

PROGRAMMING FOR SOFTWARE ENGINEERS

Table of Contents

Introduction.....	3
Analysis.....	3
Class Diagram	3
JUnit Testing	4
Project Repository and Commit Messages.....	6
Conclusion	7
References List.....	9

Introduction

With the advancement of technology, there have been several technologies and software that help to manage different tasks within an organization. Based on the object-oriented programming languages, the implementation of such tasks becomes easier. The four pillars of object-oriented programming provide ease to the system. In the given project, a case of Physiotherapy & Sports Injury Centre or PSIC has been provided, which tries to build a proper booking system for its patients. The centre deals with a number of patients and physiotherapists. Therefore, managing a proper system for booking purposes is essential. In the project, a system will be developed using the object-oriented programming language, i.e. Java, which will be able to successfully book the appointments of the visitors and patients.

Analysis

Class Diagram

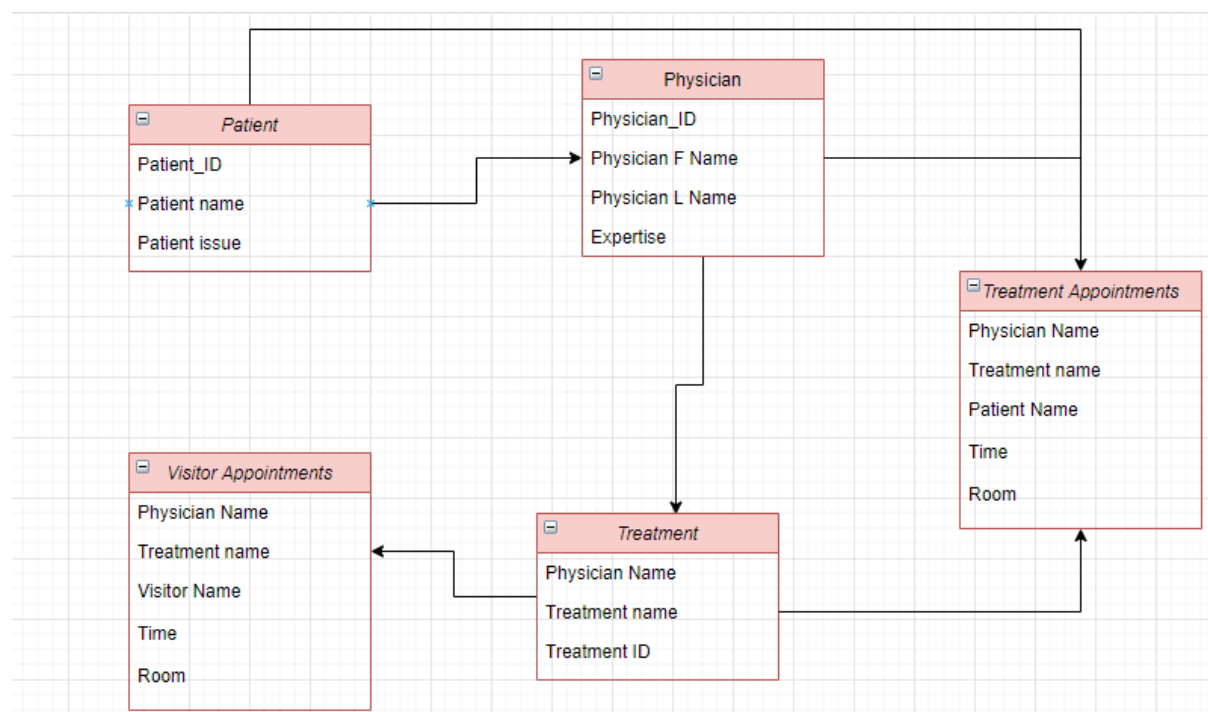


Figure 1: Class Diagram for the PSIC system

(Source: Self-created in Draw.io)

The above figure shows the UML Class diagram for the given case study of Physiotherapy & Sports Injury Centre or PSIC. The class diagram contains all the main classes of the system. As shown in the image, there are five main classes for the development of the system as

"Patient", "Physician", "Treatment Appointments", "Treatment", and "Visitors Appointments". These five classes contain all the attributes. Based on the attributes, the system designing can be made easier (Gulati *et al.* 2020). The key attributes of the classes have been mentioned in the UML Class diagrams. Therefore, the figure shows the key attributes of the "Patient" class as Patient ID, Patient Name and Patient Issue or what treatment the patient is looking for. Also, the "Treatment Appointments" class has the attributes like Physician name, Treatment name, Patient name, Time, and Room while the "Visitors Appointments" class has attributes such as Physician name, Treatment name, Visitor name, Time, and Room etc. Through class diagrams, an overview of the system can be generated. Based on this, the actual system designing is done.

JUnit Testing

The task aims to develop a system for the Physiotherapy & Sports Injury Centre so that the booking can be made easily by the patients and visitors (Haulund *et al.* 2017). The entire task has been accomplished with the help of an object-oriented programming language. The task considers a public class "PhyioSports". Under the public class, different buttons have been created, through which data regarding the patients, physicians, treatment can be entered. Another button, "Book Treatment," has also been created, which takes patient details for the booking purpose. For this purpose, JUnit testing has been implemented. JUnit testing is nothing but a regression testing framework that improves the code execution process and also enhances the quality of code. JUnit testing can be implemented in several build tools such as Maven, Eclipse, and Ant etc (Panichella *et al.* 2017). However, in this program, the JUnit testing has been implemented with the Ant build tool in Java. The purpose of the JUnit testing is to build a source code that would execute properly in a certain way. Through the testing process, it was understood if the software is working as expected or not.

The above figure shows the output window after the execution of all the tasks. The code prints "1" for each successful execution. Or, in simple words, if the PhyioSport.java file runs successfully, it will print 1 (Ivanov *et al.* 2017). After the execution, if the buttons are also executed successfully, the code will again print "1" as a symbol of successful execution. Thus the testing has been implemented with the code.

Project Repository and Commit Messages

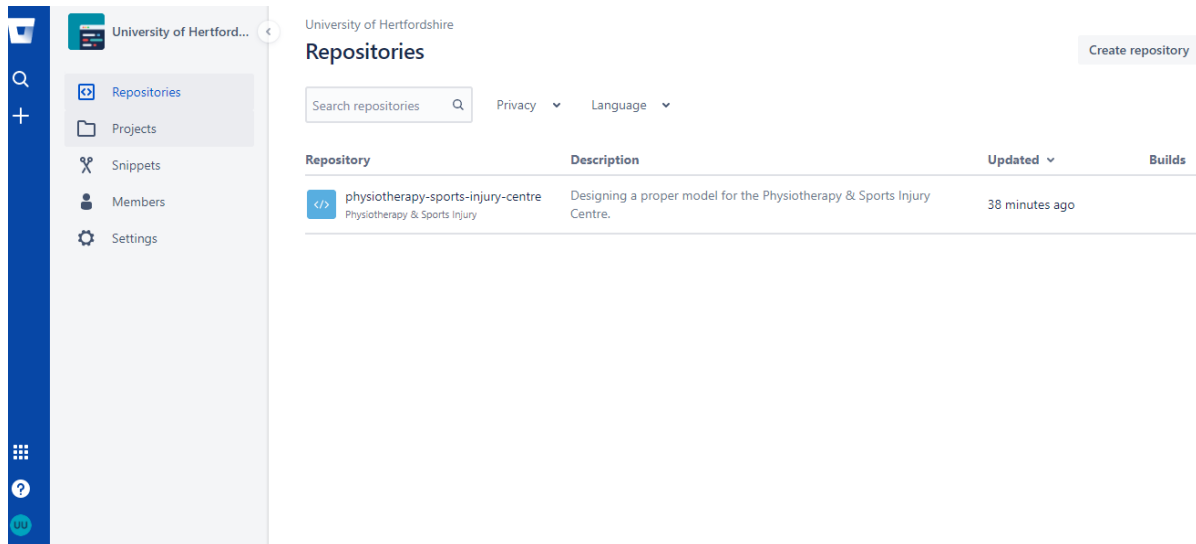


Figure 4: Creating a repository

(Source: Self-created in Bit Bucket)

The above image shows the repository that has been created in the version control software named Bit Bucket for the Physiotherapy & Sports Injury Centre. The project repository contains the actual source code of the system in the README.md file and a description of the project.

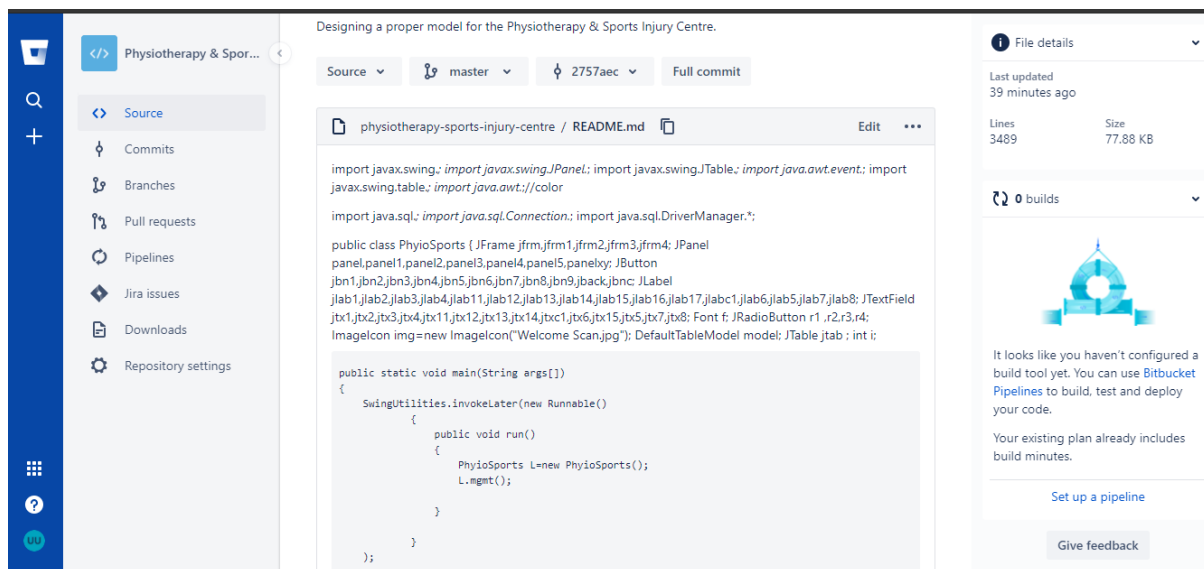


Figure 5: Implementing the code in the repository
(Source: Self-created in Bit Bucket)

The above figure depicts the source code for that PSIC system design in the repository. The repository was created in the version control software named Bit Bucket (Molina *et al.* 2018). The image implies the master program of the PSIC system and the README.md file. From the below image of the Bit Bucket repository, the commit messages can be seen. Whenever there is any change or modification in the files or code, the commit messages will appear.

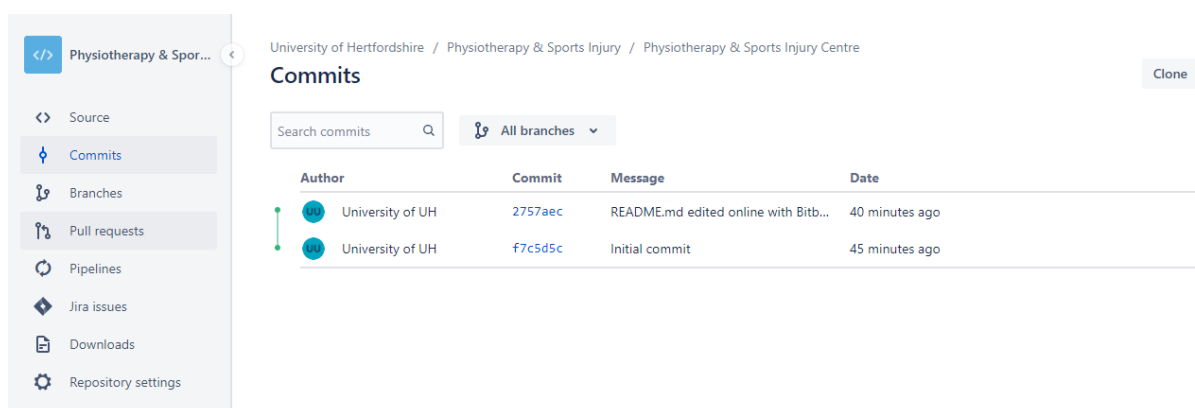


Figure 6: Commit Messages
(Source: Bit Bucket)

Conclusion

The project implements the concepts of object-oriented programming and JUnit testing in the same program to develop a system for the Physiotherapy & Sports Injury Centre or PSIC.

The tasks have been implemented using java language, as well as unit testing has been done. Through the unit testing framework, the code has been provided with more accuracy, and the quality of the code has also been ensured. Therefore, it can be concluded that the project successfully designs the booking system for the physiotherapy and sports injury centre so that the customers can easily book their appointments.

References List

Journals

Gulati, S. and Sharma, R.,2020. Java Unit Testing with JUnit 5.

Haulund, T., 2017. Design and implementation of a reversible object-oriented programming language. *arXiv preprint arXiv:1707.07845*.

Hourani, H., Wasmi, H. and Alrawashdeh, T., 2019, April. A Code Complexity Model of Object Oriented Programming (OOP). In *2019 IEEE Jordan International Joint Conference on Electrical Engineering and Information Technology (JEEIT)* (pp. 560-564). IEEE.

Ivanov, V., Rogers, A., Succi, G., Yi, J. and Zorin, V., 2017, August. What do software engineers care about? gaps between research and practice. In *Proceedings of the 2017 11th Joint Meeting on Foundations of Software Engineering* (pp. 890-895).

Molina, U.R., Kifetew, F. and Panichella, A., 2018, May. Java unit testing tool competition-sixth round. In *2018 IEEE/ACM 11th International Workshop on Search-Based Software Testing (SBST)* (pp. 22-29). IEEE.

Panichella, A. and Molina, U.R., 2017, May. Java unit testing tool competition-fifth round. In *2017 IEEE/ACM 10th International Workshop on Search-Based Software Testing (SBST)* (pp. 32-38). IEEE.