Hospital Management System

A CS814 Course Project Report

Submitted by

Jahnavi (202CS014)
Prashant Bhilwar (202CS020)

Department of Computer Science and Engineering
National Institute of Technology Karnataka
P.O. Srinivasnagar, Surathkal, Mangalore-575025
Karnataka, India
January 2021

CONTENTS

1 Introduction

Hospital management system is a computer system that helps manage the information related to health care and aids in the job completion of health care providers effectively. They manage the data related to all departments of healthcare such as,

- Clinical
- Inpatient
- Outpatient
- Operation theater
- Nursing
- Pharmaceutical
- Radiology
- Pathology etc.

HMS came into the picture of hospital management as early as 1960 and have ever since been evolving and synchronizing with the technologies while modernizing healthcare facilities. In today's world, the management of healthcare starts from the hands of the patients through their mobile phones and facilitates the needs of the patient.

1.1 Importance of Hospital Management System

Hospital management system was introduced to solve the complications coming from managing all the paper works of every patient associated with the various departments of hospitalization with confidentiality. HMS provides the ability to manage all the paperwork in one place, reducing the work of staff in arranging and analyzing the paperwork of the patients. HMS does many works like:

- Maintain the medical records of the patient.
- Maintain the contact details of the patient.
- Keep track of the appointment dates
- Save the insurance information for later reference
- Tracking the bill payments.

1.2 Advantages of Hospital Management System

- It is time-saving technology as it reduces manual paperwork.
- As all the work is done by the software it Improves efficiency by avoiding human errors.
- Data security will be ensured and correct data retrieval made possible.
- Easy access to patient data with correct patient history.
- Reduces the work of documentation.
- Better Audit controls and policy compliance.

2 System Requirements

2.1 Non Functional Requirements

Security:

- **Patient Identification:** The system needs the patient to recognize herself or himself using the email id.
- **Logon ID:** Any users who make use of the system need to hold a Logon ID and password.
- **Modifications:** Any modifications like insert, delete, update, etc. for the database can be synchronized quickly and executed only by the ward administrator.
- **Administrator rights:** The administrator can view as well as alter any information in the Hospital Management System.

Performance:

- **Response Time:** The system provides acknowledgment in just one second once the 'patient's information is checked.
- **User-Interface:** The user interface acknowledges within five seconds.

Reliability:

- **Availability:** The system is available all the time.

2.2 Functional Requirements

- **User Login:** This feature will be used by user to login into system. Before login they are required to go to their respective login page based on type of user (Patient,Doctor,Admin). They are required to enter username and password for login. The username and password will be verified using the database according to the user type. If found invalid user will not be allowed to enter the system and will be redirected to same login page. According to user type user authorization will be decide which decides what user can have access to.The user must be able to logout after they finished using system.
- **Register:** This feature allows user to register himself as patient only. User should not be allowed to register himself as doctor. For patient login system must verify all the details entered by user are correct or not. If correct system must store the user data into the database.

2.3 Software and Hardware requirements

- **Operating system:** any operating system
- **DATABASE:** MySQL is easy to use. We have to get only the basic knowledge of SQL. We can build and interact with MySQL by using only a few simple SQL statements. It is secure and follows the working of a client/server architecture.
- **Programming Language:** HTML, CSS, Bootstrap, jQuery (Front End) PHP (Back End)

2.4 Hardware Requirements

- Intel core
- RAM 1GB
- Hard Disk: 256MB

Existing Solution

- Still this work is managed manually by most of the hospitals. The employees who have to record the details must perform their job very carefully. Even a small mistake would create a lot of problems. Security of information is very less. Report generations of all the information is very tough task.

- Maintenance of patient records is very complex task. Files are required to be handled very carefully. Any degradation in maintanance may result in the failure of the entire system.

2.4 Software Tools used

The whole project is divided in two parts the front end and the back end.

Front End

- **HTML:** HTML stands for HyperText Markup Language. It is used to design web pages using a markup language. HTML is the combination of Hypertext and Markup language. Hypertext defines the link between the web pages. A markup language is used to define the text document within tag which defines the structure of web pages.HTML is a markup language that is used by the browser to manipulate text, images, and other content to display it in the required format.
- **CSS:** Cascading Style Sheets, fondly referred to as CSS, is a simply designed language intended to simplify the process of making web pages presentable. CSS allows you to apply styles to web pages. More importantly, CSS enables you to do this independent of the HTML that makes up each web page.
- **Bootstrap:** Bootstrap is a free and open-source tool collection for creating responsive websites and web applications. It is the most popular HTML, CSS, and JavaScript framework for developing responsive, mobile-first web sites. Nowadays, the websites are perfect for all the browsers (IE, Firefox and Chrome) and for all sizes of screens (Desktop, Tablets, Phablets, and Phones).
- **jQuery:** jQuery is an open source JavaScript library that simplifies the interactions between an HTML/CSS document, or more precisely the Document Object Model (DOM).

Back End

- **PHP:** he term PHP is an acronym for PHP: Hypertext Preprocessor. PHP is a server-side scripting language designed specifically for web development. PHP can be easily embedded in HTML files and HTML codes can also be written in a PHP file
- **MySQL:** MySQL server is a open-source relational database management system which is a major support for web based applications. Databases and related tables are the main component of many websites and applications as the data is stored and exchanged over the web.

3 Authorization

RBAC: Role-based access control (RBAC) systems assign access and actions according to a person's role within the system. Everyone who holds that role has the same set of rights. Those who hold different roles have different rights.

3.1 Need of RBAC based authorization

Every company has sensitive documents, programs, and records. Protect them too strictly, and company's work grinds to a halt. Leave them open, and catastrophic security issues can arise. Using RBAC we can grant access to those who need it while blocking those who don't need access. Make changes based on a person's role rather than individual attributes. We can make these changes quickly by altering access by role.

3.1.1 What is RBAC

All role-based control systems share core elements, such as:

- **Administrators:** They identify roles, grant permissions, and otherwise maintain security systems.
- **Roles:** Workers are grouped together based on the tasks they perform.
- **Permissions:** Access and actions attach to each role, and they outline what people can and cannot do.

RBAC systems do not require:

- **Differentiation of individual freedoms:** Access is defined by a person's role, not that person's preferences or wishes. This makes it easy to manage permissions.
- **Intensive maintenance:** Permissions follow roles. A new job function becomes a new role applied to dozens (or hundreds or thousands) of employees with only a small amount of work for the administrator. Promotions involve changing roles, not editing permissions as line items.

3.1.2 Roles within RBAC

Roles dictate authorization within an RBAC system. It's critical to define them properly. Otherwise, large groups of people withing company can't do their work.

Roles can be defined by:

- **Authority:** Senior management needs access to flies interns should never see.
- **Responsibility:** A board member and a CEO might hold similar authority within a company, but they are each responsible for different core functions.
- **Competence:** A skilled worker can be trusted to work within sensitive documents without errors, while a novice could make catastrophic mistakes. It's important to tailor access accordingly.

3.1.3 Role-Based Access Control Permissions

Permissions specify what people can access and what they can do in the system. Think of permissions as the rules people follow per the roles have outlined. Permissions should involve:

- **Access:** Who can open a specific drive, program, file, or record? Who shouldn't even know these things exist? Access will limit what people can see.
- **Reading:** Who can scan through these documents, even if they can't change anything inside of them? Some roles may have the ability to reference materials but not make changes to them.
- **Writing:** Who can change documents? Does someone else have to approve the changes, or are they permanent? This will define permissions.
- **Sharing:** Who can download a document or send it as an email attachment? As with the other permissions, some users will not be able to share materials even if they can reference them.
- **Finances:** Who can charge money? Who can offer refunds? Permissions could involve the ability to deal with charges and refunds, set up credit accounts, or cancel payments.

3.1.4 Role-Based Access Control Benefits

Security options abound, and it's not always easy to make the right choice for company. RBAC comes with plenty of tried-and-true benefits that set it apart from the competition. An RBAC system can:

- **Reduce complexity:** New employees gain access based on their roles, not on long lists of server and document requirements. This simplifies creating, maintaining, and auditing policies.
- **Allow global administration:** Change access for many employees all at once by altering permissions associated with a role.
- **Ease on boarding:** As people join, move withing, or are promoted within organizations, and we don't have to worry about the individual's permissions, just that they're in the right place. The roles take care of the rest.
- **Reduce Blunders:** Traditional security administration is error-prone. Adding permissions for individuals gives us plenty of options to make a mistake. Change a role's access, and you're less likely to give someone too much (or too little) power.

- **Lower overall costs:** When admin duties shrink, comanies save on security administration. This saves our organization time and money.

3.2 Components of RBAC in our application

Hospital management system has following components:

Admin

- Can login into the system with admin credentials.
- Can manage patient, View Profile, Delete.
- Can manage Doctor, Add, View Profile, Edit Profile, Delete Profile.
- Can access appointment history.
- Can access contact us queries.
- Can access Doctor session logs.
- Can access user session logs

Patients

- Can Create Account and Login
- Can update profile
- Can book appointment

Doctor

- Can access profile
- Can update profile
- Can view appointments

3.3 Components of administrative model

To ensure better functioning and smooth running of hospital, these are following components in our project:

- Planning: In this project the admin have to create a database such that it will store all the information about patient and doctor and also provide them a unique identification like ID number which allow us to uniquely identify patients and doctors.
- Organization: In the hospital management system the decision is taken as to what kind of authority is to be provided for an area of service.
- Non-Medical Administrative are necessary to the hospital’s buisness and physical plant management
- The CEO leads these administrative services and is directly responsible for the day to day operation of the facility.
- Business services manages the hospital’s admitting and discharge functions, record charges to a patient’s account and handles account receivable with third party payers such as insurance companies.
- The Finance department advices the CEO on financial policy and long range planning, establishes procedures for accounting functions, receives and deposits all monies received by the hospital, and approves the payment of salaries and other expenditures.

Conclusion

Hospital management system is the inevitable part of the lifecycle of the modern medical institution. It automates numerous daily operations and enables smooth interactions of the users. Developing the hospital system software is a great opportunity to create the distinct, efficient and fast delivering healthcare model. Implementation of hospital management system project helps to store all the kinds of records, provide coordination and user communication, implement policies, improve day-to-day operations, arrange the supply chain, manage financial and human resources, and market hospital services. Hospital management system follows RBAC policy for authorization which makes sure that roles as assigned properly with the permission and no permissions should overlap.

Reference

[1] https://www.okta.com/identity-101/what-is-role-based-access-controlrbac/
[2]https://electronichealthreporter.com/importance-of-the-hospital-management-system
[3] https://www.w3schools.com/
[4] https://en.wikipedia.org/wiki/Role-basedaccesscontrol
[5]https://django-role-permissions.readthedocs.io/en/stable/roles.html