**Introduction.**

Over the past few days, we have been developing a software application for a clinic, this program allows a user to register and manage information of appointments while keeping the important information organized and free of redundancy, this is accomplished by using a database that register the information of the user and the appointments created.

**General description.**

This software is developed so that any user can create their own account to access the options depending on the type of account, based on this, the program presents the user with a username and password validation in order to access the system, if the user uses an administrative account, the program will lead the user to the administrative menu, while if the user logs in with a normal account, he will be directed to the menu developed for the patients.

As mentioned before, we decided to split the program into two main parts, the first main part is the authorized user (patient), this type of user is going to be able to access all the options for a patient, such as creating a new appointment, deleting one, editing the information of an appointment, checking the active appointments and getting information of their account, while on the other hand is the administrative user, this type of user is going to have access to exclusive options for employes that allows them to manage the information of the clinic, the options defined for this type of user are the following: check all the users registered, delete account of unactive users, delete appointment from any specific user, create an account for a user, check the active appointments of an user.

**Functional Requirements.**

As functional requirements, there was some functions that needed to be created to accomplish the type of system asked, these options were:

* Creating a new appointment.
* Managing the information of the patients.
* Managing the information of the user’s appointments.
* Automatically delete appointments that have already been taken care of.
* Canceling appointments.

**Non-Functional Requirements.**

For this program, there was only presented one non-functional requirement, and that was a design base on Microsoft (Microsoft-like design), this type of design allows the user to get familiar with the program in a more natural way, making it easier for anyone to use the program without many issues.

**Use Case**

|  |  |  |  |
| --- | --- | --- | --- |
| Use Case ID: | 001 | | |
| Use Case Name: | Log in | | |
| Created By: | J. Bermúdez | Last Updated By: | J. Bermúdez |
| Date Created: | 07/03/2025 | Date Last Updated: | 08/03/2025 |

|  |  |
| --- | --- |
| Actors: | Usuario no autorizado |
| Description: | Solicitud de uso de recursos computacionales |
| Trigger: | Ingresar a la aplicacion |
| Preconditions: | 1. Base de datos de usuarios activo |
| Postconditions: | 1. Autorización o denegación del servicio |
| Normal Flow: | 1. Sistema presenta la Ventana de Login 2. Usuario no autorizado ingresa su correo 3. Sistema valida la información, si existe una cuenta con la base de datos de usuarios. 4. Usuario no autorizado oprimen Next 5. Usuario no autorizado incluye contraseña 6. El sistema valida la contraseña. 7. Sistema de paso al desktop de la aplicación. |
| Alternative Flows: | 1. Usuario no autorizado cierra la ventana de login 2. Usuario falla el usuario o contraseña 3. Usuario no ingresa nada |
| Exceptions: | 1. El sistema muestra un mensaje de denegación en la ventana de login. 2. Usuario no autorizado presiona el boton de Next 3. Usuario con cuenta falla su correo o contraseña |
| Includes: | Valida la información |
| Priority: | 1 |
| Frequency of Use: | Diario |
| Business Rules: |  |
| Special Requirements: | Microsoft like |
| Assumptions: | Uso de nombre y contraseña |
| Notes and Issues: | NA |

**Use Case**

|  |  |  |  |
| --- | --- | --- | --- |
| Use Case ID: | 002 | | |
| Use Case Name: | Main menu | | |
| Created By: | M. Gaviria  A. Chavarría | Last Updated By: | M. Gaviria  A. Chavarría |
| Date Created: | 07/03/2025 | Date Last Updated: | 08/03/2025 |

|  |  |
| --- | --- |
| Actors: | Usuario autorizado |
| Description: | Solicitud de uso de recursos computacionales |
| Trigger: | Usuario autorizado por el Login |
| Preconditions: | 1. Acceso válido al sistema |
| Postconditions: | 1. Acceso a funcionalidades principales |
| Normal Flow: | 1. Usuario accede mediante la Ventana de Login 2. Sistema presenta la Ventana de Menú Principal 3. Usuario escoge una opción del menú |
| Alternative Flows: | 1. Usuario no autorizado cierra la ventana del menú principal 2. Usuario accede al sistema y cierra sesión 3. Usuario accede a alguna funcionalidad.   3.1. Completa parcialmente los datos y cierra el programa |
| Exceptions: | 1. Usuario trata de acceder a opciones sin la autorización. |
| Includes: | Valida la información |
| Priority: | 2 |
| Frequency of Use: | Diario |
| Business Rules: | Usuario solo puede tener tres citas activas |
| Special Requirements: | Microsoft like |
| Assumptions: | Usuario con acceso |
| Notes and Issues: | NA |

**Use Case**

|  |  |  |  |
| --- | --- | --- | --- |
| Use Case ID: | 003 | | |
| Use Case Name: | New appointment | | |
| Created By: | A. Chavarría | Last Updated By: | 1. Chavarría |
| Date Created: | 10/03/2025 | Date Last Updated: | 10/03/2025 |

|  |  |
| --- | --- |
| Actors: | Usuario autorizado  Base de datos |
| Description: | El usuario accederá a esta funcionalidad para generar una nueva cita |
| Trigger: | El usuario accede a la opción establecida para esta funcionalidad |
| Preconditions: | 1. Acceso válido al sistema |
| Postconditions: | 1. Creación de una cita |
| Normal Flow: | 1. Usuario ingresa a la opción  2. Usuario selecciona una fecha para nueva cita  3. Usuario selecciona una de las horas establecidas para la nueva cita  4. El usuario confirma la cita establecida |
| Alternative Flows: | 1. Usuario accede a la opción y trata de agendar cita sin ingresar un día 2. Usuario selecciona un día y se sale del prrograma 3. Usuario completa todos los datos requeridos y finaliza el programa |
| Exceptions: | 1. Usuario trata de crear una cita cuando ya existen tres citas activas |
| Includes: | Validar la información  Registrar en base de datos |
| Priority: | 3 |
| Frequency of Use: | diario |
| Business Rules: | Un usuario solo puede tener tres citas activas en total, si se desea crear una cita más, el usuario debe ser detenido |
| Special Requirements: | None |
| Assumptions: | Usuario registrado en base de datos |
| Notes and Issues: | None |

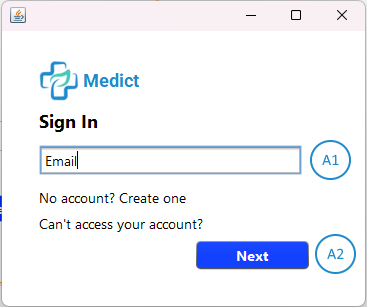


Figure 1.1.1: Email validation  
Screen that validates the email given

1.1.1.1

|  |  |
| --- | --- |
| Item | Email textbox |
| Purpose | User writes their email to be validated |
| Input | Email written by the keyboard |
| Output | None |
| Validity | Select the textbox before writting |
| Error Handling | There should be none |
| Relationships | None |
| Reference | Figure 1.1.1, item A1 |

1.1.1.2

|  |  |
| --- | --- |
| Item | Next button |
| Purpose | Calls the database to validate the email |
| Input | Mouse right-click |
| Output | The window changes and the user get asked for the password |
| Validity | Datebase has the information of the user |
| Error Handling | If the email given by the user isn’t saved in the database, an error window pops out and let the user know |
| Relationships | None |
| Reference | Figure 1.1.1, item A2 |

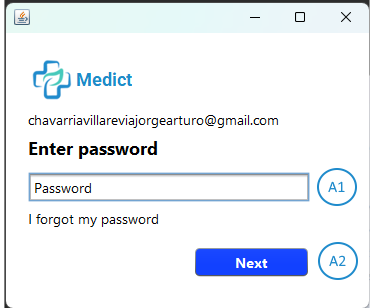


Figure 2.1.1: Password validation  
Screen that validates if the password matches the email given

2.1.1.1

|  |  |
| --- | --- |
| Item | Password textbox |
| Purpose | User writes their password to be validated and matched |
| Input | Password written by the keyboard |
| Output | None |
| Validity | Select the textbox before writting |
| Error Handling | There should be none |
| Relationships | None |
| Reference | Figure 2.1.1, item A1 |

2.1.1.2

|  |  |
| --- | --- |
| Item | Next button |
| Purpose | Calls the database to compare the password with the email given before |
| Input | Mouse right-click |
| Output | The window changes and the user enters the system |
| Validity | Datebase has the information of the user |
| Error Handling | If the password given by the user doesn’t match the password saved in the database, an error window pops out and let the user know |
| Relationships | None |
| Reference | Figure 2.1.1, item A2 |

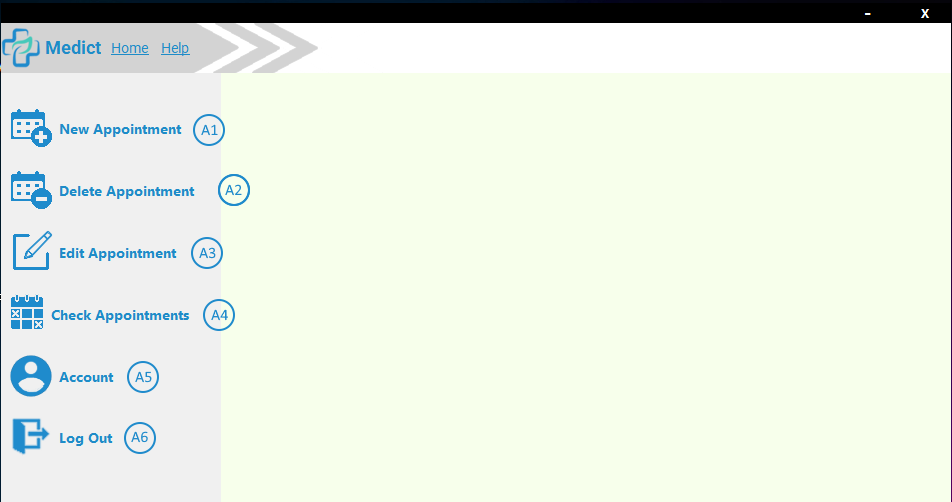


Figure 3.1.1: Main menu  
The main menu screen

3.1.1.1

|  |  |
| --- | --- |
| Item | New appointment |
| Purpose | Allow the user to see the options for a new appointment |
| Input | Mouse right-click |
| Output | The panel changes, allowing the user to set the data for a new appointment |
| Validity | Click within the button |
| Error Handling | User tries to create an appointment when he has already reached the maximum number of appointments |
| Relationships | None |
| Reference | Figure 3.1.1, item A1 |

3.1.1.2

|  |  |
| --- | --- |
| Item | Delete appointment |
| Purpose | Allow the user the ability to delete an existing appointment |
| Input | Mouse right-click |
| Output | The panel changes, allowing the user to delete an existing appointment |
| Validity | Click within the button |
| Error Handling | There should be none |
| Relationships | None |
| Reference | Figure 3.1.1, item A2 |

3.1.1.3

|  |  |
| --- | --- |
| Item | Edit appointment |
| Purpose | Allow the user to search for an appointment, and modify the information |
| Input | Mouse right-click |
| Output | The panel changes, allowing the user to modify the data for an appointment |
| Validity | Click within the button |
| Error Handling | There should be none |
| Relationships | None |
| Reference | Figure 3.1.1, item A3 |

3.1.1.4

|  |  |
| --- | --- |
| Item | Check appointments |
| Purpose | Allow the user to see the appointments created |
| Input | Mouse right-click |
| Output | The panel changes, allowing the user to see the information of the existing appointments |
| Validity | Click within the button |
| Error Handling | There should be none |
| Relationships | None |
| Reference | Figure 3.1.1, item A4 |

3.1.1.5

|  |  |
| --- | --- |
| Item | Account |
| Purpose | Allow the user to access their information |
| Input | Mouse right-click |
| Output | The panel changes, allowing the user to see the information of his/her account |
| Validity | Click within the button |
| Error Handling | There should be none |
| Relationships | None |
| Reference | Figure 3.1.1, item A5 |

3.1.1.6

|  |  |
| --- | --- |
| Item | Log Out |
| Purpose | User can log out from their account |
| Input | Mouse right-click |
| Output | The panel changes, allowing the user to Log In again or Log In with a different account |
| Validity | Click within the button |
| Error Handling | There should be none |
| Relationships | None |
| Reference | Figure 3.1.1, item A6 |