eClinic

Technical solution description

Bogdan Sukonnov

Saint-Petersburg 2019

Content.

Project goals	3
Application description	3
Used Technologies	3
Database model	5
System infrastructure	5
System architecture	6
Additional features	12
Code quality	
Tests	13
Sonar report:	14
Build and deploy	14
Future improvement	15

PROJECT GOALS

Application development for the medical rehabilitation clinic. The application should have cohesive data model, user friendly interface, separate access to different system's part, reliable system.

APPLICATION DESCRIPTION

Web application has three types of user: doctors and nurses. Admin has access to each part of the application and can do everything that the other users do.

Doctors can add and discharge patients, create, edit and cancel prescriptions. Also they have access to any information about patients, prescriptions and events.

Nurses has limited access to the service. After authentication a nurse can only see events, complete and cancel them.

There is an authentication mechanism in the system that control access to the service. Each user in application has access level that defines information for displaying and privileges.

Data of users stores in a reliable database.

USED TECHNOLOGIES

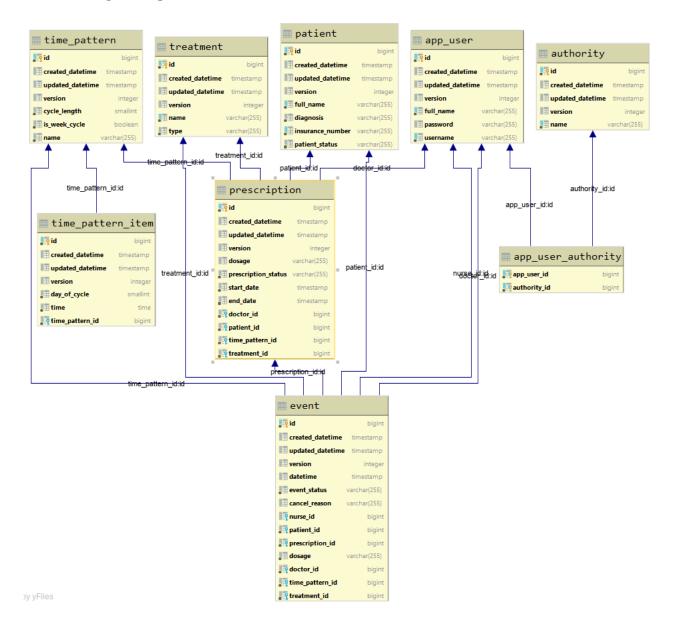
- Instruments:
 - 1. IDE IntelliJ IDEA
 - 2. Maven
 - 3. Docker

Technologies:

- 1. Java 8
- 2. Spring
- 3. JSP
- 4. JPA
- 5. Hibernate
- 6. JavaScript

- 7. jQuery
- 8. JSF
- 9. REST
- 10. Spring security
- 11. ActiveMQ
- 12. Bootstrap4
- 13. PostgreSQL
- 14. EJB
- 15. Junit
- 16. Mockito
- 17. Jenkins
- 18. SonarQube
- 19. Tomcat
- 20. WildFly
- 21. Log4j2
- 22. Flyway

DATABASE MODEL



SYSTEM INFRASTRUCTURE

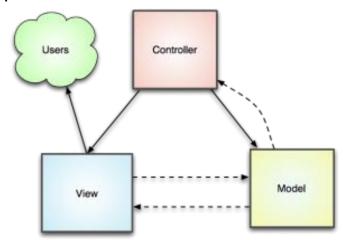
- Front-end (for doctors and nurses):
 - 1) Web-page structure HTML, JSP
 - 2) Page-design Bootstrap4
 - 3) Dynamic contents: JavaScript, Jquery

·· T ·· Systems·

- Back-end (server based level):
 - 1) Application server Tomcat
 - 2) Database PostgreSQL
 - 3) Server logic Spring Framework
- Client board application:
 - 1) Web-pages JSF
 - 2) JMS ActiveMQ
 - 3) Application server WildFly
 - 4) Server logic EJB
 - 5) WS REST

SYSTEM ARCHITECTURE

Architecture of server-based part presented by MVC - design pattern.



Class structure

According MVC-pattern application has next structure:

- com.bogdansukonnov.eclinic

 config

 controller

 converter

 dao

 dto

 entity

 exceptions

 message
- Model level:
- entity

securityservice

- (C) AbstractEntity
- AppUser
- Authority
- © Event
- EventStatus
- Patient
- PatientStatus
- Prescription
- PrescriptionStatus
- SelectorData
- TimePattern
- TimePatternItem
- Treatment
- TreatmentType

Model-service level:

- dao
 - AbstractDao
 - AbstractTableDao
 - Dao Dao
 - EventDao
 - EventDaolmpl
 - PatientDao
 - PatientDaolmpl
 - PrescriptionDao
 - PrescriptionDaolmpl
 - TableDao
 - TimePatternDao
 - TimePatternDaoImpl
 - TreatmentDao
 - TreatmentDaolmpl
 - UserDao
 - UserDaolmpl

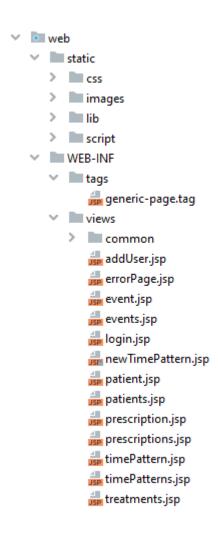
Service level:

- ✓ Image: Service
 - AppUserDetailsService
 - AppUserDetailsServiceImpI
 - EventService
 - © EventServiceImpl
 - MessagingService
 - MessagingServiceImpl
 - OrderType
 - PatientService
 - PatientServiceImpl
 - PrescriptionService
 - PrescriptionServiceImpl
 - TimePatternService
 - TimePatternServiceImpl
 - TreatmentService
 - TreatmentServiceImpl
 - UserService
 - UserServiceImpl

View-service level:

- controller
 - ErrorController
 - EventController
 - C LoginController
 - PatientController
 - PrescriptionController
 - TimePatternController
 - TreatmentController
 - UserController

View level:



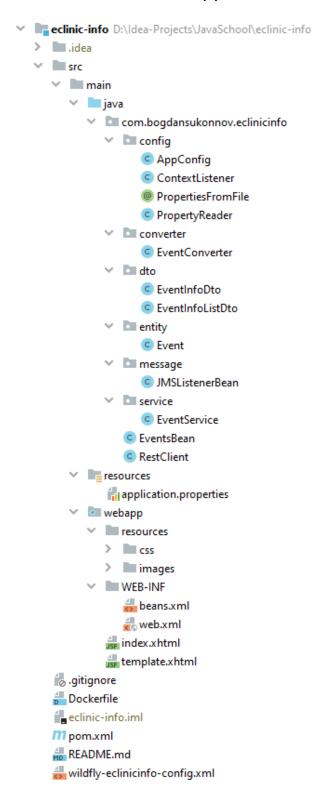
Configuration:

- config
 - AppConfig
 - Applnitializer
 - CustomRequestLoggingFilter
 - EClinicConstants
 - HibernateConfig
 - MessagingConfig
 - SecurityConfig
 - SecurityInitializer

DTO:

- ✓ dto
 - (a) AbstractDto
 - © EventDto
 - © EventInfoDto
 - © EventInfoListDto
 - © EventToCalendarDto
 - C IdDto
 - RequestEventTableDto
 - RequestPatientDto
 - RequestPrescriptionDto
 - RequestTableDto
 - © ResponsePatientDto
 - ResponsePrescriptionDto
 - SelectorDataDto
 - © SelectorDataElementDto
 - TableDataDto
 - TimePatternDto
 - □ TimePatternItemDto
 - TreatmentDto
 - Update

Rest client applications:

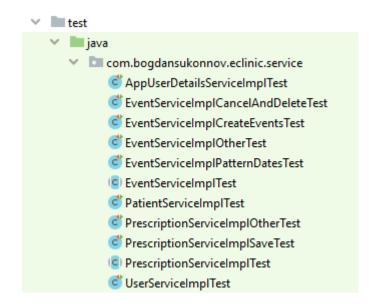


ADDITIONAL FEATURES

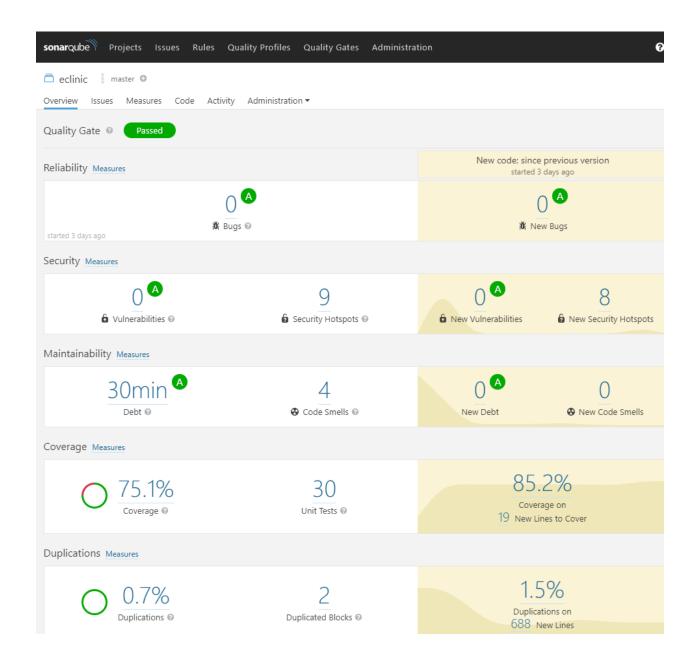
- 1.Docker. All parts of the application are stored in docker containers and starts up by docker-compose command
- 2. Jenkins + SonarCube. Jenkins starts build automatically after the push to bitbucket repository and check code with SonarScanner plugin.

CODE QUALITY

Tests:



Sonar report:



BUILD AND DEPLOY

To build the application run the command in a terminal *mvn* clean install flyway:migrate

Docker-compose up command is a part of the install maven goal.

·· T ·· Systems·

FUTURE IMPROVEMENT

- 1. Refactoring and optimization code.
- 2. Adding new functionality