**School of Media Arts and Technology**

BSc (Hons) **Software Engineering**

**Academic Year 2019-2020**

**STEFAN SHKENDEROV**

Software Product

Report

**Object Oriented Resign And Development**

**Tutor: Craig Gallen DATE 17.01.2020**

Contents

[1.DECISIONS 3](#_Toc30094936)

[2.MANUAL TEST PLAN AND STRATEGY 4](#_Toc30094937)

[3.CONCLUSION 5](#_Toc30094938)

[APPENDIXES 6](#_Toc30094939)

[Appendix A: 6](#_Toc30094940)

# 

# 1.DECISIONS

The goal of this application is to have a working system capable of automating the appointment process of a service for care workers.

The model itself consists of 5 packages – DTO, DAO, service model and web model.The DTO package has a Person class, which can be assigned to a role – Patient or Dentist, with the role being an enum. Other than that the person has as attributes both names, address and an ID which will be set by the DAO upon saving in the database. The second class is Appointment, as attributes it has description, PersonA (which is the client) and PersonB(Worker), a date, an id and duration of the appointment.

In the second package of the model are the DAO Interfaces – for Appointment, Person and the DAOFactory. The DAO Factory is used to get the other DAOs and they themselves contain functions relating to saving, deleting and finding People and Appointments by various parameters. The DAO methods of Person and Appointment are tested via Junit since they are the ones that handle insertions, deletions and find entries in the Database.

The third package contains the ServiceFacade interface, used in the Service layer of the project. It is used as a connection between the JSP files in the Web layer and the DAOs. It has the same methods of the 2 DAOs and in their implementation it just calls the corresponding methods from the DAOs. The other interface is ServiceObjectFactory, which is used to return the SeriviceFacade via a getter for the service layer. Junit tests on personOnSite and personLeavingSite methods are written, because of their role of parsing user requests and using a number of DAO methods. The checks are needed in order to see that the data is parsed correctly from the service layer reformatted and correctly entered in the methods of the DAOs.

The last interface Is WebObjectFactory, which does the same, but for the Web layer.

All of the interfaces and classes in the Model contain Javadoc information.

The testHeartbeat.jsp file has two tables – for the people and the appointments entered. In order to add a person a button is pressed, then a form appears in the same jsp where the user inputs the information. The same case is for entering an appointment but there 2 tables appear, because upon creation the appointment requires the user to enter the patient, which is PersonA in the DTO. Sending a person on site via the id’s of the person and appointment sets the PersonB attribute of the Appointment to the person with the corresponding id entered. In the table for the People the user can delete a person and in the Appointments table the user can either delete an appointment, use “PersonLeavingSite” or click the “Person Ok” button if needed to be done manually.

# 2.MANUAL TEST PLAN AND STRATEGY

Although a number of Junit tests are written in this project, a manual comprehensive test of the functionality is to be performed as well. The test plan is to first open the web app, entering a person with the role of a worker. After this an appointment will be made trough the web app, entering a second person who will be a patient. Then the information of the person in the web app will be modified and again he will be signed to an appointment. This time the web user will press “Person on Site”, “Person OK” and “Person Leaving Site”. This will test every part of this project's functionality, representing a real scenario in which this project can be use. The sequence diagram located in ..\project\UMLmodel\drawio is based on this scenario.

# 3.CONCLUSION

The design of this project is based on a modular structure, including the use of JPA with the end application being presented via JSP. The work done includes fully functioning model, DAOs with JUnit test which connect to a Service Facade for the web user, which has Junit tests as well.