

Team 35 Report

Introduction

The purpose of this report is to present our planning, designs and our query processing. The report will also include an evaluation, where we assess how well we have worked overall, and what changes could have been made.

We started by meeting up to review the scenario and evaluating the requirements. From there, we set ourselves the deadline to finish the diagrams within a week to leave as much time as possible for creating the software. In addition, we decided that after the diagrams were completed we would separate the work (implementation) between us and meet at least once a week. At each meeting, we discussed what we achieved individually so far, and what we planned on doing in the following week.

Before creating the software, we decided to separate the work using the UML information model, so that three of us would distribute the classes between us and their corresponding database tables. The last team member worked solely on the interface of the software. In the case of new work arising, we would assign the work to whose classes relate to it the most or whoever could handle the extra work. Every member was responsible for the testing of their own classes.

Design

Before committing to writing any code for the GUI, we decided it would be best to draw mockups of our design to solidify what we needed to include, and most importantly, how we were going to present the functionality most effectively.

Secretary View

The secretary view required a lot of thought to design due to the number of different tasks that the secretary needed to be able to do. It was important for them to be able to view varying information simultaneously. For this reason, we decided to design it as a split-view window. On the left-hand side, you can view all of the appointments by day and narrow them down by doctor. Since this is something that the secretary will be referring to very frequently, we thought it best to have that there permanently. On the right-hand side of the window however, we split it into 3 different sections: Register, Patients and Addresses. We could've had a 'Register' button inside the Patients tab, but since registering will be done frequently, we decided to have it accessible in one click instead. The Patients and Addresses tabs are used for searching for patients and addresses respectively.

The mockup illustrates the Secretary View GUI, which is a split-view window. The left-hand side is dedicated to viewing appointments, while the right-hand side is divided into three sections: Register, Patients, and Addresses.

Appointments Section (Left):

- Header:** "Appointments" with a "Search by doctor..." input field and a "Search" button.
- Appointment Cards:** A grid of appointment cards. Each card displays:
 - Start/End Time:** e.g., "Start 12:30", "End 13:10".
 - Doctor Name:** e.g., "Arthur Granacher", "Nuraldeeeeeen Mag...".
 - Appointment Details:** e.g., "Remedial | Appointment 1 of 3", "Cleaning | Single appointment".
 - Buttons:** "Details" and "Patient Info".
- Footer:** "Viewing appointments for" followed by a date selector (18/11/2017) and a "New appointment" button.

Right-hand Side Sections:

- Register Section:**
 - Title:** Input field "Enter title".
 - First name:** Input field "Enter first name".
 - Last name:** Input field "Enter last name".
 - Phone Number:** Input field "Enter phone number".
 - Date of birth:** Input field "05/05/1998".
 - House Number:** Input field "House number or name".
 - Postcode:** Input field "Enter postcode".
 - Search address:** Input field.
 - Register patient:** Button.
- Patients Section:** (Placeholder for patient search results).
- Addresses Section:** (Placeholder for address search results).

The final design for the secretary view differed slightly from the planned design. The major design change was having two tabs for each doctor for the secretary to be able to see their appointments separately and in a week-view (see below).

The screenshot shows two side-by-side windows from the 'Dental Clinic System'. The left window, titled 'Dentist Appointments', has tabs for 'Dentist' and 'Hygienist'. It features input fields for 'Patient First Name' and 'Patient Last Name' with a 'Search' button. Below is a calendar view for 'Monday 27', 'Tuesday 28', and 'Wednesday 29'. At the bottom, it shows 'Week from day: Nov 27, 2017' and a 'Refresh (View All)' button. The right window, titled 'Patient Registration', has tabs for 'Register', 'Patients', 'Addresses', and 'Book Appointment'. It contains a form with fields for 'Title', 'First name', 'Last name', 'Phone Number', 'Date of Birth', 'House Number', 'Street name', and 'District'. A 'Register' button is at the bottom.

Below is a screenshot of the Patients tab and a patient's details, which can be seen when View is clicked on a specific row.

The screenshot shows two side-by-side windows. The left window displays a patient's profile for 'Mr. la la'. It includes a profile icon, the name 'Mr. la la', and details: 'Date Of Birth: 1997-02-12', 'Age: 20', and 'Phone Number: 07748593488'. Below this is a 'Health plan' section showing 'hp1 | Joined: 2016-11-13', 'Checkup: 0 out of 1', 'Repair: 0 out of 3', and 'Hygiene: 0 out of 2'. At the bottom are buttons for 'Unsubscribe', 'Book', 'Delete', 'Update', and 'Due payme...'. The right window shows the 'Patients' tab with a search form for 'First name', 'Last name', 'House Number', and 'Postcode'. Below the search form is a table of patients:

ID	Title	First ...	Last N...	Date ...	House...	Postc...	Telep...	Actions
1	Mr.	la	la	1997...	Flat C...	S3 7LG	0274...	View
2	Dr.	as	as	1993...	Apt. E...	S3 7LS	0864...	View

Here you can see the Addresses tab along with the details of a specific address, which, again, can be seen when View is clicked. The address details view allows the secretary to see a list of all patients at that address.

Address details

House Number: 1

Street Name: a

District: a

City: a

Postcode: aa

Patients in this address

ID	Title	First Name	Last Name	Date Of Bi...	Telephone	Actions
3	Mr	a	a	2017-1...	1	View

Addresses

House number:

Street name:

District: [Search](#)

City:

Postcode:

House Nu...	Street Name	District	City	Postcode	Actions
1	a	a	a	aa	View Pati...
a	a	a	a	aa	View Pati...
Apt. E42F	20 Crazy ...	Sheffield	Sheffield	S3 7LS	View Pati...
Flat C43F	80 Hoyle ...	Sheffield	Sheffield	S3 7LG	View Pati...

Doctor's view

We focused on simplicity when designing the doctor's view since this works best when using touchscreen devices. To achieve this, our system shows only the necessary details, with further information and actions only a tap away. On logging in, the doctor sees the current day's appointments with an option to change the day if needed.

The doctor can choose to view an appointment's details, patient's details and can cancel too.

Appointments - 26/11/2017

Start 09:00 End 11:00	Mr. Arthur Granacher Remedial Single Appointment	View Patient Info Cancel
Start 10:00 End 11:00	Ms. Nuraldeen Magid Checkup Single Appointment	View Patient Info Cancel
Start 12:00 End 13:00	Ms. Nuraldeen Magid Remedial Single Appointment	View Patient Info Cancel
Start 13:00 End 14:00	Mr. Arthur Granacher Cleaning Single Appointment	View Patient Info Cancel
Start 14:00 End 15:00	Ms. Nuraldeen Magid Remedial Single Appointment	View Patient Info Cancel

Viewing day: 26-Nov-2017 [...](#)

The screenshot shows a window titled "Dental Clinic System" with a patient's name "Mr. Arthur Gran..." at the top. The left sidebar contains a "Remedial" section with patient details (20 years old, 1997-02-12), a "Health plan" section with "Gold HP" and progress (0 out of 1 for Checkup, 0 out of 3 for Repair, 0 out of 2 for Hygiene), a "Treatments" section with checkboxes for Amalgam (checked), Composite, and Gold Crown, and a "Notes" section with the text "Amazing patient." and a "Save appointment" button. The main area is titled "Other Appointments" and lists three appointments for Mr. Arthur Granacher, each with start/end times and a "View Above" button.

On tapping View on an appointment, they are displayed with all the relevant information they might need in an appointment. The left hand side displays information for the current appointment along with a list of treatments that can be added to the appointment using the checkboxes. The option to add the treatments which have been done in the appointment is only available if the appointment is of remedial type. On the

other side, a list of the patient's previous appointments with this doctor is shown and on top is a small box that shows details for the selected appointment from the list below. This is to help the doctor review any previous notes that might have been left in the any of the patient's previous appointments.

As you can see from the mockup below, the layout of the final design is very similar.

The wireframe mockup shows a layout with a left sidebar and a main content area. The sidebar contains a "Patient name" field, a "Patient details (age, date of birth)" field, a "health plan details and usage if any" field, a "Checkboxes with all treatments" section, a "Notes:" section with a text area, and a "Save appointment" button. The main content area has a "Selected previous appointment details. (By default, last appointment)" box at the top and a "List of previous appointments with this doctor and buttons to allow them to be viewed above" box below it.

UML Diagrams

Use Case Diagram

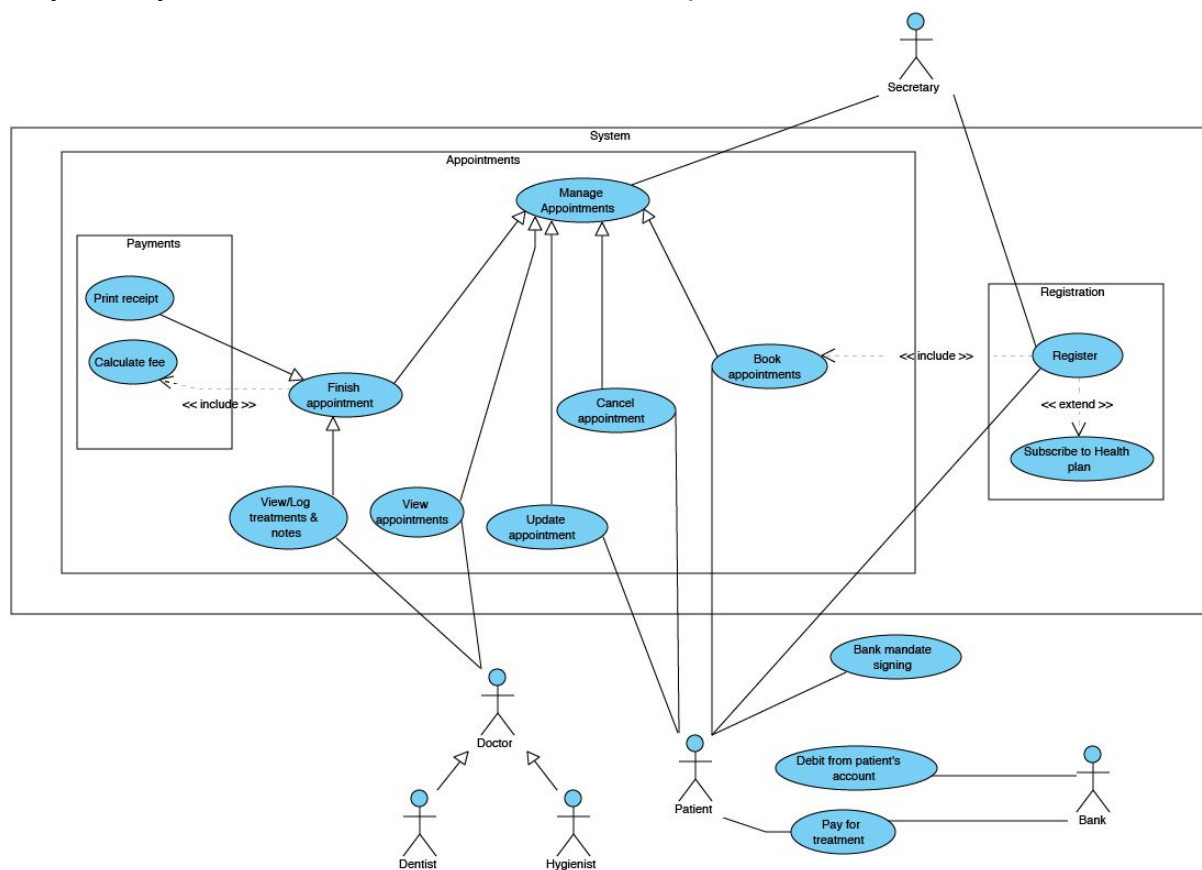
The figure below is our representation of a use case diagram, in which we tried to capture the relevant information from the scenario.

As you can see, we chose to generalize the dentist and hygienist as a doctor (actor hierarchy) because they carry out the same tasks. However the secretary and patients' tasks are separate to the doctors, so they are their own actors.

In addition, we decided to include finish appointment when dealing with calculate fee, as the appointment should be finished before it is to be paid. We also decided to include booking appointments when registering a new patient, as whenever a new patient is registered an appointment with both doctors should be booked.

Furthermore, we used a use case hierarchy for managing appointments where book, cancel, update, view and finish appointment are all similar tasks that fall under managing appointments.

Lastly we have chosen that subscription should extend registration as the patient may or may not choose to subscribe to a health plan.



Information Model Diagram

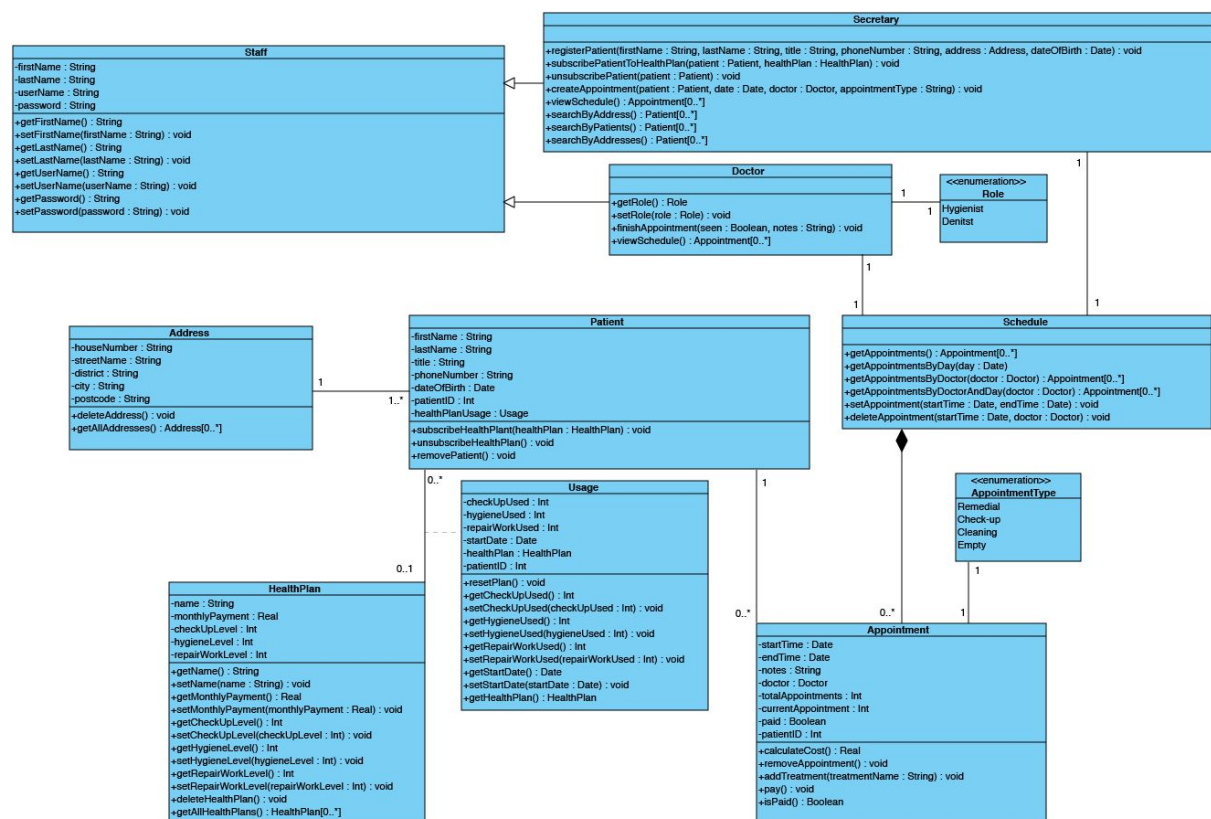
The figure below is our representation of an information model, in which we tried to capture the relevant information from the scenario.

Patients have their own class, with attributes of their details. It's methods allow a patient to subscribe, unsubscribe or reset health plan. Other classes associate with the patient class are address, health plan, usage and an appointment..

Doctors and the secretary of the clinic generalise from a staff class, where a Doctor has an attribute of role which is from a type enum (either doctor or hygienist). The secretary and doctors class contain several different methods but both of them contain the method to view schedule.

Schedule class is associated with the doctor and secretary class. This class gets all types of appointments, is associated with appointmenttype and appointment. Schedule doesn't contain any attributes, but contains methods that allow you to filter the appointment in numerous ways such as by doctor.

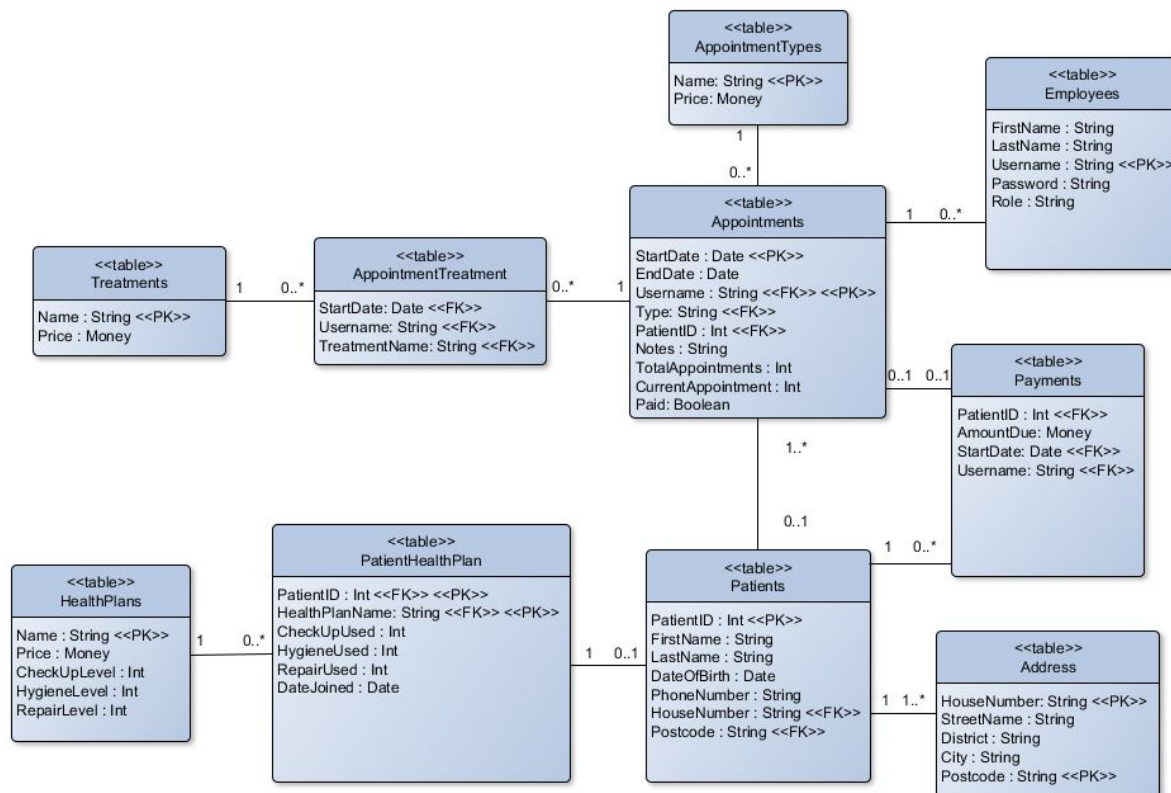
All classes have get methods, however our diagram only show classes where the methods are important to the class such as the schedule class.



Normalised Data Model Diagram

The figure below is our representation of a normalised data model, in which we tried to capture the relevant information from the scenario.

As you can see below we have decided to make two major tables, patient and appointments. These two tables are associated with each other. All other tables are linked with these two tables in some way (directly associated or indirectly associated). Evidently this can be seen by the amount of foreign keys in the appointments class. All tables have a primary key to uniquely identify its data, except for appointment treatment as all its attributes are foreign keys.



Query Processing

Registering a new patient and then showing that the new patient exists in the DB.

```
mysql> SELECT * FROM Patients;
```

PatientID	Title	FirstName	LastName	DateOfBirth	PhoneNumber	HouseNumber	Postcode
1	Mr.	la	la	1997-02-12	02748593488	Flat C43F	S3 7LG
2	Dr.	as	as	1993-06-13	08648148426	Apt. E42F	S3 7LS

2 rows in set (0.00 sec)

```
mysql> SELECT * FROM Patients;
```

PatientID	Title	FirstName	LastName	DateOfBirth	PhoneNumber	HouseNumber	Postcode
1	Mr.	la	la	1997-02-12	02748593488	Flat C43F	S3 7LG
2	Dr.	as	as	1993-06-13	08648148426	Apt. E42F	S3 7LS
3	Mr.	George	Smith	1990-01-01	12345678910	Flat C43G	S3 7LG

3 rows in set (0.00 sec)


Subscribing a patient to a healthcare plan and then showing that the patient is linked with the relevant plan in the DB, through a new record of relevant treatment-credits for the plan, for the current year

```
[mysql> SELECT * FROM PatientHealthPlan;
```

PatientID	HealthPlanName	CheckUpUsed	HygieneUsed	RepairUsed	DateJoined	PaymentsIssued
1	hp1	0	0	0	2016-11-13	0

1 row in set (0.01 sec)

Dental Clinic System



Mr George Smith

Date Of Birth: 1990-01-01 Age: 27 Phone Number: 12345678910

Health plan
No Health Plan
Repair: 0 out of 0

Subscribe to health plan

Choose:
hp2

OK Cancel

Subscribe Book Delete Update Due payments

Start 14:00 Remedial | Single Appointment Doctor: D the amazing
End 15:00
11/28/17 View Pay Delete

Start 13:00 Remedial | Appointment 1 of 4 Doctor: D the amazing
End 14:00
11/28/17 View Pay Delete

Start 11:00 Checkup | Single Appointment Doctor: D the amazing
End 12:00
11/28/17 View Pay Delete

Start 10:00 Checkup | Single Appointment Doctor: D the amazing
End 11:00
11/28/17 View Pay Delete

PatientHealthPlan

Limit to 1000 rows

1 SELECT * FROM team035.PatientHealthPlan;

Result Grid Filter Rows: Edit: Export/Import: Wrap Cell

	PatientID	HealthPlanName	CheckUpUsed	HygieneUsed	RepairUsed	DateJoined	PaymentsIssued
1	ho1		0	0	0	2016-11-13	0
3	ho2		0	0	0	2017-11-27	0
	NULL	NULL	NULL	NULL	NULL	NULL	NULL

Creating an appointment for a patient to see the dentist and showing that this appointment appears in the receptionist's week-to-view calendar for the dentist, and also appears in the dentist's appointments for that day

Register Patients Addresses Book Appointment

Appointment Booking

Start Day 27-Nov-2017

Start Time 14:00:00

End Day 27-Nov-2017

End Time 15:00:00

Doctor Dr. D the amazing

Appointment type Remedial

Book Appointment

Register Patients Addresses Book Appointment

Appointment Booking

☐ Course Treatment ☒ Single Appointment

Current appointment Number 0

Total number of appointments 0

Patient first name George

Patient house number Flat C43G

Patient Postcode S3 7LG

Book Appointment

Dentist
Hygienist

Dentist Appointments

Patient First Name: Patient Last Name:

Monday 27

Start 14:00
End 15:00
11/27/17

Mr George Smith
Remedial | Single Appointment
Doctor: D the amazing

Details Patient Info Cancel

Week from day: 27-Nov-2017 Refresh (View All)

Dental Clinic System
File

Appointments - 27/11/2017

Start 14:00
11/27/17

End 15:00
11/27/17

Mr George Smith
Remedial | Single Appointment
Doctor: D the amazing

View Patient Info Cancel

Viewing day: 27-Nov-2017

Attempting to create two appointments for a patient that are refused, either because the patient or the partner already have appointments at this time

1) *Doctor has another appointment at the same time*

Dental Clinic System
File Health Plans
Dentist Hygienist

Dentist Appointments

Patient First Name: Patient Last Name: Search

Tuesday 28

Start 10:00
End 11:00
11/28/17

Mr George Smith
Checkup | Single Appointment
Doctor: D the amazing

Details Patient Info Cancel

Start 11:00
End 12:00
11/28/17

Mr George Smith
Checkup | Single Appointment
Doctor: D the amazing

Details Patient Info Cancel

Start 13:00
End 14:00
11/28/17

Mr George Smith
Remedial | Appointment 1 of 4
Doctor: D the amazing

Details Patient Info Cancel

Week from day: Nov 27, 2017 Refresh (View All)

Register Patients Addresses Book Appointment

Appointment Booking

Start Day Nov 28, 2017

Start Time 11:00:00

End Day Nov 28, 2017

End Time 12:00:00

Doctor Dr. D the amazing

Appointment type Checkup

Course Treatment Single Appointment

Current appointment Number 0

Total number of appointments 0

Patient first name John

Patient house number G288

Book Appointment

Error fetching data from db
Clashing appointment exists.
OK

2) Patient has an appointment with the other doctor at the same time

The screenshot shows the 'Dental Clinic System' window with the 'Book Appointment' tab selected. On the left, the 'Dentist' tab is active, displaying a calendar for 'Tuesday 28' and 'Wednesday'. Three appointments for 'Mr George Smith' are listed for Tuesday 28: 10:00-11:00, 11:00-12:00, and 13:00-14:00. On the right, the 'Appointment Booking' form is visible, showing fields for Start Day (Nov 28, 2017), Start Time (11:00:00), End Day (Nov 28, 2017), End Time (12:00:00), Doctor (Dr. H the cleaner), and Appointment type (Checked). A modal error dialog is displayed in the center, stating 'Error fetching data from db' and 'Clashing appointment exists.' with an 'OK' button. The 'Book Appointment' button is at the bottom right.

Booking two days holiday for the hygienist and then showing that blank appointments fill the relevant two days on the hygienist's week-to-view calendar

The screenshot shows the 'Appointment Booking' form with the following fields: Start Day (27-Nov-2017), Start Time (00:00:00), End Day (28-Nov-2017), End Time (23:59:59), Doctor (Dr. H the cleaner), and Appointment type (Empty). The 'Book Appointment' button is at the bottom.

The screenshot shows the 'Appointments - 27/11/2017' view. It displays a single appointment slot for 'Empty Appointment' on 11/27/17, from 00:00 to 23:59, with Doctor: H the cleaner. The 'Viewing day' is set to 27-Nov-2017. Buttons for 'View', 'Patient Info', and 'Cancel' are present.

The screenshot shows the 'Appointments - 28/11/2017' view. It displays a single appointment slot for 'Empty Appointment' on 11/28/17, from 00:00 to 23:59, with Doctor: H the cleaner. The 'Viewing day' is set to 28-Nov-2017. Buttons for 'View', 'Patient Info', and 'Cancel' are present.

Recording two treatments given by the dentist to a patient and then showing that these have been added to the bill for the appointment

1) Doctor adding treatments

The screenshot shows the 'Dental Clinic System' window. On the left, under 'Mr George Smith', the appointment is for 'Remedial' treatment, dated '27 years old | 1990-01-01'. The 'Health plan' is 'hp1' with a progress bar showing 'Checkup: 0 out of 1', 'Repair: 0 out of 3', and 'Hygiene: 0 out of 2'. Below this, the 'Treatments' section has two checkboxes: 'Treatment2' (checked) and 'Treatment1' (checked). A 'Notes' text area is empty. A 'Save appointment' button is at the bottom left. On the right, the 'Other Appointments' section shows the current appointment details: 'Day: 2017-11-28', 'Appointment Type: Checkup', 'Treatments: None', and 'Notes (if any):'. Below this, a summary box shows the appointment for 'Mr George Smith' on '11/28/17' from '10:00' to '11:00', with a 'View Above' button.

2) Secretary sees treatments in payment breakdown when patient is paying

The screenshot shows the 'Dental Clinic System' window with the 'Payment breakdown' section. It lists the 'Appointment type: Remedial' and shows the costs for 'Treatment2 | £60.0' and 'Treatment1 | £50.0'. The 'Total price for appointment: £110.0' is displayed, followed by the 'Health Plan deduction: -£110.0'. The 'Final price: £0' is shown at the bottom. A 'Pay' button is located at the bottom right of the window.

Displaying the total cost of an appointment for a patient who is on a healthcare plan, showing the total cost of treatments and the amount owed by the patient

Dental Clinic System

Payment breakdown

Appointment type: Remedial

Total price for appointment: £110.0

Health Plan deduction: -£110.0

Final price: £0

Pay

Processing a payment by a patient who is on a healthcare plan, showing how their treatment-credits for that year are adjusted, and that they now owe nothing.

Dental Clinic System

Payment breakdown

Appointment type: Remedial

Total price for appointment: £110.0

Health Plan deduction: -£110.0

Final price: £0

Pay

Evaluation

On the whole, we believe we have performed very well as all requirements of the project have been fulfilled and even exceeded in many cases. Nonetheless, we felt there was room for improvement in our organisation, communication and efficiency.

Throughout the whole project we were well organised, every meeting was attended by us all with the exception of one or two meetings. However towards the end of the project we had discussed that the distribution of work could have been distributed more evenly, as some of us thought we had too much work, this leads us to believe our organisation could have been better. The failure here comes from the fact that certain aspects of the system were harder/easier than expected. Knowledge of these things comes with experience, so it's not too surprising that this happened.

What's good though is that we were all willing to help each other and discuss ways of improving our system so this eased the burden of some. Our frequent meetings, in addition to our group chat that we set up, helped in this regard.

Sometimes in-house deadlines weren't met by us all, but this is to be expected since workload varies throughout the semester.

On the whole we believe we were efficient as the software was finished on time and to a high standard.

Signed effort Deceleration

Team Member	Work Contributed	Effort
Andreas Constantino u	<ul style="list-style-type: none"> Contributed to diagrams Created schedule class Contributed to appointment class Contributed to model tests Contributed to report 	85%
Arthur Granacher	<ul style="list-style-type: none"> Contributed to diagrams Created appointment class Created appointment type Enum Created DBQueries class Create Role type Enum Created Doctors class Created Secretary class Created Staff class Created model tests Contributed to report 	100%
John Ayad	<ul style="list-style-type: none"> Main contributor to diagrams Creator of Interface package and its class (Only contributor) Creator of view package and its classes (Only contributor) Created and Initialised Database package (Only contributor) Created setupwizard package (Only contributor) Contributed health plan class Implemented design of all of the application (views and controllers) (only contributor) Contributed to the models Creator of utils package and its classes (Only contributor) Contributed to report 	115%
Nuraldeen Magid	<ul style="list-style-type: none"> Contributed to diagrams Contributed to model tests Created patients class Created address class Created usage class Created health plan class Contributed to report 	100%

Signature

Nuraldeen Magid

Arthur Granacher

John Ayad

Andreas Constantinou