Development of a tool for prediction of rare diseases based on patient symptoms

By

Rasmus Meldgaard	496341
Simon dos Reis Spedsbjerg	177609698

Mandatory Assignment 1
Course: SE6-POM
Term: Spring 2023
Handed in: March 10, 2023

The Maersk Mc-Kinney Moeller Institute
University of Southern Denmark

This tool is meant to help a general practitioner in their task of diagnosing patients. The team is without higher management and therefore not included in stakeholder analysis.

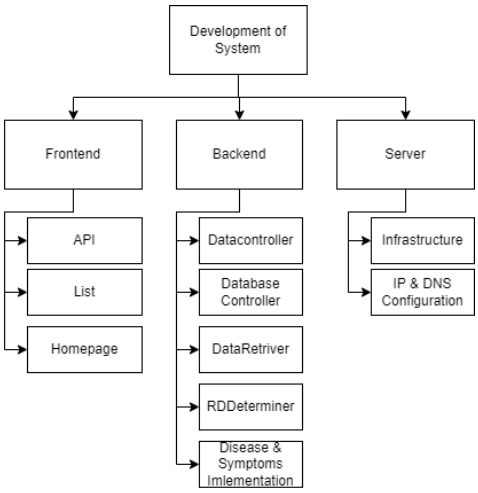
Stakeholder	Interest
General Practitioners	They are our target demographic as they use the service to provide better healthcare to their clients. Developing our program will be able to increase the confidence of their diagnoses.
Developers	They are interested in it because their business and thereby wages depends on it, delivering a good product could gain them good advantages when it comes to income and or future job prospects.
Healthcare companies	The companies that hires the General Practitioners are the one who will most likely be paying for our service, their interest is whether this will earn them more revenue either in the long run or short term.

The general partitioners is the one who would get most excited for this application, but is not expected to have much power. They will be kept informed about the program and it's features.
The Healthcare companies have a lot say in what software they will be using, and they have an interest in providing good cheap care. If we have a product which improve their service, a high interest is formed. This means that the healthcare companies are to be managed closely during the development to achieve success. Regular detailed meetings is therefore a must for this stakeholder.

In this project, the group are using a Top-Down estimation process because of the following reasons:

- It's the first time the group develops a project for healthcare and thereby don't know completely what is involved yet.
- It's a small project and the group only have two developers, the scope of the project is small.
- There is no contract involved.
- There is no current customer, only general practitioners who have offered to guide and therefore no details are needed for the customers at the current iteration.

For the development of the product, it is split up in 3 section in the WBS, Frontend, Backend and server.



The Frontend contains 3 task, API, to get the data from the backend. The list is showing the probability of different diseases and includes the function to format the disease in the certain orders depending on the practitioners wish. DataController handles the data between frontend and backend and RDDeterminer is the one that determines the risk of rare diseases. The program is webbased and has to be hosted on server which needs infrastructure setup and IP & DNS configuration.

Task	Dependency
Gather Theory	
Analysis	Theory
Design	Analysis
ISymptom	
Region	
IDisease	
Symptom	ISymptom, Region
Disease	IDisease
DatabaseController	Disease, Symptom
DataController	DatabaseController
DataRetriver	DatabaseController
IDeterminer	
RDDeterminer	IDeterminer, DataRetriver
API	DataController
Homepage	
- Infrastructure	
IP & DNS	Homepage
Write Test	Design
Review Software	RDDeterminer, API, IP & DNS, Write Test
Documentation	Review Software

This table includes all task in development of this project, Documentation includes all task after the design phase.

	Project Overview Statement	Rare disease Predictor		
Problem/Opportunity	We see a risk that a healthcare professional might miss certain rare conditions a patient might suffer from, because the rarity and lack of knowledge makes it so the doctor or nurse might not suspect it and thereby miss it.			
Goal	To develop an application that will help the health care professional to deduce whether a patient suffers from a rare disease			
Objectives	• Develop a Backend system to calculate diseases based on symptoms • Develop a Frontend to allow interaction from any computer • Establish a server to host a beta version • Deploy public version			
Success Criteria	• The system can predict diseases based on given symptoms • The system is positively received by general practitioners • The system is being used by general practitioners			
Assumptions, Risks, Obstacles	• All general practitioners have access to the internet • Server Errors might occur • Some general practitioners don't have the required computer knowledge to use the system			
		07 – 03 – 23		

Very High (71-90%)	Healthcare tech companies do not want to add the product to their catalogue of software.	The development team is composed of students meaning there is a general inexperience for developing.			
High (51-70%)			Lack of knowledge in the field area could lead to a worse quality product		
Medium (31-50%)		Some general practitioners might not have the computer knowledge to use the system.	The lack of data for diseases and symptoms.	Since it is new territory to develop medical based software, there might be certain elements we do not know of that would add elements out of scope	
Low (11-30%)					
Very Low (<10%)	The algorithm could produce a faulty result.		GP's might not use the product due to it not being on their main system.	The servers can go down leading to down time.	
Likelihood / Impact	Very Low	Low	Medium	High	Very High

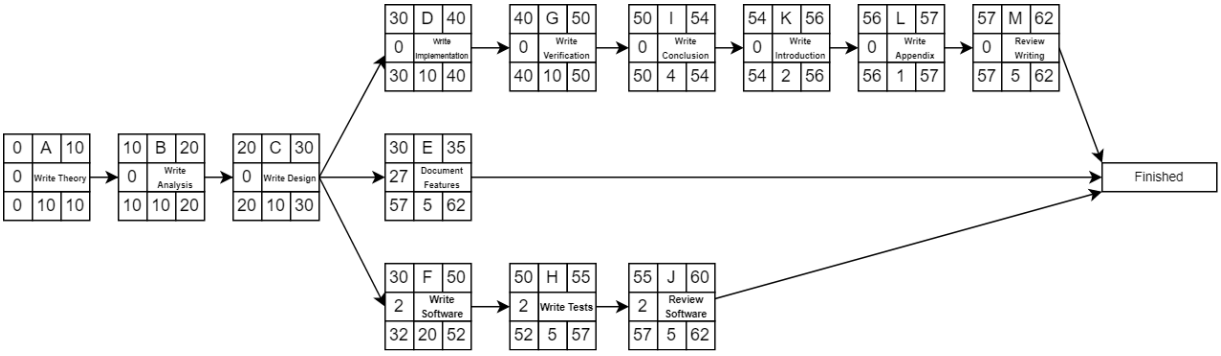
There are ways to combat some of the risks involved in the project, but some of them are risks we must accept comes with the creation of the product.

- For the risks involving the usage of the product and problems possibly arising with it, it would be possible to try and avoid that risk with the creation of a better user interface to make it easier for GP's who does not have a lot of general computer knowledge.
- For the risks involving inexperience with the medical software field, the only way to try combat it, would be to try and use the slack time to the best of the teams ability, if the slack time is used up, the function or any other non critical functions has to be pushed into the next iteration and then be completed there to ensure the iteration is finished at the deadline.
- The risk involving the actual system and algorithm, the risk can be reduced if not removed by creating multiple tests to try and break the algorithm. If the algorithm passes the tests, the risk likelihood should be lower.
- For the risks of having the servers go down, that is something we have to accept that is always a possibility that can happen for any system, the only way to reduce the chances for this to happen, would be to choose a reliable server host which has the least amount of downtime over a longer period.
- For the risk involving the lack of data, a way to avoid it making or breaking the system would be to create dummy data in its place, if the data is not acquired. Granted this will end up with a result that cannot be practically used, it would end up with a system that can showcase the functionality required of it, and it would be a simple case of swapping around data sets.
- For the Health companies not adding it to their system or GP's not wanting to use it if its not on their system is another risk we just must accept. It is not in our hands if the company does not want to add it nor if the GPs don't want to use anything outside their known toolbox. Having close contact and keeping health companies well informed does decrease the likely hood of them not adopting the software.

For the task in this project, each has been given an estimated time. This time is based on previous experience on other projects. The time that is estimated to be under 1 is calculated in total as an half.

Task	Estimated Time
Gather Theory	10
Analysis	10
Design	10
ISymptom	< 1
Region	< 1
IDisease	< 1
Symptom	< 1
Disease	< 1
DatabaseController	2
DataController	2
DataRetriver	< 1
IDeterminer	< 1
RDDeterminer	14
API	1
Homepage	< 1
- Infrastructure	< 1
IP & DNS	1
Write Test	5
Review Software	5
Documentation	32
Total	96

After which COCOMO calculation[1] has been used to increase or decrease confidence in our estimations. The estimations was based on the teams view of their own abilities. The project is expected according to the calculations to take 2.2 man months which aligns with our own expectations for a two man development team. The reasons for choosing an estimation based on our own experience, is due to the other estimation methods are still subjective in how the data is formed. The reason for the use of COCOMO is because the team is confident in the projects size which fits with COCOMOs line variable.



The diagram above is a network diagram created in the node style for the project. All the timeframes are approx. estimations we assumed how long each part would take. This network diagram is created with the entire bachelor as a project, where report writing creating the software and anything else required is included.

- The earliest finish date based on the network diagram would take 62 calendar units.
- The critical path in this network diagram is the following sequence.
  - A→B→C→D→G→I→K→L→M

References

[1] Nasa. *COCOMO Calculation*. 2023. URL: https://strs.grc.nasa.gov/repository/forms/cocomo-calculation/.