**A smart medical system for stakeholders = LEG AND FEET**

Project:

* Manage the day to day activity

Focus Questions: What do we want the system to do? Identify key terms

Get the case scenario and highlight key terms/ information to include in cmap

Good understanding of the context. Know the scope of your system.

Examples of focus questions:

What are we developing?

* A smart medical management system

Who are the stakeholders?

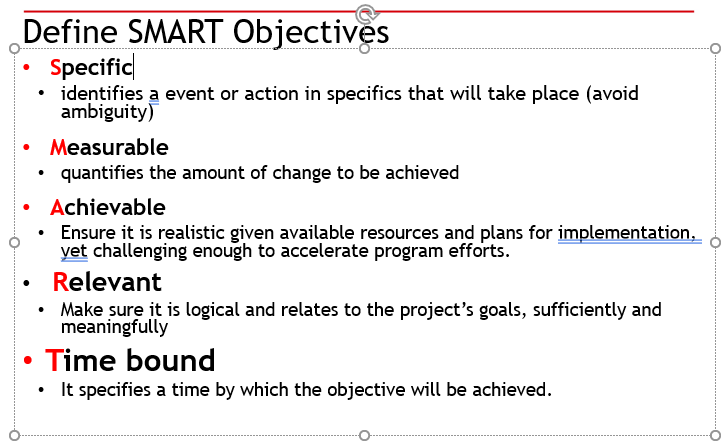
* Medical practitioners
* Patients
* Researchers
* Government
* Regulatory bodies
* Pharmacists
* Doctors
* Nurses
* Administrators

Things to look for in the maps:

Connectedness (numerous cross-linking), Link quality (good linking phrases), Link variety (diversity of linking phrases), Dynamism (changes over time and evolves), Orientation (create an overview of concept), Morphology (show understanding of what you are trying to show)

**Program Requirements!!**

* **Making appointments**
* **Self Diagnosis**
* **Normal Diagnosis**
* **Contain patient records**
* **Prescriptions**
* **Store staff records**
* **Store hospital locations**



Case Scenario

Solent Medical Service (SMS) is a conglomerate of hospitals that provided medical services to clients. These services range from treating patients, providing medical education, prescription and sale of medicine, carrying out medical examinations and research about medicine, health and wellbeing. While the organisation is renowned for its outstanding services and dealings with stakeholders, it has grown enormously over the last years. Recently SMS intends to further improve their services and the management of daily activities thus, the board has decided to employ a service for the development of a “Smart Medical Management System” for its stakeholders.

A smart medical management system is an electronic system for the management of medical activities between health organisation and corresponding stakeholders. Stakeholders of this system include medical practitioners (i.e., doctors, nurses, pharmacists and administrators), patient, researchers, government and regulatory bodies.

The term smart is emphasized to ensure the system's effectiveness in supporting decision making for stakeholders. Not only will this aid seamless organisation-stakeholders’ relationships, but will also improve the organisation's effectiveness through management

and shedding of any possible wastes, which in turn increases the organisation’s turnover and overall profits.

Therefore, concepts such as activities’ management, managing decision about ailments, medications, location and profiling among other activities should be considered as presented in the system component breakdown document.

Now the organisation has considered the above situation thoroughly and decided to employ your services as a team of the ESS software solution provider to develop a “Smart Medical Management System”. Your role involves conducting knowledge elicitation exercise on the given problem scenario, develop necessary models and implement the software system. You are required to produce a project, identify and suitable software development methodology, and develop the system following the standard software development life cycle. In addition, you will need to produce supporting documents (User and Developer guides) for the software system.

**The project is expected to be completed within the stipulated time**

**(15th May 2020 16:00)**

**and the following are expected to be considered.**

### Requirements

Doctors

See Patient Records

Update/create patient records

SMART diagnosis (get diagnosis as typing symptoms)

Prescribe medication and treatment

View appointments

Create appointment

Update/delete appointment

Add staff record

Update staff record

View staff record

Run reports: (Disease, medication, numbers of patients)

Nurse

See Patient Records

Modify patient records

View appointments

Create appointment

Modify/Delete appointment

Receptionist/Admin Staff

View appointment

Update/Delete appointment

Create appointment

View patient record

Update patient record

Create patient record

Create staff record

Update staff record

View staff record

Run reports: (Disease, medication, numbers of patients)

Patient

SMART Self-diagnosis

Create appointment

Cancel appointment

SMART Search for nearby Surgerys/hospitals

Update/delete personal information

Pharmacy

Request Information Page:

* Medication information
* Disease stats
* Patient stats

Research Company

Request Information Page:

* Medication information
* Disease stats
* Patient stats

Government

Request Information Page:

* Medication information
* Disease stats
* Patient stats
* Practicing doctor information
* Practicing nurse information

Regulatory Body

Request Information Page:

* Practicing doctor information
* Practicing nurse information

### Focus Questions

What does it do?

* It is an electronic system to handle the management of medical activities between health organisations and the corresponding stakeholders.

What services are provided?

* It should cover management of activities, manage decisions about aliments, medications, location and profiling.

What part of the body does the system cover?

* Legs and feet Including Ankle, Calf, Foot, Hip, Knee, Thigh

Team Cmap -

Who are circle lab?

What is their role?

Project Cmap -

What is a smart medical management system?

Who are the stakeholders?

How does the SMM system serve each stakeholders

### Software Development Methodology (benefits, limitations and suitability)

|  |  |  |  |
| --- | --- | --- | --- |
|  | Benefits | Limitations | Suitability |
| Waterfall | Adapts to Shifting Teams due to the large amount of documentation laid out in the first stages of the waterfall methodology.  Forces Structured Organization the developer is forced to be disciplined with the project documentation that is created. Making sure they have documentation to run for the complete life cycle, once again making the waterfall method more robust to team changes.  Allows for Early Design Changes due to the large amount of documentation created at the beginning of the project more changes can be made by the client easily in the early stages of development. However, it is much harder to make changes near the end and customers will not see a prototype for an extremely long time.  Suited for Milestone-Focused Development due to the linear nature of waterfall this is a great model for use with large organisations that require dates and a complete timeline of the project as it is much easier to develop both of those things using this model. | Nonadaptive Design Constraints the waterfall model is unable to adapt to changes quickly as a major design flaw discovered in the application at the testing phase means costly delays or potentially scrapping the project completely.  Ignores Mid-Process User/Client Feedback the waterfall model is unable to adapt to client feedback at a late stage without taking steps backwards thought the model that will be costly and time-consuming.  Delayed Testing Period testing is not started until the penultimate stage of the model potentially finding design flaws at an extremely late stage. It also breeds lackadaisical programming as testing is seen as an afterthought rather than an integral part of the development process. | Waterfall does not have the built-in flexibility of Agile but, on a project of this size it will not be difficult to move back steps if required.  It is simple for all members to understand in depth with defined start and finish points. Making it easier to write a full plan for the entire length of the project.  I believe this will also be easier to implement in the limited time we have before the hand in deadline in May. |
| Agile | The life cycle is an iterative process which means each stage will be carried out multiple times and with each iteration the software is able to become even more developed. This gives the opportunity for customer testing during the development process and changes can be made easier.The product has the opportunity to be adjusted and made better after each iteration as ‘the model produces ongoing releases, each with small, incremental changes’ (Morris, 2018). If a client is not happy with a part of the project, this methodology does give them the chance to change their system requirements at a variety of different stages within the project. | This methodology can take longer than the others due to testing and possibly changing the project at each iteration. This also creates a greater workload for developers and you need to ensure you have good communication with your clients for this methodology to be successful. The documentation for a project created using the agile methodology can often be less detailed due to the fact that it is likely that the project may change during the process of developing it so not everything can be documented straight away. | This will be beneficial for our project as we meet with the client regularly so they can see each update, test it themselves and decide whether they are happy or would like something different. |
| Rapid | Flexible and adaptable to changes  Reduces risks (by having constant communication with the client)  Due to code generators and code reuse, there is a reduction of manual coding  Due to prototyping in nature, there is a possibility of lesser defects  With less people, productivity can be increased in short time | It can’t be used for smaller projects  Requires highly skilled designers or developers  Progress and problems are hard to track as there is little documentation to demonstrate what has been done    Reduced scalability occurs because a RAD developed application begins as a prototype and evolves into a finished application    Reduced features due to limited time and features are pushed in a later version. | Software product is to be developed in a short time span (2-3 months)  The client is available throughout the development process. |
| DevOps | Similar to agile but extends the principles not only to code but how the service is delivered  Looks at how an organization collaborates and uses this too shape a project. Focuses on how the app and system interact  High app development speed, as it is possible to deploy new versions more regularly  Spend less time on unplanned work improving better service and features | Difficulty to implement the methodology, standardizing procedures and processes  Continuous testing, which could be time consuming for this project  A high amount of workload to keep on top of changes and | Devops could be beneficial to the project as there are many stakeholders that will be accessing the system therefore building a system that looks at how the company operates could make the system successful. Although it has to be mentioned it can be time consuming to get a devops methodology in place with a high workload. |
| Rational Unified Process | This methodology emphasizes on accurate documentation  It is proactively able to resolve the project risks that are associated with the clients evolving requirements for careful changes and request management  Very less need for integration as the process of integration goes on throughout the development process | The software developer needs to be expert in their work to develop software under this methodology.  The development process in this methodology is very complex and not exactly organized.  Integration throughout the process of software development adds the confusion that causes more issues during the stages of testing.  This process is too complex therefore it is very hard to understand. | This is an object-oriented and web-enabled program development methodology. Which considering the project in hand I considered rapidly that will be suitable, This model also helps software developers for providing them guidelines, templates, and examples for all aspects and stages of software development. |

Morris, A. (2018). 6 basic SDLC methodologies: Which one is best? | Robert Half. [online] Roberthalf.com.au. Available at: https://www.roberthalf.com.au/blog/employers/6-basic-sdlc-methodologies-which-one-best [Accessed 24 Jan. 2019].

**Use cases to be done:**

Add Person - Josh

Modify Person

View Person

Add Appointment

Modify Appointment - Josh

Delete Appointment

View Appointment

Add Location

Modify Location

Delete Location- Juan

View Location

Add Report Request

Delete Report Request

View Report Request

Approve Report Request - Matt

Decline Report Request

Self Diagnosis (Patient Only) - Emma

Add Diagnosis - Emma

Delete Diagnosis

View Disease

Add Treatment

Modify Treatment

Delete Treatment

View Treatment - Juan

Add Prescription - Matt

Delete Prescription

View Prescription

**View Person**

|  |  |  |
| --- | --- | --- |
|  | View Person |  |
| Step | Actor Action | System Response |
| 1 | User clicks login |  |
| 2 |  | The system prompts users for their username and password. |
| 3 | The user enters the details specified in step 2. |  |
| 4 |  | The system verifies the users username and password. |
| 5 |  | The system responds with the home page. |
| 6 | User selects the search page. |  |
| 7 |  | System checks user permissions are valid. (Is the user allowed to access the search page). |
| 8 |  | The system responds with the search page. |
| 9 | User enters possible search details of person (name, address, date of birth, role) |  |
| 10 |  | System checks validity of details, e.g. date of birth is sensible, input exists |
| 11 |  | System checks user permissions are valid. (Is the user allowed to search for that role of people). |
| 12 |  | System searches database for persons with matching details inputted by the user |
| 13 |  | The system displays search results (person first name, last name, dob) |
| 14 | User selects a person record |  |
| 15 |  | The system displays the selected person with all their details. |

Modify Person

|  |  |  |
| --- | --- | --- |
|  | Modify Person |  |
| Step | Actor Action | System Response |
| 1 | User clicks login |  |
| 2 |  | The system prompts users for their username and password. |
| 3 | The user enters the details specified in step 2. |  |
| 4 |  | The system verifies the users username and password. |
| 5 |  | The system responds with the home page. |
| 6 | User selects the search page. |  |
| 7 |  | System checks user permissions are valid. (Is the user allowed to access the search page). |
| 8 |  | The system responds with the search page. |
| 9 | User enters possible search details of person (name, address, date of birth, role) |  |
| 10 |  | System checks validity of details, e.g. date of birth is sensible, input exists |
| 11 |  | System checks user permissions are valid. (Is the user allowed to search for that role of people). |
| 12 |  | System searches database for persons with matching details inputted by the user |
| 13 |  | The system displays search results (person first name, last name, dob) |
| 14 | User selects modify button next to person they want to modify |  |
| 15 |  | System checks user permissions are valid. (Is the user allowed to modify details for that role of people). |
| 16 |  | System displays modification boxes available |
| 17 | User Modifies selected person details |  |
| 18 |  | System checks validity of details, e.g. date of birth is sensible, input exists |
| 19 |  | System updates database with the persons new details |
| 20 |  | System responds with a confirmation page. |

Add Person

|  |  |  |
| --- | --- | --- |
|  | Add Person |  |
| Step | Actor Action | System Response |
| 1 | User clicks login |  |
| 2 |  | The system prompts users for their username and password. |
| 3 | The user enters the details specified in step 2. |  |
| 4 |  | The system verifies the users username and password. |
| 5 |  | The system responds with the home page. |
| 6 | User selects the add person page. |  |
| 7 |  | System checks user permissions are valid. (Is the user allowed to access the add person page). |
| 8 |  | The system responds with the add page. |
| 9 | User enters new person details such as (name, address, date of birth, role) |  |
| 10 |  | System checks validity of details, e.g. date of birth is sensible, input exists |
| 11 |  | System checks user permissions are valid. (Is the user allowed to add a person with that role). |
| 12 |  | System inputs details of the new person into the database and saves it. |
| 13 |  | System responds with a confirmation page. |

Add Appointment

|  |  |  |
| --- | --- | --- |
|  | Add Appointment |  |
| Step | Actor Action | System Response |
| 1 | User clicks login |  |
| 2 |  | The system prompts users for their username and password. |
| 3 | The user enters the details specified in step 2. |  |
| 4 |  | The system verifies the users username and password. |
| 5 |  | The system responds with the home page. |
| 6 | User selects the appointments page. |  |
| 7 |  | System checks user permissions are valid. (Is the user allowed to access the appointment page). |
| 8 |  | System searches database for appointments matching the logged in user |
| 9 |  | The system responds with the appointment page. |
| 10 |  | The system displays search results (Doctor, patient, time, description) |
| 11 | User selects add appointment |  |
| 12 |  | The system searches the database for available appointments |
| 13 |  | The system displays search results |
| 14 | User selects an available appointment from the search results |  |
| 15 | User inputs details (patient details, description) |  |
| 16 |  | System checks the validity of the input details (patient) |
| 17 |  | System checks user permissions are valid. (Is the user allowed to add an appointment). |
| 18 |  | System inputs details of the new appointment into the database and saves it. |
| 19 |  | System responds with a confirmation page. |

Modify Appointment

|  |  |  |
| --- | --- | --- |
|  | Modify Appointment |  |
| Step | Actor Action | System Response |
| 1 | User clicks login |  |
| 2 |  | The system prompts users for their username and password. |
| 3 | The user enters the details specified in step 2. |  |
| 4 |  | The system verifies the users username and password. |
| 5 |  | The system responds with the home page. |
| 6 | User selects the appointments page. |  |
| 7 |  | System checks user permissions are valid. (Is the user allowed to access the appointment page). |
| 8 |  | System searches database for appointments matching the logged in user |
| 9 |  | The system responds with the appointment page. |
| 10 |  | The system displays search results (Doctor, patient, time, description) |
| 11 | User enters search details of appointment (doctor, patient) |  |
| 12 |  | System checks validity of details, e.g. doctor is sensible, input exists |
| 13 |  | System searches database for appointments with matching details inputted by the user |
| 14 |  | The system displays search results (Doctor, patient, time, description) |
| 15 | User selects an appointment record |  |
| 16 |  | The system displays the selected appointment with all it’s details. |
| 17 | User selects modify appointment |  |
| 18 |  | The system displays editable version of the selected appointment |
| 19 | User inputs new details |  |
| 20 |  | System checks the validity of the input details |
| 21 |  | System checks user permissions are valid. (Is the user allowed to modify an appointment). |
| 22 |  | System modifies details of the modified appointment into the database and saves it. |
| 23 |  | System responds with a confirmation page. |

Delete Appointment

|  |  |  |
| --- | --- | --- |
|  | Delete Appointment |  |
| Step | Actor Action | System Response |
| 1 | User clicks login |  |
| 2 |  | The system prompts users for their username and password. |
| 3 | The user enters the details specified in step 2. |  |
| 4 |  | The system verifies the users username and password. |
| 5 |  | The system responds with the home page. |
| 6 | User selects the appointments page. |  |
| 7 |  | System checks user permissions are valid. (Is the user allowed to access the appointment page). |
| 8 |  | System searches database for appointments matching the logged in user |
| 9 |  | The system responds with the appointment page. |
| 10 |  | The system displays search results (Doctor, patient, time, description) |
| 11 | User enters possible search details of appointment (doctor, patient) |  |
| 12 |  | System checks validity of details, e.g. doctor is sensible, input exists |
| 13 |  | System searches database for appointments with matching details inputted by the user |
| 14 |  | The system displays search results (Doctor, patient, time, description) |
| 15 | User selects an appointment record |  |
| 16 |  | The system displays the selected appointment with all it’s details. |
| 17 | User selects delete appointment |  |
| 18 |  | System checks user permissions are valid. (Is the user allowed to delete appointments). |
| 19 |  | The system deletes the appointment to the database. |

View Appointment

|  |  |  |
| --- | --- | --- |
|  | View Appointment |  |
| Step | Actor Action | System Response |
| 1 | User clicks login |  |
| 2 |  | The system prompts users for their username and password. |
| 3 | The user enters the details specified in step 2. |  |
| 4 |  | The system verifies the users username and password. |
| 5 |  | The system responds with the home page. |
| 6 | User selects the appointments page. |  |
| 7 |  | System checks user permissions are valid. (Is the user allowed to access the appointment page). |
| 8 |  | System searches database for appointments matching the logged in user |
| 9 |  | The system responds with the appointment page. |
| 10 |  | The system displays search results (Doctor, patient, time, description) |
| 11 | User enters possible search details of appointment (doctor, patient) |  |
| 12 |  | System checks validity of details, e.g. doctor is sensible, input exists |
| 13 |  | System searches database for appointments with matching details inputted by the user |
| 14 |  | The system displays search results (Doctor, patient, time, description) |
| 15 | User selects an appointment record |  |
| 16 |  | The system displays the selected appointment with all it’s details. |

Add Location

|  |  |  |
| --- | --- | --- |
|  | add Location |  |
| Step | Actor Action | System Response |
| 1 | User clicks login |  |
| 2 |  | The system prompts users for their username and password. |
| 3 | The user enters the details specified in step 2. |  |
| 4 |  | The system verifies the users username and password. |
| 5 |  | The system responds with the home page. |
| 6 | User selects the locations page. |  |
| 7 |  | System checks user permissions are valid. (Is the user allowed to access the locations page). |
| 8 |  | The system responds with the location page. |
| 9 | User selects to add new location |  |
| 10 |  | The system responds with the add location page. |
| 11 | User enters new location details and submit |  |
| 12 |  | System checks validity of new location details |
| 13 |  | The system displays message to confirm if new location has been added or errors. |
| 14 |  | If details correct, system adds location to database and displays the location page |
| 15 |  | If details incorrect, no data added to database user returned to add location page |

Modify Location

|  |  |  |
| --- | --- | --- |
|  | modify Location |  |
| Step | Actor Action | System Response |
| 1 | User clicks login |  |
| 2 |  | The system prompts users for their username and password. |
| 3 | The user enters the details specified in step 2. |  |
| 4 |  | The system verifies the users username and password. |
| 5 |  | The system responds with the home page. |
| 6 | User selects the locations page. |  |
| 7 |  | System checks user permissions are valid. (Is the user allowed to access the locations page). |
| 8 |  | The system responds with the location page. |
| 9 | User enters possible search details of location (town, postcode) |  |
| 10 |  | System checks validity of details, e.g. postcode is sensible, input exists |
| 11 |  | System searches database for locations with matching details inputted by the user |
| 12 |  | The system displays search results (location name, link to location on maps) |
| 13 | User selects a location record |  |
| 14 |  | The system displays the selected location with all its details with modifiable fields |
| 15 | User changes required details of location and presses confirm changes |  |
| 16 |  | System checks validity of details to be modified |
| 17 |  | The system displays messages to confirm if the new location has been added or errors. |
| 18 |  | If details correct, system modifies location details and updates database and displays the location page |
| 19 |  | If details incorrect, no data added to database user returned to add location page |

Delete Location

|  |  |  |
| --- | --- | --- |
|  | Delete Location |  |
| Step | Actor Action | System Response |
| 1 | User clicks login |  |
| 2 |  | The system prompts users for their username and password. |
| 3 | The user enters the details specified in step 2. |  |
| 4 |  | The system verifies the users username and password. |
| 5 |  | The system responds with the home page. |
| 6 | User selects the locations page. |  |
| 7 |  | System checks user permissions are valid. (Is the user allowed to access the locations page). |
| 8 |  | The system responds with the location page. |
| 9 | User enters possible search details of location (town, postcode) |  |
| 10 |  | System checks validity of details, e.g. postcode is sensible, input exists |
| 11 |  | System searches database for locations with matching details inputted by the user |
| 12 |  | The system displays search results (location name, link to location on maps) |
| 13 | User selects a location record to delete |  |
| 14 |  | The system displays a confirmation message/page to confirm you want to delete location |
| 15 | User clicks to confirm delete of location |  |
| 16 |  | The system updates the location to inactive |
| 17 |  | The system returns back to location screen |

View Location

|  |  |  |
| --- | --- | --- |
|  | View Location |  |
| Step | Actor Action | System Response |
| 1 | User clicks login |  |
| 2 |  | The system prompts users for their username and password. |
| 3 | The user enters the details specified in step 2. |  |
| 4 |  | The system verifies the users username and password. |
| 5 |  | The system responds with the home page. |
| 6 | User selects the locations page. |  |
| 7 |  | System checks user permissions are valid. (Is the user allowed to access the locations page). |
| 8 |  | The system responds with the location page. |
| 9 | User enters possible search details of location (town, postcode) |  |
| 10 |  | System checks validity of details, e.g. postcode is sensible, input exists |
| 11 |  | System searches database for locations with matching details inputted by the user |
| 12 |  | The system displays search results (location name, link to location on maps) |
| 13 | User selects a location record |  |
|  |  | The system displays the selected location with all it’s details |

Add Report Request

|  |  |  |
| --- | --- | --- |
|  | Add Report Request |  |
| Step | Actor Action | System Response |
| 1 | User clicks login |  |
| 2 |  | The system prompts users for their username and password. |
| 3 | The user enters the details specified in step 2. |  |
| 4 |  | The system verifies the user’s username and password. |
| 5 |  | The system responds with the home page |
| 6 | User clicks report page |  |
| 7 |  | System checks user permissions are valid. (Is the user allowed to access the search page). |
| 8 |  | System returns users current report requests. |
| 9 |  | The system responds with the report page. |
| 10 | User selects the report and inputs the report variable. (Gender, Dates, Age Ranges) |  |
| 11 |  | System verifies the user has input the required information. |
| 12 |  | System validates the dates entered. |
| 13 |  | System saves report request to the database. |
| 14 |  | System responds with a confirmation page. |

Delete Report Request

|  |  |  |
| --- | --- | --- |
|  | Delete Report Request |  |
| Step | Actor Action | System Response |
| 1 | User clicks login |  |
| 2 |  | The system prompts users for their username and password. |
| 3 | The user enters the details specified in step 2. |  |
| 4 |  | The system verifies the user’s username and password. |
| 5 |  | The system responds with the home page |
| 6 | User clicks report page |  |
| 7 |  | System checks user permissions are valid. (Is the user allowed to access the search page). |
| 8 |  | System returns users current report requests. |
| 9 |  | The system responds with the report page. |
| 10 | User clicks delete next to report. |  |
| 11 |  | The system deletes the report request from the database. |
| 12 |  | System returns the user to the report page. |

View Report Request

|  |  |  |
| --- | --- | --- |
|  | View Report Request |  |
| Step | Actor Action | System Response |
| 1 | User clicks login |  |
| 2 |  | The system prompts users for their username and password. |
| 3 | The user enters the details specified in step 2. |  |
| 4 |  | The system verifies the user’s username and password. |
| 5 |  | The system responds with the home page. |
| 6 | User clicks report page. |  |
| 7 |  | System checks user permissions are valid. (Is the user allowed to access the search page). |
| 8 |  | System returns users current report requests. |
| 9 |  | The system responds with the report page. |
| 10 | User selects an individual report. |  |
| 11 |  | The system searches the database for report details. |
| 12 |  | The system responds with the report details page. |

Approve Report Request

|  |  |  |
| --- | --- | --- |
|  | Approve Report Request |  |
| Step | Actor Action | System Response |
| 1 | User clicks login |  |
| 2 |  | The system prompts users for their username and password. |
| 3 | The user enters the details specified in step 2. |  |
| 4 |  | The system verifies the user’s username and password. |
| 5 |  | The system responds with the home page |
| 6 | User clicks report page. |  |
| 7 |  | System checks user permissions are valid. (Is the user allowed to access the search page). |
| 8 |  | System returns all report requests. |
| 9 |  | The system responds with the report page. |
| 10 | User selects an individual report. |  |
| 11 |  | The system searches the database for report details. |
| 12 |  | System responds with the report details page. |
| 13 | User clicks approve report request. |  |
| 14 |  | System updates the report to approved. |
| 15 |  | System responds with a confirmation page. |

Decline Report Request

|  |  |  |
| --- | --- | --- |
|  | Approve Report Request |  |
| Step | Actor Action | System Response |
| 1 | User clicks login |  |
| 2 |  | The system prompts users for their username and password. |
| 3 | The user enters the details specified in step 2. |  |
| 4 |  | The system verifies the user’s username and password. |
| 5 |  | The system responds with the home page |
| 6 | User clicks report page. |  |
| 7 |  | System checks user permissions are valid. (Is the user allowed to access the search page). |
| 8 |  | System returns all report requests. |
| 9 |  | The system responds with the report page. |
| 10 | User selects an individual report. |  |
| 11 |  | The system searches the database for report details. |
| 12 |  | System responds with the report details page. |
| 13 | User clicks decline report request. |  |
| 14 |  | System updates the report to declined. |
| 15 |  | System responds with a confirmation page. |

Self Diagnosis

|  |  |  |
| --- | --- | --- |
|  | Self Diagnosis | Patient |
| Step | Actor Action | System Response |
| 1 | User clicks login |  |
| 2 |  | The system prompts users for their username and password. |
| 3 | The user enters the details specified in step 2. |  |
| 4 |  | The system verifies the users username and password. |
| 5 |  | The system responds with the home page. |
| 6 | User selects the self diagnosis page. |  |
| 7 |  | System checks user permissions are valid. (Is the user allowed to access the self diagnosis page). |
| 8 |  | The system responds with the self diagnosis page. |
| 9 | User enters symptoms into page |  |
| 10 |  | System checks validity of details, e.g. symptoms are inputted correctly and only contain letters |
| 11 |  | System searches database for diseases with matching symptoms |
| 12 |  | Systems displays matching diseases |
| 13 | User selects Disease |  |
| 14 |  | System searches database for details on selected disease. |
| 15 |  | System displays disease information |

Add Diagnosis

|  |  |  |
| --- | --- | --- |
|  | Add Diagnosis |  |
| Step | Actor Action | System Response |
| 1 | User must first go through the stages of ‘View Person’ |  |
| 2 | Person selects diagnose button |  |
| 3 |  | System checks user permissions are valid. (Is the user allowed to access the diagnose patient page). |
| 4 | User enters symptoms into page |  |
| 5 |  | System checks validity of details, e.g. symptoms are inputted correctly and only contain letters |
| 6 |  | System searches database for diseases with matching symptoms |
| 7 |  | Systems displays matching diseases |
| 8 | User selects Disease |  |
| 9 |  | System searches database for details on selected disease. |
| 10 |  | System displays disease information |
| 11 | Use selects add disease |  |
| 12 |  | System saves disease to patient record on the database |

Delete Diagnosis

|  |  |  |
| --- | --- | --- |
|  | Delete Diagnosis |  |
| Step | Actor Action | System Response |
| 1 | User must first go through the stages of ‘View Person’ |  |
| 2 | User selects disease button |  |
| 3 |  | System checks user permissions are valid. (Is the user allowed to access the patient diseases page). |
| 4 |  | System searches the database for the diseases linked to the selected person. |
| 5 |  | The system displays search results (disease name, delete button, add treatment button, add prescription button.) |
| 6 | User presses the delete button next to the disease they want to remove from record. |  |
| 7 |  | System checks user permissions are valid. (Is the user allowed to delete a diagnosis from a patient record). |
| 8 |  | System deletes diagnosis from selected patient records on the database. |
| 9 |  | System displays updated diseases page. |

View Disease

|  |  |  |
| --- | --- | --- |
|  | View Disease |  |
| Step | Actor Action | System Response |
| 1 | User must first go through the stages of ‘View Person’ |  |
| 2 | User selects disease button |  |
| 3 |  | System checks user permissions are valid. (Is the user allowed to access the patient diseases page). |
| 4 |  | System searches the database for the diseases linked to the selected person. |
| 5 |  | The system displays search results (disease name, delete button, add treatment button, add prescription button.) |
| 6 | User selects disease |  |
| 7 |  | System searches database for information on selected disease linked to selected person |
| 8 |  | System displays disease information |

Add Treatment

|  |  |  |
| --- | --- | --- |
|  | Add Treatment |  |
| Step | Actor Action | System Response |
| 1 | User must first go through the stages of ‘View Person’ |  |
| 2 | User selects disease button |  |
| 3 |  | System checks user permissions are valid. (Is the user allowed to access the patient diseases page). |
| 4 |  | System searches the database for the diseases linked to the selected person. |
| 5 |  | The system displays search results (disease name, delete button, add treatment button, add prescription button.) |
| 6 | User selects add treatment next to diagnosis they would like to add it to. |  |
| 7 |  | System checks user permissions are valid. (Is the user allowed to add treatment to a diagnosis). |
| 8 |  | System searches the database for the treatments linked to the selected diagnosis. |
| 9 |  | The system displays matching treatments to the selected disease |
| 10 | User selects treatment |  |
| 11 |  | System adds treatment to patient record on the database |

Modify Treatment

|  |  |  |
| --- | --- | --- |
|  | Modify Treatment |  |
| Step | Actor Action | System Response |
| 1 | User must first go through the stages of ‘View Person’ |  |
| 2 | User must go through the stages of ‘View Disease’ |  |
| 3 | Select button to modify treatment |  |
| 4 |  | System checks user permissions are valid. (Is the user allowed to modify treatment of a diagnosis). |
| 5 |  | System searches the database for the treatments linked to the selected diagnosis. |
| 6 |  | The system displays matching treatments to the selected disease |
| 7 | User selects treatment |  |
| 8 |  | System adds treatment to patient record on the database |

Delete Treatment

|  |  |  |
| --- | --- | --- |
|  | Delete Treatment |  |
| Step | Actor Action | System Response |
| 1 | User must first go through the stages of ‘View Person’ |  |
| 2 | User must go through the stages of ‘View Disease’ |  |
| 3 | Select button to delete specific treatment |  |
| 4 |  | System checks user permissions are valid. (Is the user allowed to delete treatment of a diagnosis). |
| 5 |  | System deletes treatment from patient record on the database |
| 6 |  | System displays disease page. |

View Treatment

|  |  |  |
| --- | --- | --- |
|  | View Treatment |  |
| Step | Actor Action | System Response |
| 1 | User must first go through the stages of ‘View Person’ |  |
| 2 | User must go through the stages of ‘View Disease’ |  |
| 3 | Use selects treatment |  |
| 4 |  | System checks user permissions are valid. (Is the user allowed to view treatment of a diagnosis). |
| 5 |  | System searches the database for the treatment selected. |
| 6 |  | System displays information on treatment. |

View Prescription

|  |  |  |
| --- | --- | --- |
|  | View Prescription |  |
| Step | Actor Action | System Response |
| 1 | User clicks login |  |
| 2 |  | The system prompts users for their username and password. |
| 3 | The user enters the details specified in step 2. |  |
| 4 |  | The system verifies the users username and password. |
| 5 |  | The system responds with the home page. |
| 6 | User selects the prescription page. |  |
| 7 |  | System checks user permissions are valid. (Is the user allowed to access the prescription page). |
| 8 |  | The system responds with the prescription page. |
| 9 | User enters possible search details of person (name, postcode) |  |
| 10 |  | System checks validity of details |
| 11 |  | System searches database for person with matching details inputted by the user |
| 12 |  | The system displays search results (person name) |
| 13 | User selects a person record |  |
|  |  | The system displays all prescriptions associated with patients and the diagnosis associated with prescription |

Add Prescription

|  |  |  |
| --- | --- | --- |
|  | Add Prescription |  |
| Step | Actor Action | System Response |
| 1 | User must have selected patient and been given diagnosis before allowed to add prescription |  |
| 2 | User selects the prescription page. |  |
| 3 |  | System checks diagnosis information provided. |
|  |  | System searches the database for prescriptions. |
| 4 |  | The system responds with the prescription page with potential prescriptions. |
| 5 | User selects to which prescription to prescribe and to order prescription |  |
| 6 |  | The system checks inputs validity and displays message to confirm prescription (showing patient, diagnosis and prescription) |
| 7 | User confirms prescription |  |
| 8 |  | System adds prescription to database |

Delete Prescription

|  |  |  |
| --- | --- | --- |
|  | View Location |  |
| Step | Actor Action | System Response |
| 1 | User clicks login |  |
| 2 |  | The system prompts users for their username and password. |
| 3 | The user enters the details specified in step 2. |  |
| 4 |  | The system verifies the users username and password. |
| 5 |  | The system responds with the home page. |
| 6 | User selects the prescription page. |  |
| 7 |  | System checks user permissions are valid. (Is the user allowed to access the prescription page). |
| 8 |  | The system responds with the prescription page. |
| 9 | User enters possible search details of person (name, postcode) |  |
| 10 |  | System checks validity of details, e.g. postcode is sensible, input exists |
| 11 |  | System searches database for person with matching details inputted by the user |
| 12 |  | The system displays search results (location name, prescription) |
| 13 | User selects a person/prescription record to delete |  |
| 14 |  | The system displays a confirmation message/page to confirm you want to delete location |
| 15 | User clicks to confirm delete of prescription |  |
| 16 |  | The system updates the prescription to inactive |
| 17 |  | The system returns back to prescription screen |

* **Medication information**

|  |  |  |
| --- | --- | --- |
| **Step** | **Actor Action** | **System Response** |
| **1** | **User clicks login** |  |
| **2** |  | **The system prompts users for their username and password.** |
| **3** | **The user enters the details specified in step 2.** |  |
| **4** |  | **The system verifies the users username and password.** |
| **5** |  | **The system responds with the home page.** |
| **6** | **user selects Request information page** |  |
| **7** |  | **System checks user permissions are valid. (Is the user allowed to access the search page).** |
| **8** |  | **System responds with requested page.** |
| **9** | **User selects medication search** |  |
| **10** |  | **System responds with requested page.** |
| **11** | **User Inputs medication**  **details** |  |
| **12** |  | **System checks validity of details, e.g. date of birth is sensible, input exists** |
| **13** |  | **System searches the database for the possible matching results.** |
| **14** |  | **Systems displays the results to the user.** |
| **15** | **User select the most suitable option** |  |
| **16** |  | **System displays more in-depth information about the medication chosen.** |
| **17** | **User confirm the request** |  |
| **18** |  | **System displays a message that information was revealed successfully and go back to home page** |

Lesson Notes:

Static tests

* Review of the code, program, processes
* Test conducted without executing any program
* Static analysis
  + Analyse artefact to gain more info about it
  + White box testing and source code review
* Document review
* Walkthroughs
* Peer reviews
* Inspection

Dynamic tests

-Code is being executed and carried out when application is running

-Pick up runtime errors

* Functional Test
  + Unit testing
    - Testing individual software component or module
  + Integration testing
    - Plug multiple units together
  + System testing
    - Testing entire system
    - AKA black box testing
    - System as a whole component
  + Acceptance testing
    - Testing software against system requirements
    - Involves the users of the system
* Non-Functional Test
  + Look at non functional requirements
  + Performance testing
    - Load and stress tests etc
  + Security testing
    - Test system for confidentiality, authentication, integrity and availability.
  + Usability testing
    - Users ability to use software
    - Learnability, efficiency, satisfaction, memorability, errors
  + Compatibility testing
    - Check it works with specified software
    - Include OS, web browser etc

To do:

* Plan out how we are going to do our static tests

Feedback :

* Concept map needs to be more connected.
* Need more than one focus question (we have just a general question)
* Sub questions

Risk analysis:

* Cost is meant to be in £
* Put the risks into groups.
* Same endpoint just different ways to get to them (eg: loss of files can happen in multiple ways).

Gantt chart:

* The summary of the tasks should have a time frame of all the activities put together

Double check WBS and gantt chart match

System requirements

* The sub sections need number ranking inside

Class diagram:

* Add patient id, doctor id etc

Domain model

* Delete the row columns

Add inherited classes to the domain model

Architecture design needs to be completed