Software Test Report

DST2 group 2

Introduction

Project title: precision medicine matching system.

Next generation sequencing (NGS) promises tailored treatment by offering high throughput genetic information as a medication reference. Aiming at consolidating the bridge between NGS and clinic medication, we have developed precision medicine matching system as a solution for doctors and researchers to understand patients' NGS data by mapping clinically significant variant revealed by annotated NGS data to PharmGKB pharmacogenetics data (Barbarino *et al.*, 2018). To construct the system using PharmGKB data and display it as a user-friendly web application software, we have adopted Spring-MVC architecture which follows J2EE standards. The following software testing report describes the performance of the software during the course of development, which provides traceable information for development team, technical users and business users.

Test environment

Hardware: MacBook Pro (2019) 8C/16G/512GB.

Software: Web container: Tomcat 9.0.315; Database management system: PostgreSQL 10, pgAdmin 4; Development environment: IDEA Ultimate 2019.3.4; JDK 12.

Test design

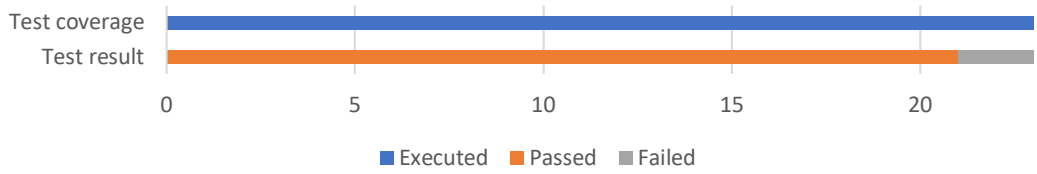
23 test cases have been designed for white box testing while 10 for black box testing. Test cases were designed based on 2 use cases defined in the requirement analysis, including doctor & scientist use case and administrator use case. For doctor & scientist use case, testing have covered file uploading, matching, knowledge base data accessing, registration and login. For administrator use case, testing have covered knowledge base data update, sample & use account management. Boundary cases, for example, blank file, huge file and wrong-format file, are covered. Test cases (Appendix 1) are implemented by Junit test integrated with Spring-MVC framework via SpringJUnit4ClassRunner.

Test objective

Date: 10-05-2020

White-box full functional testing is completed on 23 test cases (Appendix 1), the integration of backend and frontend functionality and performance was evaluated from the perspective of developer.

Test summary



Passed tests: B1~8, B10, B11.

Failed tests: B9, B12.

Description:

B9: Delete sample metadata and data by sample ID at the same time.

B12: Update tables constructed by the downloaded tables in the resource directory (clinic_meta, variant, gene, var_drug_ann) and table generated (location_annvar).

Defects

1. Table generation through joining failed.

Description: Failed to generate location_annvar in PostgreSQL database after successful running PostgreSQL stored functions. No exception is thrown in the method dedicated to generating this table (issue source file) while assertion error emerges when testing (test source file). Inspection on VarDrugAnnDAO.doImportVariant() suggested lack of SQL commitment after muting the auto commitment functionality of JDBC. Bug fixed after adding commitment codes.

Issue source file	src/main/java/DST2/Group2/DAO/VarDrugAnnDAO.java; line 162.		
Test source file	src/test/java/basic/UpdateDataTest.java		
Priority	High	**Status**	Fixed

2. Inconsistent sample ID in sample metadata and sample data.

Description: Once a file is uploaded, a sample ID is allocated to the file by SampleDAO. The VepDAO or AnnovarDAO corresponding to the sample format take sample ID returned by SampleDAO to allocate sample ID in data table. SampleDAO decrease the sample ID by 1 before returning back, which leads to inconsistency. Bug fixed after returning correct sample ID.

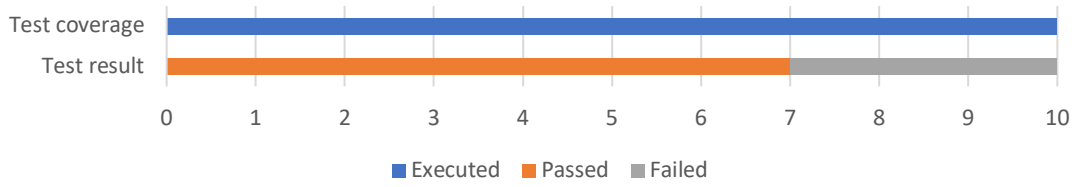
Issue source file	src/main/java/DST2/Group2/DAO/SampleDAO.java; line 65.		
Test source file	src/test/java/web/UploadControllerRedirectTest.java.		
Priority	High	**Status**	Fixed

Test objective

Date: 13-05-2020

Black-box testing is completed from the perspective of the user, overall functionality and performance was evaluated by testing 10 cases (Appendix 1) with both common and boundary cases directly through the user interface.

Test summary



Passed tests: U2~7, U9

Failed tests: U1, U8, U10

Defects

1. No form for typing in email on registration page.

Description: User email is required for registration. However, there is no place for typing in this information on the web page.

Issue source file	src/main/webapp/WEB-INF/view/register.jsp		
Priority	High	**Status**	Under fixing

2. Annotation with empty text or phenotype appears after filtering matching result.

Description: In the matching result page, after user specified a phenotype to filter the result, clinic annotation and variant-drug annotation records with no phenotype information or annotation text may appear. Bug fixed after adding not null checking in ClinicAnnDAO and VarAnnDAO search method.

Issue source file	src/main/java/DST2/Group2/DAO/VarDrugAnnDAO.java; line 36. src/main/java/DST2/Group2/DAO/ClinicAnnDAO.java; line 31.		
Priority	Medium	**Status**	Fixed

3. Less informative web page information.

Description: Dashboard page does not contain any instruction about how to use the system. Other pages are also less informative.

Issue source file	src/main/webapp/WEB-INF/view		
Priority	Medium	**Status**	Under fixing

Reference

BARBARINO, J. M., WHIRL-CARRILLO, M., ALTMAN, R. B. & KLEIN, T. E. (2018) PharmGKB: A worldwide resource for pharmacogenomic information. *Wiley Interdisciplinary Reviews: Systems Biology and Medicine*.

Appendix 1. Test Case

Id	Description	Test source file
B1	Retrieve all clinic annotation, dosing guideline, drug label, drug and variant drug annotation information (as corresponding bean class) stored in knowledge base.	src/test/java/basic /BulkRetrieveDataTest.java
B2	Retrieve all sample metadata information.	src/test/java/basic /BulkRetrieveDataTest.java
B3	Search user without username.	src/test/java/basic /RegisterTest.java
B4	Search user with only username.	src/test/java/basic /RegisterTest.java
B5	Insert new user.	src/test/java/basic /RegisterTest.java
B6	Delete user by username.	src/test/java/basic /RegisterTest.java
B7	Save VEP or annovar-annotated sample VCF data.	src/test/java/basic /SampleTransactTest.java
B8	Insert sample metadata.	src/test/java/basic /SampleTransactTest.java
B9	Delete sample metadata and data by sample ID at the same time.	src/test/java/basic /SampleTransactTest.java
B10	Given single drug name and/or phenotype as search criteria, filter beans accordingly.	src/test/java/basic /SearchTest.java
B11	Update tables constructed by the crawler (drug_label, dosing_guideline, drug) and ones generated from them (drugnames, dosing_guideline_name).	src/test/java/basic /UpdateDataTest.java
B12	Update tables constructed by the downloaded tables in the resource directory (clinic_meta, variant, gene, var_drug_ann) and table generated (location_annvar).	src/test/java/basic /UpdateDataTest.java
W1	Retrieve knowledge information upon request, redirect to new page.	src/test/java/web /KnowledgeBaseControllerTest.java
W2	Registered user signs in with correct password.	src/test/java/web /UserControllerTest.java
W3	Registered user signs in with incorrect password.	src/test/java/web /UserControllerTest.java
W4	Unregistered user signs in.	src/test/java/web /UserControllerTest.java

W5	User registration	src/test/java/web /UserControllerTest.java
W6	Upload sample annovar-annotated file (883KB) and redirect to the matching controller.	src/test/resources /annovar_test.vcf
W7	Upload sample vep-annotated file (262KB) and redirect to the matching controller.	src/test/resources /vep _test.vcf
W8	Upload blank sample	src/test/resources /blank_test.vcf
W9	Upload truncated/wrong type sample	src/test/resources /truncAnnovar_test.vcf src/test/resources /wrong_test.png
W9	Match mock annovar-VCF sample (3 records)	src/test/java/web /MatchControllerTest.java
W10	Match mock annovar-VCF sample (6 records)	src/test/java/web /MatchControllerTest.java
W11	Upload knowledge base tables upon request	src/test/java/web /AdminControllerTest.java
U1	Sign in with unregistered username	(manual visual inspection)
U2	Registration and subsequent sign in	(manual visual inspection)
U3	Annovar-annotated VCF file upload and matching	(manual visual inspection)
U4	Vep-annotated VCF file upload and matching	(manual visual inspection)
U5	Wrong/truncated file upload	(manual visual inspection)
U6	Knowledge base data display	(manual visual inspection)
U7	Introduction, about us display	(manual visual inspection)
U8	Search matching result according to drug name and/or phenotype name	(manual visual inspection)
U9	Software management with administrator account	(manual visual inspection)
U10	Overall informative and accessible information	(manual visual inspection)

B: white box base application unit testing.

W: white box web application unit testing.

U: black box web application testing directly through deployed web page.