Симулятор службы поддержки клиентов веб-сервиса

Дополнительные материалы:

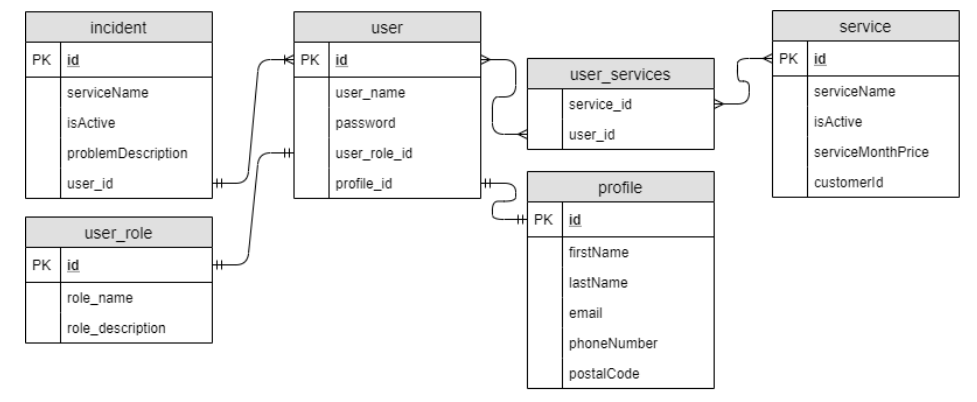
[hibernate documentation](https://hibernate.org/orm/documentation/6.0/)

[maven repository](https://mvnrepository.com/)

[пример создания приложения с hibernate](https://www.tutorialspoint.com/hibernate/hibernate_annotations.htm)

Требования:

1. Спроектировать БД согласно схеме и реализовать отношения между сущностями (аннотированными классами) с помощью hibernate;



1. Типы данных в таблицах определить самостоятельно и быть готовым объяснить их применение;
2. Таблица «user» – пользователь сервиса, в пользователя могут быть роли «USER», «ADMIN», «SUPER\_ADMIN» роли должны хранится в таблице «user\_role», «profile» – профиль пользователя, «service» – услуги которыми пользуются пользователи, «user\_services» – связная таблица для «user» и «service», «incident» – возникший инцидент, поломка сервиса (тикет на который должен ответить тех специалист);
3. Связи между сущностями hibernate и БД таблицами:

* один «user» к одному «profile»;
* один «user» ко многим «incident»;
* один «user» к одному «user\_role»;
* многие «user» ко многим «service» (каждый пользователь может пользоваться множеством услуг и в тоже время каждая услуга может иметь много подписанных пользователей);

1. При запуске приложения проводить авторизацию с вводом username и password, приложение ищет в БД, если есть соответствие, то проводим авторизацию пользователя.
2. Прием запросов можно организовать через консоль, запросы на которые приложение должно отвечать:

|  |  |  |
| --- | --- | --- |
| **запрос** | **описание** | **доступ** |
| fetch\_all\_users | отобразить в консоли список всех пользователей со всеми зависимостями | «ADMIN», «SUPER\_ADMIN» |
| fetch\_all\_incidents fetch\_all\_active\_incidents | отобразить в консоли список всех инцидентов и соответствующих пользователей без зависимостей | «ADMIN», «SUPER\_ADMIN» |
| fetch\_user\_by\_{id} | отобразить в консоли пользователя по id со  всеми зависимостями, {id} — параметр «id» | «ADMIN», «SUPER\_ADMIN» |
| add\_user | создать нового пользователя с профилем — в консоль дать статус проведенной операции, выводить вспомогательные сообщения для ввода всех полей пользователя и профиля | «ADMIN», «SUPER\_ADMIN» |
| update\_user\_{id} | обновить некоторые данные текущего пользователя по id с профилем — в консоль дать статус проведенной операции, выводить вспомогательные сообщения для ввода всех полей пользователя и профиля, {id} — параметр «id» | «ADMIN», «SUPER\_ADMIN» |
| delete\_user\_{id} | удалить пользователя по id со всеми зависимостями — в консоль дать статус  проведенной операции, {id} — параметр «id» | «ADMIN», «SUPER\_ADMIN» |
| subscribe\_service\_{id} | подписаться на услугу, {id} - параметр «id» | «ADMIN», «SUPER\_ADMIN», «USER» |
| unsubscribe\_service\_{id} | отменить подписку на услугу, {id} — параметр «id» | «ADMIN», «SUPER\_ADMIN», «USER» |
| create\_incident | создать инцидент (тикет) | «ADMIN», «SUPER\_ADMIN», «USER» |
| close\_incident | отметить статус инцидента как завершенный | «ADMIN», «SUPER\_ADMIN» |