

ECG database management system design

Annpurna Matawale
annpurnamatawale
6th semester biomedical

january 2022

ABSTRACT

Databases of electrocardiograms (ECGs) are used in medical research, pharmaceutical research, medical education, and health care. due to increasing needs in research, education, and health care Creating and maintaining ECG databases has been a challenge. It has evolved into a complicated issue.

This paper discusses the a fresh design strategy and a new application model ECG Management System by Philips (EMS). The Philips EMS isn't just for storing and managing data. Not only to collect ECG data, but also to automate the ECG workflow. to make the editing and checking of ECGs easier to acquire and compare the serial ECG recordings save auditing data and generate a serial ECG report changes, and, above all, to communicate with Hospital information systems, for example, are examples of other systems. A **relational database manages** the patient demographic and clinical data, as well as ECGs. The system allows authorised users to access ECG data, ECG recordings, and ECG graphs. In data format, measurements and ECG interpretation eXtensible Markup Language is a markup language that allows you to express yourself in a variety of ways (**XML**), and to collect clinical cases for educational and research purposes queries written in a minimal structured **query language** In conclusion, the Philips EMS is a powerful and versatile system. a simple tool for research, education, and outreach Finally, to improve patient care.

Table of content

1. Introduction
2. Technical methods and materials

3. ECG database management system
4. Database schema
5. Dataformat exchange
6. Workflow
7. System communication
8. Application
9. Conclusion
10. References

INTRODUCTION

ECG databases and **database management systems** are required for medical research, medical education, and patient care . Over the last few decades, many research institutions, healthcare organisations, and manufacturers have devised and constructed database management systems, but only a small percentage of them have been successful . The biggest challenge arises from the rapid advancement of technology. Current "state-of-the-art" technology is frequently obsolete within a few years, if not months. In an era of rapid clinical and information technology advancement, designing such a sophisticated ECG database management system has become increasingly difficult. The development and upkeep of a website consumes a significant amount of resources. Significant efforts have been made to the ECG management system as part of Philips Medical Systems' historical commitment to the medical community. The purpose of this study is to describe Philips ECG Management System's new design techniques, with a focus on its flexibility and openness in data access.

TECHNICAL METHODS AND MATERIALS

Open industry standards for data storage, convenience in user access and data inquiry, connectivity, **security**, configurable workflow, auditing, and tracking are among the novel design concepts used in the current Philips ECG Management System. SCP (standard communication protocol for computer assisted electrocardiography)-compatible files were gathered from the Wei-Gong Memorial Hospital's Emergency Department in Miao-Li County, Taiwan. The data was gathered between August 2003

and July 2006. The Philips/Agilent Page Writer M1770A Cardiograph with Option A05 storage package was used in this study. A RS232 cable was used to send the ECG data to the local host.

ECG DATABASE MANAGEMENT SYSTEM (EMS)

The system was implemented using a **three-tier architectural** approach, which has been an industry standard for decades and has been employed in numerous studies. With a web-based front-end interface, the database was built on **MySQL** and ran on Windows. The ECG data were maintained with a **MySQL database management system (DBMS)** .

The built-in file-directory system in Windows was used to store temporal data and ECG waveform data. For system implementation, open source software was chosen because of its low maintenance costs, reliability, and efficiency. **Apache, PHP, and MySQL** served as the system's backbone, all of which were freeware but performed admirably. The MySQL DBMS and a file-directory system formed the system's foundation (first-tier). To accommodate the MySQL database and PHP scripts, the file-directory structure was changed. The second tier, which was built with Apache and PHP, was built on top of the first. This was the primary component of the system. To make communication between the subsystems easier, Apache was employed as the web server. PHP was the preprocessor for web application logic interpretation.

Commands for each sub-system were interpreted and processed by PHP and then provided to the Apache server for web representation. Finally, web browsers served as the Internet's third tier for end-users (clients).