

Synoptic Assignment

BY Ashley Carruthers



May 15, 2019

**Task One -**

The piece of software that we would make is an application that would allow for patients who take certain medication to submit any side effects they have experienced while taking the medication and allow for doctors to view and monitor these side effects.

As well as making it easier for the patients to report side effects it would also allow for medical staff to use the time, they would have spent gathering the information to treating patients or other important or critical tasks.

There are three main courses of action that could be taken, and these include;

-Do nothing. We do not go through with the development of the software and due to this there is no risk to losing money or resources but also prevents any future benefits that may come from having such a piece of software in place.

-Develop it ourselves. This would mean we would have to licence the software, buy hardware, hire programmers, designers, network technicians etc to work on the development of the software on each platform (Windows, MacOS, Android & IOS).

This option has the greatest risks, but it gives us the most control and flexibility over the development of the software as we would not be dependent on an external company for the development of the software. This in turn would allow for a much more polished piece of software as we will have full control over the design, development, testing and maintain of the software.

This would allow us to prevent most of the risks involved with development and have full control over fixing any that do occur and the actions that will be taken to get around them.

-Go through with development but hire another company to do the software development aspect. While this carries less risk then the option above the cost may be significantly higher if we go with a well-respected and trustable company. We would also not have as much freedom over the design of the software itself as the company who we hire will have to agree to any design we put forward.

On top of this, unexpected delays may occur due to problems on the companies’ side which we would have no control over preventing or fixing, apart from picking an experienced company. This would lead to an increase in cost and a longer time frame for the development or might cut down on the time we would have for the testing of the application reducing polish of the end product.

Due to the benefits this would bring not only to the patients but also the medical staff the best option would be to go through with the development ourselves. This is the better option than hiring another company because despite the higher risk, we would have more control over minimizing and preventing these risks and would have greater control over what action we would take if something were to go wrong.

**Objectives**

-Have a functional application that is that is accessible for most patients to record medication side effects for the PDUA (patient Drug Care Accusation)

-Have the application securely save recorded data to an external central server through encrypted connection

-Reduce the amount of staff needed for recording drug side effects from patients

**Success Criteria**

-Must be completed in the time frame that was agreed on with the sponsor and stakeholders

-Must be within budget

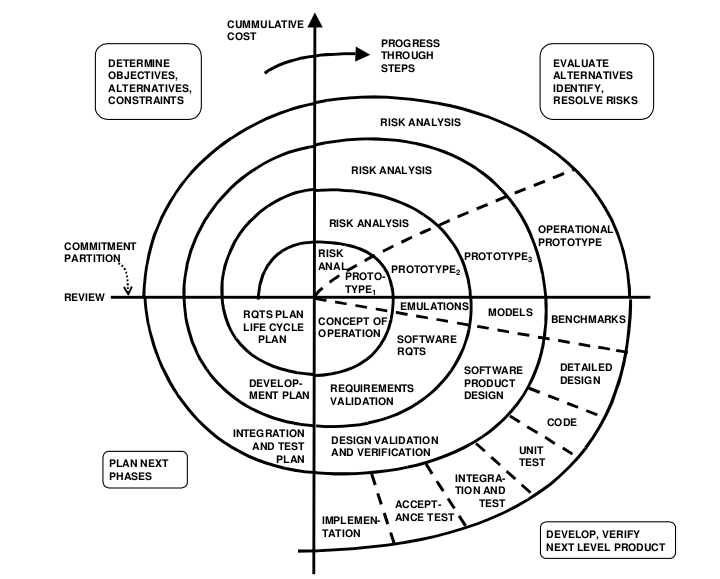
-At the end there must be a functional and secure application for user in medical faculties for patients to use to record any side-effects of their medication.

-Software must be easy to use

-Software must be easy to maintain and update.

-Software must securely and safely store and load data from a remote server.

**Software Development**

[](https://xbsoftware.com/blog/software-development-life-cycle-spiral-model/)The software development life cycle that we will use will be the Spiral Model

(Source of Picture - <https://xbsoftware.com/blog/software-development-life-cycle-spiral-model/>)

This is because the spiral model has a focus on risk awareness and management which is important for a piece of software that is used in the medical industry as any problems or harm caused due to improper function of the application can lead to greater harm to a person or a lawsuit. As well as due to the software handling sensitive personal information.

This also means that depending on the level of risk more effort will be applied to it. For example, the function of input validation has a high risk if it does not function correctly so there will be a large amount of effort put into the development and testing of this aspect of the software as it is critical that it functions as intended.

Since the application is being made for Windows, MacOS, Android and IOS we will need to licence specific software to allow for the development on each platform. These include; Visual Studios for windows, XCode for MacOS and IOS and Android Studio for Android.

The programming language we will use is C# due to it being a powerful language for applications and can also be used for development on all four platforms. This means we will save money on hiring programmers as we only need C# developers. This is also due to C# being able to run on any device without any software being installed.

On top of this C# is also very well documented mean that any issues we run into during development should be easy and quick to find a solution for saving time #.

This will also negate the risks and problems in using a different language for different platforms such as having to develop updates for each platform individually rather than for all at once which would greatly increase the time needed for updates and any fixes for issues found later on. It also allows for the developers working on each platform to communicate better as they are all using the same language.

For development for the application on Android, Android Studio will be used which is an IDE and includes an emulator for Android phones ranging from very old to modern devices with the ability of emulating almost any Android version be it old or new. This will decrease the number of physical devices needed to test the application on older versions of the OS but some will still be required to ensure it works on real devices.

Once the application has finished development work on it won’t stop right away as more bugs might be found during its user or something might need changed to make it easier to use.

**Testing**

Testing will be done throughout development and at key points throughout the software life cycle to ensure proper functionality and to detect any bugs or issues that need to be fixed.

(Test plan will be included once it’s made)

-For the testing on Android the Android studio IDE allows for the emulation of many different Android devices ranging from old to modern day smart phones as well as the majority of Android versions. This would allow for quick on the fly testing on versions of Android which we might not be able to get devices which are running them. This also means that we would only need a few different Android devices for testing on as emulation may not show certain issues that may occur on real devices.

Testing for Android also doesn’t require us to have publicly published the application.

-For testing on Windows, we will need a few different PCs with different versions of windows such as Windows XP, Windows Vista, Windows 7 as a large portion of computers in the medical sector still run older versions of Windows. This is so we can make sure that the software will run as intended and be stable on these versions of Windows.

-To test on IOS and MacOS we would need to purchase or rent a Mac and some iPhones to test on as we can’t test these versions on other devices as they wouldn’t be supported and wouldn’t accurately represent the environment that the software would run on, so problems may not show up.

To ensure that it works on a network we will also need to make a small test network with a central server and multiple end devices connected to ensure that the programme works under use from multiple devices. Darning this test, we can also ensure that the data is being transferred securely and isn’t vulnerable to attacks or theft

The majority of the testing will be white box testing with only people who have worked on the software doing the testing up until there is a piece of functioning software at which point there will be some black box testing with people from the medical sector who have never seen the software before.

**Risk management**

|  |  |  |
| --- | --- | --- |
| **Risk** | **Severity** | **Mitigation** |
| Losing data due to data corruption | Low | User brand new devices and storage devices to reduce the chance of them failing as well as backing up the files onto an external device, drive or server |
| Damage to hardware | Medium-low | Ensure proper use of the hardware and not allow liquids near the devices that development is being done on to mitigate the risk of water damage |
| Risk of hand / wrist injury | low | Ensure correct heights of desks and chairs and make sure that any mice and keyboards are ergonomic to reduce strain on wrists and arms |
| Risk of data theft | Medium-High | Ensure only certified people have access to the project’s files and Documents and if possible password protect them or have them encrypted. |
| Data loss due to power outage | Medium high | Have PDUs as backup power supplies in case the mains power goes out. This won’t keep devices online for a long period of time but will allow enough time for data to be saved and for the devices to be shutdown correctly |

**Costs**

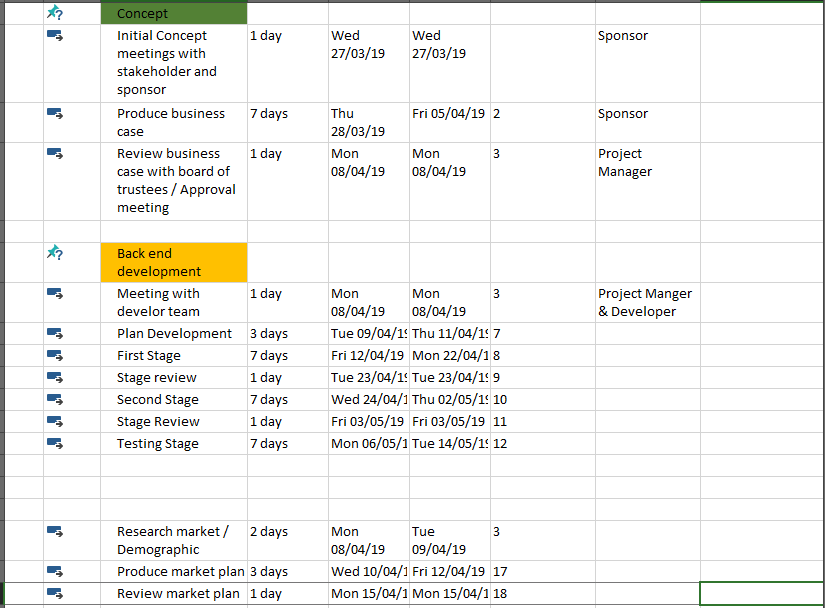
The main costs will come from the staff and the likening of the required software needed for the development on multiple platforms and operation systems

|  |  |
| --- | --- |
| Apple MacBook pro for development for macOS and IOS | £1249.00 payment – From Apples Website([https://www.apple.com/uk/shop/buy-mac/macbook-pro/13-inch-space-grey-2.3ghz-dual-core-128gb#](https://www.apple.com/uk/shop/buy-mac/macbook-pro/13-inch-space-grey-2.3ghz-dual-core-128gb)) |
| Visual Studio Professional | £35 per month (<https://visualstudio.microsoft.com/vs/pricing/>) |
| Samsung Galaxy S9 | From £499 (<https://www.samsung.com/uk/smartphones/galaxy-s9/shop/>) |
| Software Tester | £16 per hour (<https://www.payscale.com/research/US/Job=Software_Tester/Salary>) |
| Software Developer | £13.96 per hour (https://www.payscale.com/research/UK/Job=Software\_Developer/Salary) |
| Desktop Windows PC | £879 (<https://www.google.com/aclk?sa=l&ai=DChcSEwjatffDwu3hAhWjCtMKHbZWBHYYABALGgJ3Yg&sig=AOD64_0pEFy7Pulz0l-SDurmuLod5zaRmw&ctype=5&q=&ved=0ahUKEwjMvPLDwu3hAhVUSxUIHdKkDLUQ2CkImwQ&adurl=>) |

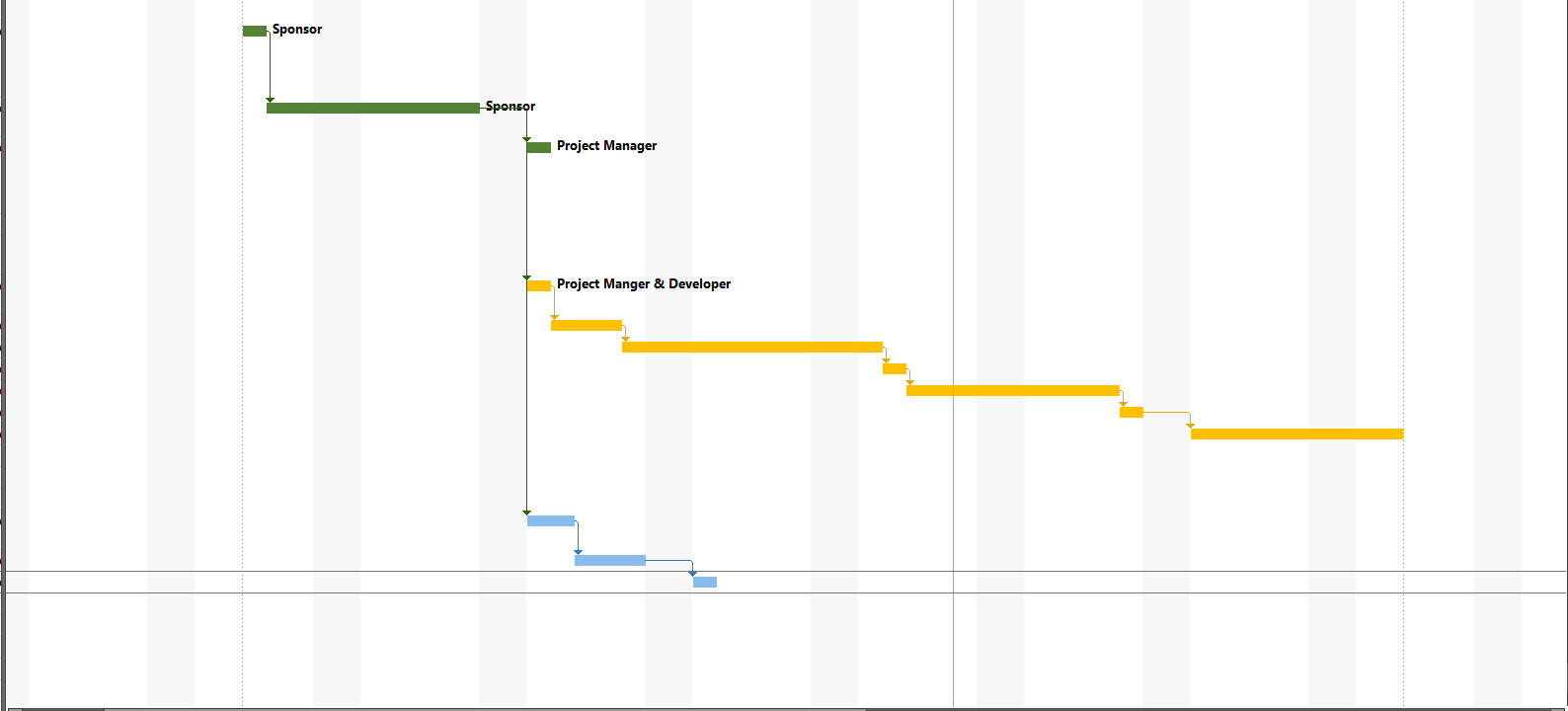
The majority of the large costs come from the hardware needed for the development, mainly the Mac and the windows PC as they need to be able to handle development on without slowing down or crashing as this would result in lost time and money.

To make sure that the project is still within budget there will be meetings with stakeholders and sponsors once every week to discuss any changes that may be needed to keep the project in budget or any action that needs to be taken to ensure it stays within budget.

Over all the project is expected to cost around £50,000 to produce the software. The majority of this cost will go to the hiring of the developers and testers.



**Legal, Ethical and Regulations**

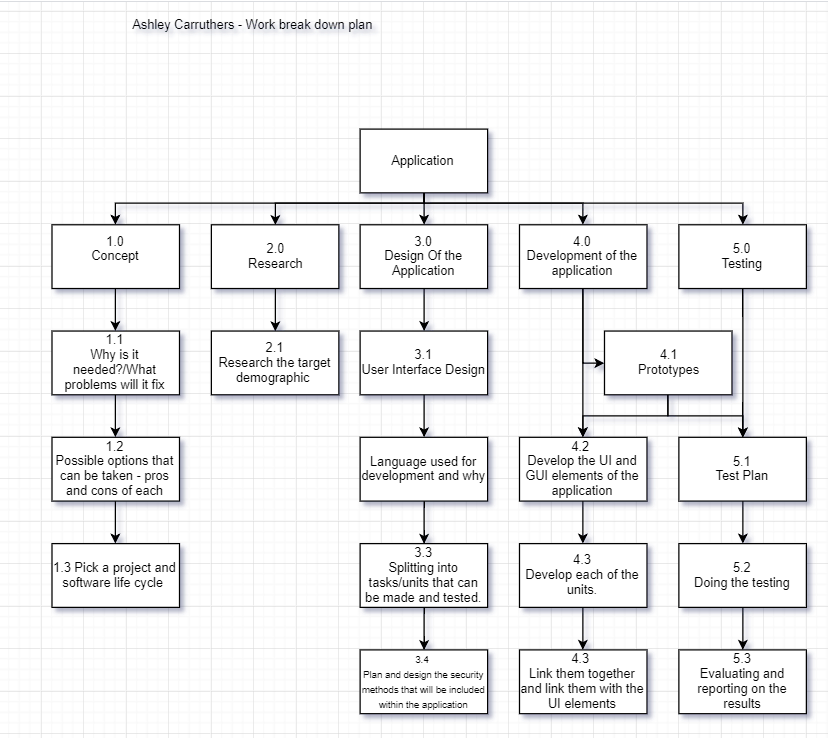


As the application will handle peoples personal and sensitive information, we will have to comply with the data misuse and data protect act to ensure the privacy of any data provided. To-do this we will ensure that the server that the data is stored on is protected from any malicious attacks such as DDOS (Distributed Denial of Service) attacks by making sure the data is encrypted at all times.

On the finished application itself before a user submits their data they will have to accept terms and conditions which will also include information about where and why their data is being stored and what it will be used for.

**Time**

The development of the software has been broken down into key parts;



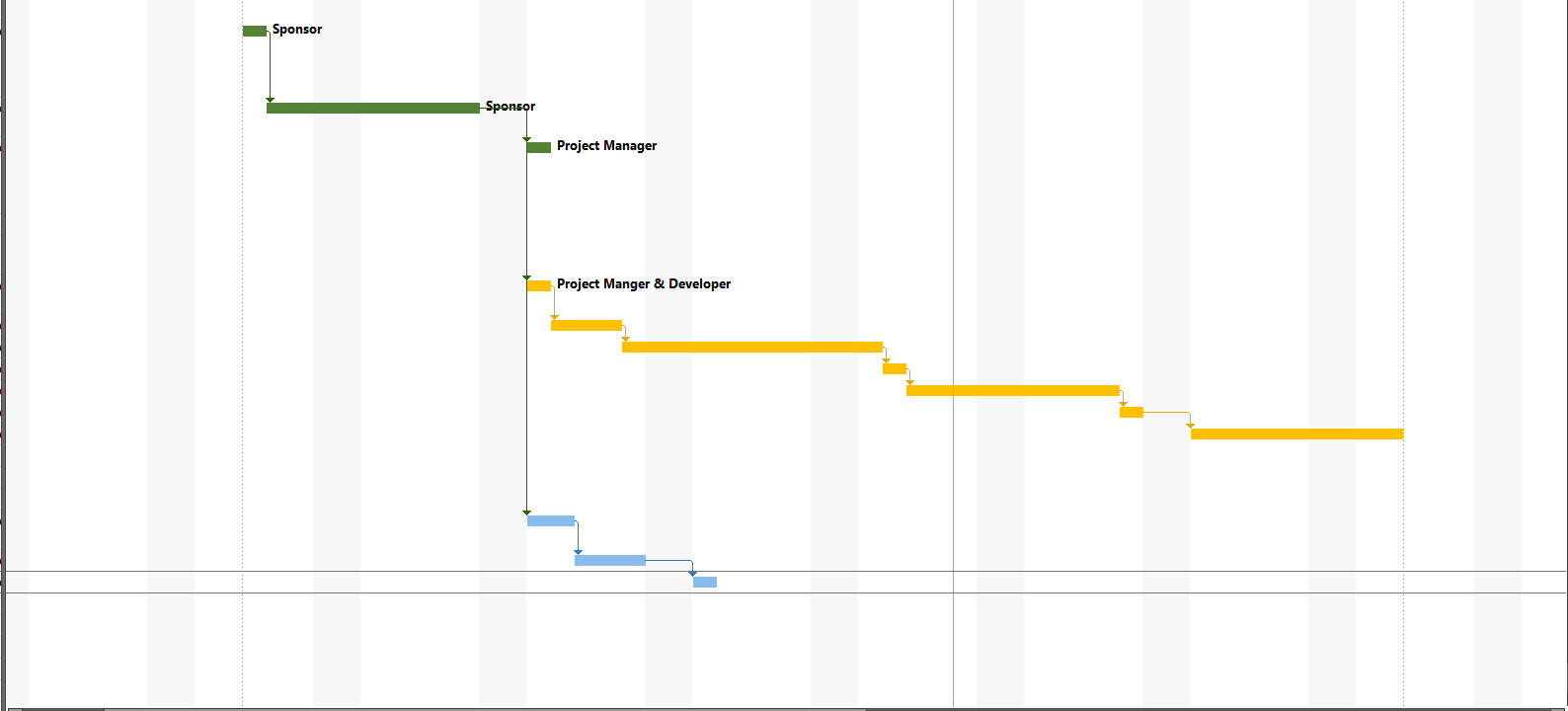
This can be applied to the development of the software on all platforms.

The development of the software for each platform can happen at the same time to decrease the overall time frame but this might increase costs are more people will be needed to work on each platform efficiently.

Another reason why I have decided to have a person working on each platform independently is it will reduce the overall workload on one person which should allow them to work to a high quality and helps to ensure a bug free and polished piece of software.

At the end of each task there will be a review to ensure that the task was completed correctly and meets quality standards and to figure out any action that need to take place to fix any issues

To make sure that the project is completed on time and is on target to be completed on time there will be meetings with stakeholders and the sponsor on the current progress and status of each part of the project and if they are predicted to be finished on time and if they aren’t what could be done / what needs to be done to get them back on target for completion on time.



**Social Media**

As the application will be used mainly by patients and medical staff the best place to advertise it would be on the inside of hospitals or other medical facilities as a lot of the people who go to these places will take regular medication which this application is designed to help.

The target demographic would be anyone over the age of 18 as these are more likely to have long term prescriptions to medication.

Advertising the application on social media wouldn’t be very effective as a lot of the elderly people who would use it would not have social media so they wouldn’t see it. It would be more effective to advertise on TV or radio.

**Task Two**

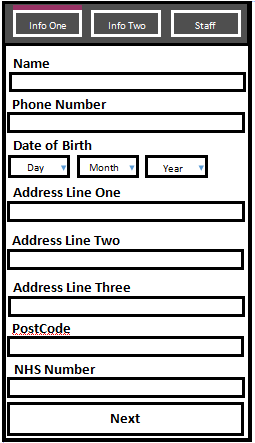
**Design Spec**

We need the programme to be use able by a wide range of people including those who have problems with their sight. To ensure this, in the final version we could have an option that upon first click on a UI element it says what it is out loud then another click would activate or trigger that UI element(s).

Another feature that could include in the final version to aids people is speech recognition for inputs to make it easier for those with motor disorders to input data on their own.

**UI**

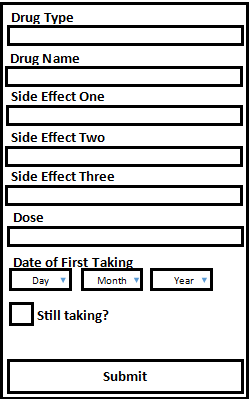
Option One:

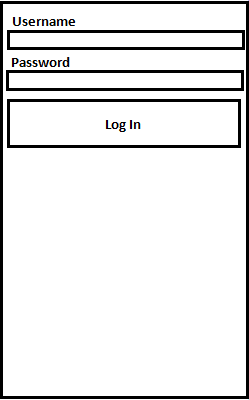
****The top bar of the form is always shown on the form as it is used for navigation and to also show which page the user is currently viewing, indicated by the bright ping / purple bar above the buttons.

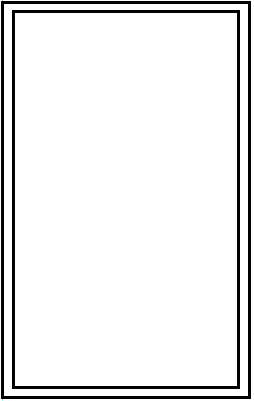
The space below this is where each page of inputs or data is shown. Due to this, it allows for each input box and the text that labels them to be larger and easier to read.

The first one which is shown has the inputs for personal information such as their name, date of birth and address.

The layout of each page is near exactly the same with there being inputs in most of the page with a ‘Next’ or ‘Submit’ button at the bottom.

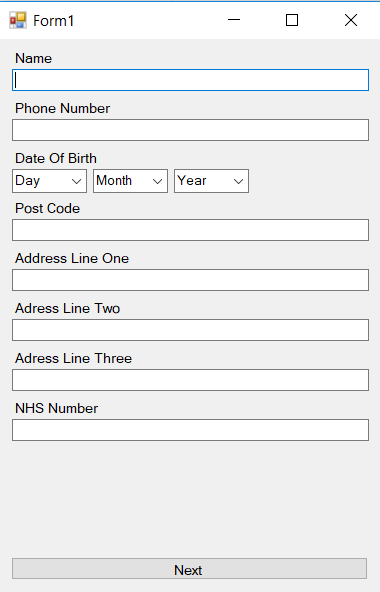
This page is visible once the inputs on the previous page have been entered and validated

This page will be visible to anyone but only staff will have the required usernames and passwords to login to the application and view a list of all data that has been entered.

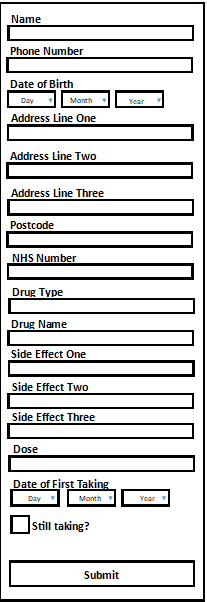
This page will only be visible to staff members as a user must log in to view it. All previously entered data is shown on the page and is split into blocks for each patient.

Each data field will be on a new line with name of the data being in **bold**

Option Two;

This layout is similar to design one, but it has a different font and lacks a navigation bar at the top. This one will not be used because the lack of a navigation bar and the different type of font would make the application more confusing to some people and increase the chance of them inputting incorrect data which could result is wasting some medical staffs time. And the feedback that I got from the UI showed that option one was the best one for functionality and looks.

Option Three

This UI design uses the same UI elements of design one but has it all visible at once rather than in sections. This design also lacks the feature that would allow staff to see any submitted data directly from the application itself with a log in.

The font that will be used is Century Gothic as it’s a simple font that has clearly formed and different characters which is important.

For the UI layout I have decided to go with option one as it will be split into a total of 5 elements of which 3 will be visible to the patients who will use it and the rest only viable to the staff. This is the better choice rather than the ones that have all UI elements shown at once because it reduces the number of items on the screen at any point in time which will reduce the chance of confusion when using the application as well as reduce the chance of incorrect data being inputted and then submitted.

The main one will hold the main body of the from and the top bar with navigational buttons of which upon launch 2 are active, one that shows the first inputs and the other which takes you to the staff login. The Middle button which is for the 2nd set of inputs only becomes active upon the first set of inputs being validated.

The main colours used will be black and white as they are the two colours with the most contrast so it will be significantly easier for most visually impaired people to distinguish between elements on the UI.

On the top bar there will be a brightly coloured bar above the button associated with the currently displayed window to indicate which window a user is currently viewing. This will make the application easier to use and reduce the chance of someone getting confused while using it and thus inputting incorrect data.

After the user has filled out each input on a selected page and clicked the next or submit button the data is validated first. If any of the inputs are invalid or blank a message box will pop up stating, why an invalid input is invalid as well as having that selected input box turning red to clearly indicated which of the inputs has an invalid value.

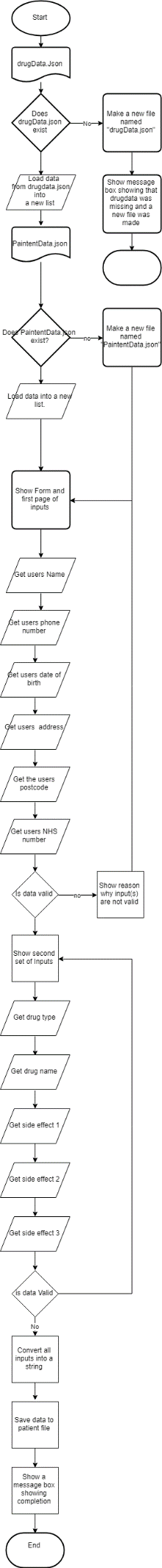
I have decided to have the data related to the drugs also stored on a file. This is because it means that the data is as secure as the user’s personal data which will prevent people from being able to change the data or delete it. It also means that if more drugs need to be added to the software, they just have to be added to the file rather than a completely new version / build of the software which would take time to release.

Overall the option to go with UI design one is to ensure that the data inputted is as accurate as possible and any errors are not caused by poor UI design as well as to ensure that it is easy to use for the majority of people including the staff.

Inputs

The majority of the inputs will be text boxes, including the ones that take numbers, but these will automatically detect if a non-numerical character has been inputted and show an error to the user. This allows for a lot of formatting options when we come to saving the data as a lot of it will be in string format.

The only inputs that are not directly text boxes are dates and the selection of side effects, drug type and drug name. These are drop downs. Although the side effect ones do take text as a direct input as a patient side effect may not be listed as one of the options.



**Other Documents**

**Test Plan**

|  |  |  |  |
| --- | --- | --- | --- |
| **Test** | **Expected Outcome** | **Actual Outcome** | **Comments** |
| File that stores drug data is not there | Should show an error box informing that the file is missing and then close out of the application | Shows error box which says that the file is missing then closes the application |  |
| Having random/unwanted chars in either the drug data file or the patient data file when loading | Should show a error message informing the user of the error then terminate the application | A message box shows up showing why the programme failed to launch / load data and the programme then closes. |  |
| Leaving any of the required inputs empty. | Should show a message box informing the user that the input is not valid and the input box with the invalid input will turn red while the message box is shown then turn back to white upon it being closed. | A message box shows up saying why a certain input is not valid as well as the input which is invalid turning red to indicate which box has the invalid input then turning back to white when the message box is closed |  |
| Inputting non numerical characters into inputs that only take numbers | Should refuse the input and inform the user that only numbers are accepted and clear the input box | Message box shows up saying that only numbers are accepted and the program does not continue to the next part – Box which had the letter in also turns red while the message is shown. |  |
| Check that only the correct usernames and passwords allow the data to be viewed inside the application | Upon incorrect username or password being entered a error message will appear informing the user that the password or username entered isn’t valid and the data is not shown to them | When invalid username is inputted it does not allow the login in and a message box is shown which shows that the username is incorrect – same for the password |  |
| The data is shown and formatted correctly in the application | The data should be formatted into blocks for each patient and all visible on the screen with a scroll bar to scroll further down if more data is present than what can fit on the screen |  |  |
| Check that data is being stored correctly | Upon clicking submit the data should be turned to text and saved to the “PaintentData.json” file under Resources inside the applications root folder – Should also be re-readable by the application upon next launch |  |  |
| Drugs are sorted/filtered correctly | Upon selecting a type of drug only drugs of that type should be shown under the drugName drop down |  |  |

**Technical Specification For Task Three -**

Minimum Hardware Requirements

-Intel Pentium or better

-2GB DDR2 or better

-At least 250MB of available disk space

-Windows XP or newer

Minimum Server Hardware Requirements – (Listed on Microsoft’s Website)

-Single 1.4ghz, 64bit processor

- 512GB of ram

-32 GB of free disk space

-Standard Ethernet network connection

Fonts – Needs to be a font that is easy to read for everyone – Needs to be large enough to read without straining people’s eyes – Needs to either be free or licenced

**UI**

**Colours**

-Top bar will be a darker grey back colour

-

Middle parts are 300 x 490

Side window colour = 41 39 40

Application its self is 300 wide by 550 long

30 between the above input and next label

25 between label and input

-For data input it will be split into three parts;

1- the first part will allow for the user to input personal information such as their name, DOB, address etc…

2- This will be for the information about the drug they have been taken and the side-effects they have had

3- The last part will mainly be for staff as it will include a login for staff members – Upon login all part patient data will be available to see for the staff with the options to remove or manually add data (last part is subject to change)

-At the end of each section there will be a next button, apart from on the last which will have a submit button, as well as a back button, apart from on the first section, which will allow the user to go back and change any data they had inputted before submission.

Designs

One –

Two –

Three –

**Classes**

**Data Storage**

-Static class as we only want one instance of it as it will write to files and if two instances of the class try to edit or read the same file it will cause an error as only one process can use a file at a time.

-Public to allow it to be accessed and its methods to be called from other classes and scripts throughout the application.

-This will have all the methods and functions related to the loading and saving of data to and from the file system included in the application for both the data of the drugs and patients.

-All methods that handle files with have error handling so if writing or loading were to fail it would not crash the application but would show a error message showing why the error happened and if possible how to fix it, for example if the data file is missing (if its missing it will automatically make one anyway)

**Drug**

-Public class as instances of it will be created from other classes through out the programme.

-This class is used to store the data of each drug and it contains the name of the drug and the class of the drug (Type Enum - DrugClass from Enum.cs)

-This is also the class which is serialized to text to be saved to the drugdata.json file

**Patient**

-This is like the drug class except this time it is used for the patient data rather than drug data

-It stores all the data for a patient as an object – Name, phone number , DOB etc

- Also includes a struct called ‘DateOfBirth’ which is used to store a users date of birth – It has 3 values, day, month and year.

**Enums**

**Data**

**Drugs**

-All the data on the drugs that can be picked will be stored in a JSON file called “drugdata.json”

-For saving the drug that a user selected it will only save the drug ID or NAME – this can then be used to get the drug that was picked from the dataset stored in the “drugdata.json” file –

-Each drug will have a ‘name’ value and a ‘type’ value which will be stored – The side-effects will be stored on the patient file once they have been selected

**Patient**

-All data on past patients will be stored in the “patientdata.json” file which they can be loaded or saved from

-For each patient all data they inputted will be saved; Name, DOB, NHS number, phone number, address, date of first taking, drug taken, side effects, and if it has been continued or not. As well as the data that the user had submitted the data.

-This data will only be loaded upon a ‘staff’ member logging into the application

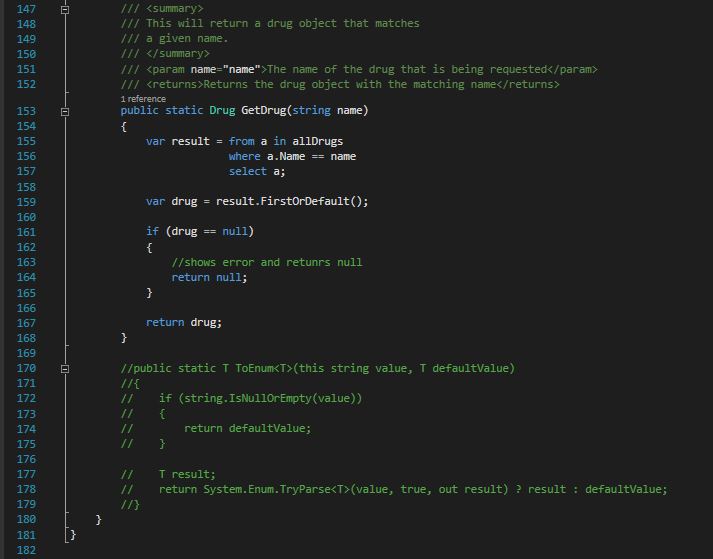
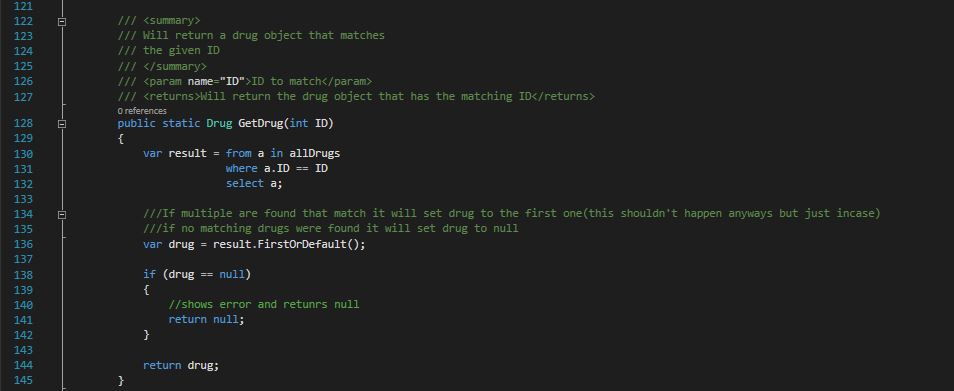
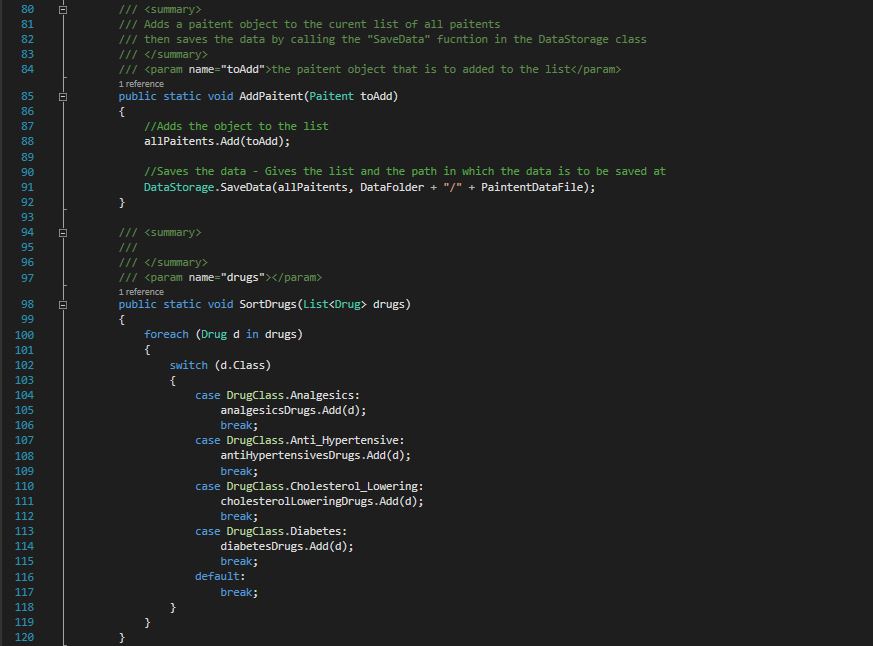
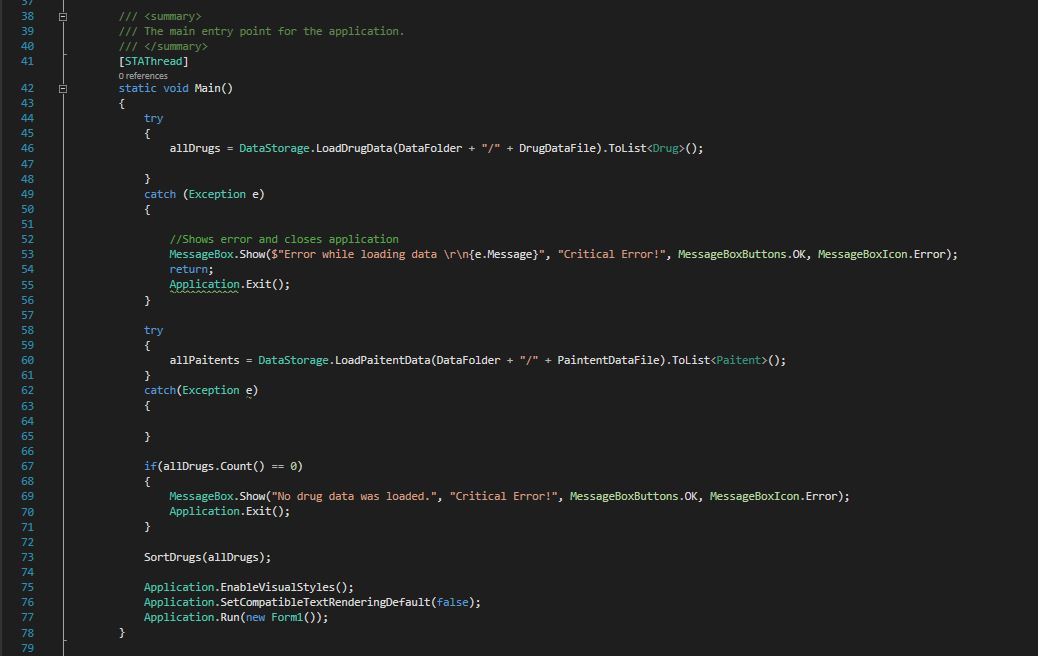
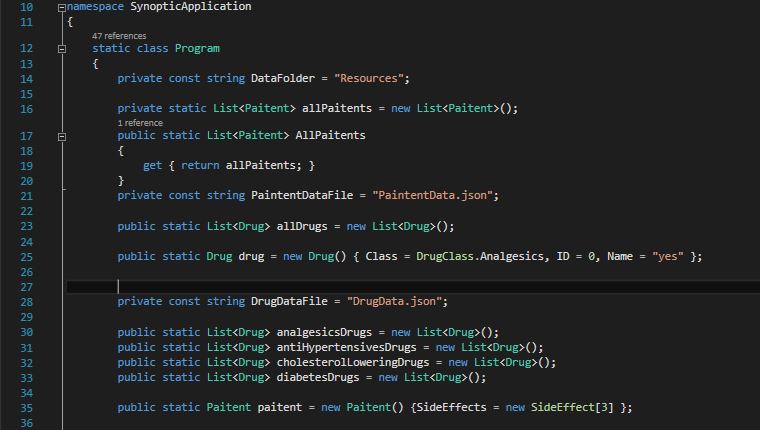
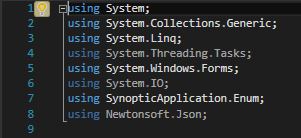
**Other**

-for loading side-effects from the enum class – Replace ‘-‘ in the name to spaces so they show up correctly in the drop down when they are being selected by the use.

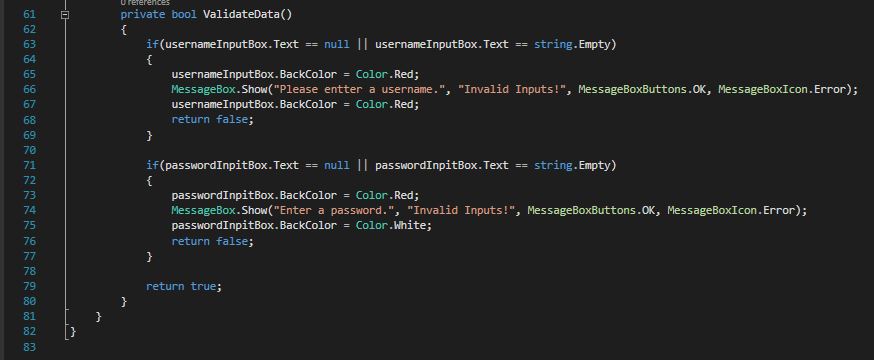
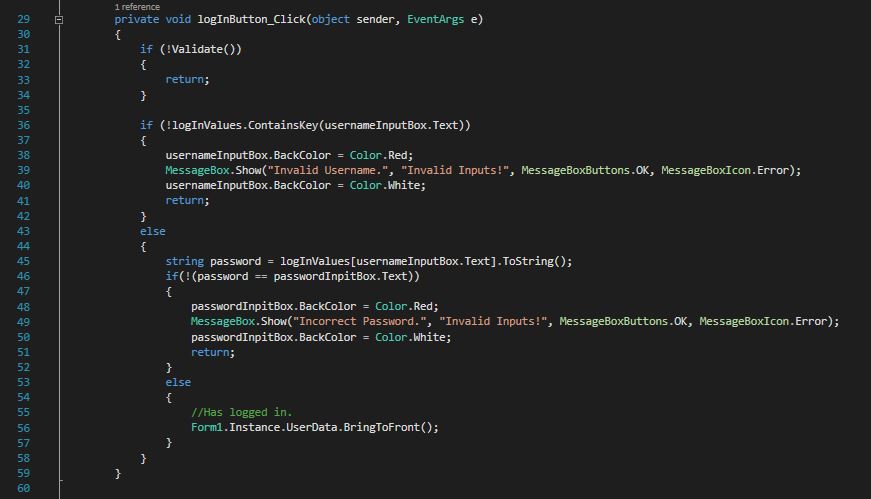
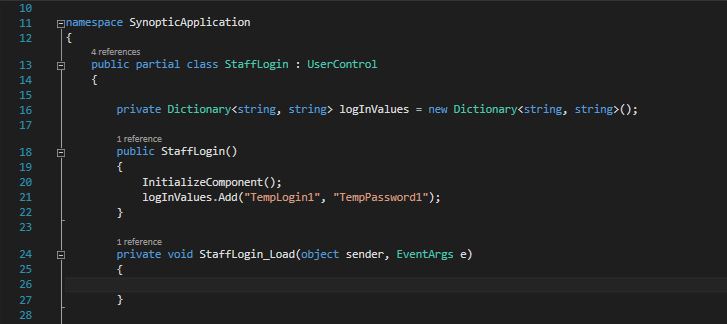
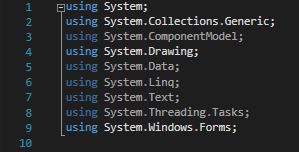
-Get side effect – If what is selected is in the enum class for side effects then class it as other

**Code**

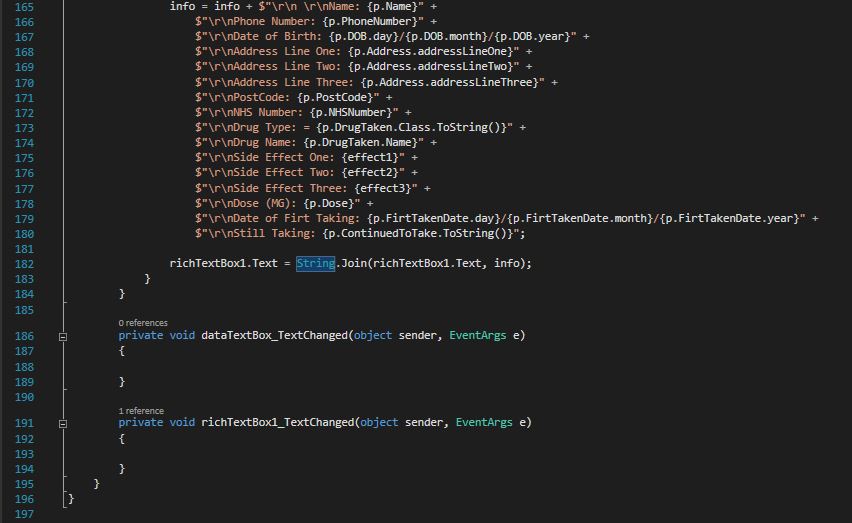
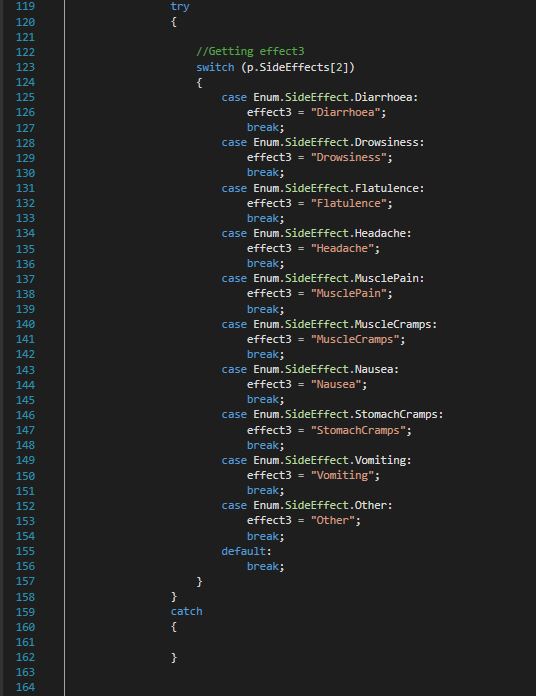
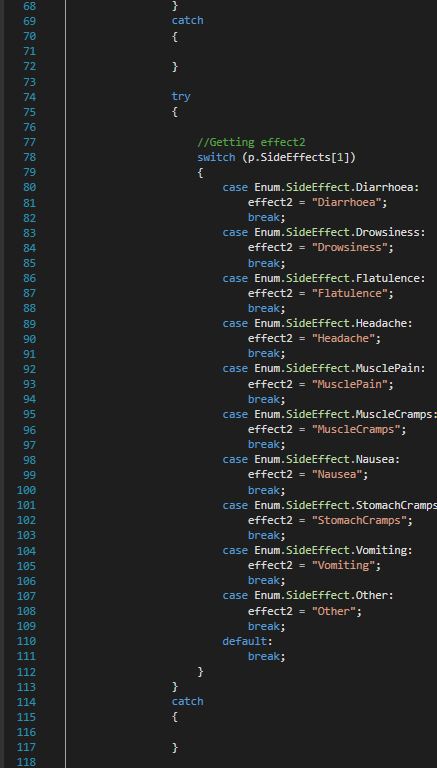
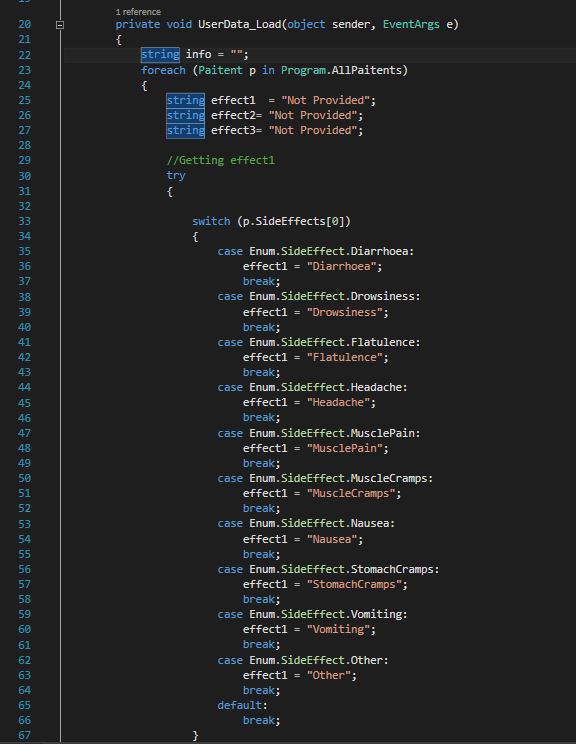
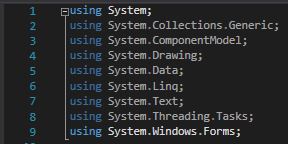
**Program.CS**



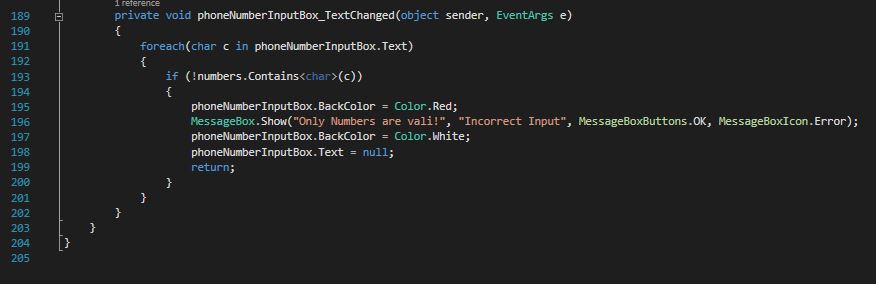
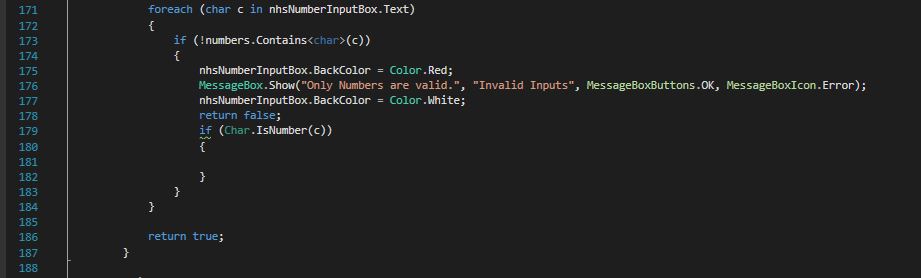
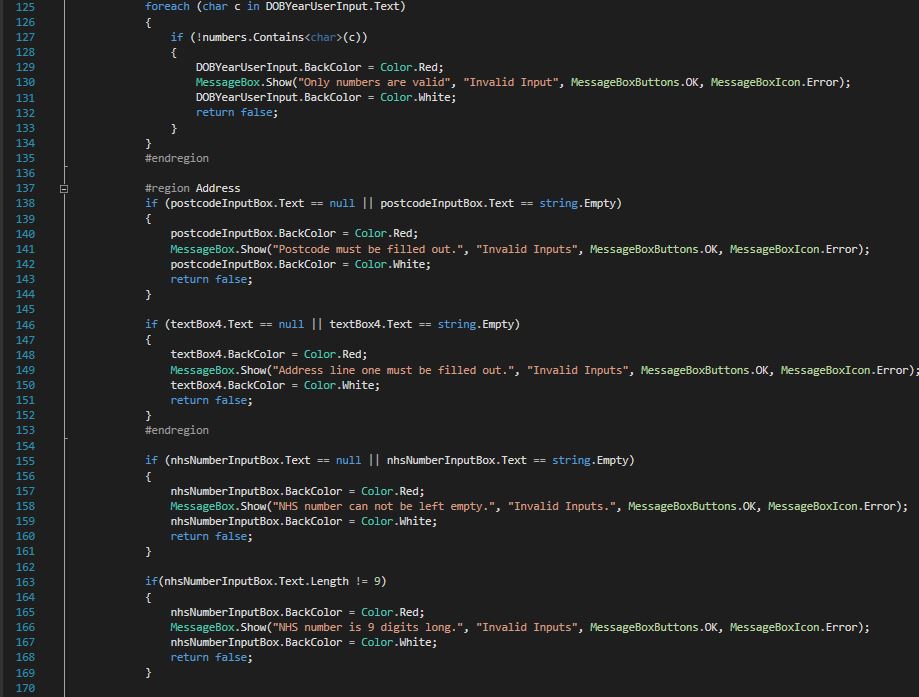
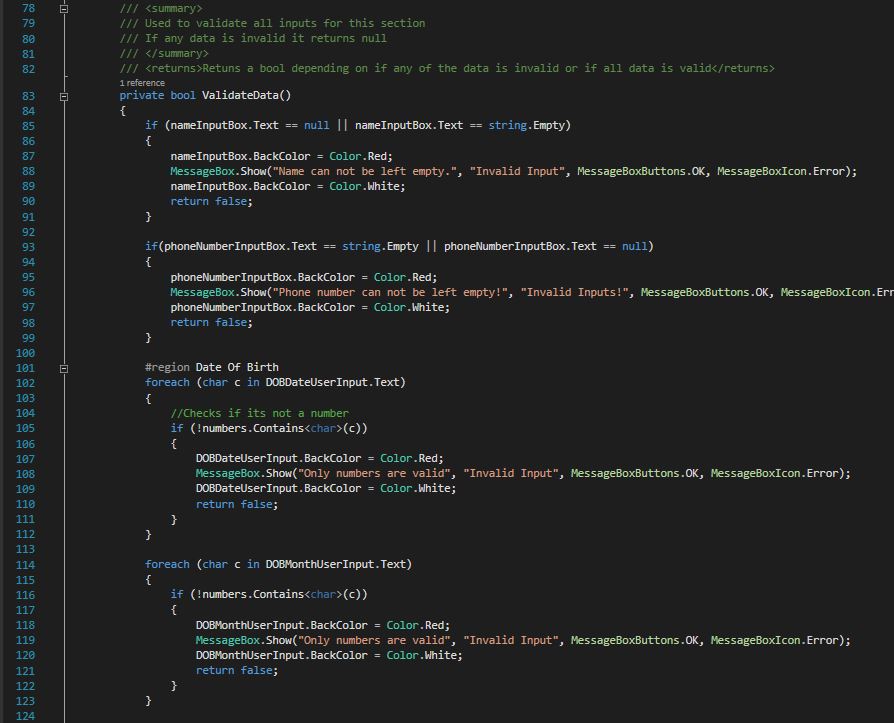
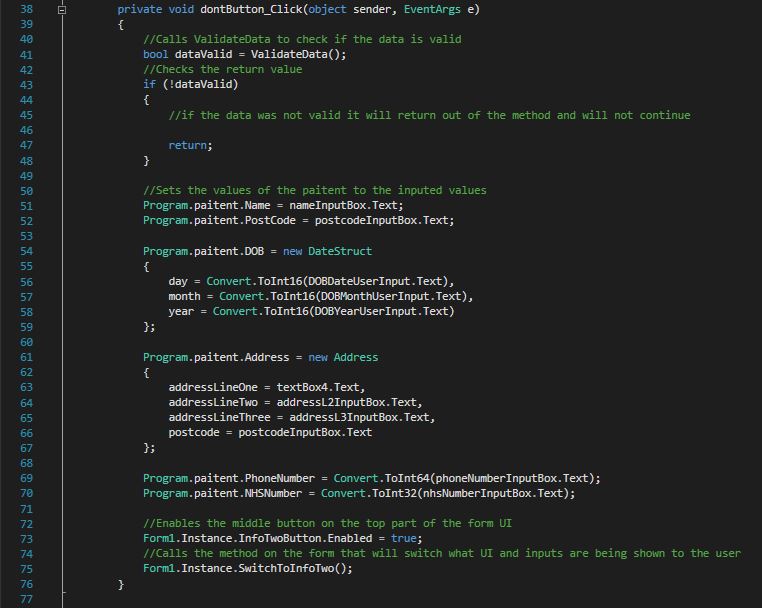
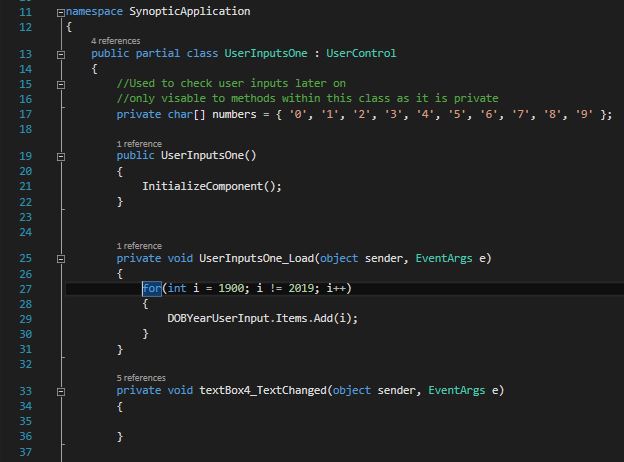
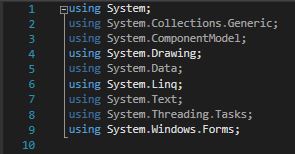
**StaffLogin.CS**



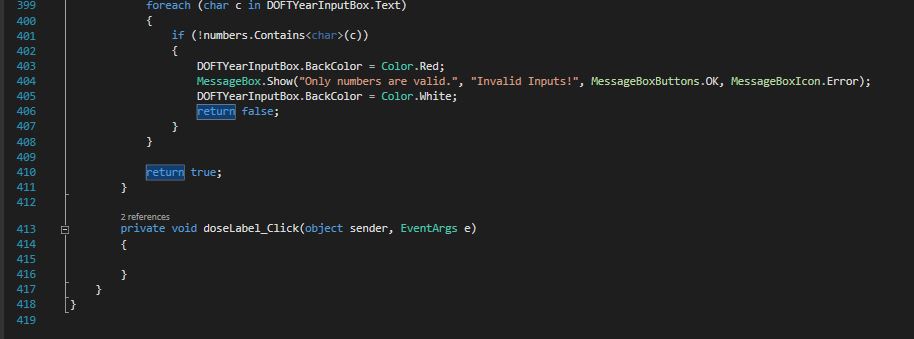
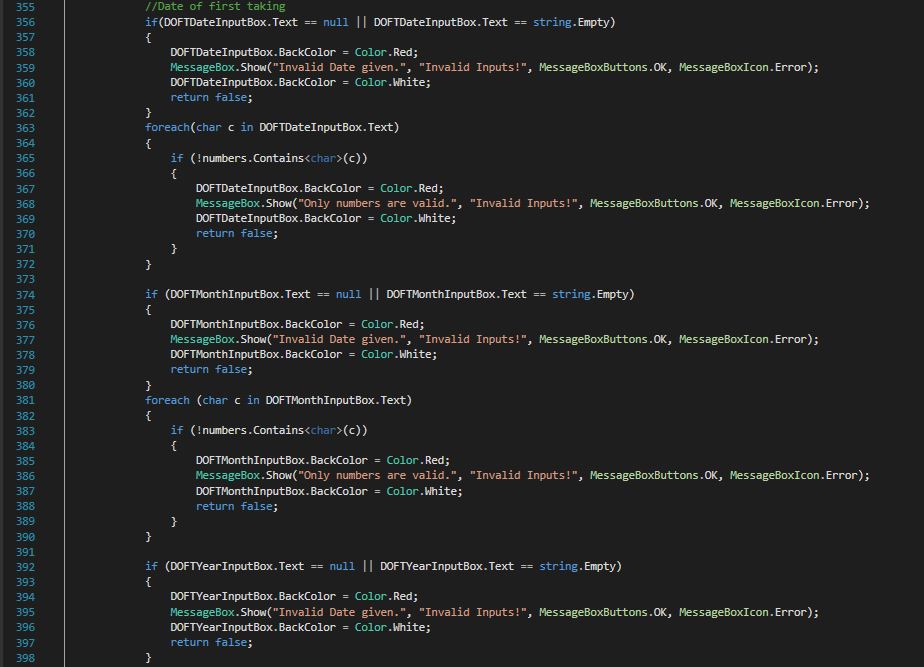
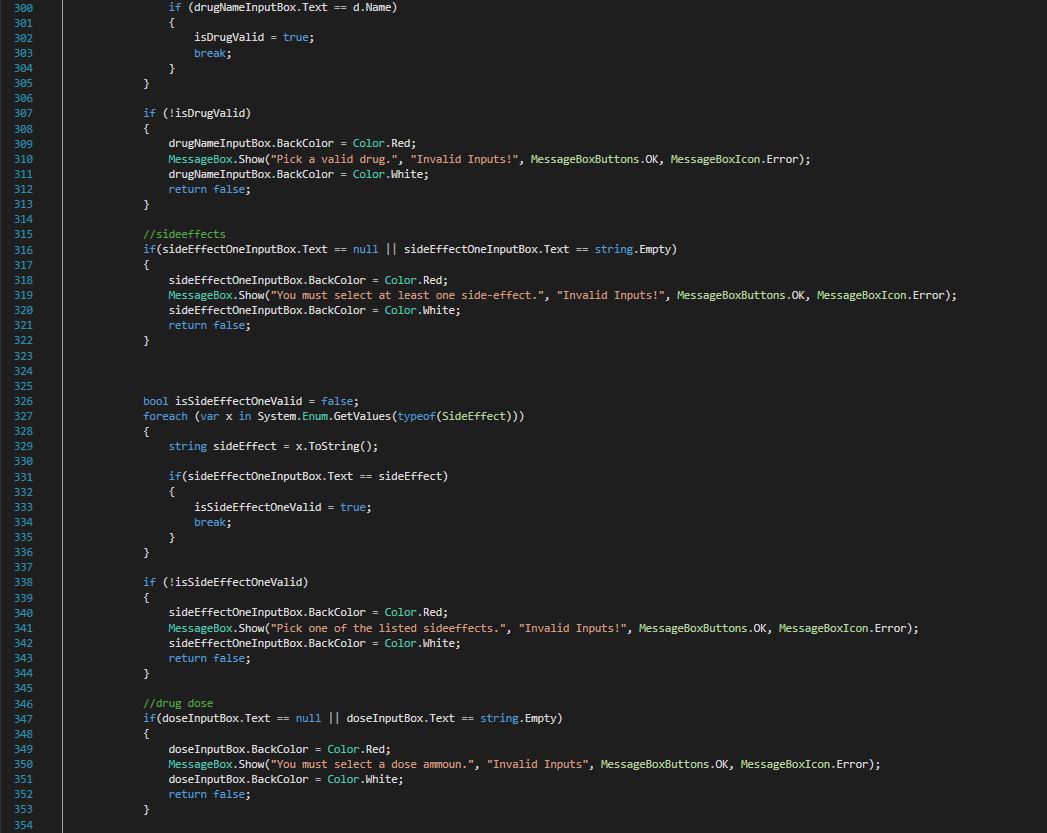
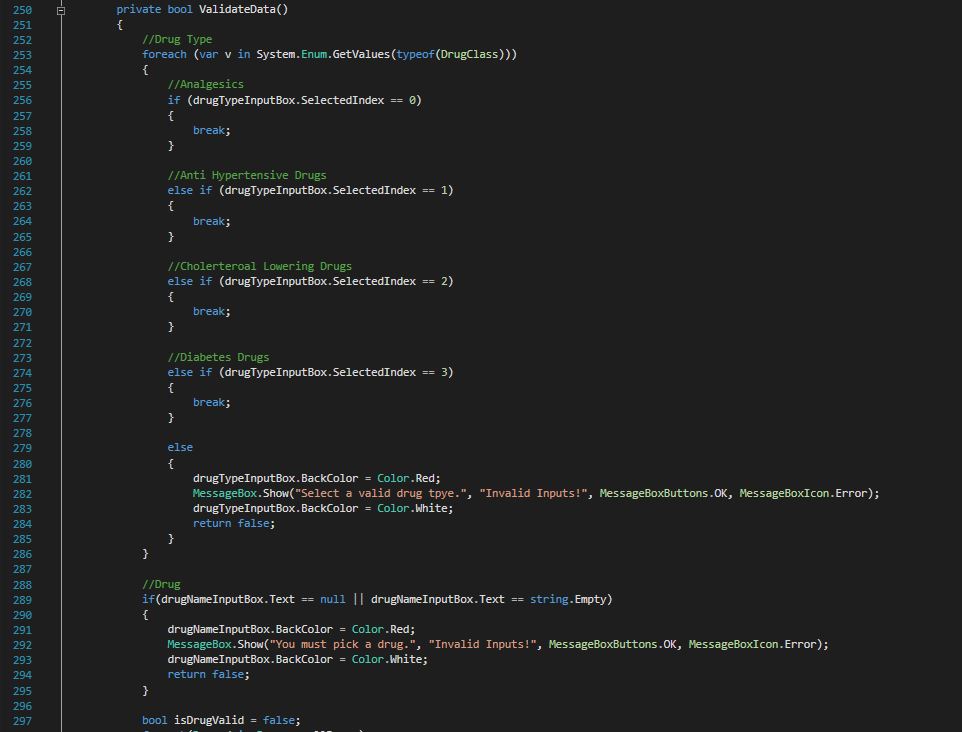
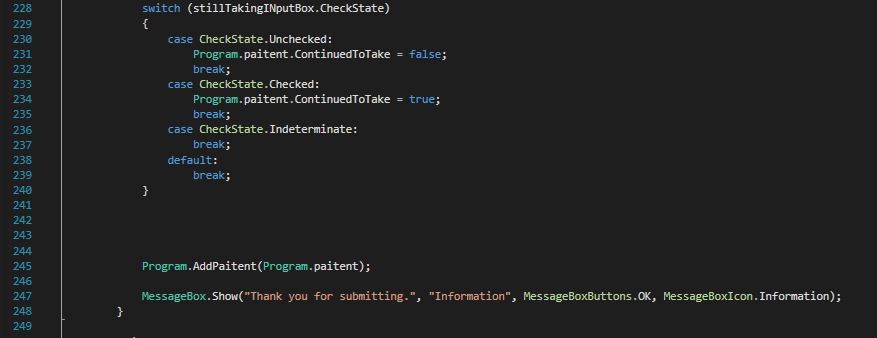
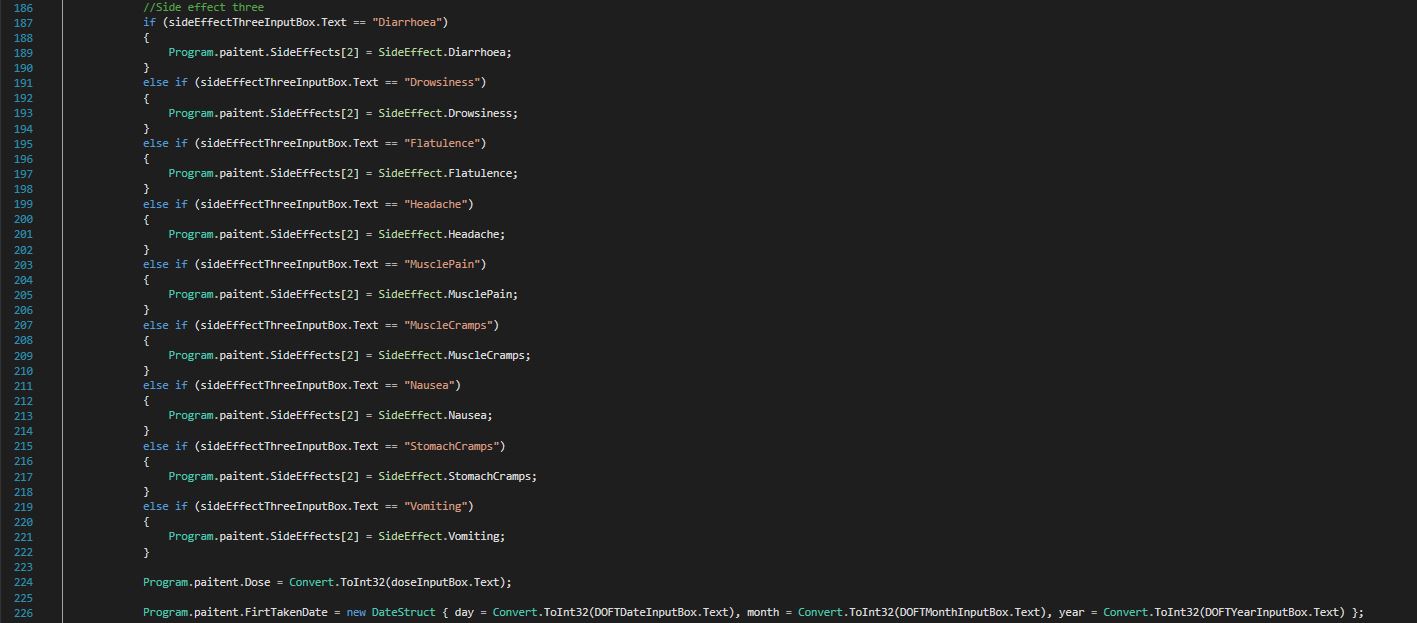
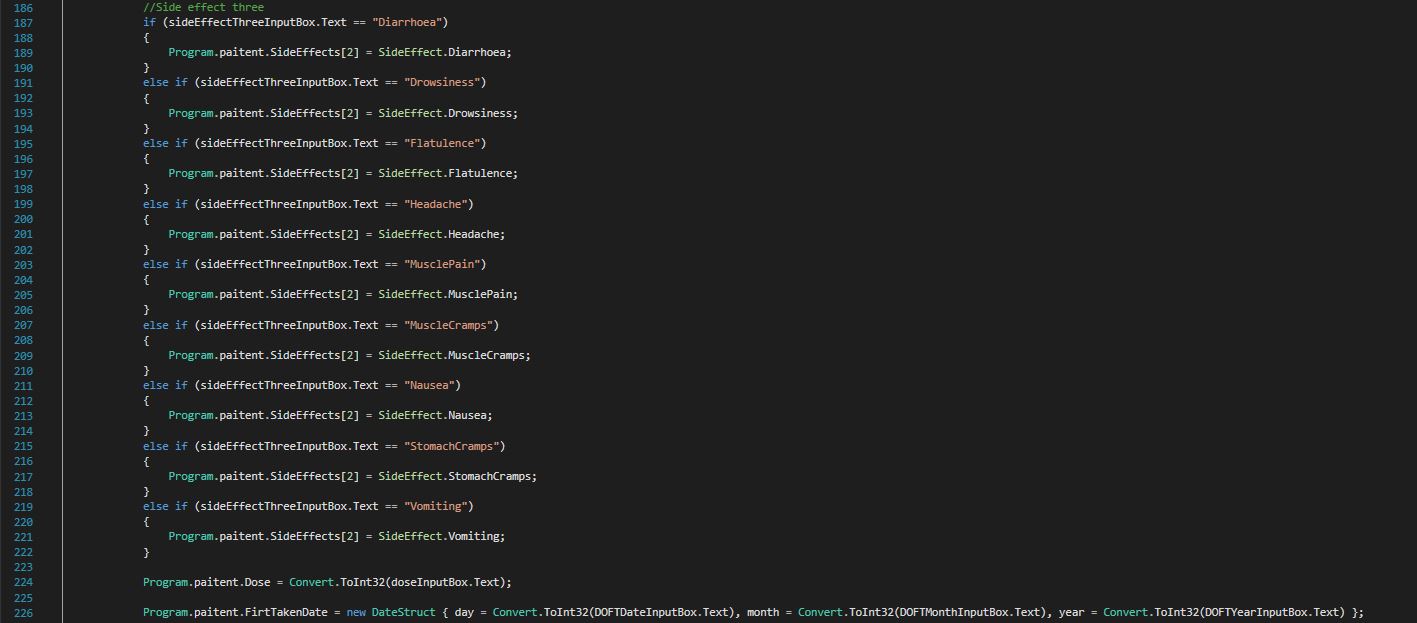
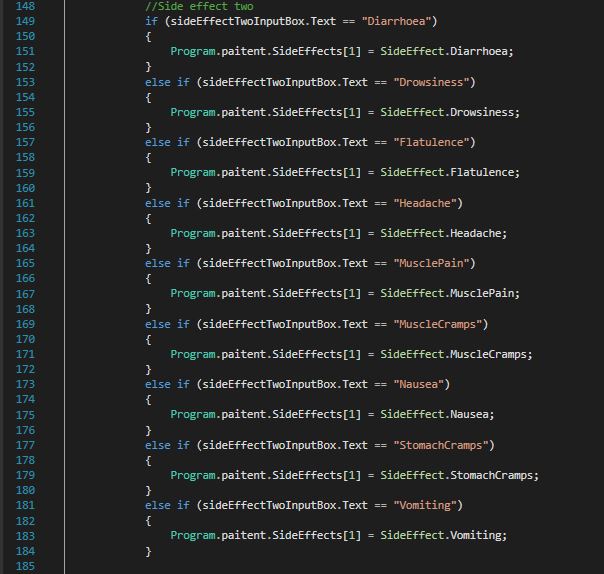
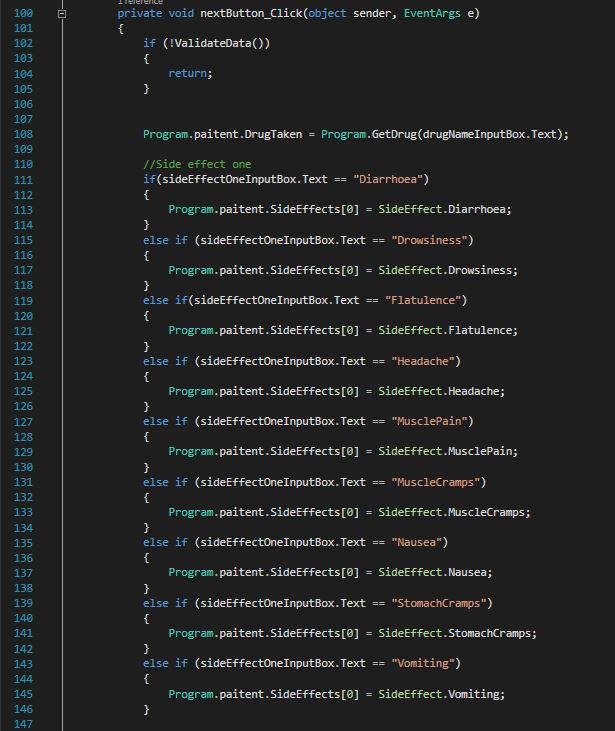
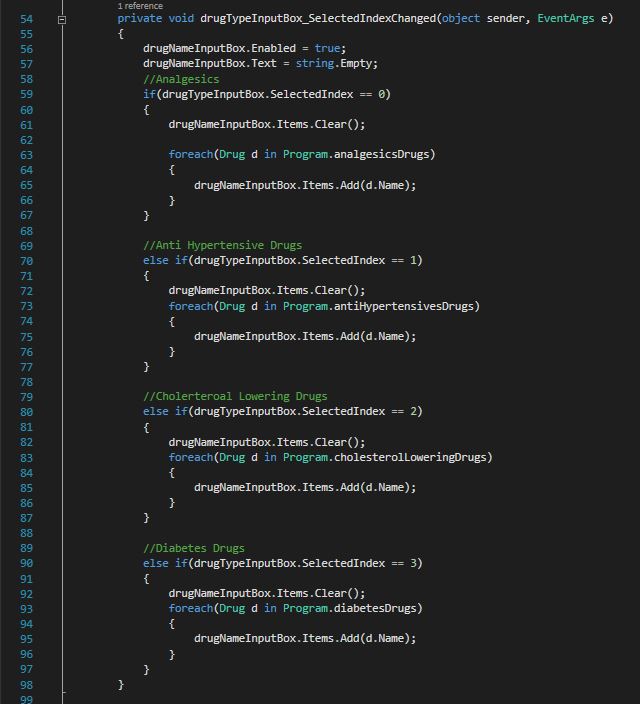
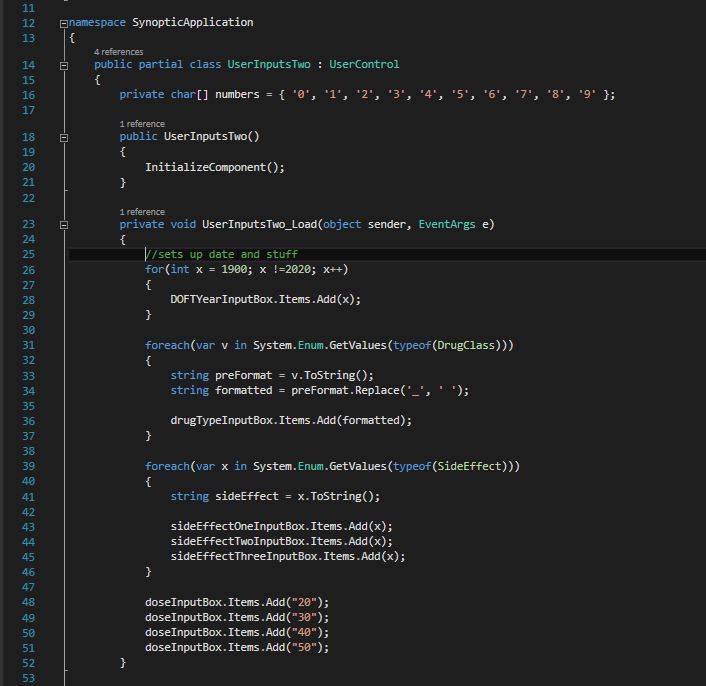
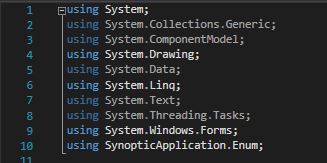
**UserData.CS**



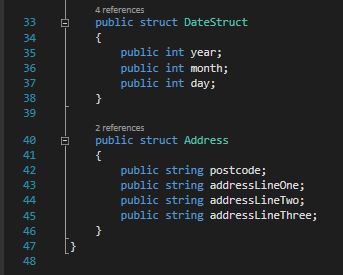
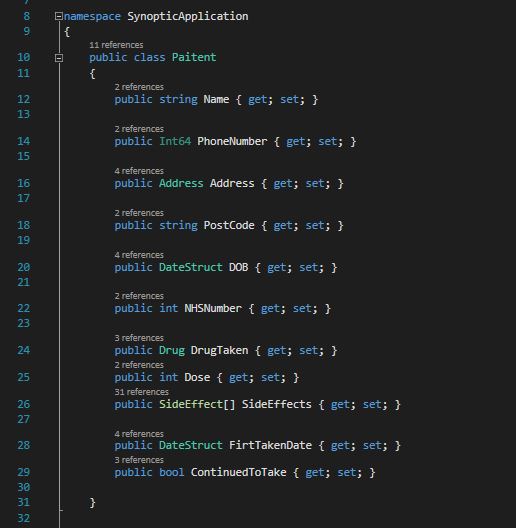
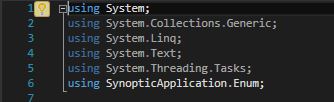
**UserInputsOne.CS**



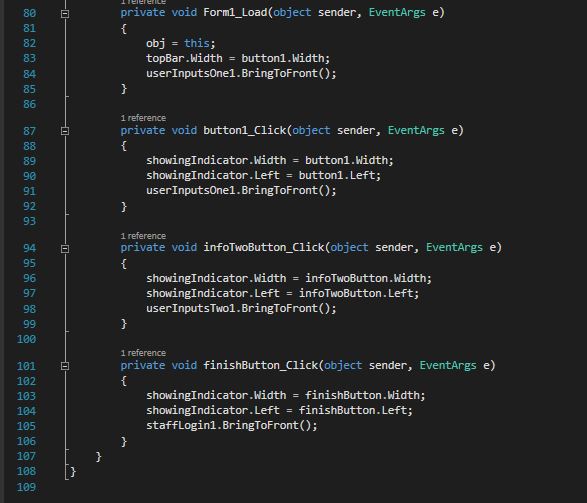
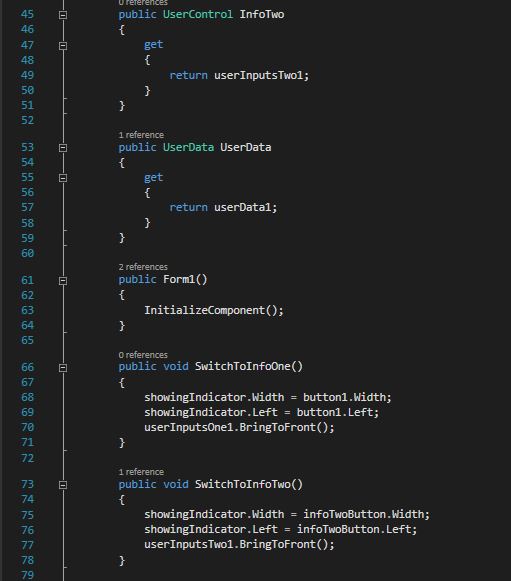
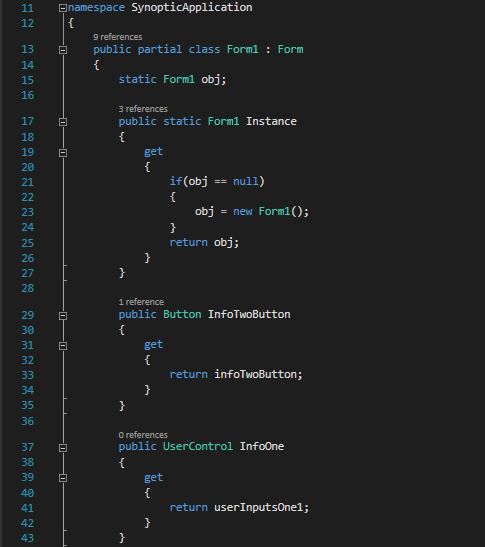
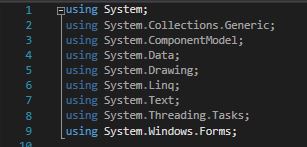
**UserInputsTwo.CS**



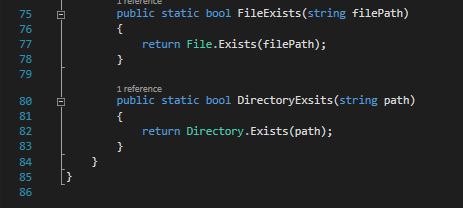
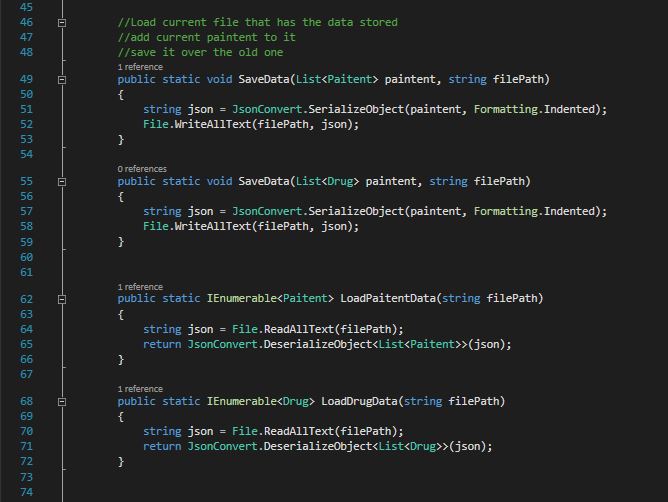
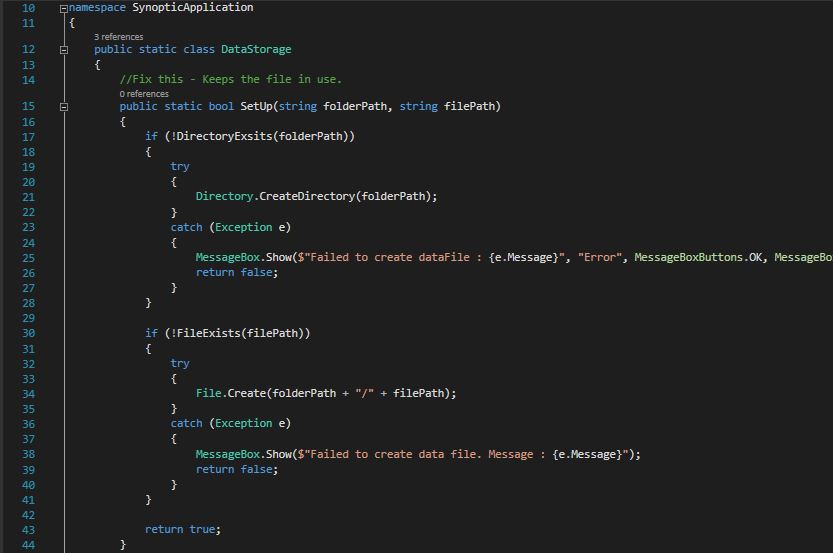
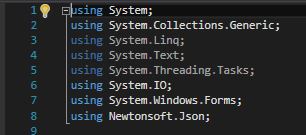
**Paitent.CS**



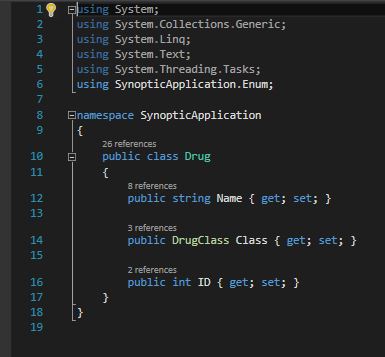
**Form1.CS**



**DataStorage.CS**



**Drug.CS**



**Enum.CS**

