Software Project Report

Full Unit – Final Report

Appointment Scheduling System

Maria Afara

A report submitted in part fulfilment of the degree of

BSc (Hons) in Computer Science

Supervisor: **Dr Rami Tawil**

Department of Computer Science

Section 5, Lebanese University

January 17, 2018

Table of Contents

Getting Started

Introduction:

If Appointments are the vehicle that moves a service organization forward, scheduling is the fuel. Vroom, vroom.

Objectives:

The appointments system allows for quick booking and managing of patients appointments while eliminating the possibility of reiteration of the same time slot for different patients.

1- Establishment of a paperless environment

2- Scheduling of the dental activities within the dental center or clinic

3- Remind Dentists and Patients of any change in appointments

4- Patient does not have to wait in queue to visit the dentist.

5- This system saves time of both dentist and patient.

Usage

This Software is controlled by the admin, who adds the dentists and receptionists and Manage all their information's.

Through this system, dentists can easily manage the appointment slots online. System will make all the empty slots visibly available to the patients, and receptionists. The system makes it easy to manage various dentists and their availability on various dates and timings. Doctors maintain the medical history of the patient into their database so that every time any particular patient logs into the system, he/she can easily access their complete medical history whenever needed. In addition, it helps the doctor to refer their patient's medical history for any further prescriptions. This allows for an automated patient dentist handling system through an online interface.

Patient Form and Login Module

For the first time visitors, they have to just enter their basic details and can enter their dashboard. System will take care of creating their new profile. For existing patients, they will have to enter their id and password hey created earlier. This module will like virtual office from where all activity can be performed.

Online Appointment Module

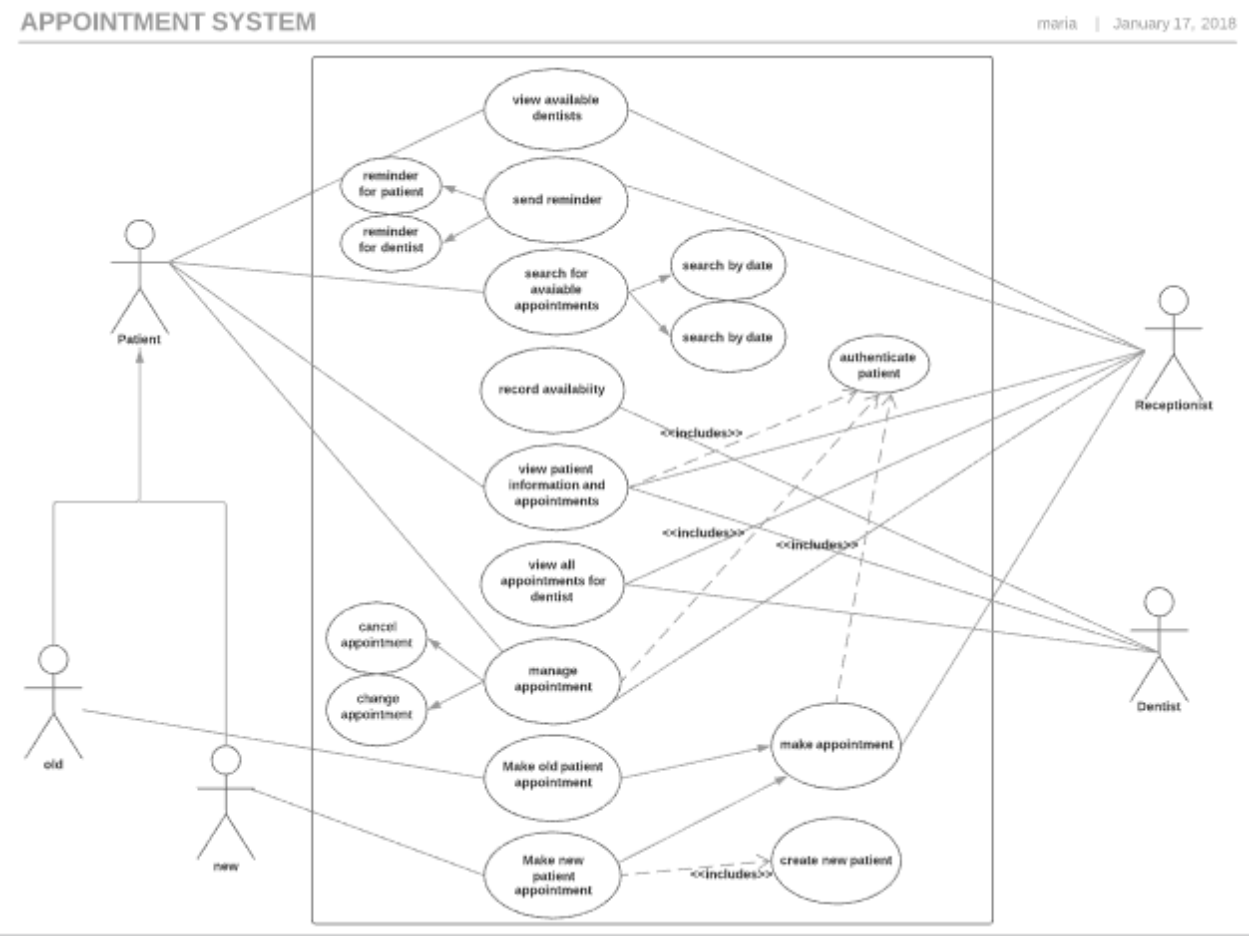
Through this module, patients can select dentists and have discussion regarding their health problems means select dentist that suits their conditions. Patients will able to get their availability time or choose from the available ones and book their appointments.

Advantages

- This system help the patient to appoint quickly whenever required.
- Patient does not have to wait in queue to visit the dentist.
- This system saves time of both dentist and patient.
- Patient can view his medical history whenever required.
- Ease the job for the receptionists.
- Patient does not have to carry medical prescription whenever he visits the doctor.

Use case model

1. If a system is to be seen as having high quality, it must meet the needs of its users.
2. So I took a user-oriented approach to systems analysis.
3. I identified the users of the system and the tasks they must undertake with the system.
4. I also seek information about which tasks are most important, so that I could and had planned the development accordingly.



Main Use Cases

Main use cases are listed below by primary actors as,

Patient

Make Appointment: The Patient ask for a new appointment or change an existing appointment or cancel an existing appointment

Receptionist

Manage Patient's Information: The Receptionist creates new patient and updates existing patient details regarding an appointment is made by the Patient

Maintain schedule: The Receptionist checks the schedule for the patient appointment with the dentist availability and confirm the appointment

Dentist

Record Availability: The Dentist updates the availability into the systems to produce schedule for the patient's appointment

Class diagram model

Identifying the right classes is one of the main skills of OO development.

1. First identifying a list of candidate classes.

→ Clinic, appointments and treatment system.
Before seeing a dentist or receptionist the patient needs to make an appointment. The appointment will be made by the receptionist in the clinic, before making the appointment the patient needs to ask the receptionist which dentist they wish to see. The receptionist will use this information, check the appointment schedule and find a free slot and make the booking. When the patient sees the Dr, the Dr will sometimes issue a description on the treatment they made. Receptionists can also cancel appointments. Each doctor has a maximum of 2000 patients registered to them.

 Underline its noun and noun phrases, that is, identify the words and phrases that denote things
 This gives a list of candidate classes, which we can then whittle down and modify to get an initial class list for the system

2. In this particular case, we discard:

 Free slot, because it is vague (we need to clarify it).
 System, because it is part of the meta-language of requirements description, not a part of the domain.
 Booking, because it's an action done by receptionist for example in this case listed above.

3. This leaves:
   - Dentist
   - Patient
   - Schedule
   - Clinic
   - Treatment
   - Receptionist
   - Appointment

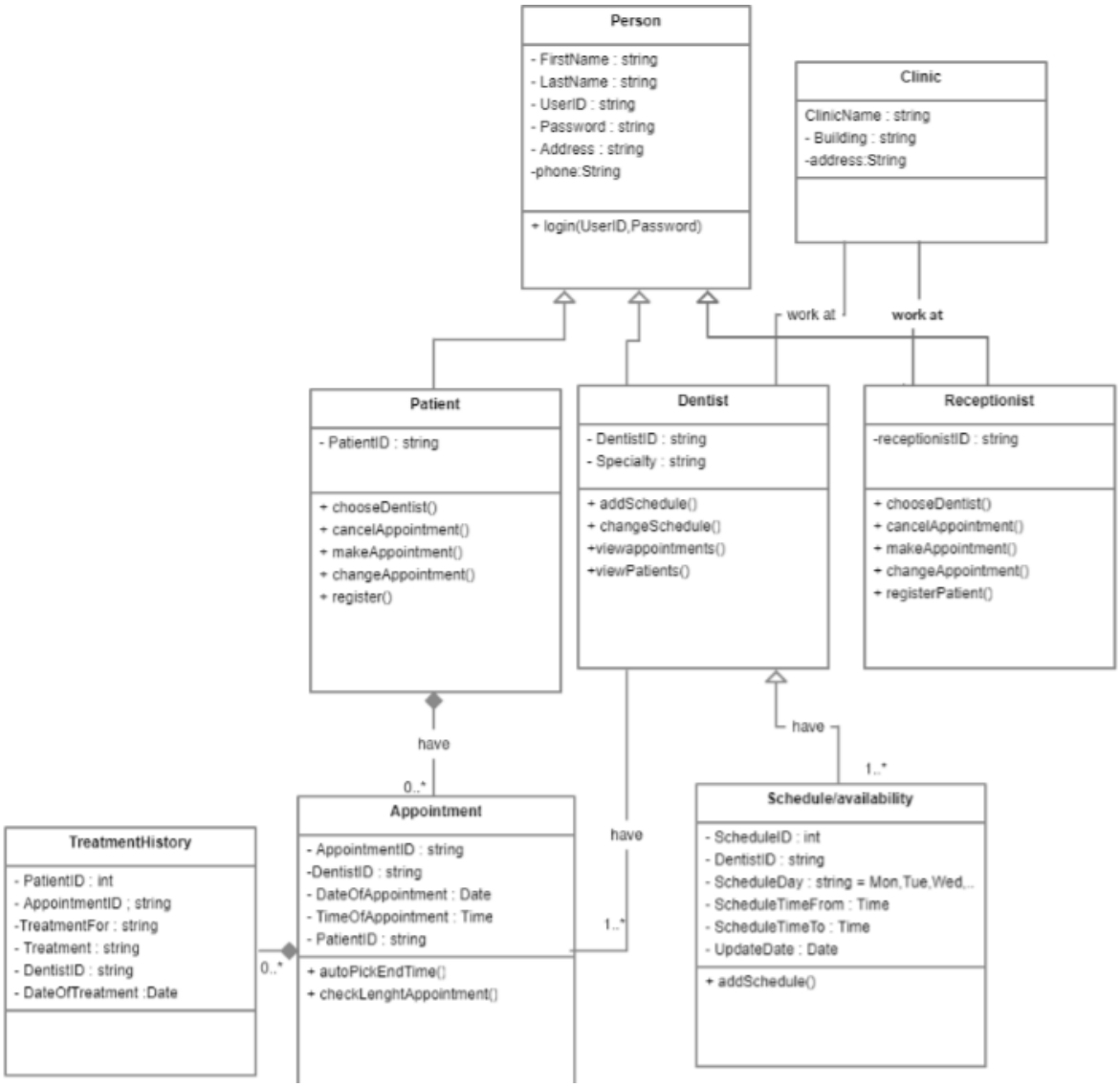
4. Relations between classes:

 Next we identify and name important real-world relationships or associations between our classes

 We do this for two reasons:

  To clarify our understanding of the domain, by describing our objects in terms of how they work together;

  To sanity-check the coupling in our system, i.e. make sure that we are following good principles in modularizing our design



Sequence diagram model

The System in Action

 A class diagram gives a static view of the system, but we know nothing about the dynamic behavior

 In UML we can use interaction diagrams to show how messages pass between objects of the system to carry out some task

-This will also show how the various classes realize the different use cases we identified in the use case diagram

An Example Sequence Diagram

 Consider what happens in the appointment booking scenario when a user wishes to make an appointment

- The receptionist must check that the person is a valid patient
- Then the doctor object must be checked to see if there are any available appointments
- If there are any suitable slots available, a new appointment should be created and assigned to the doctor.

 We now see how this is recorded in a sequence diagram

Interaction Shown on a Sequence Diagram

In this example the interaction is very simple, there is a single execution path and nothing occurs in parallel; in more complex scenarios, sequence diagrams can clarify the working of a system to a greater extent.

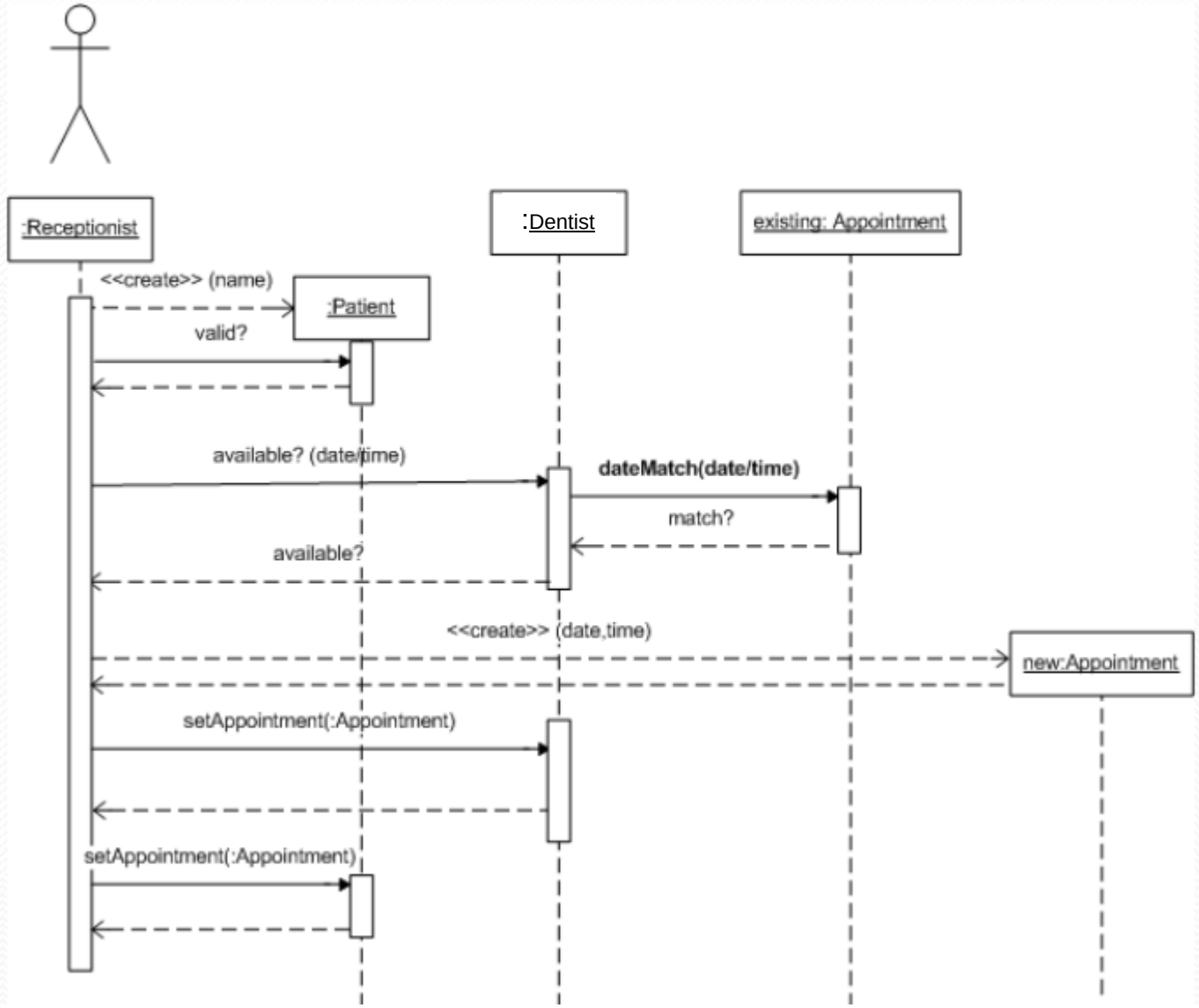

:Receptionist
:Dentist
existing: Appointment
<<create>> (name)
:Patient
valid?
available? (date/time)
dateMatch(date/time)
match?
available?
<<create>> (date,time)
new:Appointment
setAppointment(:Appointment)
setAppointment(:Appointment)

Figure1

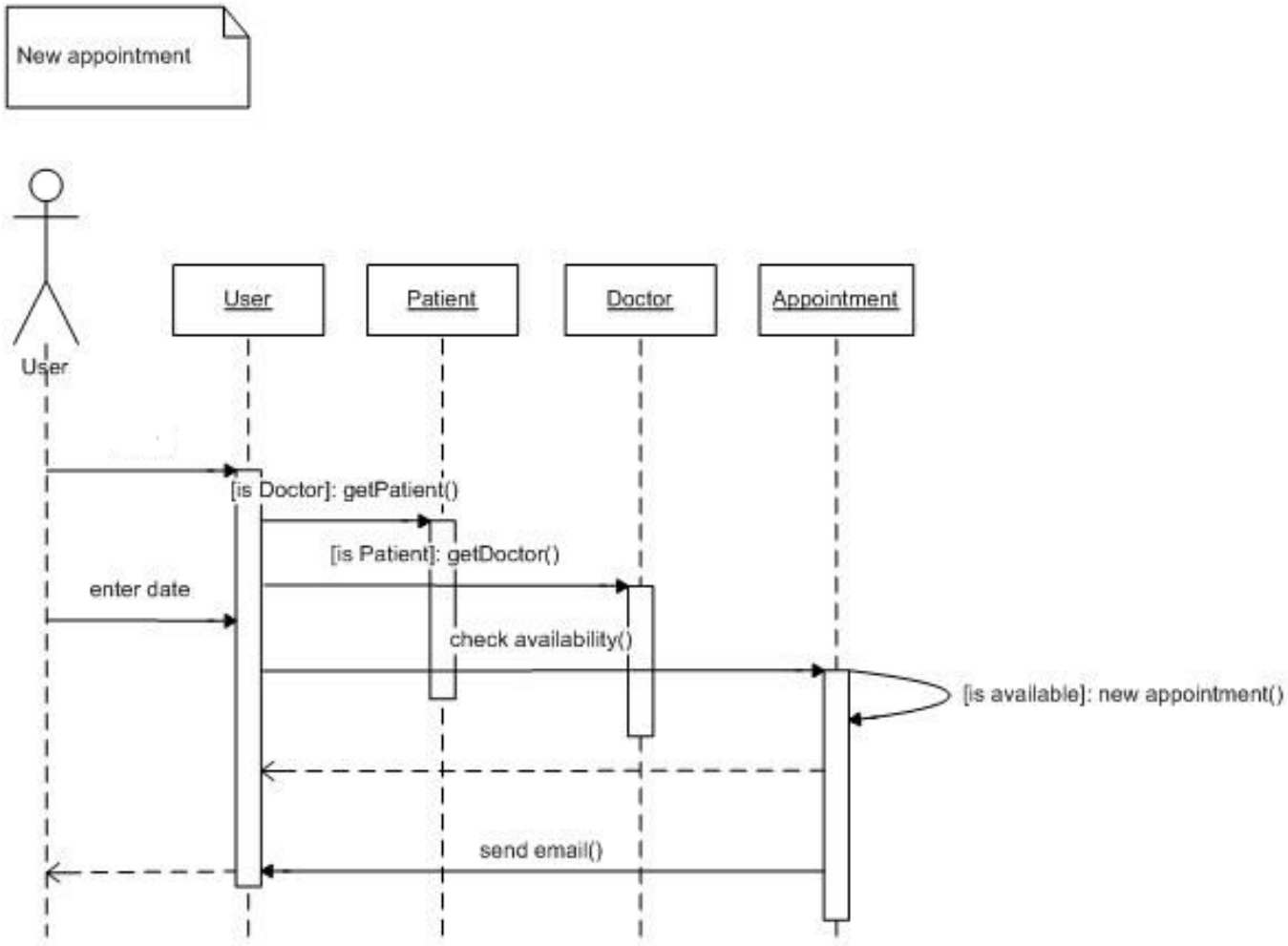

New appointment
User
User
Patient
Doctor
Appointment
[is Doctor]: getPatient()
[is Patient]: getDoctor()
enter date
check availability()
[is available]: new appointment()
send email()

Figure 2

Activity Diagram model