**Contemporary Problem Analysis**

**CLD 6000**

Professor: Prokopakis Georgios

Student: Brinza Augustin Constantin

GitHub: <https://github.com/augustinnn/Contemporary-Problem-Analysis-CLD-6000.git>

**Table of content**

1. Initial proposal

* -Definition of case problem
* -Thoughts through the process
* -Implementation mode

1. Introduction

-State of Healthcare System

* -Appointing Schedule System
* -Artificial Intelligence in Healthcare System

1. Definition of the case study, analysis, and design

* -Importance of security
* - Analysis and past tests
* - Design and implementation

1. Evaluation

* -Applications roles
* -Solving problems
* -Important requirements

1. Conclusion

* -Outcomes
* -Impact over future generations
* -Final repository

1. References

Initial Proposal

Doctor Appointments: An appointment system for a doctor requires a careful design of the database to be flexible and cover many different needs of the doctor and the patient. Design and implement a database for doctor appointments and provide a simple user interface to test it.

I chose this topic for simplicity but, after I start understanding more and more the multitude of task that are required in order to design and implement flexibility and normality to a database form, I understood the hard work that a database developer have to perform.

In order to achieve to suit our requirement the data base not only have to support large amount of data but should be able to be understood by others so they can perform their duty. To be more exact, from the start of implementation data base should have a good referring name, easy to be used, and intuitive, as same as tables and columns inside. We also should write descriptions with as much details as possible. For name format it should be in small letters, capitals letter only used for differentiating more words.

As for the user interface, it should have multiple futures such as: connecting to data base, showing existing data, modifying existing data and the possibility to add new data.

Introduction

Multiple problems and new implementations have change the Appointing Scheduling System and the difference in terms of usability and performance between our parents generations outdated technology and our current time is huge, as an example we can compare the old animal transportation that people used in the past to automobile transportation that we use today, The appointing system develop from a pen on paper appointment to the technology that almost everyone use it, online appointment. Trough the era, appointing system surpass problems such as overlapping appointments and schedule, over estimation of human capability or just of resources, we adapt to moral some moral rules that are used every were such as queuing rules, patient arrival mode and urgency.

Being a work with human service, healthcare has huge potential advantages from Artificial Intelligence, where AI-based scheduling Machine Learning models can conferee a significant improvement of hospitals healthcare services.

Definition of the case study, analysis, and design

Coming back to our problem: Doctor appointments database and user interface, this system should be design only for staff usage because of the huge amounts of sensitive data from doctors and patients. Database should be used to deliver reliable information, and also to have specific primary keys and foreign keys in order to be able to perform conations between tables and perform specific searches and data display.

The user interface as assignment require should be design simple in order to perform some tasked used to test the sustainability and response.

To obtain all of this some trials and test on have been executed in order to deliver the final project. We encounter problem such as having to implement to many data into database make it not so reliable in terms of modifying data, problems of universal representation of data inside the table.

To reach as many goals as possible, I design database in MySQL using programs such as MAMP and PhpMyAdmin. Due to its high capability of storing huge amounts of data, I manage to be able to create separate tables for our doctors and patients and also have separate data table in order to able to differentiate appointments from available dates and absences.

To use a very wanted coding language, I design the user interface using php, java, html plus css. Here we encounter tasks like connection between html and database that we completed using php, we had to implement functions for different searches and also to insert data written by customer into database trough interface.

Evaluation

By completing this project, we understand that the role of MySQL is to store big amounts of data, and that task like insert, edit and create of database can be performed external using programing language.

Before completing the project, I encounter problems, that at the begin did not look alike such as: poor design and planning of database, ignoring normalization where both sides of application developers and database designers are in need to have a common ground to understand each other and the big problem of low amount of tables holding huge amount of data, where, after understanding the process and tasks that have to be performed we can see how data should be classified.

They are a lot of things that need to be taken under consideration before implementing a healthcare appointment portal to public, like maintainability, encapsulation, security and so on.

Conclusion

From finding a good topic to finishing the project, we performed tasks as database design, interface development, testing and troubleshooting.

To conclude all of the above in term of learning from this project, where we found how databases and user interface can be connected, how data can be manipulated using certain functions and importance of design and norms for database.

As delivering final repository, with this project we achieved almost all of what we targeted.

As impact for future generation, we will sure solve problems by embracing new technologies with eyes wide open.

References

1. Appointment Scheduling Problem in Complexity Systems of the Healthcare Services: A Comprehensive Review. By Ali Ala, Feng Chen, and Academic Editor Saeid Jafarzadeh Ghoushchi. <https://www.hindawi.com/journals/jhe/2022/5819813/>
2. Hospital Management Database. <https://medium.com/@sarahalalawi0/hospital-management-database-3d19dc240446>
3. A Data Model for a Medical Appointment Booking App. By Shantanu Kher, Oracle Database Developer and Design. <https://vertabelo.com/blog/the-doctor-will-see-you-soon-a-data-model-for-a-medical-appointment-booking-app/>
4. Using MySQL with MAMP. <https://courses.cs.washington.edu/courses/cse154/19sp/resources/assets/servers/mamp_mysql.html>
5. Ten Common Database Design Mistakes. By Louis Davidson. <https://www.red-gate.com/simple-talk/databases/sql-server/database-administration-sql-server/ten-common-database-design-mistakes/>
6. Database Design Challenges. <http://www.myreadingroom.co.in/notes-and-studymaterial/65-dbms/487-database-design-challenges.html>
7. A Guide to Addressing Key Database Challenges. By Sameer Paradkar. <https://medium.com/oolooroo/common-database-design-issues-and-their-solutions-7f4c3e9f5f33>
8. PHP Introduction. <https://www.w3schools.com/php/php_intro.asp>
9. PHP vs Java: Which Should You Learn?. By Shauna Blackmon. <https://www.bestcolleges.com/bootcamps/guides/php-vs-java/>
10. Java, PHP and MySQL Bundle, from Beginner to Professional. https://www.w3schools.com/php/php\_mysql\_intro.asp