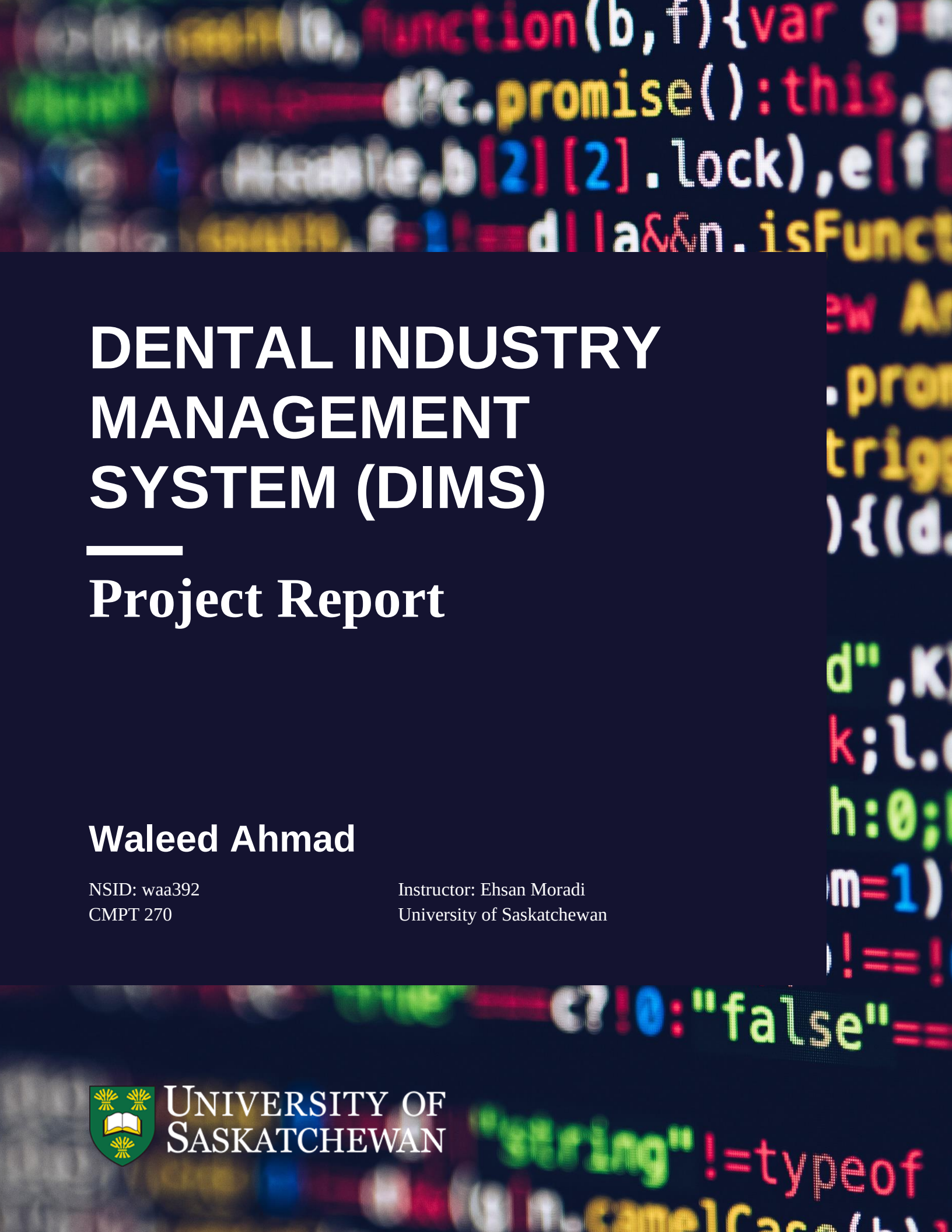

DENTAL INDUSTRY MANAGEMENT SYSTEM (DIMS)

Project Report

Waleed Ahmad

NSID: waa392
CMPT 270

Instructor: Ehsan Moradi
University of Saskatchewan

UNIVERSITY OF SASKATCHEWAN

Acknowledgement

Designing a software solution for managing dentistry industry was an interesting journey where I learned a lot about real world applications of software designs.

I would like to express my gratitude and appreciation to all those gave me the possibility to complete this project. Especially to my instructor and project supervisor Ehsan Moradi who came up with the project idea.

Further on I want to thank Campus Dentist staff for providing me with the patient registration forms being used in the industry. And to Pharmacist Alia Anber for providing me with the detailed insight of how current management systems works in medical fields.

Abstract

The dentist's industry is growing rapidly all around the globe and is expected to keep this trend in future as well. As a growing industry there is a potential to digitalize and automate the administrative tasks to increase efficiency of the entire industry and make the process of providing and getting dental care a hassle-free experience.

The DIMS provides with all the essential features to achieve the mentioned goal. It is a management system that had specialized features for the user depending on their position in the industry i.e., front desk, dentist, and HR. The system is modular and easy to maintain and update. Adding new features with time is made easy and thus make this an ever-evolving software solution for this industry.

The language used to design this system is JAVA which is not only fast but also stable to run this system. Different departments of industry are linked together to make communication between departments efficient thanks to Object Oriented approach of JAVA. Not only the system provides useful functionality but is also equipped with basic form of GUI to make the user experience better. File management is optimized in this system so that the related departments can save useful files on external storage.

The DIMS is capable automate many tasks related to dentistry industry's administration and is the best solution in the market considering the resource consumption and efficiency.

Table of Contents

Table of Figures

Chapter 1: INTRODUCTION

1.1 Overview of Project

This project is created to streamline the processes in the management of dentistry industry. It eliminates manual bookkeeping to a great extent by automating all those processes presented on a clean Graphical User Interface (GUI). The software solution simultaneously handles three different departments of dentistry industry i.e., Front Desk, Dentists, and HR.

All these department panels are accessed once the user enters a valid username and password. For the user to access the software they must be registered in the system as an employee and must enter a valid password that they set at the time of their registration.

Front Desk Panel handles all the tasks that links a patient to the dentist such as setting up appointments, keeping track of dentist's schedule, registering new patients, collection of their medical history and much more.

Dentist's Panel has options to access all the information that helps the dentist in the process of diagnosis. Features such as being able to view patient's medical history on a click of a button comes quite handy. Other features also include adding notes and prescriptions to a patient's file.

HR Panel has all the options that are related to the employees of the firm. Features such as registering a new employee, being able to view and change employee's paychecks and other attributes helps them to keep track of all the employees and keep their information up to date.

All these different departments are integrated with each other in a way that they provide seamless and hassle-free experience to the user regardless of their tasks in the firm.

1.2 Simplified UML Diagrams

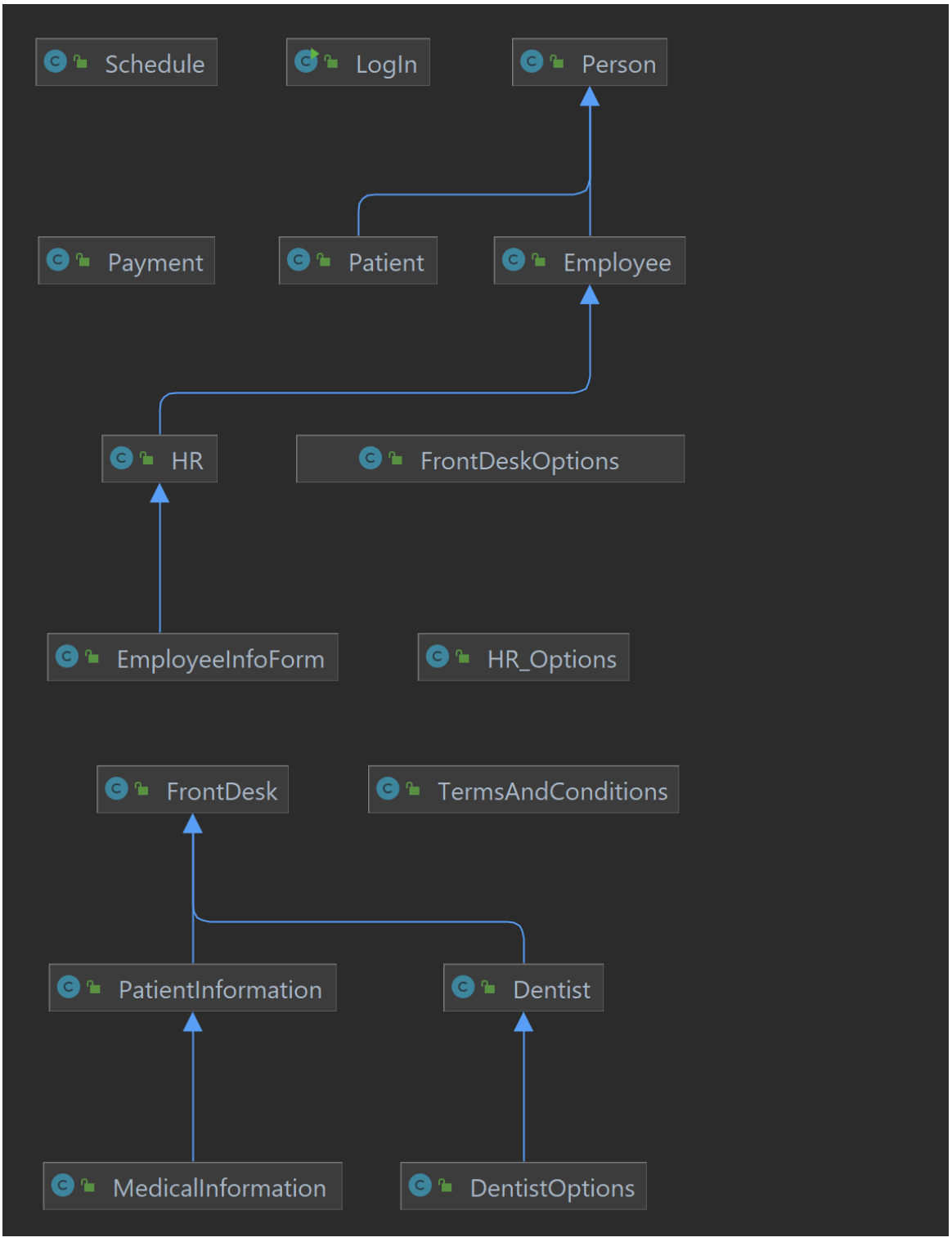

Figure 1 Simplified UML showing all the classes and inheritance hierarchy

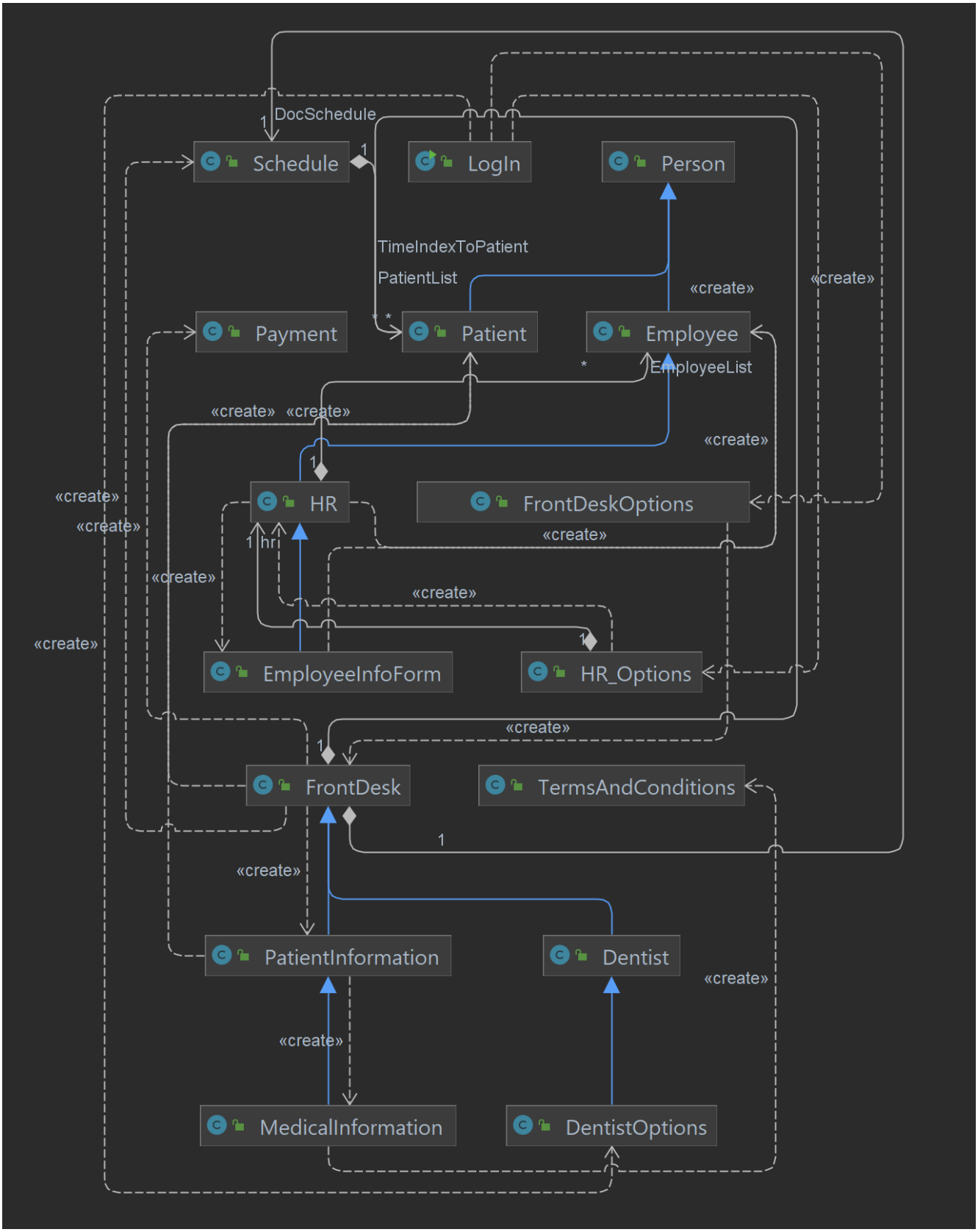

Figure 2 UML Diagram with visible dependencies

Chapter 2: DESIGN

2.1 Problem Statement

Design a management system for dentistry industry that not only provides with all the necessary features but is also fast, modular and easy to update. There must be just one management system to handle different departments of the industry.

2.2 Class Diagrams and Relationship Between Them

Following are the class diagrams showing all the attributes and methods of each class to give a detailed insight of the working code.

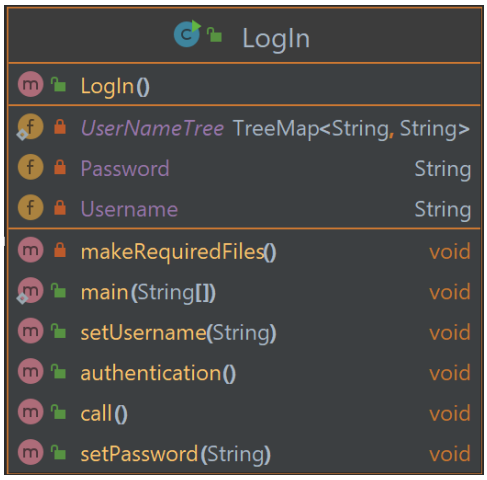

Figure 3 Login Class

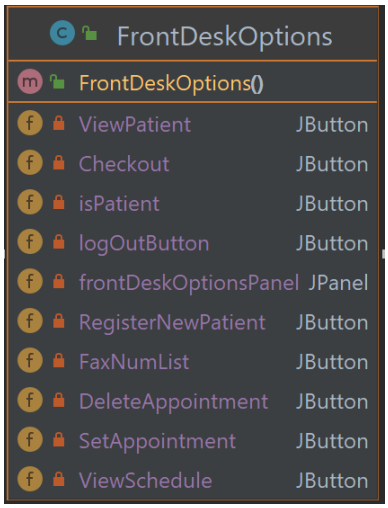

Figure 4 Front Desk Option Class

Figure 5 HR Option Class

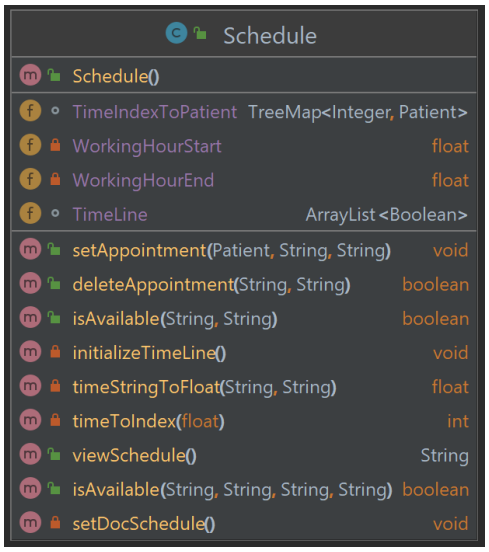

Figure 6 Schedule Class

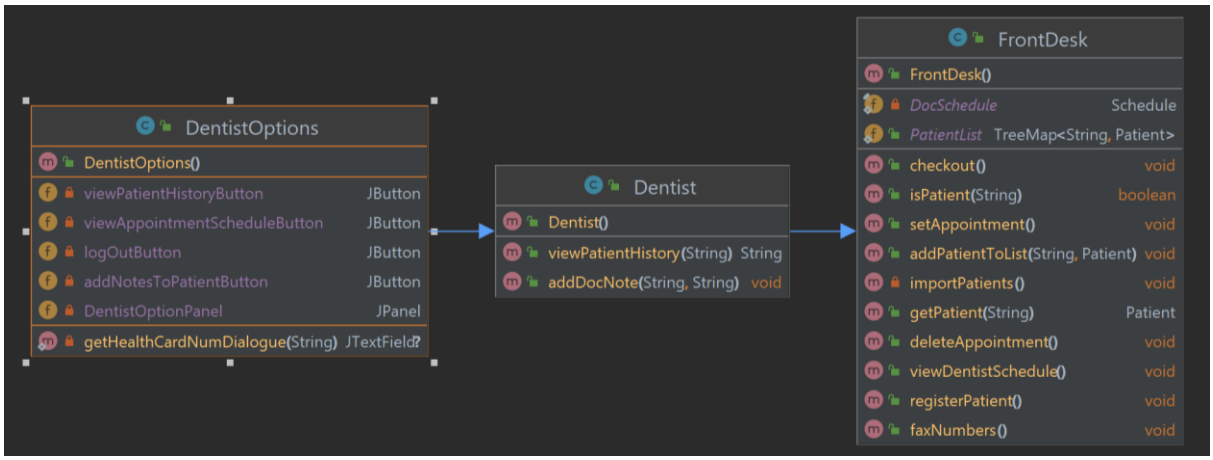

Figure 7 Dentist Options, Dentist, and Front Desk Class

Figure 8 Front Desk, Patient Information, and Medical Information Class

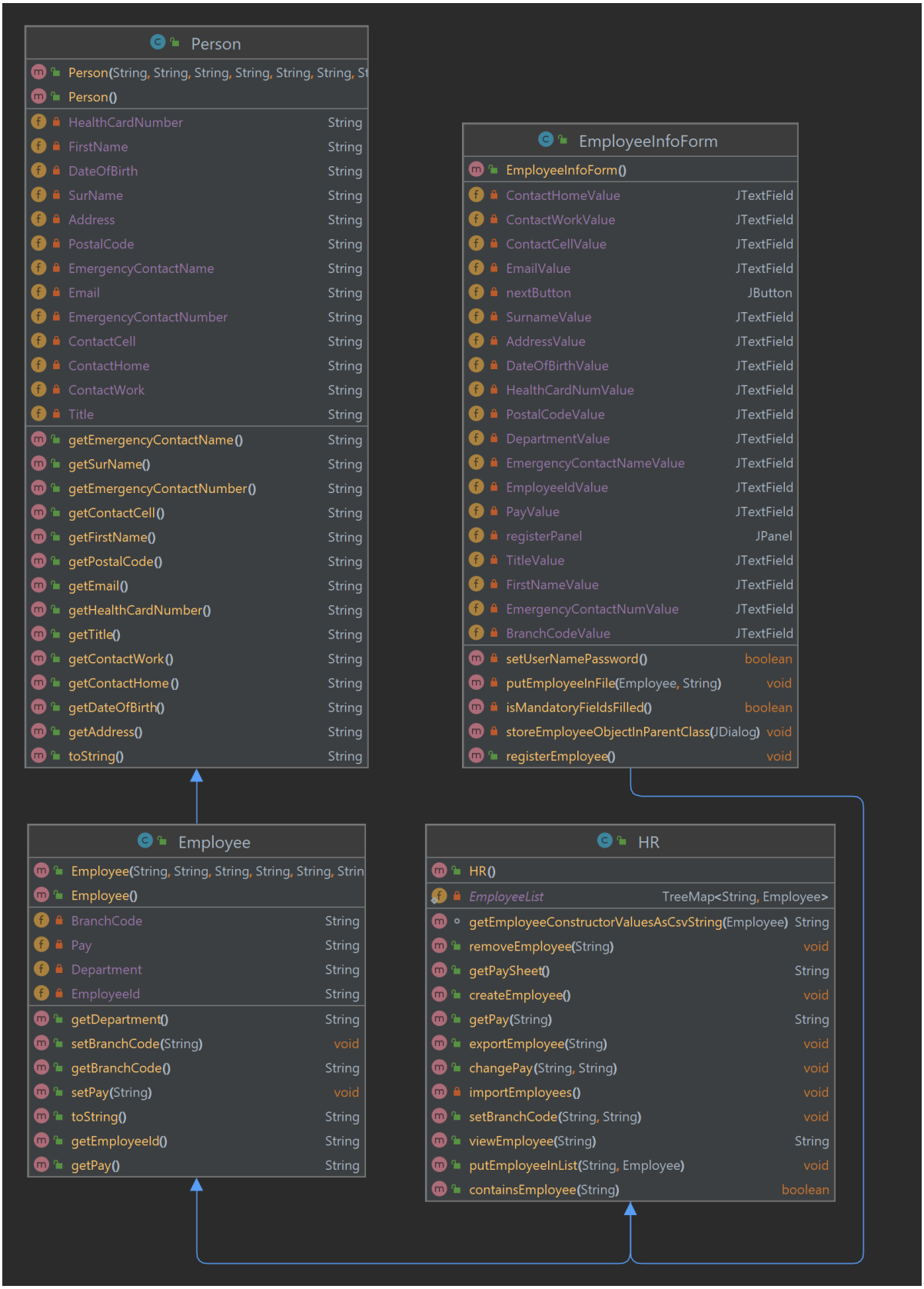

Figure 9 Person, Employee, Employee Info Form, and HR Class

Chapter 3: Brief Working of Code

3.1 Checking for dependencies

The code starts from the Login class. Before doing any other task this class checks for the files and directories that are essential for the smooth execution of the app. If those files already exist then the program starts its normal working, else it creates those files and directories first before moving on with the program. These files include

a) *A Patient Directory:* This directory stores individual file of every patient with their personal data and medical history in it.
b) *A file named username.txt* that stores all the usernames set up by employees, and their affiliated passwords
c) *A file named PatientRecord.txt:* This file stores values of constructor parameters of "Patient Class" in order, where each parameter is separated by a comma and each patient record is on a separate line. This file helps to store and recreate any patient in the system.
d) *A file named EmployeeRecord.txt:* This is just like PatientRecord.txt file but it is for employees. It stores constructor parameters of "Employee Class" in order, where each parameter is separated by a comma and each patient record is on a separate line.
e) *A file named PharmaciesInfo.csv:* This file contains the names, address, and fax number of all the affiliated pharmacies with the firm. This helps front desk workers to send prescriptions of patients directly to the patient's preferred pharmacies.

3.2 Password validation and username convention

After ensuring that all the dependencies are met, the class prompts the user to enter a username and a password (the process of setting up a username and password will be explained later). It checks for the validity of the password by going through username.txt.

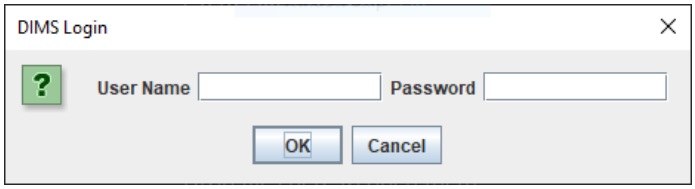

Figure 10 Login Panel

The username follows a convention *<Employee ID>@<Department>*. For example, if there is an employee at front desk with an Employee ID of waa392, their username would be *waa392@frontdesk.* Following the same convention there could be three possible types of usernames:

a) <Employee ID>@frontdesk
b) <Employee ID>@dentist
c) <Employee ID>@hr

There also exist a super user for each department. This superuser is for maintenance purposes and to get the system started. The username and password for these superusers are hard coded in the code i.e., admin@<Department> with a password “0000”.

Once the username and password has been validated the Login class call an appropriate “Options” class depending on the username type, providing the options of relevant department.

3.3 Front Desk Options

This class provides a GUI form with all the available options as buttons, but before loading all those features the front desk employee must set the shift timing of the dentist. The panels can be seen in figure 11 and figure 12. Shift timings of the dentist that must be set once a front desk employee logs into their portal

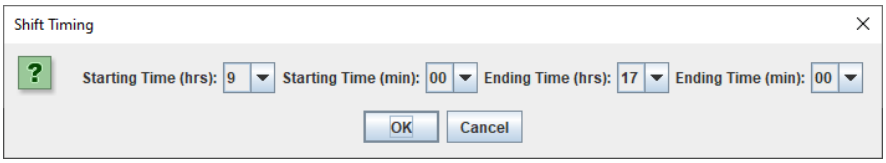

Figure 11 Shift Timings Panel

The class also performs some background tasks as soon as it is initialized. Most important of which is that it loads up all the patient’s stored in the PatientRecord.txt and stores them in a Tree Map in a class. By doing so the program can keep track of patients that were registered in previous sessions.

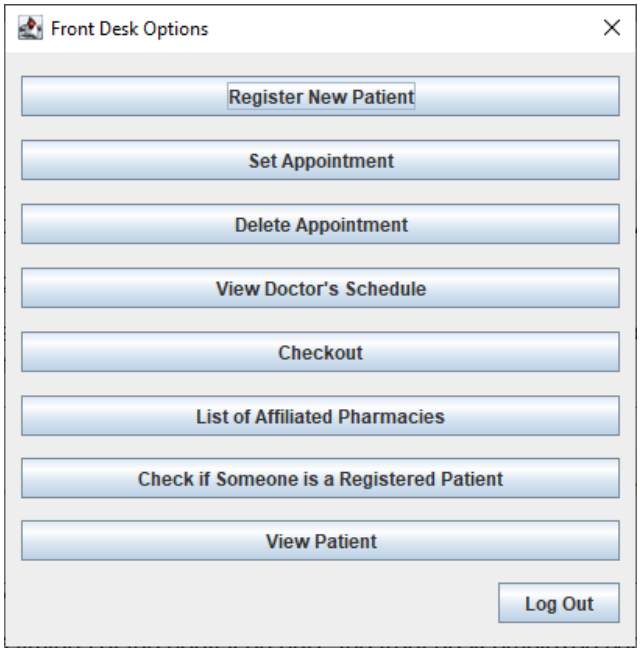

Figure 12 Front Desk Option Panel

3.3.1 Register Patient

Registering a patient is a crucial task that is carried out by using few GUI panels to collect patient’s information and medical history, getting their agreement to the terms and conditions of the clinic then storing this information in the Tree Maps of class as well as appending that information to the PatientRecord.txt file.

The create new account panel would make sure that the user fills out all the mandatory fields. if mandatory fields are not filled, the missing fields would turn red and would not proceed any further unless the fields are filled out.

The panel also makes sure that there is no other patient already stored with the same Health Card Number.

By clicking the "Next" button another form with the name of Patient Medical Information would pop-up to collect past Medical History of the patient.

This form allows the user to only select one of the three checkboxes.

Once the user clicks "Next" button another form would pop up to get the patient's consent on Terms and Conditions of the clinic.

A patient is registered if and only if they agree to the terms and conditions stated.

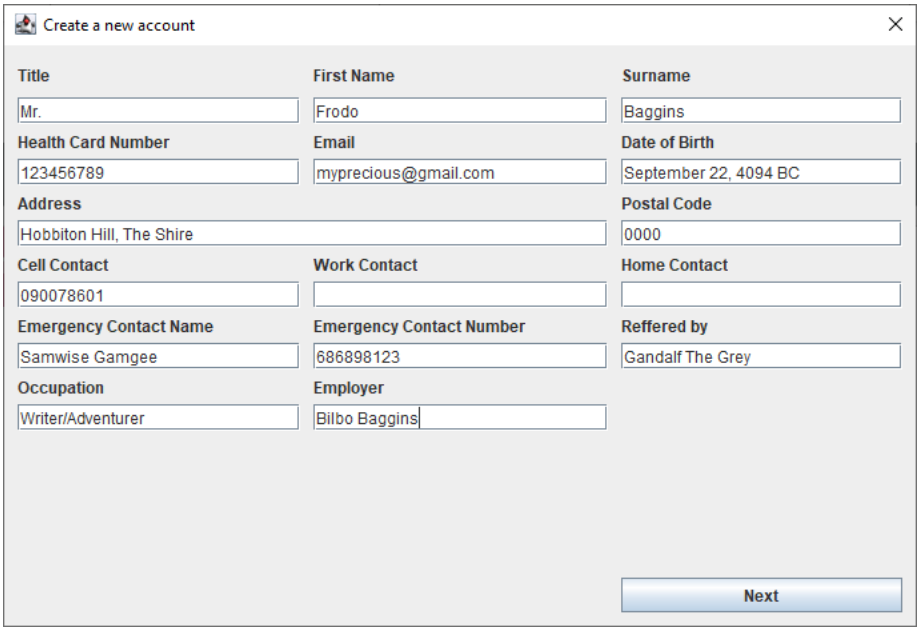

Figure 13 Patient Registration Form

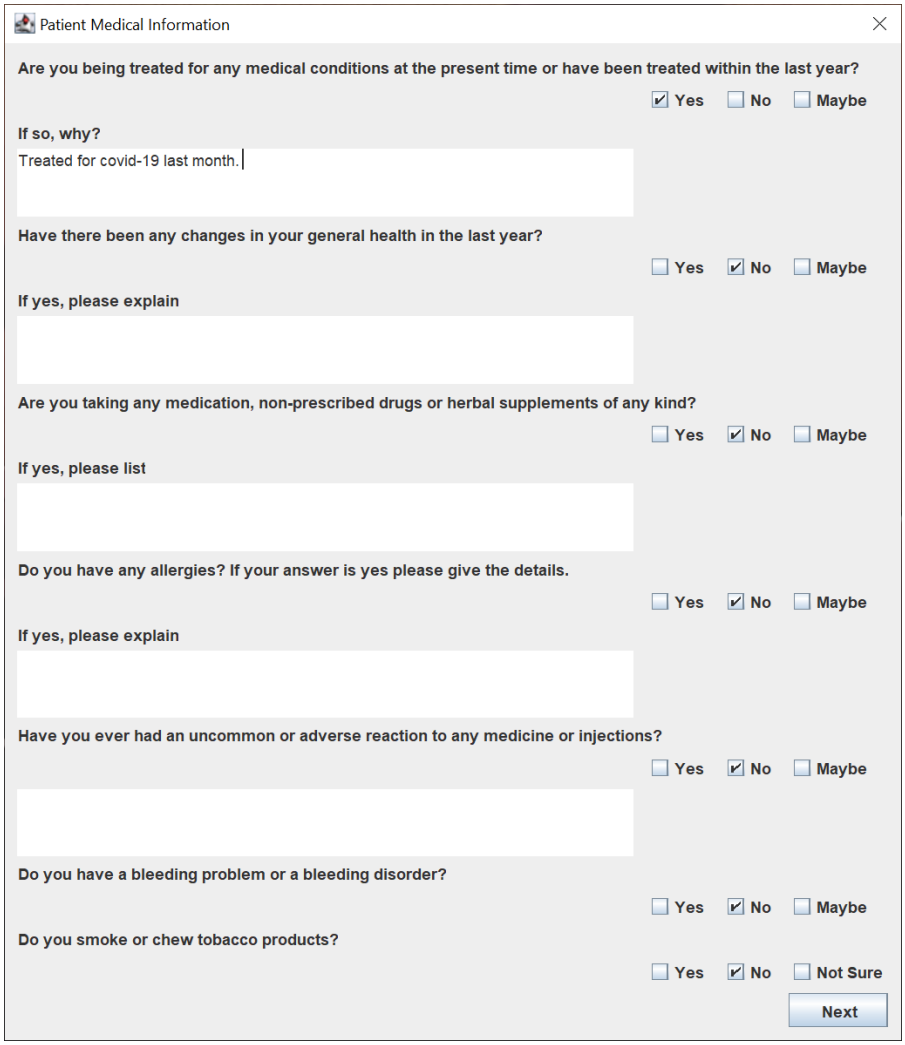

Figure 14 Medical Information/Medical History Form.

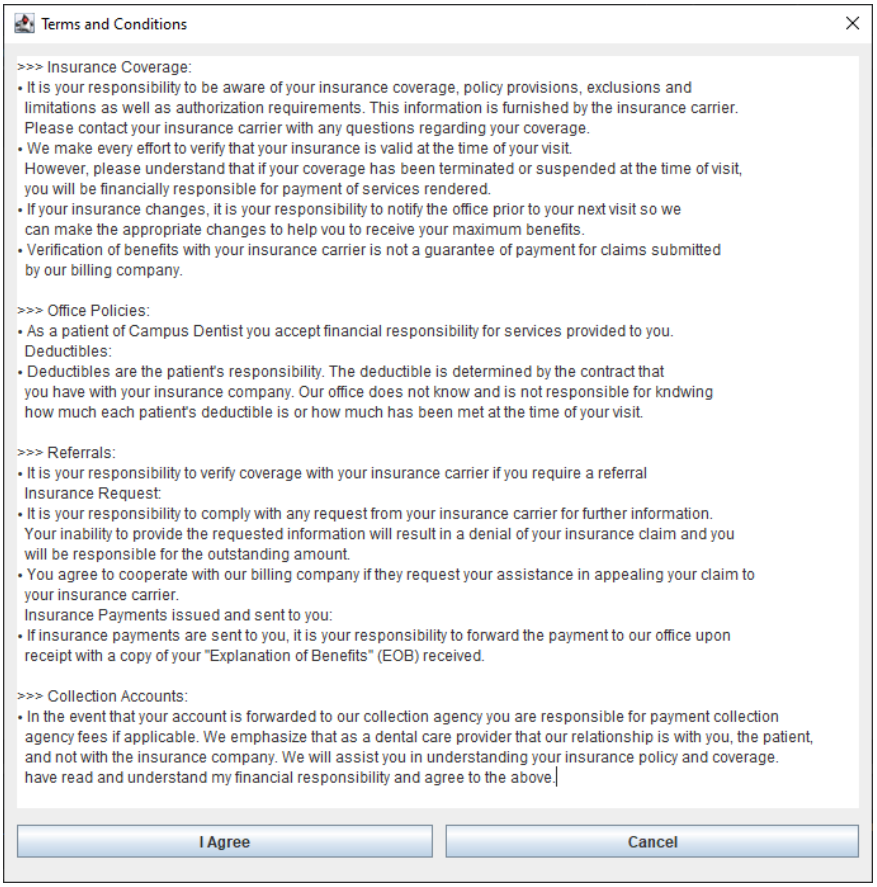

Figure 15 Terms and Conditions of the clinic.

3.3.2 Set Appointment

Setting up appointment uses schedule class functions to evaluate appointment conflicts and to assess the dentist's schedule. Each appointment is assumed to be of 30 minutes hence the dentist's schedule us divided into timeslots of 30 minutes each. The schedule class also keeps the record of each timeslot with associated patient who has reserved an appointment, using tree maps. The program Gives an error prompt if the patient provided is not already registered or if a timeslot is already booked by someone else. Pressing "OK" sets the appointment if no conflict is found.

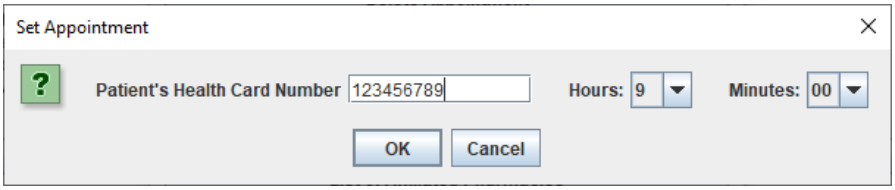

Figure 16 Set Appointment Panel

3.3.3 View Doctor's Schedule

This option displays all the timeslots during the dentist's shift and whether they are reserved or not. Only the time slots that are in the within the range of dentist's schedule are shown. In front of each timeslot the availability is displayed (either booked or Available). It also shows the health card number of the patient that has reserved that timeslot.

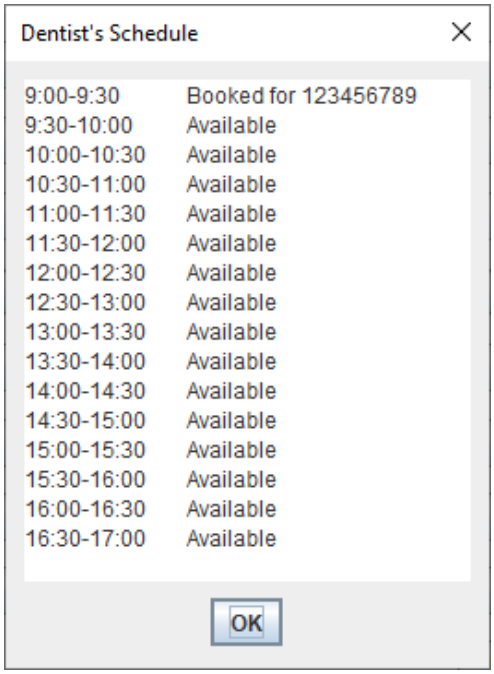

Figure 18 Dentist's Schedule Panel.

3.3.4 Delete Appointment

It asks the user to select the time at which the appointment needs to be deleted. If there exists an appointment at that time it is deleted, and timeslot is set to available otherwise the error prompt notifies the user if they try to delete an already available timeslot.

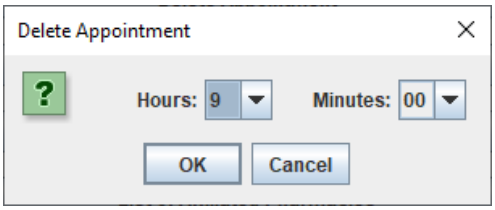

Figure 17 Delete Appointment prompt

3.3.5 Checkout

It prompts the checkout panel with some presets of costs as checkboxes. Selecting them changes the total price on the panel in real time. The panel also allows to add any additional cost that is not mentioned in the presets. Upon clicking "OK" a prompt informs about the total bill of the patient's visit.

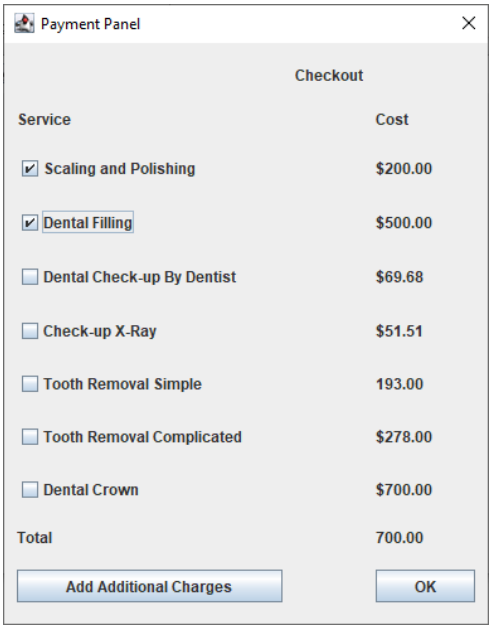

Figure 19 Checkout Panels

3.3.6 List of Affiliated Pharmacies

This option provides with a list of all the affiliated pharmacies that are read from an externally stored csv file named ***PharmaciesInfo.csv***. This helps the front desk workers to send the prescriptions of patients to their preferred pharmacies.

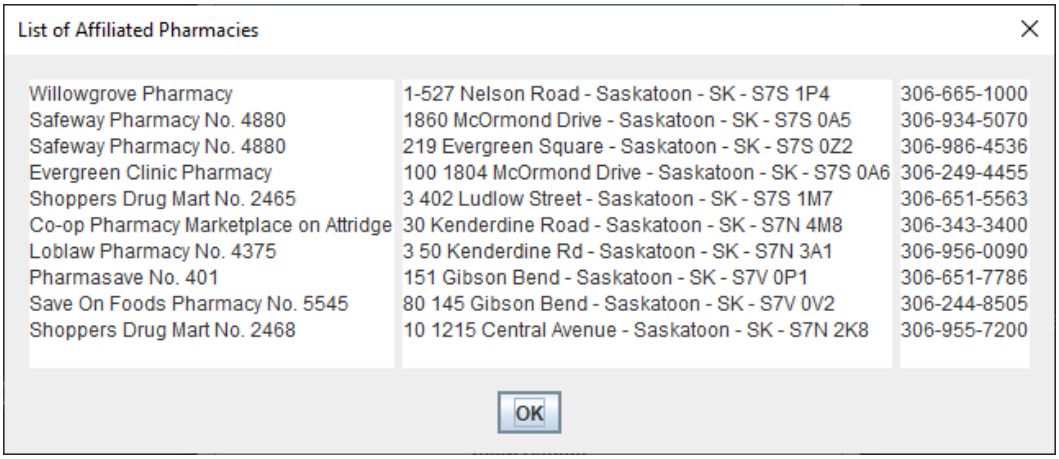

Figure 20 List of Affiliated Pharmacies

3.3.7 Check if someone is a patient

This is a quick check if someone is registered in the system or not. Asks the user for health card number and checks that number with the record of patients stored in “FrontDesk” class in the form of a tree map. If it finds the patient in the record it prompts with a message that weather the patient is registered or not.

3.3.8 View Patient

This feature gives the detailed information about the patient. The panel is shown after acquiring the health card number of the patient It also pops up a “Patient not found” prompt if an unregistered patient is tried to access.

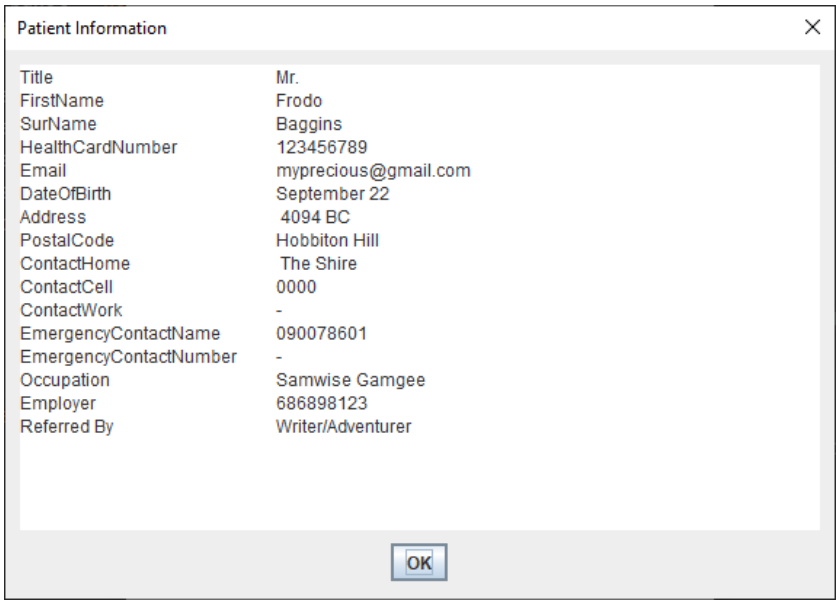

Figure 21 Patient Information Panel

3.4 Dentist's Options

This panel provides with all the features that helps the dentist in the process of diagnosis and the process dealing with the patient. Dentist's panel is equipped with only the useful features so that the already limited time of appointment does not get wasted by the dentist trying to navigate through the complexity of software. Hence most of the complex tasks of bookkeeping are assigned to front desk.
The dentist can view patient's history, add notes to patient's file, and view their own schedule of appointments.

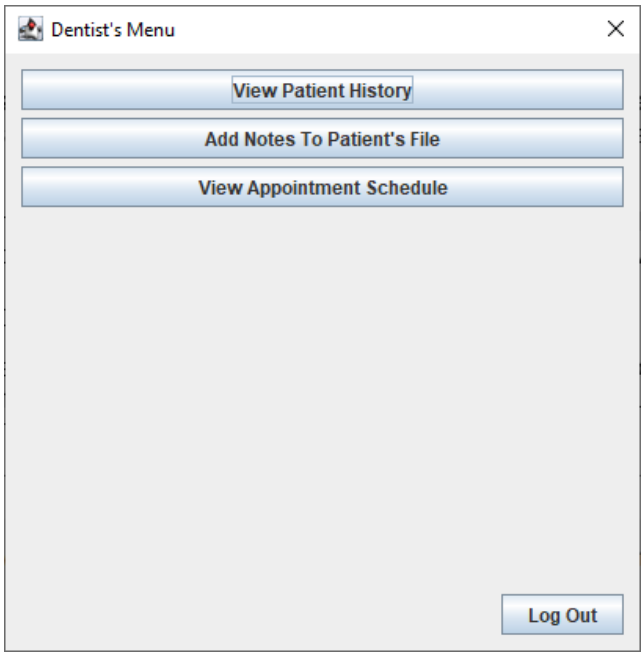

Figure 22 Dentist's Options Panel

3.4.1 View Patient's History

This feature asks for patient's health card number and then view their medical history collected at the time of registration including any noted added by the dentist later.

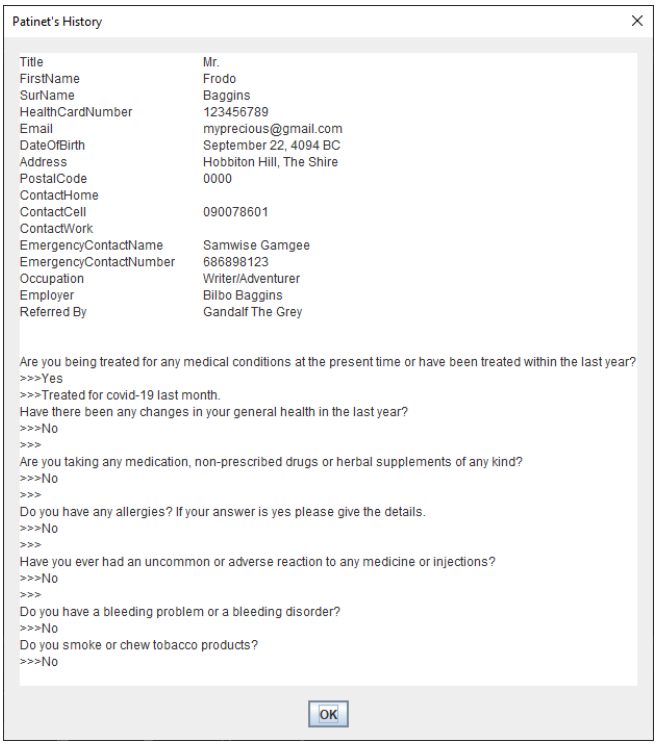

Figure 23 Patient's History Panel

3.4.2 Add Notes to Patient's file

This feature allows the dentist to add their diagnosis and/or prescription to the patient's file. Selecting this option would present the user with a text box to type in the note. This note is appended to the patient's file with a date stamp of when the note was added to the file. Hence the note would be visible to the dentist next time they display patient's history.

3.4.3 View Appointment Schedule

This feature works just like the "View Doctor's Schedule" from front desk. For more information check topic 3.3.3

3.5 HR Options

HR options provides with options related to the employees of the dental firm. For this version of DIMS the options for HR are limited for the sake of simplicity. There exist working methods of a lot more features that can be added to GUI to be accessed by users in next release. These features include accessing and/or changing individual attribute of an employee, viewing entire paysheet of the firm, viewing all the employees in the firm etc. In this release the working experience is introduced by implementing one of many similar features such as accessing and changing one attribute (Salary of an employee), features to access other attributes would have similar user experience.

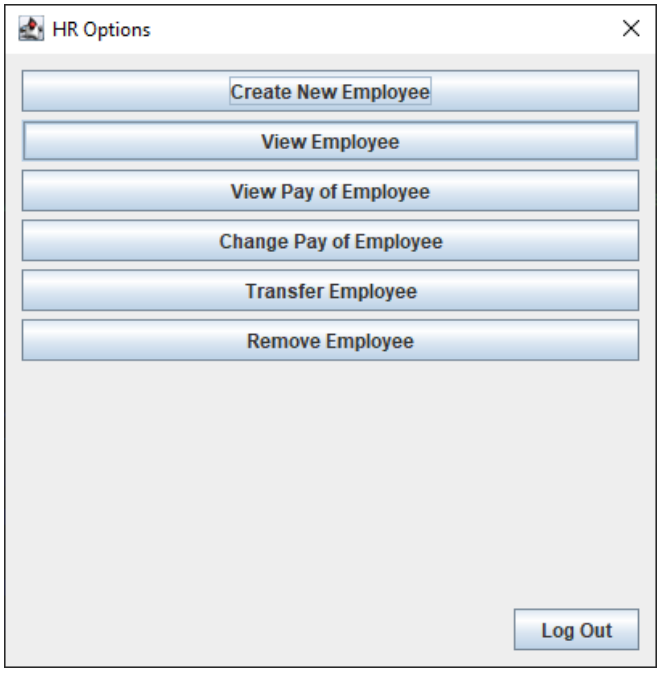

Figure 24 HR Option Panel

3.5.1 Create New Employee

This feature asks the user to fill out the Employee Registration Form. The form is designed such that it forces the user to fill out mandatory fields and makes sure that the does not already exist an employee with the employee ID that is entered in the form. Once the form is filled with valid entries the software generates a username for the employee consisting of their employee ID and department and asks the employee to set up a password. The password field must be filled and must match with confirm password field. Once this process is done an employee is registered and the record is stored in appropriate files.

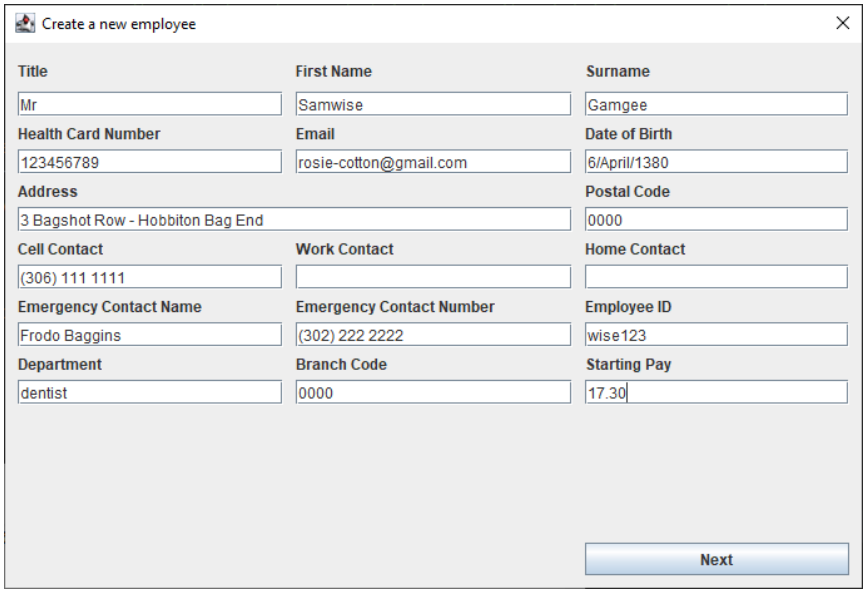

Figure 26 Employee Registration Form.

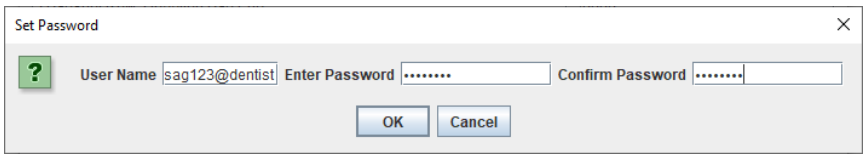

Figure 25 Set Password Panel.
The username field is auto generated and is not editable.

3.5.2 View Employee

This feature asks the user to enter an employee's ID and checks that ID in the system record. If found it displays the employee's information, if not it prompts an error popup. The popup is similar to Figure 22.

3.5.3 View Pay of Employee

After acquiring the employee ID this option displays the salary of that employee. This feature is a model for all other features that can be added to view any attribute of an employee.

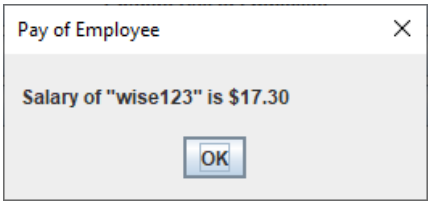

Figure 27 View Pay of Employee Panel

3.5.4 Change Pay of Employee

This feature asks for employee ID and their new pay that is to be set. This feature is a model for all other features that can be added to change any attribute of an employee.

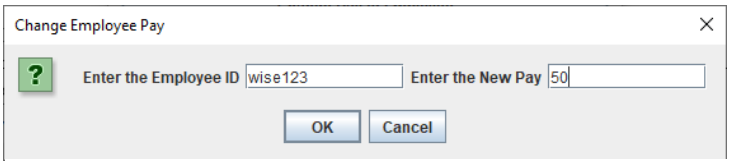

Figure 28 Change Employee Pay Panel

3.5.5 Transfer Employee

This feature changes the branch code of an employee thus transferring an employee to a different branch. The panel for this is similar to the one in figure 34.

3.5.6 Remove Employee

This feature asks for an employee ID and remove the provided employee form the system and from stored files. The panel prompt for this is similar to the one in figure 33-a

Chapter 4 Future Recommendations

DIMS version-1 does provide a fully functional practical administrative solution to dentistry industry management but like all software there is always room to improve. Here is a list of things that can be improved about DIMS making it even more powerful tool for management and making it even more resistant to bugs and unexpected behaviors.

- The biggest change that could be done to this is instead of storing all the data locally, cloud servers can be used. Doing that would make it more resistant to data loss and would provide with the advantage of access of data from anywhere around the world.
- The system creates a username file that stores all the usernames and passwords unencrypted. This poses a security threat which can be resolved by using any kind of encryption method to store password data and implementing the decryption algorithms at the time of reading the password file.
- The system checks for already existing health card numbers or employee ids and prohibits new entries to be created with the same attributes. This system is not applied on the username file allowing it to create multiple entries of username to create different passwords once again posing a security threat. This problem can be very easily solved by applying the same conditional logic that has been applied while storing patients and employees.
- For now, only few getters and setters have been implemented in the GUI of HR and Front Desk. But this can be extended using similar logic used in the ones that are implemented now.
- The registration forms for both employees and patients stores their respective constructor information in EmployeeRecord.txt and PatientRecord.txt files respectively with each entry being separated by a pipe character '|'. So, if the user puts a pipe character in any input field it would cause bugs while reading from the file. This process can be improved by implementing a conditional check on each input field making sure they don't have special character i.e., a pipe character, before storing it to file. The system already checks input fields for not being an empty string, so this check can be implemented alongside already existing check functions.
- GUI for the program can be improved making it more user friendly.
- Sending reminder emails to patients about their appointments can be implemented using Java Mail API.