Университет ИТМО, факультет ПИиКТ

Лабораторная работа №1
Дисциплина: Бизнес-логика программных систем
Вариант 413

Выполнили: Чангалиди Антон

Чайка Алексей

Группа: P33122

Преподаватель:
Пашнин Александр Денисович
Цопа Евгений Алексеевич

г. Санкт-Петербург

2020 г.

Задание

Вариант №413: Minecraft Inside - http://minecraft-inside.ru/

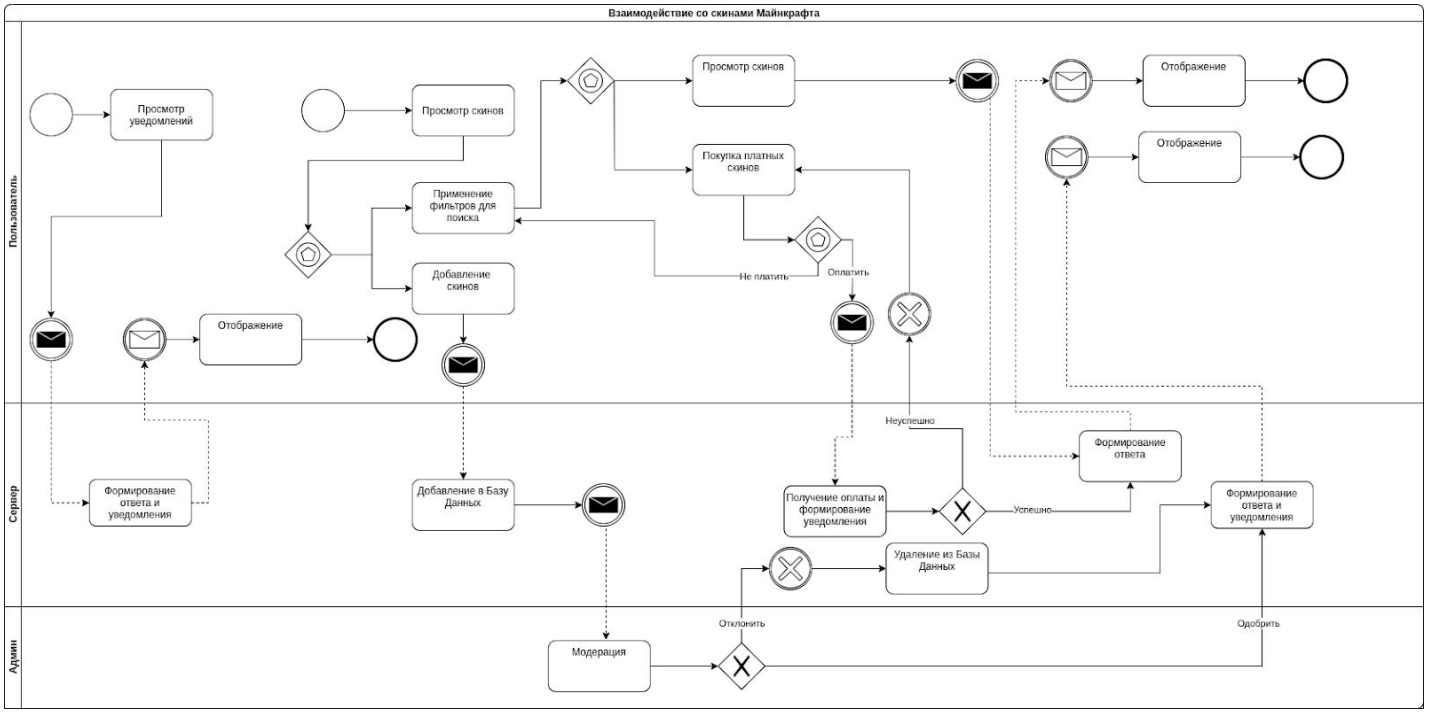
Описать бизнес-процесс в соответствии с нотацией BPMN 2.0, после чего реализовать его в виде приложения на базе Spring Boot.

Порядок выполнения работы:

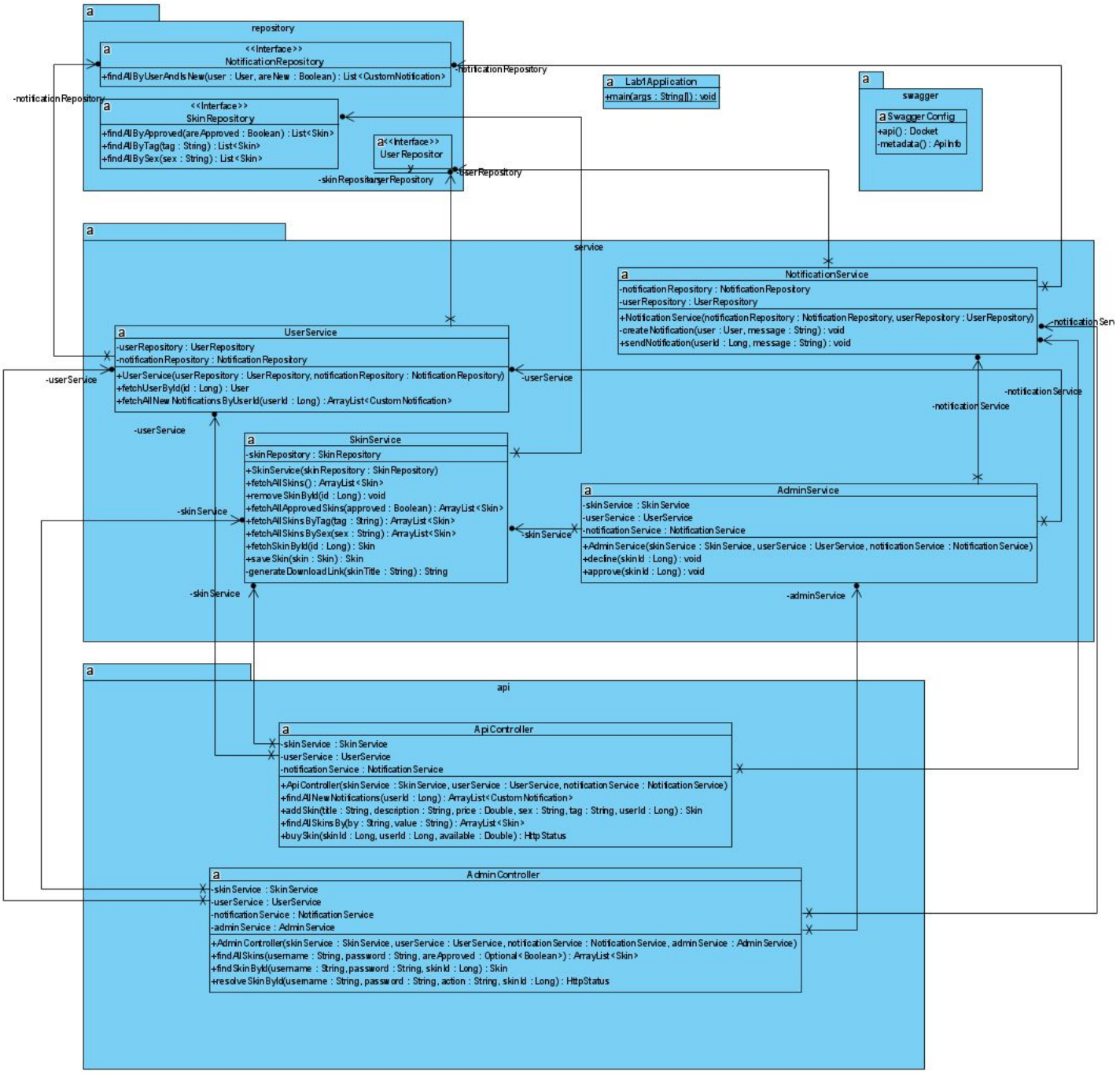
1. Выбрать один из бизнес-процессов, реализуемых сайтом из варианта задания.
2. Утвердить выбранный бизнес-процесс у преподавателя.
3. Специфицировать модель реализуемого бизнес-процесса в соответствии с требованиями BPMN 2.0.
4. Разработать приложение на базе Spring Boot, реализующее описанный на предыдущем шаге бизнес-процесс. Приложение должно использовать СУБД PostgreSQL для хранения данных, для всех публичных интерфейсов должны быть разработаны REST API.
5. Разработать набор curl-скриптов, либо набор запросов для REST клиента Insomnia для тестирования публичных интерфейсов разработанного программного модуля. Запросы Insomnia оформить в виде файла экспорта.
6. Развернуть разработанное приложение на сервере helios.

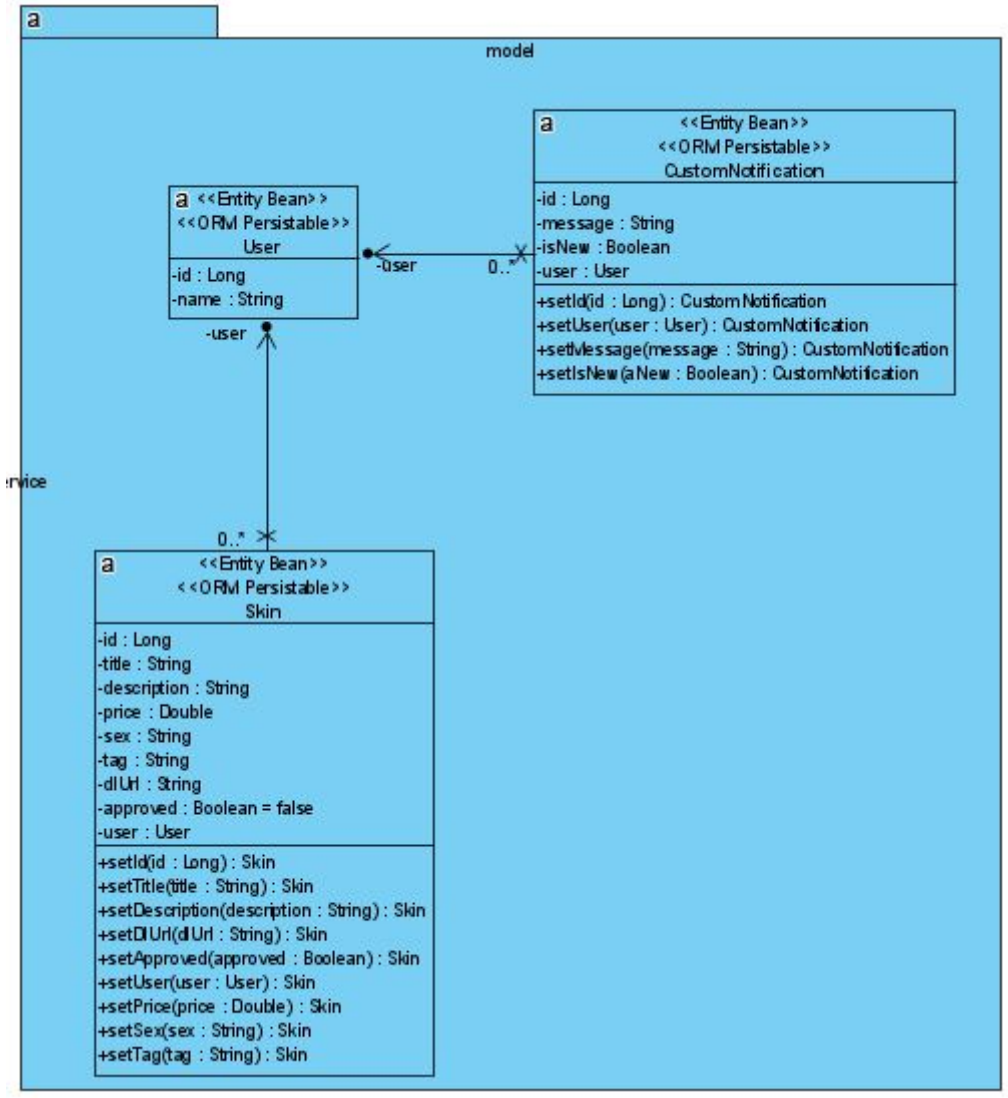
Бизнес-процесс

Бизнес процесс - взаимодействие со скинами (добавление с модерацией, просмотр, покупка с проверкой).



UML (diag_1.png, diag_2.png)

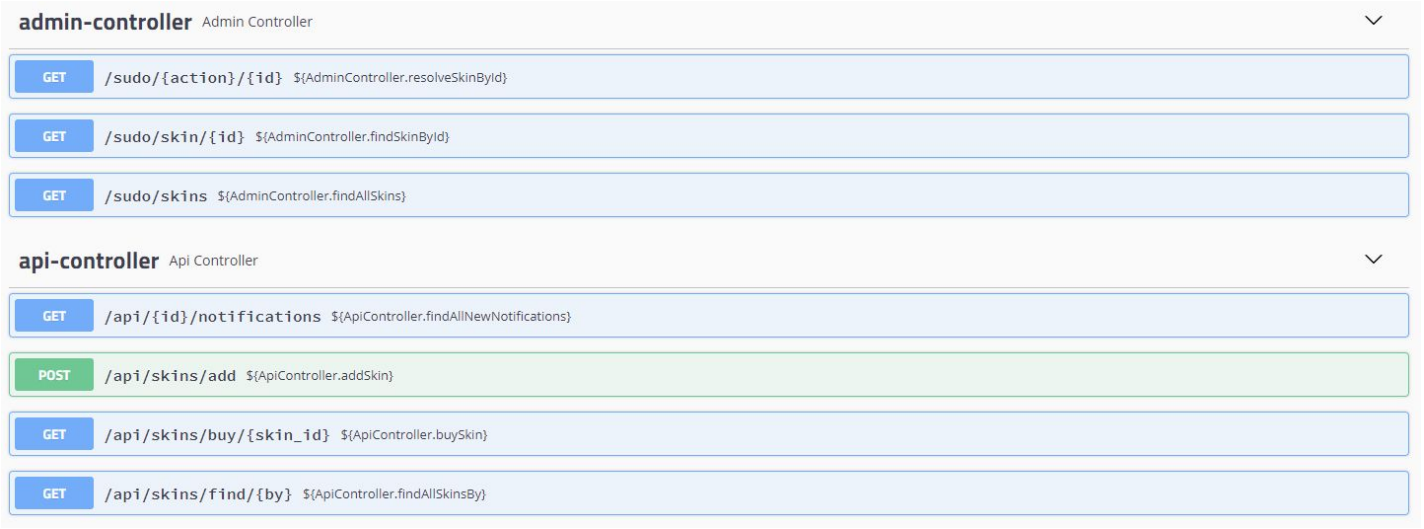



a
model
a
<<Entity Bean>>
<<ORM Persistable>>
CustomNotification
-id : Long
-message : String
-isNew : Boolean
-user : User
+setId(id : Long) : CustomNotification
+setUser(user : User) : CustomNotification
+setMessage(message : String) : CustomNotification
+setIsNew(aNew : Boolean) : CustomNotification
a
<<Entity Bean>>
<<ORM Persistable>>
User
-id : Long
-name : String
-user
0..*
-user
0..*
a
<<Entity Bean>>
<<ORM Persistable>>
Skin
-id : Long
-title : String
-description : String
-price : Double
-sex : String
-tag : String
-dlUrl : String
-approved : Boolean = false
-user : User
+setId(id : Long) : Skin
+setTitle(title : String) : Skin
+setDescription(description : String) : Skin
+setDlUrl(dlUrl : String) : Skin
+setApproved(approved : Boolean) : Skin
+setUser(user : User) : Skin
+setPrice(price : Double) : Skin
+setSex(sex : String) : Skin
+setTag(tag : String) : Skin

Спецификация REST API

Есть два пути для взаимодействия с API:

- для юзера (добавление скинов, их покупка, поиск и фильтрация, а также просмотр уведомлений о прошлых покупках и добавлениях);
- для администратора (вывести все скины (добавленные и не добавленные), или какой-то конкретный, а также их модерация).



Примеры cURL-запросов

Взаимодействие администратора со скином (approve/decline):

`/sudo/{action}/{id}` code

```
curl -X GET "http://localhost:8080/sudo/decline/2?p=admin&u=admin" -H "accept: */*"
```

Просмотр администратором конкретного скина по id:

`/sudo/skin/{id}` code

```
curl -X GET "http://localhost:8080/sudo/skin/3?p=admin&u=admin" -H "accept: */*"
```

Просмотр администратором неподтвержденных скинов (ожидающих модерации):

`/sudo/skins` code

```
curl -X GET "http://localhost:8080/sudo/skins?approved=false&p=admin&u=admin" -H "accept: */*"
```

Просмотр списка уведомлений для конкретного пользователя:

`/api/{id}/notifications` code

```
curl -X GET "http://localhost:8080/api/1/notifications" -H "accept: */*"
```

Добавление скина пользователем, который отправляется на модерацию:

`/api/skins/add` code

```
curl -X POST
"http://localhost:8080/api/skins/add?description=desc&price=150&sex=male&tag=MEGA&ti
tle=new%20skin&user_id=1" -H "accept: */*"
```

Покупка скина пользователем по его ID:

`/api/skins/buy/{skin_id}` code

```
curl -X GET "http://localhost:8080/api/skins/buy/3?available=150&user_id=1" -H
"accept: */*"
```

Поиск пользователем скина с возможным условием:

`/api/skins/find/{by}` code

```
curl -X GET "http://localhost:8080/api/skins/find/sex?value=male" -H "accept: */*"
```

Исходный код

https://github.com/trafalgande/ITMO_BUSINESSLOGIC_LAB_1

Выводы

Выполнив лабораторную работу, мы научились сначала думать над проектом и проектировать его, а затем реализовывать, что зачастую помогает избежать много ошибок на ранних этапах реализации проекта и многократного переделывания.