King Saud University College of Computer and Information Sciences Computer Science Department	
CSC 343 System Analysis and Design	Second semester 1446

Salma Project

Group 6

Member Name	ID	Your Contribution
Jumana alothman	**443202103**	**Architectural Diagram, UI**
Wajd alquwayi	**443201429**	**Design of test , UI, collect requirements, Traceability matrix**
Dimah Althunayan	**444201003**	**Interaction Diagrams, class diagram, state diagram, UI, sequence diagram**

Summary of Updates

Section	Author	Update details	date
Requirement	Dema	Req-3, Req-4, Req-5, Req-9, Req-16, Req-18 added	20-3
Requirement	Dema	Req-6, Req-7, Req-8, Req-10, Req-11, Req-12, Req-13, Req-14, Req-15 updated	20-3
Fully described use cases	Dema	UC-2 ,UC-7, UC-13	20-3

1.Customer Statement of Requirements (CSR)

1.1 Problem Statement

As a working mother, my days are a constant balancing act between my career, home responsibilities, and, most importantly, raising my child. Managing everything at once is overwhelming, and while I do my best, there are times when I need an extra pair of hands. I need time to focus on work, take care of myself, or simply rest without constantly worrying about my child's well-being.

Finding a reliable babysitter is one of the biggest challenges I face. I need someone who is not only trustworthy but also experienced and well-trained to take care of my child. Safety is my top priority—I want to ensure that the person watching over my child is qualified, responsible, and has been thoroughly vetted. However, the current process of finding a babysitter is unreliable and time-consuming.

Recommendations from friends and family are helpful but limited, and online options often lack credibility. That's where Salma comes in.

Salma is the solution we, as parents, have been searching for. It is an online babysitting platform designed to simplify the process of finding trusted and professional babysitters. Through Salma, I can browse verified babysitter profiles, check their qualifications, and even interview them before making a booking.

What makes Salma stand out is its focus on safety and quality. The platform ensures that every babysitter is legally vetted through a national identification system, giving me peace of mind. Additionally, Salma offers specialized services for children with special needs, allowing parents like me to find the right caregiver tailored to their child's unique requirements.

With Salma, I no longer have to stress about childcare. Whether I need a babysitter for a few hours or on a regular basis, I can trust that my child is in good hands. Salma empowers working parents by providing a seamless, secure, and professional babysitting service—because every parent deserves peace of mind.

1.2 Glossary of Terms

Term	Description
Babysitter	A person hired to provide temporary childcare services.
On-Demand Babysitting	A service that allows parents to book babysitters as needed, without long-term commitments.
Vetted Babysitter	A babysitter who has undergone a background check and verification process.
National Identification System	A government-based system used to verify the identity and background of individuals.
Special Needs and Requests for Child	Babysitting services tailored to children with disabilities or special care requirements, or even teaching child
Flexible Booking	The ability to schedule babysitters based on the parent's needs, whether for a few hours or recurring sessions.
Parental Peace of Mind	The assurance that a parent feels when their child is under the care of a trustworthy babysitter.
Professional Babysitter Profile	A detailed profile containing a babysitter's experience, skills, and background information.
Child Safety Measures	Procedures and policies in place to ensure the security and well-being of children under a babysitter's care.

2.System Requirements

2.1 Enumerated functional Requirements

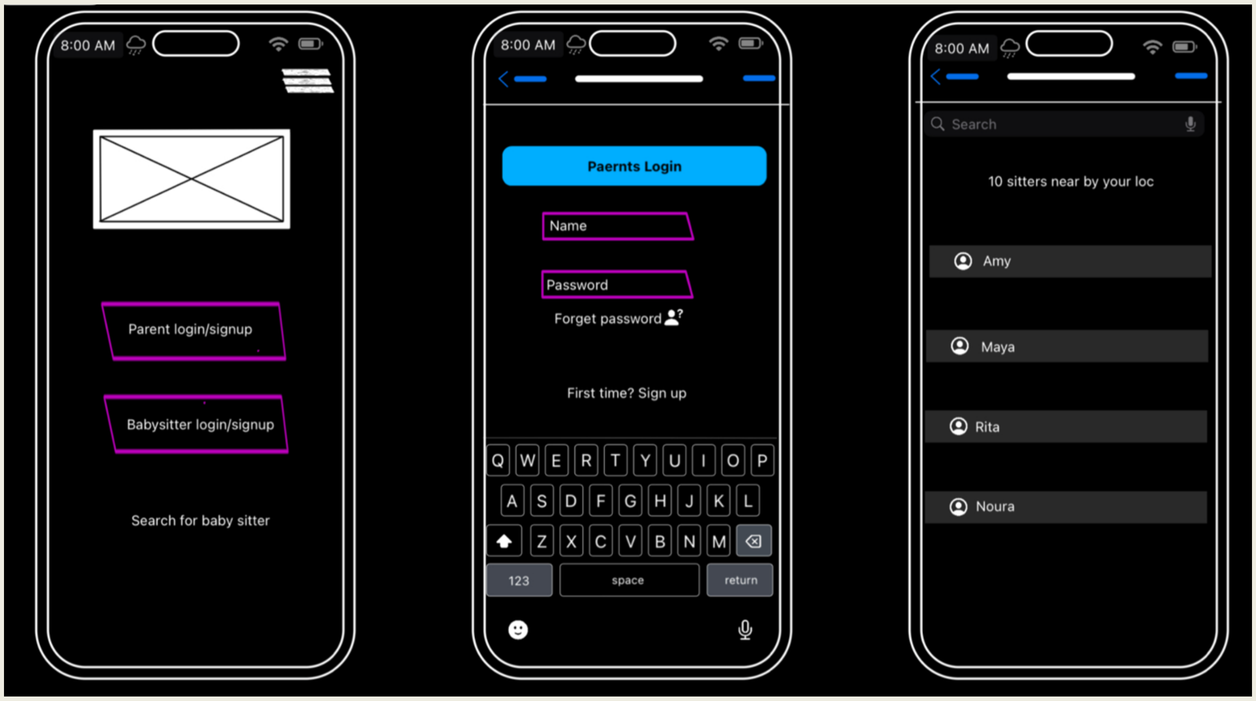
REQ-x	PW	Description
REQ-1	10	System should allow secure user authentication and log in
REQ-2	10	System should verify babysitters with national ID via Nafath
REQ-3	10	The system shall allow babysitters to create a profile that includes personal information, experience, certifications, availability calendar, booking preference (regular basis, one-time, or both), languages spoken, number of children they can care for, and special care capabilities
REQ-4	9	The system shall allow babysitters to set and update their availability calendar to manage when they are available for bookings
REQ-5	9	The system shall allow babysitters to set their hourly rate for bookings
REQ-6	8	The system shall allow parents to view babysitters' profiles, which include background information, ratings, reviews, availability calendar, the number of children they can care for, booking preferences (regular basis, one-time, or both), special care capabilities, and the option to contact them through chat
REQ-7	10	The system shall allow parents to filter babysitters based on availability, experience, ratings, languages spoken, location proximity, hourly rate, number of children the babysitter can care for, and special care capabilities
REQ-8	7	The system shall allow parents to book babysitters by specifying the booking type (one-time or regular), date, start time, duration, number of children, and the age of each child

REQ-9	10	The system shall ensure that parents can only book babysitters who are available based on their availability calendar
REQ-10	10	The system shall allow babysitters for a predefined time to accept or decline a booking request, after which it shall be automatically declined
REQ-11	8	The system shall enforce a cancellation policy preventing either party from canceling a booking less than 12 hours before the scheduled time
REQ-12	10	The system shall provide secure payment
REQ-13	10	The system shall display the babysitter's hourly rate before confirming a booking
REQ-14	10	The system shall generate an invoice for each completed booking and provide it to the parent
REQ-15	9	The system shall allow parents to track the babysitter's location in real time while they are on duty
REQ-16	10	The system shall allow parents to communicate with the babysitter while they are on duty through messaging and video calls
REQ-17	8	The system should have search history for the babysitter
REQ-18	8	The system shall provide customer support for both parents and babysitters

2.2 non-functional requirements

NFR-x	PW	Description
NFR-1	10	Performance: The system should load within 2 seconds on mobile and web platforms. Scalability: The system should support up to 10,000 concurrent users without performance issues.
NFR-2	8	Support & Maintenance: The system should provide 24/7 customer support for parents and babysitters.
NFR-3	7	Logging & Monitoring: The system should maintain logs of user activities for security and auditing purposes.
NFR-4	8	Data Backup: The system should perform automatic backups every 24 hours to prevent data loss.
NFR-5	10	Usability: The app should have an intuitive and user-friendly interface that is easy for parents and babysitters to navigate
NFR-6	8	Localization: The system should support multiple languages, including Arabic and English.
NFR-7	10	The system should provide the loc for the parent and babysitter before booking via google maps API

2.3 UX Sketch


8:00 AM
Parent login/signup
Babysitter login/signup
Search for baby sitter
8:00 AM
Paernts Login
Name
Password
Forget password
First time? Sign up
8:00 AM
Search
10 sitters near by your loc
Amy
Maya
Rita
Noura

3.Functional Requirements Specification

3.1 Stakeholders

• Parents (Users) – Need babysitting services for their children.

• Babysitters (Users) – Provide babysitting services.

• System Administrators – Manage and monitor platform operations.

• Payment Providers (Banks, Online Payment Services)

• Government/National ID System Nafath – Verify babysitters' legal identity.

• Customer Support Team – Assists users with technical issues and complaints

• Sponsors – Enlighten Our Minds School (Surve for the families , testing the app,collaboration with their teachers for extra incomes via the app)

3.2 Actors and Goals

Actors	Actors Goals
Parent/Initiating actor	Find and book a trusted babysitter.- Communicate with the babysitter.Manage payments securely.
Babysitter/ Initiating actor	Register and verify identity. Set availability and accept jobs. Communicate with parents.- Receive payments after service.
System Admin/Participating actor	Approve babysitters.- Ensure platform security.- Resolve disputes and handle complaints.

National ID System (Nafath)/ Participating actor	Verify babysitters identities.Prevent fraud.
Payment Gateway/ Participating actor	Process secure payments. Provide multiple payment options.

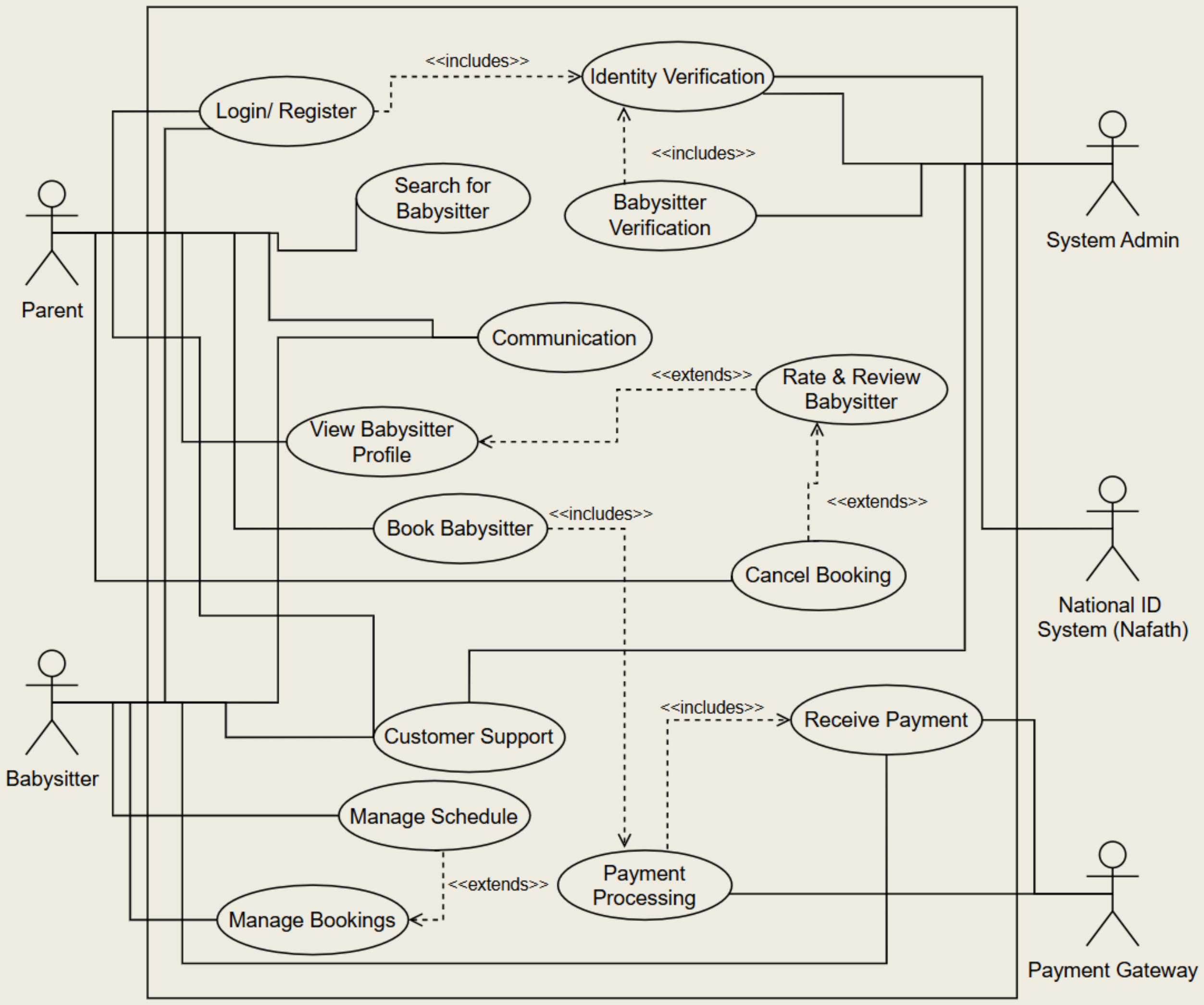
3.3 Use cases

3.3.1 Casual Descriptions of Use Cases

UC-x	Name	Descriotion
UC-1	Login/ Register	Parents and babysitters log in using their email and password. Users create an account by providing basic details and ID verification (for babysitters)
UC-2	Search Babysitter	Parents search for available babysitters based on location, skills, and availability.
UC-3	View Babysitter Profile	Parents view a babysitter's ratings, reviews, and experience before booking.
UC-4	Communicatio n	Parents and babysitters can chat or video call through the app.
UC-5	Identity Verification	Babysitters' legal identity is verified before allowing bookings.
UC-6	Babysitter Verification	The system verifies babysitters using the National ID system.

UC-7	Book Babysitter	Parents request a booking by selecting the babysitter date, time, and child’s age.
UC-8	Receive Payment	Babysitters receive have of the payments securely BEFORE completing a job and the rest after , tips are allowed ,the amount dep on the nanny and her skills
UC-9	Payment Processing	Parents complete secure payments via credit card, Apple Pay.
UC-10	Rate & Review Babysitter	Parents rate the babysitter and provide feedback after a session.
UC-11	Manage Schedule	Babysitters update their availability for bookings.
UC-12	Manage Bookings	Babysitters accept or reject incoming requests.
UC-13	Cancel Booking	Parents can cancel bookings, but cancellation policies MUST apply.
UC-14	Customer Support	Parents and babysitters contact support for technical issues or complaints.

3.3.2 Use case Diagram



3.3.3 Use Case Fully Dressed Description

UC1 <Login/Register>
This use case describes the process of parents and babysitters logging in or registering for an account on the Salma platform.
Initiating Actor: Parents, Babysitters **Participation Actor:** National ID System (Nafath)
Actors Goal: Allow users to securely log in or register an account to access the platform.
Pre-Condition: The user is on the login/registration page.
Flow of Events for Success Scenario: 1. The user presses the "Login/Register" button on the platform's homepage. 2. The system displays a form asking the user to choose between logging in or registering. 3. If logging in: • The user enters their email and password in the provided fields. • The system verifies the credentials. • If correct, the user is redirected to their dashboard. 4.If registering: • The user selects their account type (Parent or Babysitter). • The system displays a registration form where: i. Parents enter their name, email, phone number, and password. ii. Babysitters enter their name, email, phone number, password, and National ID for verification. 5. The user submits the registration form. 6. The system validates the provided information. 7. If the user is a babysitter, the system verifies their National ID via Nafath. 8.The system sends a confirmation email to the user. 9. The user checks their email inbox and clicks on the confirmation link. 10.The system verifies the email and activates the account. 11. The user is redirected to the login page and enters their credentials. 12.The system verifies the login details and grants access to the platform.

Flow of Events for Extension (Alternate Scenario): 1. Incorrect Login Credentials: • If the user enters an incorrect email or password, the system displays an error message: *"Invalid email or password. Please try again."* • The user can attempt to log in again or reset their password. 2.Duplicate Email During Registration: • If the email is already registered, the system displays an error message: *"This email is already in use. Please log in or use a different email."* • The user must provide a unique email to continue. 3. Failed National ID Verification (for Babysitters): • If the babysitter's ID verification fails via Nafath, the system displays an error message: *"ID verification failed. Please check your details and try again."* • The babysitter must provide correct information for verification. 4. Email Confirmation Not Completed: • If the user does not verify their email, they cannot log in. • The system displays a message: *"Please verify your email before logging in."* • The user can request a new confirmation email.
Post-Condition: The user is successfully logged in and can access the platform. If registering for the first time, a new user account is created.

UC2 <Search Babysitter>
This use case describes the process of parents searching for babysitters based on their preferences and availability.
Initiating Actor: Parent
Actors Goal: Find and select a suitable babysitter based on filters such as location, experience, and availability.
Pre-Condition: The parent is logged into their account and is on the babysitter search page.
Flow of Events for Success Scenario: 1. The parent navigates to the "Search Babysitter" page. 2. The system displays a search bar and filtering options (location, experience, hourly rate, ratings, availability , languages spoken , number of children, and special care capabilities). 3. The parent enters search criteria and presses the "Search" button. 4. The system retrieves babysitters from the database matching the criteria. 5. The system displays a list of available babysitters with their profiles, including: • Name • Profile picture • Experience • Ratings • Location • Hourly rate 6. The parent clicks on a babysitter's profile to view more details. 7. The system displays the babysitter's full profile with: • background information • Certifications (if available) • Ratings and reviews • Availability calendar • the number of children they can care for • booking preferences (regular basis, one-time, or both) • Contact/request options

Flow of Events for Extension (Alternate Scenario): 1. No Matching Babysitters Found: • If no babysitters match the parent's search criteria, the system displays a message: *"No babysitters found. Try adjusting your filters."* 2. System Error or Timeout**:** If a system error occurs during the search, the system displays a message: *"Something went wrong. Please try again later.*
Post-Condition: • A babysitter is successfully found and booked.

UC7<Book Babysitter>

This use case describes the process of parents booking a babysitter after selecting one from the search results.

Initiating Actor: Parent

Actors Goal: Successfully book a babysitter for a selected date and time.

Pre-Condition: The parent has selected a babysitter and is logged into their account.

Flow of Events for Success Scenario:

1. The parent selects a babysitter from the available list.
2. The system displays the babysitter's full profile.
3. The parent clicks the "Book Now" button.
4. The system prompts the parent to enter booking details:
    - Booking type (regular basis / one-time)
    - Date and time
    - Duration
    - Number of children
    - Age of each child
    - Special instructions (optional)
5. The parent reviews the booking details and confirms the request.
6. The system sends a booking request to the selected babysitter.
7. The babysitter receives the request and can either accept or decline it.
8. If the babysitter accepts the request, the system ask the parent to pay, after that it confirms the booking and notifies both the parent and the babysitter.
9. The system updates the babysitter's availability for the booked time slot.
10. The booking details are stored, and both the parent and babysitter can view them in their accounts.

Flow of Events for Extension (Alternate Scenario):

1. Babysitter Declines the Booking Request:
    - If the babysitter declines, the system notifies the parent and allows them to select another babysitter.
2. Parent Cancels the Booking Request:
    - If the parent cancels before the babysitter responds, the system withdraws the request and notifies the babysitter.
3. Babysitter Does Not Respond Within a Time Limit:
    - If the babysitter does not respond within a predefined time, the system will automatically decline the request and notifies the parent.
4. System Error or Timeout:
    - If a system error occurs, the system displays a message:
      "Something went wrong. Please try again later."

Post-Condition:

- A babysitter is successfully booked for the specified date and time.
- Both the parent and babysitter receive booking confirmation.

UC13 <Cancel Booking>

This use case describes the process of a parent or babysitter canceling an already confirmed babysitting booking.

Initiating Actor: Parent or Babysitter

Actors Goal: Successfully cancel a confirmed babysitting booking.

Pre-Condition: The parent or babysitter has an existing confirmed booking.

Flow of Events for Success Scenario:

1. The initiating actor (parent or babysitter) navigates to their "My Bookings" page.
2. The system displays a list of confirmed bookings.
3. The initiating actor selects the booking they wish to cancel.
4. The system displays the booking details along with a "Cancel Booking" button.
5. The initiating actor clicks the "Cancel Booking" button.
6. The system prompts for confirmation, displaying a message:
   "Are you sure you want to cancel this booking?"
7. The initiating actor confirms the cancellation.
8. The system processes the cancellation and updates the booking status to Canceled.
9. The system initiates a refund through payment gateway and confirms that it has been processed successfully.
10. The system notifies both parties (parent and babysitter) about the cancellation.
11. The system updates the babysitter's availability calendar based on the cancellation.

Flow of Events for Extension (Alternate Scenario):

1. Attempt to Cancel Too Close to the Booking Time:
   - If the cancellation is attempted too close to the start time (based on platform rules), the system displays a message:
     "Cancellations are not allowed within 12 hours of the booking time."
2. System Error or Timeout:
   - If a system error occurs during the cancellation process, the system displays an error message:
     "Something went wrong. Please try again later."

Post-Condition:

- The booking is successfully canceled.
- The other party (parent or babysitter) is notified of the cancellation, and the babysitter's availability is updated.

UC12 < Manage Bookings>
This use case describes the process of babysitters managing their booking requests by accepting or rejecting incoming requests.
Initiating Actor: Babysitters **Participation Actor:** Parent System Admin
Actors Goal: • Babysitters should be able to view and respond to booking requests. • Parents should receive confirmation when a babysitter accepts or rejects a booking. • The system should update schedules accordingly.
Pre-Condition: • The babysitter must be logged into their account. • The babysitter must have at least one pending booking request.
Flow of Events for Success Scenario: 1. The babysitter navigates to the "Manage Bookings" section from their dashboard. 2. The system retrieves a list of pending booking requests. 3. The babysitter selects a booking request to review its details (e.g., date, time, child's age, location). 4. The babysitter chooses one of the following actions:4.If registering: • Accept Booking: The system ask the parent to pay a. The system confirms the acceptance and updates the booking status. b. A confirmation notification is sent to the parent. c. The babysitter's schedule is updated. • Reject Booking: a. The system updates the status as "Rejected." b. A rejection notification is sent to the parent. 5. The system updates the availability status of the babysitter based on the accepted or rejected booking.
Flow of Events for Extension (Alternate Scenario): 1. Parent Cancels Booking Before Confirmation: • If a parent cancels the booking before the babysitter responds, the system removes it from the pending requests.

2. Babysitter Tries to Accept a Booking That is No Longer Available: • If another babysitter has already accepted the booking, the system notifies the babysitter that the request is no longer available. 3. Babysitter Does Not Respond Within a Time Limit: • If the babysitter does not respond within a predefined time, the system will automatically decline the request and notifies the parent 4. System Error or Timeout: • If a system error occurs during the cancellation process, the system displays an error message: *"Something went wrong. Please try again later."*
Post-Condition: • If the babysitter accepts the booking, the system updates their schedule, and the parent is notified. • If the babysitter rejects the booking, the system marks it as unavailable, and the parent is notified.

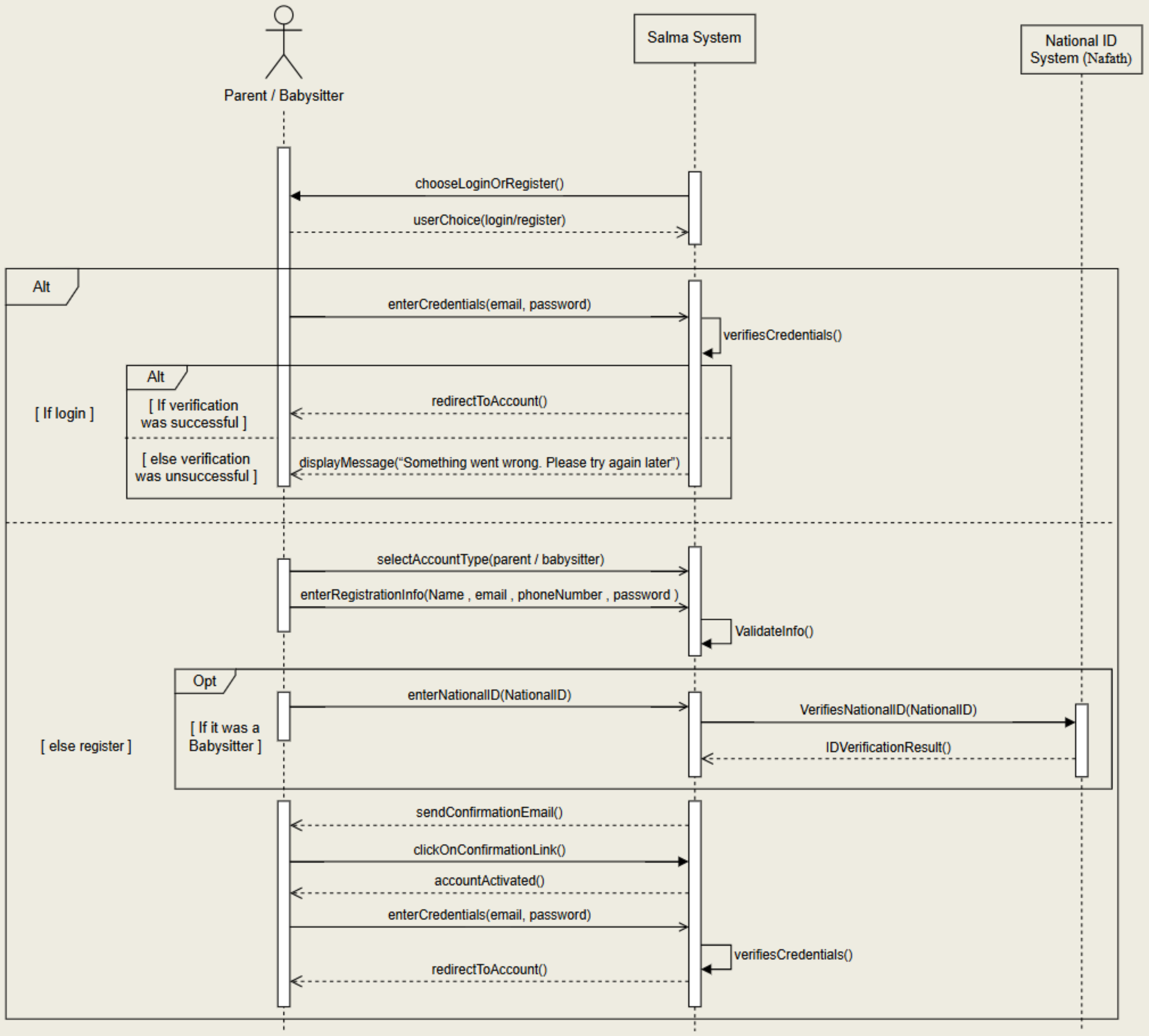
3.3.3.1Traceability matrix

REQ -x	UC 1	UC 2	UC 3	UC 4	UC 5	UC 6	UC 7	UC 8	UC 9	UC 10	UC 11	UC 12	UC 13	UC 14
REQ -1	X													
REQ -2														
REQ -3					X									
REQ -4							X							
REQ -5											X			
REQ -6		X												
REQ -7									X					
REQ -8				X										
REQ -9										X				
REQ -10								X						
REQ -11													X	
REQ -12										X				

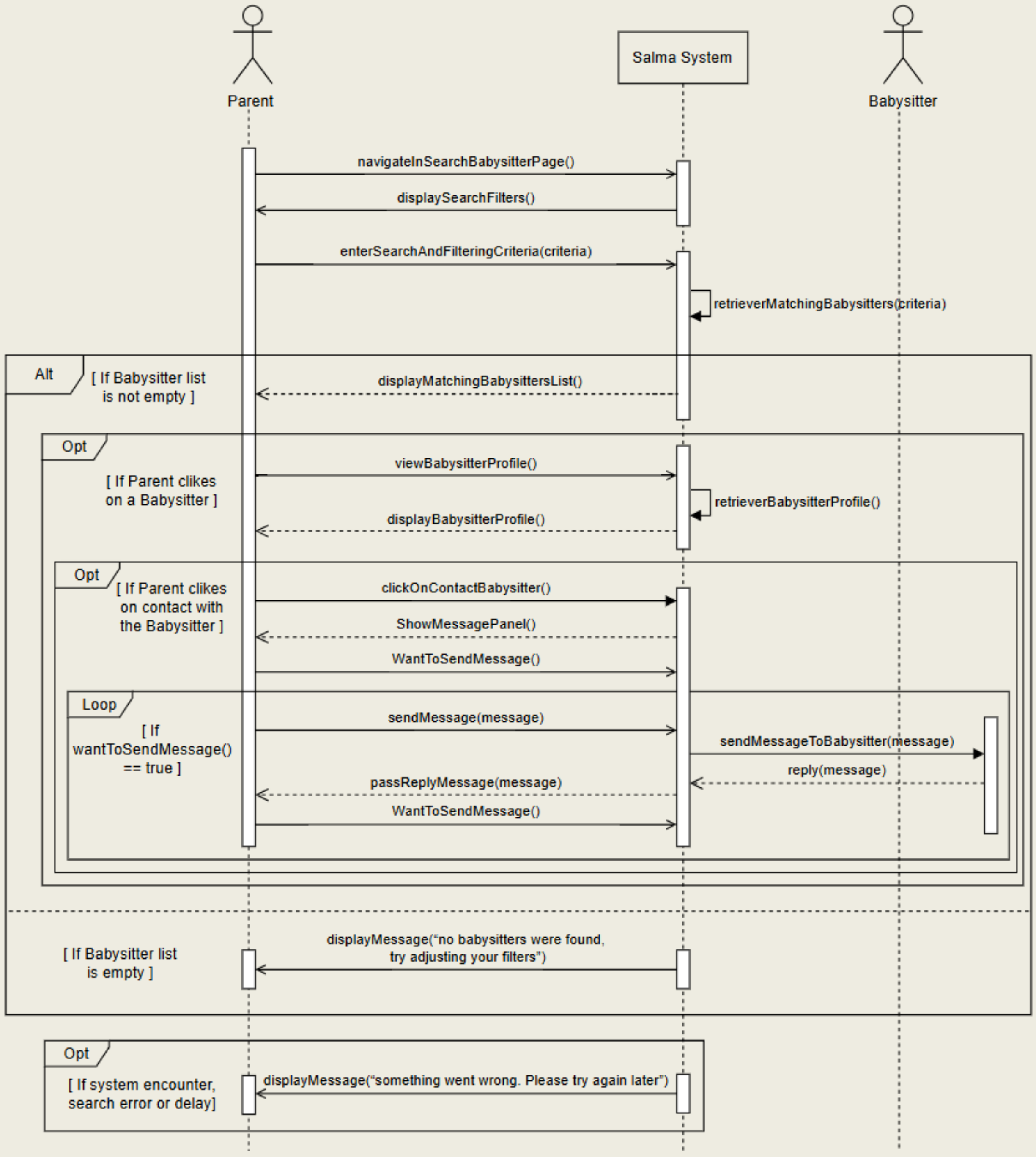
4.Interaction Diagrams

4.1 Sequence Diagrams

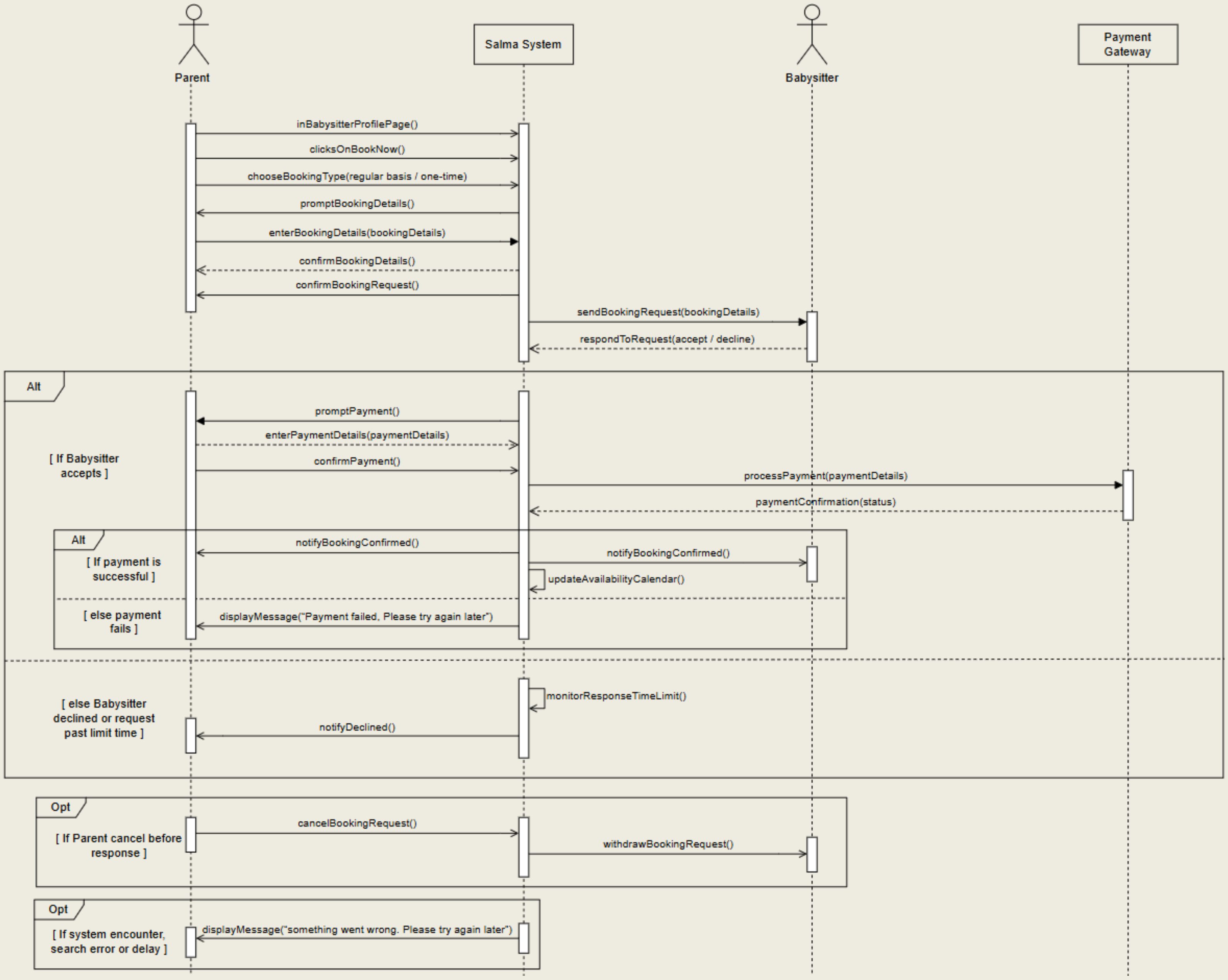
4.1.1 UC1 <Login/Register>


Parent / Babysitter
Salma System
National ID System (Nafath)
chooseLoginOrRegister()
userChoice(login/register)
Alt
enterCredentials(email, password)
verifiesCredentials()
Alt
[If login]
[If verification was successful]
redirectToAccount()
[else verification was unsuccessful]
displayMessage("Something went wrong. Please try again later")
selectAccountType(parent / babysitter)
enterRegistrationInfo(Name , email , phoneNumber , password)
ValidateInfo()
Opt
enterNationalID(NationalID)
[If it was a Babysitter]
VerifiesNationalID(NationalID)
IDVerificationResult()
[else register]
sendConfirmationEmail()
clickOnConfirmationLink()
accountActivated()
enterCredentials(email, password)
verifiesCredentials()
redirectToAccount()

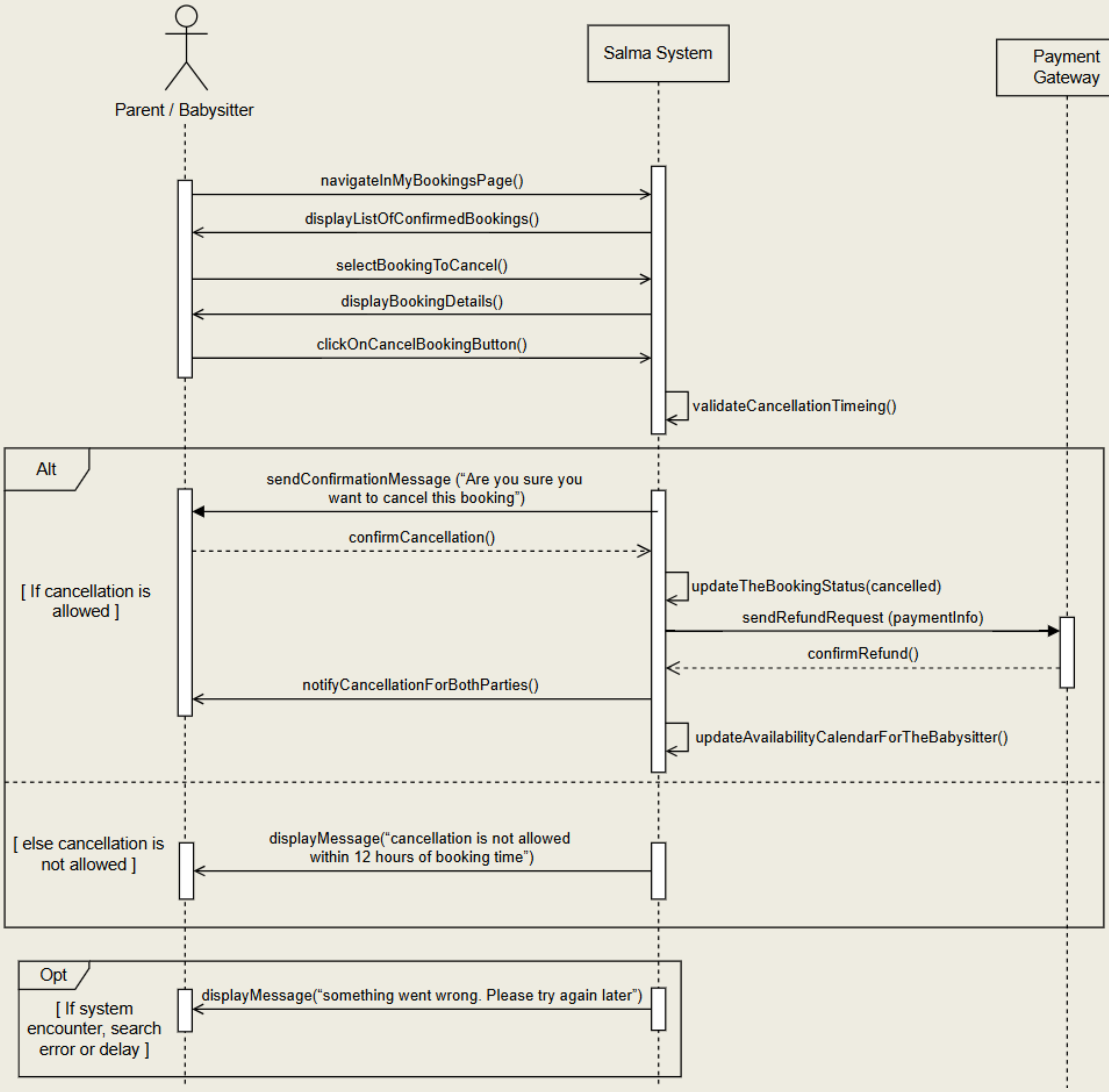
4.1.2 UC2 <Search Babysitter>


Parent
Salma System
Babysitter
navigateInSearchBabysitterPage()
displaySearchFilters()
enterSearchAndFilteringCriteria(criteria)
retrieverMatchingBabysitters(criteria)
Alt
[If Babysitter list is not empty]
displayMatchingBabysittersList()
Opt
[If Parent clikes on a Babysitter]
viewBabysitterProfile()
retrieverBabysitterProfile()
displayBabysitterProfile()
Opt
[If Parent clikes on contact with the Babysitter]
clickOnContactBabysitter()
ShowMessagePanel()
WantToSendMessage()
Loop
[If wantToSendMessage() == true]
sendMessage(message)
sendMessageToBabysitter(message)
reply(message)
passReplyMessage(message)
WantToSendMessage()
[If Babysitter list is empty]
displayMessage("no babysitters were found, try adjusting your filters")
Opt
[If system encounter, search error or delay]
displayMessage("something went wrong. Please try again later")

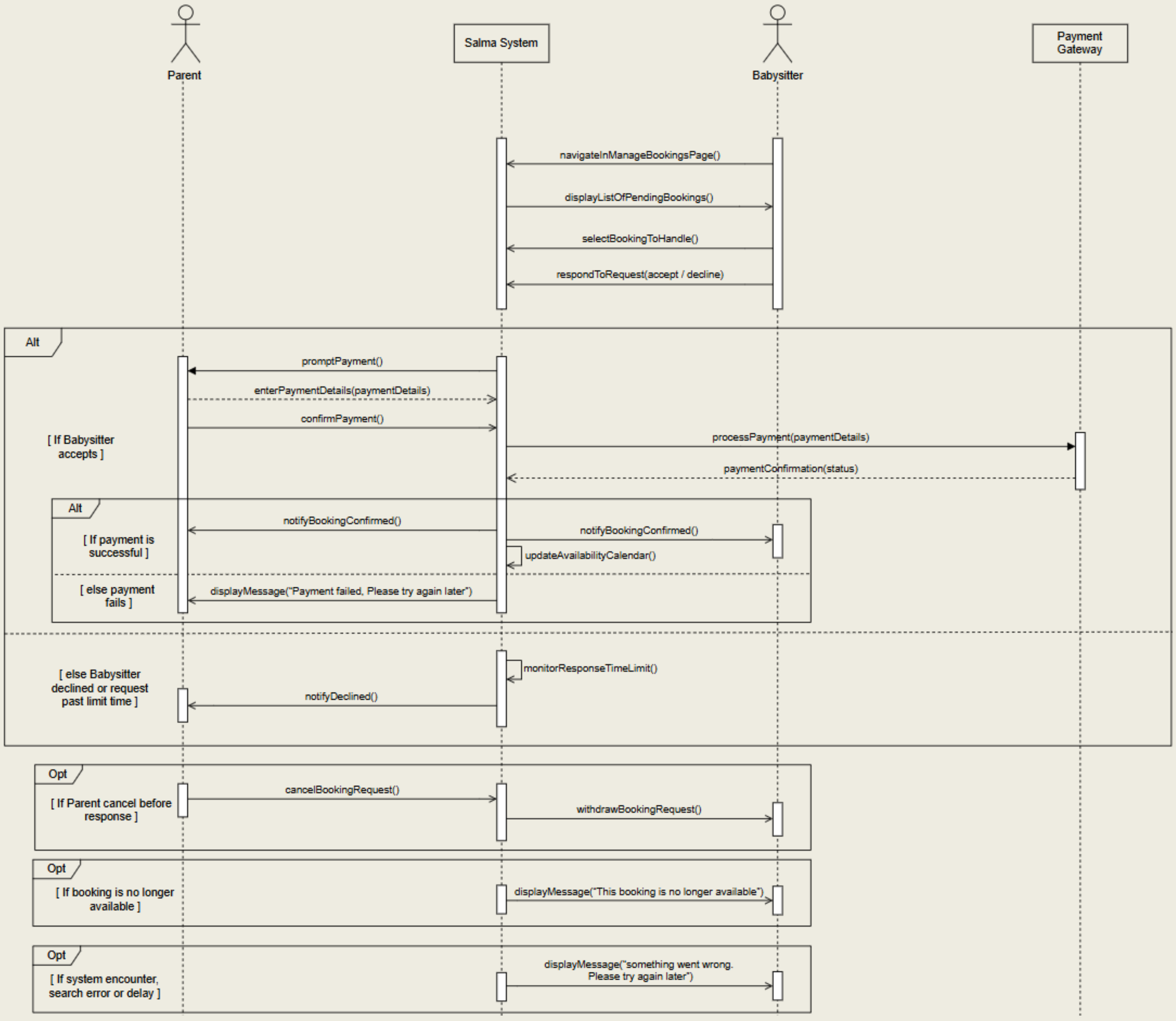
4.1.3 UC7 <Book Babysitter>


Parent
Salma System
Babysitter
Payment Gateway
inBabysitterProfilePage()
clicksOnBookNow()
chooseBookingType(regular basis / one-time)
promptBookingDetails()
enterBookingDetails(bookingDetails)
confirmBookingDetails()
confirmBookingRequest()
sendBookingRequest(bookingDetails)
respondToRequest(accept / decline)
Alt
[If Babysitter accepts]
promptPayment()
enterPaymentDetails(paymentDetails)
confirmPayment()
processPayment(paymentDetails)
paymentConfirmation(status)
Alt
[If payment is successful]
notifyBookingConfirmed()
notifyBookingConfirmed()
updateAvailabilityCalendar()
[else payment fails]
displayMessage("Payment failed, Please try again later")
[else Babysitter declined or request past limit time]
monitorResponseTimeLimit()
notifyDeclined()
Opt
[If Parent cancel before response]
cancelBookingRequest()
withdrawBookingRequest()
Opt
[If system encounter, search error or delay]
displayMessage("something went wrong. Please try again later")

4.1.4 UC13 <Cancel Booking>


Parent / Babysitter
Salma System
Payment Gateway
navigateInMyBookingsPage()
displayListOfConfirmedBookings()
selectBookingToCancel()
displayBookingDetails()
clickOnCancelBookingButton()
validateCancellationTimeing()
Alt
sendConfirmationMessage ("Are you sure you want to cancel this booking")
confirmCancellation()
[If cancellation is allowed]
updateTheBookingStatus(cancelled)
sendRefundRequest (paymentInfo)
confirmRefund()
notifyCancellationForBothParties()
updateAvailabilityCalendarForTheBabysitter()
[else cancellation is not allowed]
displayMessage("cancellation is not allowed within 12 hours of booking time")
Opt
[If system encounter, search error or delay]
displayMessage("something went wrong. Please try again later")

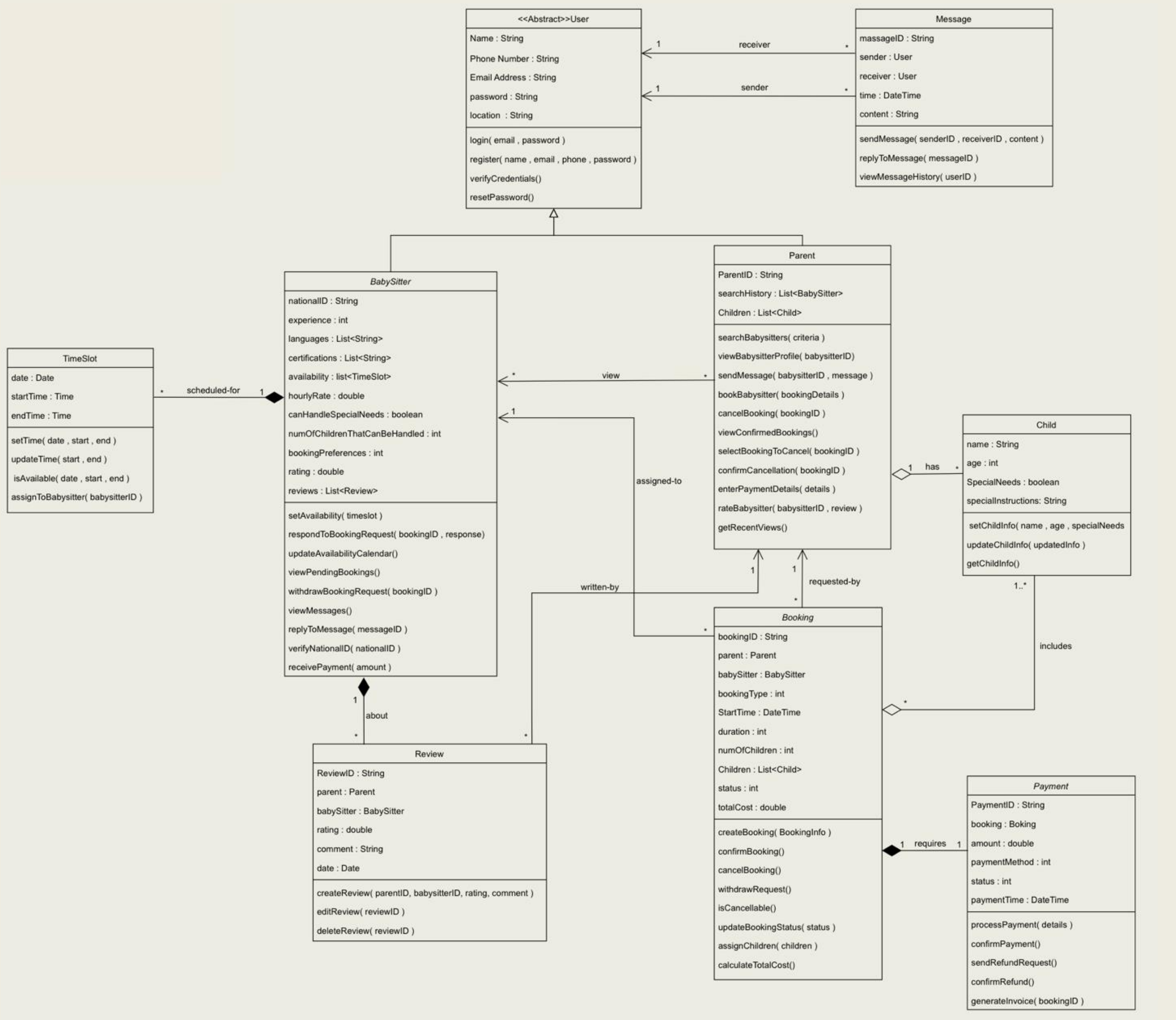
4.1.5 UC12 < Manage Bookings>


Parent
Salma System
Babysitter
Payment Gateway
navigateInManageBookingsPage()
displayListOfPendingBookings()
selectBookingToHandle()
respondToRequest(accept / decline)
Alt
[If Babysitter accepts]
promptPayment()
enterPaymentDetails(paymentDetails)
confirmPayment()
processPayment(paymentDetails)
paymentConfirmation(status)
Alt
[If payment is successful]
notifyBookingConfirmed()
notifyBookingConfirmed()
updateAvailabilityCalendar()
[else payment fails]
displayMessage("Payment failed, Please try again later")
[else Babysitter declined or request past limit time]
monitorResponseTimeLimit()
notifyDeclined()
Opt
[If Parent cancel before response]
cancelBookingRequest()
withdrawBookingRequest()
Opt
[If booking is no longer available]
displayMessage("This booking is no longer available")
Opt
[If system encounter, search error or delay]
displayMessage("something went wrong. Please try again later")

5.System Architecture and System Design

5.1 System Structural Diagram

5.1.1 Class Diagram



5.1.2 Methods and Attributes Description for Each Class

5.1.2.1 User (Abstract Class)

Attributes

- **name:** User’s full name
- **phoneNumber:** Contact number
- **emailAddress:** Email used for login and communication
- **password**: User’s hashed password
- **location:** User’s address or city

Methods

- **login(email, password):**

Input: email, password
Output: boolean (true if login succeeds)
Description: Validates credentials and starts a session

- **register(name, email, phone, password)**

Input: basic user info
Output: confirmation message or error
Description: Creates a new user account

- **verifyCredentials()**

Input: internal use
Output: boolean
Description: Checks stored credentials

- **resetPassword()**

Input: email or phone verification
Output: confirmation
Description: Allows user to reset password via email/SMS

5.1.2.2 Parent (inherits User)

Attributes

- **ParentID :** Refers to the ID of the parent
- **searchHistory:** List of babysitters that the parent viewed
- **children:** List of children tied to the parent when a booking is made

Methods

• searchBabysitters(criteria)
Input: filtering options
Output: list of matching babysitters
Description: Returns babysitters matching filters

• viewBabysitterProfile(babysitterID)
Input: babysitter’s ID
Output: Babysitter profile info
Description: Displays profile details

• sendMessage(babysitterID, message)
Input: babysitter’s ID, message
Output: success/failure
Description: Sends a message to a babysitter

• bookBabysitter(bookingDetails)
Input: selected timeslot, booking type, children
Output: booking ID or error
Description: Creates and sends booking request

• cancelBooking(bookingID)
Input: babysitter’s ID
Output: success/failure
Description: Cancels upcoming booking

• viewConfirmedBookings()

Input: None
Output: list of active bookings
Description: Display all active bookings made by the parent

• **selectBookingToCancel(bookingID)**

Input: booking ID
Output: None
Description: Allows the parent to select a booking to cancel by specifying the booking ID

• **confirmCancellation(bookingID)**
Input: booking ID
Output: success/failure
Description: Confirms and completes the cancellation of the selected booking

• **enterPaymentDetails(details)**
Input: Payment details
Output: success/error message
Description: Submits the parent’s payment information for processing

• **rateBabysitter(babysitterID, review)**
Input: babysitter’s ID , review
Output: success/failure
Description: Allows the parent to rate and review the babysitter after a completed booking

• **getRecentViews()**
Input: None
Output: success/failure
Description: Retrieves a list of babysitters recently viewed by the parent

5.1.2.3 BabySitter (inherits User)

Attributes

• **nationalID:** Government-issued ID used for identity verification

• **experience:** List of children tied to the parent when a booking is made

• **languages:** Languages the babysitter speaks

• **certifications:** Relevant qualifications or training

• **availability:** List of available time slots

• **hourlyRate:** Babysitter's hourly charge

• **canHandleSpecialNeeds:** Can care for children with special needs

• **numOfChildrenThatCanBeHandled:** Max number of children accepted

• **bookingPreferences:** Type of bookings accepted (one-time, regular, both)

• **rating:** Average rating from parent reviews

• **reviews:** List of reviews written by parents

Methods

setAvailability(timeslot)
Input: timeslot
Output: none
Description: Adds a new time slot to the babysitter's list of available working hours

• **respondToBookingRequest(bookingID, response)**
Input: bookingID, response
Output: confirmation message
Description: Accepts or rejects a booking request. The response is either approval or denial

• **updateAvailabilityCalendar()**
Input: none
Output: updated availability list
Description: Updates the babysitter's calendar to reflect any new, cancelled, or modified bookings

- **viewPendingBookings()**

Input: none
Output: list of pending bookings
Description: Retrieves all booking requests that are waiting for the babysitter’s response

- **withdrawBookingRequest(bookingID)**

Input: bookingID
Output: status message
Description: Cancels an accepted booking before the session starts, if cancellation is still allowed

- **viewMessages()**

Input: none
Output: list of message threads
Description: Retrieves all messages received or sent between the babysitter and parents

- **replyToMessage(messageID)**

Input: messageID
Output: success status
Description: Sends a response to an existing message, maintaining the chat conversation

- **verifyNationalID(nationalID)**

Input: nationalID
Output: verification result (true/false)
Description: Validates the babysitter’s identity by checking the provided ID with the national identity system

- **receivePayment(amount)**

Input: amount
Output: confirmation receipt
Description: Confirms that the babysitter has received payment for a completed booking

5.1.2.4 Child

Attributes

- **name:** Child’s full name
- **age:** Child’s age in years
- **specialNeeds:** Indicates if the child has any special needs (true/false)
- **specialInstructions:** Any specific care instructions related to the child

Methods

- **setChildInfo(name, age, specialNeeds)**

Input: name, age, specialNeeds
Output: none
Description: Sets the child’s basic information including name, age, and whether they have special needs

- **updateChildInfo(UpdatedInfo)**

Input: UpdatedInfo
Output: none
Description: Updates the child’s stored information

- **getChildInfo()**

Input: none
Output: child information
Description: Retrieves the stored information of the child

5.1.2.5 Booking

Attributes

- **bookingID:** Unique identifier for each booking
- **parent:** The parent user who creates the booking request
- **babySitter:** The babysitter assigned to the booking
- **bookingType:** Type of booking (one-time or regular)
- **startTime:** Start Date and time of the booking
- **duration**: Duration of the babysitting session
- **numOfChildren:** Number of children included in the booking
- **children:** List of children involved in the booking
- **status:** Current state of the booking (pending, confirmed, canceled)
- **totalCost:** Total cost calculated for the booking Attributes

Methods

• createBooking(BookingInfo)
Input: details needed to create a new booking
Output: Booking confirmation
Description: Creates and saves a new booking with the provided information

• confirmBooking()
Input: None
Output: Confirmation message
Description: Confirms a pending booking, updating its status accordingly

• cancelBooking()
Input: None
Output: Cancellation message
Description: Cancels the booking and updates the booking status

• withdrawBookingRequest()
Input: None
Output: Withdrawal confirmation
Description: Allows the parent to withdraw a booking request before it is confirmed

• **isCancelable()**
Input: None
Output: Boolean (true or false).
Description: Checks if the booking is eligible for cancellation based on status or timing

• **updateBookingStatus(Status)**
Input: new status value.
Output: Updated status.
Description: Updates the booking's status to the specified new state

• **assignChildren(Children)**
Input: Children list of child objects.
Output: Confirmation message.
Description: Assigns a list of children to the current booking

• **calculateTotalCost()**
Input: None
Output: Total cost as a numeric value.
Description: Calculates the total cost of the booking based on hourly rate and duration

5.1.2.6 Payment

Attributes

- **paymentID:** Unique identifier assigned to each payment
- **booking:** The booking associated with this payment
- **amount:** Total amount to be paid for the booking
- **paymentMethod:** The method used for payment (e.g., credit card, bank transfer)
- **status:** The current status of the payment (e.g., pending, completed, refunded)
- **paymentTime:** The date and time when the payment was processed

Methods

- **processPayment(details)**

Input: Payment details
Output: confirmation message
Description: Processes the payment based on the provided payment details and updates the status accordingly

- **confirmPayment()**

Input: none
Output: confirmation message
Description: Confirms that the payment has been successfully completed

- **sendRefundRequest()**

Input: none
Output: refund request status
Description: Initiates a request to refund the payment amount

- **confirmRefund()**

Input: none
Output: refund confirmation
Description: Confirms that a refund has been processed and updates the payment status

- **generateInvoice(bookingID)**

Input: bookingID
Output: invoice document
Description: Generates a detailed invoice for the specified booking

5.1.2.7 Review

Attributes

- **reviewID:** Unique identifier for each review.
- **parent:** The parent who wrote the review.
- **babySitter:** The babysitter the review is about.
- **rating:** The numeric rating given by the parent.
- **comment:** The textual feedback written by the parent.
- **date:** The date the review was submitted

Methods

- **createReview(parentID, babysitterID, rating, comment)**

Input: parentID, babysitterID, rating, comment
Output: confirmation message
Description: Creates and saves a new review for a babysitter by a parent

- **editReview(reviewID)**

Input: reviewID
Output: updated review
Description: Edits the content or rating of an existing review using its ID

- **deleteReview(reviewID)**

Input: reviewID
Output: deletion confirmation
Description: Deletes an existing review based on the provided review

5.1.2.8 Message

Attributes

- **messageID:** Unique identifier for each message
- **sender:** The user who sent the message

- **receiver:** The user who received the message
- **content:** The actual text or content of the message
- **time:** The date and time when the message was sent

Methods

- **sendMessage(senderID, receiverID, content)**

Input: senderID, receiverID, content
Output: confirmation message
Description: Sends a message from one user to another and stores it in the system

- **replyToMessage(messageID)**

Input: messageID
Output: reply status
Description: Sends a reply to an existing message using the original message ID

- **viewMessageHistory(userID)**

Input: userID
Output: List of messages
Description: Retrieves all messages sent or received by the specified user

5.1.2.9 TimeSlot

Attributes

- **date**: The specific calendar day for the time slot
- **startTime**: The starting time of the slot
- **endTime**: The ending time of the slot

Methods

• setTime(date, start, end)
Input: date, start, end
Output: updated time slot
Description: Sets a new date and time range for the time slot

• updateTime(start, end)
Input: start, end
Output: updated time slot
Description: Updates the start and end time of an existing time slot

• isAvailable(date, start, end)
Input: date, start, end
Output: boolean value
Description: Checks if a time slot is available for the specified date and time range

• assignToBabysitter(babysitterID)
Input: babysitterID
Output: confirmation message
Description: Assigns the time slot to a babysitter using their ID

5.2 Architectural Diagram

5.2.1 Architectural Style

The Salma Babysitting Platform follows a Client-Server Architecture based on the Service-Oriented Architecture (SOA) model. This architecture is well-suited for applications that rely on integration with third-party services (such as Nafath for

identity verification) and deliver modular, user-centered services through a mobile or web interface.

System Layers:

1. Client Layer (Frontend):

Mobile application used by parents and babysitters. Users can browse profiles, book services, verify identity, and communicate with one another. The frontend communicates with the backend via secure RESTful APIs.

2. Service Layer (Backend):

This layer provides essential services such as booking logic, user authentication, profile management, notification handling, and secure communication. Some functionalities are supported by external government services (e.g., Nafath). The backend is responsible for coordinating between user actions and the logic flow of the platform.

Architectural Style Characteristics:

- .**Loose Coupling:** Each service operates independently and can be updated without affecting others.
- **Integration-ready:** Allows easy interaction with external services like Nafath.
- **Scalable:** Backend services can be deployed on scalable cloud infrastructure.
- **Lightweight:** No need to manage a heavy database, reducing overhead

Why this Architecture Fits Salma:

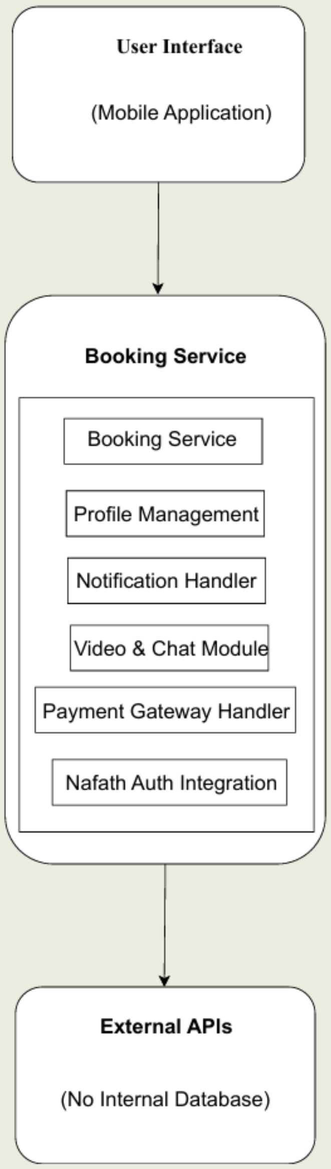
- Focuses on real-time interaction and service coordination.
- Leverages trusted third-party services instead of hosting sensitive data.
- Simplifies maintenance and deployment.

5.2.1 Identifying Subsystems

Here's a breakdown of the main logical components (subsystems) that make up

Salma's platform:

Updated Block Diagram:

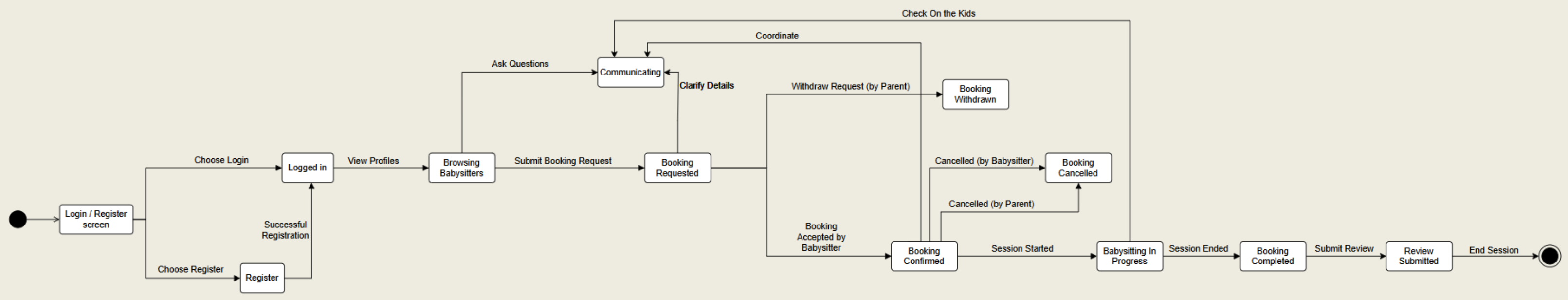


Subsystem Descriptions:

Subsystem	Description
User Interface	Mobile app allowing users to search, book, and communicate with babysitters.
Booking Service	Handles logic for creating and managing babysitting appointments.
Profile Management	Controls viewing/editing of babysitter information and user preferences.
Notification Handler	Sends real-time notifications for booking status, messages, etc.
Video & Chat Module	Enables direct communication between parents and babysitters.
Payment Gateway Handler	Coordinates payments using external financial APIs.
Nafath Auth Integration	Verifies user identity securely through the national identity system.

5.3 System Behavioral Diagram

5.3.1 State Diagram

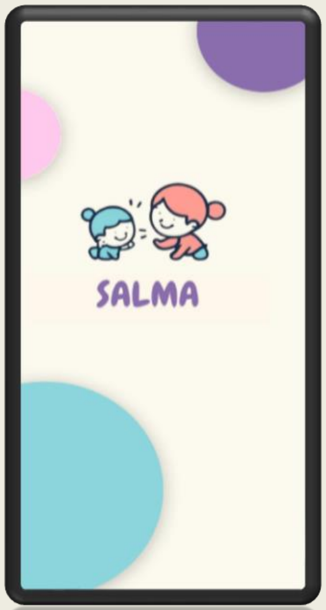


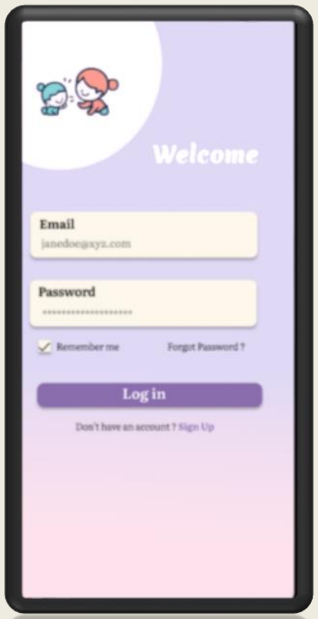
Design of Test

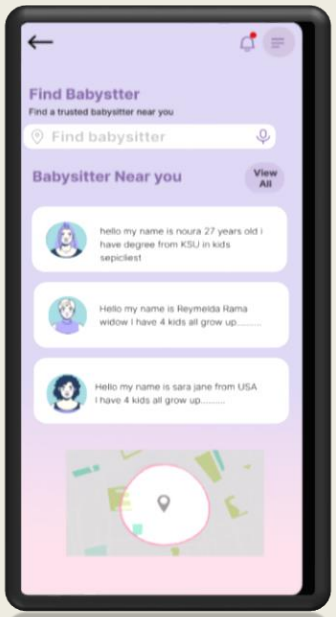
Test ID	Use Case Tested Description	Test Data / Input	Expected Output	When It's Considered Pass/Fail	Actual Output	Pass/Fail	Comment
TC01	UC-1: Login	Valid email + password	Redirect to homepage	If login successful	Homepage loaded	Pass	-
TC02	UC-1: Register (Parent)	Name, email, password	Account created	If account created & redirected	Redirected to dashboard	Pass	-
TC03	UC-1: Register (Babysitter)	Info + ID	Account pending verification	If status = pending shown	Status shown: Verification Pending	Pass	-
TC04	UC-2: Search Babysitter	Location: Riyadh, Skill: University degree in children care availability : 2-7 august	List of matching sitters	If results are correct	Displayed 2 babysitters	Pass	-
TC05	UC-3: View Babysitter Profile	Experience good with new born	Display profile info	If full profile is shown	Full profile loaded	Pass	Can chat with babysitter
TC06	UC-4: Communication	Start chat with sitter	Chat window opens	If interface loads	Chat interface opened	Pass	-
TC07	UC-5: Identity Verification	Upload legal ID	Verification submitted	If status shown to babysitter	ID submitted successfully	Pass	Related to nafth app
TC08	UC-7: Book Babysitter	Date: 20 Apr, Time :9-3 pm Child age: 4	Booking request sent	The sitter accept or reject booking	Booking request created	Pass if proved fail if rejected	-if it fail return to sitters page

Test ID	Use Case Tested Description	Test Data / Input	Expected Output	When It's Considered Pass/Fail	Actual Output	Pass/Fail	Comment
TC09	UC-8: Receive Payment	Job completed	Payment method	If sitter sees payment status	Sitter received payment	Pass	Tested with varying job lengths
TC10	UC-9: Payment Processing	Card/Apple Pay info	Payment successful	If confirmation shown	Payment processed successfully	Pass	Tested with both card and Apple Pay
TC11	UC-10: Rate & Review Babysitter	Rating from1-5, review text	Review saved	If appears on profile	Review visible on profile	Pass	-
TC12	UC-11: Manage Schedule	Update availability	Calendar updated	If new availability saved	Calendar updated	Pass	-
TC13	UC-12: Manage Bookings (Accept)	Accept request	Booking confirmed	If status = confirmed	Booking confirmed	Pass	-
TC14	UC-12: Manage Bookings (Reject)	Reject request	Booking rejected	If status = rejected	Booking rejected	Pass	-
TC15	UC-13: Cancel Booking	Cancel close to date	Refund policy applies	If correct policy shown	Cancellation fee applied	Pass	Policy applied correctly

UI Design



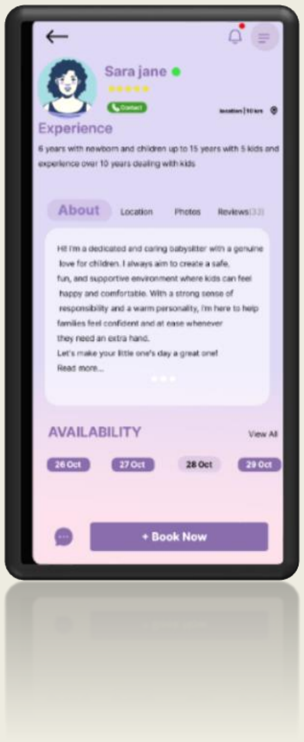




hello my name is noura 27 years old i have degree from KSU in kids sepicliest

Hello my name is Reymelda Rama widow I have 4 kids all grow up.........

Hello my name is sara jane from USA I have 4 kids all grow up.........



6 years with newborn and children up to 15 years with 5 kids and experience over 10 years dealing with kids

Hi! I'm a dedicated and caring babysitter with a genuine love for children. I always aim to create a safe, fun, and supportive environment where kids can feel happy and comfortable. With a strong sense of responsibility and a warm personality, I'm here to help families feel confident and at ease whenever they need an extra hand.
Let's make your little one's day a great one!
Read more...

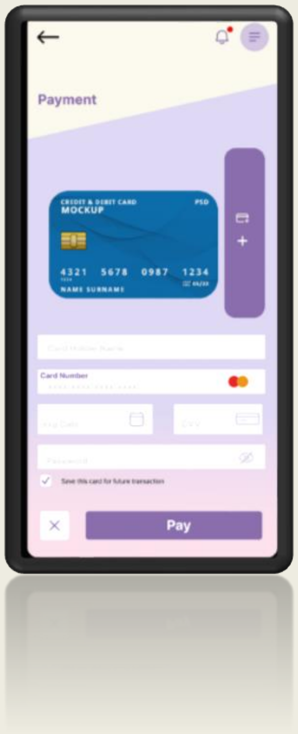


Table of Contents

Functional Requirements