5 – Post Prototype Refinement of Design

Table of Contents

5.1 Feedback

5.11 Approach to Feedback

To gain sufficient feedback on the prototype I will do what was done similar to the investigation and perform a range of methods to retrieve ideas and thoughts regarding the prototype, this will range from one to one discussions about the product to quick multichoice questionnaires. The purpose is to retrieve a wide range of feedback through different medias to provide a more whole and complete response rather than just verbal feedback, through the use of quantitative data I could also have the chance to tabulate the data and see where the prototype comes up stronger in areas and weaker in others. To help guide the users through the program a PowerPoint will also be provided to help aid the user in accessing features that otherwise would be unlikely to be seen if it was left to them, while I will be there to help them it is more up to them to get around the system, this will allow them to gain a more true experience similar to most users who won't read the guide when initially using the system.

Due to the large size of the system I feel it would be unnecessary and unideal to overload people on every section of the system. To prevent this from happening I am going to show the reviewers the system in sections that are likely to be used by them, with the exception to respondents who would give beneficial feedback on areas that could use critiques.

Name of Respondent	Relationship	Level of Competency
Yusuf Adam	Colleague	High
Joe Catton	Colleague	High
Emily Graham	Colleague	High
Phil Jackson	Tutor	Very High
Alex Nurdin	Brother/Colleague	High
Karl Nurdin	Farther- contracted Programmer	Very High
Suzanne Tomkins	Mother	Standard

Showing the user the program

This will stem from two types of feedback sessions, either the user will receive a hands-on use of the system with a document informing them on how to use the system, this will result in a more personal feedback process with more descriptive details and can produce feedback during their time using the system. Or a group session with myself going through a PowerPoint and taking questions more towards the end, this will create a group dynamic between respondents and will allow for a more collective discussion allowing the group to voice their ideas as a co-operative.

Questionnaire

One Method to explore feedback would be the questionnaire, while it is not qualitative data and I can't use it to get ideas on what to change, I feel it can be used to receive a more overall feel on the system which can be seen almost as important, while I will use other methods predominantly to see what needs to be improved this can be used to receive an initial idea. The questions in mind are basic and utilise a linear scale for specific features and propose ideas to the user. What I can gain from this would be a visual representation to the general assumption and can use this going into reviewing the feedback. It could also allow myself to have a presumption when going into discussion sessions.

Discussion

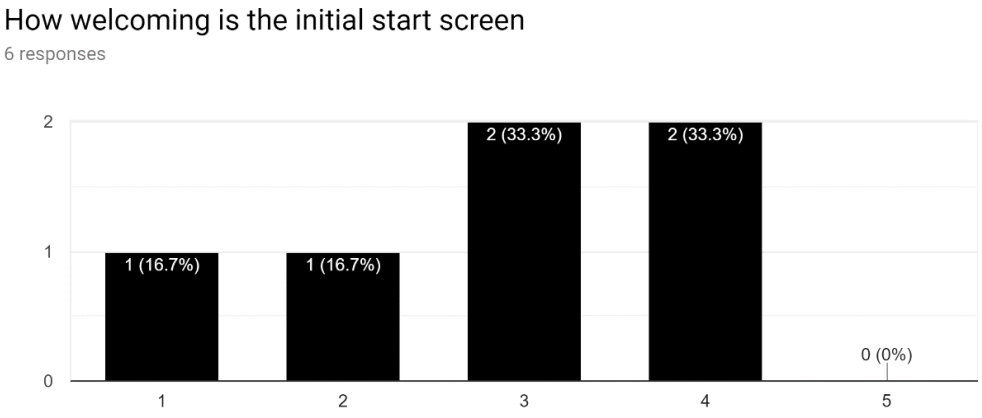
The next major area to receive input on the system will be the discussions, after having the chance to see the system, the user will be able to give feedback, giving them the chance to have a verbal discussion with me and provide feedback. I suspect predominantly I will get ideas from non-technical users and more critiques of the system from more competent users. Because of this I will try and aim towards the critical feedback from more advanced users but won't fully disregard ideas from basic users as they will be the majority who use the system.

Discussing the competency of the respondents

Before I continue I would like to talk about the reasons why the people that gave me feedback are worthy enough to be considered as third-party respondents. For my colleagues and brother who are all in the same position as myself, they possess no affiliation with the end user, and so can produce a credible feedback towards the system without knowing the exact requirements of the desired user. To add to this, like myself they too are require feedback towards their systems and so are willing to give constructive feedback as they know generalised comments have little to no use when having to redesign specific sections of the project. In regard to my parents again they have little to no affiliation towards the final system. With Karl being included the benefit of which due to his many years in the software development industry can allow for some very good critiques of the system. Finally with regards to Phil while obviously having some ideas about the system having read the investigation and discussion again like everyone else he has no link to Euxton again making him another suitable person to retrieve feedback from.

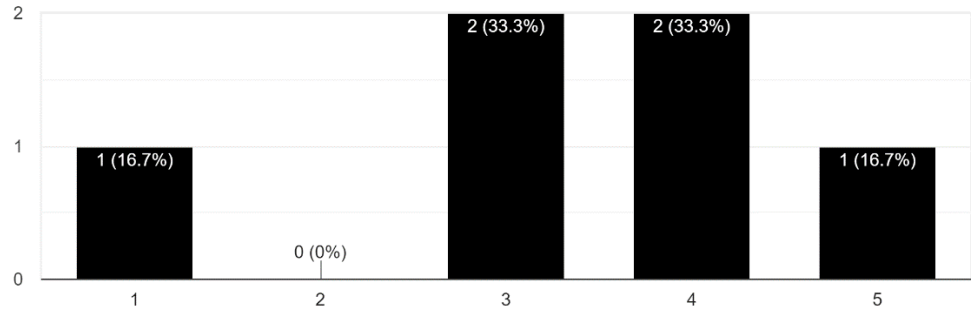
Feedback from the Questionnaire

After receiving the questionnaires, I now have a more general idea about how the normal user responds to the system. From a first glance over the data the feeling I get that general navigation and reading of the data seem to be lacking, while I assume it's not in all areas as the questions don't specify this I will have to take that all areas where hard to read and understand. However some strengths came up with regards to aspects such as account creation, admission generation and general layout. However through some specified question i.e. regarding the jargon library the feedback was similar to the expectation, however some surprises did come in the form of the answer to the question regarding consultant and data entry question 12, this was because while I have tried to reduce the fields that have been entered it seems that some more work has to be done to reduce this. However one area to consider that when less work is created for the consultant it is likely to be picked up by the patient. Other responses that surprised me where the overall feedback wasn't expected like the direct question about having a GUI was rejected by half the users.



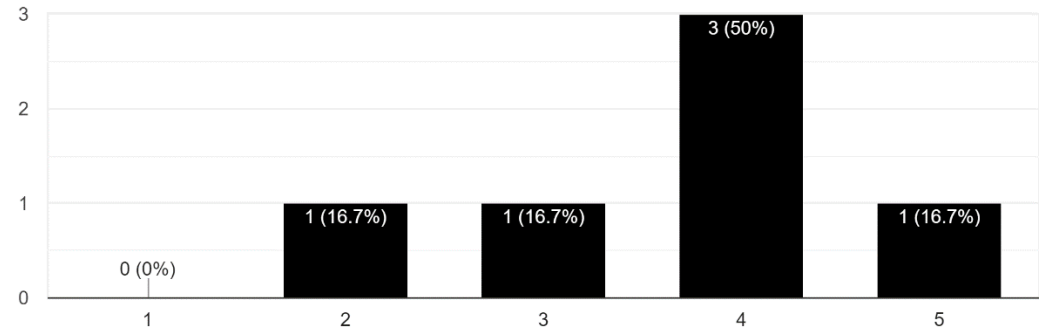
how easy would you be able to navigate the system after being initially asked to go to a particular panel.

6 responses



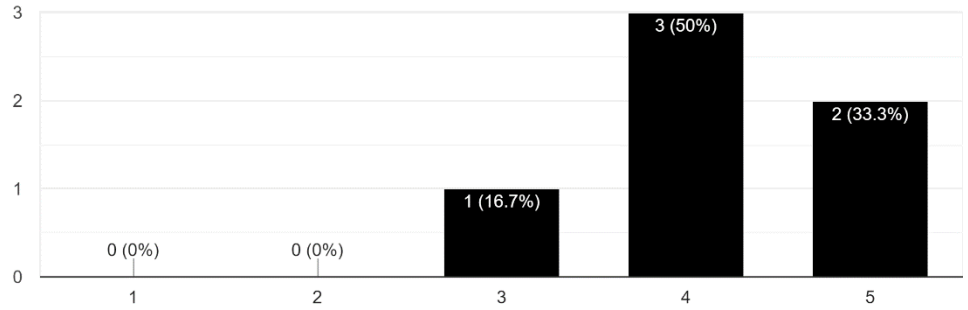
After seeing an admission being created could you create your own admission with nothing but the system's guidance

6 responses



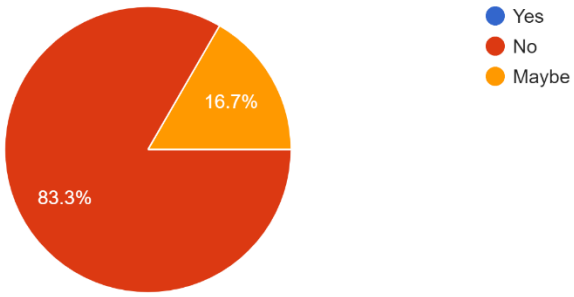
After seeing the chance to create an account, how easy did you find creating an account

6 responses



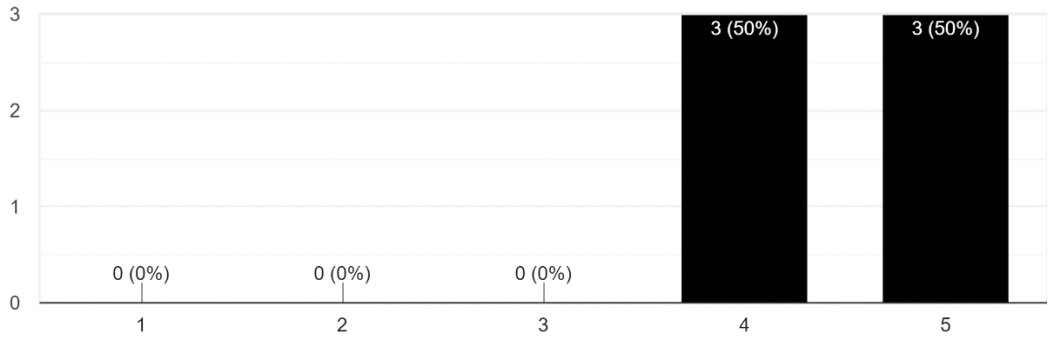
Would you feel comfortable if a non consultant had the ability to view your medical information

6 responses



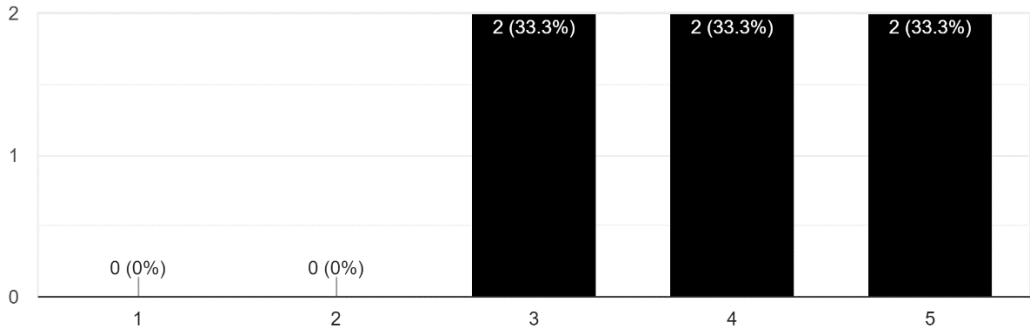
How do you find the ability to search definitions

6 responses



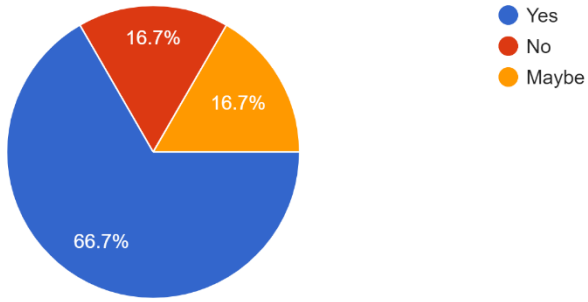
How organised do you think the demographic field layout is

6 responses



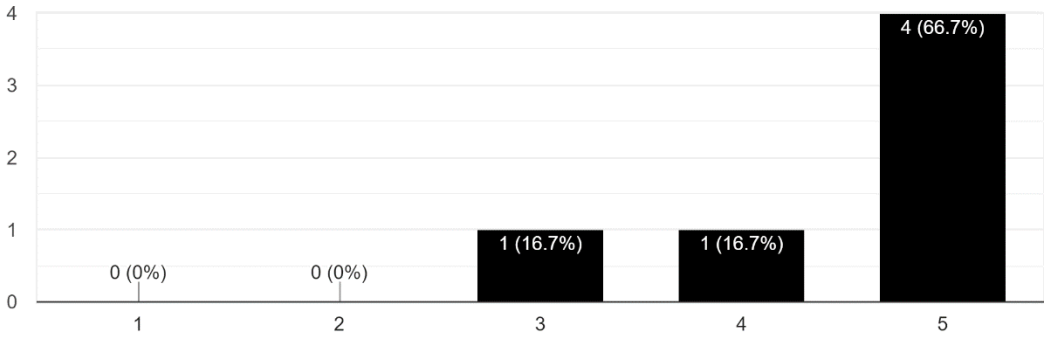
Do you think the patients should have the ability to search definitions

6 responses



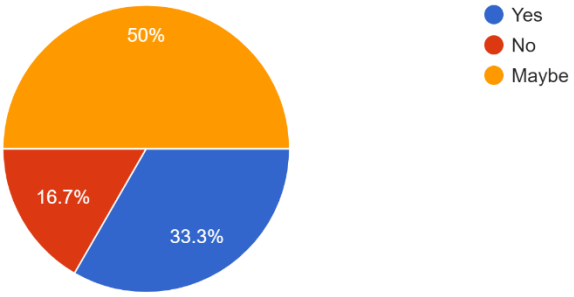
How clear is the layout of the admission homepage

6 responses



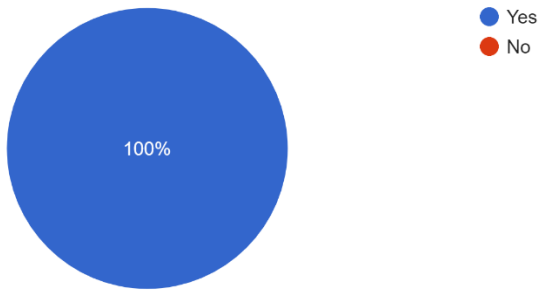
Do you think the jargon library should be on its own window

6 responses



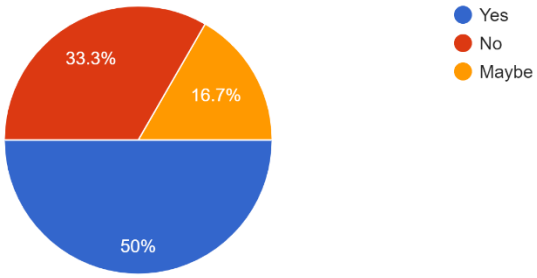
Do you think the consultant would benefit from having less fields to input data through

6 responses



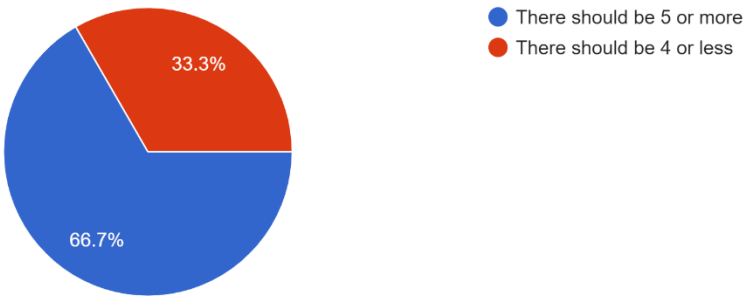
Do you think patients should have the ability to change information regarding an admission about themselves

6 responses



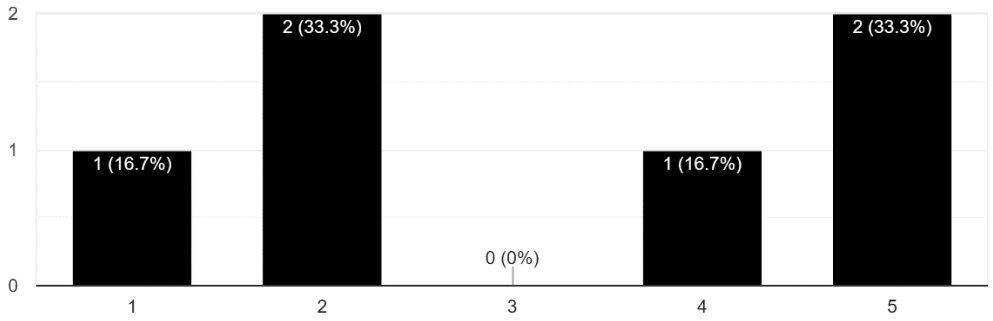
When creating documents do you think the consultant should have more than 4 document types at hand or should there be a handful at most

6 responses



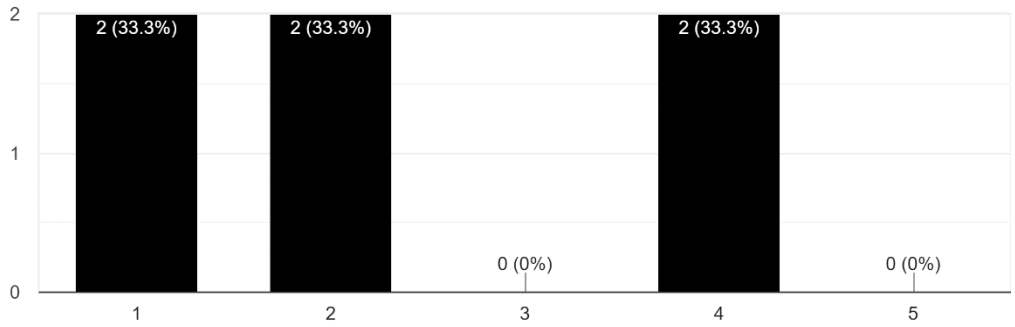
How reassured would you be to see a list of people who have amended your medical information

6 responses



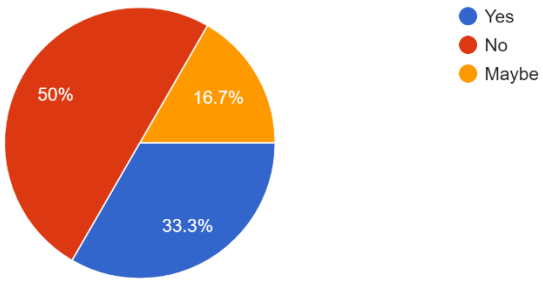
How well do you like the colour scheme

6 responses



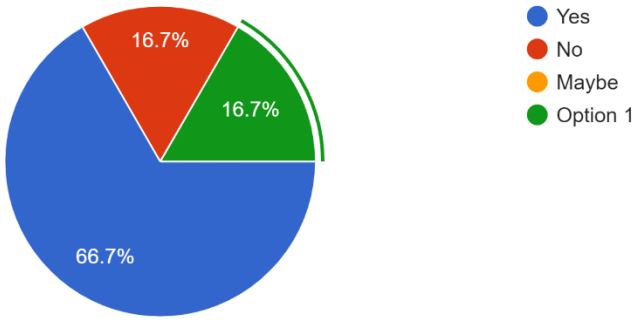
Do you think the Command line interface should be used for an entity who only has a select range of features to u...d requires little to know use for a GUI

6 responses



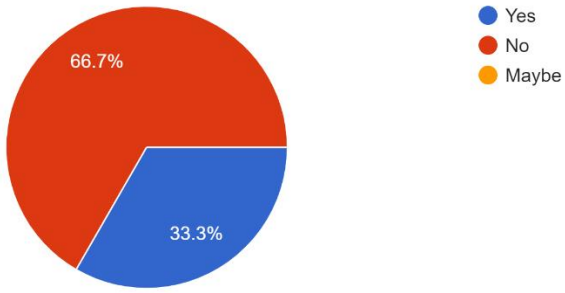
Do you think the patient should be able to see who has edited their information

6 responses



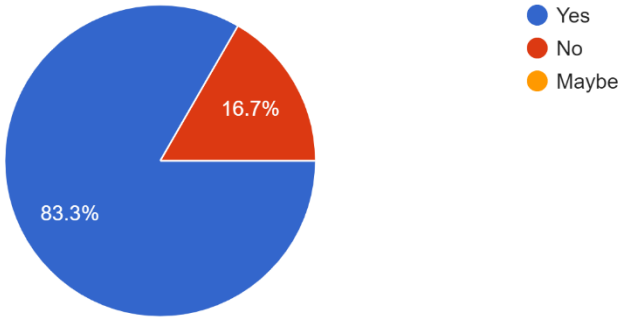
Was text easy to read and see

6 responses



Where buttons clear in general on their function

6 responses



5.12 Consideration of Feedback

People	Feedback number	Feature	Feedback	Consideration	Accept /Reject	Implications for Redesign
Emily	**1.**	Colour scheme	On having a discussion on general formatting and layout of the IO a notable result appeared to be that the colour scheme created in the design appeared to be too basic and needed some more colour to be more inviting to the user.	I am more than happy to consider the idea, on reflection the whole purpose of the GUI for the patient is to be welcoming, the bland and serious tone I went for in the prototype appears to be not working in this area. Because of this the obvious plan to fix this is overhaul the colour scheme for the system.		The implications for the visual redesign will not produce any detrimental effect on development, this is down to colour schemes being declared through variables, so changing colours will simply be the act of either the changing of a hex value or just reassigning a new colour to the component. The implications with respect to the end product will be a big improvement after looking into the feedback. As initial impressions play an important role of how well a user feels on the system the colour scheme could be a big reason to how the user feels on the system.
Emily	**2.**	Welcome screen	From Emily her feelings towards this was that the logo wasn't consistent with the formatting of the system and would need to fit in with the design. In addition to this the other criticism was that the help text was way too small to be seen without getting close to the screen.	Again, this could play an important role in initial impressions of the system. Because of this is it would again improve the user experience and so would not be an issue to include.		While the process of amending these aspects will be an important feature. The implications won't be great in any way besides a visual improvement for a screen only seen for a couple of seconds at most during the use of the program. Nevertheless, it will leave an impression of the system on the user on their first use of the system and therefore it has the utmost priority.

People	Feedback number	Feature	Feedback	Consideration	Accept /Reject	Implications for Redesign
Phil	**3.**	Welcome screen	Phil's initial impression of the screen where critical on first inspection. With him giving the impression that it was confusing to use the initial screen with little clarity towards new users on using the screen, the general consensus was that he believed it was too confusing to access the login stage or to enter the symptoms	As Phil's feedback was by far the most critical I believe that his feedback is important towards reworking features deemed to be not perfect. Because of this, the idea to rework the welcome screen is greatly appreciated and because of this I will accept this feedback and will perfect it for the final product.		While the rework will take some time to redevelop and link panels together to correctly reflect the changes, the redesign is where the implications take up time and resources. This is where I will have to rethink the initial stages of the system and plan around how this will work around current features.
Sue	**4.**	Welcome screen	The feedback from Sue was that the initial screen was basic, similar to Emily's opinion the text was too small and it looked far too dated for someone who may want to view documents	Again as it would benefit initial representation of the system I will accept the feedback, there is little reason not to know since multiple people have picked up on this		Again there is little implications to include this feature so it is a clear choice to include it on the final version of the system. All variables associated to format can easily be adjusted to redesign the initial screen. While the redesign currently at this moment is little I feel a larger change could happen if enough feedback is provided.
Karl	**5.**	Searching	On showing of the consultant homepage and the ability to search patients to Karl a few aspects of this area where talked about. Initially the primary key of the entity was an area that faced criticism, his view was that a consultant would be unlikely to search for a patient by the id but rather surname something that they are likely to know rather than a specific value.	On reflection I one hundred percent agree with him on this regard it would be unusual to search for a song on iTunes by its ID rather than the title track or album name, because of its importance I will definitely take on this feedback and will resolve the issue by allowing multiple fields to be queried.		The implication on development would be significant as it could be utilised for every other search by the user on the system. Because of this it will have to be thoughtfully worked out this time to make sure that it can return every item that fulfils the query. Alongside this the fields that can be searched for will have to be expanded on and recorded down so that every search option includes the correct fields to be used.

Karl	**6.**	Amending consultant information	After showing Karl the consultant homepage an area that came up was data revolving the consultant, while it could be static and remain the same from the point of the account's creation, he suggested that it could be updated and changed similar with regards to the patient's information	I will accept the feedback due to the fact any entity should be able to alter basic information about them. On reflection it would be pointless to include fields regarding the consultant if they would be incorrect in the future, therefore its crucial to include this		The implications of this have two results. The first one is the fact that the process of introducing it will be partially easier due to the fact that the same process already exists for the patient's demographic and admission information. Currently there has been a lack of purpose for these fields besides viewing the contact bar on the initial home screen. By having the ability to change this, this could allow new functionality and meaning to the consultants' fields.
Karl	***7.***	Reducing staff access towards patient information	While the staff entity was absent from the prototype is still cropped up a fair bit during the discussion with Karl talking about the practicality of the entity and the role the real staff play in hospitals, the feedback suggesting that the entity have a more controlled access to the patient information with a larger sense of limited access similar to how I initially planned it.	The main focus of this entity is to offload work from the consultant so that they can focus on more important matters, because of this, staff where initially designed to be main the user the patient interacts with at Euxton, by severely limiting what the staff can see in regards to patient information, this feedback makes it almost impossible for staff to interact with patients. Because of this I am going to reject the feedback.		Another reason that the idea was rejected was also due to the fact that the implications of the inclusion of this feature on the redesign meant that the entire entity for the staff would have to be reworked, in a way that limited its already restricted data access making the user have realistically no access to. This is would be troublesome down the line when the staff is trying to make judgment on whether to allow for a re-prescription of medication with little information regarding the patient being present to them.

People	Feedback number	Feature	Feedback	Consideration	Accept /Reject	Implications for Redesign
Karl	**8.**	Jargon	One aspect about the jargon library he suggested was that it was unnecessary for the consultant to have access to the feature in some regard as they are likely to be the user who understands the terminology, whereas the patient may have no idea what an endoscopy is. And so the main feedback would be the user would gain the feature instead of the consultant.	When considering the feedback again, I think this should have been realised much earlier on as it seems to be strange that only consultants can view this and be the ones who edit this. Because of this I will accept the feedback and include it on the final system. What could happen is that the current panel could be used to add definitions to the list, while a new window could be created for the definitions.		As I currently have no outstanding issues with the jargon library I feel that these actions could turn a good aspect of the system to a unique part of the system. The implications are somewhat drastic. It will mean that the current panel will now be assigned to a new window, in addition to this on most panels the option to bring up the jargon library will have to be bought up. The result of this will give an overall better perception to the system as this could be thought as a "Tool" similar to how other systems provide external features to help aid the user through the system for instance the ability to draw over webpages on Microsoft edge.
Karl	**9.**	Rename jargon	A minor addition to the Jargon library was that the term Jargon was to associative with basic terminology and seemed a little unprofessional, I believe he found the term too associated with lower levels of competency. A suggested rename would be to use "Glossary of terms".	Again, I have no reason to counter this suggestion. Personally I believe the rename suits the description much better so it is a clear choice to accept the feedback		As the tab is actually a button, the implications on development are minor, however this will again improve the user experience as the implications visually will result in a more sophisticated system due to buttons being defined much clearer.

People	Feedback number	Feature	Feedback	Consideration	Accept /Reject	Implications for Redesign
Karl	**10.**	More focus to the staff than consultant	Similar to before where Karl suggested a tighter access to information regarding staff entities, another aspect he would have liked seen was that while patient interactions would be predominately seen by nurses (staff) who would regulate medication, consultants would oversee the process and view these actions take place and could confirm changes to prescriptions or other amendments to the patient's information.	While from the initial concept the main focus revolving the staff entity was to offload work from the consultant to that entity, I will take on Karl's suggestion due to the similarity. While this current entity was missing from the prototype, it doesn't mean that the priority has changed. Because Karl here suggested the exact same thing I initially planned from the offset for this entity, I will take on board his ideas but overall the concept hasn't changed but rather has been updated to follow feedback.		Because not much is in place for this entity currently, the implications will be minor in terms of development as the same amount of time will take place. However in respect to the end product I think I will see some improvement, with a big strive now is to make sure the entity is relevant in the current system and isn't pushed aside by the consultant, due to the fact the consultant will have similar features and the fact that staff have less access to information.
Karl	**11.**	Creating an appointment before admission	Another idea presented by Karl was that when the patient creates an appointment with a consultant if no admission is selected an empty admission is created and could either create a new admission or link it to a current admission. While a little vague the idea I think he wanted to get across was like how anyone can receive a basic diagnosis the same could be for appointments .	While currently it is still possible in some hospitals and surgeries to visit before having some information kept about you, I believe with how the account system works, currently consultants only have access to patients through their admission. If a consultant would want to have an appointment made they would need the staff entity to create an admission and assign the consultant to it. Because how my system would not suit this feature in any way I am going to reject the feedback		The implications of rejecting this feature have made me realise how unaware I currently am of the features to the staff entity, because of this I feel the result of this consideration have made me want to rework the staff entity to play a larger role. Like what Karl suggested earlier on. While I wanted this in the design now I will make sure the staff plays a larger role to the lifecycle of a patients visit. While the implication of making the consultant have the ability to create appointments before an admissions creation had very little to do with this. Having to reflect on the entity has made me realise how unsure I was on what this entity would actually do.

People	Feedback number	Feature	Feedback	Consideration	Accept /Reject	Implications for Redesign
Sue	**12.**	Inclusion of GP's	An idea pushed forward by Sue was that Euxton while primarily focused with expert consultants some documents where for record purposes towards General practitioners, therefore include the GP in the system as another entity, who could also see their patient's documents.	I'm not going to delve too deep into this while I could introduce an external party of the hospital besides new patient's I feel for a system to manage documents and bookings it really has very little to do with this entity. In reality what would happen would be that the document would be sent to the GP as their input would be minor until the patient is discharged. Because of this I am going to reject the idea.		The implications of excluding the idea will not be major in any way as I had originally planned not to include the entity, there will no change to the end result. While it's my judgment to not include this entity, I feel that the entity's inclusion would not have any benefits due to how little the feature would be used. To add to this a whole new user would need to be accounted for, for such a small payoff I feel it doesn't merit an inclusion.
Phil	**13.**	Boolean Attributes for Drinking and smoking	One aspect of the account information Phil felt that could be improved was about the two Boolean state attributes for being a smoker and drinker. The idea pitched was to introduce a slider to allow further accuracy when determining how much they drink/smoke rather than a basic attribute saying whether they do or don't. It could also allow for more truthfulness with a patient as they are more likely to say they smoke a handful a week than say none at all.	In hindsight this is obviously a major aspect to the attribute on the patient as most consultants ask how many units are consumed in a week rather than whether they drink or not. This suggestion could allow for a more realistic and accurate approach to data input and because of this could also allow the attribute to be used in some minor determination of diagnosis for the admission. Because of this I will take on this feedback and make sure it is used on the final version of the system		The implications for including the attribute will be somewhat large. As the field currently relies on a radio button to change its state this will have to be replaced with a JSlider. As I have no experience it also means I will have to test and experiment until I think it works. As the slider also takes up a considerable amount of space I will also have to rearrange the layout. Another implication to take notice of will be the changeover of datatypes and the files I will have to go through changing the Boolean value to an int, while not hard, it will take time.

Phil	**14.**	Login screen buttons	Phil had another issue with the login screen with the fact that in order to create an account the use had to select login as that user, the issue being the user is unlikely to select login when they don't have an account on the system. To add to this while it contradicts the previous statement in some manner, the create account button should not be the same size if it is only going to be used once for that user.	I feel that the feedback presented here is constructive towards the initial use of the system, obviously based on the current feedback the area the system is currently lacking in is the initial few screens because of its constant reappearance in feedback I will accept the feedback and will consider it in the re-design of the prototype		Again the implications of this feature on the redesign aren't too much of an issue this shouldn't be too much time to rename a few components and move a few buttons. However this will have large implications towards the initial impressions, as of now the majority of people have picked up on this aspect. Because of this, it leaves a negative initial impression on the user, by resolving this it should make the user feel impressed rather than confused. The implications on the redesign will also result in the first two panels needing to be reworked.
Phil	**15.**	Login Screen Design	Phil found the login screen was unclear of the function of the login button as the next page also provides more actions other than just logging in, he suggested redesigning the screen be more current and similar to most modern login screens, with also renaming the button to be clearer on its purpose.	I believe the feedback to be honest of Phil and appreciate the truthfulness of the comment and believe that end result will benefit the system much more than what currently exists. An overhaul could also allow for an improved start screen and login page. Because of this I will accept the feedback and will rework the start screen.		This as of yet is the largest feedback I have received because of the nature of it, the implications are substantial meaning the entirety of the initial screens are going to be re-worked so that it is easier to use. Because of this I am going to have to go back to the designing of these panels and make sure that I like them. However this will only be up to the part where the user logs in, the general consensus was that the home screens themselves where fine besides a few colour changes.
Phil	**16.**	Colour code	While Phil wasn't too critical on the layout of the general homepage or the components' sizes or fonts, one critique was the colour of the back panels suggesting they were too similar making it look like it was bland, he commented that the next appointment box was a step in the right direction and to go for a more variant colour palette.	I think this feedback delves deeper into the issue Emily initially raised when discussing areas for improvement. While Phil wasn't entirely against the two-tone gradient aesthetic I was going for I can accept the feedback and incorporate it on the final version of the system.		Similar to Emily's view towards the colour code I will also discuss the implications of this feedback. In regard to development it will be a matter of minutes to reassign colours. In respect to the redesign I will need to make sure it looks great whilst still looking professional. To make sure the theme looks consistent I will go through the system to make sure that nothing looks out of place due to the changes.

Phil	**17.**	Discharge as false	Another critique Phil had a particular aspect of the system, while it seems minor and inconvenient it needs to be mentioned. This was that the wording of the dischargement status sounded wrong and lengthy to him and suggested if it was to be changed to an easier description that every user could understand.	While this was a temporary aspect and was likely to be changed at a later point, I am more than happy to see it get picked up on now and will include the feedback on the final version to make sure it sounds correct while still being able to be changed due to the current status of the admission.		Again this is another visual issue, this will have little implications on development or redesign. What I will need to look out for and make sure is that the text is worded correctly. In addition to this is to make sure that the wording makes sense for when both states occur: so when discharged and not.
Joe	**18.**	Two states for gender and sex	Here Joe suggested while showing him some aspects of the system was that while in current society gender is a spectrum rather than a binary state, to include two attributes one describing the gender they identify as, and also the genitalia they were born with. In addition to this the user may not feel comfortable to disclose what gender they are here and should have the option to not say.	while I could dive deeper down the rabbit hole and list every known gender for inclusivity's sake , I believe that the standard male, female other and joes suggested "prefer not to say" will suffice the user. Because of who important gender rights are in current society I will adhere to this feedback and make sure it is in the final system		The implications of this feature is much larger, for the redesign I am going have to redesign the file to make way for a new attribute attached to the object, in addition I will have to move around a few components on the demographic page to make room for the extra component. However the implications on the final product will mean that there is inclusivity for the user's gender identification alongside a medical aspect for consultants and other professionals.
Phil	**19.**	Quick suggest Allergies	As most people with allergies are allergic to common items like latex or penicillin it may be beneficial to include a quick suggest similar to the quick symptoms panel used in admission generation of your system. While you will need to allow manual entry for allergies a quick suggest may be very helpful towards the user experience.	This request is minor and so without much conversation on the issue I will accept the feedback and include it on the final system. The only thing I will talk about this is that it's only being included due to the fact the feature already exists on the system and will greatly improve efficiency using the system.		The implications of the quick suggest is that the layout of the new patient panel will have to be reworked to allow room for the new component. Alongside this the implications towards the end result will improve the user experience as most people have similar allergies. The implications for the development will mean some time will have to be set aside to allow for a new component and making sure it works alongside the manual entry part

Phil	**20.**	Remember me	One feature Phil suggested was a remember me feature similar to how desktops that are often used allow the use of usernames to be remembered on the system, here Phil was suggesting a feature where a user may be recognised as the predominant user and could be automatically assumed to be the desired user of the system, where all they would have to do would be to enter their password.	Again I can see no reason to reject the idea, its small enough not to impact development drastically but would create a better experience between users. To add to this it could also play an important role in the login screen redesign by allowing this feature it could improve responses towards this screen.		The implications of this will be large. While in general the re-work of the initial panels will probably be the majority of the post prototype, I will need to make sure that this feature works its way into the final system. An implication on the final system will also improve usability as it will be one less for the user to enter. For the redesign it will mean an extra hour or so making sure it looks well enough to include.
Joe	**21.**	Text field length	When looking at the demographic one area he felt that could be changed was the main address field length, with the idea that the value if short enough would recognise as an int or if longer would be converted to as a String.	While it is an ambitious idea and clever to think about, even from a technical standpoint completely doable, it would generate some confusion between users having to enter their house number and house name. If labelled incorrectly or I am unable to get this idea across simply enough the idea will just fail. Because of this I am going to have to reject the idea unfortunately		The implications of this are minor, all this means is that I will have to include a new field to accept the name or number, using validation to make sure at least one field has been entered. While this will take some time, I believe that the effort in doing this outweighs the pain in making sure every user of the system can enter their information to some degree.
Sue	**22.**	Account creation	Sue found the process of creating an account to be an easy task with most fields being clear on what data the system requires from her. One critique was that the text was a bit too hard to read and enter without having to look closer to make sure.	Due to the number of fields the user will have to enter for this panel I was somewhat confined to using a small area for data. However coming from the feedback the text was too small I could introduce a scroll pane and allow the fields to be a bit larger in size to allow data entry. I think as it will be doable I will accept the feedback.		The implications on the redesign are minimal at best, this will be down to just making the panel scrollable for the user. Nothing else besides rescaling text to a more readable font will be needed because of this, development time should be reduced. This as a result will help aid the user in creating an account and should allow for an easier use of the system in general.

Sue	**23.**	Patient homepage	While Sue was overall very impressed by the system in general, one area she found could use some more work on was the patient home screen, describing it to be "Pointless in the nicest way possible". While it does give her reassurance that she has logged into her account she feels there was little else to do on the page that couldn't be done anywhere else. To put it bluntly she would be happy seeing it go.	I feel that while somewhat true it's too important to greet the user on the launch of the system. While I can see that the panel provides little function most screens provide a welcome to the user that logs in. As I am going for a great patient user experience I have nothing to go against the screen and will reject the feedback and will still include the screen. Despite this to try and resolve the issue I will see to it that the panel gets some attention in the next section		The implications would be very large on the redesign as an entire panel will be moved and all the features associated uniquely will have to be given a new location on the system. This is too demanding on the redesign and would be too much for an issue felt by only one person. However while thinking about it I think I will also move the notifications to a larger part of the system and might reshuffle the layout of the homepage a little.
Sue	**24.**	Demographic formatting	Similar to sues issues with the account creation and the formatting of the text on the demographic she feels the text here was too small to be viewed by anyone without straining. And that some of the fields had an inconsistent size to the layout of text fields.	Again I will accept the feedback without much hesitation due to the fact that little is needed to actually fix the issue. While I feel that the issue isn't too much of an underlying problem of the system I think these minor issues overtime create an impression on the user so they do need to be addressed.		The implications on the redesign while initially at first thought where minor in order to make the text more readable the fields will need to be readjusted to allow room. While the panel as it currently has it doesn't have enough real estate to undertake these changes a scroll panel will need to be introduced to allow the changes.
Sue	**25.**	Admissions useless button	While inspecting the view admission panel she commented that the update admission button was pointless to have as it was already disabled, she recommended that it was just hidden from the patient completely	Again, a notable comment and that while this was just an aspect leftover during development I completely understand it is a feature that doesn't need to be seen because of this I will accept the feedback and will make sure that it is hidden in the final version		The implications on the redesign are minor it just mean that it won't be visible for the patient. For development it won't be too hard either as wherever the button is disabled I will just replace it with the visible function. Besides this there is nothing else to comment about as it is a small aspect anyway.

Sue	**26.**	Document attribute	As sue pointed out while an admission can be passed around between consultants she asked outside the action log how would the patient know from a glance who the document was made by, because of this in her own way she suggested to include an attribute about the author.	This aspect I will see included as it will be simple enough to include and it will allow me to include some more information about the document on the card, as if a staff entity brings over a document from the old system it will record that they were the ones who did it.		The implications of including this on the redesign are somewhat significant this will be due to the fact that development will mean that a new attribute to the object is needed and also that the field will need to be included on every area a document is present on the system. To add to this it will also mean that every current document will need to be updated to prevent null pointer exceptions
Sue	**27.**	Sorting documents	Here Sue felt that the documents have too little sorting attributes to find the document she wants to find, because of this shoe wants the ability to sort documents by other fields. For instance having the system group certain collections of documents together i.e. notes as one group.	While it will be possible to make these changes and have the chance to include extra sort options, the whole point was to allow faster access to documents while sorting is fine to have, searching will be the main option. Because of this I will reject the feedback		The implications of not including this is minor. While I could include every piece of feedback given to me it would just take as long as recreating the system as my time is finite to finish the system it will mean having to trade-off certain features, with this being an example of that, while the implications of this feature are minor.
Sue	**28.**	Abbreviating tab names	Another aspect Sue felt could be improved was the tabs of the system, while she felt they did what buttons are supposed to do she was unsure about the names and felt they shouldn't be shortened	Again I completely agree with Sue on the feedback and as a result will accept the feedback. While it isn't too much to ask of me it will mean the repositioning of the buttons to make sure they correctly fit in the box.		The implications of the redesign revolve around the relocating of the tabs to make sure that they include the new size of the text and any new aspects I may include based on other feedback. With respect to development it will be based around making sure that all the components are lined up with that being about it.
Sue	**29.**	Complete archive with GDPR	While it's not included yet sue thought that the ability to permanently delete an account without recovery should be an option after telling her about the option to "delete an account"	<See below>		The implications on the redesign would be minimal at first glance as it would mean that the field information would just be deleted not much to really design about it. However for actual use the files associated would have to be deleted including but not limited to: Personal Admission

	While for most other suggestions I have kept the response in the box here I wanted to go a little more in depth. While it is completely understandable that a user may not feel comfortable to know that their information may not have the ability to be permanently deleted, I think for the system having the ability to delete/archive an account completely is absurd. Obviously to comply with GDPR I have to show all the user's information that we hold about them, which I do, however if they want to access the account at any later point they would be unable to retrieve the information that was held due to it being deleted. Because of this I think the best compromise would be to disable the account and prevent access to it in any until the patient reenables the account.					
Sue	**30.**	User homepage button that takes them to their next appointment	Here when Sue looked at the homepage she saw the next appointment box for the user and asked how they would see the admission information. From this I struggled to explain that they would have to go to that specific admission like normally. From this she suggested that maybe a direct link to that admission may help solve the issue.	On reflection it does seem quite strange that the system informs the user of the admission the user is seeing next but then the user has to go through the effort of finding it to see the information about it. Because of this I will accept the feedback and will include it o the system		The implications of the redesign entail adding a button to the section indicating the next appointment and attaching it to load the admission panel showing the corresponding the correct admission. On development it will mean just figuring out which admission to load and assign the tabs and correct features to, it won't be too much of a stretch but it should work.
Sue	**31.**	Account credentials	On account creation she was unsure on the details she would need to login, she suggested that maybe some sort of window could be used to display the username and inform them of it when the account is created.	While the password is currently not included on the system I think the idea of having a popup window informing the user of their username is a great idea, because of this I will include it on the final version of the system		The implications of the design are somewhat substantial as their will need to be a new interface design. It will also mean that the panel/window will need to be linked to the process of creating account. One reflection this should work with the redesign set in place for the new start screen and reworks of the account creation.
Alex	**32.**	Navigation	On loading of the system, without much direction besides the aid of a document explicitly telling him how to access the systems features he came out and said that he found it really hard to navigate the system and find what he was looking for.	While I appreciate the honesty I can't see much to agree with. While yes it is a valid point to make, he however only had a brief few minutes to use the system before making the comment. I believe this to come from the idea that the document had little in the way of navigation and rather than him really struggling to move through the system on intuition.		The implications on the redesign if I were to follow through with Alex's feedback would be quite large as a large portion of the system would have to be overhauled to be made easier to navigate. While the design would take some time, actually implementing this on the system would take a much larger time.

People	Feedback number	Feature	Feedback	Consideration	Accept /Reject	Implications for Redesign
Alex	**33.**	Not clearing text field after logging out	While more of a bug than an issue here, Alex pointed out when the user logs out the field they entered to log back in is not removed allowing other users to log in as the last user.	Obviously this is a major flaw in the security of the system. Because of this without much thought as it doesn't need it, I will accept the feedback and make sure the issue is resolved way before it is finished on the system		The implications are minor to say the least, because of the issue at hand nothing needs to get redesigned but rather the issue needs to be patched before the product is tested. The issue around it is very serious however, the point of a login is to allow a user to leave the application on their account and allow others to use it. If this was not fixed this would seriously breach data protection.
Alex	**34.**	Admission tab contradicts the initial screen	Here Alex suggested that the use of the two admission boxes on the homepage contradict the purpose of the need for the tabs as they are more likely to be used so negates the actual implementation of the tabs. To sum it up he feels they are unneeded.	While somewhat linked to Sues idea that the homepage was little in features and bordered on the point on not even including it. Here I believe that the thought is correct as the button replaces an extra click and uses much of the precious real-estate of the home screen which could be utilised by more important aspects, because they are unlikely to be used I will accept the feedback and remove them from the current version of the system.		The implications on the redesign entail in the removal of the cards, it will also focus on the inclusion on new parts of the system to fill up that space. While in terms of development this will largely factor on what is eventually decided to be included. However some time will be relieved as the removal of these boxes will mean the code linking them to the corresponding admissions won't have to be developed.
Emily	**35.**	Primary key generation	When creating an account, and being told how the primary key generation works, she asked how do accounts with less than three characters in the surname have an ID. Obviously she has pointed out a flaw in the generation of an ID, however also suggested to solve the problem that being to repeat the last character until the length equals three.	The main reason I will accept the feedback is to increase inclusivity of the system, while there is no argument against including this on the system the main draw is that it will allow for more people to use the system without having to struggle to use it.		While nothing in terms of redesign will be needed, the only inclusion will be in development using an if statement checking that the field entry checks for the length being less than three and above zero, it should then hopefully iterate x many times until the length of three is reached. Obviously however this new version will not be used to save as a password but rather the username.

People	Feedback number	Feature	Feedback	Consideration	Accept /Reject	Implications for Redesign
Emily	**36.**	Button relocating for the login page	While only minor this feature is just about adjusting the layout of the login page. This is only to align the buttons so that they match the formatting of the rest of the system	In general I completely agree, when I eventually redesign these screens for the other feedback, I will make sure to take this into account because of this I will take this feedback on.		While the implications of the re-design of the buttons alone are small all it will be is just the relocating of the button to match the rest of the system. However as the entirety of the start screens are being readjusted the implications will be the quite large like previously stated.
Emily	**37.**	Transaction log	Here Emily suggested while a CMD is fine to show some text she believes that using a table to display the actions would be a more visually pleasing way to display the information to the management entity.	On hearing the idea, I was completely on board with the suggestion. From a technical standpoint while it will be possible it will need to be tested a little until it is fully developed. The only concern is that while some fields will have little data some will be paragraphs and may be awkward however I will manage to avoid it.		The implications for this will be somewhat complicated I will have to design a table that will be scrollable and the average size to fill in the fields of data. However a benefit for this will be that the number of actions will be known so by using a JScrollPane I should literally create the field and assign the values to the sign. To add to this a key point to make is that the frame will also have to be developed to hold the component in place.
Emily	**38.**	Remove dropdown box for blood type for updating the information	Here Emily suggested that while a dropdown box was convenient the user should have to enter the blood types as text to ensure that they are correct to avoid any errors. In addition to this once they are selected it shouldn't need to be changed if correct.	While in some respect I think Emily is correct as the patient should only have to enter the fields information once. However from a point of accuracy it needs to be editable from some point to prevent mistakes, and since it involves blood it needs to be correct so while not all of her suggestion will be included most will.		The implications of the redesign mean that the ability to enter the field for the patient will be only once for the patient, however the staff entity and consultant should always have the option. The implication for development means that the field should be disabled for editing for the patient but not for the rest of the entities.

People	Feedback number	Feature	Feedback	Consideration	Accept /Reject	Implications for Redesign
Emily	**39.**	If no changes disable save button	While currently the user can save the information file to for the demographic at any point, Emily suggests that from a practical point it would waste resources if a user could press it theoretically multiple time, to resolve this she suggests that its ca only be done once when a field is updated	While it may be a slight issue having to check that every field is the same when an event occurs I believe that overall it will appear better than if the user has the option to save the changes despite the fact that no changes have occurred. However as overall the benefits outweigh the draw backs I am going to accept the feedback		The implications on development will be minor to say the least. At most all will be needed is the ability to disable the button. However on development a lot will be needed to make sure that the information is correctly monitored. This will consist of checking every field to see if it updates and only then enabling the button to save, from this these new fields will be used as the fields determine whether the user can update the field.
Emily	**40.**	Remove popup for the error in searching	Similar to the no notifications on the user homepage Emily thinks that the popup should be moved to a label and displayed to the user as it seems more professional in her opinion.	While I think it could be useful in a way webpage errors display on the screen. I feel that a pop up would be better as it would just be quick and overall displays the same message. As it is already in place and there is no benefit to including it as a label I will decline the feedback		The implications on the redesign mean that for every search option the label would need to take the place of the popup window. Alongside this, the effort into replacing these windows for a feature that provides no extra benefit at this time would be pointless and a waste of resources.
Emily	**41.**	Improve search criteria	Here while she found that while the ID does work as a search term she thinks in general it should not be the predominant search term for any of the searches on the system for the user due to the likelihood of remembering these values is unlikely.	The idea to just improve the search term list, while a good suggestion is one that is just a result of the prototype stage of development. Nevertheless it will be an important feature of the system and I will make sure that it is included in the final version		The implications on the redesign minimal at best, very little would be needed to be redesigned. However in terms of development some effort will be needed to make sure the correct field is used in the query and in some searches the data type as Date may be useful for documents etc. Besides that very little else happens on the development of the system.

People	Feedback number	Feature	Feedback	Consideration	Accept /Reject	Implications for Redesign
Emily	**42.**	Archiving the consultant	Another feature Emily suggested was how would I respond to patient admissions if the consultant was archived and removed their account	While this would be inevitably thought about, while Emily has brought it up now means I can answer it sooner. The question is an important one and because of it I will accept the feedback due to fact that patients attached to the consultant will need to be dealt with. An obvious answer would be to remove the consultant as an option and move every patient under the admissions to a similar consultant.		While not essentially a redesign but more as a design the implications mean I will need to think about I will approach this idea and make sure that no errors are left or that not every area is left unsatisfied. The implications of development will mean that any entity attached to the consultant will have to be relocated. The best attempt to move every patient would be to look at the diagnosis and compare the consultants, however if a wrong guess is made the system does allow for readjustments. At this point it could even be a task given to the staff and for them to make the call on it and let them do it manually.
Emily	**43.**	Patients can self-discharge	Another feature that Emily suggested was that she thought like how in real life patients can self-discharge themselves whenever they want at hospitals, patients should also have the option to do that on my system.	While this idea did cross my mind at some point I felt that it wasn't worth it in the prototype. However while the idea has been voiced I feel that it is imperative that it is now included, this could be partially due to the self-submission of the admission, where a patient may feel that it is in their best interests that they discharge themselves for whatever reason. Because of this I am accepting the feedback		The implications will be fairly large. This will need to include a button to activate the feature alongside this a new panel will be needed to inform the user of the implications of their actions and finally a backend attribute that indicates that the dischargement was due to the user wanting to be discharged. For development it may also take a while to include as it could be much larger than anticipated.

People	Feedback number	Feature	Feedback	Consideration	Accept /Reject	Implications for Redesign
Joe	**44.**	Forgot my password	Here Joe suggested that a system might need to be in place for account recovery in case they forgot their login credentials, similar to how every other online system provides this feature. To add to this he suggested as the account details are very important so it would be wise to include some form of method accessing the account.	While I really liked the idea of having an automated email distribution system for emails having them be able to email out passwords, the main issue associated with this, after researching the idea I would have to download nonstandard java plugins such as activation and mail, while I want to use them they won't be allowed		The implications on the redesign could have also meant that other items could utilise the email, such as login credentials. To add to this instead of notifications emails could have been sent out with the information so the user could have constant updates about their account.

5.2 Input and Output

Considerations

After considering the feedback presented to me and consolidating what is to be included and omitted from the final version of the system. Now the input and outputs of the system need to be redesigned and planned out. An important aspect will be to make sure that none of the accepted feedback is missed and because of this the current designs will be reworked alongside the feedback making sure that every critique is resolved. To do this I will bring up the old Moqups designs and recreate them so that they meet the feedback suggested to me. For proof of adjustments I will compare the current screen used in the program if present otherwise the old design will be used as a substitution and show them against the new design alongside the suggestion. While not all screens actually need to be updated I will make sure that every screen gets a more accurate representation of what was achieved in the prototype, the main culprit of this being the screen size. While I would have liked to have a scalable size window I think at the current systems size it would just be too impractical with everything else being added and changed on the system, this will surely be a point to reflect on later on in the project.

House Style rework

Before the Input Output is started as a large critique of the system was that the colour scheme lacked any visual impression on the user, with comments suggesting it was too basic and serious for a patient. While at first in the design I was hesitant to create a too colourful system, I believe this fear of avoiding colour has become a determinant with the system now lacking any whatsoever. To resolve this issue I have added a few extra colours to the colour scheme. While in general the house style remains somewhat retained I have tried to add some more colour in the system. However to prevent the system from looking too informal and trivial, I have kept the original style but have given it a new look in some areas that needed it. However as the system still has time to see minor changes if I still feel dissatisfied before the final submission I will see any changes through the system in order to feel completely fulfilled with the system.

Overhauling the start screen

As the start screen has been reworked entirely a few changes have occurred, meaning the original structure has been altered to the point it hardly resembles the original system in any way. The large aspects are this:

- **There is no longer a symptom checker on the system the feature will now only be found in generating an admission**, meaning that only a login is available on the launch of the system
- There is a **onetime login and remember me aspect of the system**, the program should hopefully remember who was on last and will also **on the systems launch will open the homepage**. If not a welcome screen should be displayed and then followed by a login page.
- The user **specific logins will now be merged into one area** so that there is only on set of fields to enter credentials.

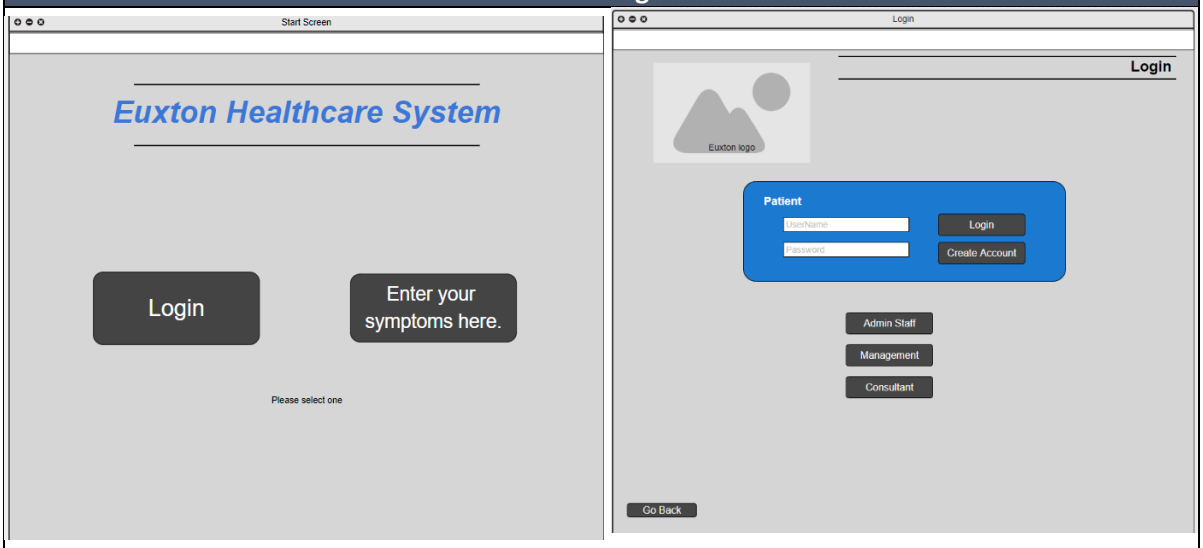
The reasons in doing this where due to the feedback and reflection on its practicality, when designing the login screen I weighed up the usefulness of the symptom entering part of the system and concluded the predominant userbase will be people with accounts and that maybe one in thirty people who start up the system may use the feature. Because of this I thought that an aspect hardly being used would not be worth the resources into developing. In addition to excluding that feature, also meant that the only part of the start of the system left was the login, and so following one of Phil's suggestions of a remember me feature, I could bypass the need of a start screen and could theoretically just load the homepage if the user who was on the system would never change.

Because of all of this the following changes like I stated where made. When a user starts the system the following should occur. If there is no remembered user then the welcome screen should be shown followed by the login screen from that the user either logs in or creates an account, from this it remains unchanged. If there is a remembered user they will be prompted for the password, which then allows them to login, from this like before operations remain unchanged.

5.21 New I/O Designs

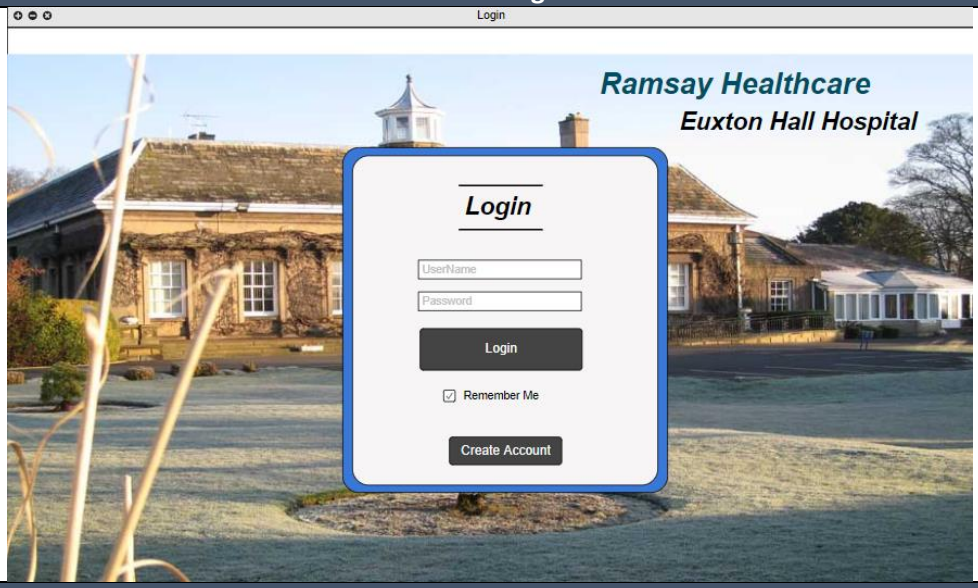
Start - login screen

Old design



Initial feedback: Buttons where originally in a counter intuitive layout, Formatting was bland, Slightly Confusing to login

New design



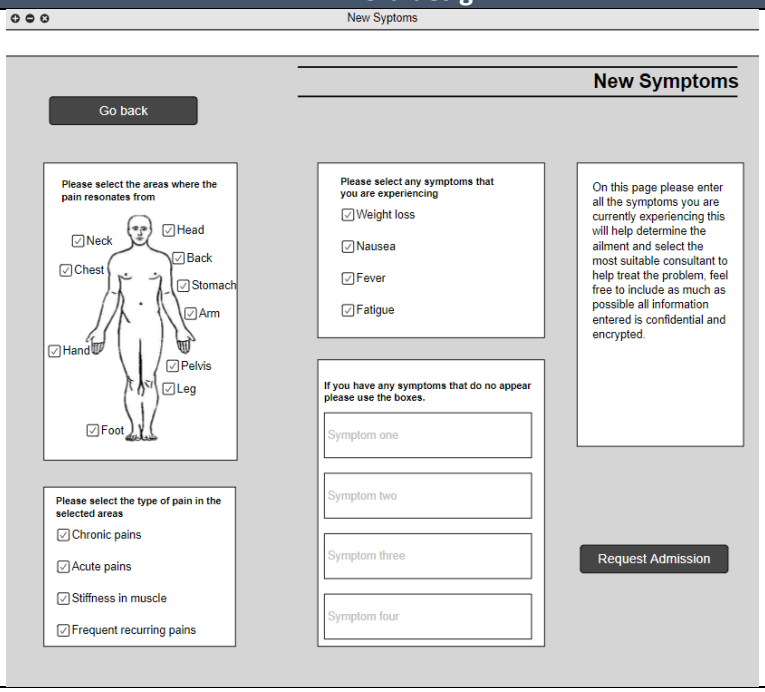
Consideration of redesign

The purpose of the redesign was to make the login screen more appealing to the user and more usable. A large inspiration was actually by the current Microsoft login screen with the large picture background which really avoids a bare solid colour background. The image on the back is actually of Euxton, however the ratio of the image as meant that the screen size is slightly adjusted to suit the image better a new image will be taken to fit the final screen ratio. Finally the gradient of the inner box is used to highlight the two text fields as it is a slight shade of darker white.

A large difference is that the start screen was reinvented to be a one-time deal until the user logs out.

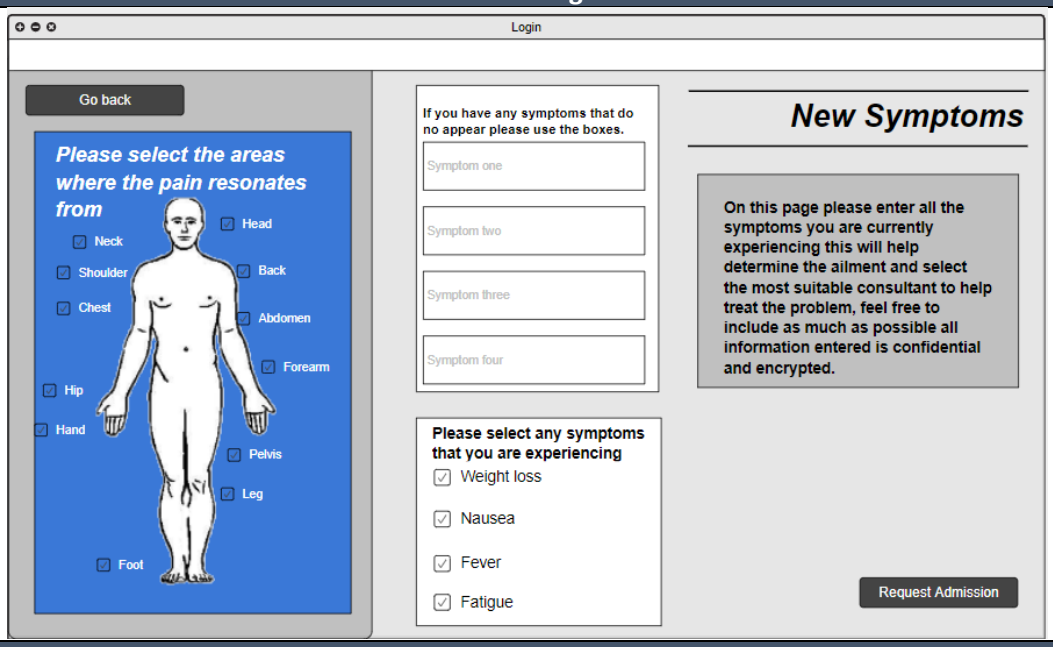
Symptom entry page

Old design



Initial feedback: Formatting was boring and bland

New design

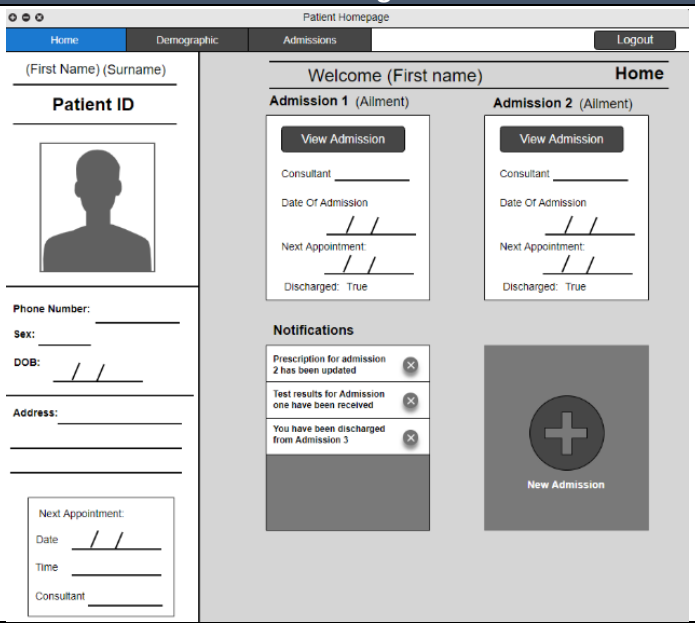


Consideration of redesign

For this panel, while little needed to be changed a few alterations in convenience and style where introduced. The only major change made was that the types of pain this was because not all symptoms may be painful because of this it wouldn't need to I be included as this can be asked by the consultant. In addition to this the text was made larger and bold to take advantage of the new screen ratio, this was a bid to make the screen easier to use. While the quick access symptoms could have been a drop-down box it would have mean that any other symptoms that they had seen wouldn't be selectable, because of that the checkboxes where included. While not visible the highlighting is another nice little detail to now include allowing for a more accurate description of what the user enters

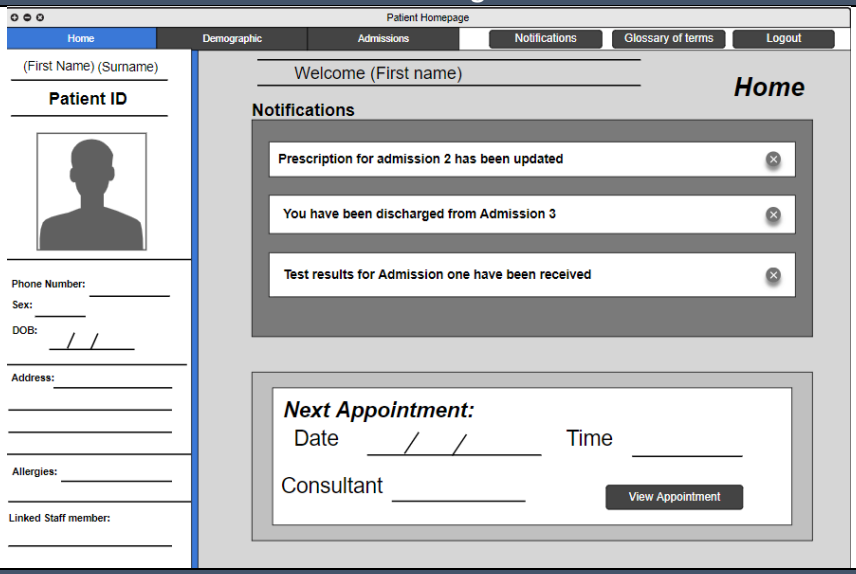
Patiet Homepage

Old design



Initial feedback: Somewhat featureless, all features are present at other sections of the system, defeats the point in using them

New design

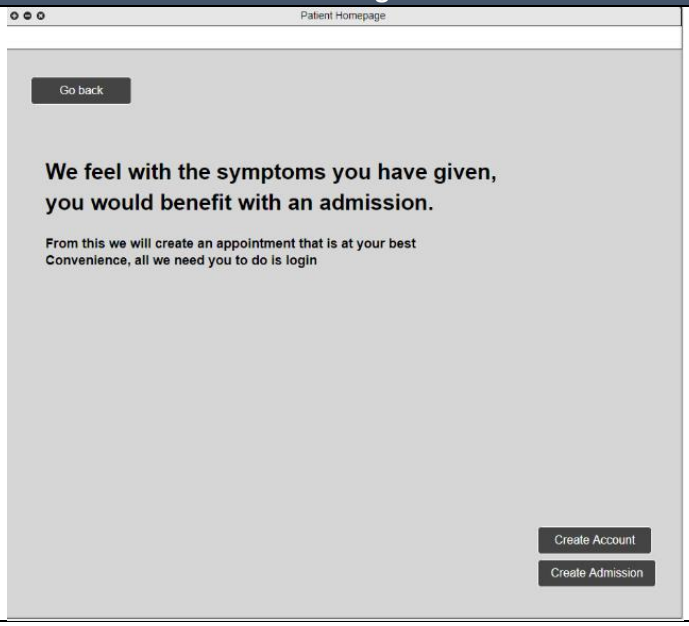


Consideration of redesign

The consideration of this redesign stemmed from Sues Comment about the homepage being pointless, along with my plans to remove the two admission cards. To add to this to fill more real estate on the panel the next appointment has been enlarged along with the notifications to make it feel more like a homepage. To add to this I followed Sues other comment, which was to include the View appointment button, here it should allow for a much easier navigation of the system. Finally as the contact panel used to include the next appointment box was removed, I had to fill in the information with other fields. These being the allergies of the user along with the linked staff member. I thought the last one would be a good field as it means the patient can have a good idea who handles their information other than the consultant.

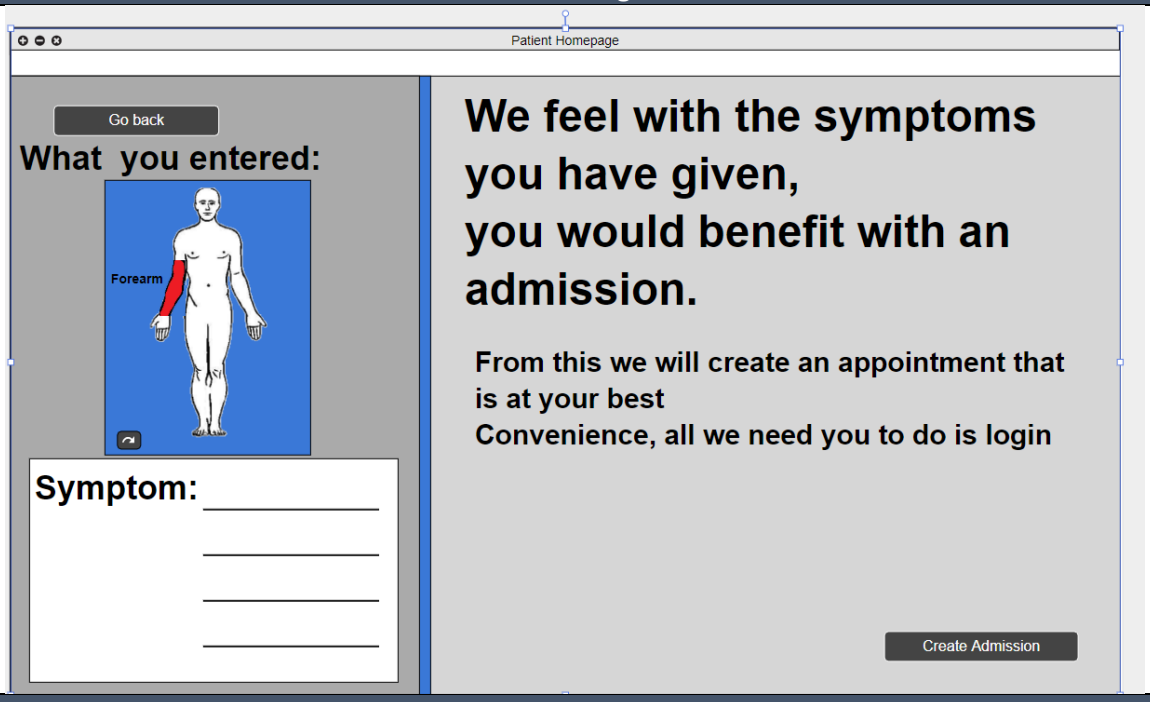
Symptom suggegstions

Old design



Initial feedback: N/A my own consideration

New design



Consideration of redesign

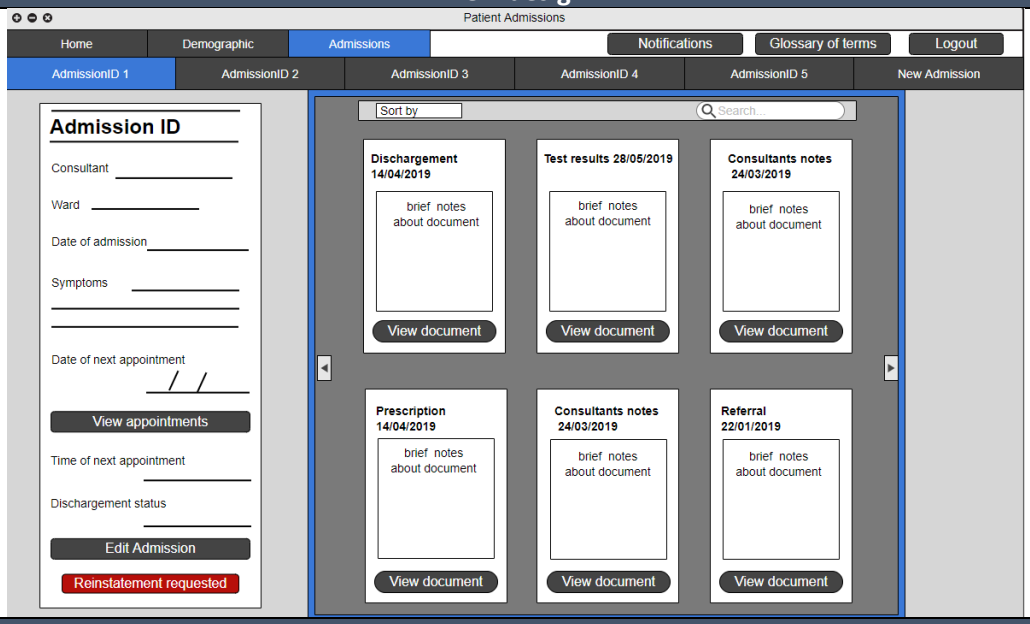
The point of this redesign wasn't given from anyone else but was my dissatisfaction of the screen. The main gripe I had was that the screen pretty much it had nothing to show but instead gave the patient the option to create an account or admission. The whole point of the panel was to either accept or decline the admission, before there where 4 possible outputs but now with their only being two I almost scrapped the screen entirely. However I thought similarly to how the chosen symptoms where shown in command line, I thought it could be shown like as a receipt of sorts in the GUI. After designing it now I feel really happy how it turned out and are now eager to have it included on the final system. This screen was inspired some aspects of the prior screen however no input on the symptoms can be entered here.

Admission Homepage

Old design

Initial feedback: The screen could use some more colour

New design

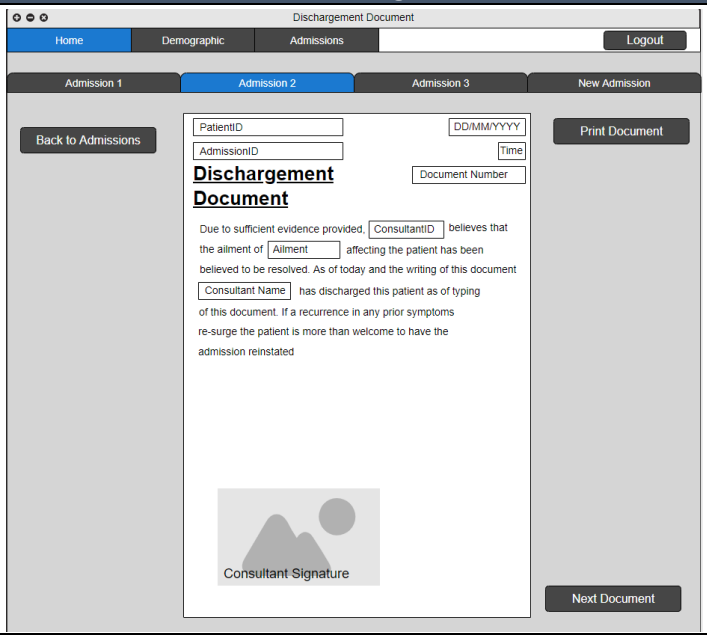


Consideration of redesign

While I did want to follow the feedback I said I would, and in some ways I believe I did. The aspect I really liked about the panel was that it had this formal and serious look to it. Because this, to and try please both parties I feel that I have reached a point in which it does this. Besides reformatting the size to include the new window ratio, the inclusions include the new buttons on the top tabbed bar, in addition to this the highlighting of the main part of the panel with the same blue to brighten up the page. Overall very little in the terms of new components has been added to the screen and so stays truthful to the original design. This is down to my overall original satisfaction with the panel and my gut feeling of it being close to it needing no changes.

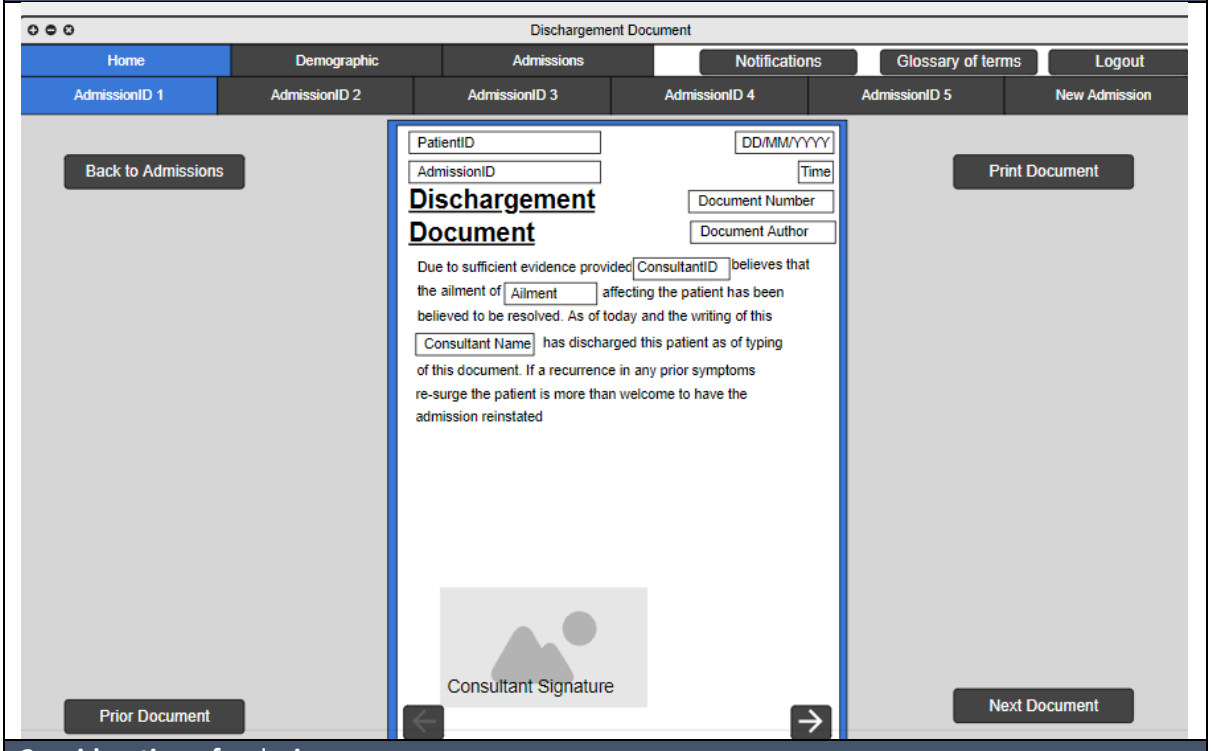
Viewing a document

Old design



Initial feedback: Multiple pages

New design

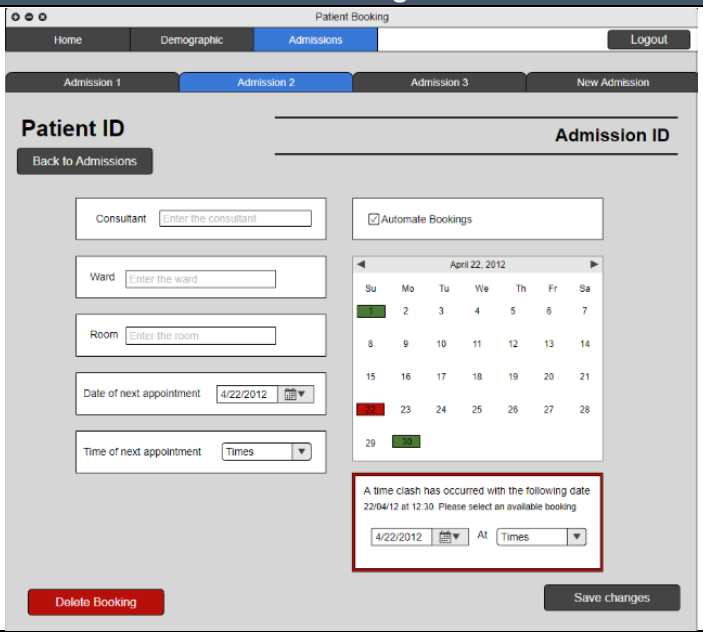


Consideration of redesign

Again, this is a screen that has seen little in the way in overhauling and because of this, looks pretty much identical to the original design. However some work has been done to improve and bring the panel up to current standards. This being that the tabbed bar now includes the suggested feedback given to me. The other inclusions predominantly being that the two buttons to move through the pages for the document now exist. To add to this a blue tinted highlight is given to the document to make it stand out against the background of the panel. Finally a field is now included indicating the author of the document, as suggested by one of my respondents. These changes are general for all types of documents, the discharge document was an example.

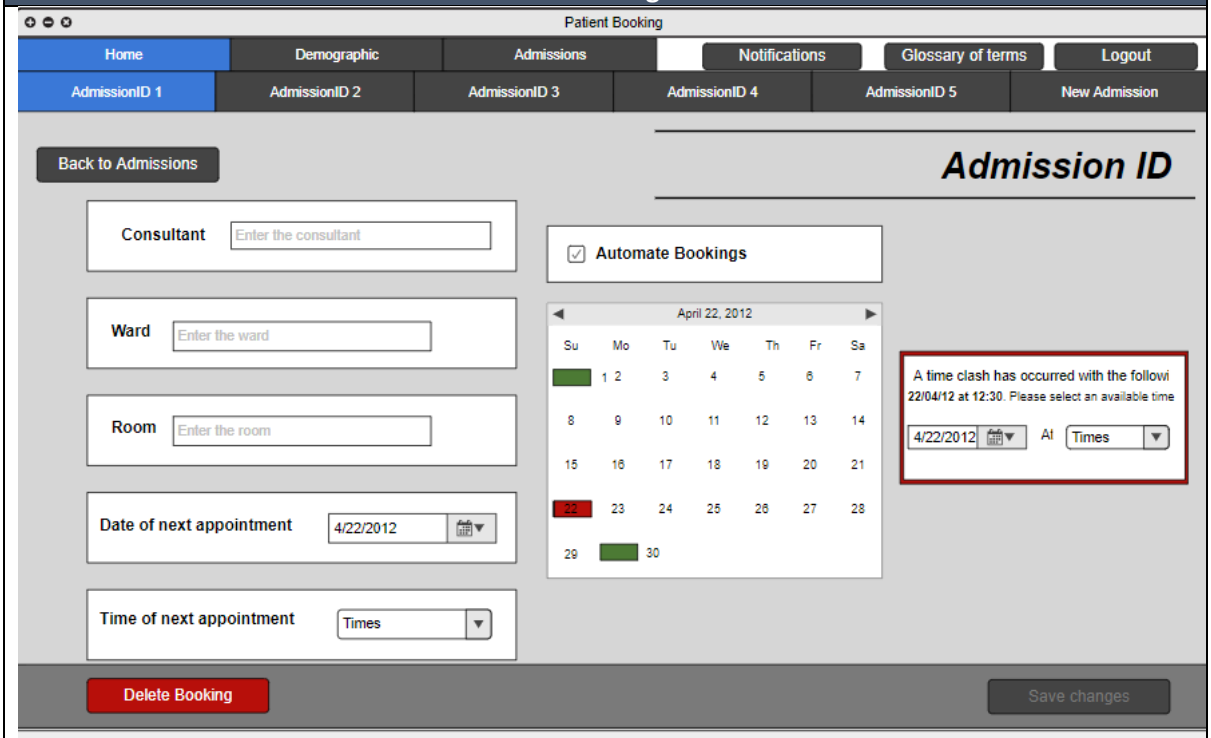
Bookings

Old design



Initial feedback: N/A

New design

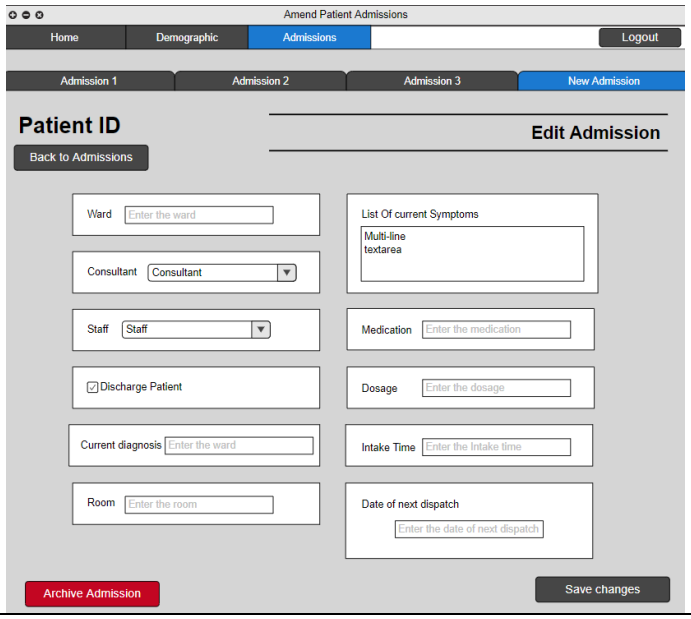


Consideration of redesign

Nothing has been done to overhaul the panel, this because while most where created in the prototype this one wasn't, however as there have been a few changes to how the system works some inclusions and alterations have occurred. For example the top bar has been the majority of the screen redesign including the layout of the screen. The inclusion of the two buttons is probably the best example of this with these being new additions coming from the feedback suggested by others. While my prototype was lacking a data picker in general this was because of the booking being missing but now with it back in the plan the data picker should hopefully see its inclusion in the final system.

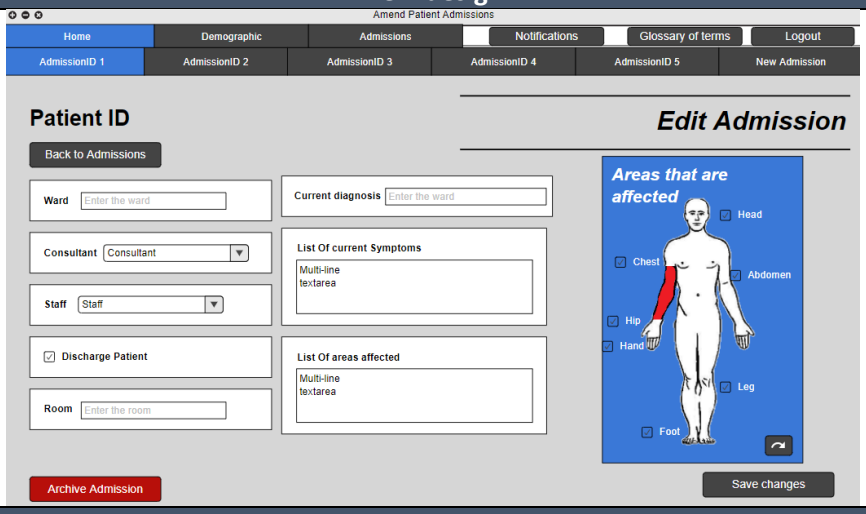
Veiwing/Editing the Admission

Old design



Initial feedback: N/A

New design

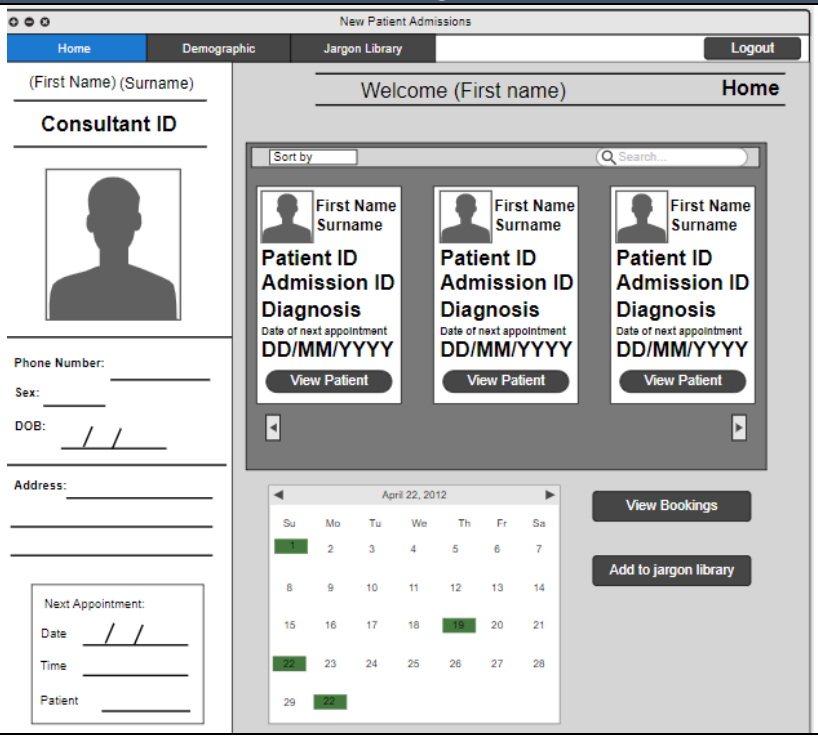


Consideration of redesign

Nothing has been done to overhaul the panel, this because while most where created in the prototype this one wasn't, however as there have been a few changes to how the system works, some inclusions and alterations have occurred. For example the top bar has been the majority of the screen redesign including the layout of the screen. The inclusion of the two buttons is probably the best example of this with these being new additions coming from the feedback suggested by others. One area that has received is format following the layout that was used in the prototype and praised by in the feedback. A last minute inclusion was the idea of utilising the affected areas of the system. To improve the visibility and have a clearer understanding in which areas where affected the body has been included along with a highlighted section to imitate what it will look like in the end product. To add to this the ability to flip the silhouette will also be available also showing the rest of the options to the user, this will be available in the symptoms screen also.

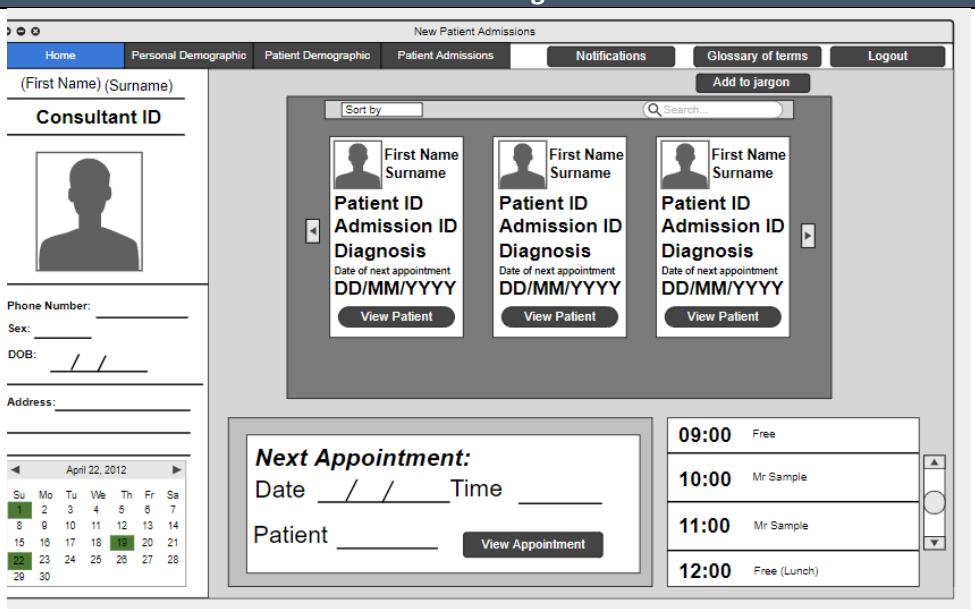
Consultant Homepage

Old design



Initial feedback: Include a view admission button

New design

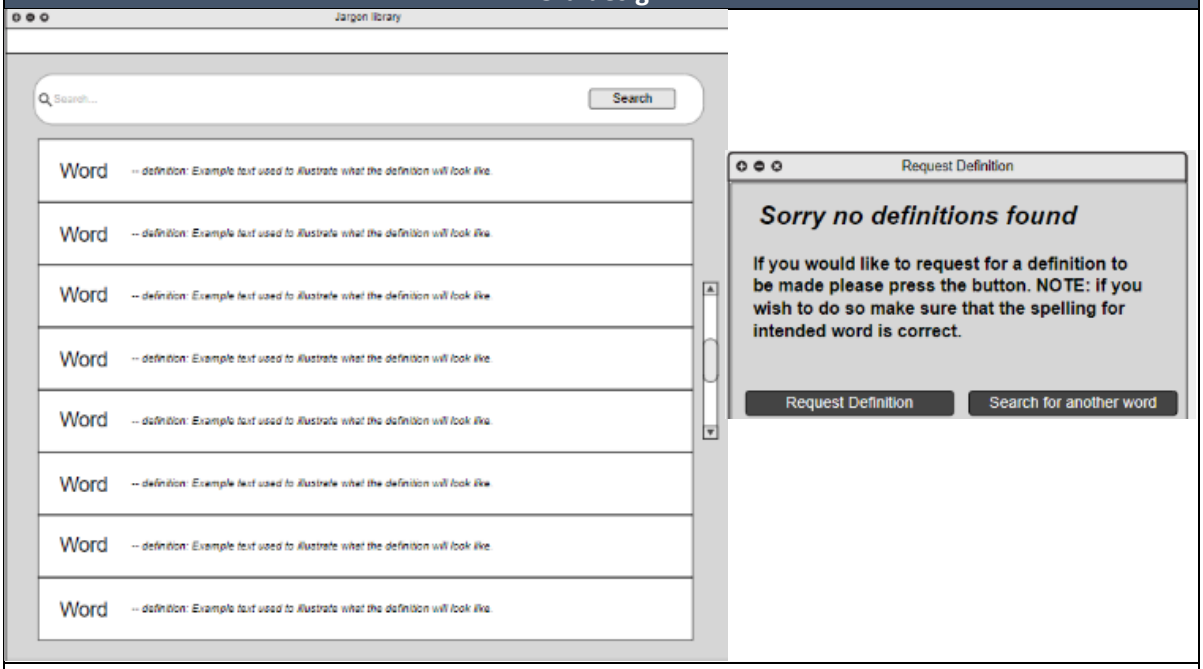


Consideration of redesign

While this screen could have remained the same, I felt that as the patient's homepage is getting reworked slightly, this panel might as well use some attention. Because of this like every other panel the generalised actions have been made. These being the new window aspect ratio alongside the tab bar buttons. However the main redesign is the flipping of the calendar and the next appointment box. This was done as it was suggested that the idea of the next appointment box was well received and while the calendar is a nice thought the next appointment would be viewed much more, so naturally I made it bigger. Finally to add to this I also included the view button as suggested by Sue along with a day planner as it may come as in much use as the other feature.

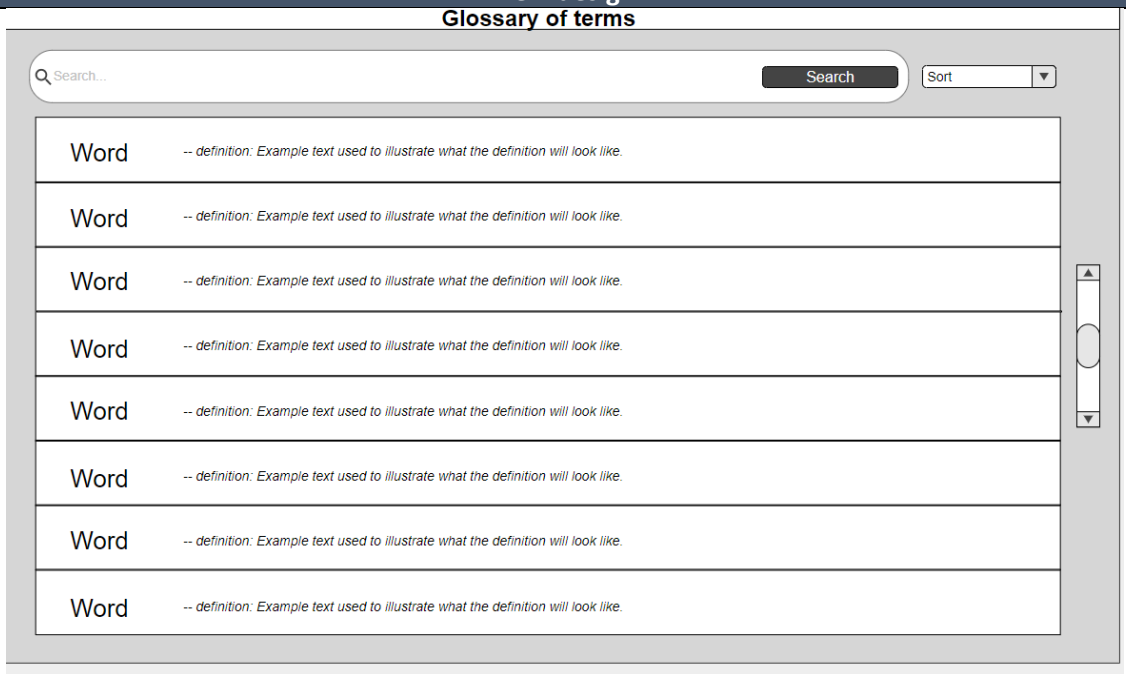
Glossary of terms

Old design



Initial feedback: N/A

New design

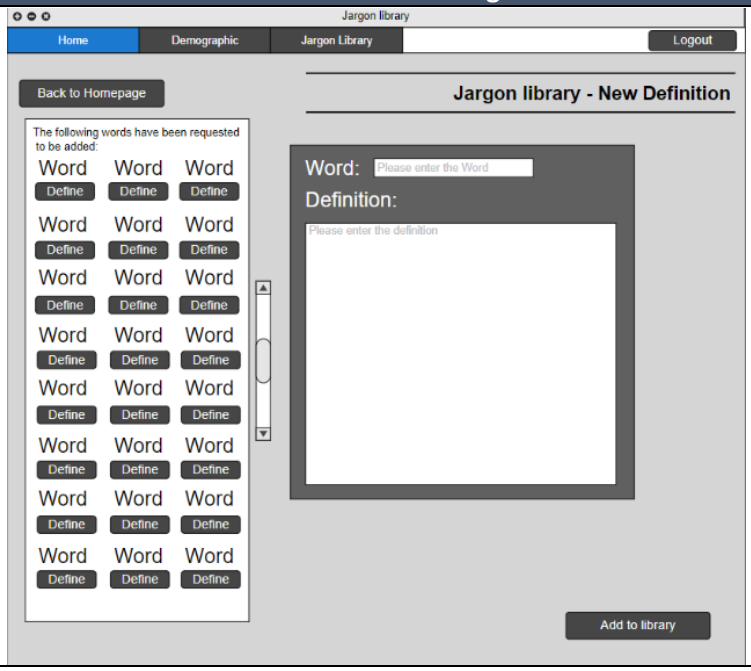


Consideration of redesign

Besides updating the screen to include the new Tabbed bar and fit the new window screen size nothing else too significant was included on the redesign. This is because it will be on a new window so having to make it look appealing, while obviously a priority is not a main focus, with that being having the panel on a new window. One thing that has changed however is that the pop box has now been removed and instead will appear of the location of the definitions.

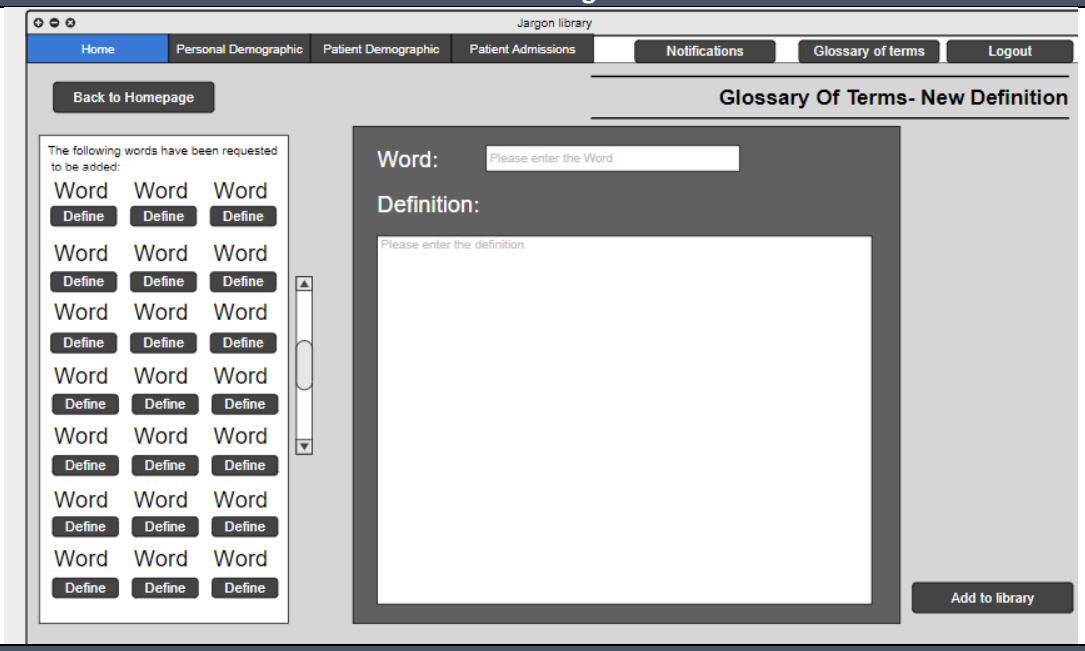
Adding to the glossary of terms

Old design



Initial feedback: N/A

New design

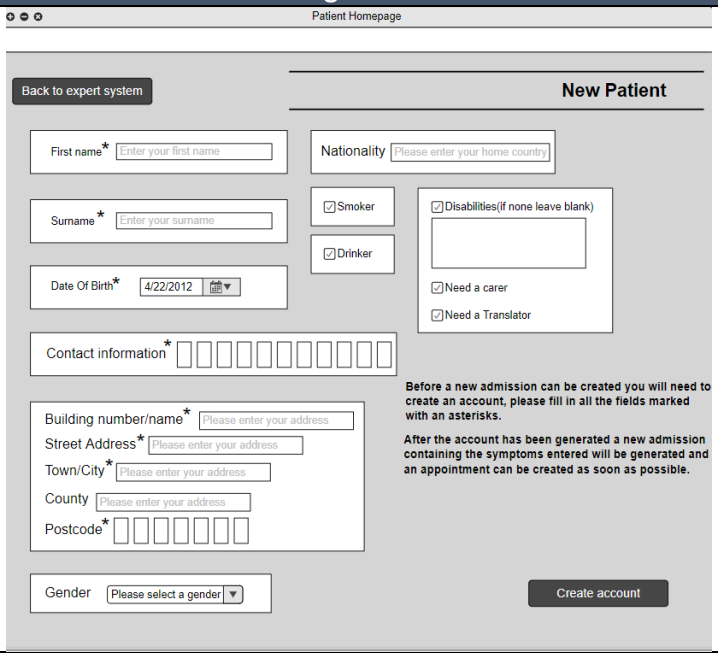


Consideration of redesign

Nothing has been done to overhaul the panel, this is because while most where created in the prototype this one wasn't, however as there have been a few changes to how the system works some inclusions and alterations have occurred. For example the top bar has been the majority of the screen redesign including the layout of the screen. The inclusion of the two buttons is probably the best example of this with these being new additions coming from the feedback suggested by others. Finally some work has been done with the tab buttons now including account information for the staff along with the same tabs included for the patient.

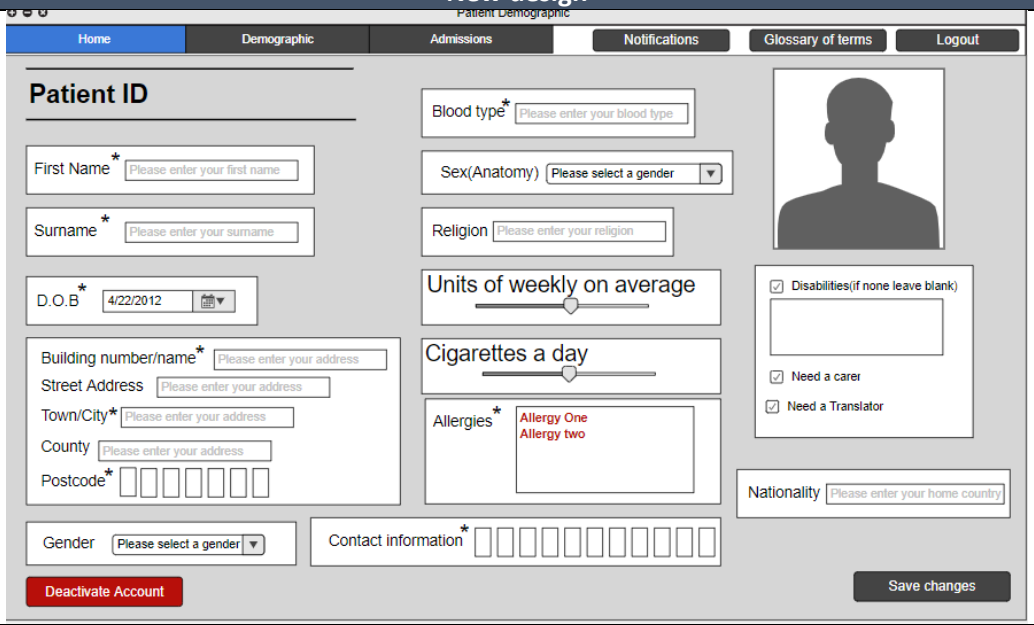
User Demogrpahic

Old design



Initial feedback: The binary state fields should be slider values

New design

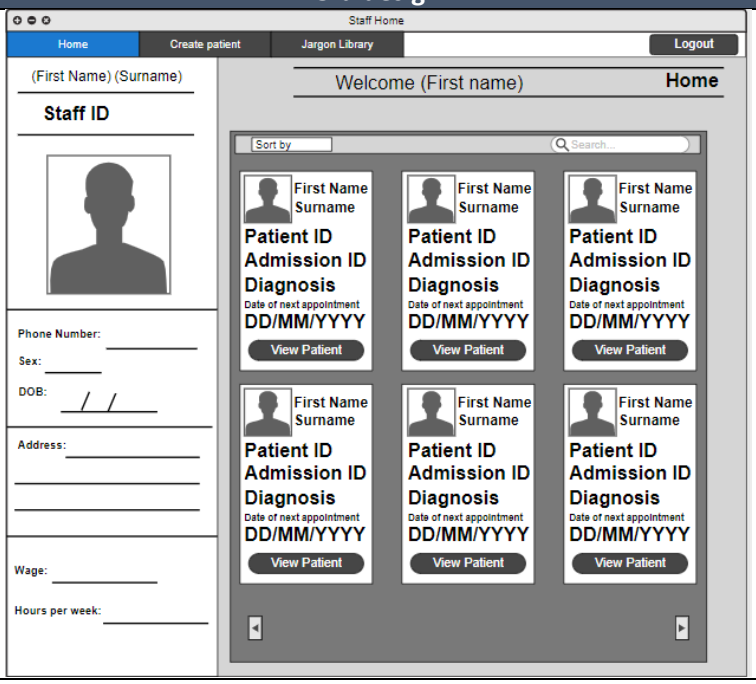


Consideration of redesign

Besides the login screen, this panel received the most attention despite being one of the least criticised aspects of the system. This was because the screen was actually updated before now, during the development of the prototype and because of this some alterations where made but never fully documented. The main being the new layout of the fields. However one aspect that was brought up by the respondents was that the contact and address fields need to be individual fields. As you can see that they are already like this in the old design but where left out in the prototype. Because of this I will make sure that these are continued through onto the new system. In addition to that the two Boolean values have now been replaced with sliders.

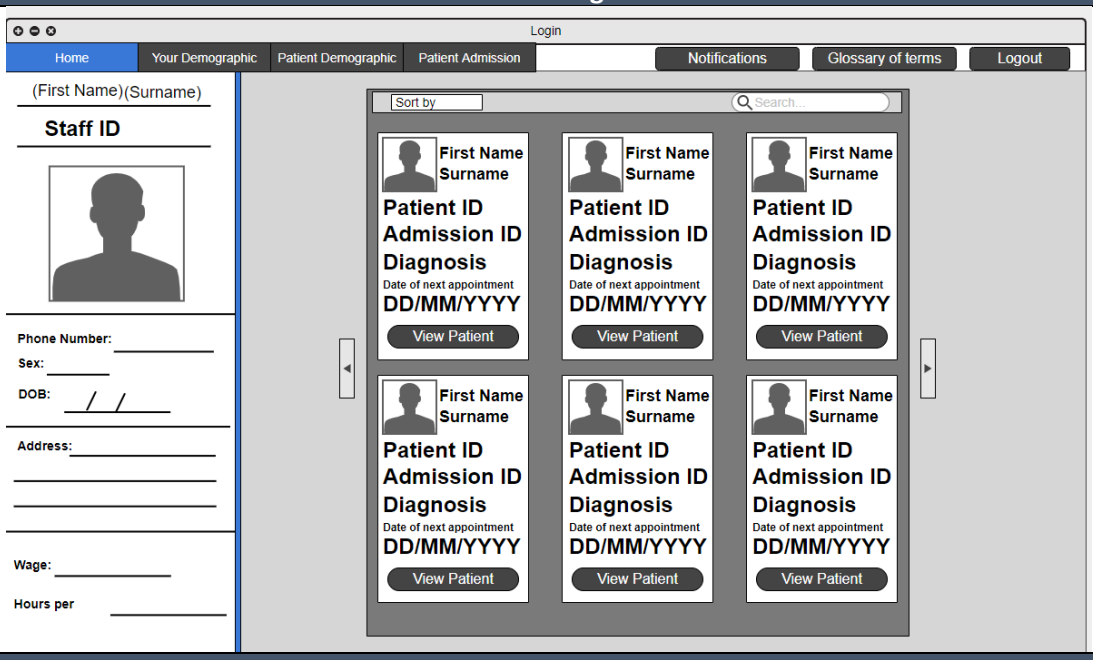
Staff Homepage

Old design



Initial feedback: N/A

New design

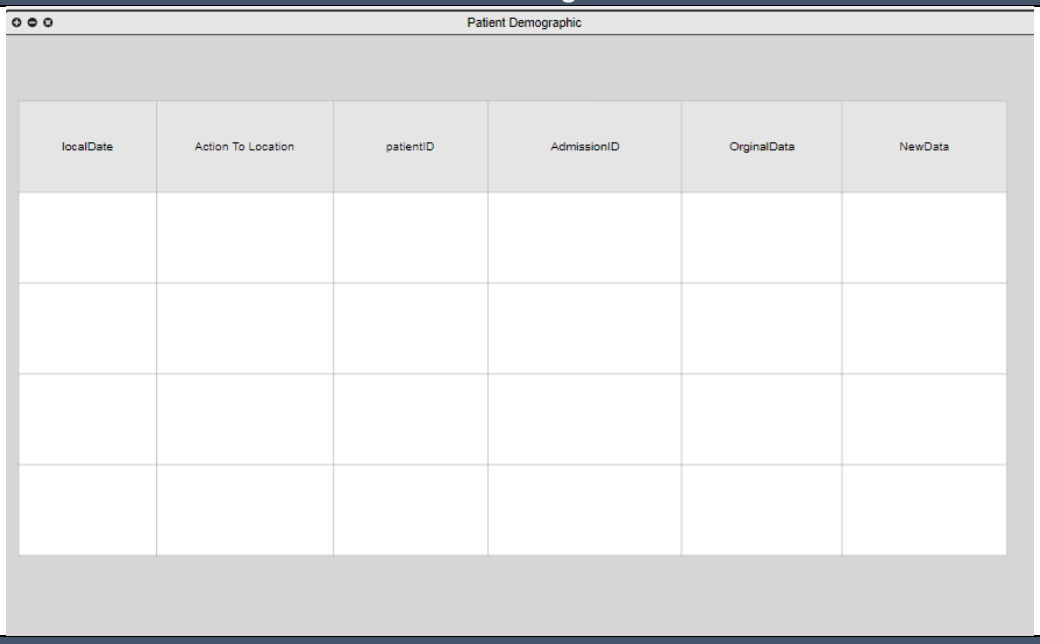


Consideration of redesign

Nothing has been done to overhaul the panel, this because while most where created in the prototype this one wasn't, however as there have been a few changes to how the system works some inclusions and alterations have occurred. For example the top bar has been the majority of the screen redesign including the layout of the screen. The inclusion of the two buttons is probably the best example of this with these being new additions coming from the feedback suggested by others. Finally some work has been done with the tab buttons now including account information for the staff along with the same tabs included for the patient. However on inspecting a patient's admission the consultant will also be able to add old documents to the admission by just selecting the pdf they want to attach.

Action log tabel

New design

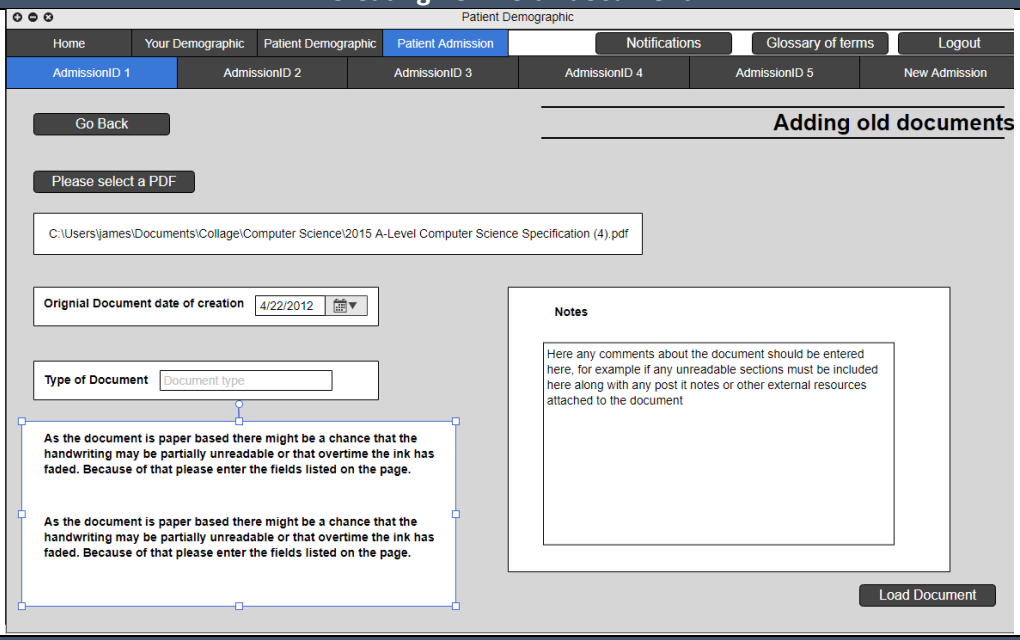


Consideration of redesign

Following Emily's direct suggestion to include a table the following table was made to have a quick idea on how it will look. The reason it was given little time in design was due to the fact on the final version a table would be produced in a window with no links. Maybe the idea to print of the actions could be quite the unique idea to have included, A benefit of this of course being that the table being the only component on the window, nothing would be needed to only print the table. However a thing to point out would be the eventual need for a JScrollpane. This will be due to the fact that there will be so many entries it will mean that the table will leave the viewable range for the user and so a scroll pane will be needed. There are no tabbed bars due to the fact this entity is solely based in command line and has no other GUI other than this, because of this there is no need to have a bar to attach links to so none was included.

Action log tabel

Creating new “old” document



Consideration of redesign

While not a redesign it is a screen I forgot to include on the design and after considering the need for such a screen it was obvious one would need to be designed. While not visually amazing it doesn’t need to be as it will only be used by the staff entity. The function is to allow the staff entity to select the pdf document they want attached to the patient’s admission. The main focus is to allow all the correct information to be brought over also. As some documents have degraded overtime or the fact that handwriting may become an issue. The current attributes already in place can be used to store information in the notes field for this field the staff entity will try to make sure any sections that may be hard to read are noted in this section.

5.3 Data Structures and Files

Considerations

Now that the I/O has been reworked I can focus on the backend of the program; this is where most of the system will see changes overall. The purpose of this section will be to rework and overhaul the current poorly designed file system currently in place. While it currently gets the job done and achieves everything it needs to, it is no way a perfect method to dealing with data and in order to become more efficient and effective with file management a change is surely needed. While data structures may experience slight tweaking in order to accommodate changes in regard to the newly designed I/O screens, overall I believe the object-oriented class diagrams to be a highlight of the system and I would consider it a waste of resources to spend too long adjusting them. The section that really needs to see change is the data storage in documents. This is mainly down to the fact that data in some places is on a per admission basis. While convenient when accessing a few lines of data for an account, overtime the folder containing the data has become a mess and no user would tolerate so many files. It was clear from the offset of development that the change would come into fruition, however it is now time to go through with it.

5.31 Data Structures

While I said that all current data structures seem to be fine, it doesn't mean that nothing new has occurred. The statement was made due to the fact the changes will be minor in comparison to the fact the entire data storage system is being redesigned. One data structure that will see some attention is the employees due to the fact that the entity will now retain all the attributes possessed by the demographic of the patient. However other fields such as the primary key will be kept the same due to the fact the code uniquely generates correct values. while currently the generation works fine the code will have to be changed to accommodate the new changes but the format and current designs for said fields will be the same.

To avoid creating extra unneeded work for myself, all list classes will not need to have their data structures changed, because their function is to store information regarding the object there wouldn't be any new attributes that the single class wouldn't have. To add to this the objects list classes actually extends from the original so no need to worry about retaining any changes.

Validation of current attributes

Before the new data structures are shown here I wanted to briefly explain the lack of validation on some variables some would want validated. As not all attributes are entered manually some through hardcoding and other means, to minimise unnecessary actions on the system any automatically generated attribute like current date time or static attributes like the name of the hospital will not be validated simply because there is no need to.

The need for new data

Overall as there have been alterations to the original entities, they have gained quite a few attributes. This is partially due to the fact during development they where needed in order for certain processes to happen for instance the inclusion of the number of admissions allowed for the correct amount to be retrieved from file. On the other hand the rest of the new data will be due to the changes and redesigns planned in this document, the main reason they will be included is because they will overall play an important enough role in the final system and will be needed for every object of that entity.

User

Overall this entity has seen a lot of changes and inclusions to it. Besides the moving over of pre-existing attributes some new ones suggested in feedback such as Joe's idea about sexual genitals and gender [18]. The main changes being that some of the fields have been altered to accept more than a few characters and for gender the character data type has been changed to as string so that the conventional F/M will still apply but the user can also now enter other gender identifications. While I was somewhat hesitant to include said attributes to said class the result will mean that every user will now retain a fully updated demographic and so will have no data redundancy issues. While the file size will be much larger for each entity because of this I feel that the quality of life improvement made here will make the system seem fuller to other users other than the patient, who in reality are the predominant userbase as it will be them who use the program every day.

Field Name	Key Field	Data Type	Length	Example Data	Validate	Same? New? Altered? Removed
firstName	-	*String*	*>0*	*James*		
surName	-	*String*	*>0*	*Nurdin*		
gender	-	*String*	*>0*	M		
dob	-	LocalDate	10	28/05/2002		
houseNum	-	*Integer*	*3*	*167*		
houseStreet	-	String	<25	Town Road		
Town	-	String	<20	Wigan		
Postcode	-	String	7	PR26 9RA		
contactNumber	-	String	11	07484727992		
nationality	-	String	<25	English		*
smoker	-	Boolean	1	0		*
drinker	-	Boolean	1	1		*
county	-	String	<25	Lancashire		
disability	-	String	1	0		*
Carer	-	Boolean	1	1		*
Translator	-	Boolean	1	0		*
Password	-	String	>6	Dosed12!		
DaysSinceLastUpdate	-	*Integer*	*>0*	*90*		
Sex	-	Character	1	M		

* These asterixis indicate that the fields while new for the entity actually stem from the objects' child class of a patient however following Karl's feedback[6] I have moved the information to the user entity meaning all users will retain the attributes on the system.

Employee

I removed the two attributes here because it would have been a nightmare in the long run, only having one admissions assigned to a patient is crazy, to fix the issue the attribute was slightly changed and given to the patient so that all admissions are looked over by one staff member but the staff can theoretically have as many patients as needed. Also as the employee class is a parent class to the consultant also it would additionally mean that the consultant too would inherit the attribute which is something we do not want. Other than this I see no issue with the entity as the data present seems to qualify for what the staff will need to do on the system. On thing to mention is the lack of validation.

Field Name	Key Field	Data Type	Length	Example Data	Validate?	Same? New? Altered? Removed
employeeID	*PK*	*String*	*11*	*ENUR1234567*		
wagePerHour	-	*Real*	*4 (2 int,2dp)*	*12.80*		
hoursPerWeek	-	Integer	2	42		
currentAction	-	*String*	>0	Created Document		
DateActionPerformed	-	*Date*	8	12/04/2019		
PatientInvloved	-	*String*	>0	PNUR0000001		
AdmissionInvolved	-	*String*	>0	ANUR0000001		
newData	-	*String*	>0	*"Test results look good no worries"*		
oldData	-	*String*	>0	n/a		

Staff

For this entity again very little was needed added to it due to the fact that the entity wasn't present in the prototype. To add to this the staff's parent classes employee and user also retain the majority of the entities attributes so it is unneeded to add any new attributes to just this entity, when others would need it too. Overall while on the attribute side while most of it is inherited the majority of the entity will reside in functions and methods that can be performed. The number of patient's field is for the generation of new patients so that staff entities are not over encumbered when new patients are made and that an even distribution occurs between staff. Finally there are little attributes simply because what attributes attained are kept in their parent classes and so following my hierarchical class diagram the most efficient way to store data will be to follow that structure.

Field Name	Key Field	Data Type	Length	Example Data	Validate?	Same? New? Altered? Removed
staffID	*PK*	*String*	*11*	*SNUR1234567*		
numberOfPatients	-	*Integer*	*>0*	*21*		
archived	-	Boolean	1	0		

Consultant

Again this is another entity I feel needs no additional amendments or inclusions to there seems to be enough attributes currently going for them I think there is no need to include any more. Since the patient demographic attributes have been added to the user they should now also have the ability to view their demographic information, which is a good feature to add, I think the lack of attributes was just a biproduct of the prototype as I had seemed to have planned a lot more in design. The only change that has been made is since now these fields will be amendable they will need to all be validate and so I made sure that they will be.

Field Name	Key Field	Data Type	Length	Example Data	Validate	Same? New? Altered? Removed
consultantID	*PK*	*String*	*11*	*CNUR1234567*		
ward	-	*String*	*<15*	*Cardiology unit*		
practisesList[]	-	String	>1	Anesthesiology,Dermatology		
nextPatient	-	Object		PNUR1234567 James Nurdin ANUR0000001		
timeOfNextApp	-	LocalTime	4	17:18		
dateOfNextApp	-	LocalDate	10	DD/MM/YYYY		
archived	-	Boolean	1	0		
numOfPatients	-	Integer	3-2	99		
listOfAppointments	-	Object		PNUR1234567 17:18 DD/MM/YYYY R206		

Management

With management again this is an entity that within the prototype had very little attributes assigned to them. However as their use on the system has become more potent it has come to my attention that they need to be fully fleshed out. Again similar to staff the attributes are associate with their parent class and because of this it appears that the entity itself seems underwhelming when this is not the case as the attributes are inherited.

Field Name	Key Field	Data Type	Length	Example Data	Validate	Same? New? Altered? Removed
managmentID	*Pk*	*String*	*11*	*MNUR1234567*		
archived	-	Boolean	1	0		

Patient

Similar to the user entity the attributes here that have been deleted have been moved to their parent class to allow other users to access the attributes while, some development will be needed so that any current instances will still work the changeover should be fine. As you can see on the last newly added attribute the staff involved has been included after being removed from the employee object this was done to make it easier on deciding which employee receives which patient. The final attribute notification will be for any new events that have happened being recorded to the patient's account.

Field Name	Key Field	Data Type	Length	Example Data	Validate	Same? New? Altered? Removed
patientID	*Pk*	*String*	*11*	*PNUR1234567*		
nationality	-	String	<25	English		
bloodtype	-	String	1-2	O+		
smoker	-	Boolean	1	0		
drinker	-	Boolean	1	1		
disability	-	String	1	0		
numOfAdmissions	-	Integer	2	5		
Carer	-	Boolean	1	1		
Translator	-	Boolean	1	0		
numberOfNotifications	-	Int	Any positive int	12		
notifications	-	String[]	N/A	Appointment Was Set to 19/05/2019		
StaffInvolved	-	String	<25	*ENUR1234567*		
notification						

Admission

In regard to the rest of the attributes I have decided to finally sort out what is going on with the prescription with it finally being declared as a document type and therefore have all remnants here removed before development takes place. To add to this the ward attribute was changed to make that the field validated as it will be important. Because there has been a big change towards the caring of admissions here I would like to address it. Currently in the prototype the admission has two names of staff associated with them, them being the staff and consultant. The staff's name is being removed and placed in the patient object as a staff will care for a collection of patient so it would be more efficient to have the same name stored once instead of multiple instances. The final new attribute will now account for all the areas affected for the user. While it was disregarded initially as it would only be used in symptom diagnosis, now I feel that the information could be used to help display what areas are affected for the consultant and staff.

Field Name	Key Field	Data Type	Length	Example Data	Validate	Same? New? Altered? Removed
admissionId	*PK*	*String*	*11*	*ANUR0000001*		
ward	-	*String*	*<15*	*Cardiology unit*		
consultantID	FK	*String*	*11*	*CNUR1234567*		
staffID	FK	*String*	*11*	*SNUR1234567*		
timeOfNextApp	-	LocalTime	4	17:18		*
dateOfNextApp	-	LocalDate	10	22/05/2018		*
active	-	Boolean	1	1		
numOfDocuments	-	Integer	2-3	12		
medication	-	String	>20	Azithromycin		
dosage	-	String	5	10.0mg		
intakeTime	-	LocalTime	4	17:18		
dateOfNextDispatch	-	LocalDate	10	22/05/2018		
listOfSymptoms[]	-	Sting	>1	Excessive hunger Weight Gain Fatigue Nausea		
currentDiagnosis	-	String	>15	Diabetes type 2		
archived	-	Boolean	1	0		
room	-	String	4	P001		*
AreasAffected[]	-	String	N/A	Chest Abdomen		

* With this attribute as the importance of modular programming as become ever more apparent to me, right now a new object has been decided to be included on the system. That being booking. See more later on.

Document

For this entity some changes have been made, firstly the prescription attributes have been altered so that they are validated, I don't understand why they weren't in the first place just happy to see it now. In addition to this two new attributes have been included. The first one indicating to the system which type of document it is as seen in the prototype, and also the author of the document as suggested by Sue [26]. Besides this very little has been included to the entity. However one major inclusion being an idea of my own, that being multipage documents, while the feature won't be rampant throughout, if for instance the text box for notes is full a new document will be made.

Field Name	Key Field	Data Type	Length	Example Data	Validate	Same? New? Altered? Removed
documentID	*Pk*	*String*	*11*	*DNUR1234567*		
timeOfDocument	-	Time	4	17:18		
dateOfDocument	-	Date	*10*	22/05/2018		
consultantNotes	-	String	N/A	MR Jones failed to attend the consultation today, I will postpone is surgery to a later date until he visits again to discuss final preparations		
Hospital	-	String	N/A	Euxton Hall		
testResults	-	String	N/A	Ammonia concentration 0.5% etc		
Medication	-	String	N/A	Paracetamol		
dosage	-	String	3/4	5mg		
IntakeTime	-	String	N/A	09:00		
dateOfNextDispatch	-	Date	10	22/05/2018		
DocumentType	-	String	N/A	Old Document		
Authorofdocument	-	String	N/A	Dr Smith		
testResults	-	String	N/A	Inconclusive		
AbsolutePath	-	String	N/A	C:\Users\james\Documents\Collage\Computer Science\2015 A-Level Computer Science Specification (4).pdf		
numberOfPages	-	Int	>0	2		

(New Entity) Booking

As this entity is new I have included the key attributes that will be associated it before the class diagram is redone. The attributes here are to show the information regarding the booking, in addition to this to uniquely identify every booking on their own instead of utilising a compound key like previously thought in the prototype, a new primary key can be used which will be created using the same conventions as previously described before. While the data isn't necessarily new the entity holding it is and so I would like to think that the data is because of it.

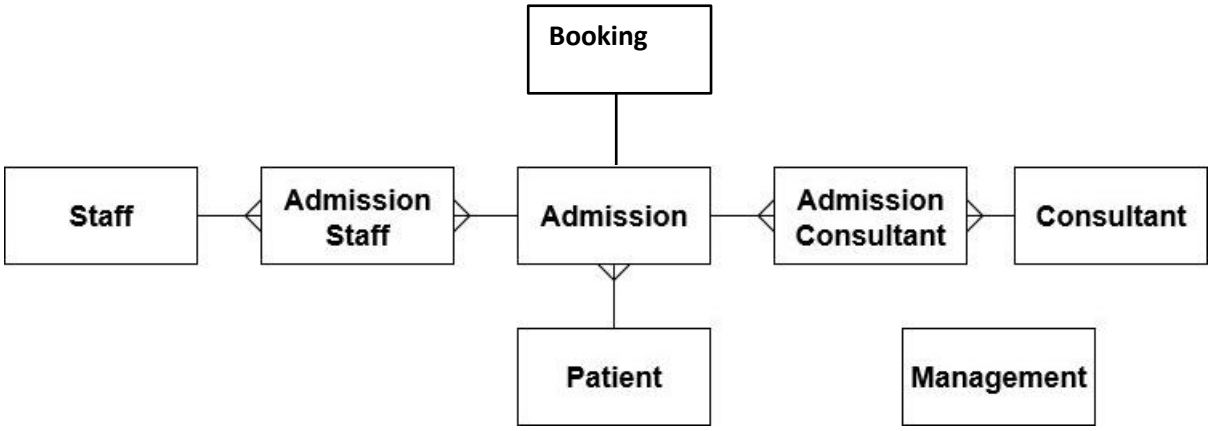
Field Name	Key Field	Data Type	Length	Example Data	Validate
timeOfNextApp	-	LocalTime	4	17:18	
dateOfNextApp	-	LocalDate	10	22/05/2018	
room	-	String	4	P001	
admissionID	FK	String	11	*ANUR1234567*	
PatientID	FK	String	11	*PNUR1234567*	
AutomaticBooking	-	int	1	0	
ConsultantID	FK	String	11	*CNUR1234567*	
BookingReference	PK	String	11	*BNUR1234567*	

(New Entity) BookingList

Again this entity isn't anything too drastic only a few attributes to make sure that the correct information is stored about the entity. This will be the exact same entities as every other object just associated with bookings.

Field Name	Key Field	Data Type	Length	Example Data	Validate
allbookings[]	-	*Object*	*100*	*A list of Bookings*	
tempBooking	-	*Object*	*15*	*Any Booking*	
nextPosition	-	Integer	3	99	

5.32 Refinements to the entities



Finally before the file system is adjusted here I just wanted to discuss any major changes to the entities of the system. To start I want to mention the inclusion of the booking entity. This has happened due to the fact while the attributes and functions could have been held in the admission entity it seem more appropriate to have it as a new class, this will also allow myself to create new instances for the employees of the system meaning the attributes don't need to be copied over. However the main reason the entity has been included as of now is that while developing the prototype I felt that a majority of the system lied within the GUI construction class. While most of the code there was about the GUI, I felt that I need to utilise the classes to my advantage and so because of this I have included a new entity. Because there is a new entity I want to address its purpose and impact it will have on the system.

The Booking entity

The main purpose of the booking entity will be to **create/Amend/Delete** bookings on the system. While initially appointments will be initiated by the patient, after the initial creation as long as an admission is active appointments will need to be created until a discharge occurs(Either through the consultant or patient[43]), other than the patient the bookings can be only be made by staff on the system. While this may sound counter intuitive not to give this feature to the consultant, from my investigation I found that the consultant tells the patient when to next see them with the staff then creating the appointment. With appointments as shown with the data structure they will possess an array of attributes along with the following functions

- **Create Booking** Either the patient or staff entity can create booking, however attributes such as the room and ward are dependent on the consultant assigned to the admission, along with this the date and times attributes being dependant on the availability of the consultant.
- **Cancel Booking** For this function again either user can perform the action however to prevent wasting time like before on the previous design a deadline will be needed to prevent last minute cancellations.
- **Amend Booking** Finally this is the last user enabled feature for the patient and staff, this function will be for adjusting the booking if plans change and for whatever reason the booking needs to change slightly. Again similar to cancelling it won't be changed after the deadline has been met.
- **Retrieve Bookings** This is a non-entity function that will be needed to retrieve all the appointments the consultant or patient has on the system. This will then be used to determine the closest appointment which will be outputted on the respective screens.
- **Closest Booking** This is an extension of the former function it is needed to in order to determine which booking is the closest for the user. It will achieve this by comparing the dates to see which is the closest when there are appointments on the same date then the time will be used to find the closest.
- **Sort Bookings** This will be needed for the consultant's timetable where the order is really important. Similar to finding the closest appointment the order will only need to be in ascending order.
- **Automate Booking** Finally the last major function will be for the ability to automate bookings. However an important aspect to make sure works is that double bookings occur

obviously the way to circumvent this would be to produce a list of times that have no appointments are already included.

Current Entities

Besides the staff entity, all the current entities are all the same in terms of features, and don't need any adjustments to them this results in no need having to re-divulge already known information already present in the original design document. This is because I believe in all regards they were effectively designed and so perform all the aspects I required them to do for the prototype. However despite this while most of their features from the prototype are complete to compensate the new entity some additions will need to be made in order to accommodate bookings on the system. However the functions are already described in the paragraph above.

The staff entity

While the other entities will not need to receive this treatment due to my happiness towards them at the moment, currently I feel that this entity like before has received little representation on the project as a whole and because of this the staffs design and purpose has become unclear over the past few months with development of the prototype. Here I wanted to reclarify what I expect from the entity along with any new features that have not been previously mentioned. I believe without this consideration the entity would not feel necessary on the system, and could may be misconstrued as an attempt to include unnecessary aspects to the system.

Currently I believe the entity should be a representative for the consultants, when a new patient is created it will be assigned to a staff whose job it will be to make sure that everything is ok for them. For example when an admission is created the first job for the staff will be to assign a consultant for the admission, following the changes to the expert system. It will also be up to them having to bring over documents from the old system and saving them as pdfs. Finally the last major feature is that like before any appointments can be made by the staff and will allow them to amend that information at any time. In regard to privacy, the information available will be retained with no private information like contact information or sexual identification being able to be viewed because coming from feedback the questionnaire suggested most people don't want the sort information divulged. To summarise the entity will be heavily involved with bookings, bringing over documents from the old system and arranging which consultant is right for the admission.

5.32 File redesigns

Considerations

As the part of the system I am least impressed coming from the prototype, now I will go into much detail in order to make sure that the eventual file designs are fully working an in a usable state by any user of the system. The whole aim is to minimise the number of files that are needed for the system. Because of this the aim is to have only one document for every patient, I will go into further detail now. Besides that the current method of retrieving the correct data seems to be perfect as of this moment so current file organisation methods will remain as of now. However as the redesigns will consider many attributes and the fact the number of documents may vary only one version will be included however multiple may be needed in the end, all the examples have been done outside a table to take advantage the longer lines. To indicate the start of new lines pointers have been included. NOTE just to indicate which sections of code iterate these sections are indexed in the final version there would be no benefit to include this. Another new reason that these files will be I included will also be due to the fact that the newly included fields will need to also be accounted for on the system.

A large factor into the file overhaul

While it was hardly mentioned in previous sections, here I wanted to give another reason why the filing system needed a new approach in place. This was largely due to the fact that the file system resulted in a very negative programming experience, whenever an issue was found it was mainly due to an incorrect value in a file causing the issue to be worse. This was due to the fact, when the issue was found it would also mean the file discrepancy would have to be resolved even before the issue was fixed. This process sometimes took me hours just to realise there was a space where there shouldn't be. While yes this sort of issue will still be relevant in the main stage of development, hopefully the new filing system and inclusion of validation mean that these sort of issues are a rare occurrence unlike the prototype.

(PatientID)_file.txt

To reduce the number of files it can dramatically go down by including this representation below, it will be used for every patient and will hold every piece of information about the patient. This will also help having to reduce the need to access different files when amending information. By having one file, it should help improve writing needs as multiple filenames won't be needed. As documents may be pdf's instead of having empty fields they will be included as the attributes for instance the consultant notes field will be used as the general notes for the pdf, however the absolute path will be needed as it is important and can't be used as another field. In regard to the demographic the format structure is the same as it went so well in the prototype no new renditions have been made to the structure except the fact that the new attributes have been included to make sure that the current feedback has been met. For the notifications there is no change as it's literally one attribute. For the admission information like shown in the structure redesign the booking fields have been separated into a new entity along with a new line for the file.

(**patientID)(Password),(surname),(firstname),(houseNum),(houseStreet),(town),(Postcode),(contactNumber),(nationality),(bloodtype),(smoker),(drinker),(numOfAdmissions),(D.O.B),(Religion),(Allergies),(Gender),(disablities),(carer),(translator),(Sex),(county),(DaysSinceLastUpdate)**

=====New Line=====

(**notification**),(**notification**) *This repeats for the number of notifications*

=====New Line=====

{*===== For every admission=====*

(admissionID),(consultantID),(numOfDocuments),(ward),(listOfSymptoms[]),(archived),(dateAdmissionCreated),(currentDiagnosis),(areasAffected)

=====New Line=====

(*BookingReference*),(timeOfNextApp),(dateOfNextApp),(*AutomaticBooking),(room)*

=====New Line=====

{ *=====For every document associated with the admission=====*

(documentID),(dateOfDocumentCreation),(timeOfDocumentCreation),(doctype),(medicationName),(medicationDosage),(medicationIntakeTime),(medicationDateOfNextDispatch),(notes),(testResults),(Authorofdocument),(*Hospital*),(AbsolutePath*),(***numberOfPages*)

}

=====New Line=====

}

(StaffID)_file.txt

For this file while there isn't too many fields to concern it is the fact that they will repeat a lot and so it needs to be made sure that no attributes are missed out. Similar to every other user-based file the first section is their demographic information which has been standardised following general feedback, the only inclusion is the employee specific attributes them being wage and hourly rate. The next section is the patients under the staff entities supervision, these will the patients the staff will organise appointments for and add documents to their account. Now finally this is where the bulk of the file will appear, the action log for every event the staff performs on a patient's information it will be stored here. Because of this overtime the file will become quite large. However as the number of staff on the system is finite I don't see this becoming an issue.

(**StaffID),(employeeID),(Password),(surname),(firstname),(houseNum),(houseStreet),(town),(Postcode),(contactNumber),(nationality),(smoker),(drinker),(D.O.B),(Religion),(Allergies),(Gender),(disabilities),(carer),(translator),(Sex),(county),(DaysSinceLastUpdate),(wage),(hourlyPerWeek),(archived)**

=====New Line=====

(**patientID**),(**patientID**),(**patientID**),*This repeats for the number of patients the staff has*

=====New Line=====

(numberOfActions),(localDate),(currentAction),(patientID),(AdmissionID),(OrginalData),(NewData)

This will repeat for every action the entity has performed

(ConsultantID)_file.txt

Here for this final user the data storage will be used in a way that is the exact same for every other entity. However the main aspect of this file is that it does contain some repeated data that being the booking information. While it could be seen as inefficient to hold data twice I believe that the time taken to rewrite these files when being updates would outweigh the time taken to search the individual file to isolate then call the line of code for one booking. Anyway, the first section like the staffs, contains the demographic information. The next section contains a list of admissions associated with that patient that is with the consultant. After this in the same order as the admissions the list of booking details can be found. Finally like the staff the data for the action log can be found here at the last part of the file.

(**ConsultantID),(employeeID),(Password),(surname),(firstname),(houseNum),(houseStreet),(town),(Postcode),(contactNumber),(nationality),(smoker),(drinker),(D.O.B),(Religion),(Allergies),(Gender),(diablities),(carer),(translator),(Sex),(county),(DaysSinceLastUpdate),(wage),(hourlyPerWeek),(archived)**

=====New Line=====

***(practisesList[]),(*numOfPatients)**

=====New Line=====

{*===== For every patient=====*

(**patientID**),(**admisisonID**),(**admisisonID**),(**admisisonID**)*This will repeat for every patient the entity has affiliations with*

(*BookingReference*),(timeOfNextApp),(dateOfNextApp),*AutomaticBooking),(room)*

(*BookingReference*),(timeOfNextApp),(dateOfNextApp),*AutomaticBooking),(room)*

(*BookingReference*),(timeOfNextApp),(dateOfNextApp),*AutomaticBooking),(room)*

}

=====New Line=====

(numberOfActions),(localDate),(currentAction),(patientID),(AdmissionID),(OrginalData),(NewData)

This will repeat for every action the entity has performed

Glossary_Of_Terms.txt

Again as this is a minor file there is no need to amend any of the information about it. To add to this, there is no other file it would benefit from merging with. Because of this I intend to keep the way it is and keep the design from the initial design. This is also due to the fact that the file works perfectly in the prototype and because of this I see no point in changing how the current version of the system works. As the primary key is the word itself there is no need for any other values other than the word and its corresponding definition. While the length of each line in the file will vary on the definition of the word, the actual size will be quite large with their going to be many hundreds of lines. While I could instead of using new lines, use one the fact is either way the actual length would be considerably large. Because of this I found looking vertically down through a document much easier than going horizonal and so have decided to do it like this. To add to this the code either way I similar enough as I would either have to step through a new line or use selection to determine whether a new definition is occurring.

(Word),(Definition)

=====New Line=====

This will repeat for every definition

General.txt

This is a file meant to hold general options about the system. While no data here is associated to general entity fields here fields could be from a certain entity in particular. For example the username of the remember me aspect suggested here by Phil [20], the main reason why this file is needed. It is here as the system otherwise would have to search through every text file on the system to see which use has the remember me attribute applied, the other issue would have been updating it as not only is the attribute of the new user needs changing but the old one also. The only other current plan to have on this file will be holding information about every surname on the system for storing in the generation of user based primary keys.

(PatientID)/(EmployeeID)

=====New Line=====

(Abbot,1),(Armitage,3)//A based surnames

(Baxendale,10),(Bell,1)//B based surnames

(Curtis,5)//C based surnames

…..

(Young,12)//Y based surnames

()//Z based surnames

Management.txt

This is a file that was realised just as the processes where being included. As the management entity will retain the basic default attributes inherited by the classes parent ones, they can therefore all be combined onto one file. To add to this as no new accounts will be created of this entity, the size should relatively stay the same. Despite this while hardcoding could be an option I do intend on having the fields be editable so therefore this is the only way to easily do so.

(**ManagmentID),(employeeID),(Password),(surname),(firstname),(houseNum),(houseStreet),(town),(Postcode),(contactNumber),(nationality),(smoker),(drinker),(D.O.B),(Religion),(Allergies),(Gender),(diablities),(carer),(translator),(Sex),(county),(DaysSinceLastUpdate),(wage),(hourlyPerWeek),(archived)**

This will repeat for every management entity on the system

=====New Line=====

(**ManagmentID),(employeeID),(Password),(surname),(firstname),(houseNum),(houseStreet),(town),(Postcode),(contactNumber),(nationality),(smoker),(drinker),(D.O.B),(Religion),(Allergies),(Gender),(diablities),(carer),(translator),(county),(wage),(hourlyPerWeek),(archived)**

5.4 Processes

Now that most of the system has been redesigned the last major section of the document to discuss and plan is the improved and redesigned processes. To prevent having to recreate the entirety of all the pseudocode written in the design 3.52 document. Prior to doing this, I weigh up the current processes against my plans and discuss whether or not they need to be reworked, to add to this if any new processes of the system are required to meet new conditions brought on by certain feedback they will also be included and will be labelled as such.

Old Processes

Overall, while I thought my current processes aren't necessarily lacking to an extent, after having the chance to actually program and develop the system, I feel there is definitely room to adjust the objectives so that they will suit the overall new needs of the system much better. While I feel spending hours making sure that every processes present is made perfect is a good idea, there is no need as specific processes can be generalised to some degree and still provide the same amount of usefulness when it comes to development, not forgetting to mention some objectives will not benefit from a redesign anyway. The implications on the redesign however may impose some areas in the prototype having to be redeveloped to follow new specifications laid out in this section. While this is to be expected, the prototype is basis of what the final system will be, by changing too many processes the system will begin to no longer resemble the original idea, with it having the chance for myself to lose focus when thinking of ways to improve certain processes.

Process/ Objective number	Description	Justification	Array / Files Used
1. **Input/select symptoms into expert system**	The purpose of this process is to allow the user to input their symptoms onto the system, in this new version there are less fields to entire as they were hardly relevant to the admission. The fields in general try to depict what the patient is experiencing by allowing symptoms to be entered manually.	The reason this process has been redesigned is to allow and make use of the new symptom fields that have been included on the system during the later stages of development and the redesign. In the new code the pseudocode will need to recognise if any Boolean attributes or body parts have or have not been selected. In addition having to highlight body parts to show areas that have been selected. The final thing will be that some of the unneeded options have been removed	**Symptoms**[] – To hold all the symptoms entered by the user **listOfAreas**[] – This will store every area of the body the patient is affected by and can correctly display it to the user

Process/ Objective number	Description	Justification	Array / Files Used
2. **Determine suggestion**	This process is generate a basic diagnosis nothing major, a change is that the consultant is no longer determined and left to the staff entity to decide. The fields are used from what the user entered from objective 1. It works by counting similarities with symptoms already on the system and figuring out which diagnosis is the most like the one entered by the user.	The reason the process hasn't been redesigned is because the objective was designed to be simple as possible from the design stage, while before I thought the process was going to need to more complex pseudocode, on reflection the code already shows what I was planning on including. The process will therefore not need any changes to it fortunately.	**Symptoms**[] – To hold all the symptoms entered by the user **listOfAreas**[] – This will store every area of the body the patient is affected by and can correctly display it to the user **Diagnosable symptoms**[][] – these will be hardcoded onto the system and will be used to determine the suggested diagnosis
3. **Generating new patients**	Using the fields from objective 22 once the patient creates an account their information is saved into a file and the patient is logged in and attached to a staff entity. The fields that they enter are mostly the demographic along with some account fields. When created they are informed of their username and password.	The main reason that this process will need to be redesigned will be to make use of the new attributes included on the patient account. To add to this the process will also call upon the new file structuring system and will use some other new features that have been included over the course of the redesign stages of development. In general the processes worked really well in the prototype besides having issues in file management the actual entity creation worked well.	**(StaffID)_file.txt** – this will be needed to store the information regarding the patient that has just been created (**PatientID)_file.txt** – This will be created and will initially store all the information inputted by the aforementioned patient **AllPatients**[] – this field will only be used if the staff entity logs in after an account's creation.

Process/ Objective number	Description	Justification	Array / Files Used
4. Generating a new admission	Similar to objective 3 once an admission is created the fields relating to the entity are written to the patients' file and the staff uses the data from objective 2 to determine a consultant. Once created the admission should then be added to the patient's admission list so it can be immediately seen by the user.	Similar in regard to objective 3, the process has been decided for a redesign in order to include the new attributes associated with it, that have just recently been included. To add to this it will also need to call upon some other newly defined processes for instance objective 46. Finally the processes needs to make sure the newly designed files also are included correctly. The only issue that needed to be redesigned was the file management part of the process, as that has been taken care of little is needed to change.	**Symptoms**[] – To hold all the symptoms entered by the user **listOfAreas**[] – This will store every area of the body the patient is affected by and can correctly display it to the user **(PatientID)_file.txt** – This will be create a new entry for the admissions and will store all the information inputted by the patient **AllAdmissions[] –** The array will be needed as once the admission has been created it will added to the patient's list if admissions.
5. Booking a new appointment	When a consultant is determined for the admission the staff entity will also arrange an appointment for the patient and will enter the fields for the appointment, once created until the deadline the patient can amend any appropriate fields. With concerns about dates, unavailable times will be unable to be selected. Any action performed by the employee will be recorded in the action log	The reason this process has been redesigned is to allow for a wider range of features to be included, this means that the pseudocode will begin to reflect the final system much better. For example the action log objective 13.B will be called if the booking is created by anyone other than the patient. In addition to this as the booking is now an entity, and because of this will need to call upon a selection of new attributes. Finally as the file system is new, there will be changes in the location of the booking.	**listofAppointments**[] – this will hold all the appointments the patient currently has, the newly created appointment will be added to this **(PatientID)_file.txt** – This will be create a new entry for the appointment in the correct admission and will store all the information inputted by the user **(ConsultantID)_file.txt** This file will be needed as not only will the patient file be updated but the consultant one as well, this however won't contain data that would be obvious like the consultant

6. Login users	Using the credentials given on the account creation the user will enter them to get onto their own part of the system. The system will first use the username to find the account and then the password to validate the user. As part of this once this has been finished the newly added objective 50 may also be called	The redesign of this processes is to utilise the new filing system, as the system no longer requires individual entity specific login files. Because of this the actual pseudocode used will be much smaller as a result and will depend on more calling of more objectives. The overall aim for this redesign was make a simple task more efficient. Instead of having to include 4 different search statements only one will now be needed. As there was no official login on the prototype the actual process itself can't be justified or dismissed	**Management.txt** **(ConsultantID)_file.txt** **(StaffID)_file.txt** **(PatientID)_file.txt** All these files will be needed as the username will be used as a search query of the file to return the correct user.
7. Display menu options	N/A	N/A	No files or arrays are used
8. Add employees	The purpose of this process is for the management entity having the ability to add employees (either consultants or staff) onto the system, all the fields about the entity is entered here such as hourly rate or address.	The purpose of this redesign will be to include the new fields associated with the user entity to now be included. Besides the new filing system, no other changes will be needed to be made to this process as the objective works well currently on the prototype, this is in regard to general new entity generation obviously.	**practisesList[]** ***allConsultants[]*** ***allStaff[]*** **(ConsultantID)_file.txt** **(StaffID)_file.txt** Some of these data structures are dependent on whether or not the user is a staff or consultant entity either way once an employee is created their respective file is created adding their inputted information to the file.

Process/ Objective number	Description	Justification	Array / Files Used
9. Archive employees	While objective 8 deals with including employee entities on the system, this one deals with taking them off. While it won't physically remove the data from file, it will prevent said data from ever being accessed. The other side to it is making sure no patient or admission is disregarded in the time being this is a new consideration following feedback from Emily [42].	The reason the process doesn't need changing is mainly down to the fact the actual archiving process only relies on a single attribute which can easily be amended, most of the new work will come from the new objective which will sort with the resolving of any patient details. The process can't be discussed in too much depth because of its omission on the prototype	**practisesList[]** ***allConsultants[]*** ***allStaff[]*** **(ConsultantID)_file.txt** **(StaffID)_file.txt** Similar to 8 the files and arrays are dependent on what user have been selected. Either way the information in the file is updated to change the archive attribute to true.
10. Sort for employees	When the management entity sees all the employees on the system, they might decide to have a different order in which they want them viewed. In order for this to be achieved the sorting algorithm re-orders them into the desired way.	The main reason that all sorts aren't getting a redesign is down to a few reasons. First they play a minor role on the system and only allow for data to be rearranged, when there are so many items per list it is much more convenient to search for items instead. In addition to this the algorithm itself I really good for sorting and the system would only see a light improvement by utilising more advanced sort.	***allEmployees****[]* The reason this data structure is needed is because the employee that is desired will be stored in this array and can be retrieved from it, this will be a much easier processes than having to re-read the particular file that contains the particular employee.
11. Search for employees	The purpose of this function is to allow either the system or user find the employee they are looking for using the primary key to find the entity in particular.	The reason that the search doesn't need changing is because the search at the current moment of time can't be improved without extra resources into research. While I would like to create the most advanced searching algorithm possible it simply just isn't feasible due to the deadlines I have to keep up with. To add to this while the size of the arrays may be quite large, the binary search speed of O(log n) tells us that the search is more than fine for the system.	***allEmployees****[]* Similar to 11 this data structure is needed because it will be used for directly searching for specific employees on the system

12. View an employee's transaction log	After objective 11 has finished the system will ask for two time parameters and will output a table containing every action performed by the employee. The data will include other relevant information including data, patient involved etc.	The reason I feel no changes are needed to view the action log is because while amending to it will require changes in term of the file location, this process just performs the action of displaying it to the user. While the process is slightly different being in a GUI the pseudocode won't be any different so no changes are needed.	**(ConsultantID)_file.txt** **(StaffID)_file.txt** One an employee is found their specific data will need to be retrieved this will utilise this objective and these files in order to do so as their information will be stored on these files. The file is respective on what type of employee is being searched obviously
13. A. Read transaction log from file	In order for objective 12 to occur the information needs to be read from the employees file, this resolves in the two time parameters being utilised as boundaries for what actions to include and not include. All this information is then saved to the appropriate fields and outputted	The purpose of this redesign is to allow the system to now correctly read the information from the new files and also account for the new attributes, besides this there is no other parts that need redesigning. This is simply because the process worked well on the system and I have no issue leaving the majority of the objective the same	**(ConsultantID)_file.txt** **(StaffID)_file.txt** Once the information has been found the fields are then read from file and then displayed, without this file no data could be generated
13. B. Write transaction log to file	When an action by an entity is performed it is then saved to their file, along with what action is performed some others are included for the sake of safekeeping and so more context can be given without having to question the user directly. The general purpose of the feature is to make sure that unacceptable behaviour goes on. This is achieved by recording every major action which will deter users from doing things they shouldn't.	Unlike objective 12 this process will need to see some refinements. This will be mainly down to writing the data into a different part of the file, where normally the file just adds the action onto the current staff's line, as the file is employee specific every action can now require its own line, and so these changes need to be made.	**(ConsultantID)_file.txt** **(StaffID)_file.txt** Similar to how data is read, when an action is performed the action along with any subsequent data is then written into file where it is kept. Because the actions only relate to employees, only employee files are needed.

Process/ Objective number	Description	Justification	Array / Files Used
14. Staff can sort for patients	When the staff is looking through their patients they can view in which order they see the patients. The standard currently are ascending and descending (in terms of their primary key). However this would not be the main method to finding the patient as this would be down to searching/ objective 15.	The main reason that all sorts aren't getting a redesign is down to a few reasons. First they play a minor role on the system and only allow for data to be rearranged, when there are so many items per list it is much more convenient to search for items instead. In addition to this the algorithm itself I really good for sorting and the system would only see a light improvement by utilising more advanced sort.	***allPatients[]*** **(StaffID)_file.txt** These structures are needed because in order for staff to select patients they need to be in some list to be displayed and the only way of retrieving the list is pulling that information from the staff's profile which contains eligible patient's
15. Staff can search for a patient	When the staff need locate and find a particular patient they will enter a field that is acceptable. Currently the primary key, and the system will return all the patients that meet the query. To add to this once they no longer need the returned search items by pressing the X they will clear the items and original list will appear back to them.	The reason to update this search is twofold. First the search should get improved to a recursive quick sort, so that access times can be improved. To add to this the algorithm needs to be made so that multiple search fields can be utilised and that more than one value can be returned to the user.	***allPatients[]*** **(PatientID)_file.txt** **(StaffID)_file.txt** – The reason for these data structures and files is because that all the patients are needed to be retrieved, this is found in the staff file, all these names are then stored in the array. Finally once the desired patient is found their information is pulled from the file containing that primary key.
16. View patient details	The purpose of this field is to allow the staff entity to view the basic attributes about their patient, this will be fields that aren't sensitive to the user and could be basic information about them.	The reason this objective doesn't need to be updated is because the staff entity will only need to see the basic attributes about the patient, since that is already the case there is no need to reimplement the pseudocode as there have been no changes in this regard. The process itself should be more than suitable after having the chance to test it in the prototype.	***allPatients[]*** **(PatientID)_file.txt** As the following process after objective 16 the list of patients will be needed to hold the information about all the patients, from this the information should then be fully read from the desired file after being selected by the user. Which would result in the information being retrieved

17. Add archived notes from old system	A key process of the entity, as of the systems launch a changeover period will be in direct affect and as the idea would be a parallel crossover, not accepting new bookings on the old system, it will be the staff entities role to scan old documents and then attach them to the correct admission.	The reason this objective needs to be redesigned is to make way for the new file organisation of the system. To add to this in the redesign some new attributes have been included on the document entity and therefore in order to correctly show the user the process the changes need to be made. While I only had a finite amount of time to test the feature on the prototype the time spent should mean that the process should be fine for the system.	**(PatientID)_file.txt** **(StaffID)_file.txt** The purpose of these files is to store both the documents and actions to their correct locations respectively. For the patient file, it will store the absolute file path to the pdf to open it. For the staff it will record that a document was created and copied over from the old system
18. Amend bookings	As circumstances change, either the patient or consultant may be unable to attend an appointment and because of this either the staff or patient can change the information regarding the booking. While not all fields will be amendable and that there will be an eventual deadline until this process will stop, it should be open to ant affiliated user.	The purpose of the redesign of the amending of bookings is to make sure that all the correct entities receive the booking, an aspect neglected in the initial design document. In addition to this the file system will be different from the prior one so it needs to account for. Finally the pseudocode needs to have a deadline date included so that the information can be prevented from being amended.	**(ConsultantID)_file.txt** **(StaffID)_file.txt** **(PatientID)_file.txt** The purposes of these files is either to record the new booking on the file to save it correctly, in this case either the patient or consultant. Or it is to record that the amendments were made to the booking in the first instance
19. View patient bookings	The purpose of this process is to allow the staff entity the ability to view the booking information from selecting it from their account. Once they have the booking information they need, they can alter it assuming the time constraints haven't been met.	The reason that the objective has been included is because it was left over from the initial design of the prototype. This makes more sense when thinking about the abilities now enabled to the staff entity, the main one being having the objectives to now alter the bookings. While it is a process that may be thought as irrelevant now, I believe it will be important in finding the correct booking.	(**PatientID)_file.txt** **The reason this file is needed is because in order to view the booking the information will need to be retrieved from file.** **AllBookings[]** The reason the array is needed is because the booking needs to be selected in the first place out of all the patient's admissions

20. View patient admissions	Similarly linked to objective 19 this process will allow the fields of the admission to viewed by the staff entity. While I feel that after the consultant has been decided(by staff) these fields should become more restricted it should show the general information, such as diagnosis while more sensitive information obscured.	The reason that this prototype has been kept the same is because the fact the staff entity while having the ability to amend information should have a limited access to the information presented to them. This is also the case because the attributes held by the admission are the same so therefore there will be no need to amend this information.	**(PatientID)_file.txt** The reason this file is only needed is because all the information needed on the account will be stored here. While the array could have been included here it is already stored on the file so it would be converted anyway.
21. Have patients view their Admissions and Demographic information	Being the most important objective for the patient, this process utilises the retrieving of their information from file and then displaying it to them for them to view in their entirety unlike the staff. The fields here will be everything held about the patient.	This process has been kept the same because their information will be brought over by a new objective and so little would be changed to adapt this change. When it comes to accessing this information the process in doing so will remain the same. Because of this I feel the objective, ultimately has no real reason to change how the current processes work. Because of how the prototype went I feel that in general this process was perfect and didn't need changing at all.	**Symptoms**[] – To hold all the symptoms entered by the user **listOfAreas**[] – This will store every area of the body the patient is affected by and can correctly display it to the user **(PatientID)_file.txt** Similar to 20 all the information will be read from file; however the difference is that the admission will also be included so the arrays will be included.
22. Amend demographic information	The purpose of this function will be to allow the user to alter the information regarding them as a general person. Once these fields are updated, all the information is then sent to the account and then overwritten, therefore saving the details. A counter will be kept for every user indicating when was the last time they updated their information, in order to prevent information becoming out of date and invalid.	This process has been redesigned in order to account for the new attributes that have been both removed and added, to add to this it will also need to include the action log in case of an employee alerting that information. Finally the system files have been update so the locations the data is being written to has changed.	**(ConsultantID)_file.txt** **(StaffID)_file.txt** **(PatientID)_file.txt** As the information that is being edited by the user will vary on the entity, all three files will be needed. In the case for the employees they will be needed for the transaction log system to write that this process had occurred. For the patient as you would expect the information regarding the demographic would be written to file.

Process/ Objective number	Description	Justification	Array / Files Used
23. Validate information	The purpose of this objective is a system side process. As the system deals in data types of varying types the system will not run solely of input as strings, and so fields will need to be validated in order to make sure that any input meets the standard input required.	There is no need to change the validation algorithms as they will always be the same from the point of the codes conception I feel perfectly happy to have these processes stay the same on the system. While validation was missing in the prototype I feel that the validation individually couldn't see an improvement from the original design.	No files or arrays are used
24. View bookings in entirety	This process will be now for all users, what will happen is that every detail regarding the appointment will be shown to the user. Unlike the ability to amend bookings like objective 18, this will come into effect once the deadline has been met and changes can no longer occur.	The reason that this process has been changed is to make way for the method of retrieving them in the first place is correct. Other than that and any new attributes included nothing else has changed the process. While the booking entity wasn't present on the prototype I feel that the changes made recently would need to be seen reflected in the pseudocode before development starts again.	**(ConsultantID)_file.txt** **(PatientID)_file.txt** **AllBookings**[] The reason the files are needed is because the fact the information would need to be read by a file in the first place. With respect to the array, in order to view the booking one will need to be selected this can only be achieved from selecting the booking from the options available hence the array is needed.

Process/ Objective number	Description	Justification	Array / Files Used
25. Add bookings	The purpose of this feature is to allow the use to create their own booking, while the option to automate bookings may be required this will happen for admissions that may only need to be seen twice or similar events. The fields once entered will then be written to file where the booking can then be view amended or deleted, assuming the deadline of alterations has not been reached.	When comparing the pseudocode of the old process one section that I feel now needs to be included is the inclusion of the action log, besides that the other major thing that would need to have included was the generation of a primary key. While the process could have included some more features I feel that the renditions made along with keeping the rest of the process is the best course of action for this objective.	**(ConsultantID)_file.txt (StaffID)_file.txt (PatientID)_file.txt AllBookings**[] For the files, for the consultant and patient, when one is created they will both be needed in order to correctly store the information. With respect to the consultant and employee their files will be needed to record that an action has been performed to the patients account. Finally as the system will be running the booking will be needed to be added to the list of current bookings to prevent from the system needing to be reloaded.
26. A) Sort documents	The purpose of this objective is to allow either the patient, consultant or the staff entity to sort the documents that they can see into a particular order. This will be achieved by swapping items continuously until the list is in the new order. The options for the order they will be in are either ascending or descending.	The main reason that all sorts aren't getting a redesign is down to a few reasons. First they play a minor role on the system and only allow for data to be rearranged, when there are so many items per list it is much more convenient to search for items instead. In addition to this the algorithm itself I really good for sorting and the system would only see a light improvement by utilising more advanced sort.	**(PatientID)_file.txt AllDocuments[]** Similar to other sorts, the list of documents are needed as the items to have ordered need to be in a data structure otherwise the process couldn't effectively work. When it comes to the file, once the desired item has been found it then can be returned and then any following process can occur.

Process/ Objective number	Description	Justification	Array / Files Used
26. B) Search documents	The purpose of this objective is to allow the user to search for particular documents that meet the search query entered by the user, however, to prevent search times from being too long only a finite amount of search fields will be available to the user. More than one item will also be returned if they all meet the condition.	The reason that the search doesn't need changing is because the search at the current moment of time can't be improved without extra resources into research. While I would like to create the most advanced searching algorithm possible it simply just isn't feasible due to the deadlines I have to keep up with. To add to this while the size of the arrays may be quite large, the binary search speed of O(log n) tells us that the search is more than fine for the system.	**(PatientID)_file.txt AllDocuments[]** Similar to 26A the list is needed so that the processes can occur correctly and efficiently. And in respect to the file, that will be needed as it will contain the data that is returned from the search.
27. Print documents	The purpose of this feature is to allow the user to print of a physical rendition of the document on screen in front of them. This should be a A4 copy of the document as during design all documents where designed to follow an A4 scale to allow copies to be printed of and used like traditional copies if the patient wanted. The reason it is included, is because the patient may want to keep a physical copy for legal reasons or for safekeeping.	This process doesn't need to change because the action of printing won't change, while I could have altered it to just for the sake of making changes, as it is a standardised process I feel that little else can be done to improve it. The process while absent on the original system I feel could be made into a really special feature as long as the system works well.	N/A The reason no file or array is needed is because there is no amendments to any data so no writing to file Is required, all the data is already in memory so the system doesn't need to read from file either. As we are only dealing with a single documents, there is no need to hold it in an array either.

Process/ Objective number	Description	Justification	Array / Files Used
28. Search for patients	The purpose of this feature is to allow the consultant to enter the patient they are looking for and a list of patients that meet the search query should be returned. As the list can vary in size the list could range from 0 patients, indicating no results were found, up to all patients with the surname Smith for example.	The reason that the search doesn't need changing is because the search at the current moment of time can't be improved without extra resources into research. While I would like to create the most advanced searching algorithm possible it simply just isn't feasible due to the deadlines I have to keep up with. To add to this while the size of the arrays may be quite large, the binary search speed of O(log n) tells us that the search is more than fine for the system.	**AllPatients[] (PatientID)_file.txt** The purpose of this file is so that once the correct patient has been found they can then have their information retrieved. With respect to the array, this will be needed so that a search algorithm can effectively be used to find the desired results from the query.
29. Consultant can view patient files	The process of this objective is to use the prior objective to find the location of the object then to allow further options to be selected to navigate the consultant through the rest of the patient's menu Its role is to allow for further navigation of the patient's data and information on the system	The reason the current process is the same because that in the prototype the process of retrieving the patient's file seems to be successful, because of this I see no need to have the option to change how things currently are. While the location of data and the data itself is different now these processes are being handled in new objectives. As the process had already existed in the prototype it should provide sufficient evidence to support the idea that this process didn't need changing.	**AllPatients[] (PatientID)_file.txt** The purpose of this list is that the consultant can view all the patients under his watch efficiently. With the text file this will be sued to store the information about the desired patient that is selected once a patient is found.

Process/ Objective number	Description	Justification	Array / Files Used
30. View patient Demographic information	The process of this objective is to output all the attributes associated along with the demographic Its role is to show to the user all the fields that are in the demographic. Besides this there is little else to it.	I believe that the ability to view demographic information needs some changing. While the current method of retrieving the data was fine, now with the new filing system is in place I believe large improvements can be made which will benefit the system. These will mainly be how data is stored in the file like previously explained in other parts of the document, not only this but the system can also have included the outputting of the new attributes created in the data structure section mentioned before.	**(PatientID)_file.txt** This file is needed as it will be used to store information about the patient held in objective 29, once selected all their information will be read and outputted.
31. Sort admission	The process of this objective is to allow the user to sort the list of documents in the array. It does this by retrieving the documents from file, saving them to file then allowing user input to determine the order in which they are sorted. It does this by using the insertion sort and inserting the document into the correct place Its role is to allow the array or essentially the file to be put into the correct order determined by the user	The main reason that all sorts aren't getting a redesign is down to a few reasons. First they play a minor role on the system and only allow for data to be rearranged, when there are so many items per list it is much more convenient to search for items instead. In addition to this the algorithm itself I really good for sorting and the system would only see a light improvement by utilising more advanced sort.	**(PatientID)_file.txt** **AllAdmissions[]** The purpose of the file is so that the information about the found desired admission can be read from file and then outputted to the user. The purpose of the list is so that the order can be effectively rearranged and ordered in a new list.
32. Edit Prescriptions	N/A	N/A	N/A

Process/ Objective number	Description	Justification	Array / Files Used
33. Add Admission information	The process of this objective is to retrieve the desired patient and find the desired admission from this any new fields are brought up for the consultant to enter the data into those fields. The action is written to the consultant's action log and the new data is added to the file after it has been encrypted Its role is to allow the user to add information to the admission, fields may be left due to human error or could be left until the data in the field is eventually known like diagnosis etc.	The main reason this process needs reworking is to allow for the newly included attributes and screen designs to be fully utilised. The best example of this is the inclusion of the human silhouette, where the newly added body parts can now be used to provide a clear depiction of where the symptoms resonate from. Finally the new filing system will also need to be implemented to the process so that the data is writ correctly, along with the action also.	**Symptoms**[] – To hold all the symptoms entered by the user **listOfAreas**[] – This will store every area of the body the patient is affected by and can correctly display it to the user **(PatientID)_file.txt** Once the amendments have been mad they will be written onto this file.
34. Edit Admission information	The process of this objective is to retrieve the desired admission and bring up all the information regarding the admission. All the fields are decrypted and then saved into an array the user is then allowed to enter the new data in the fields. The new data is then saved into the action log and then saved over in the array. Finally, the admission is then concatenated encrypted and written to file Its role is to allow updates to occur to the admission, this should allow in case fields that are not permanent like address change	The reason that this process needs to be redesigned is similar to objective 33. The main point being that in the redesign a lot of changes were made in respect to the data structures and so attributes on the old process may no longer be a part of the process, to add to this once this information is changed, it will also need to be written to a different file so the process will also have to accommodate this too. While in general the process of amending admission worked great it's just that the file management could have been better.	**Symptoms**[] – To hold all the symptoms entered by the user **listOfAreas**[] – This will store every area of the body the patient is affected by and can correctly display it to the user **(PatientID)_file.txt** **(ConsultantID)_file.txt** **(StaffID)_file.txt** These files are needed for multiple reasons for patient, it will be needed for writing the actual information to file. For the two employee files they will be needed to update the action log if either user edits the admission of the patient.

Process/ 35. Objective number	Description	Justification	Array / Files Used
36. Add notes	The process of this objective is to allow the document type for the user to be declared and then accordingly set the individual fields to what the user inputs, this data is then validated and encrypted and then saved to the employee's action log. Its role is to allow the user to add documentation onto the patient's admission and allow new information to be saved.	The purpose of this redesign is to allow the process to contain the new aspects of the system, while the objective of creating new documents hasn't seen much changed to it, the actual process of having to write the document to file has changed as a new file is being used and so overall the process has been redesigned to follow these amendments. To add to this the process will now also have to account for whether a new page may be needed for a document following changes made.	**(ConsultantID)_file.txt** **(StaffID)_file.txt** **(PatientID)_file.txt** These files will be needed for two reasons, for the patient it will be needed to write the actual document to file. For the employees similar to objectives 34 they will be a long record of the actions performed on the patient's account and information.
37. Encrypting data before being written to file	The process of this objective is to pass the string through the objective then the ASCII value for each index is then moved up 5 values. The word is then concatenated and then saved as a whole string after this it is then returned to the desired field. Its role is to allow any data to be written to file to be encrypted to make sure it is secure and allows sensitive data to be protected.	While the standard Caesar shift is useful enough for demonstration of its potential, as the system will contain people's sensitive information I think it's time I improve the security of the system. While java does provide its own cipher encryption algorithms, as the entire system has currently had no reliance on external classes or downloads I think I want to keep it that way. Because of this I will utilise the ATBASH Latin cypher as it is more complex than the Caesar shift but also provides a more complex result to understand.	N/A

Process/ Objective number	Description	Justification	Array / Files Used
38. Decrypting that has been read from file	The process of this objective is to retrieve the encrypted data from this the index of the individual string are then converted back into the original data, this is done by reducing the ascii value down by 5 after this it is then concatenated again and then returned to the user Its role is to allow the user to decrypt any data that is read from file, without it the user would be unable to understand any data returned from file	Similar to objective 36 I think the system needs to utilise a more secure encryption algorithm. Obviously I won't be able to provide a level of security used by large companies however, it should be a more standard algorithm as Caesar while suitable for a prototype needed a more advanced encrypting algorithm.	N/A
39. Using the Jargon library	The process of this objective is to retrieve a desired word the list of definitions is also retrieved then a binary search is performed to find the location of the word. After it has been found the world has been split the word is then returned Its role is to allow the definition to be retrieved and then showed to the user	For this process I think it can stay the same due to the fact that the prototype went so well with it doing everything I expected from it. While the purpose was basic from the beginning I feel that there is nothing that can be done to it to improve on it. Because of it no pseudocode is required from it.	**Glossary_Of_Terms.txt AllDefenitions**[] The file will be needed to retrieve all the definitions from file sot that they can be added to the array. The array is needed so that they can be outputted and shown to the entity, long side this the sorts and searches can also be applied to this.
40. Adding to the Jargon library	The process of this objective is to allow the consultant to enter the word they want to use and the definition is then concatenated. Finally, a binary search is used to find the location, afterwards it is then written to the correct location Its role is to allow new words to be added by the consultant otherwise no new definitions will be included besides the initial declaration	For this process I feel that a change isn't needed to compensate for the changes in regard to file design. While the objective was missing from the prototype I feel this part of the system hasn't seen much change and can therefore carry on the way it was originally intended. While it may seem that the decision and discussion about the process seemed brief there isn't too much to talk besides the fact that the ability to create definitions doesn't need to be altered.	**Glossary_Of_Terms.txt AllDefenitions**[] While for objective 38 the file was needed for reading; in this instance it will be for writing the most current definition to file. The array will be for updating the current list so that the definitions are not of synchronisation.

Process/ Objective number	Description	Justification	Array / Files Used
41. Search through Bookings	The process of this objective is to use the search value entered by the user and compare it to the list of items in the array, it will then perform a binary search to locate the position the entity is located allowing for further manipulation of the file. The search consists of comparing the search value with the midpoint and disregarding the side that is irrelevant until the entity is found. Its role is to find the user the location in which the entity is stored at and return it to them for further use. If it is not present in the array an error message is shown	The reason that the search needs changing is because the search utilises the old filing system and because of this is trying to retrieve information from a file that would no longer exists, to add to that while the other files are somewhat similar to older versions this file has been merged with another and so needs to be updated now to coincide changes made to the files.	**AllBookings**[] **(PatientID)_file.txt** For the array, this will be needed as it will be the most effective way for searching values as the searching algorithm uses it. The file will be needed as it can be used to retrieve the information from the desired entity from the search algorithm once found.
42. Search through demographic information	The process of this objective is to use the search value entered by the user and compare it to the list of items in the array, it will then perform a binary search to locate the position the entity is located allowing for further manipulation of the file. The search consists of comparing the search value with the midpoint and disregarding the side that is irrelevant until the entity is found. Its role is to find the user the location in which the entity is stored at and return it to them for further use. If it is not present in the array an error message is shown	This process is no longer required as it relied on an old file system to retrieve old demographic information, in that system every piece of demographic information was read from a single file, so a search was needed. However with the new system once a username is known the first line ill only contain the demographic so it is much easier as you can tell.	N/A

New processes

Now that all of the old process and objectives have been accounted for, like I said here I want to go into detail about the new objectives that will be included onto the system. These were thought about of from the start of the development up until now. The point of these processes are either to provide an aspect of the system that was forgotten during design or has been included for a newly suggested feature. While they will all be included for one reason or another I think it is important to give enough process a real justification on its inclusion on the system. However what lacks from this table is the justification column this is because these objectives are new and therefore don't have to be redesigned and so therefore will not need to be justified for why they need changing.

Process/ Objective number	Description	Justification	Array / Files Used
43. Add/Edit employee demographic information	The purpose of this objective is to allow the employees of the system all three included the ability to add and edit their personal information, to add to this it will also entail the 90 update period that the patient has to abide by. Once their information is inputted it is then written to their correct location so it can be accessed at a later point.	The reason that the process is now needed on the system is because of the inclusion of the new reflection I had with the system, this being that all entities should have amendable access to their information. Otherwise it would be pointless to have the information on the system if overtime it would become obsolete and incorrect	**(ConsultantID)_file.txt** **(StaffID)_file.txt** **Management.txt** The reason these files are used is because in order for the information to be saved they needed to be written to file. As there are different types of employees there is a file for each of the respective types of employee: consultant staff and management.
44. Logout Users	When the user is finished with the system and for examples sake are on a public computer they will want to log out of the system. In turn this will remove any information or variables that currently hold information about them and will prevent the user from re-entering the system without any credentials being entered. While the system saves automatically there is no penalty for logging out just after hitting save. However fields will be wiped if not saved correctly.	The reason this process is now being included is because to be honest was forgotten about the design. While obviously it wasn't forgotten on the prototype, as now I have the chance to formally include new objectives it seems now is the perfect time to include. The reason to include this feature is to allow public computers using this software the ability to logout without having to worry about their account.	No files or arrays are used

45. Remember Me	The purpose of this feature is to allow the user to help aid the constant user by allowing their username to be saved to as the remembered user. While this won't entail their password is remembered this will allow for some reduction time as their username will be already in the text field on launch of the system. This will be automatically changed if the remember me button is selected and someone else logs in.	This process is being included because it was an excellent piece of feedback given to me. The reason I want it as an objective is because it is a fairly large task and will need to be planned out. This is mainly due to the fact it deals a lot with file reading and writing and will be needed on the start of the systems launch regardless if the files empty.	**General.txt-** As of now the file only contains two fields the primary key of the user along with their password. The reason the file is needed is because as the system loads up the desired account that has remember me set up will need to have the credentials loaded to the fields. However as there could be possible hundreds of accounts it would be impossible to do so. To get around this, a single file is used which can be hardcoded to the system. Not only this but there is no need to find the old account and remove the remember me privilege as it would already be done.
46. Cancel bookings	The purpose of this process is to allow the user to cancel a booking when the date is within the boundaries of the accepted window. While there isn't a current punishment for not attending an appointment it obviously is not the most appreciated action as the hospital's time is being wasted. This is the reason the system was included, however, to prevent frustratingly late cancelations a deadline has been included to prevent this from occurring.	The reason I want this process is because, again while in the original design I believe this process was intended I think it was shoehorned into with the amending of the bookings, because of it made the objective too large on reflection. In terms of the actual process itself it should be included as the objective will be in charge of dealing with at least two files and will be important in terms of severity as it is now a key feature due to the fact that booking has now been made a main entity.	**(ConsultantID)_file.txt** **(StaffID)_file.txt** **(PatientID)_file.txt** The reason that these files are included is twofold. For the patient and consultant these actually need to store the information regarding the booking and any changes that are made to them, with the consultant and staff they will also hold the action log and will record any changes that occur to them. **Allbookings[]** as the deleting of bookings utilises the objective which returns the index of bookings it is most suitable to use this to just copy the list of bookings and exclude this array.

47. Determining a consultant for an admission	A new feature concepted in this document the process will be for the staff entity to determine the most suitable consultant for the patient's admission. While not the perfect idea it is to replace the old idea of having software reach the conclusion. However so the entity isn't completely dumbfounded the information about both the consultants and the admission will be on hand to make a judgment call, maybe even a recommendation from the system could be possible.	This process has now been included onto the system due to the fact the process has come into fruition as the staff entity has been repurposed somewhat. The purpose itself is important as it will be one of the initial phases of an admissions creation and could determine the treatment a patient receives.	**(PatientID)_file.txt** **(ConsultantID)_file.txt** The reason that these files are needed is because that once the staff entity has determined which consultant suits the patient the best it them needs to be recorded down. It needs to be in the patient's because it is where the admission is naturally stored. For the consultant's file it is here to improve access times it will also be stored here.
48. Read Patient information	For this process and subsequently 32 33 and 34 all these processes are for the retrieving of the information from their respective files. While a per entity process could have been utilised this will allow for all their information to be read from file and then assigned to the object allowing for the user to access their information effectively. For the patient their demographic, admissions along with bookings are all read from file to be assigned to individual objects respectively.	For this entity and a few others like it, the process has now been included due to the fact it is probably the most commonly used feature of the system. It wasn't on the old design document simply because the process wasn't thought of being that important, however this is no longer the case. The reason that these processes are now included is because they initialise the entity and therefore the user for the system. While It should have been in originally it is better to have it in now rather than not at all.	**(PatientID)_file.txt** For this file in particular the reason it is here is pretty obvious; this is down to the fact that in order for the entity to be declared and assigned attributes it first needs to be read from file, otherwise there will be no data to amend or use on the system. To add to this if any changes do occur this method can be recalled so that the newest data is used on the system.

49. Read Staff information	The purpose of this process like described in 31 will read from the respective file and will generate an object with the corresponding information for that entity, alongside this there will be other data such as the action log and other important information. For the staff entity the unique part about this would be that the patients associated with the employee. Once all the information is assigned to the object the system runs as normal.	For this entity and a few others like it, the process has now been included due to the fact it is probably the most commonly used feature of the system. It wasn't on the old design document simply because the process wasn't thought of being that important, however this is no longer the case. The reason that these processes are now included is because they initialise the entity and therefore the user for the system. While It should have been in originally it is better to have it in now rather than not at all.	**(StaffID)_file.txt** For this file in particular the reason it is here is pretty obvious; this is down to the fact that in order for the entity to be declared and assigned attributes it first needs to be read from file, otherwise there will be no data to amend or use on the system. To add to this if any changes do occur this method can be recalled so that the newest data is used on the system.
50. Read consultant Information	The purpose of this process like described in 31 will read from the respective file and will generate an object with the corresponding information for that entity, alongside this there will be other data such as the action log and other important information. For the consultant the unique part about this would be that the patients booking that they have with them. Once all the information is assigned to the object the system runs as normal.	For this entity and a few others like it, the process has now been included due to the fact it is probably the most commonly used feature of the system. It wasn't on the old design document simply because the process wasn't thought of being that important, however this is no longer the case. The reason that these processes are now included is because they initialise the entity and therefore the user for the system. While It should have been in originally it is better to have it in now rather than not at all.	**(ConsultantID)_file.txt** For this file in particular the reason it is here is pretty obvious; this is down to the fact that in order for the entity to be declared and assigned attributes it first needs to be read from file, otherwise there will be no data to amend or use on the system. To add to this if any changes do occur this method can be recalled so that the newest data is used on the system.

51. Read management information	The purpose of this process like described in 31 will read from the respective file and will generate an object with the corresponding information for that entity. As the management entity has no major attributes other than the demographic because of this the little number of attributes all the management entities might as well be converted at the same point of the systems launch	For this entity and a few others like it, the process has now been included due to the fact it is probably the most commonly used feature of the system. It wasn't on the old design document simply because the process wasn't thought of being that important, however this is no longer the case. The reason that these processes are now included is because they initialise the entity and therefore the user for the system. While It should have been in originally it is better to have it in now rather than not at all.	**Management.txt** For this file in particular the reason it is here is pretty obvious; this is down to the fact that in order for the entity to be declared and assigned attributes it first needs to be read from file, otherwise there will be no data to amend or use on the system. To add to this if any changes do occur this method can be recalled so that the newest data is used on the system.
52. Searching definition	The purpose of this process is to allow the user to search for the definition using a query. Unlike in pervious searches where the new redesign was including more than one search field, this objective only has one due to the nature of the data related to it. However one thing that separates this from the old version in the prototype is the inclusion of multiple results	The purpose of this process was because it was a feature that was included in the prototype as a quality of life aspect and was kept on due to the fact it worked really well. The other main reason I want this to be its own process is due to the fact it was also a criticised feature of the feedback, with not enough output being produced.	**Glossary_Of_Terms.txt** The reason the file is needed is because the definitions on the file are changing and because of this it would be easiest that they are in a separate file away from everything else on the system. As the number of fields per definition is minimum it makes searching pretty easy due to how fast items can be searched.

Process/ Objective number	Description	Justification	Array / Files Used
53. Sorting definitions	The purpose of the sort in this feature is to allow the user to sort the definitions into two distinct orders either the words are in ascending order from A to Z or descending order from Z to A. There will be no other sorting orders available.	Again similar to 52 this process has been made due to the fact that it was included on the prototype but not on the design document. The main reason I wanted this feature to be put down as an objective was because every other sort was included. When the project of the system is finished it will mean I have to discuss the effectiveness of every objective and as this feature was responded to by feedback I can talk about the improvements made to it.	**Glossary_Of_Terms.txt** **AllDefenitions[]** The reason that the array is needed is because in order for a sort to be performed on the data the elements need to be reliably swapped. The only way to effectively do this is by utilising an array. The file is needed because it will read the definitions from this location into the program.
54. Primary key generation	The purpose of this process is to produce unique fields that identify entities on the system. This is achieved by indicating initially what type of entity it is then followed by the user, which consist of their surname and a unique number. While all the keys are unique to some degree some are compound keys due to the fact that they need two unique keys.	Finally like other features on the section this is a feature that should have definitely been included on the design stage but was forgotten about. The reason however it is needed in general is because it will be the main reason entities are found/created/sorted /accessed on the system. The system will revolve around these fields so it is of the upmost importance they are now described as objectives.	**(ConsultantID)_file.txt** **(StaffID)_file.txt** **(PatientID)_file.txt** **Management.txt** Finally the reason that all 4 of the main file types are needed is because the field that is generated will either be the name of the file or will be used to identify sections in the file. Without having the files there would be no possible way that data is held on the system.

Redesigned Pseudocode for the processes of the system

Now that the current processes of the system have been redesigned and included, these changes will now be represented in a standardised format to allow others to see the changes in their full extent. While I have said some of these process have changed this does not mean they have been overhauled and so to fully show the changes to them the sections will be done in red, with the rest being the original code found in design. For objectives that have not been redesigned they will be left out as there is no need to make this document any longer and simply because they can be found in the design document.

Pseudocode for Objectives 1 to 10

1. Entering symptoms

```
START program
    SET Array symptoms[4]//creates an empty array for the patient to enter the symptoms
    SET Array listOfAreas[4]//creates an empty array for the patient to select body parts
    SET counter AS 0//sets the counter for the symptoms array as zero
    WHILE input IS NOT empty
        OUTPUT "Please enter symptoms you are experiencing"//asks patient for the
        symptoms they are experiencing
        Symptoms[counter] IS INPUT//input symptoms into the array
        SET listOfSympotms AS Symptoms[counter]
        Counter ++//increments
    END While//ends while loop
    WHILE input IS NOT empty
        listOfAreas AS [counter] IS INPUT//inputs areas affected into the array
        Counter ++//increments
    END While//ends while loop
    SET admission.listOfAreas AS listOfAreas//saves the admission attribute of the list of
    symptoms to the entity
    SET admission.ListOfSymptoms AS listOfSympotms//saves the admission attribute of the list
    of symptoms to the entity
END program
```

3. Generating New Patients

```
START program
    SET NEW user
    SET user.firstname AS INPUT//Sets the attribute of the new object to the input
    SET user.surname AS INPUT//Sets the attribute of the new object to the input
    SET user.DOB AS INPUT//Sets the attribute of the new object to the input
    SET user.gender AS INPUT//Sets the attribute of the new object to the input
    SET user.sex AS INPUT//Sets the attribute of the new object to the input
    SET User AS patient//Sets this new object of user as a patient as well
    ~~SET randomNUM (0,9999999)//generates a random number~~
    ~~SET patient.patientID AS P + user.surname(0,2) + randomNUM//creates a new ID for the~~
    ~~patient using their surname and a random number~~
```

SET patient.patientID AS **RUN OBJECTIVE 53(**patient)//Calls the objective which generates the primary key
SET **user**.*houseNum* **AS INPUT//**Sets the attribute of the new object to the input
SET **user**.*houseStreet* **AS INPUT//**Sets the attribute of the new object to the input
SET **user**.Postcode **AS INPUT//**Sets the attribute of the new object to the input
SET **user**.contactNumber **AS INPUT//**Sets the attribute of the new object to the input
SET **user**.nationality **AS INPUT//**Sets the attribute of the new object to the input
SET **user**.smoker **AS INPUT//**Sets the attribute of the new object to the input
SET **user**.drinker **AS INPUT//**Sets the attribute of the new object to the input
GENERATE NEW text file (patientID)_file.txt**//**creates a new text file for the patient's admissions
ADD New Patient **TO** AllPatients[]**//**Adds the patient to the list of patients
RUN WRITETOFILE**(**(patientID)_file.txt, New Patient **)//**writes the patient to file
RUN OBJECTIVE 13B(employeeID, actionToLocation, patientID, AdmissionID, OrginalData, NewData**)//**this passes the action just performed into the objective to add it to their action log

END program

4. Generating New Admissions

START program

RETREIVE patient**//**Finds the relevant patient to access the file
RETRIEVE symptoms[]**//**Finds the symptoms that they have entered
RETRIEVE diagnosis**//**Finds the corresponding diagnosis for the symptoms
~~**SET** filename **AS** patient.patientID + _Admissions**//**creates a suitable filename to find the actual file~~
RETREIVE filename.txt**//**Uses the filename to find the text file
~~**SET** randomNUM (0,9999999)**//**Generates a random number~~
~~**SET** admission.admissionID **AS** A + user.surname(0,2) + randomNUM creates a new ID for the patient's admisison using their surname and a random number~~
SET admission.ward **AS INPUT//**Sets the attribute of the new object to the input
~~**SET** admission.*consultantID* **AS INPUT//**Sets the attribute of the new object to the input~~
~~**SET** admission.*staffID* **AS INPUT//**Sets the attribute of the new object to the input~~
SET admission.active **AS TRUE//**Sets the attribute of the new object to the input
SET admission.listOfSymptoms **AS** symptoms[]**//**uses the imported list of symptoms and saves it to the variable
SET admission.currentDiagnosis **AS** diagnosis**//**uses the imported diagnosis and saves it to the variable
SET admission.admissionID AS **RUN OBJECTIVE 53(**admission)//Calls the objective which generates the primary key
RETIREVE admission.consultant **RUN OBJECTIVE 46**// calls objective which sets consutlant
RUN WRITETOFILE**(**PatientID.txt, **New** Admission **)//**Writes the new object to file
RUN OBJECTIVE 13B(employeeID, actionToLocation, patientID, AdmissionID, OrginalData, NewData**)//**this passes the action just performed into the objective to add it to their action log

END program

5. Booking a new appointment

START program

RETRIEVE availableDates(consultantID)**//**retrieves all the available dates for the consultant
OUPUT nonfull dates**//**outputs all the dates that are not full
SET NEW booking//creates a new object of booking
SET booking.dateOfNextAppointment **AS INPUT//**saves the date selected as that date
RETRIEVE availableTimes(consultantID,date)**//**retrieves all the available times from the selected date
OUTPUT nonbooked times**//**Outputs all the available timeslots
SET booking.timeOfNextAppointment **AS INPUT//**saves the time selected from the user's choice
SET room **AS INPUT//**saves the room inputted
SET AutomaticBooking **AS INPUT//**saves the room inputted
SET booking.refenerceNumber **AS RUN OBJECTIVE 53(**booking)//Calls the objective which generates the primary key
ADD NEW booking **TO** listofAppointments[]**//**Adds the booking to the consultants list
RUN WRITETOFILE**(**PatientID+"_file.txt", AdmissionID,booking **)//**Writes the new object to file
RUN WRITETOFILE**(**consultantID+"_file.txt", AdmissionID,booking **)//**Writes the new object to file
RUN OBJECTIVE 13B(employeeID, actionToLocation, patientID, AdmissionID, OrginalData, NewData**)//**this passes the action just performed into the objective to add it to their action log

END program

6. login users

START program

SET Loggedin **AS FALSE//**Sets the login credentials as false to loop the initial login screen
WHILE loggedin **IS FALSE//**loops the screen so allows for multiple attempts
THEN SET username **AS INPUT//**user enters initial credential
SET password **AS INPUT//**saves the users input to the variable
OPEN username+"_File.txt"**//** tries to open the file using the correct username
IF NOT FOUND//if an error is thrown(file doesn't exist) this is ran
THEN OUTPUT "wrong username"**//**informs the user that the username is wrong
ENDIF
ELSE//If their username is found then this is ran
THEN IF password **EQUALS** file.password**//**Checks to see if the password is correct
THEN GRANT ACCESS//allows them access to the system
SET loggedin **AS TRUE //**Exits the loop
RUN OBJECTIVE 44//runs the remember me process
ENDIF
ELSE//this is ran if the password is wrong

THEN OUTPUT”password is wrong”**//**informs user that password is wrong

ENDELSE

ENDELSE

ENDWHILE

END program

8. Generating New Employees

START program

RETRIEVE NEW user**//**Once a new user is generated, they will be retrieved

SET type **AS INPUT//**The user when determine if this new user is a consultant or staff

SET user **AS** Employee**//**The system will set the new user as an employee

SET user.sex AS INPUT//Sets the attribute of the new object to the input

SET user.*houseNum* **AS INPUT//**Sets the attribute of the new object to the input

SET user.*houseStreet* **AS INPUT//**Sets the attribute of the new object to the input

SET user.Postcode **AS INPUT//**Sets the attribute of the new object to the input

SET user.contactNumber **AS INPUT//**Sets the attribute of the new object to the input

SET user.nationality **AS INPUT//**Sets the attribute of the new object to the input

SET user.smoker **AS INPUT//**Sets the attribute of the new object to the input

SET user.drinker **AS INPUT//**Sets the attribute of the new object to the input

~~**SET** randomNUM (0,9999999)**//**generates a random number~~

~~**SET** employee**.**employeeID **AS** E +~~ user.surname(0,2) + randomNUM**//**System assigns the user a employeeID

SET employee**.**wagePerWeek **AS INPUT//**Sets the attributes for the wage as the input

SET employee**.**hoursPerWeek **AS INPUT//**Saves the employees hourly work rate from the input

SET employee.employeeID AS **RUN OBJECTIVE 53(**employee)//Calls the objective which generates the primary key

IF type **IS** consultant**//**If management decides to save the employee as a consultant the following will set the rest of the attributes

THEN SET employee **AS** consultant**//**updates the entity to obtain the new attributes

~~**SET** randomNUM (0,9999999)**//**generates a random number~~

~~**SET** consultant**.**staffID **AS** C + user.surname(0,2) +randomNUM**//**generates the id for the entity~~

SET consutlant.consultantID AS **RUN OBJECTIVE 53(**consultant)//Calls the objective which generates the primary key

GENERATE NEW text file (consultantID)_file.txt**//**creates a new text file for the consultant

SET Ward **AS INPUT//**saves the attribute as the input they have entered

SET archived **AS FALSE//**saves that the archived field as false and they are currently working

SET counter **AS 0//**adds an empty counter for array indexes in the iterative section

DO//Will immediately loop until a condition has been satisfied

SET practices **AS INPUT////**saves the attribute as the input they have entered

SET practisesList[counter] **AS** practices**//**will save the input to the array

Counter**++//**Will incremient and move onto the next index

UNTILL INPUT IS NULL//termination condition until no more input is left to be entered

ADD NEW consultant **TO** *allConsultants[]***//**adds the new entity to the array of entities

RUN WRITETOFILE**(**consultantID)_file.txt, **New** Consultant **)//**Writes the entity to staff file also

END IF

ELSE IF type **Is** staff**//** If management decides to save the employee as staff the following will set the rest of the attributes

THEN SET user **AS** staff**//** updates the entity to obtain the new attributes

~~**SET** randomNUM (0,9999999)**//** generates a random number~~

~~**SET** staff.*staffID* **AS** S + user.surname(0,2) + randomNUM**//** generates the id for the entity~~

SET staff.*staffID*-AS **RUN OBJECTIVE 53(**staff)//Calls the objective which generates the primary key

GENERATE NEW text file (*staffID*)_file.txt**//**creates a new text file for the consultant

SET archived **AS FALSE////**saves that the archived field as false and they are currently working

ADD NEW staff **TO** *allStaff[]***//**adds the new entity to the array of entities

RUN WRITETOFILE**((***staffID*)_file.txt, **New** staff **)//** Writes the entity to staff file also

RUN OBJECTIVE 13B(employeeID, actionToLocation, patientID, AdmissionID, OrginalData, NewData**)**

//this passes the action just performed into the objective to add it to their action log

ENDIF

END program

Pseudocode for Objectives 11 to 20

13A. Read transaction log from file

START program

RETRIEVE employee**//**this retrieves the patient the management entity wishes to view

RETRIEVE startPostision**//**This retrieves the start position of the transaction log the management entity wants to start at

RETRIEVE endPostision**//** This retrieves the start position of the transaction log the management entity wants to end at

IF user IS CONSULTANT//the selection here is to determine the correct state of the employee determining the right filename

THEN SET filename **AS** consultant.*consultantID***//**this sets the filename to the correct one associated with the user

ENDIF

ELSE

THEN SET filename **AS** staff.*staffID***//**this sets the filename to the correct one associated with the user

ENDELSE

~~**SET** filename **AS** employee.*employeeID***//**this sets the filename to the correct one associated with the patient~~

OPEN filename
FOR line **FROM** startPostision **TO** endPostision**//**Declaration of the for loop running from the start point to the end point for every line. This will go through every action performed by the employee between the selected date
THEN READ filename_Actionlog.txt**//**This will then read that line
SET text **AS Line//**The line read is then saved to the text variable
SET decryptedtext **AS (RUN OBJECTIVE 38(**text**))//**Encrypts the data allowing it to be securely written to file
OUTPUT decryptedtext**//**the text is then outputted to the management entity
NEXT line**//**The line now increments to the next one
CLOSE filename
END program

13B. write transaction log to file

START program
IF user IS CONSULTANT//the selection here is to determine the correct state of the employee determining the right filename
THEN SET filename **AS** consultant.*consultantID***//**this sets the filename to the correct one associated with the user
ENDIF
ELSE
THEN SET filename **AS** staff.*staffID***//**this sets the filename to the correct one associated with the user
ENDELSE
SET localDate**//**sets the date the action was performed
SET localTime**//**sets the time the action occurred
RETRIEVE actionToLocation**//**retrieves the action and where it was applied to
RETRIEVE patientID**//**retrieves the patient ID to allow for tracing of who's information was affected
RETRIEVE AdmissionID**//**retrieves the admission ID to trace what admission was affected
RETRIEVE OrginalData**//**retrieves the data in the field before it was amended
RETRIEVE NewData**//**retrieves the new data that was added
SET text **AS** localDate + "," + localTime + "," + actionToLocation + "," + patientID + "," + AdmissionID + "," + OrginalData + "," + NewData**//**Concatenates all the data retrieved and saves it as a single string
SET filename **AS** filename +"_file.txt"**//**Sets the correct filename to that of the employeeID
RETRIEVE filename**//**retrieves the action log for the individual employee
OPEN filename**//**opens the file allowing for amending of data
SET encryptedtext **AS (RUN OBJECTIVE 38(**text**))//**encrypts the text
WRITE encryptedtext **TO** filename**//**Writes the concatenated data to file, as it is ordered by date then time no ordering issues occur so can be added to the bottom of the file
CLOSE filename**//**
END program

17. Add archived notes from old system

START program

RETRIEVE patient**//**retrieves the patient desired
SET filename **AS** (PatientID)+"_file.txt"**//**sets up the filename of the text file where all other scans are saved
SET admisisonID **AS INPUT//**Saves the admission ID so it can be used to distinguish scan names between admissions
~~**SET** scanName **AS INPUT//**Sets the scan name of the scan so it can be used to identify its contents~~
SET AbsolutePath **AS INPUT//**The staff entity will select the file path the document is from
SET date **AS INPUT//**declares the date of the document to distinguish documents in the same admission that are of the same type
SET time **AS INPUT//**declares the time of the document to distinguish documents in the same admission that are of the same type
SET AuthorOfDocument **AS INPUT**//Assigns the correct author of the document
SET numberOfPages**AS INPUT**//Assigns the correct number of pages of the document
SET doctype **INPUT**//Assigns the type of the document
SET notes **INPUT**//Assigns any notes that help aid readability of the document
RETRIEVE SCAN//this will retrieve the scan of the document
~~**SET** scan.filename **AS** admisisonID + "_" + scanName + "_" + date**//**concatenates the three attributes and saves them as a string~~
OPEN filename**//**opens the file to allow for it to be used
SET employeeID**//**sets the employeeID to the employee using the system
SET actionTOloaction **AS** adding old document to admission**//**saves the action the employee is doing to adding old documentation
SET documentdocumentID AS **RUN OBJECTIVE 53(**document)//Calls the objective which generates the primary key
SET encryptedtext **AS (RUN OBJECTIVE 38(**document**))//writes the entirety of the entity to the encryption algorithm**
WRITE encryptedtext **TO** filename**//**Writes the concatenated string to file
SET originalData **AS NULL//**AS no original data was amended this is set to null
SET NewData **AS** scan.filename**//**sets the new data to the scan's filename
CLOSE filename**//**closes the file to prevent it from corrupting
RUN OBJECTIVE 13B(employeeID, actionToLocation, patientID, AdmissionID, OrginalData, NewData**)//**this passes the action just performed into the objective to add it to their action log
END program

18. Amend bookings

START program

RETRIEVE patient**//**finds the patient who's booking needs amending

IF date - appointmentDate **IS GREATER THAN 12 HOURS//**checks to see if the appointment is before the deadline

THEN SET field **AS INPUT//**the field wanting to be edited is chosen

IF field **IS** timeOfNextApp**//**selection determining if this field needs to be edited

THEN SET orignialData **AS** Admisson.timeOfNextApp**//**updates the original data variable to add to the action log

SET Admisson.timeOfNextApp **AS INPUT//**updates the field to contain the desired data

SET newData AS Admisson.timeOfNextApp**//**uses the new data and saves it to an external variable

END IF

ELSE IF field **IS** dateOfNextApp**//**selection determining if this field needs to be edited

THEN SET orignialData **AS** Admisson.dateOfNextApp**//**updates the original data variable to add to the action log

SET Admisson.dateOfNextApp **AS INPUT//**updates the field to contain the desired data

SET newData AS Admisson.dateOfNextApp**//**uses the new data and saves it to an external variable

END IF

ELSE IF field **IS** room**//**selection determining if this field needs to be edited

THEN SET orignialData **AS** Admisson.dateOfNextApp**//**updates the original data variable to add to the action log

SET Admisson.room **AS INPUT//**updates the field to contain the desired data

SET newData AS AS Admisson.dateOfNextApp**//**uses the new data and saves it to an external variable

END IF

ELSE IF field **IS** ward**//**selection determining if this field needs to be edited

THEN SET orignialData **AS** Admisson.ward**//**updates the original data variable to add to the action log

SET Admisson.ward **AS INPUT//**updates the field to contain the desired data

SET newData AS Admisson.ward**//**uses the new data and saves it to an external variable

END IF

ELSE IF field **IS** consultant**//**selection determining if this field needs to be edited

THEN SET orignialData **AS** Admisson.consultant**//**updates the original data variable to add to the action log

SET Admisson.consultant **AS INPUT//**updates the field to contain the desired data

SET newData AS Admisson.consultant**//**uses the new data and saves it to an external variable

END IF

ELSE IF field **IS** AutomaticBooking**//**selection determining if this field needs to be edited

```
                THEN SET orignialData AS booking.AutomaticBooking//updates the original data variable to add to the action log
                SET booking.AutomaticBooking AS INPUT//updates the field to contain the desired data
                SET newData AS booking.AutomaticBooking//uses the new data and saves it to an external variable
            END IF
            SET actionToLocation AS amendment to booking field//sets the action to amending the booking field
            SET newText AS patientID + "," + admissionID + "," + timeOfNextApp + "," + dateOfNextApp + "," + ward + "," + room//concatenates all the fields together onto one string
            RUN WRITETOFILE(PatientID+"_file.txt", AdmissionID,booking  )//Writes the new object to file
            RUN WRITETOFILE(consultantID+"_file.txt", AdmissionID,booking )//Writes the new object to file
            RUN OBJECTIVE 13B(employeeID, actionToLocation, patientID, AdmissionID, OrginalData, NewData)//this passes the action just performed into the objective to add it to their action log

            CLOSE filename//closes the file to prevent corruption
            RUN OBJECTIVE 13B(employeeID, localDate, localTime, actionToLocation, patientID, AdmissionID, OrginalData, NewData)//this passes the action just performed into the objective to add it to their action log
        ENDIF
END program
```

22. Amend demographic information

```
START program
    IF user IS employee
        THEN SET original data AS patientDemographic//sets the old data to what is currently on file
    ENDIF
    SET patientID AS INPUT//Declares the patientID from the input
    SET patient AS (RUN OBJECTIVE 41(patientID))//Runs the objective to retrieve the demographic
    RETURN LINE//returns the line the patient was in on the file
    SET patientDetails[] AS (split.patientDetails[])//Splits up the line to hold the individual data in each index
    SET detailToChange AS INPUT//User sets the field they wish to change
    IF detailToChange is patientID//selection to see if they want to amend this field
        THEN SET patientDetails[0] AS INPUT//sets the correct field in the array to the new data
    ENDIF
    ELSE IF detailToChange is password//selection to see if they want to amend this field
        THEN SET patientDetails[2] AS INPUT//sets the correct field in the array to the new data
```

ENDIF
ELSE IF detailToChange **is** surname**//**selection to see if they want to amend this field
THEN SET patientDetails[2] **AS INPUT//**sets the correct field in the array to the new data
ENDIF
ELSE IF detailToChange **is** firstname**//**selection to see if they want to amend this field
THEN SET patientDetails[3] **AS INPUT//**sets the correct field in the array to the new data
ENDIF
ELSE IF detailToChange **is** houseNum**//**selection to see if they want to amend this field
THEN SET patientDetails[4] **AS INPUT//**sets the correct field in the array to the new data
ENDIF
ELSE IF detailToChange **is** houseStreet**//**selection to see if they want to amend this field
THEN SET patientDetails[5] **AS INPUT//**sets the correct field in the array to the new data
ENDIF
ELSE IF detailToChange **is** Postcode**//**selection to see if they want to amend this field
THEN SET patientDetails[6] **AS INPUT//**sets the correct field in the array to the new data
ENDIF
ELSE IF detailToChange **is** contactNumber**//**selection to see if they want to amend this field
THEN SET patientDetails[7] **AS INPUT//**sets the correct field in the array to the new data
ENDIF
ELSE IF detailToChange **is** nationality**//**selection to see if they want to amend this field
THEN SET patientDetails[8] **AS INPUT//**sets the correct field in the array to the new data
ENDIF
ELSE IF detailToChange **is** bloodtype**//**selection to see if they want to amend this field
THEN SET patientDetails[9] **AS INPUT//**sets the correct field in the array to the new data
ENDIF
ELSE IF detailToChange **is** smoker**//**selection to see if they want to amend this field
THEN SET patientDetails[10] **AS INPUT//**sets the correct field in the array to the new data
ENDIF
ELSE IF detailToChange **is** drinker**//**selection to see if they want to amend this field
THEN SET patientDetails[11] **AS INPUT//**sets the correct field in the array to the new data
ENDIF
ELSE IF detailToChange **is** disability**//**selection to see if they want to amend this field
THEN SET patientDetails[12] **AS INPUT//**sets the correct field in the array to the new data
ENDIF
ELSE IF detailToChange **is** numOfAdmissions**//**selection to see if they want to amend this field

```
        THEN SET patientDetails[13] AS INPUT//sets the correct field in the array to the new
        data
ENDIF
ELSE IF detailToChange is D.O.B//selection to see if they want to amend this field
        THEN SET patientDetails[14] AS INPUT//sets the correct field in the array to the new
        data
ENDIF
ELSE IF detailToChange is Religion//selection to see if they want to amend this field
        THEN SET patientDetails[15] AS INPUT//sets the correct field in the array to the new
        data
ENDIF
ELSE IF detailToChange is Allergies//selection to see if they want to amend this field
        THEN SET patientDetails[16] AS INPUT//sets the correct field in the array to the new
        data
ENDIF
ELSE IF detailToChange is Gender//selection to see if they want to amend this field
        THEN SET patientDetails[17] AS INPUT//sets the correct field in the array to the new
        data
ENDIF
ELSE IF detailToChange is diablities//selection to see if they want to amend this field
        THEN SET patientDetails[18] AS INPUT//sets the correct field in the array to the new
        data
ENDIF
ELSE IF detailToChange is carer//selection to see if they want to amend this field
        THEN SET patientDetails[19] AS INPUT//sets the correct field in the array to the new
        data
ENDIF
ELSE IF detailToChange is translator//selection to see if they want to amend this field
        THEN SET patientDetails[20] AS INPUT//sets the correct field in the array to the new
        data
ENDIF
ELSE IF detailToChange is Sex//selection to see if they want to amend this field
        THEN SET patientDetails[21] AS INPUT//sets the correct field in the array to the new
        data
ENDIF
ELSE IF detailToChange is county//selection to see if they want to amend this field
        THEN SET patientDetails[22] AS INPUT//sets the correct field in the array to the new
        data
ENDIF
ELSE IF detailToChange is DaysSinceLastUpdate//selection to see if they want to amend this
field
        THEN SET patientDetails[23] AS INPUT//sets the correct field in the array to the new
        data
ENDIF
FOR counter FROM 0 TO 23//creates a loop that will iterate 12 times for a total of 13 runs
        THEN SET text AS text + ”,” + patientDetails[counter]//merges the current text
        variable with the next data in the array, essentially re-joins the array back together
        just replaces the old data
```

```
        END FOR
        IF user IS employee
                THEN SET newData AS text//sets the new text as the new attributes
                RUN OBJECTIVE 13B(employeeID, localDate, localTime, actionToLocation, patientID,
                AdmissionID, OrginalData, NewData)//this passes the action just performed into the
                objective to add it to their action log
        ENDIF
        SET encryptedtext AS RUN OBJECTIVE 37(text)//encrypts the data
        OPEN PatientID+_file.txt //opens the text file containing all demographic information
        WRITE encryptedtext TO PatientID+_file.txt AT line//writes the encrypted data to file
        CLOSE PatientID+_file.txt //closes file to prevent corruption
END program
```

24. View bookings in its entirety

```
START program
        IF user IS consultant//the system is determining whether or not the user is a consultant
                THEN SET filename AS (consultant+"_file.txt")//sets the correct filename
        ENDIF
        IF user IS patient//the system is determining whether or not the user is a patient
                THEN SET filename AS (patient+"_file.txt")//sets the correct filename
        ENDIF
        SET booking AS (RUN OBJECTIVE 40(patientID))//finds the patient in the file we are looking
        for along with them we receive all their attributes
        OUPTUT booking.timeOfNextApp//outputs to the user the attribute
        OUTPUT booking.dateOfNextApp//outputs to the user the attribute
        OUTPUT booking.ward//outputs to the user the attribute
        OUTPUT booking.room//outputs to the user the attribute
        OUTPUT admission.consultantID//outputs to the user the attribute
        OUTPUT admission.admissionID//outputs to the user the attribute
        OUTPUT booking.bookingreference//outputs to the user the attribute
        OUTPUT booking.AutomaticBooking//outputs to the user the attribute
END program
```

25. Add bookings

START program
SET field **AS INPUT//**the field wanting to be edited is chosen
IF field **IS** timeOfNextApp**//**selection determining if this field needs to be edited
THEN SET booking.timeOfNextApp **AS INPUT//**updates the field to contain the desired data
END IF
ELSE IF field **IS** dateOfNextApp**//**selection determining if this field needs to be edited
THEN SET booking.dateOfNextApp **AS INPUT//**updates the field to contain the desired data
END IF
ELSE IF field **IS** room**//**selection determining if this field needs to be edited
THEN SET booking.room **AS INPUT//**updates the field to contain the desired data
END IF
ELSE IF field **IS** AutomaticBooking**//**selection determining if this field needs to be edited
THEN SET booking.AutomaticBooking **AS INPUT//**updates the field to contain the desired data
END IF
SET BookingReferenceAS **RUN OBJECTIVE 53(**booking)//Calls the objective which generates the primary key

SET newText **AS**
(*BookingReference*)+,+(timeOfNextApp)+,+(dateOfNextApp)+,+(*AutomaticBooking)+,+(room*)**//**concatenates all the fields together onto one string
SET filename **AS** (PatientID+"_filename.txt")**//**sets the correct filename
OPEN filename//opens the file to allow amendments
WRITE text **TO** filename(Line)**//**writes the new booking to file
SET filename **AS** (consultantID+"_filename.txt")**//**sets the correct filename
OPEN filename//opens the file to allow amendments
WRITE text **TO** filename(Line)**//**writes the new booking to file
CLOSE filename**//**closes the file to prevent corruption
END program

30. View patient Demographic information

START program
SET patient **AS RUN OBJECTIVE(47)**
OUTPUT patient.patientID**//**outputs to the user the attribute
OUTPUT patient.surname**//**outputs to the user the attribute
OUTPUT patient.firstname**//**outputs to the user the attribute
OUTPUT patient.houseNum**//**outputs to the user the attribute
OUTPUT patient.houseStreet**//**outputs to the user the attribute
OUTPUT patient.Postcode**//**outputs to the user the attribute
OUTPUT patient.contactNumber**//**outputs to the user the attribute
OUTPUT patient.nationality**//**outputs to the user the attribute
OUTPUT patient.bloodtype**//**outputs to the user the attribute
OUTPUT patient.smoker**//**outputs to the user the attribute

OUTPUT patient.drinker**//**outputs to the user the attribute
OUTPUT patient.disability**//**outputs to the user the attribute
~~**OUTPUT** patient.numOfAdmissions**//**outputs to the user the attribute~~
OUTPUT patient.D.O.B**//**outputs to the user the attribute
OUTPUT patient.Religion**//**outputs to the user the attribute
OUTPUT patient.Allergies**//**outputs to the user the attribute
OUTPUT patient.Gender**//**outputs to the user the attribute
OUTPUT patient.carer**//**outputs to the user the attribute
OUTPUT patient.translator**//**outputs to the user the attribute
OUTPUT patient.Sex**//**outputs to the user the attribute
OUTPUT patient.county**//**outputs to the user the attribute
END program

Pseudocode for Objectives 31 to 40

33. Add Admission information

START program
RETRIEVE patient **FROM** allPatients[]**//**the desired patient is retrieved from the array containing all patients
SET allAdmisisons[]**//**initialises the array containing all the admissions
SET filename **AS** ((patientID)+"_filename.txt")**//**concatenates the information creating the text filename
SET desiredAdmission **AS INPUT//**the user enters the desired admission
~~**OPEN** filename**//**opens the file to allow it to be amended~~
~~**DO**~~
~~**THEN READ LINE//**the line is read~~
~~**SET** decryptedtext **AS RUN OBJECTIVE 38**(line)**//**the line is decrypted~~
~~**SET** allAdmisisons[**lineCounter**] **AS** decryptedtext**//**the line is then added to the array~~
~~**SET** lineCounter **AS** lineCounter**++//**the index counter increments~~
~~**IF LINE IS** desiredAdmission**//**selection determining if the admission the user enters is in the array~~
~~**THEN SET** index **AS** lineCounter**//**the line containing the desired admission is saved to the variable index~~
~~**ENDIF**~~
~~**UNTIL LINE IS NULL//**this loop keeps iterating until the file contains no more text~~
~~**THEN SET** admission **AS** alladmissions[index]**//**the admission is then set to the correct one containing all the relevant information~~
IF Admission**.**consultantID **IS EMPTY**
THEN SET Admission.consultantID **AS INPUT//**At this point the new data is filled in
SET NewData **AS** Admission.consultantID **//**the new data is saved to the attribute for the transaction log
SET originalData **AS NULL//**As no original data existed the attribute for the transaction log is set to null
SET actionToLocation **AS** "Adding to admission"**//**the attribute for the attribute is saved to contain the correct information

ENDIF

IF Admission**.**numOfDocuments **IS EMPTY**

THEN SET Admission.numOfDocuments **AS INPUT//**At this point the new data is filled in

SET NewData **AS** Admission.numOfDocuments **//**the new data is saved to the attribute for the transaction log

SET originalData **AS NULL//**As no original data existed the attribute for the transaction log is set to null

SET actionToLocation **AS** "Adding to admission"**//**the attribute for the attribute is saved to contain the correct information

ENDIF

IF Admission**.**ward **IS EMPTY**

THEN SET Admission.ward **AS INPUT//**At this point the new data is filled in

SET NewData **AS** Admission.ward **//**the new data is saved to the attribute for the transaction log

SET originalData **AS NULL//**As no original data existed the attribute for the transaction log is set to null

SET actionToLocation **AS** "Adding to admission"**//**the attribute for the attribute is saved to contain the correct information

ENDIF

IF Admission**.**listOfSymptoms **IS EMPTY**

THEN SET Admission.listOfSymptoms **AS INPUT//**At this point the new data is filled in

SET NewData **AS** Admission.listOfSymptoms **//**the new data is saved to the attribute for the transaction log

SET originalData **AS NULL//**As no original data existed the attribute for the transaction log is set to null

SET actionToLocation **AS** "Adding to admission"**//**the attribute for the attribute is saved to contain the correct information

ENDIF

IF Admission**.**archived **IS EMPTY**

THEN SET Admission.archived **AS INPUT//**At this point the new data is filled in

SET NewData **AS** Admission.archived **//**the new data is saved to the attribute for the transaction log

SET originalData **AS NULL//**As no original data existed the attribute for the transaction log is set to null

SET actionToLocation **AS** "Adding to admission"**//**the attribute for the attribute is saved to contain the correct information

ENDIF

IF Admission**.**dateAdmissionCreated **IS EMPTY**

THEN SET Admission.dateAdmissionCreated **AS INPUT//**At this point the new data is filled in

SET NewData **AS** Admission.dateAdmissionCreated **//**the new data is saved to the attribute for the transaction log

SET originalData **AS NULL//**As no original data existed the attribute for the transaction log is set to null

SET actionToLocation **AS** "Adding to admission"**//**the attribute for the attribute is saved to contain the correct information

```
        ENDIF
        IF Admission.currentDiagnosis IS EMPTY
            THEN SET Admission.currentDiagnosis AS INPUT//At this point the new data is filled in
            SET NewData AS Admission.currentDiagnosis  //the new data is saved to the attribute for the transaction log
            SET originalData AS NULL//As no original data existed the attribute for the transaction log is set to null
            SET actionToLocation AS "Adding to admission"//the attribute for the attribute is saved to contain  the correct information
        ENDIF
        IF Admission.areasAffected   IS EMPTY
            THEN SET Admission.areasAffected AS INPUT//At this point the new data is filled in
            SET NewData AS Admission.areasAffected  //the new data is saved to the attribute for the transaction log
            SET originalData AS NULL//As no original data existed the attribute for the transaction log is set to null
            SET actionToLocation AS "Adding to admission"//the attribute for the attribute is saved to contain  the correct information
        ENDIF
        FOR counter UNTILL NULL//declares a counter that runs for an iteration of 10 times
            SET text AS AdmissionFiled+ ","//the text for all the array is concatenated to add new information to the string
        NEXT counter//the counter increments
        SET encryptedtext AS RUN OBJECTIVE 37(text)//the concatenated string is then encrypted using the objective
        WRITE encryptedText TO filename AT LINE//this newly encrypted text is then wrote to file
        CLOSE filename//the file is closed to prevent corruption
        RUN OBJECTIVE 13B(employeeID, actionToLocation, patientID, AdmissionID, OrginalData, NewData)//this passes the action just performed into the objective to add it to their action log
END program
```

34. Edit Admission information

```
START program
        RETRIEVE patient FROM allPatients[]//the desired patient is retrieved from the array containing all patients
        SET allAdmisisons[]//initialises the array containing all the admissions
        SET filename AS ((PatientID)+"_file.txt")//concatenates the information creating the text filename
        SET desiredAdmission AS INPUT//the user enters the desired admission
        OPEN filename//opens the file to allow it to be amended
        DO
            THEN READ LINE//the line is read
            SET decryptedtext AS RUN OBJECTIVE 37(line)//the line is decrypted
```

```
        SET allAdmisisons[lineCounter] AS decryptedtext//the line is then added to the
        array
        SET lineCounter AS lineCounter++//the index counter increments
        IF LINE IS desiredAdmission//selection determining if the admission the user enters
        is in the array
                THEN SET index AS lineCounter//the line containing the desired admission is
                saved to the variable index
        ENDIF
UNTIL LINE IS NULL//this loop keeps iterating until the file contains no more text
THEN SET admission AS alladmissions[index]//the admission is then set to the correct one
containing all the relevant information
SET field AS INPUT//the field wanting to be edited is set as input
IF field IS ward//Checks to see if the field wanting changed is acceptable
        THEN ACCEPT INPUT//allows the field to be amended
ENDIF
IF field IS Consultant//Checks to see if the field wanting changed is acceptable
        THEN ACCEPT INPUT//allows the field to be amended
ENDIF
IF field IS active//Checks to see if the field wanting changed is acceptable
        THEN ACCEPT INPUT//allows the field to be amended
ENDIF
IF field IS symptoms//Checks to see if the field wanting changed is acceptable
        THEN ACCEPT INPUT//allows the field to be amended
ENDIF
IF field IS currentDiagnosis//Checks to see if the field wanting changed is acceptable
        THEN
ENDIF
IF field IS archived//Checks to see if the field wanting changed is acceptable
        THEN ACCEPT INPUT//allows the field to be amended
ENDIF
IF field IS areasAffected//Checks to see if the field wanting changed is acceptable
        THEN ACCEPT INPUT//allows the field to be amended
ENDIF
ELSE//If no valid field is entered this will run
        THEN OUPUT field not recognised//informs the user that the field does not exist
ENDELSE
SET unvalidatedData AS INPUT//the new data to be added is set as input
SET validated AS RUN OBJECTIVE 23(unvalidatedData)//The Boolean attribute is set to true
or false if correctly or incorrectly validates respectively
IF validated IS TRUE//if the attribute is accepted the selection condition is satisfied
        THEN SET oldData AS admission.feild//before updating the new field the old data is
        saved for the action log
        SET newData AS unvalidatedData//Once the new data has been validated it has
        been saved to the transaction log attribute
        SET allAdmissions[index] AS newData//the new data now overwrites the old data
        allowing for an update to occur
        SET  actionToLocation AS "Adding to Admision"//the action to location attribute is
        updated to be used in the action log
```

ENDIF
FOR feild **IN RANGE** 9**//**a for counter runs for the number of attributes on the line normally
SET text **AS** alladmissions[feild]**+ “,”//**the new text to be added is concatenated to be written to the line
NEXT counter**//**the next attribute is then used by the index pointer
SET encryptedtext **AS RUN OBJECTIVE 37**(text)**//**the line is then encrypted using the objective and using the text as a parameter
WRITE encryptedText **TO** filename **AT LINE//**the new text now overwrites the information previously found at the beginning
CLOSE filename**//**the file is closed to prevent data corruption
RUN OBJECTIVE 13B(employeeID, actionToLocation, patientID, AdmissionID, OrginalData, NewData**)//**this passes the action just performed into the objective to add it to their action log
END program

35. Add notes

START program
SET docType **AS INPUT//**the user decides what document they want to use
SET filename **AS “**((patientID)+”_file.txt.”)”**//**information regarding the patient is retrieved and the respective filename is created
OPEN filename**//**the file is opened to allow new documents to be added
SET oldData **AS NULL//**as the document is new no old data is available and so field is set as null
THEN SET date **AS INPUT//**user sets start date for when the document was written
SET time **AS INPUT//**user sets time for when the document was written
SET documentID **AS RUN OBJECTIVE 53(document)//**Runs the objective which generates a new primary key for the entity
~~**SET** consultantID **AS** consutltant.consultantID**//**the consultants id is retrieved from the user’s attribute~~
SET patientID **AS** patient.patientID**//**the patient’s patient id is also retrieved from the patient’s attribute
SET summary **AS INPUT//**allows the user to determine the summary of the document
SET documentOfAuthor **AS INPUT//**determines the correct author of the system
IF doctype **IS** prescription**//**if the user wants this document then this will run
SET dataMedication **AS INPUT//**the intended input is entered by the consultant
SET validated **AS RUN OBJECTIVE 23A**(dataMedication)**//**it is then validated too make sure it is acceptable
IF validated **IS TRUE//**If it is allowed to be used then this is ran
THEN SET medication **AS** dataMedication**//**the new field in the document is now ran
ENDIF
SET dataDosage **AS INPUT//**the intended input is entered by the consultant
SET validated **AS RUN OBJECITVE 23.A(**Data**) AND RUN OBJECTIVE 23.D(**Data,start,end**) AND RUN OBJJECTIVE// 23.E(**dataDosage listOFDosages[]**)//**this selection statement passes through out the new data through all three of the validation functions making sure that they are correct if they all return true or accept the condition is met

IF validation **IS TRUE//**If it is allowed to be used then this is ran

THEN SET dosage **AS** dataDosage**//**once validated the actual field is updated to contain the user's input

ENDIF

SET dataIntaketime **AS INPUT//**the intended input is entered by the consultant

SET validated **AS RUN OBJJECTIVE 23.E(**Data listOFDates[]**) AND RUN OBJECTIVE 23.D(**dataIntaketime ,start,end**)//**this selection statement passes through out the new data through all two of the validation functions making sure that they are correct if they all return true or accept the condition is met

IF validated **IS TRUE//**If it is allowed to be used then this is ran

THEN SET Intaketime **AS** dataIntaketime**//**once validated the actual field is updated to contain the user's input

ENDIF

SET datadateOfNextDispatch **AS INPUT//**the intended input is entered by the consultant

SET validated **AS RUN OBJECITVE 23.B(**Data, **string) AND RUN OBJECTIVE 23.D(**Data,start,end**)//**the data is entered and checked to see if acceptable

IF validated **IS TRUE//**If it is allowed to be used then this is ran

THEN SET dateOfNextDispatch **AS** datadateOfNextDispatch **//**once validated the actual field is updated to contain the user's input

ENDIF

~~**SET** newdata **AS** doctype+","+date+","+ time +","+documentNum+","+ consultantID +","+PatientID +","+medicationdosage**//**saves the new data field for the action log as a concatenation of all the fields~~

SET ActionToLocation **AS** adding to prescription**//**updates the action log field to indicate what action has occurred

ENDIF

IF doctype **IS** Consultant Notes**//**if the user wants this document then this will run

THEN SET notes **AS INPUT//**the intended input is entered by the consultant

~~**SET** newdata **AS** doctype+","+date+","+ time +","+documentNum+","+ consultantID +","+ notes**//**saves the new data field for the action log as a concatenation of all the fields~~

SET ActionToLocation **AS** adding to Consultant Notes**//**updates the action log field to indicate what action has occurred

ENDIF

ELSE IF doctype **IS** Test Results**//**if the user wants this document then this will run

THEN SET testResults **AS INPUT//**the intended input is entered by the consultant

~~**SET** newdata **AS** doctype+","+date+","+ time +","+documentNum+","+ consultantID +","+ testResults**//**saves the new data field for the action log as a concatenation of all the fields~~

SET ActionToLocation **AS** adding to Test Results**//**updates the action log field to indicate what action has occurred

ENDIF

ELSEIF doctype **IS** oldDocument**//**if the user wants this document then this will run

THEN RUN OBJECTIVE 17

ENDELSE

SET numberOfPages **AS document.numberOfPages//**determines the correct number of pages

```
        SET hospital AS "EUXTON"//writes the hardcoded value to file
        SET newdata AS
        (documentID)+","+(dateOfDocumentCreation)+","+(timeOfDocumentCreation)+","+(doctyp
        e)+","+(medicationName)+","+(medicationDosage)+","+(medicationIntakeTime)+","+(medic
        ationDateOfNextDispatch)+","+(notes)+","+(testResults)+","+(Authorofdocument)+","+(Hos
        pital)+","+(AbsolutePath)+","+(numberOfPages) //saves the new data field for the action log
        as a concatenation of all the fields
        RUN OBJECTIVE 13B(employeeID, actionToLocation, patientID, AdmissionID, OrginalData,
        NewData)//this passes the action just performed into the objective to add it to their action
        log
        SET encryptedData AS RUN OBJECTIVE 37(newData)//encrypts the data using the objective
        WRITE encryptedData TO filename//writes the encrypted data to file, as it is the latest entry
        it is automatically in order
END program
```

36. Encrypting data before being written to file

```
START program
        RETRIEVE dataToEncrypt//the value intending to be encrypted is passed through
        SET data//initialising the encrypted value
        SET length AS dataToEncrypt.length//the length of the attribute is found
        FOR index FROM 0 TO length //declaration of a for loop that will cycle through every
        character in the string
                SET letternum AS  data.index-96//converts the ascii value to the characters
                numerical value
                SET letternum AS 27-letternum// finds the opposite value so 1 => 26
                SET data.index AS letternum + 96// finds the new ascii value
                SET data.index AS (char) data.index// converts the ascii value to the character it
                represents
        NEXT index//moves onto the next index to encrypt
        RETURN data//the newly encrypted value is returned
END program
```

37. Decrypting that has been read from file

```
START program
        RETRIEVE dataToDecrypt//the value intending to be decrypted is passed through
        SET data//initialising the decrypted value
        SET length AS dataToDecrypt.length//the length of the attribute is found
        FOR index FROM 0 TO length //declaration of a for loop that will cycle through every
        character in the string
                SET letternum AS  data.index-96//converts the ascii value to the characters
                numerical value
                SET letternum AS 27-letternum// finds the opposite value so 1 => 26
                SET data.index AS letternum + 96// finds the new ascii value
                SET data.index AS (char) data.index// converts the ascii value to the character it
                represents
```

NEXT index**//**the next index is then used
RETURN data**//**the newly decrypted value is returned
END program

40.Searching through booking

START program
SET *allBookings[]***//**initialises the array containing all the respective booking information
SET filename **AS** (userID+"_file.txt")**//**declares the file name by using the correct attributes of the user
OPEN filename**//**opens the file
DO
READ LINE//reads the file at the line
DO
THEN READ LINE//reads the file at the line
UNTILL(LINE CONTAINS B AT INDEX 0)
SET *allBookings[counter]* ***AS*** *LINE***//**saves the contents of the line to the index of the array
SET COUNTER **++//**increments counter
UNITL LINE IS Contains LETTER D//iteration condition seeing if the line read has finished reading all the appointments
PERFORM Binary search ON bookings// carries on as the same from the old design
END program

Pseudocode for the new processes of the system

Now that the old processes are redesigned t reflect the new system I will now go onto the final part of the pseudocode that being for the new process. To improve the overall appearance of the document as all the code will be new there is no need to have the text in red and instead will be the original scheme.

42. Add/Edit employee demographic information

```
START program
    RETRIEVE user//retrieves the user
    IF user IS staff//selection determining if the current user is this entity
        THEN SET employee AS (RUN OBJECTIVE 48(user.employeeID+"_file.txt"))//calls the objective which retrieves the user's information
        SET typeID AS user.staffID//sets the correct id of the employee
    ENDIF
    IF user IS consultant//selection determining if the current user is this entity
        THEN SET employee AS (RUN OBJECTIVE  49(user.consultantID+"_file.txt"))//calls the objective which retrieves the user's information
        SET typeID AS user.consultantID//sets the correct id of the employee
    ENDIF
    IF user IS management//selection determining if the current user is this entity
        THEN SET employee AS (RUN OBJECTIVE 50(user.managmentID+"_file.txt"))//calls the objective which retrieves the user's information
        SET typeID AS user.managementID//sets the correct id of the employee
    ENDIF
    WHILE INPUT ISN'T ROUGE //termination condition checking to see if employee done with changing details
        THEN IF INPUT IS surname//Checks to see if this is the desired field
            THEN RUN OBJECTIVE 23(INPUT)//validates the field so input is valid
            SET user.surname AS INPUT//Finally the attribute is set to the object
        ENDIF
        THEN IF INPUT IS firstname//Checks to see if this is the desired field
            THEN RUN OBJECTIVE 23(INPUT)//validates the field so input is valid
            SET user.firstname AS INPUT//Finally the attribute is set to the object
        ENDIF
        THEN IF INPUT IS houseNum//Checks to see if this is the desired field
            THEN RUN OBJECTIVE 23(INPUT)//validates the field so input is valid
            SET user.houseNum AS INPUT//Finally the attribute is set to the object
        ENDIF
        THEN IF INPUT IS houseStreet//Checks to see if this is the desired field
            THEN RUN OBJECTIVE 23(INPUT)//validates the field so input is valid
            SET user.houseStreet AS INPUT//Finally the attribute is set to the object
        ENDIF
        THEN IF INPUT IS Postcode//Checks to see if this is the desired field
            THEN RUN OBJECTIVE 23(INPUT)//validates the field so input is valid
            SET user.Postcode AS INPUT//Finally the attribute is set to the object
        ENDIF
```

```
        THEN IF INPUT IS nationality//Checks to see if this is the desired field
                THEN RUN OBJECTIVE 23(INPUT)//validates the field so input is valid
                SET user.nationality AS INPUT//Finally the attribute is set to the object
        ENDIF
        THEN IF INPUT IS smoker//Checks to see if this is the desired field
                THEN RUN OBJECTIVE 23(INPUT)//validates the field so input is valid
                SET user.smoker AS INPUT//Finally the attribute is set to the object
        ENDIF
        THEN IF INPUT IS drinker//Checks to see if this is the desired field
                THEN RUN OBJECTIVE 23(INPUT)//validates the field so input is valid
                SET user.drinker AS INPUT//Finally the attribute is set to the object
        ENDIF
        THEN IF INPUT IS disability//Checks to see if this is the desired field
                THEN RUN OBJECTIVE 23(INPUT)//validates the field so input is valid
                SET user.disability AS INPUT//Finally the attribute is set to the object
        ENDIF

        THEN IF INPUT IS D.O.B //Checks to see if this is the desired field
                THEN RUN OBJECTIVE 23(INPUT)//validates the field so input is valid
                SET user.D.O.B AS INPUT//Finally the attribute is set to the object
        ENDIF

        THEN IF INPUT IS Religion//Checks to see if this is the desired field
                THEN RUN OBJECTIVE 23(INPUT)//validates the field so input is valid
                SET user.Religion AS INPUT//Finally the attribute is set to the object
        ENDIF

        THEN IF INPUT IS Gender//Checks to see if this is the desired field
                THEN RUN OBJECTIVE 23(INPUT)//validates the field so input is valid
                SET user.Gender AS INPUT//Finally the attribute is set to the object
        ENDIF
        THEN IF INPUT IS carer//Checks to see if this is the desired field
                THEN RUN OBJECTIVE 23(INPUT)//validates the field so input is valid
                SET user.carerAS INPUT//Finally the attribute is set to the object
        ENDIF
        THEN IF INPUT IS translator//Checks to see if this is the desired field
                THEN RUN OBJECTIVE 23(INPUT)//validates the field so input is valid
                SET user.translatorAS INPUT//Finally the attribute is set to the object
        ENDIF
        THEN IF INPUT IS Sex//Checks to see if this is the desired field
                THEN RUN OBJECTIVE 23(INPUT)//validates the field so input is valid
                SET user.SexAS INPUT//Finally the attribute is set to the object
        ENDIF
        THEN IF INPUT IS county//Checks to see if this is the desired field
                THEN RUN OBJECTIVE 23(INPUT)//validates the field so input is valid
                SET user.county AS INPUT//Finally the attribute is set to the object
        ENDIF
ENDWHILE
```

SET newDemographic **AS**
((typeID)+","+(user.employeeID)+","+(user.Password)+","+(user.surname)+","+(user.firstname)+","+(user.houseNum)+","+(user.houseStreet)+","+(user.town)+","+(user.Postcode)+","+(user.contactNumber)+","+(user.nationality)+","+(user.smoker)+","+(user.drinker)+","+(user.D.O.B)+","+(user.Religion)+","+(user.Allergies)+","+(user.Gender)+","+(user.diablities)+","+(user.carer)+","+(user.translator)+","+(user.county)+","+(employee.wage)+","+(employee.hourlyPerWeek)+","+(user.archived))**//**concatenates the entirety of the employees demographical information

SET encryptedData **AS RUN OBJECTIVE 37(**newDemogrphic**)//**Encrypts the demographic information

SET filename AS (typeID+"_file.txt))**//**Declares the correct filename

WRITE encryptedData **TO** filename **AT LINE 1//**Writes the new information to the first line

END program

43. Logout Users

START program

IF logoutButton **HAS BEEN PRESSED//**

THEN OPEN NEW WINDOW//opens a confirmation window making sure that the user is ok with logging out, verification basically

OUTPUT"Are you sure you want to log out?**"//**Informs the user they are about to log out

IF OPTION IS YES//Selection determining that the input was correct, they want to log out

THEN SET user **AS NULL//**Clears the users object so no fields are accessible no information is now stored in RAM

SET listOfPanels **AS NULL//**Clears the stack containing their panels

SET currentPanel **AS** loginPanel**//**Moves the user to the homepage

CLOSE WINDOW//closes the popup window

ENDIF

ELSE

THEN CLOSE WINDOW//selection determining if the pressing of the logout button was a mistake

ENDIF

ENDIF

END program

44. Remember Me

START program
    IF LOGIN SUCCESSFUL AND (rememberMeTB **IS SELECTED)//**selection determining if the user wanted to actually be remembered in the first place
        THEN SET newLoginCredentials **AS** (username)**//**Sets the new username as the one used to get onto the system
        SET filename **AS "General.txt"//**sets the correct filename
        OPEN filename**//**Opens the file to allow changes to be made
        OVERWRITE filename **WITH** newLoginCredentials**//**overwrites over any old usernames, this could be the same user but looking into using selection to prevent this it saves more resources to just allow the process to happen
        CLOSE filename**//**closes the file as it is now finished
    ENDIF
END program

45. Cancel bookings

START program
    RETRIEVE patient **RUN OBJECTIVE 47(**patientID**)//**retrieves the correct patient from file by calling method
    RETRIEVE consultant **RUN OBJECTIVE 49(**consultantID**)//**retrieves the correct consultant from file by calling method
    SET desiredBooking **AS INPUT//**user selects booking they want cancelled
        SET bookingIndex **AS RUN OBJECTIVE 40//**runs objective that determines the booking needing removed
    SET arraylength **AS** allBookings[].length**//**sets the length of the array
    OPEN ((patientID)+"_file.txt")**//**opens the patient's file with all their information
    WRITE demographicInfo**//**writes all their demographic information to file
    WRITE notifications**//**writes all their notifications information to file
    FOR counter 0 **TO** bookingIndex -1**//**runs a counter that iterates up to the last booking before the desired one
        THEN WRITE Alladmissions[counter]**//**writes the admission information
        WRITE allBookings[counter]**//**Writes the booking information about that admission
        WRITE allAdmissions{alldocuments[counter]]**//**Writes all the documents associated with the admission
    NEXT FOR//repeats process
    WRITE Alladmissions[bookingIndex]**//**writes the admission information
    WRITE cancelled**//**informs the document that this booking was cancelled
        WRITE allAdmissions{alldocuments[bookingIndex]]**//**Writes all the documents associated with the admission
    FOR counter bookingIndex +1 **TO** arraylength **//**runs a counter that iterates up to the last item in the array staring after the desired booking
        THEN WRITE Alladmissions[counter]**//**writes the admission information
        WRITE allBookings[counter]**//**Writes the booking information about that admission

```
            WRITE allAdmissions{alldocuments[counter]]//Writes all the documents associated
            with the admission
        NEXT counter //repeats process
        CLOSE ((patientID)+"_file.txt")//closes the file
        OPEN ((consultantID)+"_file.txt")//
        WRITE demographicInfo//writes all their demographic information to file
        FOR counter 0 TO arraylength //runs a counter that iterates up to the through the entire
        array
            THEN WRITE Allpatients[Alladmissions[counter]]//
        NEXT FOR//repeats process
        FOR counter 0 TO bookingIndex -1//runs a counter that iterates up to the last booking
        before the desired one
            THEN WRITE allBookings[counter]//Writes the booking information about that
        admission
        NEXT counter //repeats process
        WRITE cancelled//informs the document that this booking was cancelled
        FOR counter bookingIndex +1 TO arraylength //runs a counter that iterates up to the last
        item in the array staring after the desired booking
            THEN WRITE allBookings[counter]//Writes the booking information about that
            admission
        NEXT counter //repeats process
        CLOSE ((consultantID)+"_file.txt")//closes the file
END program
```

46. Determining a consultant for an admission

```
START program
        RETRIVE allconsultants[]//retrieves the list of consultants to choose from
        SET length AS allconsultants.length//finds the umber of consultants or size of the array
        RETRIEVE admission//retrieves the admission
        OUTPUT admission.symptoms[]//Outputs the symptoms the patient is experiencing to the
        staff
        OUTPUT admission.areasAffected[]//Outputs the areas the patient is affected to the staff
        FOR counter IN RANGE 0 TO length //runs a for loop that iterates through every consultant
            THEN OUTPUT allconsultants[counter].expertise//Outputs the particular expert
            areas of the consultant
        NEXT counter//iterates onto the next consultant
        SET Consultant AS INPUT//staff determines which consultant they want for the admission
        RUN OBJECTIVE 23(INPUT)//validates the input, just determines a consultant has been
        selected
        SET admission.consultant AS INPUT//saves the consultant to the correct field
        RETURN admission.consultant//returns the value to the prior method
END program
```

47. Read Patient information

START program
OPEN filename**//**opens the file to allow it to be amended
READ LINE//
SET decryptedtext **AS RUN OBJECTIVE 38**(line)**//**the line is decrypted
SET demoAttributes**[] AS** decryptedtext **.SPLIT**
SET patient**.**consultantID **AS** demoAttributes**[0]//**saves the attribute from the array
SET patient**.**Password **AS** demoAttributes**[1]//**saves the attribute from the array
SET patient**.surname AS** demoAttributes**[2]//**saves the attribute from the array
SET patient**.firstname AS** demoAttributes**[3]//**saves the attribute from the array
SET patient**.houseNum AS** demoAttributes**[4]//**saves the attribute from the array
SET patient**.houseStreet AS** demoAttributes**[5]//**saves the attribute from the array
SET patient**.town AS** demoAttributes**[6]//**saves the attribute from the array
SET patient**.Postcode AS** demoAttributes**[7]//**saves the attribute from the array
SET patient**.contactNumber AS** demoAttributes**[8]//**saves the attribute from the array
SET patient**.nationality AS** demoAttributes**[9]//**saves the attribute from the array
SET patient**.bloodtype AS** demoAttributes**[10]//**saves the attribute from the array
SET patient**.smoker AS** demoAttributes**[11]//**saves the attribute from the array
SET patient**.drinker AS** demoAttributes**[12]//**saves the attribute from the array
SET patient**.numOfAdmissions AS** demoAttributes**[13]//**saves the attribute from the array
SET patient**.D.O.B AS** demoAttributes**[14]//**saves the attribute from the array
SET patient**.Religion AS** demoAttributes**[15]//**saves the attribute from the array
SET patient**.Allergies AS** demoAttributes**[16]//**saves the attribute from the array
SET patient**.Gender AS** demoAttributes**[17]//**saves the attribute from the array
SET patient**.disablities AS** demoAttributes**[18]//**saves the attribute from the array
SET patient**.carer AS** demoAttributes**[19]//**saves the attribute from the array
SET patient**.translator AS** demoAttributes**[20]//**saves the attribute from the array
SET patient**.county AS** demoAttributes**[21]//**saves the attribute from the array
SET patient**.DaysSinceLastUpdate AS** demoAttributes**[22]//**saves the attribute from the array
SET patient**.Sex AS** demoAttributes**[23]//**saves the attribute from the array
RETURN patient
END program

48. Read Staff information

START program
OPEN filename**//**opens the file to allow it to be amended
READ LINE//
SET decryptedtext **AS RUN OBJECTIVE 38**(line)**//**the line is decrypted
SET demoAttributes**[] AS** decryptedtext **.SPLIT**
SET staff**.**staffID **AS** demoAttributes**[0]//**saves the attribute from the array
SET staff**.**employeID **AS** demoAttributes**[1]//**saves the attribute from the array
SET staff**.**Password **AS** demoAttributes**[2]//**saves the attribute from the array
SET staff**.surname AS** demoAttributes**[3]//**saves the attribute from the array

```
        SET staff.firstname AS demoAttributes[4]//saves the attribute from the array
        SET staff.houseNum AS demoAttributes[5]//saves the attribute from the array
        SET staff.houseStreet AS demoAttributes[6]//saves the attribute from the array
        SET staff.town AS demoAttributes[7]//saves the attribute from the array
        SET staff.Postcode AS demoAttributes[8]//saves the attribute from the array
        SET staff.contactNumber AS demoAttributes[9]//saves the attribute from the array
        SET staff.nationality AS demoAttributes[10]//saves the attribute from the array
        SET staff.bloodtype AS demoAttributes[11]//saves the attribute from the array
        SET staff.smoker AS demoAttributes[12]//saves the attribute from the array
        SET staff.drinker AS demoAttributes[13]//saves the attribute from the array
        SET staff.D.O.B AS demoAttributes[14]//saves the attribute from the array
        SET staff.Religion AS demoAttributes[15]//saves the attribute from the array
        SET staff.Allergies AS demoAttributes[16]//saves the attribute from the array
        SET staff.Gender AS demoAttributes[17]//saves the attribute from the array
        SET staff.disablities AS demoAttributes[18]//saves the attribute from the array
        SET staff.carer AS demoAttributes[19]//saves the attribute from the array
        SET staff.translator AS demoAttributes[20]//saves the attribute from the array
        SET staff.county AS demoAttributes[21]//saves the attribute from the array
        SET staff.DaysSinceLastUpdate AS demoAttributes[22]//saves the attribute from the array
        SET staff.Sex AS demoAttributes[23]//saves the attribute from the array
        SET staff.wage AS demoAttributes[24]//saves the attribute from the array
        SET staff.archive AS demoAttributes[25]//saves the attribute from the array
        SET staff.hoursperWeek AS demoAttributes[26]//saves the attribute from the array

        RETURN staff
END program
```

49. Read consultant Information

```
START program
        OPEN filename//opens the file to allow it to be amended
        READ LINE//
        SET decryptedtext AS RUN OBJECTIVE 38(line)//the line is decrypted
        SET demoAttributes[] AS decryptedtext .SPLIT
        SET consultant.consultantID AS demoAttributes[0]//saves the attribute from the array
        SET consultant.Password AS demoAttributes[1]//saves the attribute from the array
        SET consultant.surname AS demoAttributes[2]//saves the attribute from the array
        SET consultant.firstname AS demoAttributes[3]//saves the attribute from the array
        SET consultant.houseNum AS demoAttributes[4]//saves the attribute from the array
        SET consultant.houseStreet AS demoAttributes[5]//saves the attribute from the array
        SET consultant.town AS demoAttributes[6]//saves the attribute from the array
        SET consultant.Postcode AS demoAttributes[7]//saves the attribute from the array
        SET consultant.contactNumber AS demoAttributes[8]//saves the attribute from the array
        SET consultant.nationality AS demoAttributes[9]//saves the attribute from the array
        SET consultant.bloodtype AS demoAttributes[10]//saves the attribute from the array
        SET consultant.smoker AS demoAttributes[11]//saves the attribute from the array
        SET consultant.drinker AS demoAttributes[12]//saves the attribute from the array
```

SET consultant**.D.O.B AS** demoAttributes**[14]//**saves the attribute from the array
SET consultant**.Religion AS** demoAttributes**[15]//**saves the attribute from the array
SET consultant**.Allergies AS** demoAttributes**[16]//**saves the attribute from the array
SET consultant**.Gender AS** demoAttributes**[17]//**saves the attribute from the array
SET consultant**.disablities AS** demoAttributes**[18]//**saves the attribute from the array
SET consultant**.carer AS** demoAttributes**[19]//**saves the attribute from the array
SET consultant**.translator AS** demoAttributes**[20]//**saves the attribute from the array
SET consultant**.county AS** demoAttributes**[21]//**saves the attribute from the array
SET consultant**.DaysSinceLastUpdate AS** demoAttributes**[22]//**saves the attribute from the array
SET consultant**.Sex AS** demoAttributes**[23]//**saves the attribute from the array
SET consultant**.wage AS** demoAttributes**[24]//**saves the attribute from the array
SET consultant**.archive AS** demoAttributes**[25]//**saves the attribute from the array
SET consultant**.hoursperWeek AS** demoAttributes**[26]//**saves the attribute from the array
READ LINE//
SET decryptedtext **AS RUN OBJECTIVE 38**(line)**//**the line is decrypted
SET demoAttributes2**[] AS** decryptedtext **.SPLIT**
SET consultant**.*practisesList[]* AS** demoAttributes2**[0]//**saves the attribute from the array
SET consultant**.numOfPatientsAS** demoAttributes2**[1]//**saves the attribute from the array
RETURN consultant
END program

50. Read management information

START program
OPEN filename**//**opens the file to allow it to be amended
READ LINE//
SET decryptedtext **AS RUN OBJECTIVE 38**(line)**//**the line is decrypted
SET demoAttributes**[] AS** decryptedtext **.SPLIT**
SET managment**.**ManagmentID AS demoAttributes**[]//**saves the attribute from the array
SET managment.employeID **AS** demoAttributes**[]//**saves the attribute from the array
SET managment.Password **AS** demoAttributes**[]//**saves the attribute from the array
SET patimanagmentent**.surname AS** demoAttributes**[]//**saves the attribute from the array
SET managment**.firstname AS** demoAttributes**[]//**saves the attribute from the array
SET managment**.houseNum AS** demoAttributes**[]//**saves the attribute from the array
SET managment**.houseStreet AS** demoAttributes**[]//**saves the attribute from the array
SET managment**.town AS** demoAttributes**[]//**saves the attribute from the array
SET managment**.Postcode AS** demoAttributes**[]//**saves the attribute from the array
SET managment**.contactNumber AS** demoAttributes**[]//**saves the attribute from the array
SET managment**.nationality AS** demoAttributes**[]//**saves the attribute from the array
SET managment**.bloodtype AS** demoAttributes**[]//**saves the attribute from the array
SET managment**.smoker AS** demoAttributes**[]//**saves the attribute from the array
SET managment**.drinker AS** demoAttributes**[]//**saves the attribute from the array
SET managment**.D.O.B AS** demoAttributes**[]//**saves the attribute from the array
SET managment**.Religion AS** demoAttributes**[]//**saves the attribute from the array
SET managment**.Allergies AS** demoAttributes**[]//**saves the attribute from the array
SET managment**.Gender AS** demoAttributes**[]//**saves the attribute from the array

SET managment**.disablities AS** demoAttributes**[]//**saves the attribute from the array
SET managment**.carer AS** demoAttributes**[]//**saves the attribute from the array
SET managment**.translator AS** demoAttributes**[]//**saves the attribute from the array
SET managment**.county AS** demoAttributes**[]//**saves the attribute from the array
SET managment**.DaysSinceLastUpdate AS** demoAttributes**[]//**saves the attribute from the array
SET managment**.Sex AS** demoAttributes**[]//**saves the attribute from the array
SET managment**.wage AS** demoAttributes**[]//**saves the attribute from the array
SET managment**.archive AS** demoAttributes**[]//**saves the attribute from the array
SET managment**.hoursperWeek AS** demoAttributes**[]//**saves the attribute from the array

RETURN managment
END program

51. Searching definition

START program
RETRIEVE *allDefenitions[]***//**retrieves the entire array of all definitions available
RETRIEVE searchDefinition**//**Returns the document needing to be searched
SET lengthOfArray **AS** length.*allDefenitions*[]**//**sets the length of the array
SET startPosition**//**Declares the start position index variable
SET middlePosition**//**Declares the middle position index variable
SET endPosition **AS** lengthOfArray**//**Declares the end position index variable
Initially setting it as the last index of the definition list
SET found **AS FASLE//**sets the value to be false as nothing is found as of yet
SET Position**//**sets the found position of the definition if found within the array
DO
SET middlePosition **AS (**startPosition + endPosition) **DIV 2//**Finds the middle position of the array by averaging the start and end points
IF searchDefinition**IS** *allDefenitions[*middlePosition*]//*Selection determining if the search value is the same as the document at the current index
THEN SET found **AS TRUE//**updates the found attribute
SET position **AS** middlePosition**//**sets the position of the index where it was found
OUTPUT patient found at index (position)**//**outputs that the patient was found
RETRIEVE patient**//**retrieves all the data regarding the definition
ENDIF
ELSE IF searchDefinition IS **LESS THAN** *allDefenitions[*middlePosition*]//* Selection determining if the search value is less than the definition at the current index
THEN SET endPosition **AS** middlePosition – 1**//**correctly adjusts the new end position removing any irrelevant definitions
ENDIF
ELSE IF searchDefinition **IS GREATER THAN** *allDefenitions[*middlePosition*]//*
Selection determining if the search value is greater than the definition at the current index

```
                THEN SET startPosition AS middlePosition + 1// correctly adjusts the new
                start position removing any irrelevant patient
            ENDIF
        UNTIL found IS TRUE OR endPosition IS LESS THAN startPosition//iteration condition of
        termination until found or that the start index exceeds the end point indicating that no other
        values exist and the search value is not there
        IF found IS FASLE//selection if seeing that the definition was not found
            THEN OUTPUT no document found//informs that the definition was not found
END program
```

52. Sorting definitions

```
START  DEF(Quicksort) program
        RETRIEVE arrayOfDefenitions//retrieves the parameter of array
        RETIREVE lowestIndex//finds the lowest index that is unsorted in the array
        RETIREVE highestIndex//finds the highest point in the array that not sorted
        SET tempLow AS lowestIndex//sets a floating pointer used to eventually determine new min
        pointer
        SET tempHigh AS highestIndex//sets a floating pointer used to eventually determine new
        min pointer
        SET swapVar//sets a temporary buffer
        SET pivot AS(lowestIndex + highestIndex)DIV 2//finds midpoint
        WHILE tempLow IS LESS THAN OR EQUAL TO indexhigh//checks that if the variables are not
        out of bounds in respect to the assigned limits
            THEN WHILE arrayOfDefenitions[templow] IS LESS THAN pivot AND tmplow IS LESS
            THAN highestIndex//finds value that should be on other side of array
                THEN tempLow ++//moves pointer on
            ENDWHILE
            WHILE arrayOfDefenitions[temphigh] IS GREATER THAN pivot AND tmphigh IS
            GREATER THAN lowestindex //finds value that should be on other side of array
                THEN tempHigh --//moves pointer on
            ENDWHILE
            IF templow IS LESS THAN OR EQUAL TO temphigh//checks to see is a swap is
            needed
                THEN SET swapVar AS arrayOfDefenitions[tmplow]//sets buffer variable
                SET arrayOfDefenitions[tmplow] AS array[tmphigh] //swaps upper value
                SET arrayOfDefenitions[tmphigh] AS swapVar//swaps lower value
                Tmplow ++//increments variables
                Tmphigh –//increments variables
            ENDIF
        ENDWHILE
        IF(lowestindex IS LESS THAN temphigh)//Selection determining that if there is more than
        one value in between the pointers
            THEN CALL Quicksort(arrayOfDefenitions,lowestIndex,tempHigh)//recalls the
            method this time using new pointers
        ENDIF
```

```
        IF(tmplowLESS THAN highestIndex)//Selection determining that if there is more than one
        value in between the pointers
                THEN CALL Quicksort(arrayOfDefenitions,tempLow,highestIndex)//recalls the
                method this time using new pointers
        ENDIF
END program
```

53. Primary key generation

```
START program
        RETRIEVE entityType //passes through which type of user is needed
        IF entityType IS patient//selection determining if that entity
                THEN SET letter AS "P"//setting the type of key needed
        ENDIF
        IF entityType IS staff//selection determining if that entity
                THEN SET letter AS "S"//setting the type of key needed
        ENDIF
        IF entityType IS consultant//selection determining if that entity
                THEN SET letter AS "C"//setting the type of key needed
        ENDIF
        IF entityType IS management //selection determining if that entity
                THEN SET letter AS "M"//setting the type of key needed
        ENDIF
        IF entityType IS  booking//selection determining if that entity
                THEN SET letter AS "B"//setting the type of key needed
        ENDIF
        IF entityType IS admission//selection determining if that entity
                THEN SET letter AS "A"//setting the type of key needed
        ENDIF
        IF entityType IS document//selection determining if that entity
                THEN SET letter AS "D"//setting the type of key needed
        ENDIF
        IF entityType IS patient OR consultant OR staff OR management //selection determining the
        id will be for a user
                THEN OPEN ("general.txt")//opens the general file where all the surnames are
                stored
                SET shortname AS user.surname(FROM INDEX 0 TO 2)//sets the value to the first
                three parts of the surname
                RETRIEVE lastInt FROM FILE AT SURNAME(shortname)//searches for the surname
                and retrieves the int value associated with it
                IF lastInt IS NULL//checks to see if there are no users with that surname
                        THEN SET lastint 0//defaults the surname and then adds
                        ADD surname TO FILE// adds the surname to file so it is remembered that it
                        exists
                ENDIF
                SET lastint ++//increments the value so it is not the same
        ENDIF
```

```
    IF entityType IS booking OR admission OR document//selection determining the id will be
    patient based but not for the patient itself
        THEN OPEN(user.patientID+"_file.txt")//opens the patients file
        IF entityType admission//selection determining the id will be admission
            THEN SET lastInt AS patient.numberOFAdmissions +1//increments the value
        ENDIF
        FIND CORRECT ADMISSON//finds the desired admission where the item is destined
        IF entityType document OR booking//selection determining the id will be document
        or booking
            THEN SET lastInt AS admissions.numberOFDocuments +1//increments value
            so it is correct
        ENDIF
    SET ID AS (letter+shortname+lastInt)//concatenates all the values together so it is one
coherent value
    RETURN(ID)//returns the value to the user.
END program
```