

BEAUTY SALON APPOINTMENT SYSTEM

Requirement Analysis

Modern beauty salons often struggle with manual appointment tracking, which causes scheduling conflicts and inefficiency.

This project aims to develop a digital **Appointment Management System** that allows customers to book, cancel, or update their appointments easily through a web or mobile interface.

Employees (such as hairdressers) can view and confirm their own appointments, while administrators manage services, staff, and system operations.

Objective:

To digitalize the appointment process, reduce scheduling errors, and improve both customer satisfaction and staff efficiency.

Use Case Diagram

Actors

- **Customer:** Creates, cancels, and views appointments or services.
- **Employee:** Views assigned appointments and confirms them.
- **Administrator:** Manages employees and salon services.
- **System:** Automatically sends reminders to customers before their appointment time.

Use Cases

No	Use Case	Description	Actor
1	Book Appointment	Customer selects a service and schedules a time	Customer
2	Cancel Appointment	Customer cancels an existing booking	Customer
3	View Services	Customer views available services and prices	Customer
4	Approve Appointment	Employee confirms or rejects a booked appointment	Employee
5	Add/Delete Service	Administrator manages service listings	Administrator
6	Add/Delete Employee	Administrator manages employee accounts	Administrator
7	Send Reminder	System sends automatic reminders	System

Detailed Use Case Scenario

- **Main Flow:**

• Step	• User Action (Customer)	• System Response
• 1	<ul style="list-style-type: none">• The Customer selects the "Book Appointment" option from the main menu.	<ul style="list-style-type: none">• The System retrieves and displays a list of available services (e.g., Haircut, Coloring, Manicure) and their prices.
• 2	<ul style="list-style-type: none">• The Customer selects a specific service (e.g., Haircut).	<ul style="list-style-type: none">• The System displays a list of available Employees and their open time slots for the selected service.
• 3	<ul style="list-style-type: none">• The Customer selects a preferred Employee (optional) and chooses a specific date and time slot.	<ul style="list-style-type: none">• The System validates the availability of the selected slot (Check Availability) and prompts the Customer to confirm the booking details.
• 4	<ul style="list-style-type: none">• The Customer reviews the details and clicks the "Confirm Booking" button.	<ul style="list-style-type: none">• The System changes the status of the time slot to "Booked," stores the appointment in the database, and generates a booking reference number.
• 5	<ul style="list-style-type: none">•	<ul style="list-style-type: none">• The System displays a success message on the screen and sends an automated confirmation email to the Customer's registered address.

Detailed Use Case Scenario

Use Case Name: UC-01: Book Appointment

Primary Actor: Customer

Scope: Beauty Salon Appointment System

Precondition: The Customer is logged into the system².

Postcondition: The appointment is successfully stored in the system, and a confirmation email is sent to the Customer³.

Main Flow of Events:

Alternative Flows:

- **A1: Selected Time Slot becomes Unavailable (At Step 3)**
 - **Condition:** If the selected time slot was taken by another user just before confirmation.
 - **System Response:** The System displays an error message stating "The selected time is no longer available" and suggests the nearest alternative time slots (Suggest Alternative Slot).
 - **User Action:** The Customer selects a new time slot or cancels the process.
- **A2: Customer Cancels the Booking**
 - **User Action:** The Customer clicks the "Cancel" button.
 - **System Response:** The System clears any selected data, releases any held resources, and returns the Customer to the main dashboard. The booking process ends.

Special Requirements (Non-functional):

- The system must confirm the appointment and send the email within 30 seconds of confirmation.

E1: Selected Time Slot is Unavailable (<<extend>> Suggest Alternative Slot)

Step	User Action (Customer)	System Response
3.a	The Customer selects a time slot.	<p>System Condition: The System detects the slot was just taken by another user.</p> <p>System Action: The System triggers the Suggest Alternative Slot use case. It displays an error message ("Slot no longer available") and shows the nearest open slots.</p>
3.b	The Customer selects one of the suggested alternative slots.	The System updates the selection and returns to Step 4 of the Main Flow to proceed with confirmation.
3.c	(Optional) The Customer cancels the process.	The System terminates the use case.

E2: Customer Cancels the Booking (General Exception)

Step	User Action (Customer)	System Response
All	The Customer clicks the "Cancel" button.	<p>System Action: The System clears all input data, releases any temporarily held resources, and redirects the Customer to the Dashboard. The use case ends unsuccessfully.</p>

Harika, bu bölümü Use Case diyagramındaki <<extend>> ve <<include>> ilişkileriyle birebir örtüşecek şekilde teknik bir dille yeniden yazdım.

Kitapta ve profesyonel analiz dokümanlarında, "Alternative Flows" başlığı yerine, diyagramdaki <<extend>> ilişkisini vurgulamak için "**Extensions**" veya "**Exception Flows**" başlıkları kullanılır. Ayrıca, ana akışın (Main Flow) içine <<include>> edilen Use Case'ler referans olarak eklenir.

İşte "Book Appointment" senaryosunun, ilişkileri (Include/Extend) açıkça belirten ve "User Action - System Response" formatına sadık kalan son hali:

Detailed Use Case Scenario

Use Case Name: UC-01: Book Appointment

Primary Actor: Customer

Relationships:

- **Includes:** Log In, Check Availability
- **Extends:** Suggest Alternative Slot

Precondition: Customer is logged into the system (<<include>> Log In performed).

Postcondition: Appointment is stored, and confirmation is sent.

Main Flow of Events

Step	User Action (Customer)	System Response
1	The Customer selects "Book Appointment".	The System displays the list of services.
2	The Customer selects a service (e.g., Haircut).	The System displays available Employees and time slots.
3	The Customer chooses a specific date and time.	(<<include>> Check Availability): The System verifies that the selected slot is free.
4	The Customer confirms the booking.	The System stores the appointment and generates a booking ID.
5		The System sends a confirmation email.

Extensions (Handling <<extend>> and Exceptions)

E1: Selected Time Slot is Unavailable (<<extend>> Suggest Alternative Slot)

- **Extension Point:** Occurs at **Step 3** of Main Flow if Check Availability returns false.

Step	User Action (Customer)	System Response
3.a	The Customer selects a time slot.	<p>System Condition: The System detects the slot was just taken by another user.</p> <p>System Action: The System triggers the Suggest Alternative Slot use case. It displays an error message ("Slot no longer available") and shows the nearest open slots.</p>
3.b	The Customer selects one of the suggested alternative slots.	The System updates the selection and returns to Step 4 of the Main Flow to proceed with confirmation.
3.c	(Optional) The Customer cancels the process.	The System terminates the use case.

E2: Customer Cancels the Booking (General Exception)

- **Extension Point:** Can occur at any step before **Step 4**.

Step	User Action (Customer)	System Response
All	The Customer clicks the "Cancel" button.	<p>System Action: The System clears all input data, releases any temporarily held resources, and redirects the Customer to the Dashboard. The use case ends unsuccessfully.</p>

Special Requirements (Non-functional)

- **Performance:** The System must confirm the appointment and send the confirmation email within 30 seconds of the user confirming the booking.
- **Usability:** The interface must clearly highlight unavailable slots in grey to prevent selection errors.

Conclusion

The **Beauty Salon Appointment System** provides a modern digital solution for managing appointments and salon services.

Through UML diagrams, the system's **actors, interactions, and data structure** are clearly modeled.

Use Case and Class diagrams together help visualize both the **functional** and **structural** aspects of the system making it easier to develop, test, and extend in future.

Ebrar İkbāl ÇALIŞKAN

210202030