Modeling Languages Project Report

Juan Jose Soriano Escobar

January 2, 2018

Contents

List of Figures

A Introduction

Modeling is an important part of any System or product development that helps to understand the model system an holistic point of view and to ensure that all the functionalities and stakeholders requirements are cover. One of the most popular modeling tools nowadays in the software and product development environment is UML.

UML is a communication standard for the world of software development. It consists of different types of diagrams that will, together, help you describe the boundary of the system, the structure of the system, and the behavior of the system.

In this report, a Hospital appointment and billing system management are described and modeled in UML as result of the concepts and diagrams learn in the course of Modeling Languages.

The description of the model in this report will be split by diagram types, covering use case, sequence, activity, state and class diagram.

B Use Case Diagram

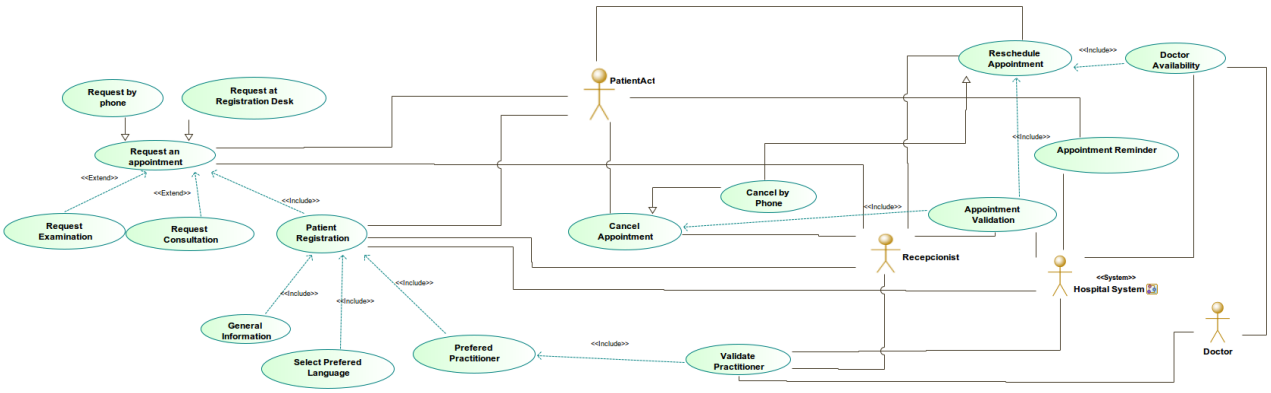
A use case captures a contract between the stakeholders of a system about its behavior describes the systems behavior under various conditions as it responds to a request from the primary actor (Delligatti, Steiner, Soley, 2013).

The case diagram is also good to ensure that every member in the development team understand the main activities and behavior of the system, recognizing the actors and the global interactions between them

The use case diagram is divided in three main behaviors, the appointment management, clinical history management and payments.

B.1 Appointment Use Case

In the figure 1 the use case diagram for the appointment system is Illustrated. It contains four Actors, three Human actors such as the Patient, the Receptionist and the Doctor. The remaining actor is the Hospital System which is consider as a Software from the hospital that is used to manage all the logistics, billing and service operations by different users.



Created with Modelion 3.7.

Figure 1: Appoint Use Case Diagram

This diagram presents how the patient communicates with the receptionist by phone or at the Hospital desk, to Schedule, cancel or modify an appointment. To do so, the Receptionist has to use the system, registering the patient and validating if the doctor is available for the required appointment.

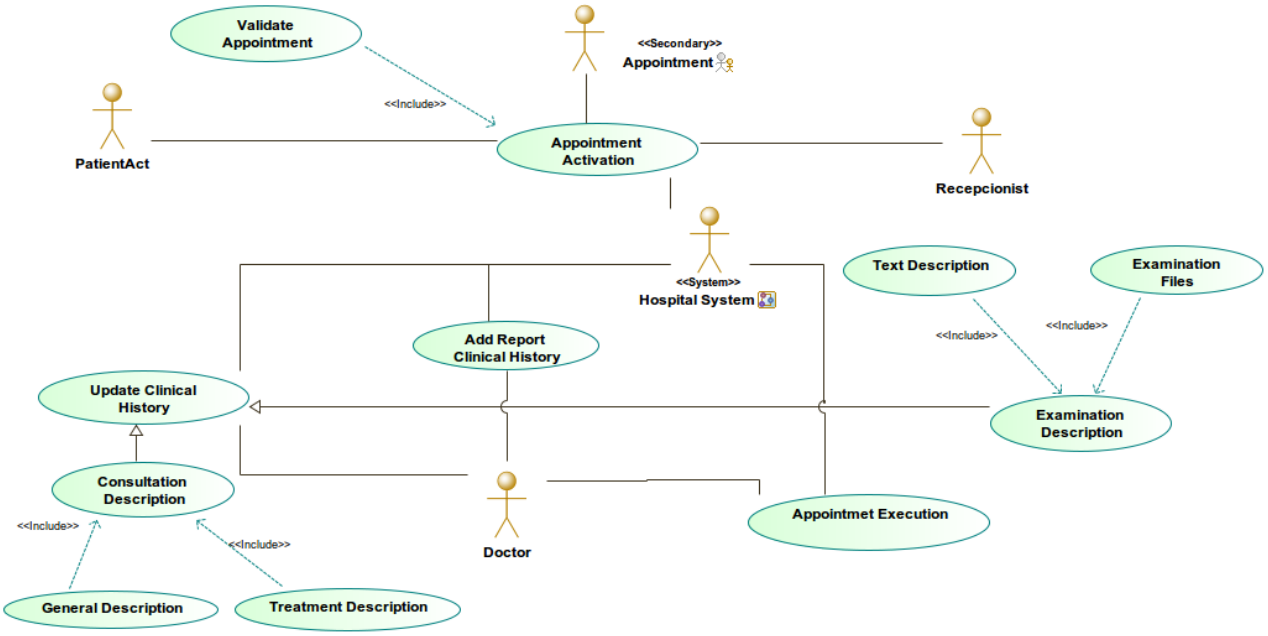
B.2 Examination Use Case

Once the patient has an appointment scheduled, the next step is to understand the appointment behavior. As actors, there are the same four included in the previous Use Case Diagram plus the appointment considered as a secondary one.

The day of the appointment, the Patient will go to the Hospital desk by the time of the appointment as is expected. The receptionist will activate the appointment in the system, to let know to the Doctor and to the billing staff that the patient has come to the appointment as planned.

Once in the execution of the appointment with the Doctor, depending on the type of the appointment (Examination or consultation), the Doctor can rad and Update the Patient Clinical History by using the Hospital System. During the reportm the Doctor is allowed to add files, treatment description and some text explaining the patient condition.

Other interaction between the Doctor and the system such as creating reports and email notifications will be explained later in the Sequence diagram.

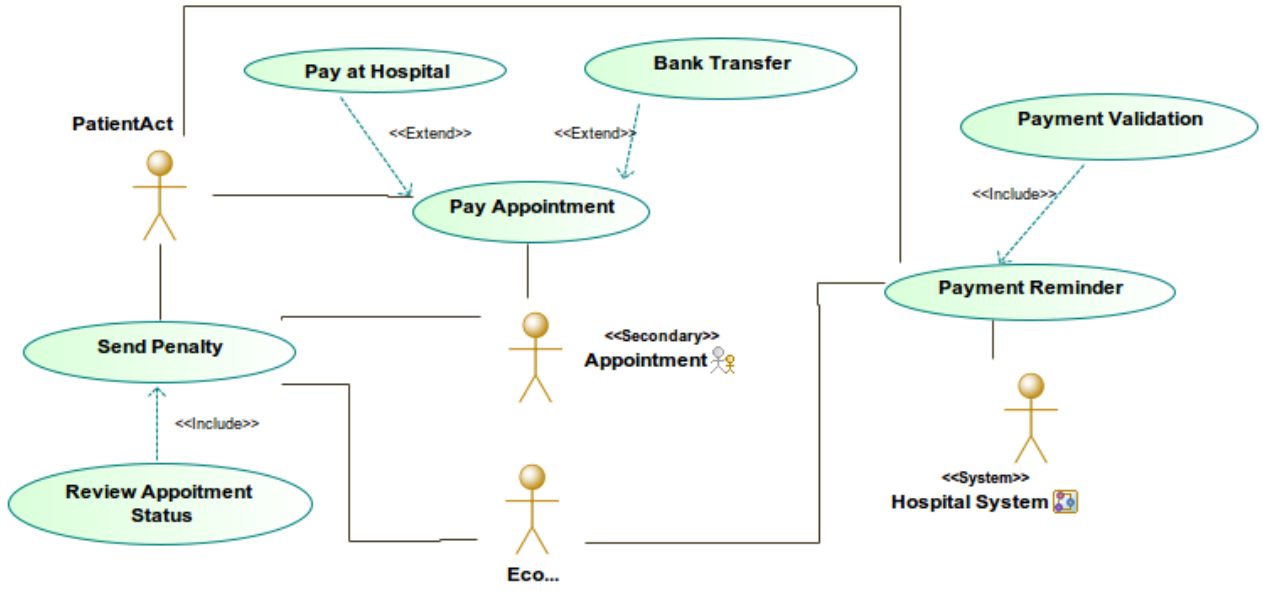


Created with Modelion 3.7.

Figure 2: Examination Use Case.

B.3 Payments Use Case

In the payments Use Case, the appointment becomes a secondary actor by itself because the appointment is the connecting factor (somehow the product consume by the patient) between the Patient and the Hospital. In the diagram included in the Figure 3, the Economy staff Actor from the hospital is introduced to review the payment status of the appointments and create penalty fees whenever is needed. The Economy staff uses the Hospital system to access to the appointments an patient payments.



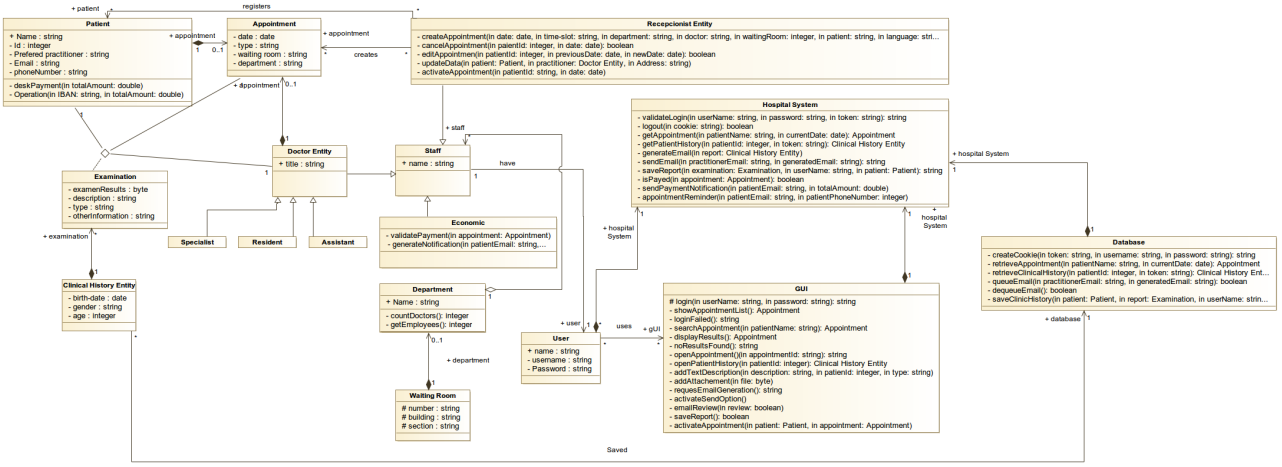
Created with Modelion 3.7.

Figure 3: Payment Use Case Diagram

The Patient have the option of paying at the hospital or by a bank transfer.

C Class Diagram

Class diagram describes the attributes and operations of a class and also the constraints imposed on the system. The class diagrams are widely used in the modeling of objectoriented systems because they are the only UML diagrams, which can be mapped directly with object-oriented languages.

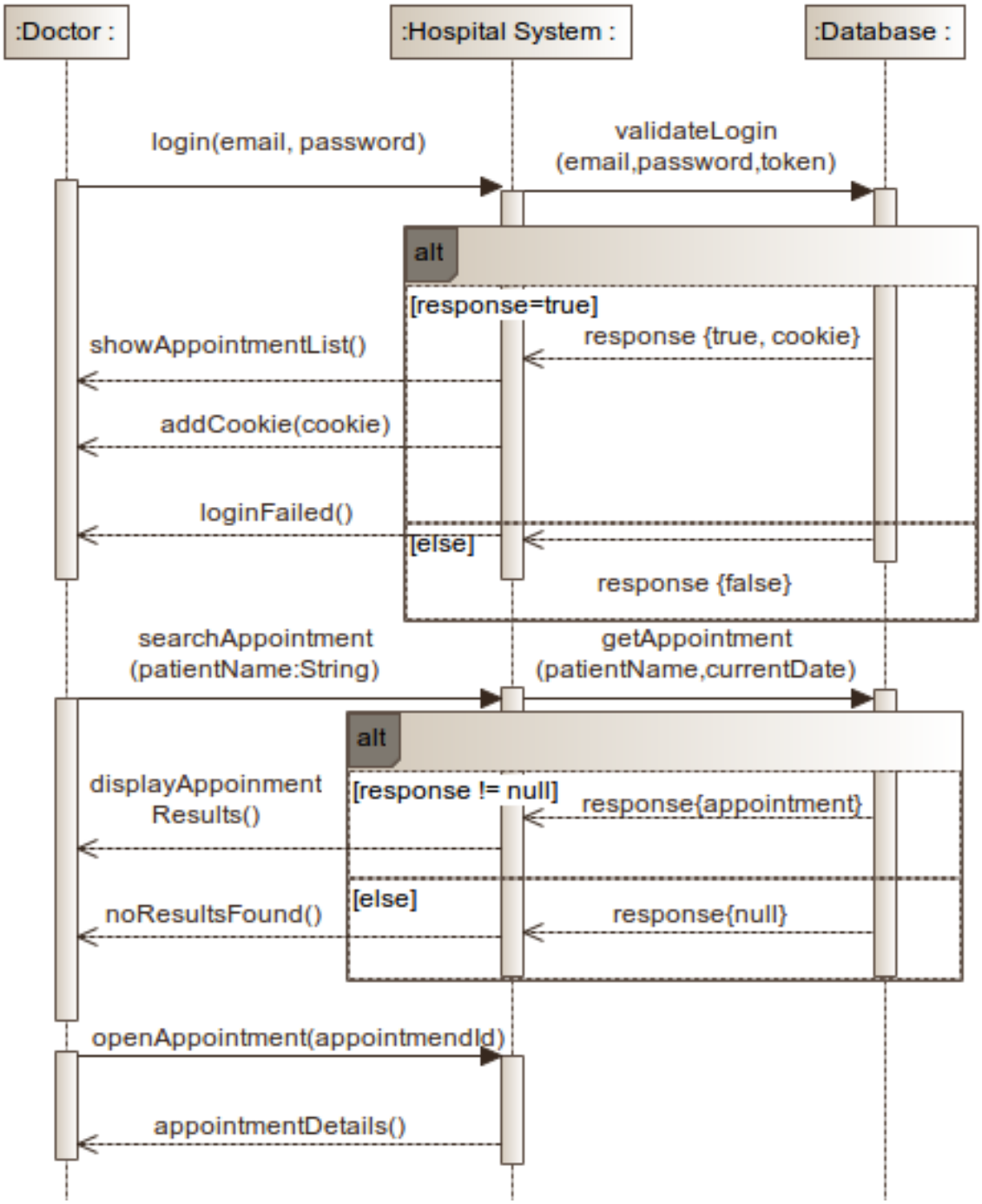


self-made

Figure 4: Distributed Computing Implementation.

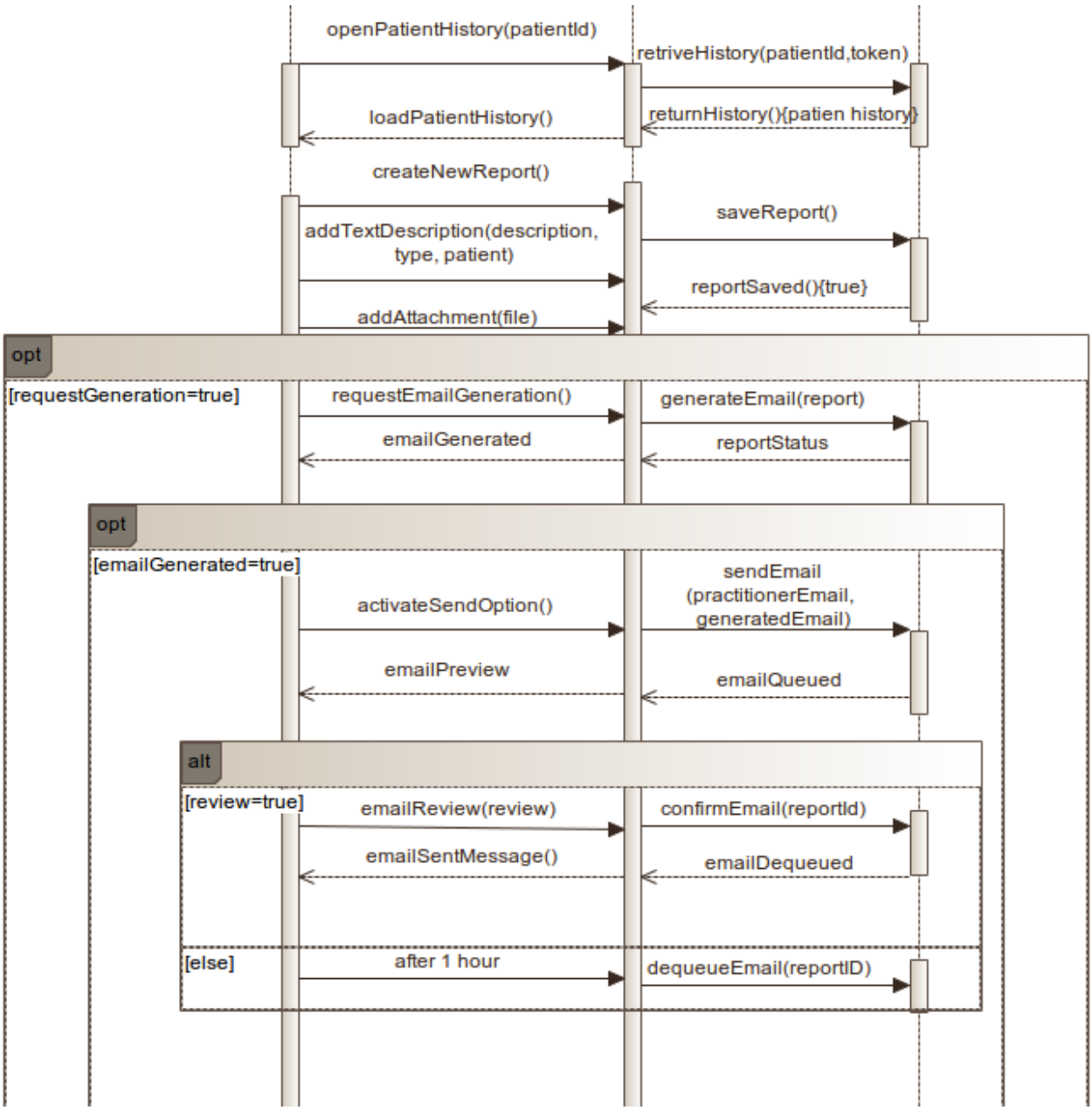
D Sequence Diagram

Next to the activity diagram is the State Machine Diagram, which is also a behavior diagram focus on how a structure within a system changes in response to event occurrences over time. It refers to the behavior that begins executing the moment a block is instantiated and generally finishes executing when that instance is destroyed



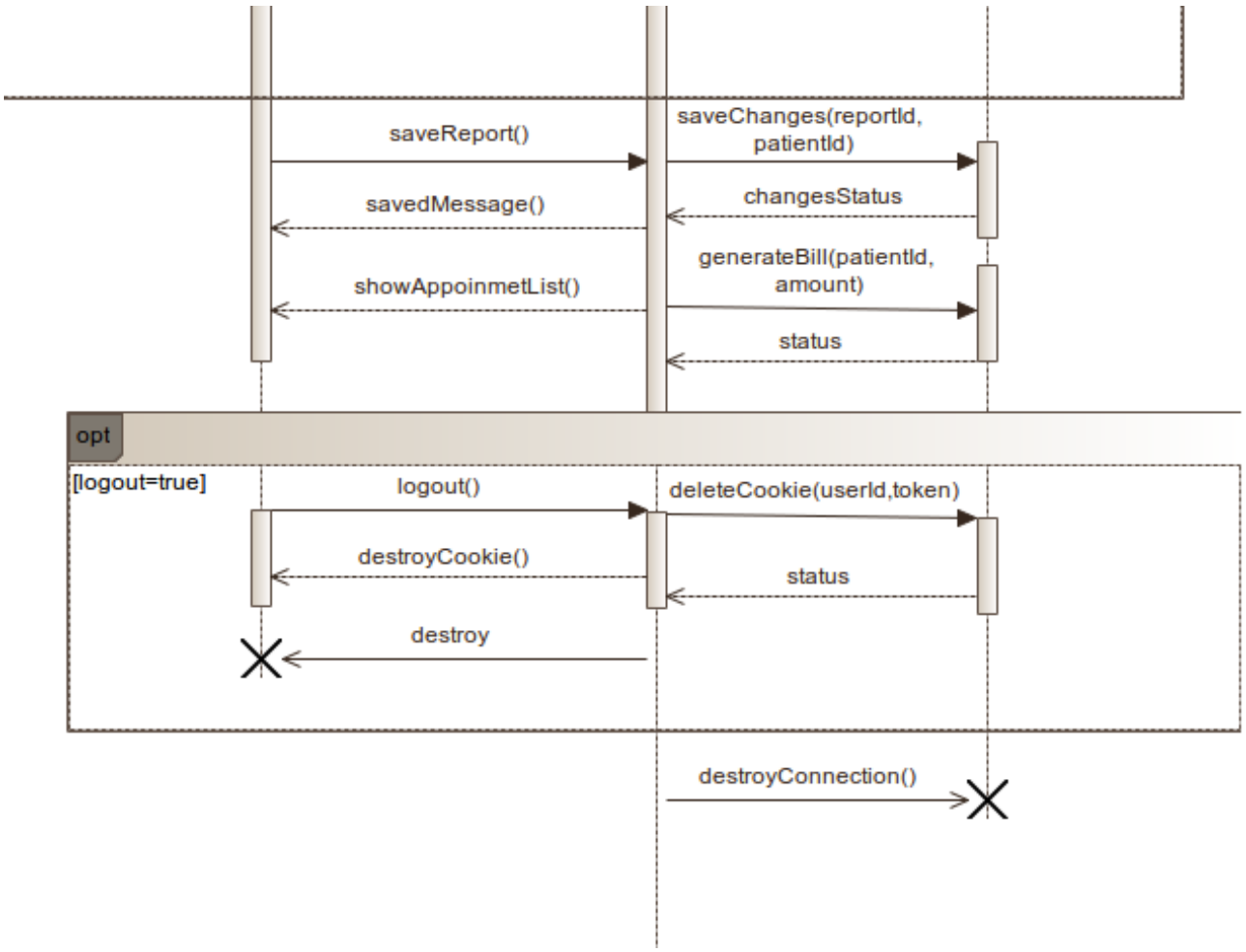
self-made

Figure 5: Distributed Computing Implementation.



self-made

Figure 6: Distributed Computing Implementation.

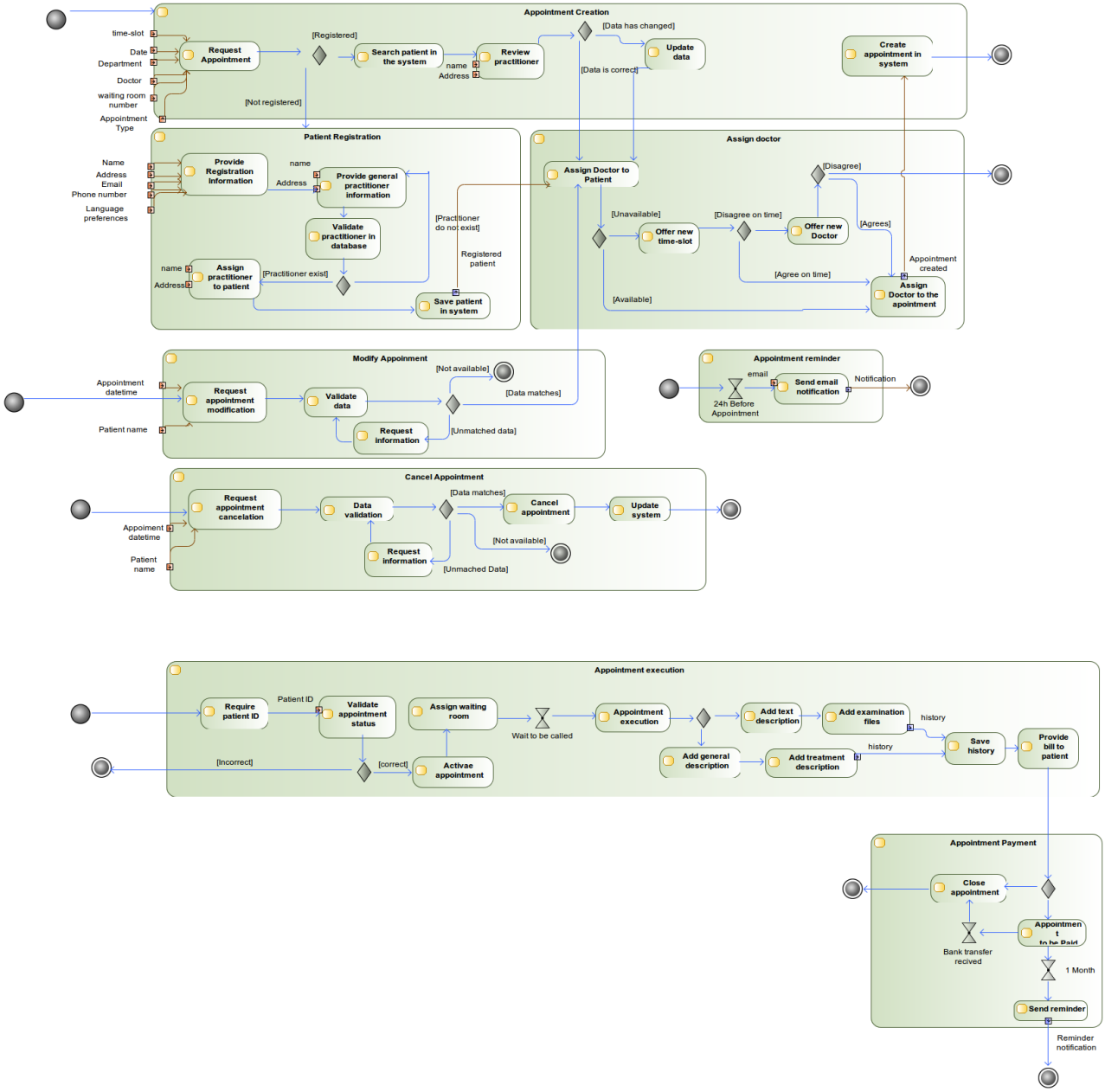


self-made

Figure 7: Distributed Computing Implementation.

E Activity Case Diagram

The activity diagrams are dynamic views of the system that expresses sequence of behaviors and event occurrence over time (Delligatti, Steiner, Soley, 2013). Together with the state machine diagram, the system behavior is expressed.

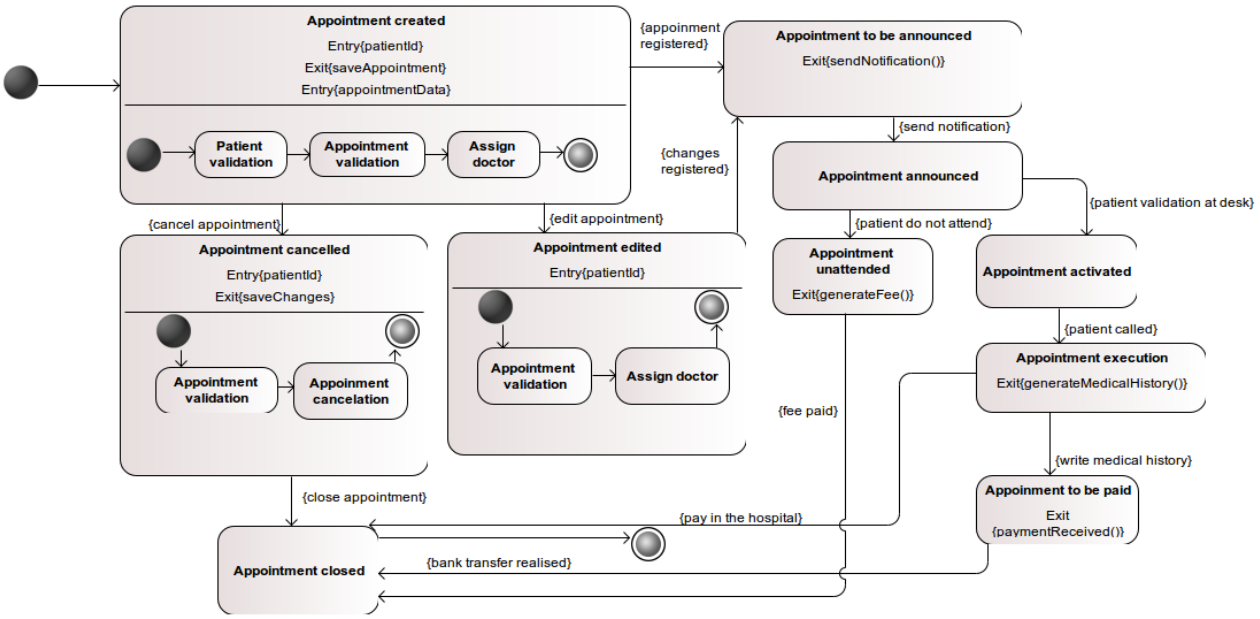


self-made

Figure 8: Distributed Computing Implementation.

F State Diagram

In this project, we tried to go further with the implementation by combining concepts that we learned in previous courses of the master, such as Web technologies and Databases. We tried to simulate in a "real case" environment, where several devices with different applications or different back-end (e.g. mobile application in Python and web application in Nodejs) are used to collect a specific information and save it in a pre-defined database.



self-made

Figure 9: Distributed Computing Implementation.