Cover Page

Barbershop Appointment Management System

---

https://github.com/SBelagode/BarbershopAppointmentManagementSystem

Group#
Sohum Belagode
Srikar Vempaty

May 10, 2025

Table of Work

(Please write x in the boxes to mention what each student achieved in this project)

	Student-1 Sohum	Student-2 Srikar	N/A
Project Description	Sohum and Srikar worked together to describe the project.	Sohum and Srikar worked together to describe the project.	
Uses Cases Diagram(s)	Sohum checked the use cases.	Srikar designed the use cases as both students discussed what they would be.	
Sequence Diagrams	Sohum helped clarify working of code for sequence diagrams.	Srikar was responsible for ensuring the sequence diagrams matched the code.	
Class diagram(s)	Sohum helped clarify design of classes for class diagrams.	Srikar was responsible for ensuring the class diagrams matched the code.	
Implementation	Sohum was mainly responsible for coding the project.	Srikar helped fix some bugs and helped code some of the clientGUI code.	
Conclusion	Sohum and Srikar wrote this together.	Sohum and Srikar wrote this together.	

Table of Contents

Project Description

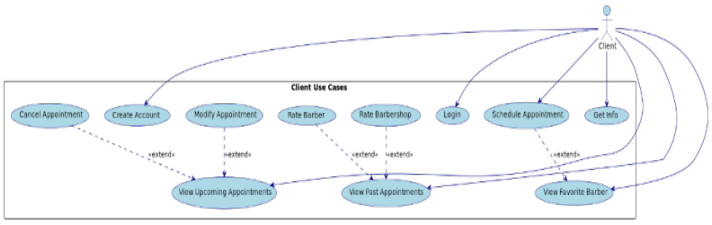
The Barbershop Appointment Management System is a GUI-based application that allows clients and barbers to efficiently manage appointments, profiles, and scheduling. The system supports two personas: Clients and Barbers, providing distinct features tailored to their respective needs. It enables streamlined booking, rating, and scheduling, while allowing barbers to manage their availability and track client engagement. Clients can create accounts, get info about the barbershop, schedule appointments, view upcoming appointments, modify appointments, delete appointment, view favorite barbers, view past appointments, rate appointments, rate the barbershop, and delete their account. Barbers can create profiles, edit availability, view appointments, cancel appointments, mark appointment complete, delete accounts, and view their ratings

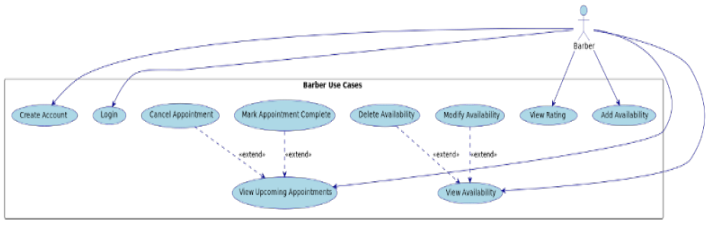
The barbershop management system operates under several key assumptions that shape its design and functionality. First and foremost, the system operates entirely in-memory with no persistent storage implementation, as confirmed by the teaching assistant (Arsalan). This means all data is lost upon program termination, and no database or file system integration is required. Additionally, the system assumes all user inputs are valid and correctly formatted, implementing minimal error handling as these requirements were not explicitly stated in the initial project proposal. The system starts with no pre-existing data, requiring all accounts, appointments, and reviews to be created during runtime, with no initial data seeding necessary. The interface is graphical user interface implementation, and all interactions are text-based without special formatting requirements. The system assumes single-user access at a time, with no concurrent user handling, session management, or timeout mechanisms required.

The system's workflow follows a sequential process that begins with barber account creation and availability setup. A typical system walkthrough starts with a barber creating an account, signing in, and establishing their availability slots. This process can be repeated for multiple barbers, each creating their own account and setting up their individual availability schedules. Once the barbers are set up, a client can create their account, log in, and proceed to schedule availabilities. The client can log out after scheduling, and a barber can then log in to manage these appointments. Barbers have the capability to cancel appointments, modify their availability slots, and mark appointments as completed. When a client logs back in, they can view their past appointments and engage with the rating system. The rating system allows clients to rate the barbershop as a whole or rate individual barbers for specific appointments.

Use Case Diagram

Note because of the technology we used and because of our larger project size, we have cleared by the TA Arsalan, that it is okay we include the use case diagram and class diagram in a separate .svg file for clear view that is within the compressed submission and also in the github link. Please refer to those for clearer views. We have included them as best we could here, but to clearly access these diagrams (which didn't fit into a pdf), please refer to the svgs.





Use Case Descriptions:

Use Case Descriptions:

UC Reference Name/Number	UC-01 Client Create Account
Overview	A client registers by entering personal information, login, and payment details.
Related Use Cases	N/A
Actors	Client

UC Reference Name/Number	UC-02 Client Login
Overview	Client supplies username + password; if verified, the system opens Client Home.
Related Use Cases	N/A
Actors	Client

UC Reference Name/Number	UC-03 Schedule Appointment
Overview	Client chooses a barber, views available slots, and books an appointment.
Related Use Cases	UC-11 (is extended by UC-03)
Actors	Client

UC Reference Name/Number	UC-04 View Upcoming Appointments
Overview	Lists all future appointments for the client.
Related Use Cases	UC-05 (extends UC-04), UC 10 (extends UC-04)
Actors	Client

UC Reference Name/Number	UC-05 Cancel Appointment
Overview	Client selects an upcoming appointment and deletes it on both client and barber accounts.
Related Use Cases	UC-04 (is extended by UC-05)
Actors	Client

UC Reference Name/Number	UC-06 Rate Barber
Overview	After a past appointment the client leaves a 1-5 rating and comments for that barber.
Related Use Cases	UC-09 (is extended by UC-06).
Actors	Client

UC Reference Name/Number	UC-07 Rate Barbershop
Overview	Client submits an overall rating/comment for the barbershop.
Related Use Cases	UC-09 (is extended by UC-07)
Actors	Client

UC Reference Name/Number	UC-08 Get Info (Barbershop)
Overview	Shows barbershop address, phone, email, average rating, and all reviews.
Related Use Cases	N/A
Actors	Client

UC Reference Name/Number	UC-09 View Past Appointments
Overview	Displays completed appointments; client may review barbers, review shop, or favorite a barber.
Related Use Cases	UC-06 (extends UC-09), UC-07 (extends UC-09)
Actors	Client

UC Reference Name/Number	UC-10 Modify Appointment
Overview	Clients can reschedule an upcoming appointment by picking a new slot.
Related Use Cases	UC-04 (is extended by UC-10)
Actors	Client

UC Reference Name/Number	UC-11 View Favorite Barbers
Overview	Lists client's favorite barbers and lets the client book directly with one if they want to.
Related Use Cases	UC-03 (extends UC-11)
Actors	Client

UC Reference Name/Number	UC-12 Barber Create Account
Overview	Prospective barber registers with profile and specialization details.
Related Use Cases	N/A
Actors	Barber

UC Reference Name/Number	UC-13 Barber Login
Overview	Barber enters credentials; on success the system opens Barber Home.
Related Use Cases	N/A
Actors	Barber

UC Reference Name/Number	UC-14 View Upcoming Appointments
Overview	Barber sees all scheduled future client appointments.
Related Use Cases	UC-18 (extends UC-14), UC-19 (extends UC-14)
Actors	Barber

UC Reference Name/Number	UC-15 View Availability
Overview	Displays all availability slots posted by barber.
Related Use Cases	UC-16 (extends UC-15), UC-17 (extends UC-15)
Actors	Barber

UC Reference Name/Number	UC-16 Modify Availability
Overview	Barber edits the details of a new selected availability slot to replace the selected slot.
Related Use Cases	UC-15 (is extended by UC-16)
Actors	Barber

UC Reference Name/Number	UC-17 Delete Availability
Overview	Barber removes an availability slot from their schedule.
Related Use Cases	UC-15 (is extended by UC-17)
Actors	Barber

UC Reference Name/Number	UC-18 Mark Appointment Complete
Overview	Barber marks an appointment as completed (client's copy moves to past list).
Related Use Cases	UC-14 (is extended by UC-18)
Actors	Barber

UC Reference Name/Number	UC-19 Cancel Appointment
Overview	Barber cancels a future appointment, removing it from both barber and client lists.
Related Use Cases	UC-14 (is extended by UC-1()
Actors	Barber

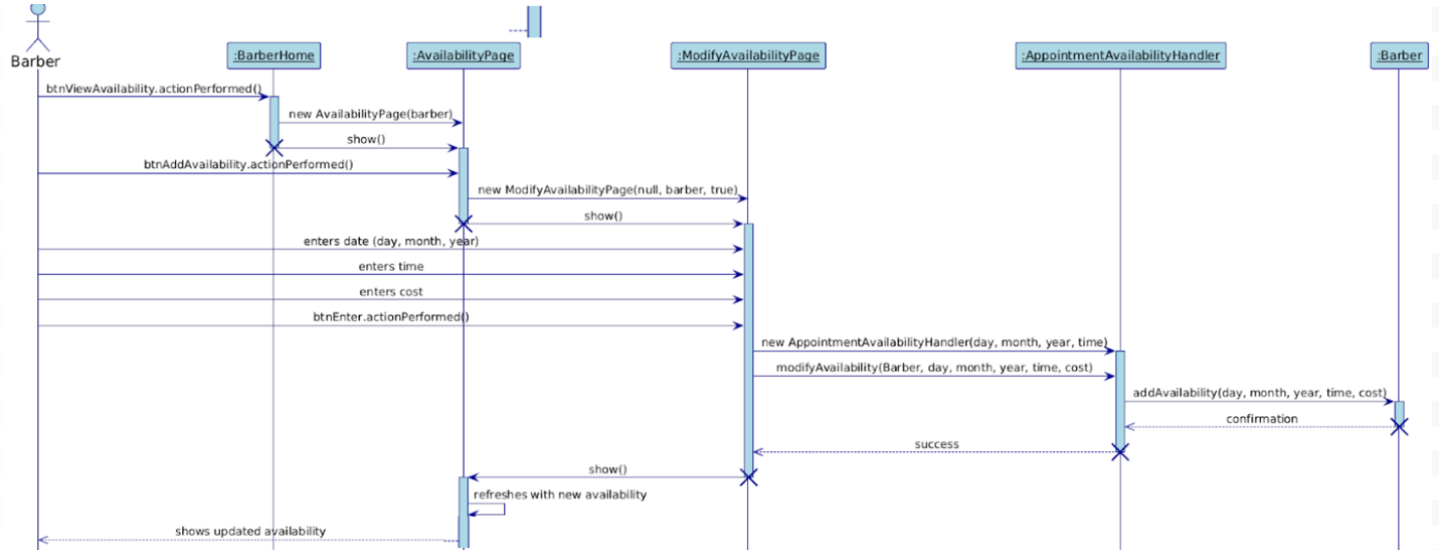
UC Reference Name/Number	UC-20 View Rating
Overview	Barber views their average star rating and all client reviews.
Related Use Cases	N/A
Actors	Barber

UC Reference Name/Number	UC-21 Add Availability
Overview	Barber adds a new availability slot for a potential appointment.
Related Use Cases	N/A
Actors	Barber

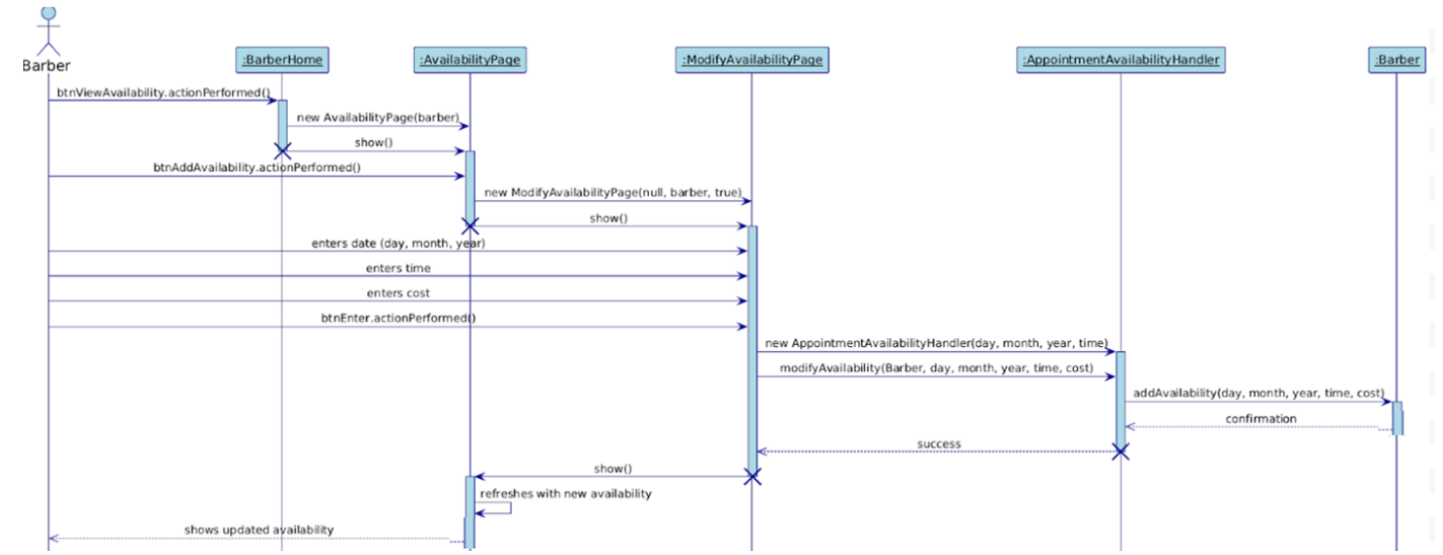
Sequence diagrams

Some minor rendering issues with activation boxes, please ignore those slight discrepancies.

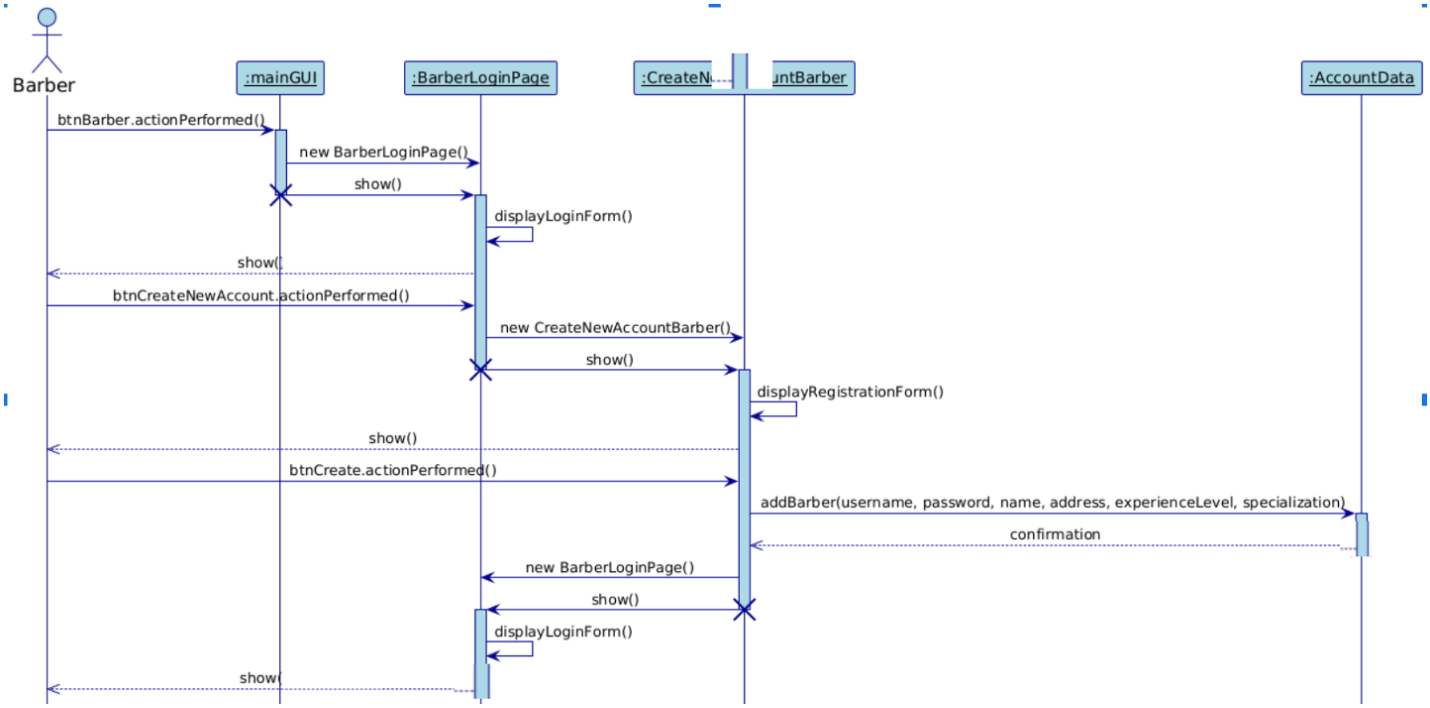
Barber Add Availability Sequence Diagram:



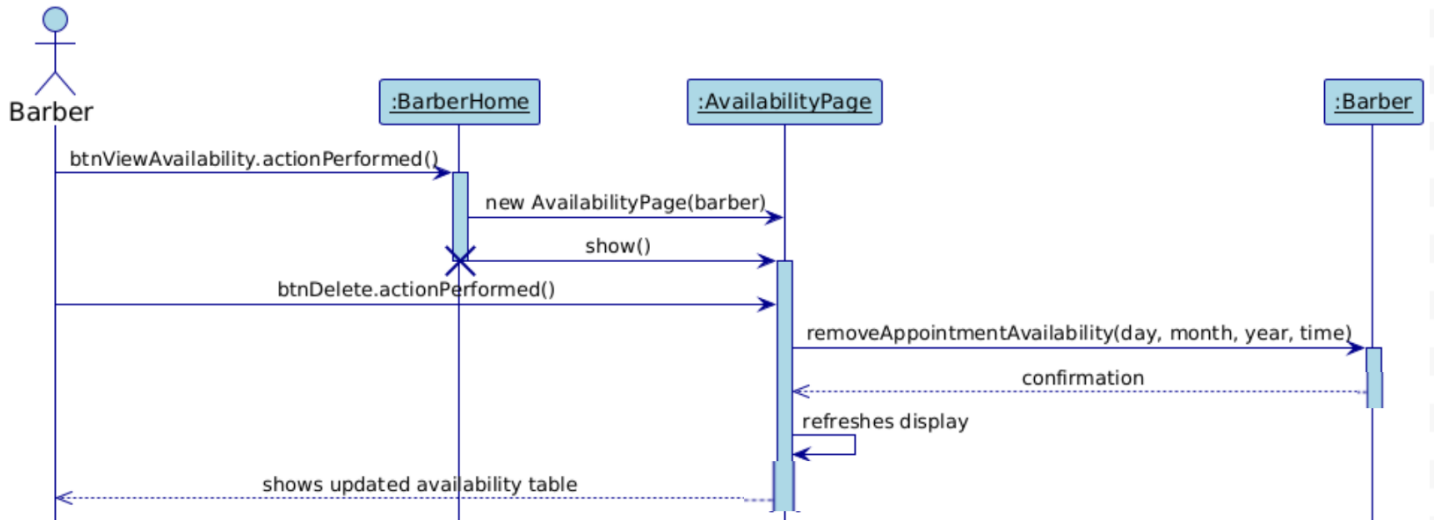
Barber Cancel Appointment Sequence Diagram:



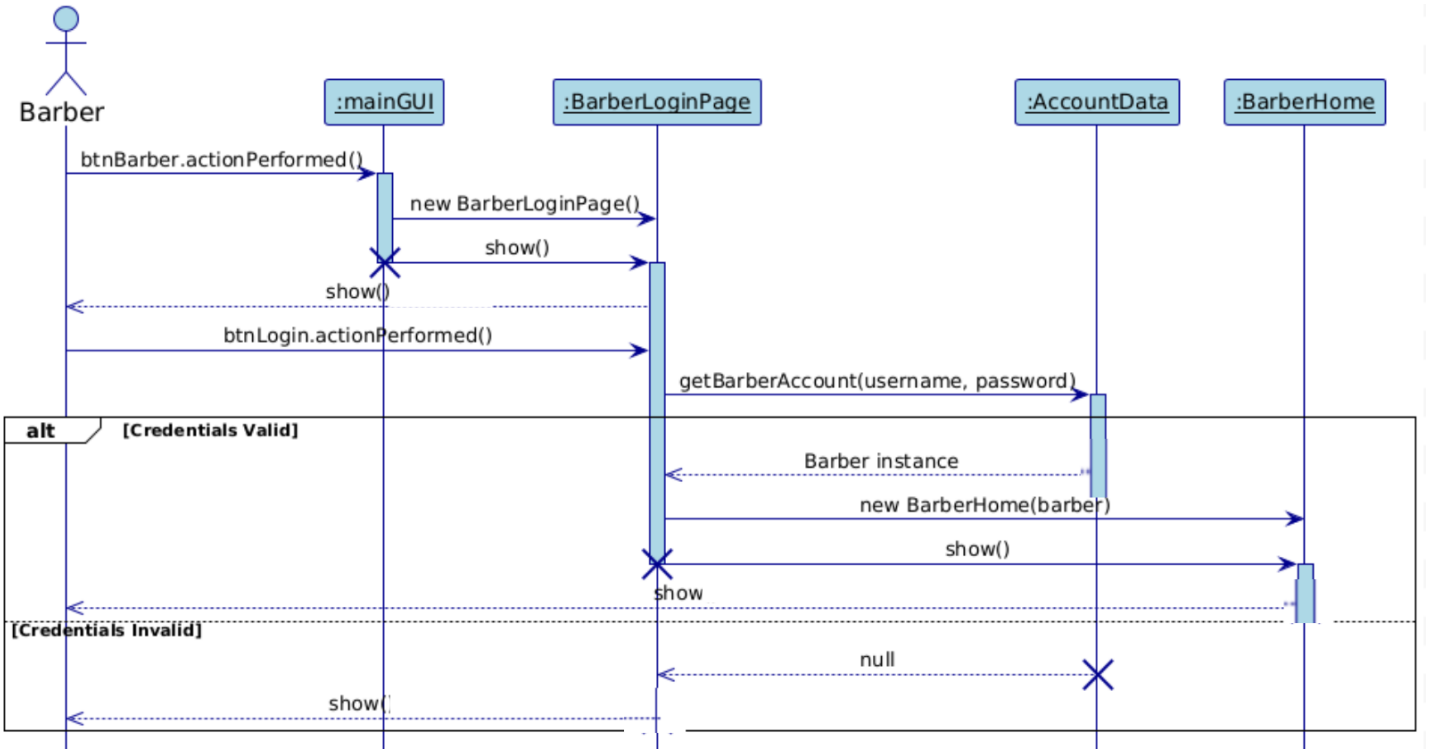
Barber Create Account Sequence Diagram:



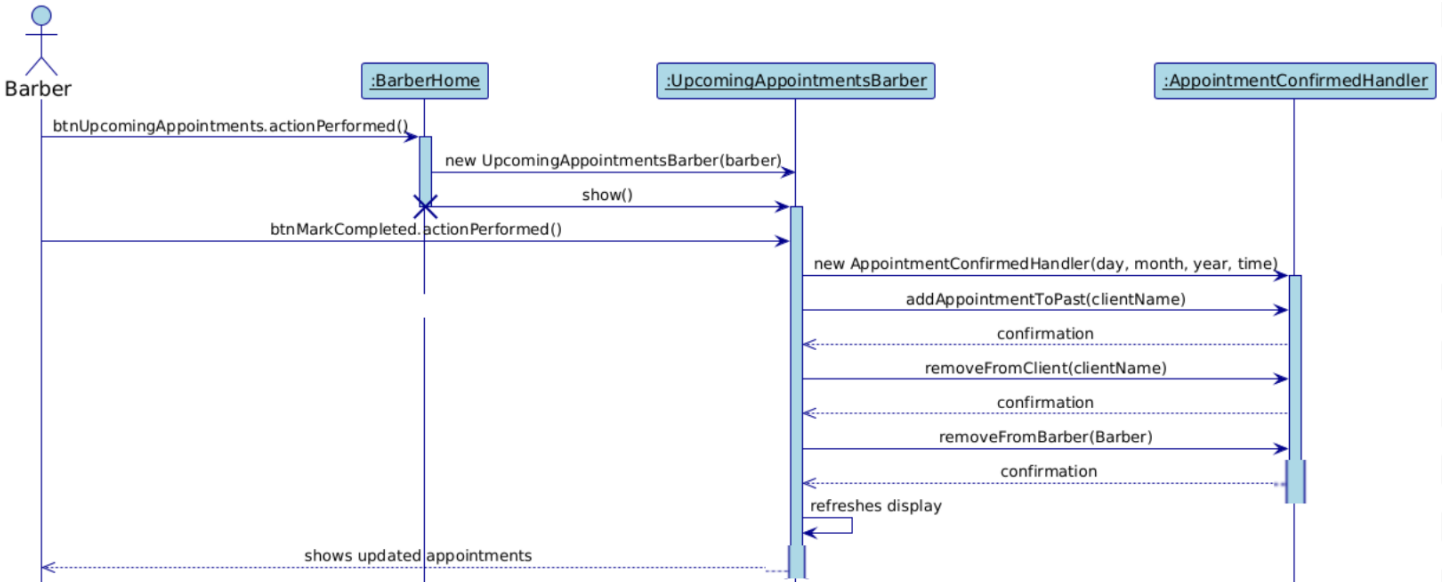
Barber Delete Availability Sequence Diagram:



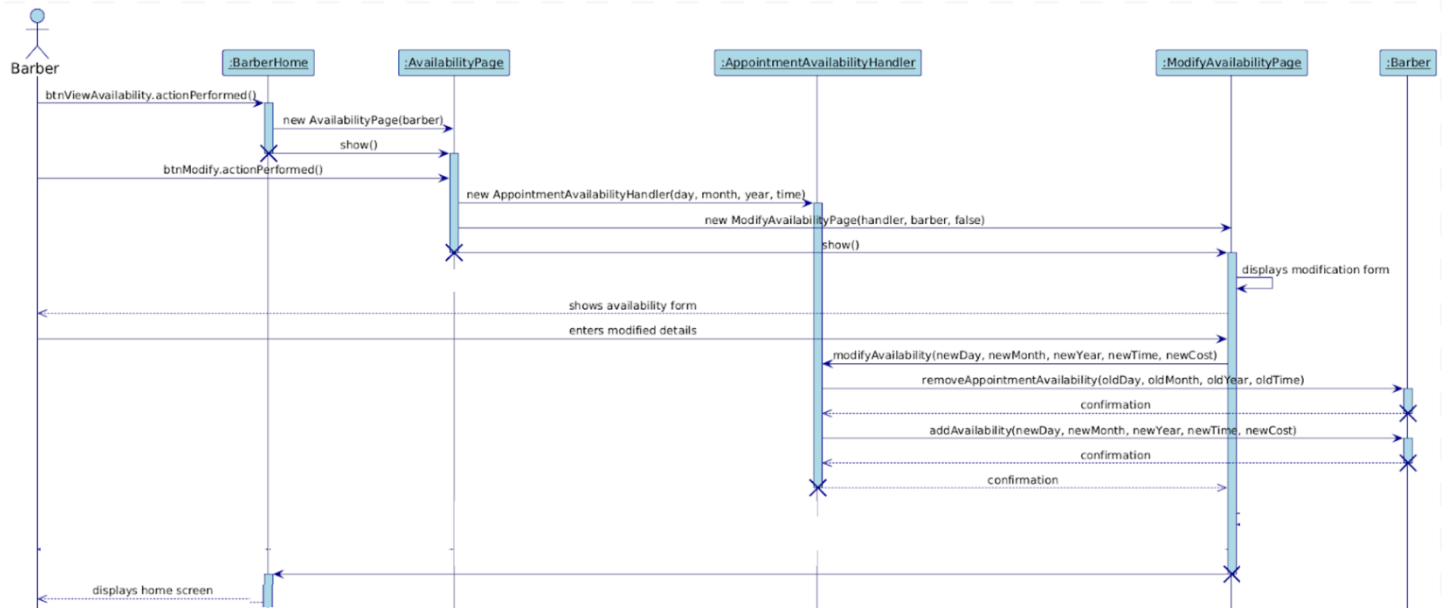
Barber Login Sequence Diagram:



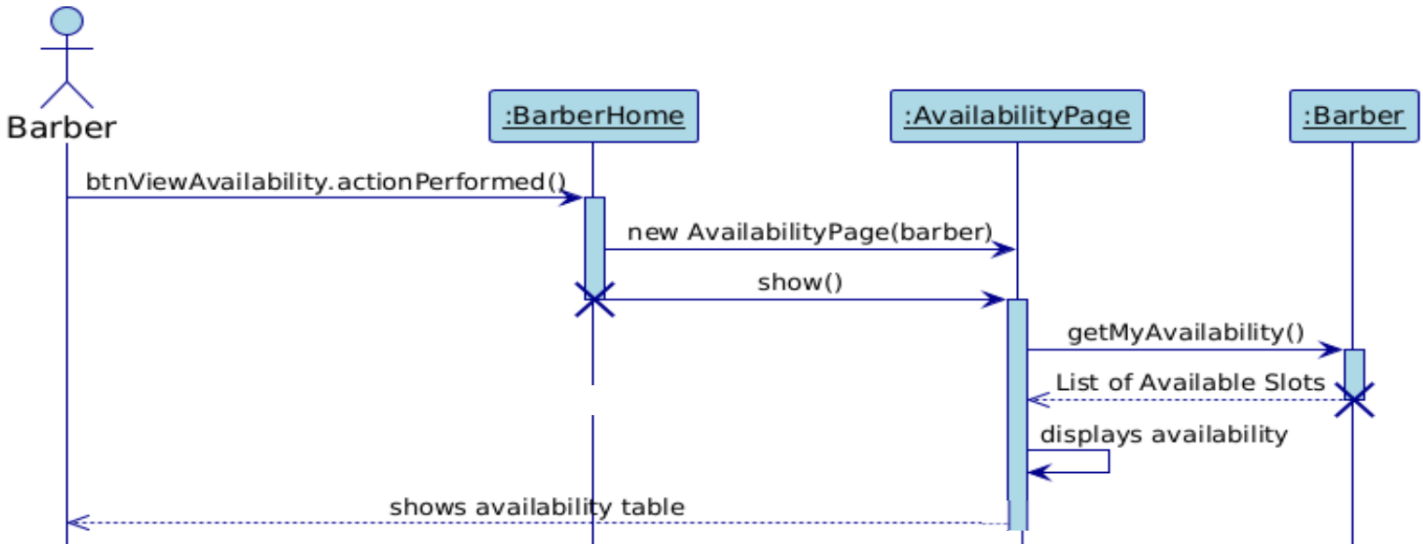
Barber Mark Appointment Complete Sequence Diagram:



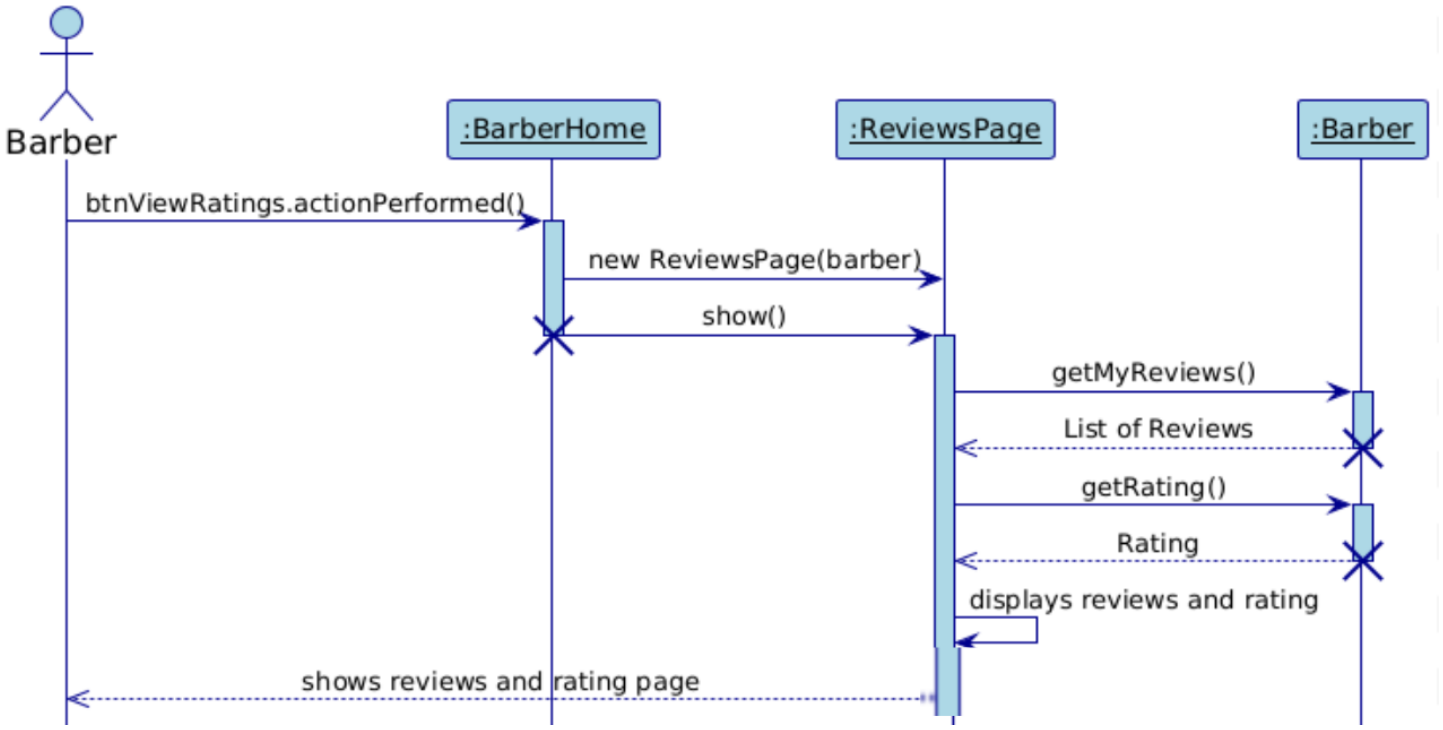
Barber Modify Availability Sequence Diagram:



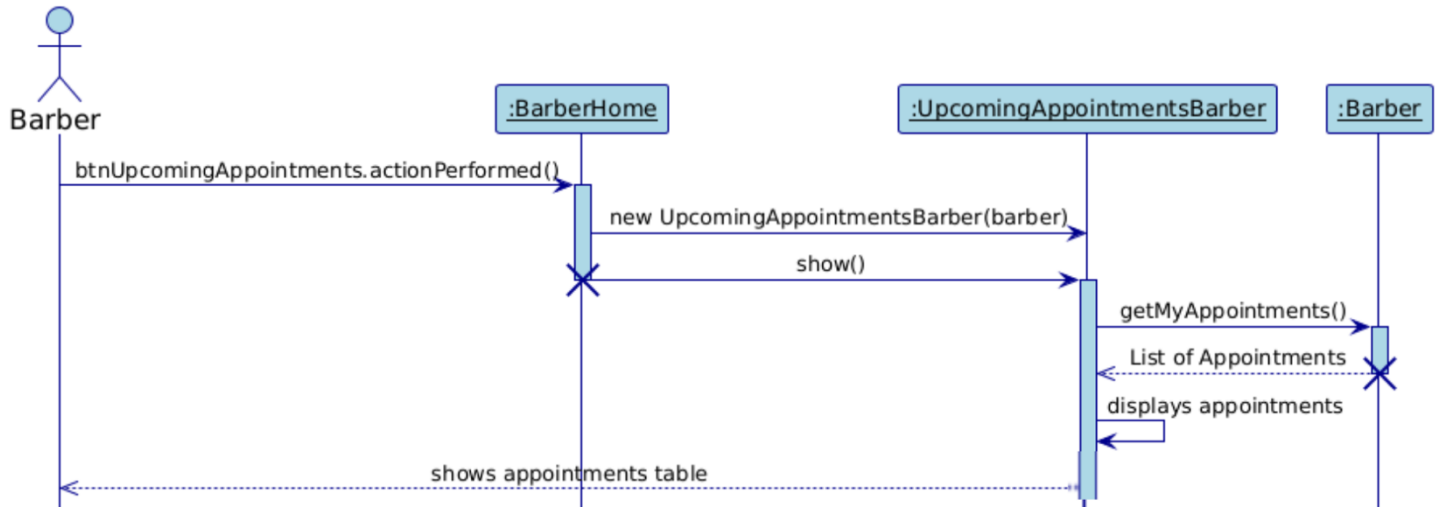
Barber View Availability Sequence Diagram:



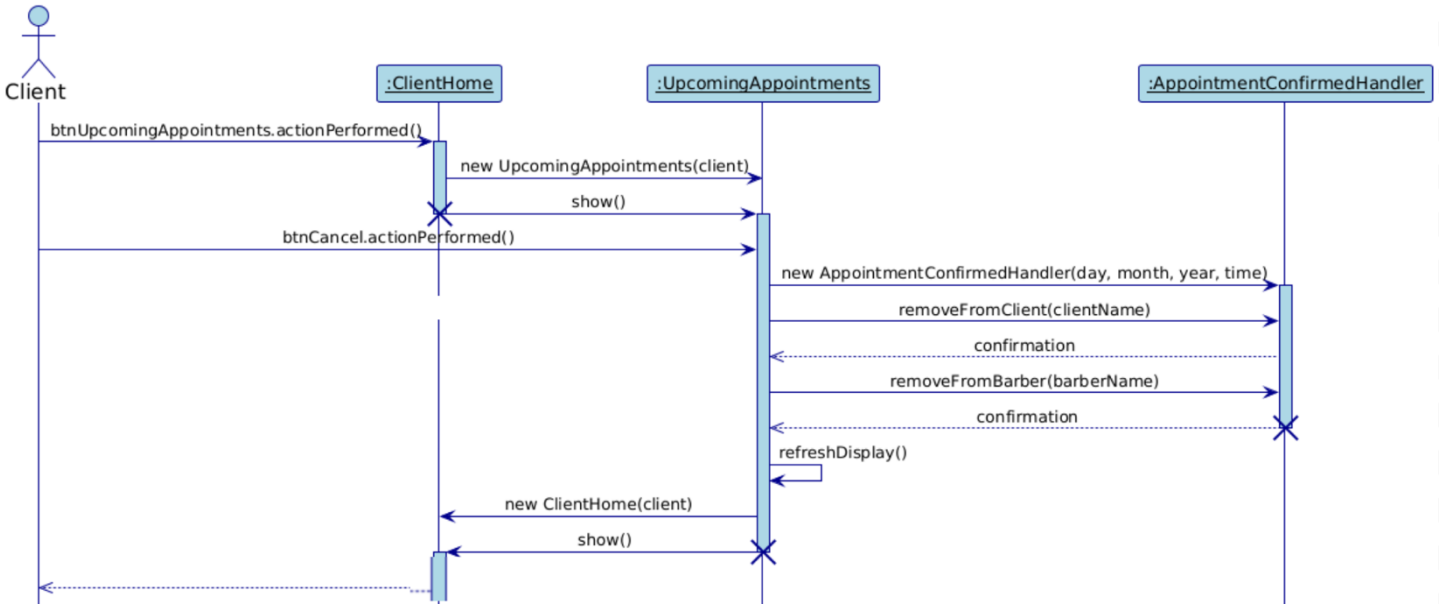
Barber View Rating Sequence Diagram:



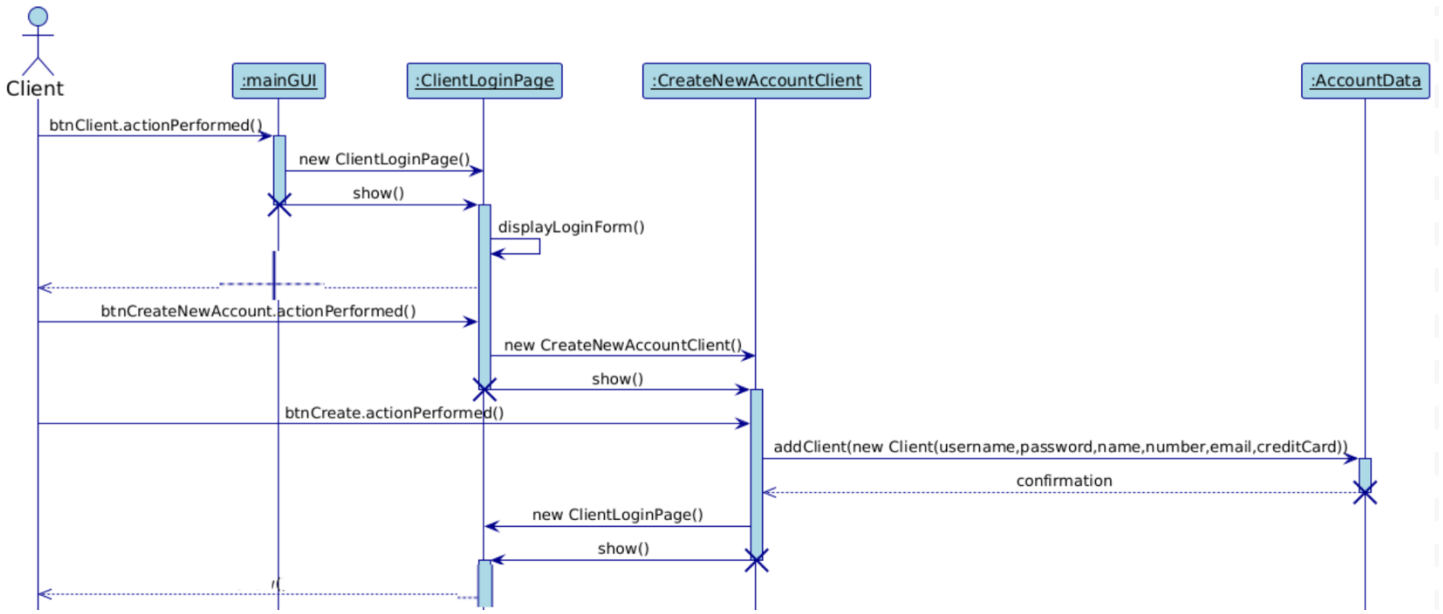
Barber View Upcoming Appointments Sequence Diagram:



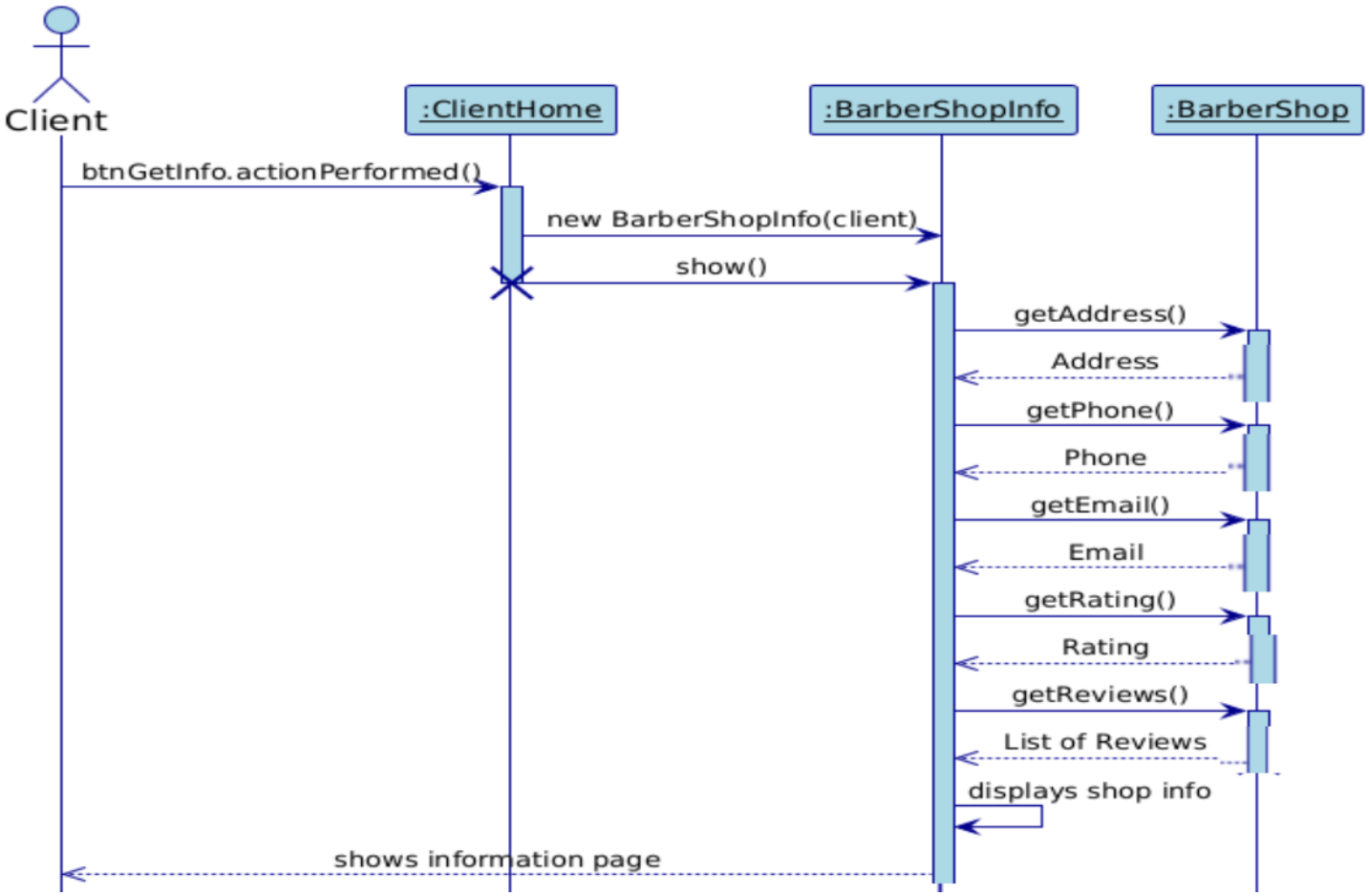
Client Cancel Appointment Sequence Diagram:



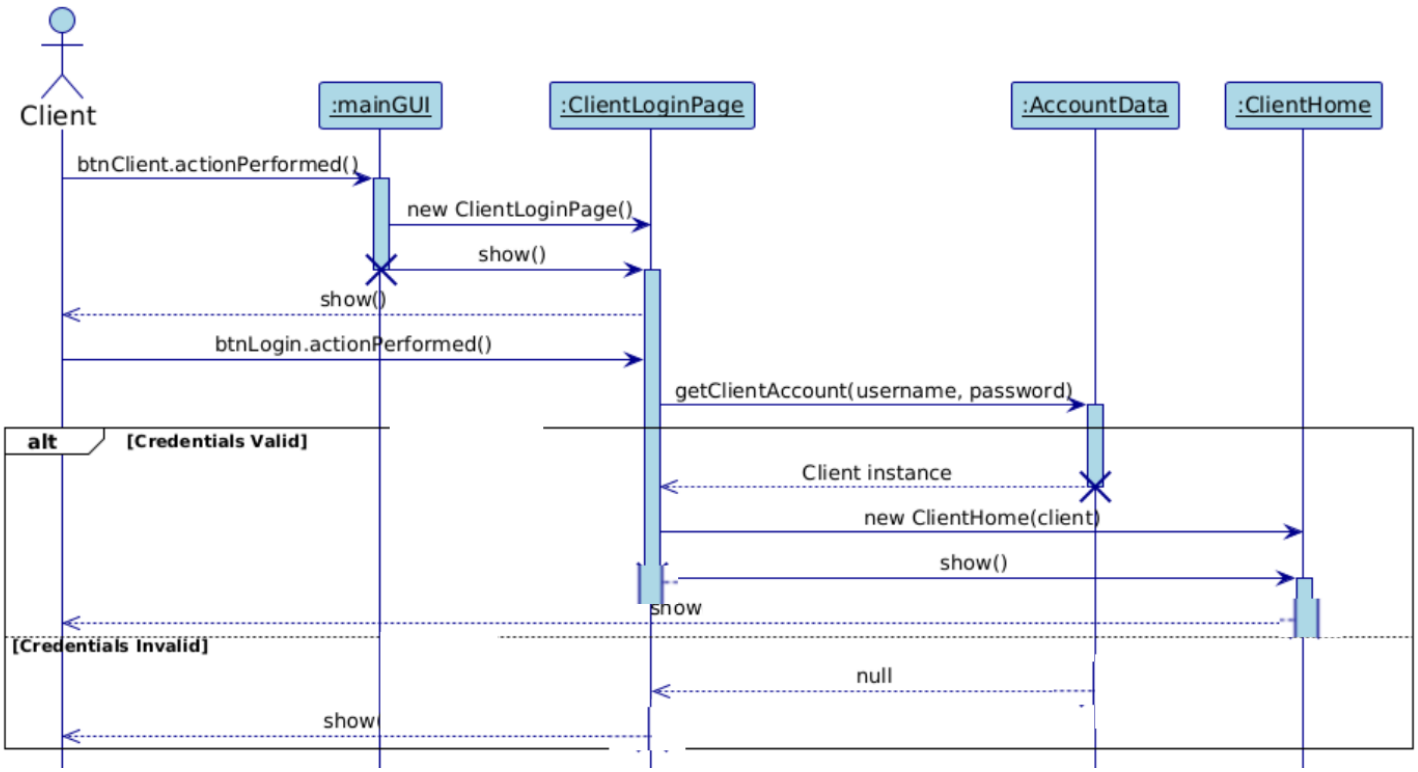
Client Create Account Sequence Diagram:



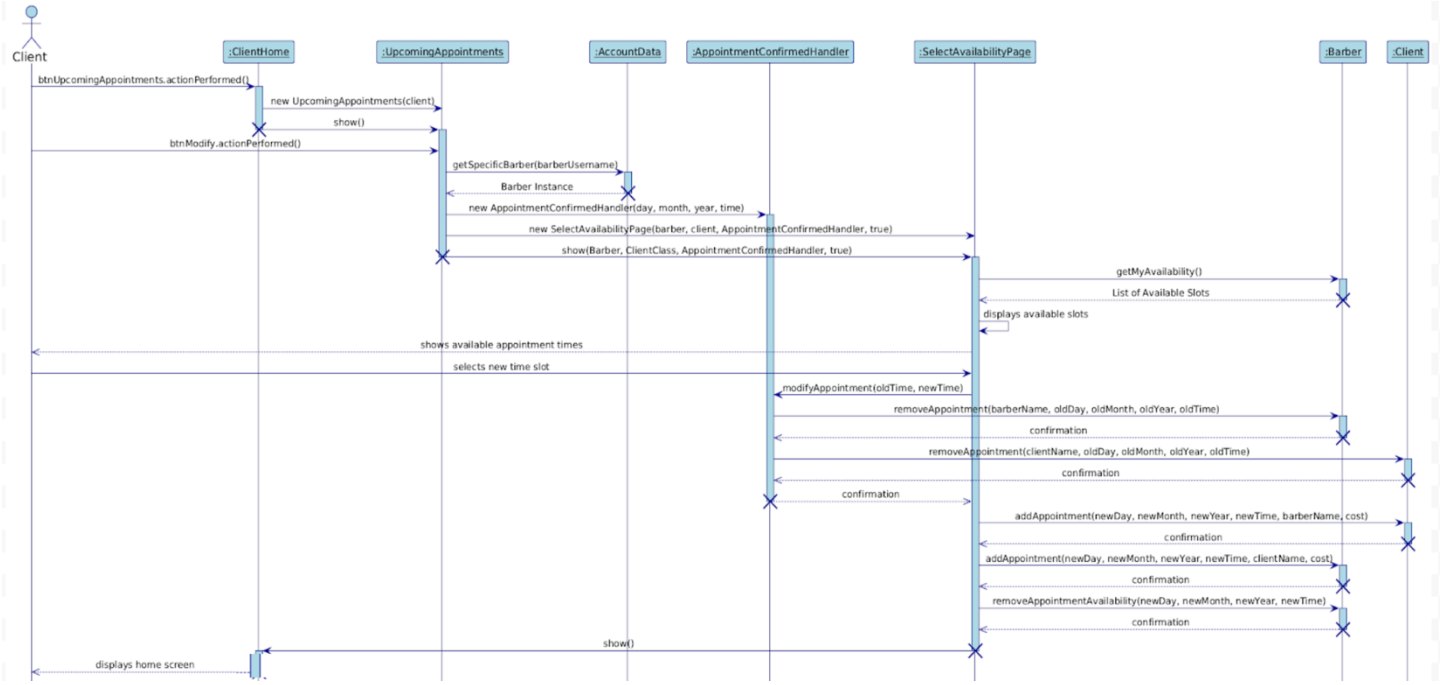
Client Get Information Sequence Diagram:



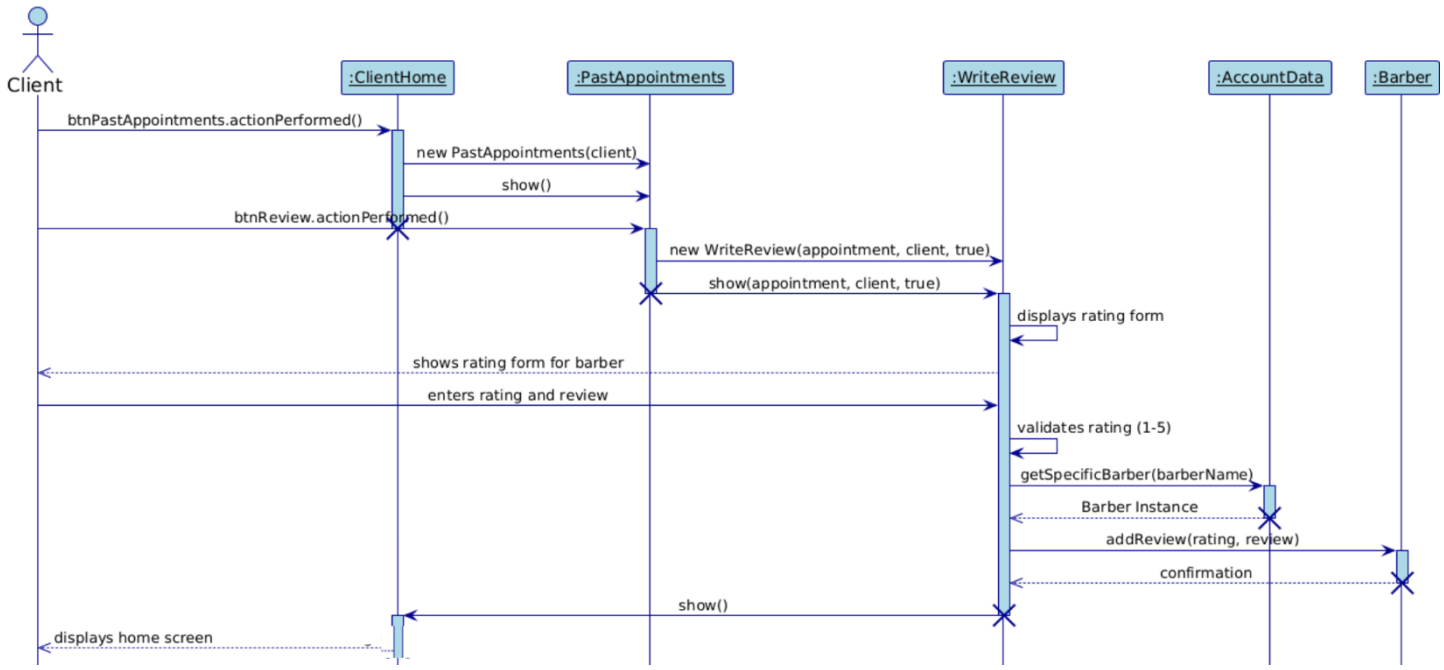
Client Login Sequence Diagram:



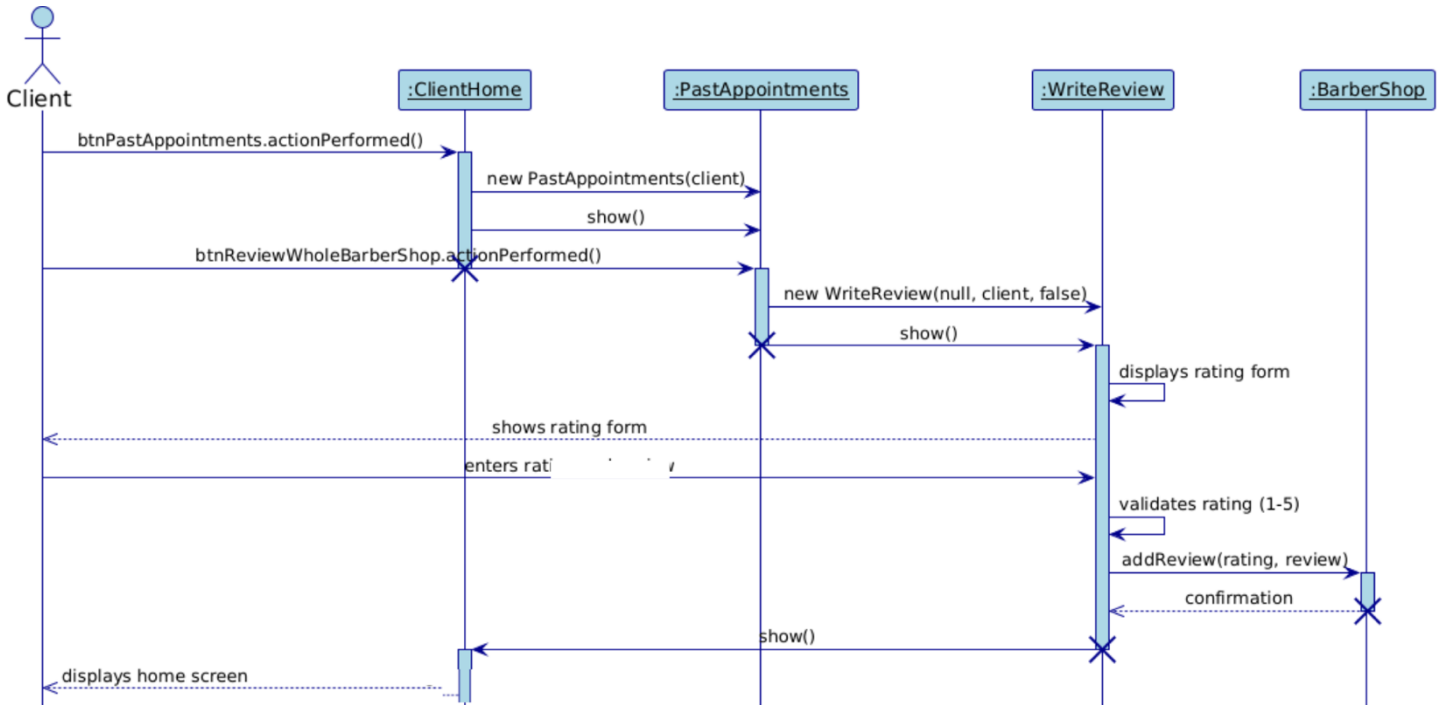
Client Modify Appointment Sequence Diagram:



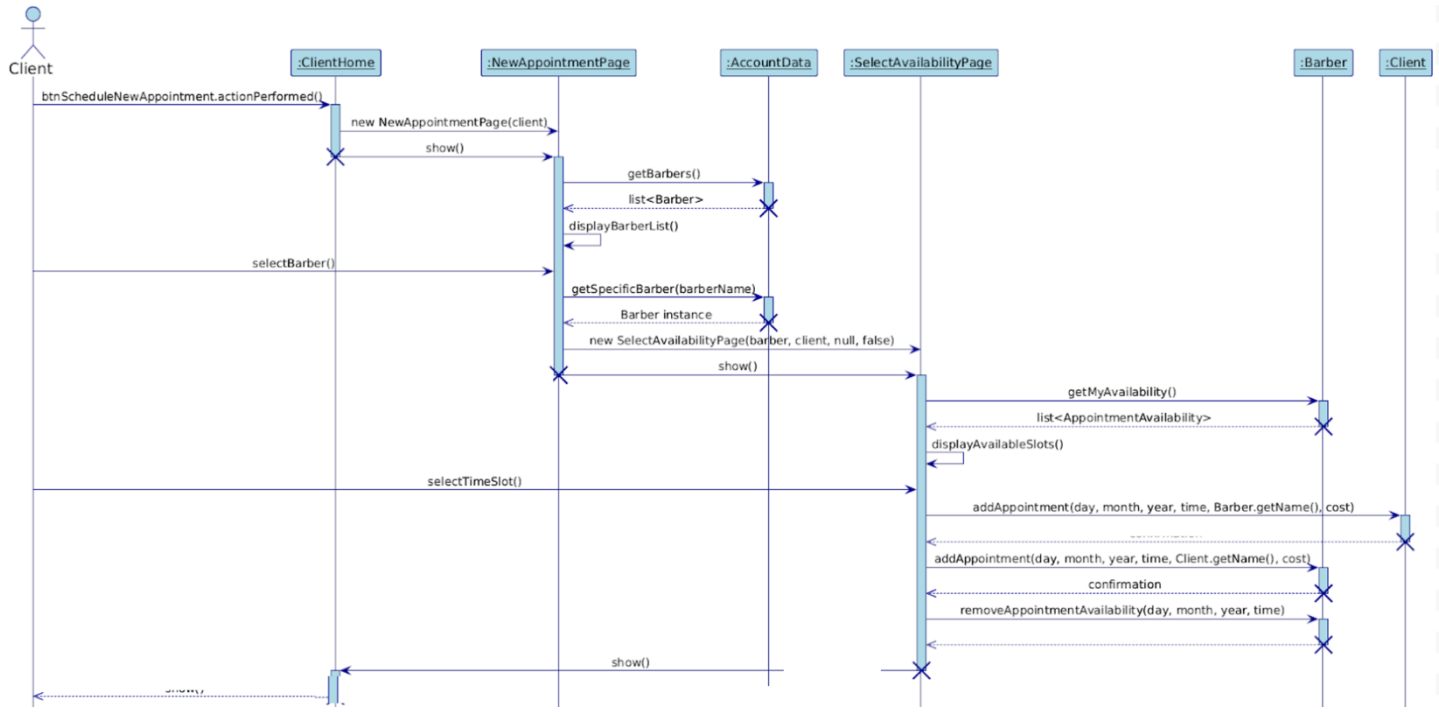
Client Rate Barber Sequence Diagram:



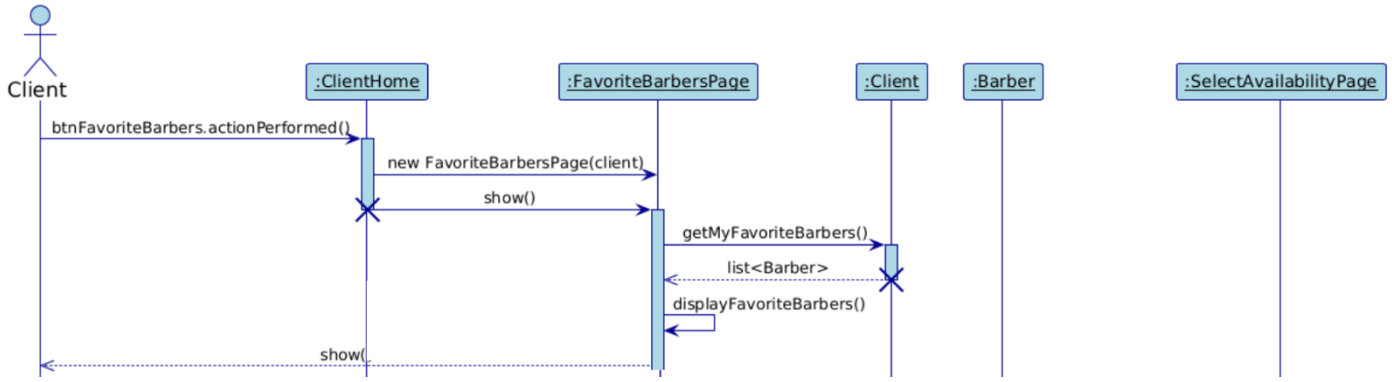
Client Rate Barbershop Sequence Diagram:



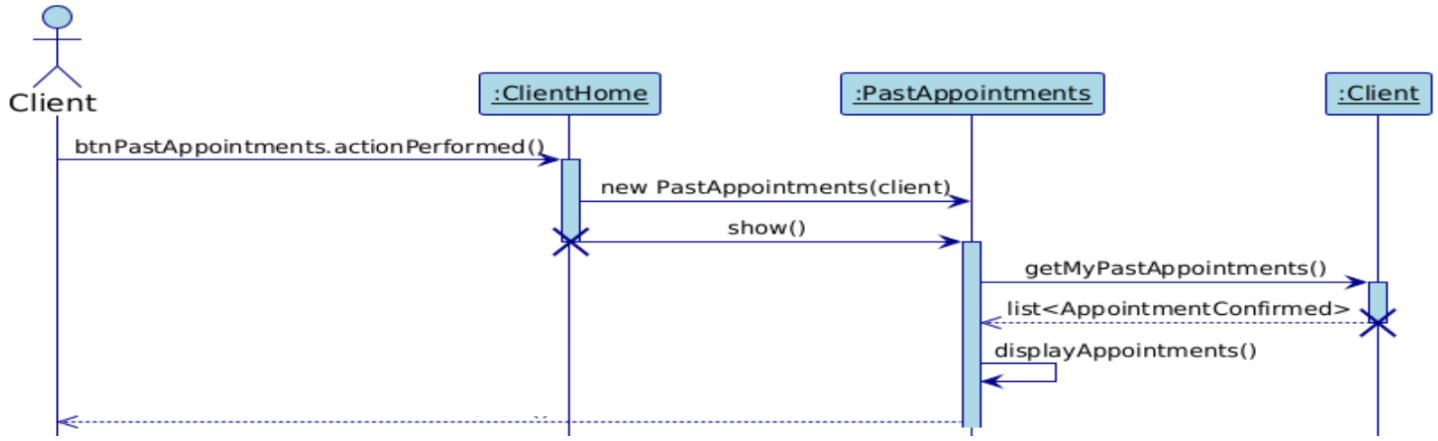
Client Schedule Appointment Sequence Diagram:



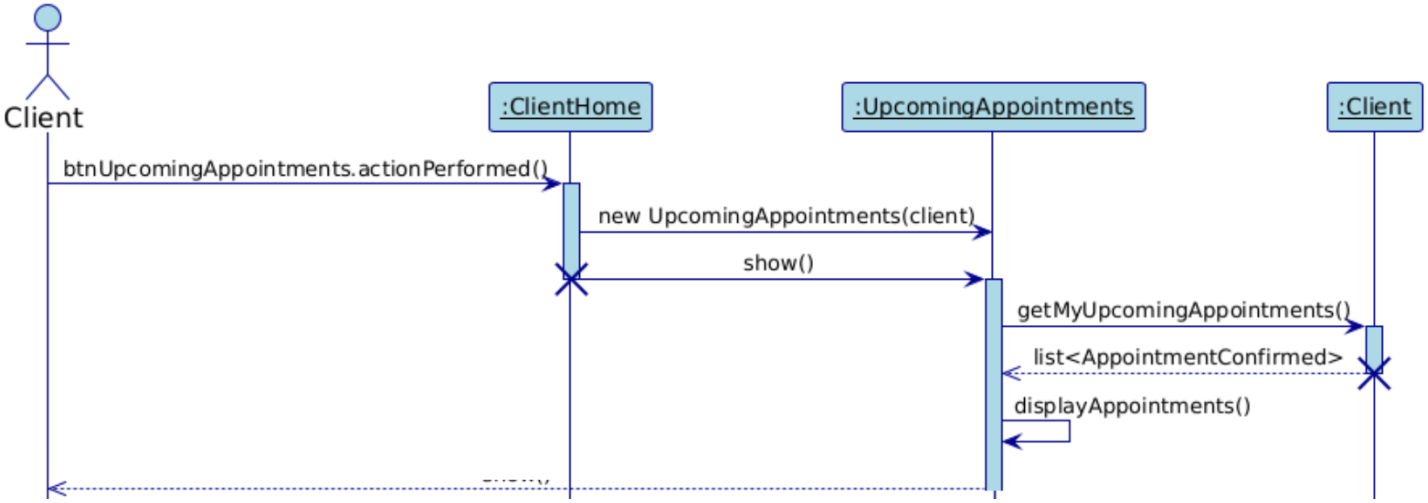
Client View Favorite Barbers Sequence Diagram:



Client View Past Appointments Sequence Diagram:

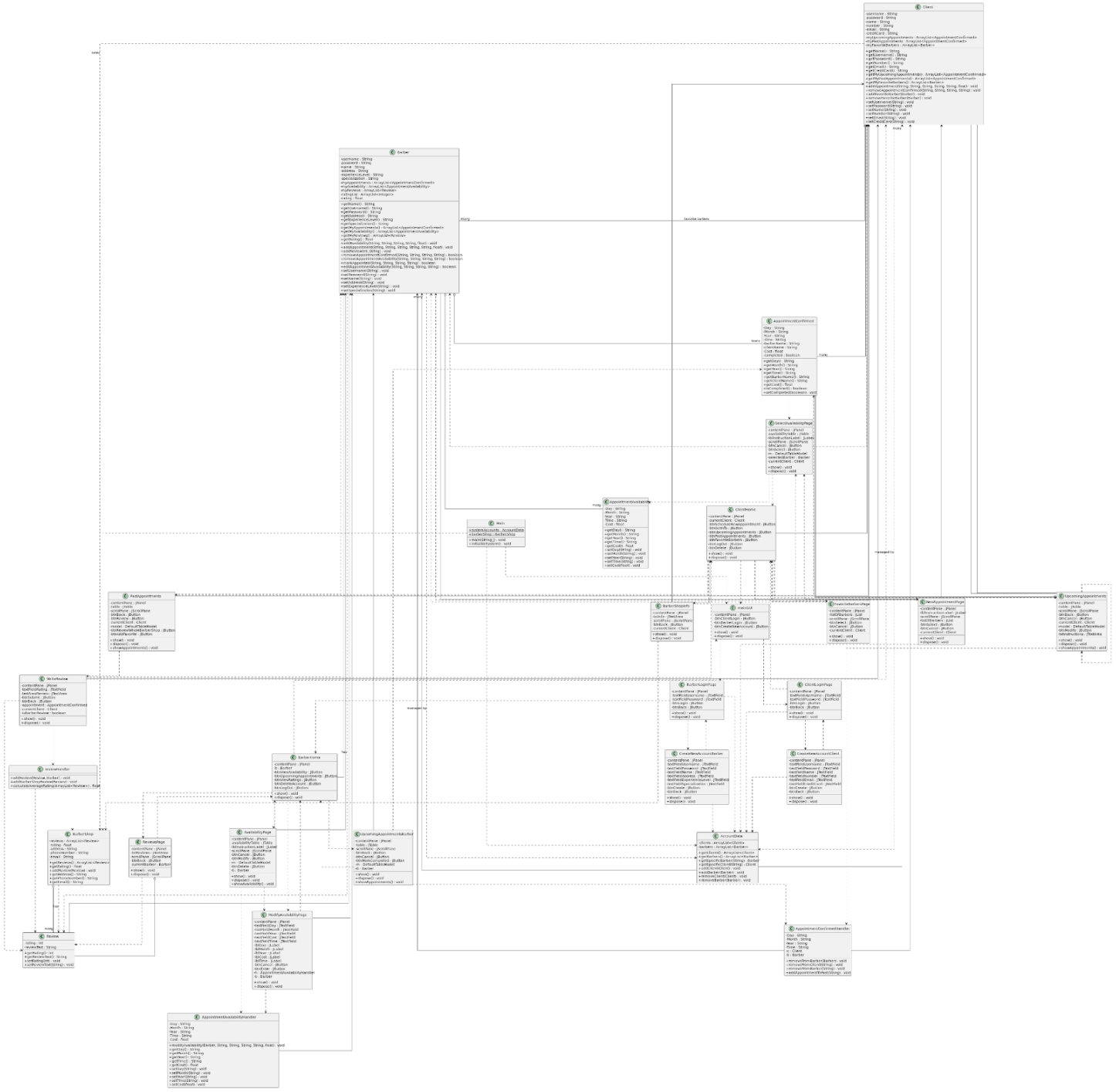


Client View Upcoming Appointments Sequence Diagram:



Class Diagrams

Note because of the technology we used and because of our larger project size, we have cleared by the TA Arsalan, that it is okay we include the use case diagram and class diagram in a separate .svg file for clear view that is within the compressed submission and also in the github link. Please refer to those for clearer views. We have included them as best we could here, but to clearly access these diagrams (which didn't fit into a pdf), please refer to the svgs.



Conclusion

Through this project, we were able to learn about the utility of the SOLID principles when coding/designing the project, as they really helped us when we needed to repeatedly reconfigure aspects of our project. The system's architecture is built upon the SOLID principles of object-oriented design, ensuring a robust and maintainable codebase. The Single Responsibility Principle is implemented through the separation of distinct functionalities into different classes, such as `AccountData` handling user authentication, `BarberShop` managing business operations, and `AppointmentConfirmedHandler` dealing with appointment processing. The Open/Closed Principle is demonstrated through the system's extensibility, where new features can be added without modifying existing code, such as the ability to add new types of appointments or rating systems without changing the core appointment management logic. The Liskov Substitution Principle is maintained through the proper inheritance hierarchy, as none of the classes required inheritance that would make sense. The Interface Segregation Principle is applied by ensuring that classes only implement the methods they need. Finally, the Dependency Inversion Principle is followed by depending on abstractions rather than concrete implementations, allowing for flexible component interactions and easier testing and maintenance. This adherence to SOLID principles results in a system that is not only functional but also maintainable, extensible, and well-structured, making it easier to add new features or modify existing ones without compromising the system's integrity.

Thank you for reading through our project!

Appendix

As stated above, the svg images of the use case diagram and class diagram are not fitting in this pdf. Please look at them in their respective .svg files attatched in the submission.