**AEGCM: Efficient Scheduling for Medical Appointment Management.**

**FRANKLIN PARRALES-BRAVO1, (Senior Member, IEEE), TOMALA GONZALES FEELIX 1, MALAVE BRIONES JOSE 1, (Senior Member, IEEE), ROMERO SERANO JEAN1, FALQUEZ PICO STEVEN1, AND FALCONI VILLAMAR MARCOS 1**

**1Software Engineering Research Group, Faculty of Mathematical and Physical Sciences, University of Guayaquil, Guayaquil 090514, Ecuador.**

**I. ABSTRACT:** This document has been prepared by sixth-semester students of the Software career at the University of Guayaquil, under the guidance of the subject of Software Construction, of the parallel SOFT-VE-6-2.

Today, efficient medical appointment management is essential to providing quality healthcare and improving the patient experience. In this context, we present the technical manual for the implementation of the Medical Appointment Scheduling system. The main objective of this system is to optimize the management of medical appointments, improve the patient experience and facilitate the administration of medical schedules and resources.

**II. INDEX TERMS:** The growing demand for health services has generated a pressing need to improve the efficiency of administrative processes in medical centers. Medical appointment management, in particular, plays a critical role in the organization of patient care. However, traditional systems often have limitations that affect both patients and medical staff.

This article presents a proposal for an integrated system for the optimization of medical appointment management, designed to address the shortcomings of current systems. The proposed system combines information and communication technologies to automate the processes of requesting, confirming and managing appointments, offering an intuitive and easy-to-use interface for both patients and medical staff.

**The main contributions of this work are:**

The manuscript introduces an open and flexible system, available on GitHub, that allows customization and adaptation.

* **Development of a flexible and scalable scheduling system:** A modular and open architecture is presented that allows easy customization and adaptation to different clinical environments, in addition the system has been designed to integrate with various medical record systems.
* **Intuitive and user-friendly user interface:** An intuitive and easy-to-use user interface has been developed for both patients and administrative staff. The interface provides a smooth and efficient user experience.
* **Appointment management optimization:** The system implements optimization to allocate appointments efficiently, taking into account physician availability, patient preferences, and the urgency of consultations. This allows you to reduce waiting times and improve resource utilization.
* **Comprehensive system evaluation:** A rigorous evaluation of the system has been carried out, both from a quantitative and qualitative perspective. The results of the evaluation show that the system is capable of significantly improving the efficiency and quality of medical appointment management.

**III. SYSTEM DESCRIPTION:** The development of this system will be carried out using the Scrum methodology, an agile strategy that is based on team collaboration, incremental delivery and constant feedback from users. The choice of Scrum is based on the following characteristics that align with the objectives of the project:

* **Focus on Collaborative Work:** Scrum promotes a collaborative work environment in which team members meet regularly to plan, execute, and evaluate project progress. This collaboration encourages open communication, joint problem-solving, and shared decision-making.
* **Obtaining Better Results through Partial Deliverables:** Scrum divides project development into short cycles, each with a defined set of objectives and deliverables. This methodology allows tangible results to be obtained on a regular basis, adapting to the changing needs of the project and providing early feedback to users.
* **Prioritization Based on User Value:** System functionalities are prioritized based on their value to end users. This ensures that more resources are dedicated to the features that have the greatest impact on user experience and overall satisfaction.
* **Agile Management:** Scrum is based on the principles of agile management, which emphasize adapting to change and responding quickly to feedback. This allows the project to adjust to the changing needs of the market and users, ensuring its long-term success.

**IV. SYSTEM MODULE**

**System Modules**

The system is made up of two main modules:

* **Management Module**: Is responsible for the administration of human and material resources, including information on patients, doctors, and appointments.
* **Operation Module:** Handles the coordination and follow-up of appointments, from request to appointment completion.

System Advantages

**The Medical Appointment Scheduling system offers several advantages:**

* Optimization of appointment management, allowing efficient scheduling.
* Improvement in the patient experience, providing fast and personalized care.
* Generate reports with key indicators such as the number of scheduled appointments, canceled appointments and patient satisfaction.

This system is presented as an innovative tool that improves the efficiency and effectiveness in the management of medical appointments, which contributes to the growth and competitiveness of medical institutions.

**Description of Databases**

The database of the Medical Appointment Scheduling system has been designed to efficiently support the processes defined in the software. The tables are described below and diagrams are included for a better understanding of the structure and relationships.

# 

# **Test Environment**

The program was tested on Windows 11 in conjunction with, Mstest, Visual Studio and SQL SERVER with its proper connection server

## **Base Hardware Requirements**

The following items are sufficient to execute the unit tests required

|  |  |  |
| --- | --- | --- |
| **Resource** | **Quantity** | **Name and type** |
| Connection Server | 1 | SQL SERVER – database |
| Electronic equipment | 1 | PC o laptop |

## **Base Software Requirements in the Test Environment**

The following table defines the software elements required in the test environment.

|  |  |  |
| --- | --- | --- |
| **Software Element** | **Version** | **Guy** |
| Windows 11 | 11 | S. O. |
| Visual Studio Community 2022 | 11 | IDE |
| SQL Server | 2022 | SSMS |
| Mstest | 8 | Framework |

This article presents a detailed guide on the best practices to optimize and maintain the medical appointment scheduling software, ensuring its correct operation and security.

**1. Operating System Update:** Keeping the operating system updated to the latest version of Windows 10 is crucial to ensure compatibility with scheduling software and ensure optimal performance. Regular updates fix bugs, improve security, and optimize overall system operation.

**2. Driver Update:** Drivers are responsible for communication between the operating system. It is essential to verify that the drivers of the computer where the scheduling software is installed are up to date. Driver updates can improve software performance, stability, and compatibility.

**3. Relevant Software Update:** It is important to keep all software related to the scheduling system up to date, including medical software, patient management software, and any other complementary software. Regular updates fix bugs, improve security, and ensure compatibility with the latest versions of the operating system and scheduling software.

**4. Removal of Unnecessary Software:** An operating system overloaded with unnecessary software can affect the performance of scheduling software. It is recommended that you perform a regular scan to identify and remove programs that are not being used. This will free up system resources and improve the overall performance of the scheduling software.

**5. Maintenance of the Updated Antivirus:** The security of patient data is paramount. It is essential to maintain an up-to-date antivirus to protect the scheduling system against viruses, malware, and other cyber threats. Regular antivirus updates ensure maximum protection against the latest threats.

**6. Review of Planned Events:** It is important to periodically review the events scheduled in the scheduling system's calendar to ensure that they are accurate and do not generate conflicts or errors. This will help prevent scheduling issues and ensure a smooth experience for patients.

**7. System Performance Monitoring:** Implementing system performance monitoring tools allows you to detect potential bottlenecks, performance issues, or failures before they affect users. Regular monitoring of system performance helps to identify and resolve issues proactively, ensuring optimal operation of scheduling software.

**8. Software Configuration Management:** Establishing a configuration management process to control and manage changes to the software system is crucial to maintaining the stability and integrity of the system. It is recommended to use version control tools and keep a detailed record of the modifications made. This makes it easier to identify and resolve issues should they arise.

**9. Testing and Quality Control:** Performing regular tests of the scheduling system is essential to verify its correct operation and detect possible errors or failures. Testing should include integration testing, performance testing, security testing, and functional testing.

# **V. Scrum Methodology**

The development of this system will be carried out using the Scrum methodology, an agile strategy that is based on team collaboration, incremental delivery and constant feedback from users. The choice of Scrum is based on the following characteristics that align with the project's objectives: **Focus on Collaborative Work:** Scrum promotes a collaborative work environment in which team members meet regularly to plan, execute, and evaluate project progress. This collaboration encourages open communication, joint problem-solving, and shared decision-making. **Obtaining Better Results through Partial Deliverables:** Scrum divides project development into short cycles called "sprints", each with a defined set of objectives and deliverables. This methodology allows tangible results to be obtained on a regular basis, adapting to the changing needs of the project and providing early feedback to users. **Prioritization Based on User Value:** System functionalities are prioritized based on their value to end users. This ensures that more resources are dedicated to the features that have the greatest impact on user experience and overall satisfaction. **Agile Management:** Scrum is based on the principles of agile management, which emphasize adapting to change and responding quickly to feedback. This allows the project to adjust to the changing needs of the market and users, ensuring its long-term success.

**VI. SYSTEM DESCRIPTION:**

**System Functionalities:**

* **Patient Registration:** Patients can easily register in the system by providing their basic personal and medical information.
* **Appointment Search and Selection:** Patients can search for available appointments by doctor, specialty, date, and time through an intuitive user interface.
* **Appointment Requests:** Patients can request appointments online with a few clicks, eliminating the need for phone calls.
* **Appointment confirmation and reminders:** The system automatically sends appointment confirmations via email or SMS to patients. In addition, pre-appointment reminders can be set to prevent forgetfulness.
* **Medical agenda management:** Medical staff can view their agenda in real-time, manage appointments requested by patients and schedule new appointments.
* **Appointment cancellation**: Patients can easily cancel appointments through the system, freeing up space in the doctor's schedule for other patients.

**System Benefits:**

* For patients:
  + Reduce wait times to schedule appointments.
  + Increases the convenience and flexibility to request appointments online.
  + Reduce the risk of missing appointments thanks to automatic reminders.

**VII. Analysis of the Diagram:**

The diagram represents a system called "Private Practice". Within this system, several packages or modules can be identified, each with a specific function:

* **Private Practice:** This is the main package that encompasses all the others. It represents the complete system.
* **Data**: This package seems to take care of data management. Contains:
* **DbOperations:** Probably responsible for performing operations directly with the database, such as queries, inserts, updates, and deletions.
* **Interface:** This could be an interface that defines the basic operations that can be performed on the data, serving as a contract for other classes.
* **Model:** This package probably contains the classes that represent the objects in the problem domain, that is, the real-world entities that you want to model in the system (e.g., patients, appointments, and so on).
* **Controller:** This package usually contains the classes that are responsible for coordinating the interaction between the view and the model. It receives user input (via the view) and translates it into operations on the model.
* **View:** This package contains the classes that are responsible for presenting the information to the user, such as graphical interfaces or reports.
* **Utility:** This package could contain classes that offer auxiliary or general utility functionalities for the system, such as validation tools, date and time utilities, etc. Within this package are:
* **Data:** It could contain classes related to data that is not persistent in the database, such as configuration data or temporary data.
* **Forms:** Probably contains classes that represent the forms that the user uses to interact with the system.
* **Resources: It** could contain resources such as images, configuration files, etc.

Diagram

Auto-generated description

**VIII. Component diagram:**

The diagram represents a system called "Private Practice". Within this system, several packages or modules can be identified, each with a specific function:

* **Private Practice:** This is the main package that encompasses all the others. It represents the complete system.
* **Data:** This package takes care of data management:
* **DbOperations:** Probably responsible for performing operations directly with the database, such as queries, inserts, updates, and deletions.
* **Interface:** This could be an interface that defines the basic operations that can be performed on the data, serving as a contract for other classes.
* **Model:** This package probably contains the classes that represent the objects in the problem domain, that is, the real-world entities that you want to model in the system (e.g., patients, appointments, and so on).
* **Controller:** This package usually contains the classes that are responsible for coordinating the interaction between the view and the model. It receives user input (via the view) and translates it into operations on the model.
* **View:** This package contains the classes that are responsible for presenting the information to the user, such as graphical interfaces or reports.
* **Utility:** This package might contain classes that provide auxiliary or generally useful functionalities for the system, such as validation tools. Within this package are:
* **Data:** It could contain classes related to data that is not persistent in the database, such as configuration data or temporary data.
* **Forms:** Probably contains classes that represent the forms that the user uses to interact with the system.
* **Resources: It** could contain resources such as images, configuration files, etc.

Diagram

Auto-generated description