Уеб Проект:

Регистрация за нормални юзъри;

-дроп меню за вид агенагенцията:

<https://ntr.tourism.government.bg/Registration.nsf/alltato.xsp>

STEP1:

* Username;
* Password;
* Email;
* FirstName, LastName;
* phoneNumber;

STEP 2:

* Лична карта Снимки;
* селфи снимка.

STEP 3:

* Удостоверителен номер;
* Име на фирмата;
* ЕИК;
* Населено място;
* Адрес по регистрация;
* Офиси;
* Прикачване на документ ( разрешително);

Опционално: Лицето което регистрира трябва да изпрати селфи + лична карта снимка за верификация.  
  
 Трябва да има 1 админ юзър:

В админ панела трябва да излизат всички нови опити за регистрация като прозорчета

Под формата на списък, при натискане на прозорчето да се разширява и отпечатва цялата информация, трябва да има бутончета за удобряване и отхвърляне на регистрацията.

При удобряване, регистрацията трябва да се запише при нормалните акаунти, които вече са активни, по този начин меню с възможност добавяне на оферти ще излезе и акаунта ще може да бъде функционален.

В противен случай при влизане в акаунта ще излиза бутон за добавяне на снимките/файловете

За повторна регистрация, или цялостната форма за регистрация за попълване на данните отново.

При успешна регистрация, юзърите ще могат да създават оферти за пътувания, като ще има предварителен текстов шаблон, който да попълнят:

Държава:

Дестинация:

Звезди:

Времетраене:

Вид транспорт:

Цена за човек:

Отстъпки:

Текст: - свободен текст за описание

База Данни:

@MappedSuperclass

1.BaseEntity:

id;

@MappedSuperclass

2.BaseUser extends BaseEntity;

userName;

password;

3.AdminUser extends BaseUser;

firstName; -> nullable

lastName; -> nullable

Set<UsersForApproval>

4.RegularUser extends BaseUser:

Dto incoming info {

firstName; -> notNull;

lastName; -> notNull;

phoneNumber; -> notNull;

idCardPictureFront; -> notNull

idCardPictureBack; -> notNull

selfiePictureForApproval; -> notNull;

}

credentialNumber; -> nullable;

companyName; -> nullable;

uniqueIdentifier; -> nullable;

registrationAddress; -> nullable;

List<String> Offices; -> nullable;

licenceDocument; -> nullable;

@OneToMany

List/Set<TravelOffer> addedOffers; nullable -> трябва ли да е двойна връзката

И от офертите към самия акаунт??

5.TravelOffer extends BaseEntity

Country;

Destination;

HotelName;

Stars;

TravelOfferTime;

Transport;

Money;

Discounts;

Description;

UsersForApproval extends RegularUser

или е по-добре да има само users при които има boolean в който се вижда дали е админ или не е, вместо да се отделя в отделна таблица. Адекватно ли е всички регистрации да са на едно място admin/regular/forApproval?

друга идея е да се пълни таблица отделна и от сорта на users\_for\_approval и след одобрение да се мести при users или това не е оферта??

fetch = FetchType.LAZY, cascade = CascadeType.ALL