

Use Case Descriptions

Use Case: Register

Goal:

- Allow new users to create an account and register in the system.

Primary Actor(s):

-Patient

-Doctor

Preconditions:

- The user must not have an existing account.
- The system is working.

Success end:

- The user account is successfully created.

Failure end:

- The account registration fails .

Trigger:

- When the user clicks the "Register" button.

Main Success Scenario:

- 1.The user provides the necessary details (name, email, password,...).

2.The system verifies the information provided.

3.The system confirms account creation and sends a confirmation email.

Extensions (error scenarios):

1a. Incomplete information:

- The system prompts the user to fill in the missing details before submitting.

1b. Invalid email address:

- The system informs the user that the email provided is not valid and requests a new one.

Use Case: Login

Goal:

- Allow users to log into the system using their login details.

Primary Actor(s):

- Patient
- Doctor
- Admin

Preconditions:

- The user must have an active account in the system.
- The system is connected and working.

Success end:

- The user is logged into the system.

Failure end:

- Invalid login details.

Trigger:

- When the user clicks the "Login" button.

Main Success Scenario:

1. The user enters their username and password.
2. The system verifies the details.
3. If valid, the user is logged in, and the homepage is displayed.

Extensions (error scenarios):

1a. Incorrect username or password:

- The system informs the user that the username or password is incorrect and try again.

1b. Account locked:

- If the user enters incorrect details multiple times, the system locks the account and tell the user to reset the password.
-

Use Case: Search Doctor

Goal:

- Search for doctors based on specific criteria (specialty, location).

Primary Actor(s):

- Patient

Preconditions:

- The patient logged into the system.
- The system is working.

Success end:

- Doctors matching the specified criteria are found.

Failure end:

- No doctors match the specified criteria.

Trigger:

- When the patient clicks the "Search Doctor" button.

Main Success Scenario:

1. The patient selects search criteria .
2. The system displays a list of matching doctors.
3. The patient reviews the available doctor profiles.

Extensions (error scenarios):

1a. No doctors found matching the search criteria:

- The system informs the patient that no doctors match the criteria .
-

Use Case: View Doctor Profile

Goal:

- Allow patients to view doctor profiles, including qualifications, specialties, and availability.

Primary Actor(s):

- Patient

Preconditions:

- The patient is logged into the system.
- The system is working.

Success end:

- The doctor profiles are successfully displayed.

Failure end:

- No doctor profiles are found.

Trigger:

- When the patient clicks on a doctor's name to view their profile.

Main Success Scenario:

- 1.The patient selects a doctor from the search results.
- 2.The system displays the doctor's profile, including qualifications and availability.

Extensions (error scenarios):

1a. No doctor profiles found:

- The system informs the user that no profiles are available for the selected doctor.

Use Case: Book Appointment

Goal:

- Book an appointment with a doctor.

Primary Actor(s):

- Patient

Preconditions:

- The patient is logged into the system.
- The system is working.

Success end:

- The appointment is successfully booked.

Failure end:

- The appointment is unavailable.

Trigger:

- When the patient clicks the "Book Appointment" button.

Main Success Scenario:

1. The patient selects a doctor and reviews available appointment slots.
2. The patient selects a specific time.
3. The system confirms the appointment booking.

Extensions (error scenarios):

1a. Appointment booking failure:

- The booking fails due to a system error, and the user is informed to try again later.
-

Use Case: View Appointment History

Goal:

- Allow patients to view their past appointment history.

Primary Actor(s):

- Patient

Preconditions:

- The patient is logged into the system.
- The system is working.

Success end:

- The patient is able to view a detailed history of past appointments.

Failure end:

- No appointment history is found.

Trigger:

- When the patient clicks on "View Appointment History".

Main Success Scenario:

1. The patient selects the option to view past appointments.
2. The system displays the patient's appointment history.

Extensions (error scenarios):

1a. No appointment history found:

- The system informs the user that no past appointments are available.

Use Case : Payment with Visa

Goal:

- Enable patients to pay for services using Visa online.

Primary Actor(s):

- Patient

Preconditions:

- The patient is logged into the system.
- The system is working.
- The patient has a valid Visa card for online payment.

Success End:

- The payment is processed successfully using Visa online.

Failure End:

- The payment cannot be processed due to card issues or system errors.

Trigger:

- When the patient selects the "Pay with Visa" option on the payment page.

Main Success Scenario:

1. The patient selects the "Pay with Visa" option.
2. The system tell the patient to enter Visa card details .
3. The patient enters the required information and confirms the payment.
4. The system processes the payment and confirms the success to the patient.

Extensions (Error Scenarios):

1a. Payment failure with Visa:

- The system fails to process the payment due to insufficient funds or card issues, and the patient is prompted to try another method or card.

1b. Validation failure for Visa:

- The system detects invalid Visa card details, and prompts the patient to re-enter the correct information.

Use Case : Payment with InstaPay

Goal:

- Enable patients to pay for services using InstaPay online.

Primary Actor(s):

- Patient

Preconditions:

- The patient is logged into the system.
- The system is working.
- The patient has an active InstaPay account for online payment.

Success End:

- The payment is processed successfully using InstaPay .

Failure End:

- The payment cannot be processed due to account issues or system errors.

Trigger:

- When the patient selects the "Pay with InstaPay" option on the payment page.

Main Success Scenario:

1. The patient selects the "Pay with InstaPay" option.
2. The system tell the patient to enter account information or the phone number linked to the InstaPay account.
3. The patient enters the required details and confirms the payment.
4. The system processes the payment and confirms the success to the patient.

Extensions (Error Scenarios):

1a. Validation failure for InstaPay:

- The system detects invalid InstaPay account information, and prompts the patient to re-enter the correct details..

Use Case: Payment with Cash

Goal:

- Enable patients to pay for services using cash.

Primary Actor(s):

- Patient

Preconditions:

- The patient is logged into the system.
- The system is working.
- The patient is physically present to make a cash payment.

Success end:

- The cash payment is processed successfully.

Failure end:

- The cash payment cannot be processed due to system errors or issues with the cash register.

Trigger:

- When the patient selects "Pay with Cash" on the payment page.

Main Success Scenario:

1. The patient selects the "Pay with Cash" option.

2. The patient will pay when he arrive to the clinic

Extensions (error scenarios):

1a. The patient didn't pay when he arrive

- The system will send it to the admin, who will cancel the booking and block the patient
-

Use Case: Edit Patient Profile

Goal:

- Allow patients to edit their profile information in the system.

Primary Actor(s):

- Patient

Preconditions:

- The patient is logged into the system.
- The system is working.

Success End:

- The profile is successfully updated, and the information is saved in the system.

Failure End:

- The profile cannot be updated due to system errors or invalid data entry.

Trigger:

- When the patient selects the "Edit Profile" option on the profile page.

Main Success Scenario:

1. The patient clicks the "Edit Profile" button.

2. The system displays a form with the current profile details
3. The patient edits the required fields.
4. The patient submits the changes.
5. The system validates the entered data and updates the profile.
6. The system confirms the update and shows a success message to the patient.

Extensions (Error Scenarios):

1a. Invalid data :

- The system detects invalid input and tell the patient to correct the information before resubmitting.
-

Use Case: Change Appointment

Goal:

- Allow the patient to change the date or time of an existing appointment.

Primary Actor(s):

- Patient

Preconditions:

- The patient is logged into the system.
- The system is working.
- The appointment is available and can be changed.

Success end:

- The appointment is successfully changed to the new date/time.

Failure end:

- The appointment cannot be changed due to scheduling conflicts or system issues.

Trigger:

- When the patient clicks "Change Appointment" from the appointment management options.

Main Success Scenario:

1. The patient selects an existing appointment to change.
2. The system displays available dates and times for rescheduling.
3. The patient selects a new date and time.
4. The system checks for availability and confirms the changes.
5. The appointment is updated with the new date/time, and both patient and service provider are notified.

Extensions (error scenarios):

1a. Unavailable time slot:

- The system notifies the patient that the selected time is unavailable and prompts them to choose another.

1b. Restricted change period:

- The system prevents changes if the appointment is too close and displays an error message.

1c. System failure:

1c.1 If there's a system error, the patient is notified and asked to try again later.

Use Case: Cancel Appointment

Goal:

- Allow the patient to cancel an existing appointment.

Primary Actor(s):

- Patient

Preconditions:

- The patient is logged into the system.
- The system is working.
- The appointment exists and is available for canceling.

Success end:

- The appointment is successfully canceled, and both the patient and doctor are notified.

Failure end:

- The appointment cannot be canceled due to system errors .

Trigger:

- When the patient clicks "Cancel Appointment" from the appointment management options.

Main Success Scenario:

1. The patient selects an existing appointment to cancel.
2. The system asks the patient to confirm the cancellation.
3. The patient confirms the cancellation.
4. The system processes the cancellation and notifies both the patient and doctor.

Extensions (error scenarios):

1a. Restricted cancellation period:

- The system notifies the patient that the appointment cannot be canceled due to proximity to the appointment time and displays an error message.

1b. Cancellation fee:

- If applicable, the system informs the patient of a late cancellation fee, requiring confirmation before proceeding.

Use Case: Edit Doctor Profile

Goal:

- Allow doctors to manage their profiles (update information, availability).

Primary Actor(s):

- Doctor

Preconditions:

- The doctor is logged into the system.
- The system is working.

Success end:

- The doctor's profile is updated successfully.

Failure end:

- The profile update fails.

Trigger:

- When the doctor clicks "Manage Doctor Profile".

Main Success Scenario:

1. The doctor selects the option to update their profile.
2. The system tell for profile updates and availability changes.
3. The doctor submits the updates, and the system confirms the changes.

Extensions (error scenarios):

1a. Profile update failure:

- The system fails to save the changes to the doctor's profile, and the doctor is prompted to retry.

Use Case: Delete Doctor

Goal:

- Allow the admin to delete a doctor's profile from the system.

Primary Actor(s):

- Admin

Preconditions:

- The admin is logged into the system.
- The system is working.
- The doctor's profile exists in the system.

Success End:

- The doctor's profile is successfully deleted from the system.

Failure End:

- The doctor's profile cannot be deleted due to a system error.

Trigger:

- When the admin selects the "Delete Doctor" option from the doctor management page.

Main Success Scenario:

1. The admin navigates to the doctor management section.
2. The admin selects the doctor they want to delete from the list.
3. The system prompts the admin to confirm the deletion.
4. The admin confirms the action.
5. The system deletes the doctor's profile from the database.
6. The system confirms the deletion and notifies the admin.

Extensions (Error Scenarios):

1a. System failure:

- The system fails to delete the doctor's profile and informs the admin to try again later.

Use Case: Delete Patient

Goal:

- Allow the admin to delete a patient's profile from the system.

Primary Actor(s):

-Admin

Preconditions:

- The admin is logged into the system.
- The system is working.
- The patient's profile exists in the system.

Success End:

- The patient's profile is successfully deleted from the system.

Failure End:

- The patient's profile cannot be deleted due to a system error.

Trigger:

- When the admin selects the "Delete Patient" option from the patient management page.

Main Success Scenario:

1. The admin navigates to the patient management section.
2. The admin selects the patient they want to delete from the list.
3. The system prompts the admin to confirm the deletion.
4. The admin confirms the action.
5. The system deletes the patient's profile from the database.
6. The system confirms the deletion and notifies the admin.

Extensions (Error Scenarios):

1a. System failure:

- The system fails to delete the patient's profile and informs the admin to try again later.

Use Case: Add Doctor

Goal:

- Allow the admin to add a new doctor's profile to the system.

Primary Actor(s):

-Admin

Preconditions:

- The admin is logged into the system.

-The system is working.

Success End:

-The doctor's profile is successfully added to the system.

Failure End:

-The doctor's profile cannot be added due to a system error.

Trigger:

-When the admin selects the "Add Doctor" option from the doctor management page.

Main Success Scenario:

1. The admin navigates to the doctor management section.
2. The admin selects the "Add Doctor" option.
3. The system prompts the admin to enter the doctor's details (name, specialty, contact information, ...).
4. The admin enters the required details and submits the information.
5. The system validates the entered information.
6. The system adds the doctor's profile to the database.
7. The system confirms the addition and notifies the admin.

Extensions (Error Scenarios):

1a. System failure:

-The system fails to add the doctor's profile and informs the admin to try again later.

Use Case : View List of Users

Goal:

- Allow the admin to view a list of all registered users in the system.

Primary Actor(s):

- Admin

Preconditions:

- The admin is logged into the system.
- The system is working.
- Users are registered in the system.

Success End:

- The admin successfully views a list of all users.

Failure End:

- The list of users cannot be retrieved due to a system error.

Trigger:

- When the admin selects the "View Users" option from the management dashboard.

Main Success Scenario:

1. The admin navigates to the user management section.
2. The admin selects the option to view all users.
3. The system retrieves the list of users from the database.
4. The system displays the list of users to the admin.

Extensions (Error Scenarios):

1a. System failure:

- The system fails to retrieve the list of users and informs the admin to try again later.
-

Use Case : View List of Doctors

Goal:

- Allow the admin to view a list of all registered doctors in the system.

Primary Actor(s):

- Admin

Preconditions:

- The admin is logged into the system.
- The system is working.
- Doctors are registered in the system.

Success End:

- The admin successfully views a list of all doctors.

Failure End:

- The list of doctors cannot be retrieved due to a system error.

Trigger:

- When the admin selects the "View Doctors" option from the management dashboard.

Main Success Scenario:

1. The admin navigates to the doctor management section.
2. The admin selects the option to view all doctors.
3. The system retrieves the list of doctors from the database.
4. The system displays the list of doctors to the admin.

Extensions (Error Scenarios):

1a. System failure:

- The system fails to retrieve the list of doctors and informs the admin to try again later.
-

Use Case : Appointment Completed

Goal:

- Allow doctors to mark an appointment as completed.

Primary Actor(s):

- Doctor

Preconditions:

- The doctor is logged into the system.
- The system is working.
- The appointment is scheduled and not yet marked as completed.

Success End:

- The appointment is successfully marked as completed.

Failure End:

- The appointment status cannot be updated due to a system error.

Trigger:

- When the doctor selects the "Mark as Completed" option for an appointment.

Main Success Scenario:

1. The doctor views the list of scheduled appointments.
2. The doctor selects the option to mark an appointment as completed.
3. The system updates the appointment status to "Completed."
4. The system confirms the update and notifies the doctor.

Extensions (Error Scenarios):

1a. System failure:

- The system fails to update the appointment status and informs the doctor to try again later.
-

Use Case : Cancel Appointment

Goal:

- Allow doctors to cancel a scheduled appointment.

Primary Actor(s):

- Doctor

Preconditions:

- The doctor is logged into the system.
- The system is working.
- The appointment is scheduled and has not yet been completed or canceled.

Success End:

- The appointment is successfully canceled.

Failure End:

- The appointment cannot be canceled due to a system error.

Trigger:

- When the doctor selects the "Cancel Appointment" option for a scheduled appointment.

Main Success Scenario:

1. The doctor views the list of scheduled appointments.
2. The doctor selects the option to cancel an appointment.
3. The system prompts the doctor to confirm the cancellation.
4. The doctor confirms the cancellation.
5. The system updates the appointment status to "Canceled."
6. The system confirms the update and notifies the doctor.

Extensions (Error Scenarios):

1a. System failure:

- The system fails to cancel the appointment and informs the doctor to try again later.
-

Use Case: Logout

Goal:

- Allow patients, doctors, and admins to log out of the system.

Primary Actor(s):

- Patient
- Doctor
- Admin

Preconditions:

- The user is logged into the system.
- The system is working.

Success End:

- The user successfully logs out of the system.

Failure End:

- The logout process fails due to a system error.

Trigger:

- When the user clicks the "Logout" button.

Main Success Scenario:

1. The user selects the "Logout" option.
2. The system invalidates the user's session.
3. The system logs the user out and redirects them to the login page.
4. The system displays a confirmation message indicating a successful logout.

Extensions (Error Scenarios):

1a. System failure:

- The system fails to log out the user and displays an error message, prompting them to try again later.
-

Use Case: Send Email Reminder

Goal:

- Allow the system to send email reminders to patients regarding their upcoming appointments or important health notifications.

Primary Actor(s):

- Admin

Preconditions:

- The patient is logged into the system.
- The patient has a valid email address registered in the system.
- The system is working.

Success End:

- The email reminder is successfully sent to the patient.

Failure End:

- The email reminder fails to send due to a system error or invalid email address.

Trigger:

- Scheduled to send reminders for upcoming appointments or events.

Main Success Scenario:

1. The system identifies patients with upcoming appointments or health-related notifications.
2. The system prepares an email reminder containing the necessary details (appointment date, time, location).
3. The system sends the email reminder to the patient's registered email address.
4. The system logs the status of the email sent (successful or failed).

Extensions (Error Scenarios):

1a. Invalid email address:

- The system detects that the registered email address is invalid and prompts to update it.

1b. Email sending failure:

- The system fails to send the email due to a server issue and logs the error, prompting the system admin to investigate.
-