

**DEPARTMENT OF COMPUTER SCIENCES**

**SOFTWARE ENGINEERING CAREER**

**FIRST PARTIAL**

**MEMBERS:** MICHAEL COBACANGO-ADRIAN IZA-ANDRES JACOME-JUNIOR JURADO-JOHNNY LOACHAMIN

**TEMA:** MEDICBYTE

**Edison Lascano, PhD.**

**NRC 6382**



**SANGOLQUÍ - ECUADOR**

1. **Title**

MedicByte

**1.1 Definition and justification of the problem**

In our country, technology is advancing slowly and the bureaucratic systems that have passed are still part of our daily life. This is very evident in the health area, specifically in the request for medical appointments, which in addition to appearing simple and logical, no one implements an online appointment system in public sector hospitals, so to take a medical appointment Long procedures, such as calls to call centers or long lines, still need to be done, unnecessary and age-old difficulties faced by the majority of the population to obtain a busy medical appointment.

**1.2 Introduction**

This document describes the project development plan to be presented by the students of the Software Engineering degree in the determined MedicByte system, its technical specifications and an overview of the proposed development approach. The objective of the software is to make it easier for people to access medical appointments more efficiently, making the software a tool for the user to adapt medical appointments according to their needs. As for users, the system has a simple interface that is user-friendly.

1.3 **Purpose**

The development of this software is to simulate a solution to the problems presented day by day by hundreds of users who want to get a medical appointment, leaving behind the many calls, long lines and having to get up early, saving time on their day.

The developed software belongs to the same developers, and project managers in such a way that there is no express request from an entity for the development of this software.

**1.4 Scope**

The software that we will implement due to its focused characteristics and its relationship with medicine is called MedicByte

-Application for the administrator

* Create appointment
* Add doctor
* show doctor

-Application for the user

* To ask for a date
* Select specialty

**1.5 Personnel involved**

|  |  |
| --- | --- |
| Name | Andres Jacome |
| Role | Project manager |
| Professional category | Software engineer student |
| Responsibilities | the majority of the project |
| Contact information | ksprwhite@gmail.com |
| Approval | Yes |

|  |  |
| --- | --- |
| Name | Junior jurado |
| Role | Inspector |
| Professional category | Software engineer student |
| Responsibilities | Identify errors in the code |
| Contact information | stalinjuniorbrayan@gmail.com |
| Approval | Yes |

|  |  |
| --- | --- |
| Name | Michael Cobacango |
| Role | Inspector |
| Professional category | Software engineer student |
| Responsibilities | Identify errors in the code and logic of the program |
| Contact information | michaelpm63@gmail.com |
| Approval | Yes |

|  |  |
| --- | --- |
| Name | Adrian Iza |
| Role | notary |
| Professional category | Software engineer student |
| Responsibilities | Keep a record of errors presented in the program |
| Contact information | Bryanadrian38@hotmail.com |
| Approval | Yes |

|  |  |
| --- | --- |
| Name | Johnny Loachamin |
| Role | notary |
| Professional category | Software engineer student |
| Responsibilities | Analyze present errors |
| Contact information | jsloachamin1@espe.edu.ec |
| Aprobación | Yes |

**1.6. Summary**

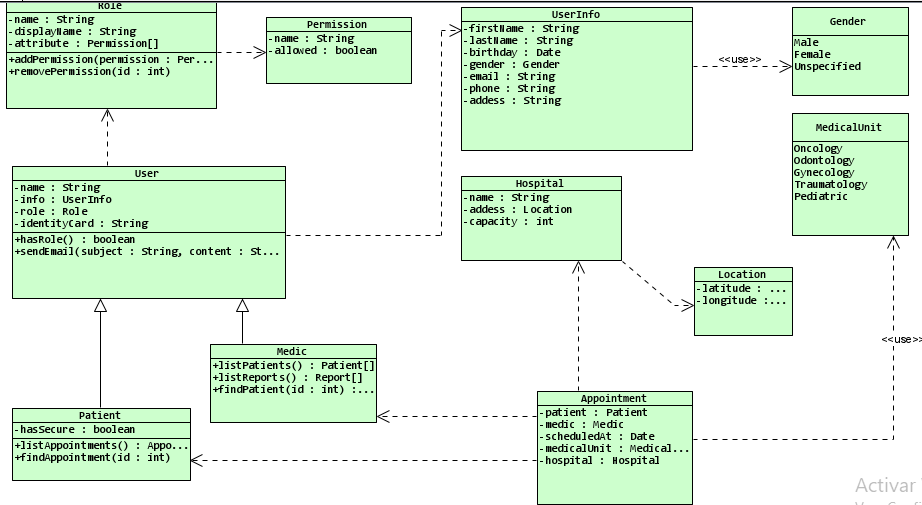
This document consists of three sections. The first section introduces it and provides an overview of the specification of functional and non-functional requirements of the software product, its objectives, as well as the scope, as well as a description of the personnel involved in the development of the system. In the second section of the document, a general description of the system is made, in order to know the main functions that it must perform, the associated data and the factors, restrictions, assumptions, main characteristics and dependencies that affect development, without going into excessive details. The third section of the document is the one that defines in detail the functional and non-functional requirements that the system must meet.

**2. General Description**

**2.1. Product perspective**

"MedicByte" software It is a software product that allows access to users and administrators, to schedule appointments, in the user interface we can appreciate its functions such as creating an appointment, selecting the specialty, as well as entering personal data and Preference date / time data for shift allocation

**2.2. Funcionalidad del producto**



**2.3. User characteristics**

|  |  |
| --- | --- |
| Type of user | User |
| Formation | Basic Education |
| Activities | Register in the system with your personal data, Request a shift with time and date of preference |

|  |  |
| --- | --- |
| Typo of user | Administrator |
| Formacion | Basic Education |
| Activities | Create an appointment, add doctors, show doctors |

**3. Specific requirements**

**3.1. Functional Requirements**

**Functional Requirement 1 (RF1):** Administrator logging. The program should allow administrator access.

**Functional Requirement 2 (RF2):** User interaction with the program. The program it should display a menu with options and allow the user to choose one.

**Functional Requirement 3 (RF3):** Option exit. The program must allow the user and administrator exit the program.

**Functional Requirement 4 (RF4):** Menu validation. The menu start will count 3 options and can only be entered numbers from 1 to 3.

**Functional Requirement 5 (RF5**): to ask for an appointment. The program should allow the user the option to request a medical appointment.

**Functional Requirement 6 (RF6):** Choose one of the 3 specialties. The program should allow the user the option to choose any of the 3 specialties available.

**Functional Requirement 7 (RF7):** Schedule Appointment. The program must generate a medical appointment.

**Functional Requirement 8 (RF8):** Cancel a medical appointment. The program must cancel a previously scheduled medical appointment.

**Functional Requirement 9 (RF9):** Show medical appointments. The administrator will be able to choose the option to show the medical appointments that have previously been scheduled with their respective specifications.

**Functional Requirement 10 (RF10):** Add doctor. Administrators should be allowed to add different new doctors with their respective personal information and professional.

**Functional Requirement 11 (RF11):** Show doctors. The administrator will be able to display the list of doctors that the hospital has divided into their respective specialties.

**Functional Requirement 12 (RF12): Consult** appointment. The user can consult the information of a previously scheduled appointment.

The development of this software is to simulate a solution to the problems presented day by day by hundreds of users who want to get a medical appointment, leaving behind the many calls, long lines and having to get up early, saving time on their day.

**3.2. Non-Functional Requirements**

Non-Functional Requirements (N-FR1): The system must have properly structured user manuals.

Non-Functional Requirements (N-FR2): The system must be able to operate properly with normal number of users with concurrent sessions

Non-Functional Requirements (N-FR3): The access permissions to the system can only be changed by the data access administrator.

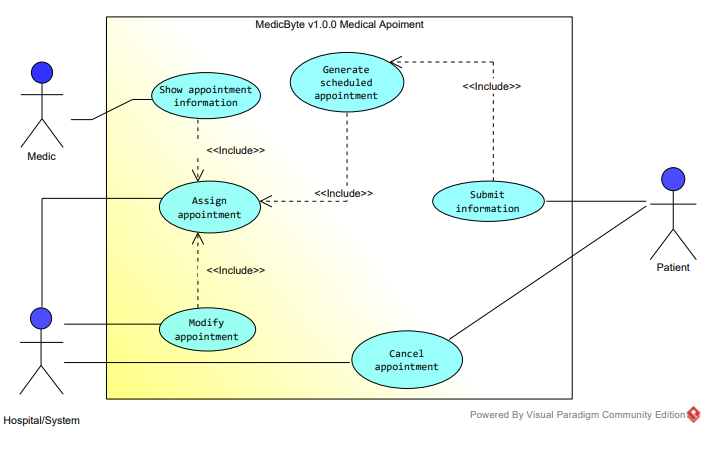
Non-Functional Requirements (N-FR4): The system will not reveal to its operator’s other personal data of users other than names and reference numbers.

Non-Functional Requirements (N-FR5): Software testing will be managed with a software testing management tool.

Non-Functional Requirements (N-FR6): Medical data systems: The new system and its data maintenance procedures must comply with medical data protection laws and regulations. Non-Functional Requirements (N-FR7): A user friendly design

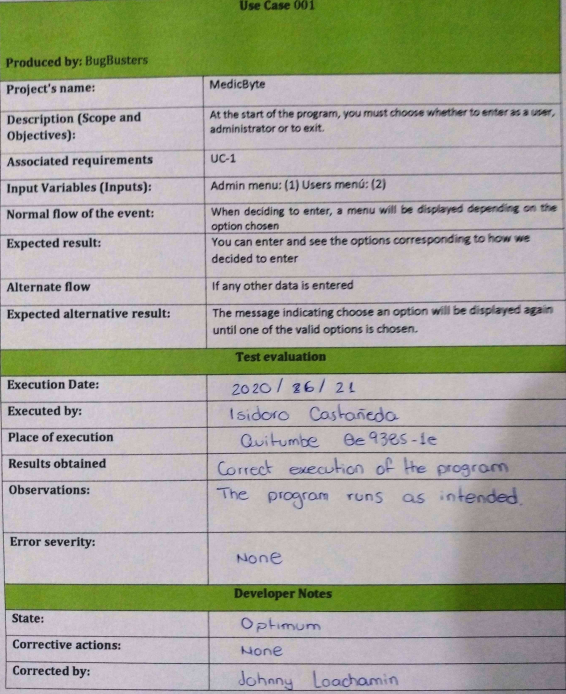
Non-Functional Requirements (N-FR8): The software must be portable. Therefore, moving from one operating system to another does not create a problem

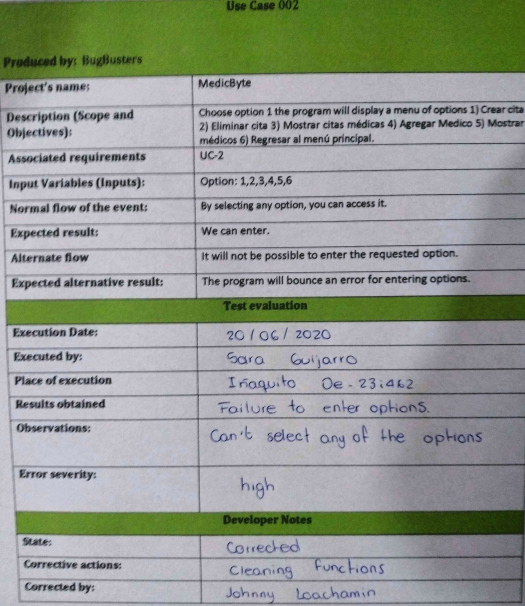
**4. Use Cases Diagram**

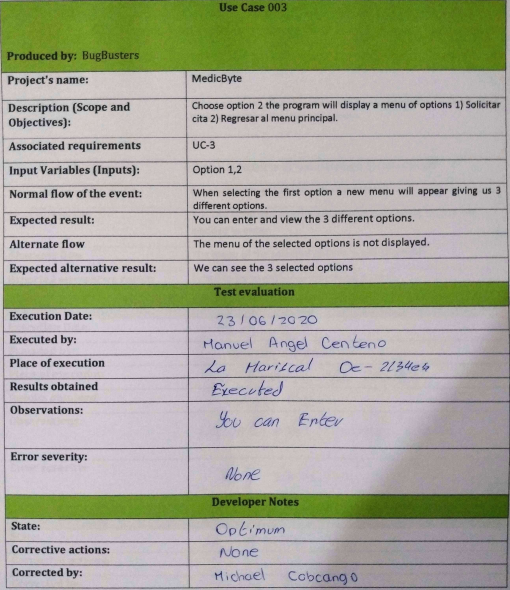


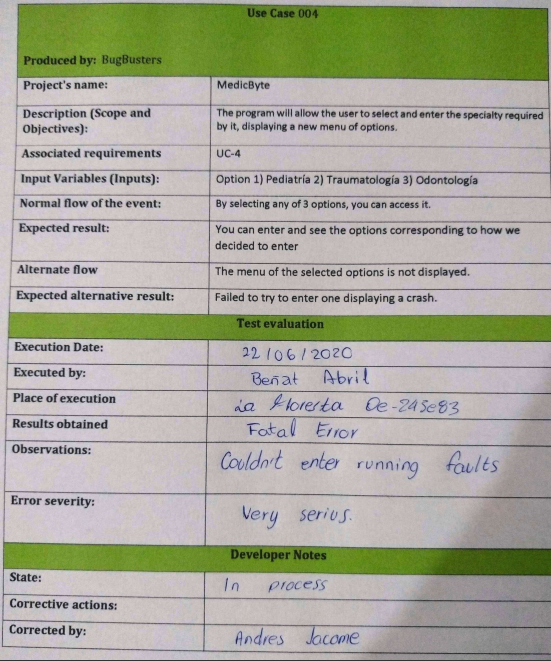
**5. Specification Use cases**

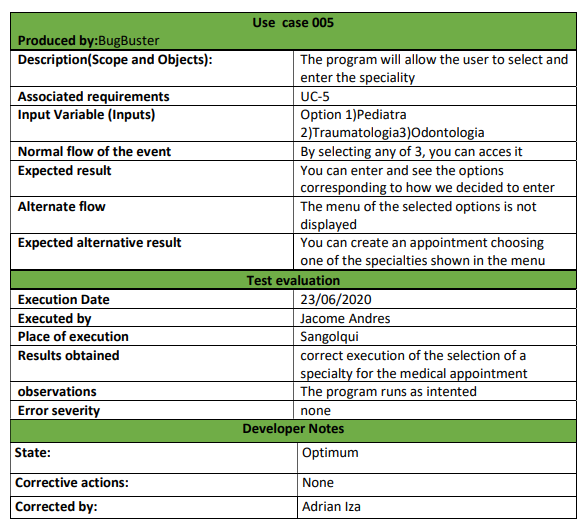
**5.1 Shift registration application**





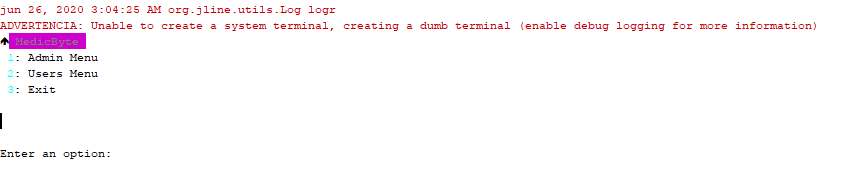






**6. User interface**

Main menu



Admin menu

