Project Hospital 2006

Table of Contents

- Organisation
- Primary Goals
- System Overview
- User Interaction
- Functional Specifications
- Hospital Scheduling System: The User's View
- HSS vital information and attributes
- Credits

Organisation

HealthCare is private hospital owned by a group of leading doctors specialised within hip operations and ear, nose and throat operations. HealthCare started with 15 employees in total and has expanded every year since the foundation of the company in 2000. Today the hospital is staffed with 49 full-time employees: 14 doctors, 30 nurses, and five administrative persons. Two of the administrative employees (receptionists) are hired from a recruitment company. The hospital consists of:

- Two wards with three operation rooms each
- 68 waiting beds divided in 20 rooms
- 24 recovery beds divided in 12 rooms

Primary Goals

There are two primary goals that the development of the Hospital Scheduling System (HSS) should meet:

1) Allocation of time/human resources.
2) Allocation of time/space resources.

To clarify, time/human resources reflect the basic scheduling of employees (both doctors and nurses), while time/space resources include the overall control of patient traffic within wards to ensure precise bed allocation and estimations of bed availability, as well as the scheduling of expected and unexpected operations. The desired system should be able to manipulate either of these resources in a variety of ways.

System Overview

The HSS (Hospital Scheduling System) should provide its users a plethora of scheduling abilities of which include:

- reliable operation scheduling and updating of waiting lists, based on patient status.

- automatic determination of bed availability status for wards.

- advance employee schedule viewing inclusive of on-call status.

- password security system for employees.

With the HSS assuming the majority of the hospital scheduling responsibilities, the system's users should be able to: spend less time amidst the shuffling of paper, spend more time with patients, have access to immediate and accurate information on the status of hospital functions, and in turn, generate confidence within the workplace. As a result, hospital efficiency will increase on the whole. Users of the HSS should also be able to find transition to the new system a gentle one with lucid, easy to understand commands and actions. This will establish a comfort with the HSS, rather than an alienation which so many other systems generate. With this combination of user confidence and comfort fostered by the HSS, the daily on-goings of the hospital will be simple and orderly.

User Interaction

To further detail the interpretation of the Hospital Scheduling System, we are providing a listing of the functions recognized as vital to the system's operation. A distinction between wards is identified and split into two basic categories: operating rooms (O/R) and wards. While most of the requirements are similar, recovery time has to be taken into consideration for functions which dealt with O/R. This add another level of complexity to the scheduling techniques necessary for the HSS, but in general to cope with scheduling from the O/R perspective a meeting number should be assigned to all operations to monitor all present and all upcoming scheduled operations. The designation of this unique meeting number will allow for easy cancellation and re-bookings for operations.

Functional Specifications

1. Schedule Operation

1.1 Schedule Patient to Operation Room

Function: A patient is scheduled to an operation room for a given treatment. A recovery room is also assigned.

Inputs : Operation room number, operation date/time, operation duration, recovery date/time, recovery duration, treatment, patient ID number, doctor(s) ID, nurse(s) ID

Error Conditions:

- Syntax errors: flag user to re-enter data
- Schedule conflicts: flag user to re-enter date/time
- Patient ID does not exist: prompt user for re-entry
- Doctor ID does not exist: prompt user for re-entry
- Nurse ID does not exist: prompt user for re-entry
- Recovery room schedule conflicts: prompt user for schedule re-entry
- Operation room schedule conflicts: flag user to re-enter date
- Treatment does not exist: flag user to re-enter treatment
- Recovery room number does not exist: flag user to re-enter room number
- Operation room number does not exist: flag user to re-enter room number
- Patient name does not exist: flag user to re-enter name
- Operation date/time is outside of Recovery date/time: prompt the user to re-schedule both dates/times

- Operation/Recovery dates/times are made too far in advance: prompt the user to re-schedule dates/times

1.2 Modify Previously Scheduled Operation

Function: Any of the original inputs for an existing operation may be updated. This function will automatically remove all staff from the operation roster if the date/time is modified.(Staff schedules will have to be re-entered)

Inputs : Operation room number, operation date/time, operation duration ,recovery date/time, recovery duration, treatment, patient ID number, Doctor ID, Nurse ID, Meeting number

Error Conditions:

- Syntax errors: flag user to re-enter data
- Schedule conflicts: flag user to re-enter date/time
- Patient ID does not exist: prompt user for re-entry
- Doctor ID does not exist: prompt user for re-entry
- Nurse ID does not exist: prompt user for re-entry
- Meeting number does not exist: prompt user for re-entry
- Recovery room schedule conflicts: prompt user for schedule re-entry
- Operation room schedule conflicts: flag user to re-enter date
- Treatment does not exits: flag user to re-enter treatment
- Recovery room number does not exist: flag user to re-enter room number
- Operation room number does not exits: flag user to re-enter room number
- Patient name does not exist: flag user to re-enter name
- Operation date/time is outside of Recovery date/time: prompt the user to reschedule both dates/times
- Operation/Recovery dates/times are made to far in advance: prompt the user to reschedule dates/times

1.3 Operation Cancellation

Function: An operation is canceled for a given patient. This function will clear recovery room schedules, as well as operation schedules and any schedules of staff that have been added to the operation.

Inputs : Meeting number

Error Conditions:

- Syntax errors: flag user to re-enter data
- Meeting number does not exist: prompt user for re-entry

1.4 Assign Staff to Operation

1.4.1 Assign Doctor to Operation

Function: Assign a doctor to an operation. Checks for doctor schedule conflicts.

Inputs : doctor ID, operation date/time, room number, Meeting number

Error Conditions:

- Syntax Errors: prompt user to re-enter data.
- doctor ID does not exist: flag user to re-enter data.

- operation date/time does not exist: prompt user to re-enter date/time.
- room number does not exist: prompt user to re-enter room number.
- doctor schedule conflicts: prompt user to either use a different doctor or exit the function and advise user to change doctor schedule.
- doctor ID and meeting number already exist in combination:
- prompt user to re-enter data.

1.4.2 Remove Doctor from Operation

Function: Deletes a doctor from an operation.

Inputs : doctor ID, Meeting number

Error Conditions:

- Syntax Errors: prompt user to reenter data.
- doctor ID does not exist: flag user to reenter data.
- meeting ID does not exist: prompt user to reenter data.
- operation date/time does not exist: prompt user to reenter date/time.
- room number does not exist: prompt user to reenter room number.

1.4.3 Assign Nurse to Operation

Function: Assign a nurse to an operation. Checks for nurse schedule conflicts.

Inputs : Nurse ID, operation date/time, room number, meeting number

Error Conditions:

- Syntax Errors: prompt user to re-enter data.
- Nurse ID does not exist: flag user to re-enter data.
- Operation date/time does not exist: prompt user to re-enter date/time.
- Room number does not exist: prompt user to re-enter room number.
- Nurse schedule conflicts: prompt user to either use a different nurse or exit the function and advise user to change nurse schedule.
- Nurse ID and Meeting number already exist: prompt user to re-enter data.

1.4.4 Remove Nurse from Operation

Function: Removes a nurse from an operation.

Inputs : nurse ID, Meeting number

Error Conditions:

- Syntax Errors: prompt user to re-enter data.
- Nurse ID does not exist: flag user to re-enter ID.
- Meeting ID does not exist: prompt user to re-enter data.
- Operation date/time does not exist: prompt user to re-enter date/time.
- Room number does not exist: prompt user to re-enter room number.

1.4.5 Assign Nurse to Ward

Function: Assigns a nurse to a ward.

Inputs : nurse ID, ward number

Error Conditions:

- Syntax Errors: prompt user to re-enter data.
- Nurse ID does not exist: flag user to re-enter ID.
- Ward number does not exist: prompt user to re-enter number

2. Wait List

2.1 Assign Patient to Wait List

Function: Assigns a patient to the wait list

Inputs : patient ID, duration, treatment, priority number

Error Conditions:

- Syntax Errors: prompt user to re-enter data.
- Patient ID does not exist: prompt user to re-enter ID.
- Treatment does not exist: prompt user to re-enter treatment.
- Priority number does not exist: prompt user to re-enter number.

2.2 Remove Patient from Wait List

Function: Removes patient from the wait list.

Inputs : patient ID, queue number

Error Conditions:

- Syntax Errors: prompt user to re-enter data.
- Patient ID does not exist: prompt user to re-enter ID.
- Patient ID and queue number does not exist: prompt user to re-enter data.

2.3 Modify Wait List Information

Function: Changes Priority number or treatment for a given patient.

Inputs : patient ID, queue number, priority number, treatment, duration

Error Conditions:

- Syntax Errors: prompt user to re-enter data.
- Patient ID does not exist: prompt user to re-enter ID.
- Queue number does not exist: prompt user to re-enter ID.
- Treatment does not exist: prompt user to re-enter treatment.
- Priority number does not exist: prompt user to re-enter number.
- Patient ID and queue number does not exist: prompt user to re-enter data.

2.4a View Wait List Information

Function: Lists all patients currently awaiting an operation(s).

Inputs : none

Error Conditions: none

2.4b View Patient Wait List Information

Function: Lists a patient current waiting status for an operation(s).

Inputs : Patient ID

Error Conditions: none

- Syntax Errors: prompt user to re-enter data.
- Patient ID does not exist: prompt user to re-enter ID.

The following is a listing of reports which can be generated for operating room information. These include reports for: operation room schedules, employee schedules in O/R, daily patient reports in O/R, and patient operation reports.

3. Report functions for the O/R

3. 1 Create report for operation rooms

Function: a report is generated for operation rooms.

Inputs : room number, start date, <finish date> (optional)

Outputs : Reports operations to be done in the operation room listing in order of date and time of operation. Reports can be done daily, weekly, or any other length of time determined by start date and finish date of the search.

If no finish date is entered only a day's report is done for the room. This report will also list:

- the patient name/ID
- the doctor's name/ID
- the scheduled nurses name/ID
- the type of operation
- the length of the operation

Error Conditions:

- Syntax incorrect in date or room number.
- Start date is > finish date
- Date doesn't exist. i.e.: Jan. 32, 1997
- Room doesn't exist as a room or as an operating room.

3.2 Create report for employees

Function: generates report for employees' schedule

Inputs : doctor/nurse ID, start date, <finish date>

Outputs : Reports a doctor/nurse's schedule ordered by date and time. The report will be as long as is specified by start date and finish date. As well, if a finish date is not specified the function does a day's schedule for the employee.

The report will also list:

- length of operation or appointment
- the patient being seen or operated on i.e.: name/ID
- the type of operation or reason for the appointment.

Error Conditions:

- Syntax incorrect in date or room number.
- Start date is > finish date
- Date doesn't exist. i.e.: Jan. 32, 1997
- Room doesn't exist as a room or as an operating room.

3.3 Create report of daily patients

Function: generates report of patients for the day (for O/R)

Inputs : date

Outputs : lists patients for the day, alphabetically by name. Also reports:

- patient's ID
- length of operation
- doctor's name/ID.

Error Conditions:

- Syntax incorrect for the date.
- Date doesn't exist.

3.4 Create report for patient

Function: generates report for patient (query)

Inputs : patient name/ID#

Outputs : lists a patient's operation and appointments, listed in order of date. Searches on both patient's name and ID. Also lists:

- the doctor the patient is seeing.
- length of stay either for an operation or for a ward.

Error Conditions:

- syntax incorrect for patient name/ID#
- patient name/ID# doesn't exist

The next section details the functions deemed to be necessary for the HSS to monitor and manipulate data regarding wards other than the operating and recovery wards. This includes data regarding the three principle members of any hospital situation: patient, nurse and doctor.

4. Ward/Bed Functions (Basisløsning)

4.1 Function Name: Add a bed

Input: - All attributes of bed (availability field optional)

Output: - Will add a bed to the system

Can be added to any ward or room

default setting will make a bed available

Error conditions:

- Field input missing (all fields must be defined, except for availability)

4.2 Function Name: Delete a bed

Input: - Ward number, Room number, Bed number

Output: - Will delete a bed from system

Error conditions:

- bed currently being used (i.e. not available)
- bed scheduled for future use
- bed does not exist

4.3 Function Name: Change bed info

Input: - Ward number, Room number, Bed number, {fields that need to be changed}

Output: - Modifies bed fields

Used when beds moved around in the hospital

Also used to manually modify bed availability

Error conditions:

- bed does not exist

4.4 Function Name: Display bed info

Input: - Ward number, Room number(Optional), Bed number(Optional)

Output: - Displays bed information for any particular bed specified

Will display info based on parameters passed:

1. Ward number

2. Ward number AND room number

3. Ward number, room number, AND bed number

Option 1 will list bed info for all beds in the ward.

Option 2 will list bed info for all beds in a particular room.

Option 3 will list bed info for a specific bed.

Error conditions:

- Bed does not exist

5. Ward/bed reports (Basisløsning)

5.1 Report Name: Bed Availability Report (ward)

Input: - Ward number

Output: - list of beds still available in a certain ward specified by user

- also includes a total of the number of beds still available in the ward

Error conditions:

- Ward does not exist

5.2 Report Name: Bed Availability Report (entire hospital)

Input: - (none)

Output: - list of available beds for entire hospital separated by wards

also includes a total of the number of beds still available broken down by ward and the entire hospital

5.3 Report Name: Room Type Availability Report (ward)

Input: - Room type

Output: - lists all rooms of certain type specified by the user that still have available beds

- also lists the specific beds which are available

Error conditions:

- Room type does not exist

5.4 Report Name: Room Type Availability Report (entire hospital)

Input: - Room type

Output: - as above, except for the entire hospital, and broken down by ward

Error conditions:

- Room type does not exist

The functions listed below outline the basic functions needed to manipulate and keep track of information regarding nurses within wards.

6. Nurse Functions (Basisløsning)

6.1 Function Name: Add a Nurse

Input: - all nurse fields, except for nurse number

Output: - adds a nurse into the database

- assigns a unique nurse number to each nurse

Error conditions:

- Field input missing (all fields must be defined)

6.2 Function Name: Delete a Nurse

Input: - nurse number

Output: - deletes a nurse from the database

Error conditions:

- Nurse does not exist

6.3 Function Name: Change Nurse info

Input: - nurse number, {all fields that need to be changed}

Output: - modifies nurse info fields (i.e. change of name, phone numbers, etc)

Error conditions:

- Nurse does not exist

6.4 Function Name: Display Nurse info

Input: - nurse number or nurse name

Output: - displays a nurse's whole record

Error conditions:

- Nurse does not exist

7. Nurse Reports

7.1 Report Name: Daily schedule report

Input: - date or (none)

Output: - outputs a schedule for a particular day listing the nurses working that day - default is current day

Error conditions:

- Incorrect date

7.2 Report Name: Weekly schedule report

Input: - date or (none)

Output: - as above, except for an entire week, broken down by day - default is current week

Error conditions:

- Incorrect date

7.3 Report Name: Monthly schedule report

Input: - date or (none)

Output: - as above, except for an entire month, broken down by week and day - default is current month

Error conditions:

- Incorrect date

7.4 Report Name: Nurse list report

Input: - ward number or (none)

Output: - outputs a list of all nurses in a particular ward, or the entire hospital if no input is given

Error conditions:

- Ward does not exist

7.5 Report Name: On-call list

Input: - (none)

Output: - generates output listing all nurses that are on call currently

There are two types of patients who may stay at the hospital: a patient who has checked-in because of medical reasons, or the patient who is scheduled for an operation. The following set of functions allow for this distinction, as well as the ability to generate patient reports as specified below.

8. Patient Functions (Basisløsning)

8.1 Function Name: Add a patient

Input: - All patient fields

Output: - A unique identification number is assigned - Patient is added to the database

Error Conditions:

- Cannot add a patient which already exists
- Field input missing (all fields must be defined)

8.2 Function Name: Modify a patient

Input: - Patient identification number or patient name

Output: - Displays patient information - User able to change information in field(s)

Error Conditions:

- Patient doesn't exist

8.3 Function Name: Check-in patient

Input: - If patient is new then, Add a patient.

If patient is in database then, Modify a patient.

Specify ward that patient will stay in

Output: - Assigns patient a room and bed number specific to the ward

Error Conditions:

- No available beds in specified ward

8.4 Function Name: Check-out patient

Input: - Patient identification number

Output: - Displays patient information

Entered by user, an update of medical history

Makes specified bed available

Calculates total cost incurred (days of stay * cost per day)

Error Conditions:

- Patient does not exist in database; patient is not currently checked-in the hospital

8.5 Function Name: Display Patient Information

Input: - Patient identification number or patient name

Output: - Displays all information in patient fields

Error Conditions:

- Patient does not exist in database

8.6 Function Name: Patient Schedule

Input: - Patient identification number or patient name

Output: - Displays appointments with doctors or operation date/time(s)

Error Conditions:

- Patient does not exist in database

9. Patient Reports (Basisløsning)

9.1 Report Name: Patient Listing

Input: - Specify a particular ward(s) or entire hospital

Output: - Listing of all patients, their identification number, name, and location

Error Conditions:

- Ward specified does not exist

Each of the identified functions required for doctor information is listed below in detail, including the generation of reports.

10. Doctor Functions (Basisløsning)

10.1 Function Name: Add a doctor

Input: - All doctor fields

Output: - A unique identification number is assigned - Doctor is added to the database

Error Conditions:

- Cannot add a doctor which already exists
- Field input missing (all fields must be defined)

10.2 Function Name: Delete a doctor

Input: - Doctor identification number of doctor to be deleted, doctor identification number of replacement doctor

Output: - Remove the doctor from the database, re-assign doctor's schedule to another doctor

Error Conditions:

- Cannot delete a doctor which does not exist
- Cannot use a replacement doctor if they do not exist

10.3 Function Name: Modify a Doctor

Input: - Doctor identification number or doctor name

Output: - Displays doctor information, user able to change information in field(s)

Error Conditions:

- Doctor doesn't exist

10.4 Function Name: Display Doctor Information

Input: - Doctor identification number or doctor name

Output: - Displays all information in doctor fields

Error Conditions:

- Doctor doesn't exist

11. Doctor Reports

11.1 Report Name: Doctor Listing

Input: - Specify a particular ward(s) or entire hospital

Output: - Listing of all doctors, their identification number, name, and their specialization

Error Conditions:

- Ward specified does not exist

11.2 Report Name: Doctor Daily Schedule

Input: - Doctor identification number, date (default is current date)

Output: - Display the specified day schedule for the specified doctor

Error Conditions:

- Doctor doesn't exist

- Date is invalid

11.3 Report Name: Doctor Weekly Schedule

Input: - Doctor identification number,

week (default is current week - Sunday to Saturday)

Output: - Display the specified weekly schedule for the specified doctor

Error Conditions:

- Doctor doesn't exist
- Week is invalid

11.4 Report Name: Doctor Monthly Schedule

Input: - Doctor identification number, month (default is current month)

Output: - Display the specified monthly schedule for the specified doctor

Error Conditions:

- Doctor doesn't exist
- Month is invalid

11.5 Report Name: Doctor's Patient Listing

Input: - Doctor identification number

Output: - Lists all the patients currently under the supervision of specified doctor

Error Conditions:

- Doctor doesn't exist

Hospital Scheduling System: The User's View

The most important considerations in the design of the user interface are simplicity, consistency, clarity and speed of access. A difficulty that surfaces when designing any non-trivial system is the trade-off between simplicity and speed of access. For the experienced user, much time can be lost in a day wading through several levels of menus. On the other hand, menus provide the easiest method of navigating the system.
With the understanding that the system will be accessed by various hospital employees, we require a menu-based system. However, since the system will be used consistently by administrative personnel who will spend much of their time setting up the scheduling of operations and beds, we have limited the maximum depth of menus to three. In addition, we require "hot keys" (or shortcut keys) for direct access to most of the system tasks. This should allow rapid access to all essential system tasks.

From the user's perspective, we view the system as essentially two main functional units. The primary unit consists of the scheduling functions. For this reason, the HSS should provide direct access to these tasks from the main menu. The secondary unit is the supporting database, consisting of the patient, doctor, nurse, ward and operating room information. We expect that these entities will need to be updated from time to time, but less frequently than the main scheduling of beds and operating rooms. Nevertheless, the ability to make changes to any of the information stored should involve no more than two menu selections from the main screen.

As a customized system, the HSS should attempt to simplify the scheduling tasks as much as possible. We require that the user is permitted to arrange a time for an operation or a bed for a patient by providing a minimal amount of information, assuming that the patient data has already been entered. By using a graphical user interface (GUI), mouse support should also be available; this will further simplify the procedure.
The system should have a familiar look and feel to any experienced computer user. This, we believe, can accomplish this by using Java technology.

A final consideration in the design of the user interface is the generation of various reports. This system should have the capability to generate any report that is needed. In addition, there should be a set of standard reports that are predefined in the system. We require, for example, that operating room schedules can be printed out on demand.

HSS vital information and attributes

The list below contains examples of vital attributes that the system must store. By no means is this list exhaustive, but instead an indication of what we have recognized as vital information for some of the classes. Feel free to contact any member of our organization for clarification.

Doctors:

Basic doctor information:

- Name
- Adress
- Phone_number
- Pager_number

Nurses:

Basic nurse information:

- Name
- Adress
- Phone_number
- Pager_number

Patients:

Basic patient information:

- Name
- Adress
- Phone_number
- History (A memo field.)

Beds:

Basic information about beds:

- Ward number
- Room number
- Bed number
- Available

Credits

This system and its requirements description were originally developed by Christine Anderson, Norman Chow, Chiasen Chung (Charles), Jonathan Chang, Jordan Goudreau, Don Luu, Liz Wong, Brandon Kurucz, Michael Shareski, Peter Macmurchy, Donald Tetreault and Conway Yu at the University of Calgary, Canada. Our description is based on the original, but modified to fit the course INF3120/INF4120.

Blindern 22.08.2006; Hans Gallis, Magne Jørgensen, Dag Sjøberg and Bente Anda