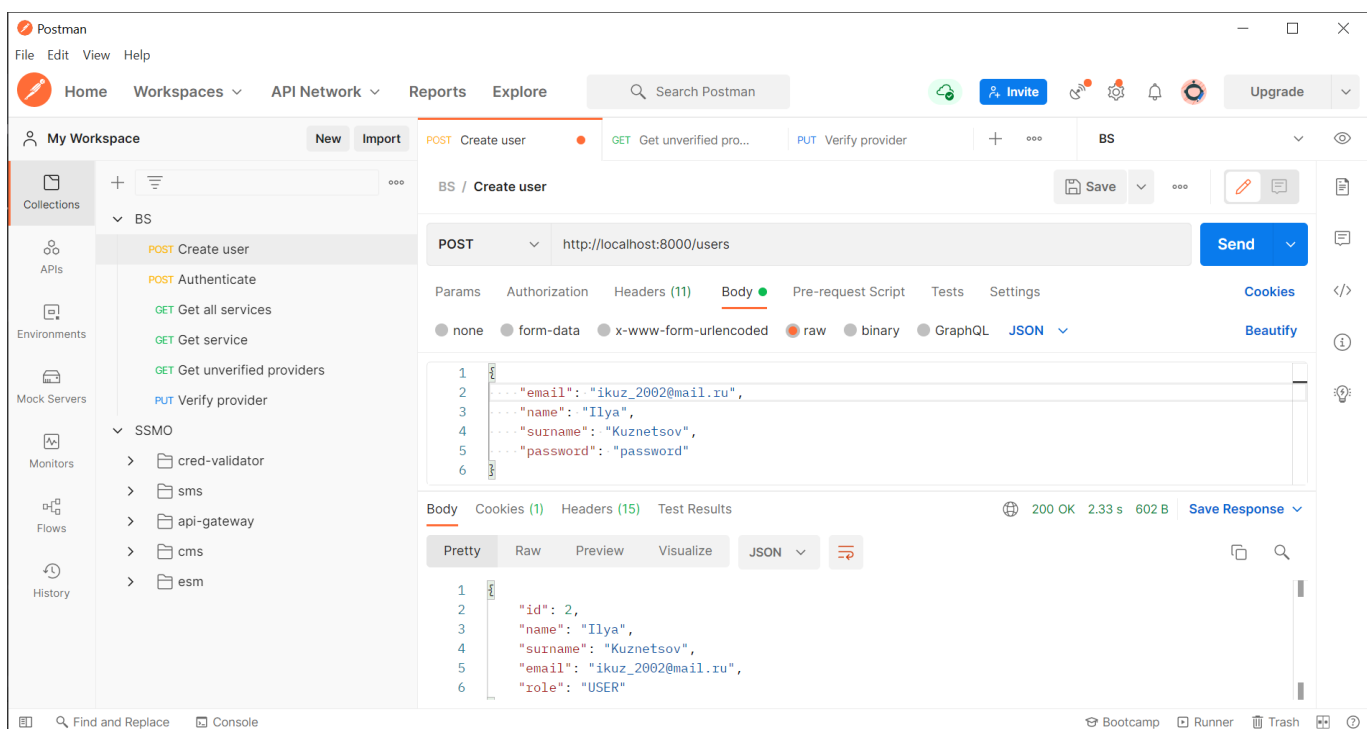


Powered By: VisualParadigm Community Edition

## State diagrams



## Screenshots



Postman

File Edit View Help

Home Workspaces API Network Reports Explore Search Postman

My Workspace New Import

POST Create us... POST Authentic... GET Get unverif... PUT Verify prov... BS

BS / Authenticate

POST http://localhost:8000/auth

Params Authorization Headers (11) Body Pre-request Script Tests Settings Cookies Beautify

none form-data x-www-form-urlencoded raw binary GraphQL JSON

```

1 {
2   "email": "ikuz_2002@mail.ru",
3   "password": "password"
4 }

```

Body Cookies (1) Headers (14) Test Results 200 OK 86 ms 640 B Save Response

Pretty Raw Preview Visualize JSON

```

1 {
2   "token": "eyJhbGciOiJIUzUxMiJ9.eyJzdWIiOiIyIiwicm9sZSI6IiVTRVIiLCJpYXQiOiJlY2MzYzUxODQsImV4cCI6MTY0MDc4NDc4NH0.4kzPxDkRqvCKmMeKh5Z14N0R8nbdxVA902aGH3Z37-3tZyAJvsYomHbQ941t1LgVB-_E3eu5N9FfsUD959xw"
3 }

```

Find and Replace Console Bootcamp Runner Trash

Postman

File Edit View Help

Home Workspaces API Network Reports Explore Search Postman

My Workspace New Import

GET Get all services BS

BS / Get all services

GET http://localhost:8080/services

Params Authorization Headers (9) Body Pre-request Script Tests Settings Cookies

Query Params

KEY	VALUE	DESCRIPTION	Bulk Edit
Key	Value	Description	

Body Cookies (1) Headers (15) Test Results 200 OK 2.34 s 584 B Save Response

Pretty Raw Preview Visualize JSON

```

1 {
2   "id": 1,
3   "name": "First service",
4   "description": "Sample description"
5 }
6
7

```

Find and Replace Console Bootcamp Runner Trash

Postman

File Edit View Help

Home Workspaces API Network Reports Explore

Search Postman

My Workspace New Import

BS / Get service

GET http://localhost:8080/services/1

Params Authorization Headers (9) Body Pre-request Script Tests Settings

Body Cookies (1) Headers (14) Test Results

Status: 200 OK Time: 24 ms Size: 679 B Save Response

Pretty Raw Preview Visualize JSON

```

1 {
2   "id": 1,
3   "name": "First service",
4   "description": "Sample description",
5   "limits": [
6     {
7       "id": 1,
8       "limit": 100,
9       "price": 10000
10    }
11  ],
12  "template": {
13    "id": 1,
14    "items": [
15      {
16        "id": 1,
17        "content": "Sample text",
18        "type": "TEXT"
19      },
20      {
21        "id": 2,
22        "content": "Sample input",
23        "type": "INPUT"
24      }
25    ]
26  }
27 }

```

Find and Replace Console

Bootcamp Runner Trash

Postman

File Edit View Help

Home Workspaces API Network Reports Explore

Search Postman

My Workspace New Import

BS / Get unverified providers

GET http://localhost:8080/admin/providers/unverified

Params Authorization Headers (9) Body Pre-request Script Tests Settings

Headers 9 hidden

KEY	VALUE	DESCRIPTION
Key	Value	Description

Body Cookies (1) Headers (14) Test Results

Status: 200 OK Time: 14 ms Size: 561 B Save Response

Pretty Raw Preview Visualize JSON

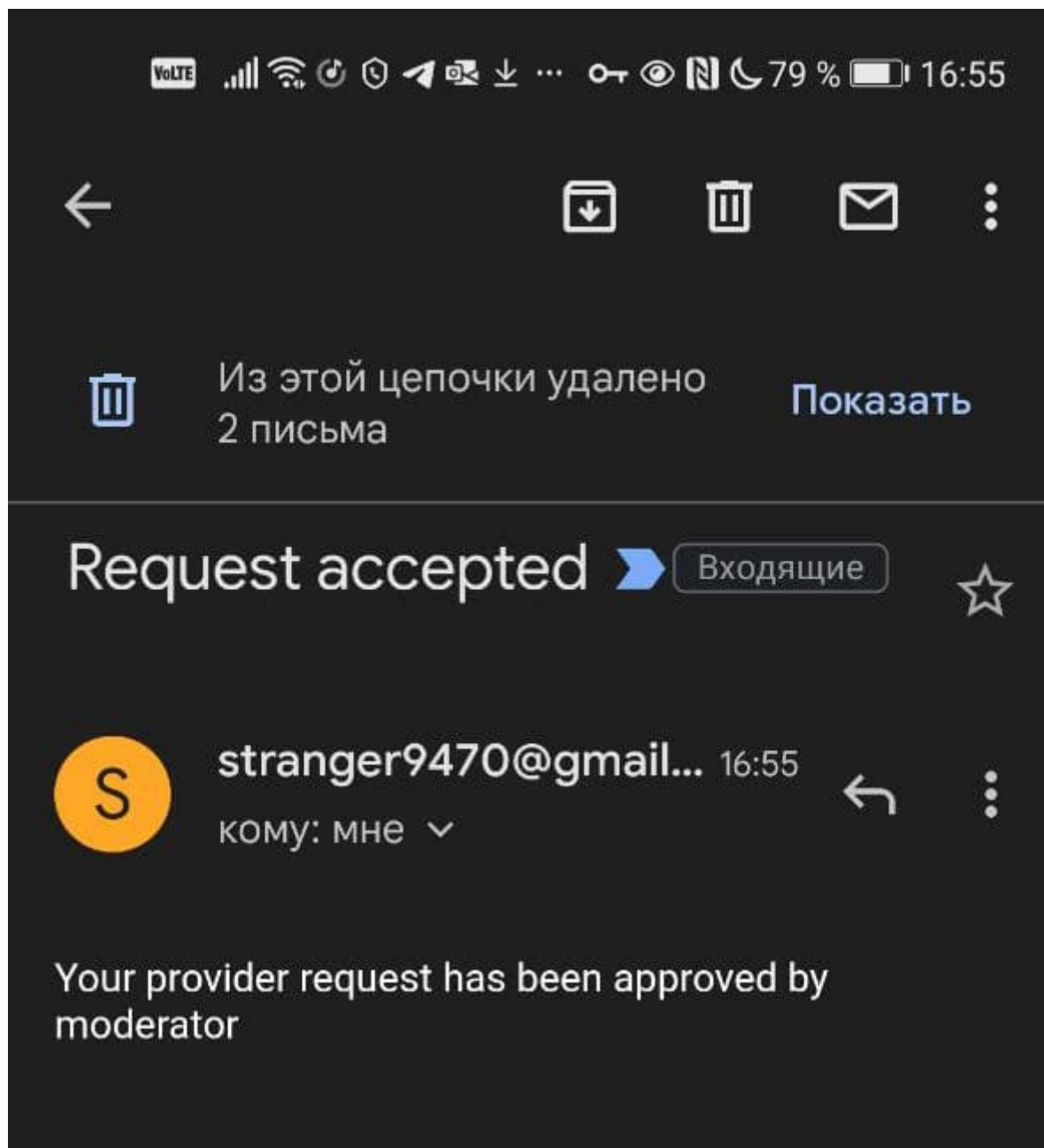
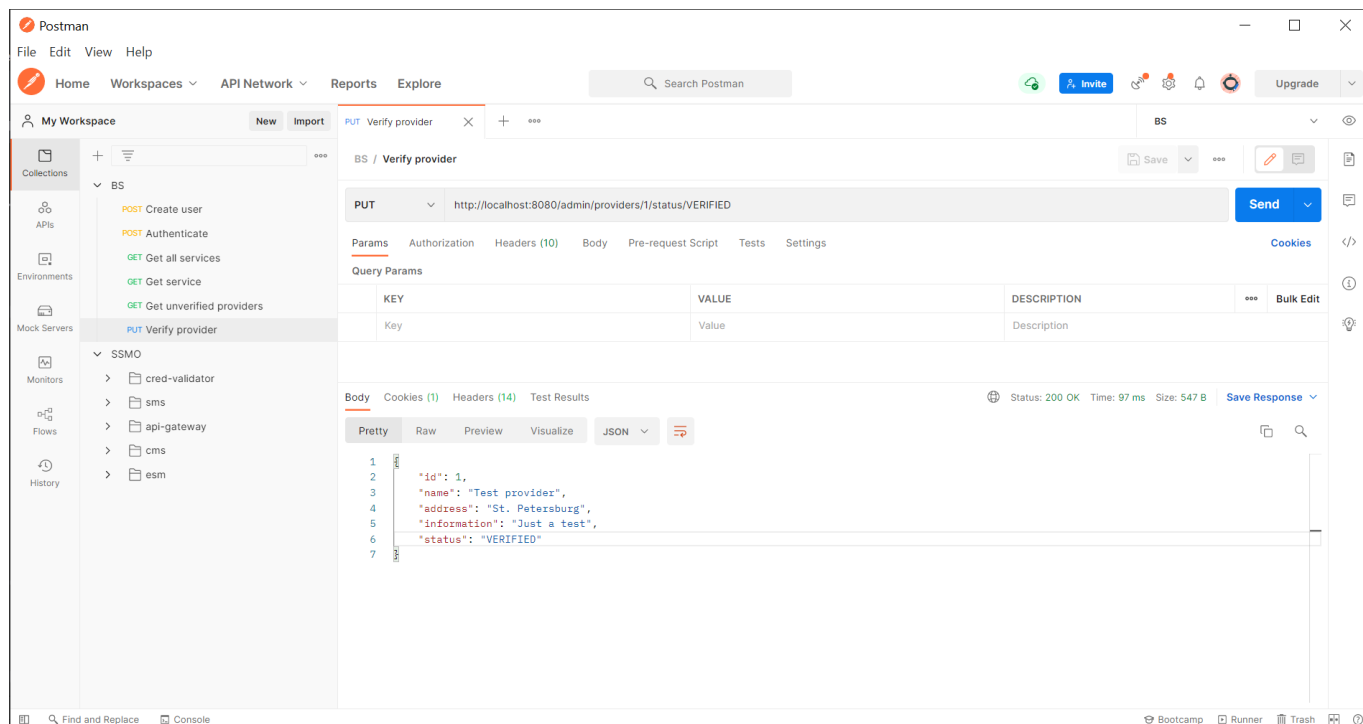
```

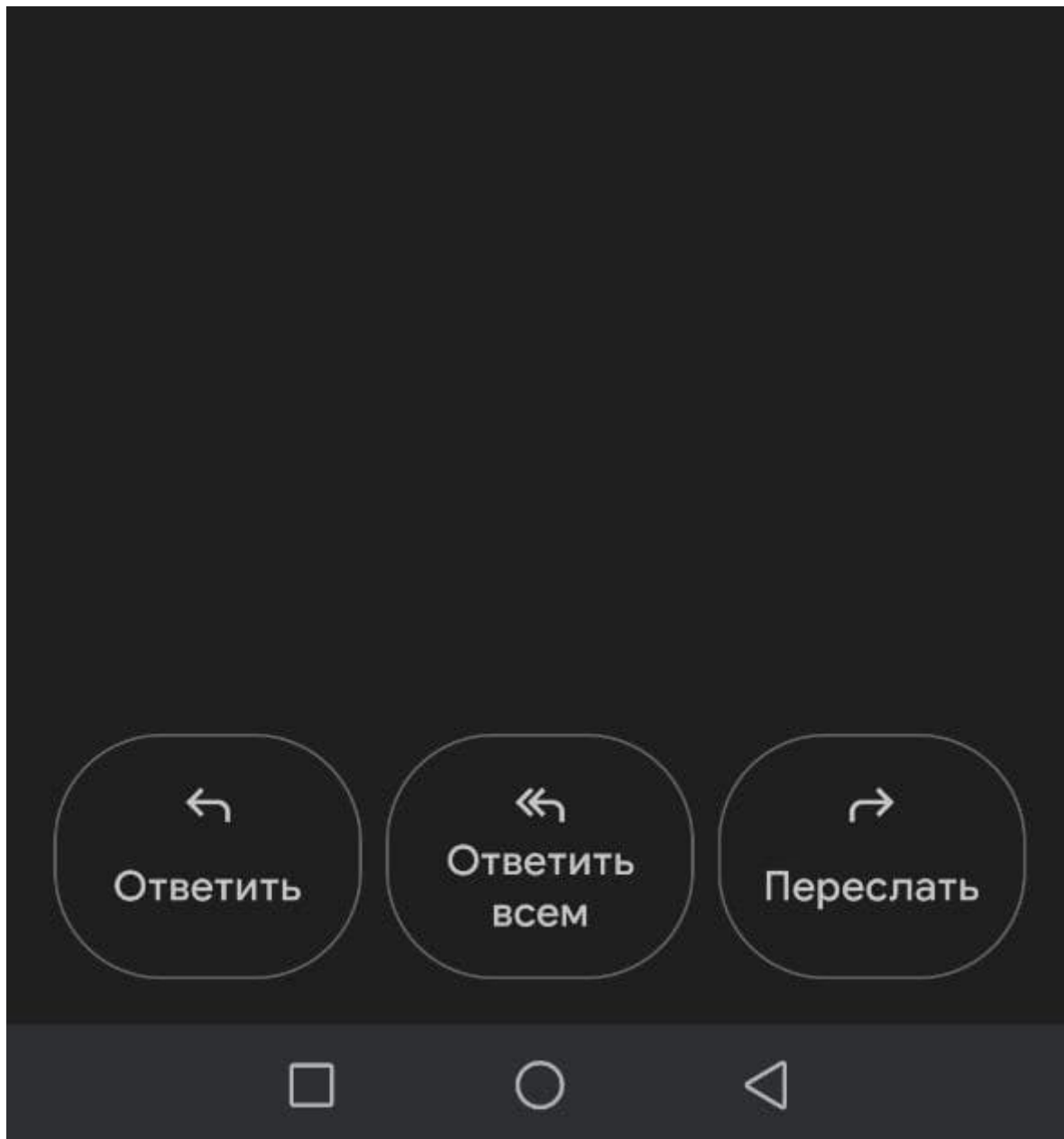
1 [
2   {
3     "id": 1,
4     "name": "Test provider",
5     "address": "St. Petersburg",
6     "information": "Just a test",
7     "status": "PENDING_VERIFICATION"
8   }
9 ]

```

Find and Replace Console

Bootcamp Runner Trash





Main

localhost:15672/#/queues

Refreshed 2021-12-15 16:58:07
Refresh every 5 seconds
Virtual host All
Cluster rabbit@9622b6fab14f
User guest
Log out

RabbitMQ
RabbitMQ 3.8.18
Erlang 24.0.3

Overview
Connections
Channels
Exchanges
Queues
Admin

Queues

All queues (3)

Pagination

Page 1 of 1 - Filter:
☐ Regex ?
Displaying 3 items , page size up to: 100

Overview				Messages			Message rates			
Name	Type	Features	State	Ready	Unacked	Total	incoming	deliver / get	ack	
notifications.email	classic	<span>D</span>	idle	0	0	0	0.00/s	0.00/s	0.00/s	
users.created	classic	<span>D</span>	idle	0	0	0	0.00/s	0.00/s	0.00/s	
users.requests	classic	<span>D</span>	idle	0	0	0				

Add a new queue

[HTTP API](#)
[Server Docs](#)
[Tutorials](#)
[Community Support](#)
[Community Slack](#)
[Commercial Support](#)
[Plugins](#)
[GitHub](#)
[Changelog](#)



```
[main] c.g.k.b.n.service.UserService : Created new connection: 10.151.1.100:8080
[main] c.g.k.b.n.NotificationServiceApplication : Started NotificationServiceApplication in 10.151 sec
[ntContainer#0-1] c.g.k.b.n.service.UserService : User 1 with email ikuz2002@gmail.com saved
[ntContainer#1-1] c.g.k.b.n.service.MailSenderService : User 1 notified: Request accepted
[ntContainer#0-1] c.g.k.b.n.service.UserService : User 3 with email user1@user.com saved
[ntContainer#0-1] c.g.k.b.n.service.UserService : User 4 with email user2@user.com saved
[ntContainer#0-1] c.g.k.b.n.service.UserService : User 5 with email user3@user.com saved
```