C:\Users\Messom\Desktop\Monash.jpg

# FIT5032 Design Report

# Major Application Development

# High Distinction

# Tamrul General Practice Innovations

Nicolas Pallant : 28785959

Contents

[FIT5032 Design Report 1](#_Toc117380686)

[Major Application Development 1](#_Toc117380687)

[High Distinction 1](#_Toc117380688)

[Tamrul General Practice Innovations 1](#_Toc117380689)

[Web Application Title and Description 3](#_Toc117380690)

[Title 3](#_Toc117380691)

[Description 3](#_Toc117380692)

[User Stories and Use Case Diagram 3](#_Toc117380693)

[User Stories 3](#_Toc117380694)

[Use Case Diagram 4](#_Toc117380695)

[Block/Functional Diagram 5](#_Toc117380696)

[Your Selected Approach when Constructing the Application 5](#_Toc117380697)

[UML Class Diagram 6](#_Toc117380698)

[Data Dictionary 7](#_Toc117380699)

[Mockup Prototype and implementation with user Registration & Authentication 10](#_Toc117380700)

[Git 10](#_Toc117380701)

[Usability Design Review 11](#_Toc117380702)

[Development Methodology 11](#_Toc117380703)

[The code-and-fix model has been used for the development of this web application solution. The reason why this was used is because I find it much simpler to be able to try and code as much as I can, try it out, and then fix the issues that occur along the way. 11](#_Toc117380704)

[On top of this, I do not have the time, money, or resources to be able to plan out far in advance, get user data, collect testing data, so it was much more attainable for this style of development methodology. 11](#_Toc117380705)

[Versioning 12](#_Toc117380706)

[Git repository 12](#_Toc117380707)

[Commits 12](#_Toc117380708)

[Checklist of Site Functionality 13](#_Toc117380709)

## Web Application Title and Description

### Title

Tamrul General Practice Innovations

### Description

Online Booking System for Large General Practice Clinic. Customers can contact reception staff, who fill out a meeting between the client and the doctor. Reception staff have the highest level of security, doctors have the next step down, and then patients have little to nothing. Patients can login to their booking number and date and cancel/move the appointment. The website also allows patients to lookup location and opening times of the clinic.

## User Stories and Use Case Diagram

### User Stories

(All portraits generated using AI on <generated.photos>)

As an aging father, I want to be able to have a history of all of my past doctor’s certificates so that my and my children’s financial wellbeing aren’t on the line if my employer ever asks for a forgotten doctor’s appointment.

(<https://generated.photos/face/neutral-white-middle-aged-male-with-short-brown-hair-and-brown-eyes--5e6888dd6d3b380006f21a91>)

As a GP reception working, I want to be able to easily add new clients and doctors to our records without having to print off physical paper so that we don’t need to waste any additional paper and create additional paperwork.

(<https://generated.photos/face/joyful-white-middle-aged-female-with-short-brown-hair-and-brown-eyes--5e6886566d3b380006f18667>)

A person with dark hair

Description automatically generated with low confidenceAs a busy university student, I want to be able to receive calendar invites when I am booked into an appointment, as I easily forget exactly what date and time I have commitments.

(<https://generated.photos/face/joyful-asian-young-adult-female-with-medium-brown-hair-and-brown-eyes--5e6889316d3b380006f22d51>)

### Use Case Diagram

Diagram

Description automatically generated

## Block/Functional Diagram

Diagram, schematic

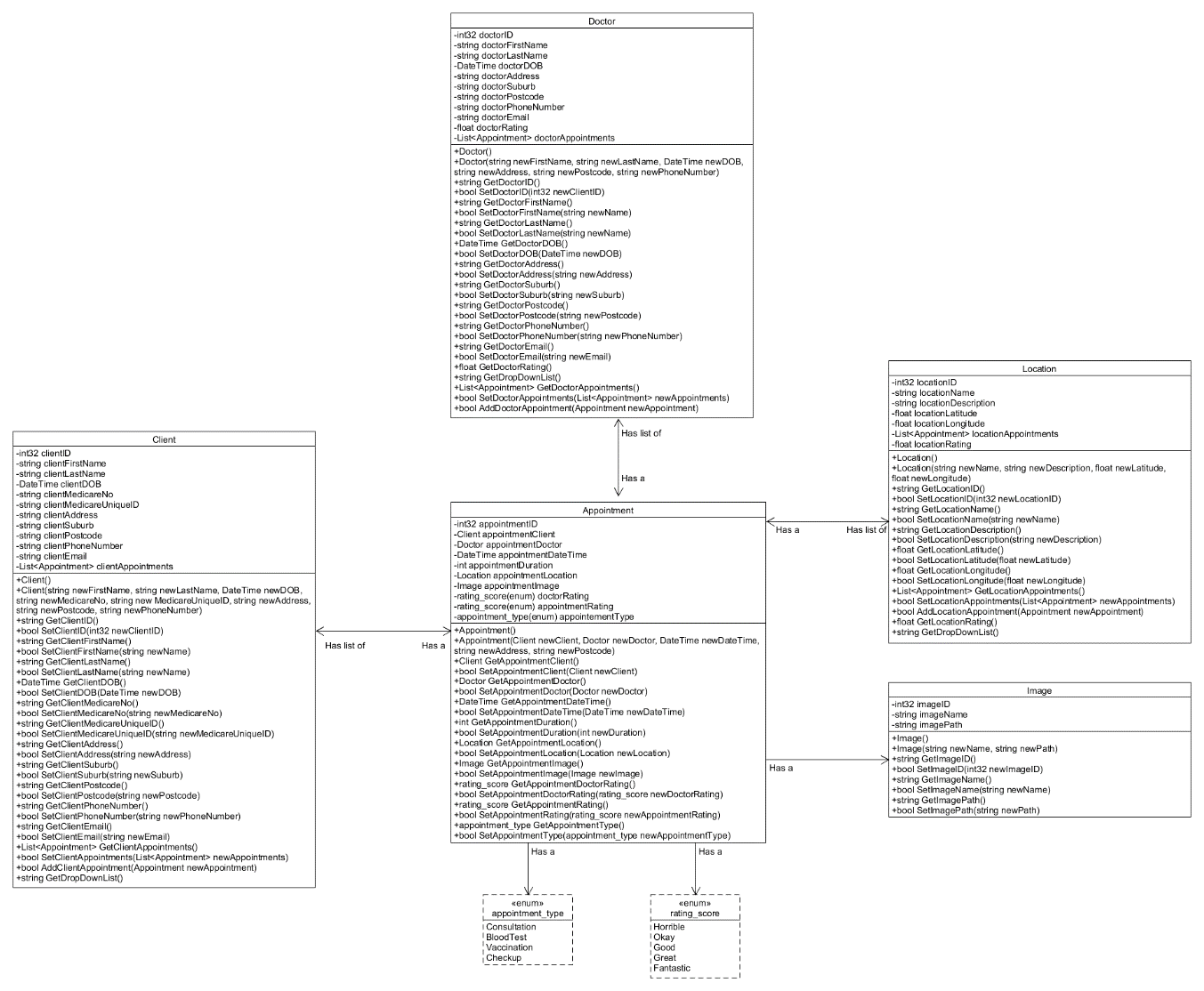
Description automatically generated

## Your Selected Approach when Constructing the Application

I decided upon using Code First for my selected approach with constructing the application as I feel that you have significantly more control over the application development in C#, and I am much more familiar with C# than the other languages like SQL.

On top of this, validation and validation messages are much easier to navigate and flexible using C# and the code first approach than Database First and Model First.

## UML Class Diagram

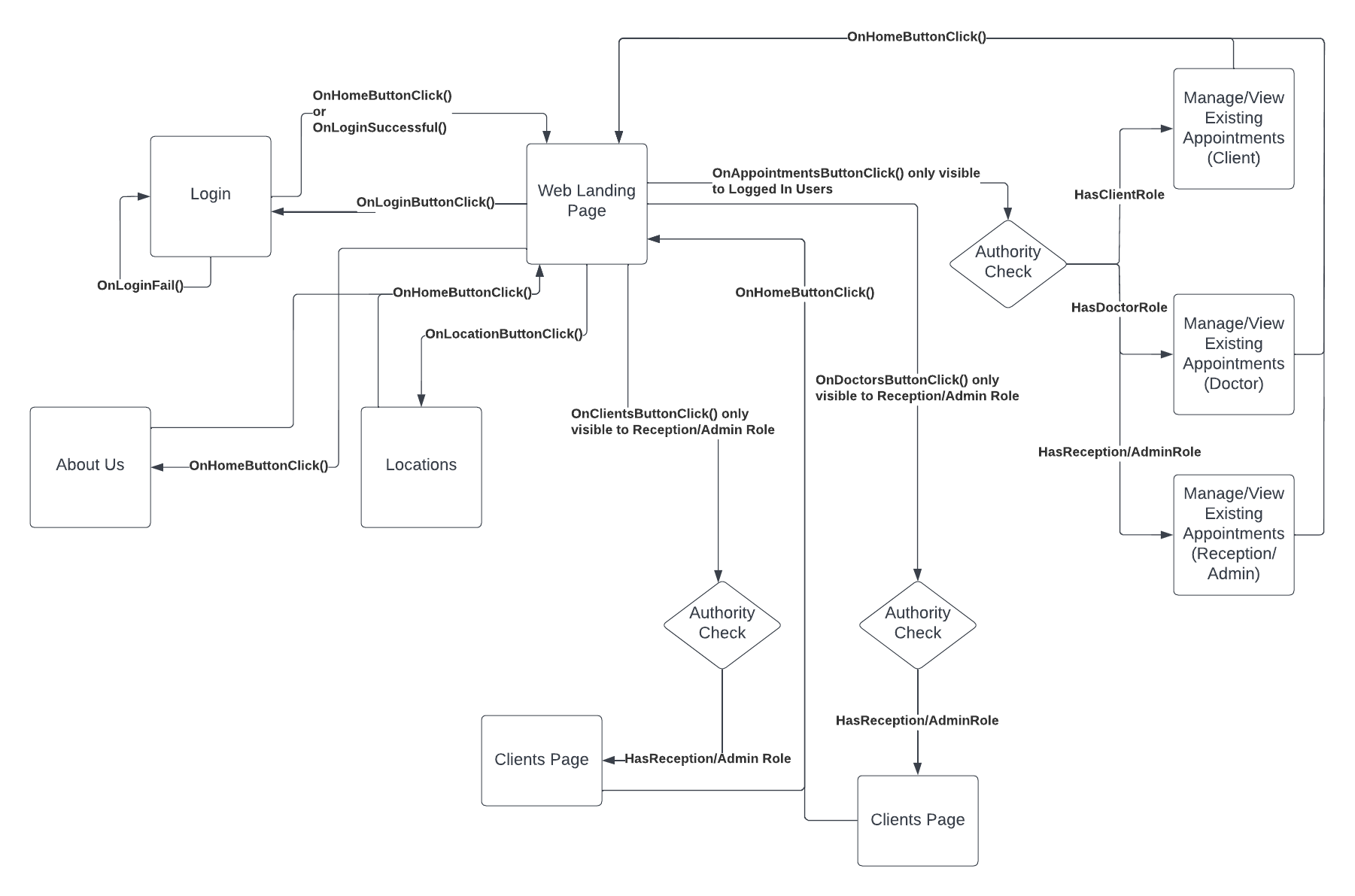


## Data Dictionary

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Field Name | Data Type | Data Format | Field Size | Decription | Example | Justifcation |
| First Name | String |  | 50 | Fist Name for Clients & Doctors | Nicolas | String used as names are arrays of characters |
| Last Name | String |  | 50 | Last Name for Clients & Doctors | Pallant | String used as names are arrays of characters |
| DOB | DateTime | DD/MM/YYYY |  | Date of Birth for Clients & Doctors | 25/01/1999 |  |
| MedicareNo | Int | NNNNNNNNN | 9 | Medicare Number for Clients | 111111111 |  |
| MedicareUniqueID | Int | N | 1 | Medicare Unique Identifier for Clients | 1 |  |
| Address | String |  | 100 | Address for Clients & Doctors | 21 Berringarra Road | String used as addresses are arrays of characrters, can have numbers and letters |
| Suburb | String |  | 50 | Suburb for Clients & Doctors | Officer | String used as suburbs are arrays of characters |
| Postcode | String | NNNN | 4 | Post Code for Clients & Doctors | 3809 | String used as Post codes can have a leading 0 |
| PhoneNumber | String | NNNNNNNNNN | 10 | Phone Number for Clients & Doctors | 0422592901 | String used as Phone numbers regularly have a leading 0 |
| Email | String |  |  | Email Address for Clients & Doctors (Used for login) | npal0002@student.monash.edu | String used as emails are made up from arrays of characters |
| DropDownList | String |  |  | Concatanation for FirstName + “ ” + LastName for Dropdown lists | Nicolas Pallant | Concatanation of strings, so a string made sense |
| Appointments | List<Appointment> |  |  | List of Appointments for Clients, Doctors, and Locations |  | Lists are dynamic, so size can be dynamically adjusted |
| Rating | Float |  |  | A rating score derived from the average of rating\_score enums | 2.56 | An average of an enum / amount of appointments, so float made sense as can be a decimal number |
| Appointment\_type | Enum |  |  | Enum for type of appointment | Consultation | Enum used as only certain specific options allowed |
| Rating\_score | Enum |  |  | Enum for rating of Doctors & Apppointments | Horrible | Enum used as only certain specific options allowed |
| Client | Client |  |  | Client attending the Appointment | N/A | N/A |
| Doctor | Doctor |  |  | Doctor attending the Appointment | N/A | N/A |
| DoctorRating | Rating\_score |  |  | Rating of the Doctor | Great | Rating\_score enum used as rating can only be from specific options |
| AppointmentRating | Rating\_score |  |  | Rating of the Appointment | Fantastic | Rating\_score enum used as rating can only be from specific options |
| AppointmentType | Appointment\_type |  |  | Type of the Appointment | Bloodtest | appointment\_type enum used as certain appointment types can only be from specific options |
| ClientString | String |  |  | Concatanation of Client’s FirstName + “ ” + LastName | Nicolas Pallant | Concatanation of strings, so a string made sense |
| DoctorString | String |  |  | Concatanation of Doctor’s FirstName + “ “ + LastName | Kristopher Pallant | Concatanation of strings, so a string made sense |
| LocationString | String |  |  | Concatanation of Location’s Name + “ ” + Description | Healthcare Services Clayton Practice | Concatanation of strings, so a string made sense |
| Location | Location |  |  | Location of the Appointment | N/A | N/A |
| DateAndTime | DateTime | DD/MM/YYYYTHH:MM |  | Date and Time of the Appoitnment | 23/10/2022 05:00PM | As appointments are made up of a date and also a specific time, DateTime made sense |
| Duration | Int |  |  | Duration in minutes of the Appointment | 20 | This clinic’s appointment duration can only be a number of minutes, not decimals, so integer made sense |
| Image | Image |  |  | Image uploaded to Appointment | N/A | N/A |
| Name | String |  | 100 | Name of Image/Location | Certificate | String used as names are arrays of characters |
| Description | String |  | 100 | Description of Location | Clayton Practice | String used as descriptions are arrays of characters |
| Latitude | Float | 0:###.######## |  | Location’s Latitude bearing | -35.2453 | Float used as latitude can consist of decimals |
| Longitude | Float | 0:###.######## |  | Location’s Longitude bearing | 146.3245 | Float used as longitude can consist of decimals |
| Path | String |  |  | Image’s file path on the server | /Uploads/d27da623add72367d | String used as file paths are combination of letters and numbers |

## Mockup Prototype and implementation with user Registration & Authentication

Text

Description automatically generated

## Git

<https://github.com/Ryukawastaken/FIT5032-Internet-Apps-Dev/tree/main/FinalAssignment>

## Usability Design Review

The usability of my current web application is very in-depth, such as what is currently included in my Client class. My client class currently features many constraints which don’t allow users to input invalid options, like entering a negative number for a phone number, or not including an @ symbol in an email field.

On top of this, my web application uses visibility to its advantage by always including the navigation menu on the top of the page to allow the user to know that they can always move around. This also falls under the usability principle of consistency, as this is a point of the website that is always in the exact same spot every time, giving the user a sense of familiarity and consistency.

Another usability principle my web application adheres to is the concept of feedback, as whenever the user hovers over one of the navigation bar buttons, they highlight themselves and also become a different colour, this is to indicate that they can be pressed, and that the user is doing the correct thing by hovering over them.

## Development Methodology

## The code-and-fix model has been used for the development of this web application solution. The reason why this was used is because I find it much simpler to be able to try and code as much as I can, try it out, and then fix the issues that occur along the way.

## On top of this, I do not have the time, money, or resources to be able to plan out far in advance, get user data, collect testing data, so it was much more attainable for this style of development methodology.

## Versioning

Git has been used for versioning of the project with commits made every time I completely certain functionality of the project. No branches were used. If a mistake was made and the project was unrecoverable, the repository was rolled back to a previous commit.

Specifically GitHub Desktop was used as it is a free option for a Git GUI. GitLFS has not been used as commits have been less than 100mb.

### Git repository

<https://github.com/Ryukawastaken/FIT5032-Internet-Apps-Dev/tree/main/FinalAssignment>

### Commits

Text

Description automatically generated

## Checklist of Site Functionality

|  |  |
| --- | --- |
| **1. (Layout Page)** |  |
| Good Design | ✓ |
| Stylesheet | ✓ |
| JavaScript | ✓ |
| Menu | ✓ |
|  |  |
| **2. (Home page)** |  |
| Design and content | ✓ |
| Banner Image | ✓ |
|  |  |
| **3. (User Log in)** |  |
| Web form and validation controls | ✓ |
| Formatted data entry display | ✓ |
| Overall page design | ✓ |
|  |  |
| **4. (Customised Views and Controllers)** |  |
| Customised Views | ✓ |
| Customised Controllers | ✓ |
| Other customisations | ✓ |
|  |  |
| **5. (Documentation)** |  |
| Code Comments | ✓ |
| Attribution of Source of any code used | ✓ |
|  |  |
| **6 Business Requirements** |  |
| **BR(A1): for P** | ✓ |
| **BR(A2): for P** | ✓ |
| **BR(B1): for C to C+** | ✓ |
| **BR(B2): for C to C+** | ✓ |
| **BR(C1): for C+ to C++** | ✓ |
| **BR(C2): for C+ to C++** | ✓ |
| **BR(C3): for C+ to C++** | ✓ |
| **BR(C4): for C+ to C++** | ✓ |
| **BR(D1): for D to D++** | ✓ |
| **BR(D2): for D to D++** | ✓ |
| **BR(D3): for D to D++** | ✓ |
| **BR(D4): for D to D++** | ✓ |
| **BR(E1): for HD to HD+** | ✓ |
| **BR(E2): for HD to HD+** | ✓ |
| **BR(F1): for HD+ to HD++** |  |
|  |  |
| **Audit** |  |
| No breaking of copyright | ✓ |