

User Registration Service

In the next diagram (*Figure 1*) there are some of the core components of the service. The *UserController* receives a *SaveUserRequest* with the user details and phones, then it uses a validator method implemented in *UserControllerHelper*. The helper, in turn, uses the *UserRepository* to validate if the email is not already taken. If all validations succeed, the password is encoded using the *SecurityService* and the user is persisted to database.

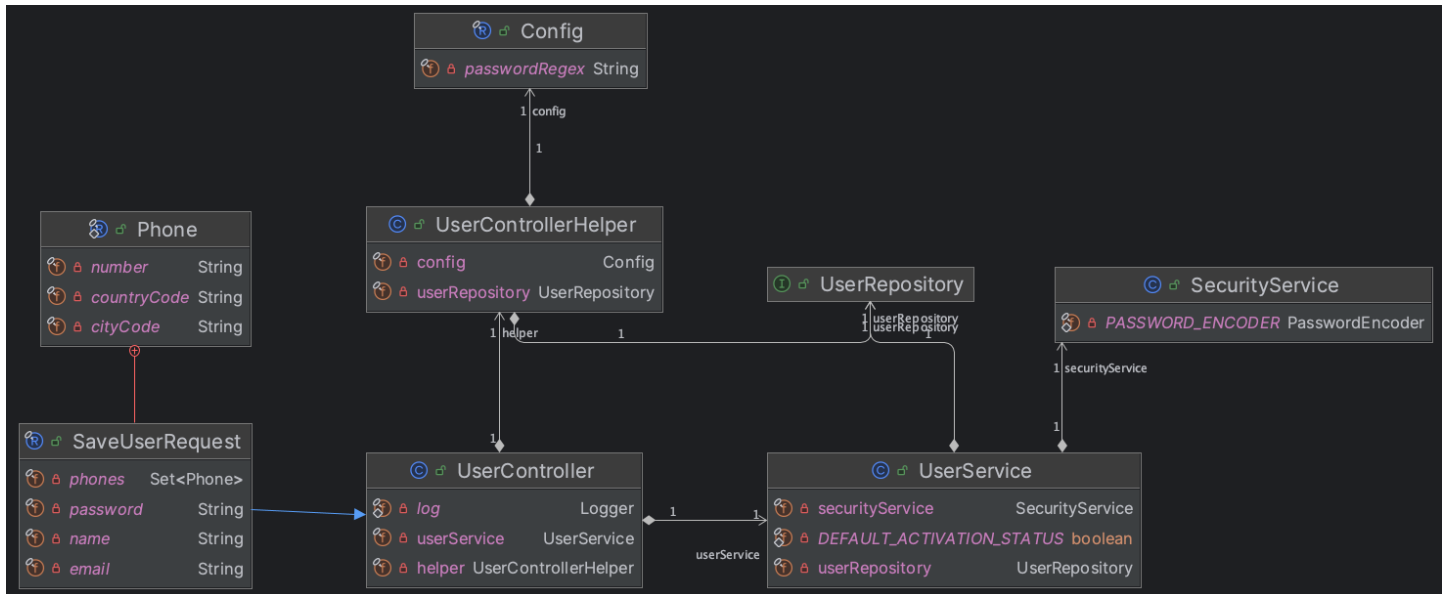


Figure 1- Core components

Once the user is persisted, the *UserMapper* converts the *User* entity to *UserRegistrationInfo*, the DTO that will be returned as response. See *Figure 2*.

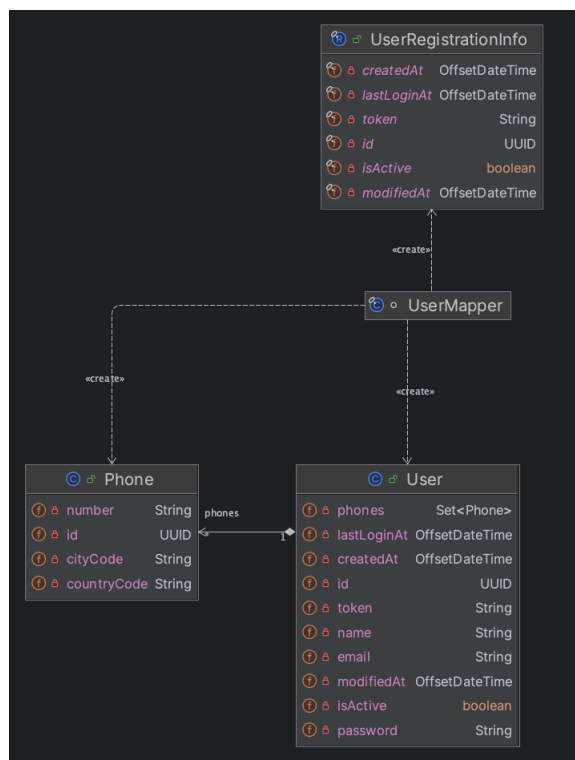


Figure 2 - DTO mapping

The data model

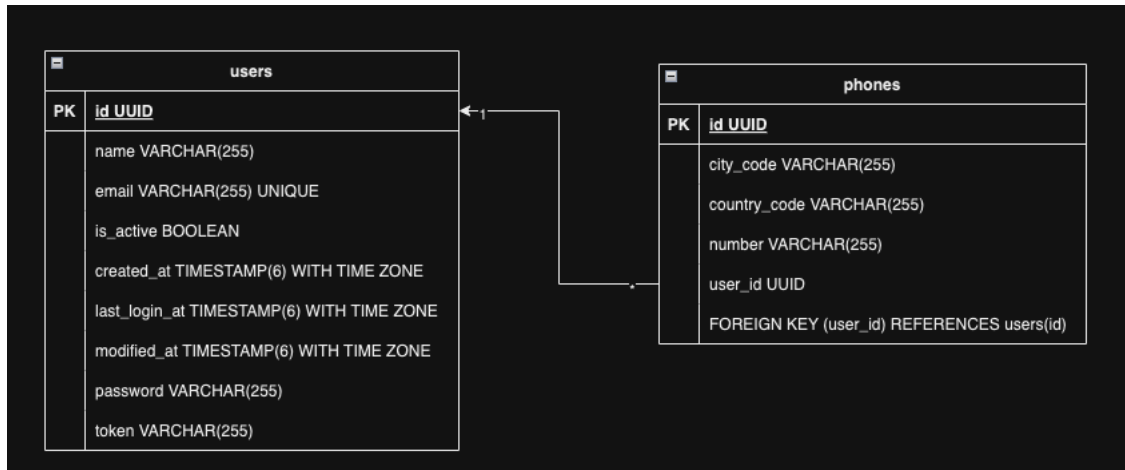


Figure 3 - ER diagram

The script

```
DROP TABLE IF EXISTS phones CASCADE;
DROP TABLE IF EXISTS users CASCADE;

CREATE TABLE users (
  id UUID PRIMARY KEY,
  name VARCHAR(255),
  email VARCHAR(255) UNIQUE,
  is_active BOOLEAN,
  created_at TIMESTAMP(6) WITH TIME ZONE,
  last_login_at TIMESTAMP(6) WITH TIME ZONE,
  modified_at TIMESTAMP(6) WITH TIME ZONE,
  password VARCHAR(255),
  token VARCHAR(255)
);

CREATE TABLE phones (
  id UUID PRIMARY KEY,
  city_code VARCHAR(255),
  country_code VARCHAR(255),
  number VARCHAR(255),
  user_id UUID,
  FOREIGN KEY (user_id) REFERENCES users(id)
);
```