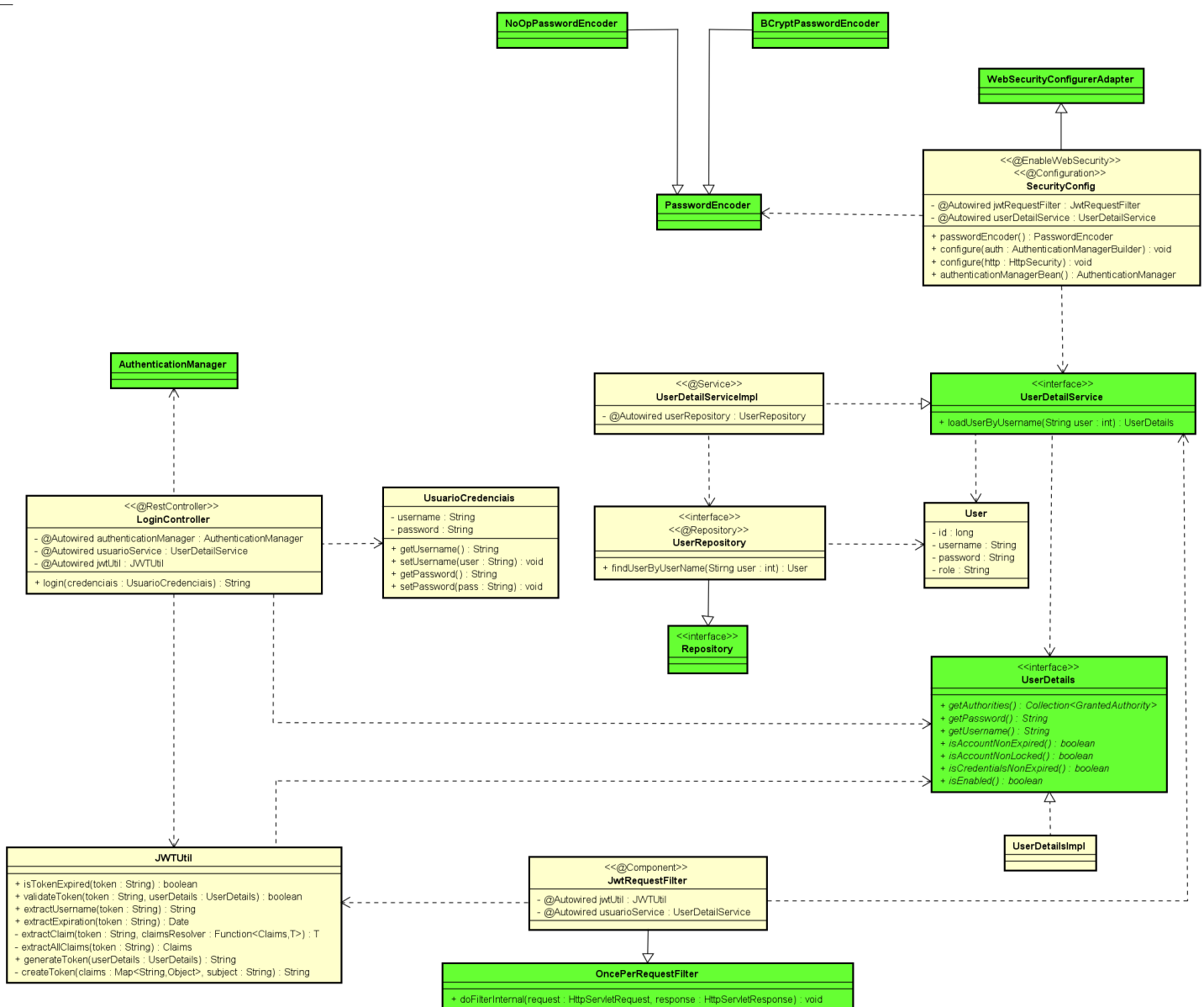


## Exercício – Migração do projeto PETS para fazer a autenticação e autorização usando Spring Security

Atividade, migrar o projeto PETS feito na atividade anterior para utilizar o Spring Security com JWT

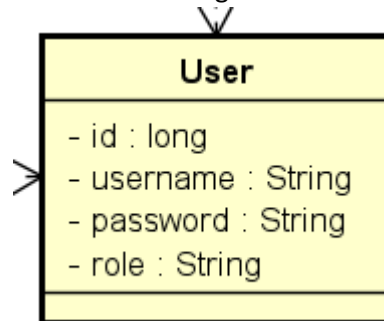
A estrutura de classes necessárias para a implementação do Security com JWT é mostrada no quadro 1, as classes e interfaces em verde já são fornecidas pelo Spring Framework ou Spring Security.



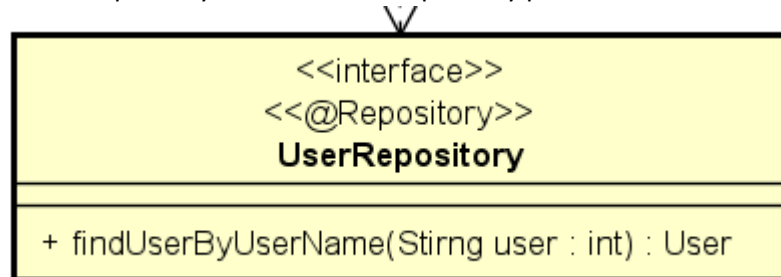
Quadro 1 - Estrutura de classes e interfaces envolvidas na implementação do Security com JWT

Para implementar deverão ser executadas as seguintes etapas:

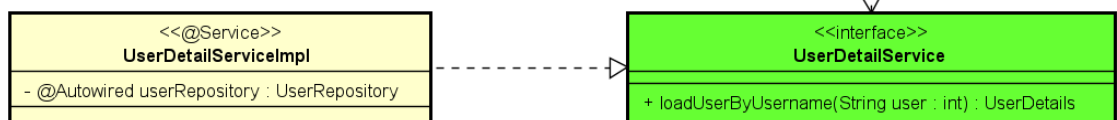
- Adicionar no arquivo build.gradle na seção de dependências a seguinte linha:  
**implementation group:'io.jsonwebtoken', name:'jjwt', version:'0.9.1'**
- Criar as classes utilizadas para pesquisar pelo usuário no banco de dados
  - Crie uma classe chamada User com a estrutura a seguir:



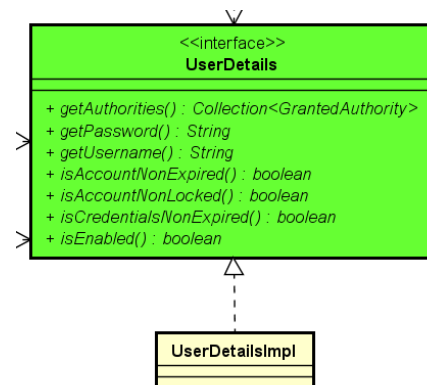
- Crie um Repository chamado UserRepository para acessar o banco de dados



- Crie uma classe de serviço chamada UserDetailsServiceImpl que irá acionar o repositório para buscar os usuários do banco de dados. Este serviço em especial deve implementar a interface UserDetailsService fornecida pelo SpringSecurity, sobrescrevendo o método loadUserByUsername(), a qual irá usar o repositório para buscar pelo usuário no banco de dados:



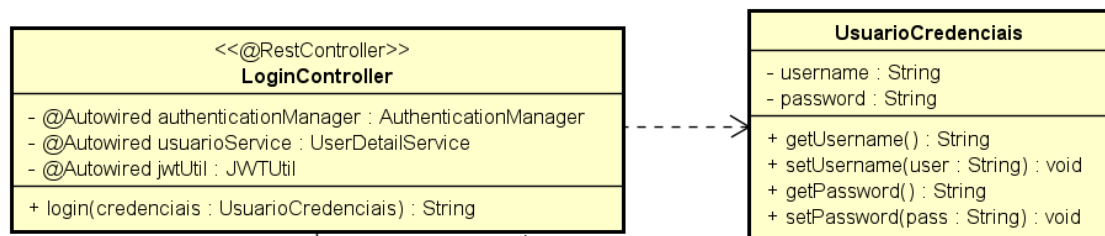
- O método loadUserByUsername() da classe de serviço criará uma instância de UserDetails e irá popular suas propriedades conforme os dados encontrados no banco de dados para este usuário. Para isto crie a classe UserDetailsImpl implementando a interface UserDetails para que os objetos dela possam conter as informações dos usuários:



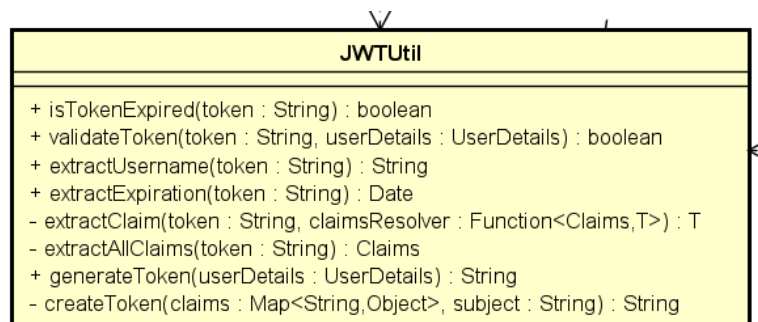
- Criar as classes a serem usadas na autenticação do usuário
  - Crie uma classe RestController chamada LoginController para fazer o Login, que receberá um objeto JSON contendo
 

```
{
            "username": "nome do usuário",
            "password": "senha do usuário"
          }
```

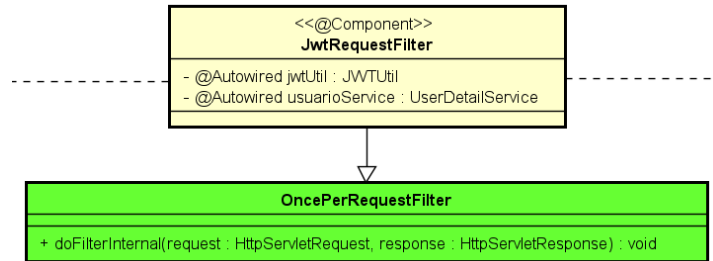
    - Este objeto JSON será representado no Backend Spring por uma classe estilo JavaBean chamada UsuarioCredenciais que contenha duas propriedades uma chamada username e outra chamada password, ambas do tipo String, contendo seus respectivos GETs e SETs. Para receber este objeto JSON, o controller deverá receber um parâmetro anotado com @RequestBody e um objeto desta classe UsuarioCredenciais
    - O controller deverá criar um objeto do tipo UsernamePasswordAuthenticationToken contendo o usuário e a senha recebidas pelo objeto do tipo UsuarioCredenciais
    - O objeto recém-criado deverá ser encaminhado para o AuthenticationManager na função authenticate. O objeto do tipo AuthenticationManager deverá ser injetado na classe controller.



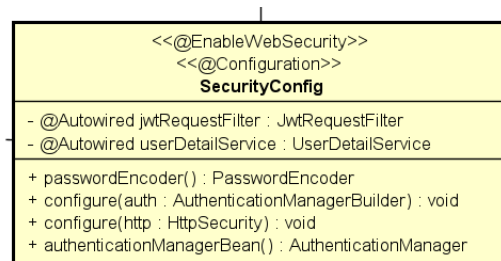
- Agora resta criar as classes para manipular o Token
  - Crie uma classe com as funções para consultar e criar Tokens, esta classe deve ser chamada JWTUtil e deve ser anotada com a annotation @Service para que um objeto dela possa ser injetado nas classes do SpringSecurity que irão utilizá-la.



- Crie a classe que executará o filtro de autorização, esta classe deverá se chamar JwtRequestFilter e herdar da classe OncePerRequestFilter:



- Crie uma classe chamada SecurityConfig que herde de WebSecurityConfigurerAdapter anotada com as annotations @Configuration e @EnableWebSecurity
  - Crie o método passwordEncoder() anotado com @Bean, que deverá retornar um objeto do tipo NoOpPasswordEncoder
  - Sobrescrever o método configure(AuthenticationManagerBuilder auth) para que passe ao objeto auth a instância da classe UserDetailsService. A instância desta classe UserDetailsService deverá ser injetada por meio do @Autowire.
  - Sobrescrever o método authenticationManagerBean deixando-o anotado com @Bean para que ele retorne a instância super.authenticationManagerBean();
  - Sobrescrever o método configure(HttpSecurity http) para configurar quais são as URLs autorizadas, informando também qual será a classe de filtro do JWT



Ao postar a atividade, coloque apenas as classes criadas, apenas os arquivos .java envolvidos nesta atividade.