Problem Specification

Since patients cannot always visit the same doctor or if they do visit their doctor regularly the doctor seeing them may not be aware of pass ancients that could potentially be linked to their current symptoms/ illness. If a symptom seems to be nothing a doctor who is unaware of the patients pass medical history may over look this symptom without realising the links to pass symptoms and illness. Since certain diseases are difficult to diagnose as they show very few symptom or these symptoms build over the scale of months or years. Not know the patients fully medical history in advance could seem symptoms that point to serious other lying illness go unnoticed. As it is possible for doctors to know the full medical history of every single patient they see and this would be a very repetitive task after or whilst seeing a patient to read over all of their records in case a symptom which seems small has the potential to be more serious.

There are other symptom checker alternatives on the market however they rely on doctors’ knowing the symptoms and manually writing out these symptoms so that they can be analysed, to see if they match the symptoms of other diseases. Although these programs are powerful and can compare symptoms against the known symptoms of other diseases. This approach can be problematic as it assumes that the doctor already knows all of the patients’ symptoms. Also if he or she feels that the problem is too small to deem further inquiry than there is the risk that serious illness could potentially go unnoticed.

A software application which records and monitors patterns or symptoms as a doctor is taking their notes means that the problem of whether or not a he or she deems something to be important or not affect the diagnose software, everything would be taken into account. For example, if symptoms that appeared over the course of a years might not show importance until diagnosed together, so until on their own they are not important the recording and monitoring system may be able to spot an underlying pattern. Since the application would recording the patients information in search of threads if it found the symptoms matched up the application cold notify the doctor as they are recording the patients’ notes. Meaning that even if the doctor is new to this patient or unware of relating pass information they will be aware of this thread.

Although there is no need to re-invent the wheel and create a new surge system for doctors when they already has a functioning surge system to take their patients notes on. It would be useful to has a prototype surge system were it could be demonstrated how the recording and monitoring application would fit in with the current surge system. The focus of the design is to create an alert system which gets the attention of the professional user however does not alarm the patient. So the warning needs to be subtle but effective. Also if the professional user selects the

Project Plan

Also there will be iterations through this project as testing will throughout the project instead of starting near the end these deadlines are a marker of when certain actives should be completed. So the last 20 days should be reversed for making sure everything is completed and functional to make sure there are no loose ends.

Deadlines

6th of July – Problem specification

Dissertation

16th of July – Requirements and user stories (10 days)

26th of July – Design (10 days)

4th of September – Implementation (40 days)

22th of September – Testing (19 days)

23rd September – Project Dissertation, Code and Journal (80 days)

24th September – Project Demonstration and Expo

Abstract

This is a summary (100 words) outlining the subject matter and conclusions of the dissertation.

Contents

A complete list of chapters, sections, appendices etc. with page numbers.

Main Body of the Dissertation

The following section outlines the sequence of chapters that should be included in the dissertation. Some dissertations may be organized differently from the chapter headings given below. However, you should attempt to address each of the following aspects in your dissertation. Further advice will be available from your supervisor.

Introduction

This should outline the structure and key sections of the rest of the document.

Chapter 1: Problem Specification

This should primarily consist of the original specification (as outlined in the initial Problem Specification / Project Plan submission). Any changes to the original specification should be identified and explained. The student must clearly describe the perceived problem and the target audience. It should be obvious from the chapter that the student has a thorough understanding of the problem domain and current applications used (if available) to address the problem at present.

Chapter 2: Proposed solution and justification of the development model

This chapter should clearly follow on from where chapter 1 left off. The proposed solution must be sensible and clearly justified and should demonstrate some flair and originality. The development strategy must be clearly described, adequately justified and appropriate. Also included within this chapter should be a clear identification of the design details in line with the chosen development strategy.

Chapter 3: Requirements analysis and specification

This chapter should explain and justify the process by which the requirements have been elicited. It should then also clearly identify the requirements of your project, which can be later tested. Depending on the chosen development strategy this chapter may be written retrospectively. The student’s academic supervisor will provide additional advice on this where required. The following are some of the details which would be expected in this chapter:

\* A clear and precisely defined data model

\* A complete set of function definitions

\* Details of error conditions

Chapter 4: Design

This chapter should describe, illustrate and justify the design of your proposed system. This should normally have two parts:

1. User Interface Design

2. Software System Design: the system should be decomposed into components with the interfaces between components and the role of each component described. A clear correlation should exist between your design and your specification.

Chapter 5: Implementation

The methodology used in the development of the proposed project has to be clearly described in this chapter with a description of the main functions of the software. It should include:

\* A precise specification of the most significant aspects of the systems implementation. This may include detailed descriptions of the implementation of important components, functions or algorithms.

\* Specification and justification of the testing strategies employed

In keeping with the principle of making the dissertation as readable and fluent as possible, it is not necessary to present all of the detailed program code in the main text. Some illustrative examples of the most significant sections of code would suffice. The remainder of the program code can be presented as a listing in one of the appendices. The program listing must be appropriately annotated with comments and each component should have a complete module header. Programs should be written in a clear style with good program structure and well-defined data structures. The program code should reflect its design.

Chapter 6: Evaluation and Conclusion

The evaluation chapter should demonstrate that the student understands the process of evaluation and testing in Software Development and can relate this to the project being developed. This section should include a general evaluation of the success of the project measured against the criteria stated in the introduction and/or requirements. An evaluation of the hardware/software environment and language used may also be presented. Suggestions for further work should also be discussed. Do not be afraid to be critical or to draw a negative conclusion; not all projects will be successful. This section should provide a thorough and honest reflection on the process followed in the project and the results of that process. To do this well, the student should not leave any blind spots in their reflection and should identify the most and least successful aspects of the project.

References/Bibliography

A list of relevant references to other documents (books, papers, web-sites, etc.) which are referred to in the main body of the text.

For books, record:

\* The author’s name or the editor’s name

\* The year the book was published

\* The title of the book

\* The edition of the book

\* The name of the publisher

\* The ISBN of the book

For electronic resources, record:

\* The author’s name or the editor’s name where available

\* The year the article was published where available

\* The title of the article

\* The date the source of the information was accessed

\* The electronic address or email

\* The type of electronic resource (email, discussion forum, WWW page, etc)

Appendices

These may include:

\* Test results, if appropriate.

\* Printout of code for section/sections of the application developed in relationship to the submission.

