Patient Handling System: Final Report

Contents

[Introduction 1](#_Toc447238815)

[Technical Review and Previous Work 3](#_Toc447238816)

[Technical Issues 3](#_Toc447238817)

# Introduction

Handling patients who have mobility problems, is not surprisingly, an intricate task. It involves the consideration of a number of different variables, and the use of many different movement tasks, with different processes required for each. The patients themselves, for one thing, can vary significantly in a number of ways, such as in weight, co-operation level, weight bearing capacity, whether they are injured or not, and more. Other factors that also need to be included are the equipment available to the patient handler, and the features of each one. With different movement tasks, the status of some or all of these attributes need to be included in the final action taken to move the patient. Neglecting any could result in poor outcomes, ranging from injury to the patient or patient handler, to undue stress for both parties.

To aid patient handlers with the above process, a series of algorithms were designed by [Name of company?], which guide the patient handler through an algorithm, in which the path taken through the algorithm, and the eventual solution, or handling plan generated by the algorithm is based on the value of various different patient and equipment attributes. Figure 1.1 shows an example of such an algorithm.

Image of Algorithm 4

Figure 1.1

In total, there are sixteen algorithms that a patient handler has to either learn off or have on their person at all times when on duty. They also may need to have on hand the various patient/equipment attributes, such as weight, co-operation level, etc. in order to iterate through the algorithm. This results in a process that, if due care was not taken, could produce many errors - for example if the patient handler was under time constraints and guessed some of the patients attributes, or if they read a logical operator incorrectly. The purpose of creating an application for this process is to avoid these errors, which would lead to a better quality of service, and ultimately more satisfied patients and patient handlers.

With all of the above in mind, it is clear that there is a need for an application that would aid in this process. This application could be beneficial in different ways, with the main one being that the application would iterate through each algorithm and could produce handling plans for a set of patients. How this is integrated into the patient handler’s day to day job could vary, however a likely scenario is that they would print off the required handling plans for each patient, and store them in an easily accessible place next to the patient. Other implementations would also be possible, such as using a tablet to access the handling plan for a patient. This would be a more expensive option however, but could be more effective if the patient or equipment attributes changed regularly.

This information led to a decision that a web application would be the best type of implementation for these particular requirements. Given the sensitive data that may be stored in this application it would be best if the handling facility had its own internal network where application could be hosted. However, this may not be possible in some facilities, and in that case, the application and associated database could then be hosted on cloud based services, such as Microsoft Azure, which, on a basic subscription plan, consisting of a web application and a database, would cost €600 annually. The web application itself would store all of the patients’ details, along with the various handling plans, which would both be editable.

# Technical Review and Previous Work

There have been two previous attempts at building an application for the company for this specific process. Both attempts were done by final year students, and were functioning software, were never brought into a working environment by the company. Having access to the reports for both projects (however, not the actual software itself), the previous work done provided a good resource for the work on this application. Mainly, the reports provided an excellent introduction to the problem domain. The software developed for this report was quite different than the software developed by the other two attempts, so for this reason, the previous work was not as beneficial as could have been hoped.

From analysing the previous work done, it was clear there were some advantages and disadvantages to the approaches taken by both. Ultimately, however, it was decided to go with the proven environment of ASP.NET MVC. This is a popular choice when building applications with many CRUD (Create, Read, Update and Delete) type operation, which make up a significant portion of this one – i.e. the storing and editing of patient, user, and equipment data.

# Technical Issues