



Parking Reservation Application

CS542 Design Patterns and OO Design

Presented by: Adityan Elangovan, Adriana Caetano, Aruna Elangovan

Agenda

- Project Objectives
- Requirements
- System Architecture
- Use Case Diagram
- Sequence Diagram
- UML Class Diagram
- Design Patterns
- Demo Video

Project Objectives

Develop a system that enables students to manage their 'CSUSM Campus Parking Reservations.

Implement 'Design Patterns and Object-oriented Analysis' concepts to build an efficient code base in terms of reusability, maintainability, etc.



REQUIREMENTS



Functional Requirements



FR ID #	FUNCTIONAL REQUIREMENT DESCRIPTION
FR01	The application must have a user-friendly UI for all users.
FR02	A user must be able to create an account with their unique ID and basic information to use the application.
FR03	A user registered on the application must be able to sign into the application. Upon successful sign in, the user must land on a home page.
FR04	Upon sign in failure, an error notification must be sent to the user and the user must be able to reset their password using the 'Forgot Password' option.
FR05	A user must be able to view their user profile with their basic information.
FR06	A user must be able to navigate to the 'Make a Reservation' page and select the date for which they wish to search for a free parking spot.
FR07	A user must be given a choice to search for free parking spots in a specific parking lot (Eg: Lot A, B, etc.) for all available time slots.
FR08	A user must be given a choice to search for free parking spots during a specific time slot (Eg: 8:00 am - 9:00am) in all available parking lots.

Functional Requirements



FR ID #	FUNCTIONAL REQUIREMENT DESCRIPTION
FR09	Upon successful search, a user must be able to reserve a free parking spot for the selected date, time slot and parking lot.
FR10	Upon successful reservation, a user must receive an information notification indicating a successful transaction.
FR11	Upon a reservation failure, a user must receive an error notification indicating a failed transaction.
FR12	A user must be able to view a list of all their upcoming confirmed reservations on the home page.
FR13	A user must receive a reminder notification 10 mins prior to their scheduled check-in time.
FR14	A user must be able to check-in to a parking spot reserved by them on the scheduled date and time slot.
FR15	Upon successful check-in, the user must receive an information notification indicating a successful transaction and must be able to view “Parked” status with the check-in timestamp on their reservation.
FR16	Upon successful check-in, the user must also be navigated to a ‘Ratings’ page where they can provide ratings for the previous user of the same parking spot with regards to punctuality in check-out, leaving the parking spot clean, etc.
FR17	Upon a check-in failure, the user must receive an error notification indicating a failed transaction.

Functional Requirements



FR ID #	FUNCTIONAL REQUIREMENT DESCRIPTION
FR18	A user must receive a reminder notification 10 mins prior to their scheduled check-out time.
FR19	A user must be able to check-out of a parking spot checked-in by them on the scheduled date and time slot.
FR20	Upon successful check-out, the user must receive an information notification indicating a successful transaction and must be able to view “Vacated” status with the check-out timestamp on their reservation.
FR21	Upon a check-out failure, the user must receive an error notification indicating a failed transaction.
FR22	A user must be able to cancel a reservation previously reserved by them for a specific date and time slot.
FR23	Upon successful cancelation, the user must receive an information notification indicating a successful transaction and must be able to view “Vacated” status with the check-out timestamp on their reservation.
FR24	Upon a cancelation failure, the user must receive an error notification indicating a failed transaction.
FR25	A user must be able to sign out of their account in the application.

Non-Functional Requirements



NFR ID #	TYPE	NON-FUNCTIONAL REQUIREMENT DESCRIPTION
NFR01	USABILITY	The application must be user-friendly for CSUSM's diverse population in terms of navigation, font style and size, color palette, etc
NFR02	SUPPORTABILITY	The application must be supported by Web for this phase of the project.
NFR03	RELIABILITY	The application must be reliable with an appropriate disaster recovery plan.
NFR04	PERFORMANCE	The application must be lightweight with an SLA < 3 seconds for each interface.
NFR05	SECURITY	The application must use secured databases and comply with PII Protection standards of CSUSM.
NFR06	REUSABILITY	The application must implement a reusable code base to account for additional functionalities in the future phases.
NFR07	MAINTAINABILITY	The application must be easy to maintain in terms of version upgrades and code fixes.

SYSTEM

UI : Qt Creator

Database : SQLiteStudio

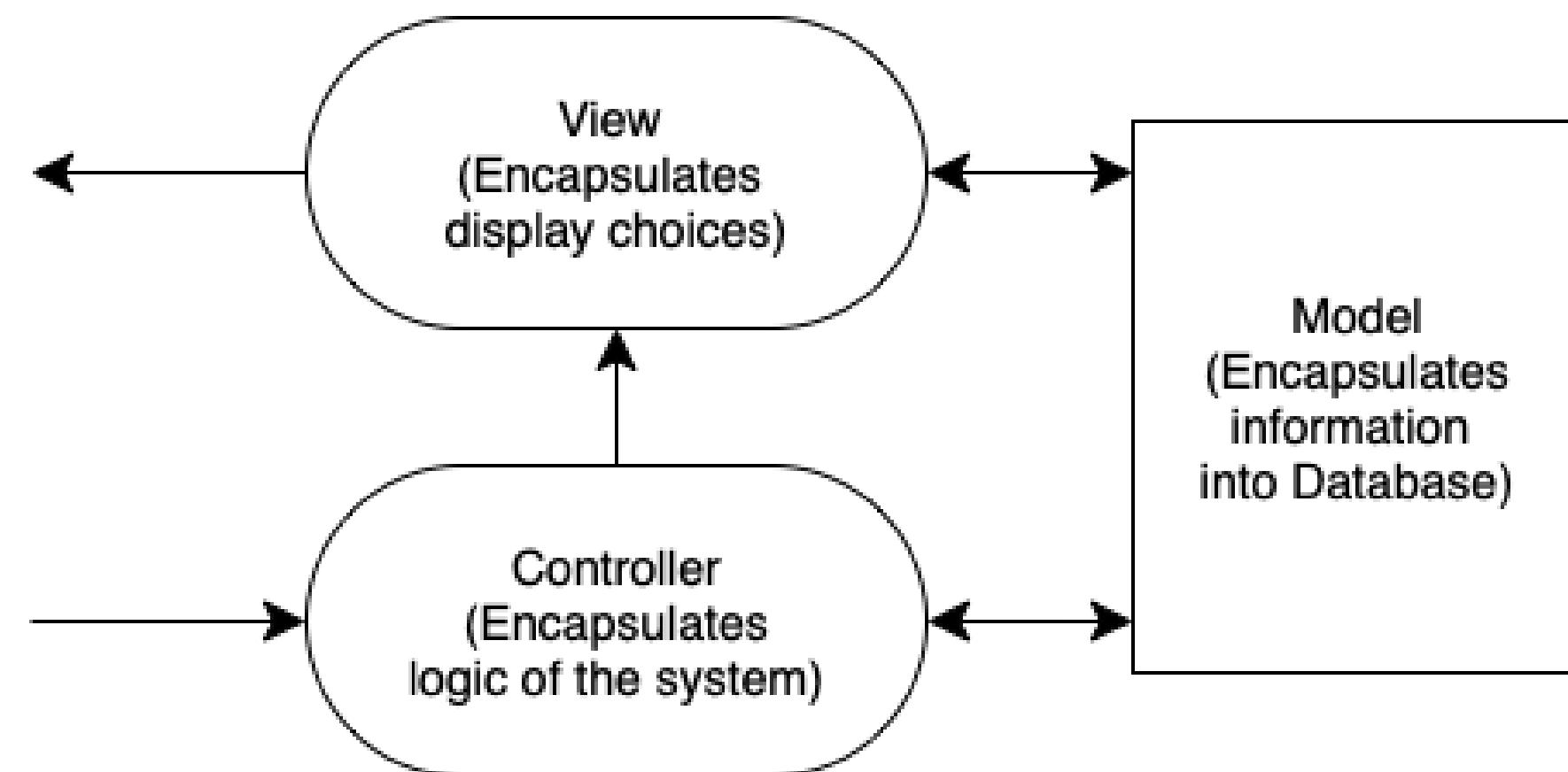


System Architecture - MVC

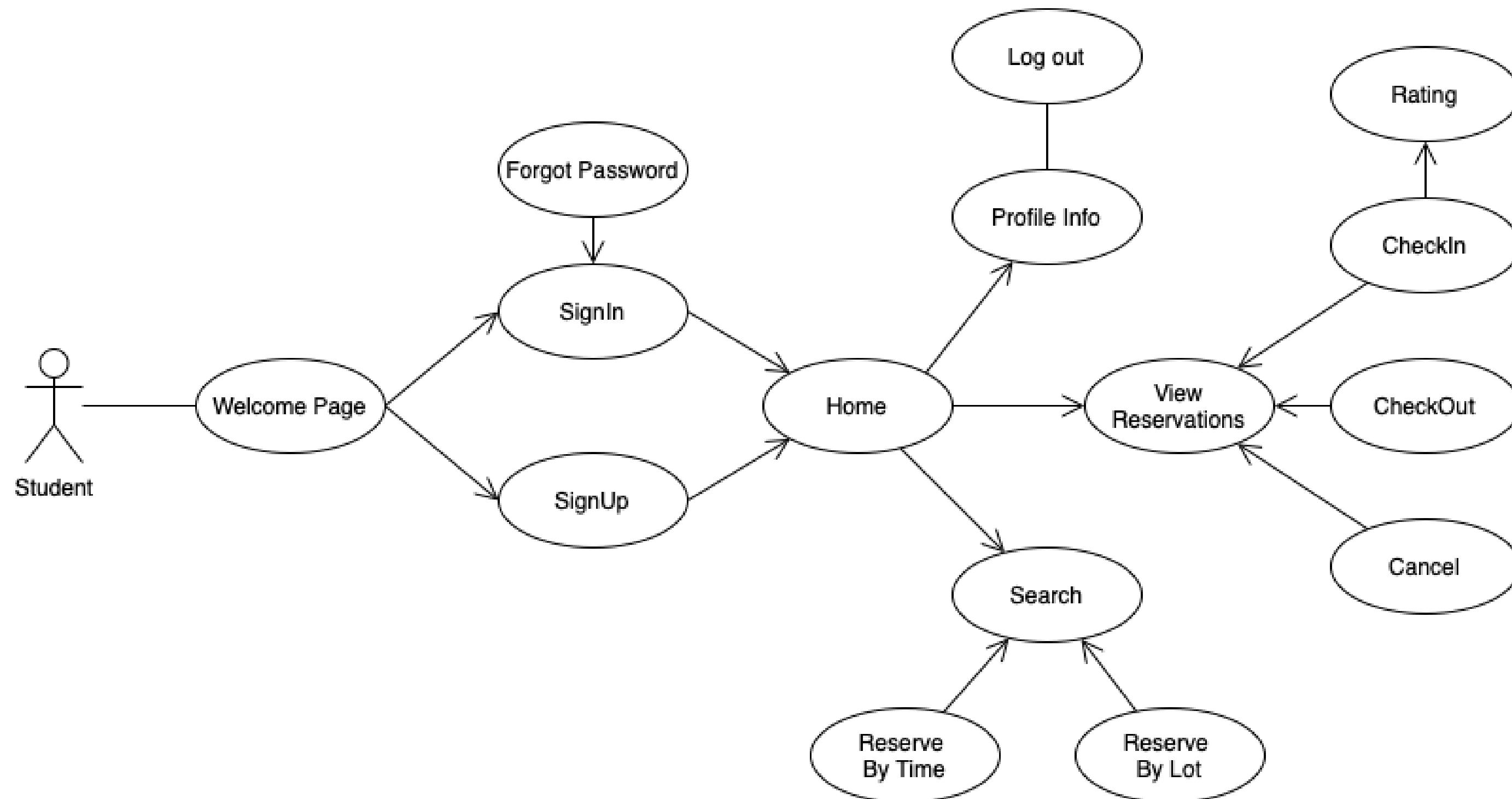
Model: the information stored in the database

View: the .ui files that display the information

Controller: the .cpp files that drive the logic of the system

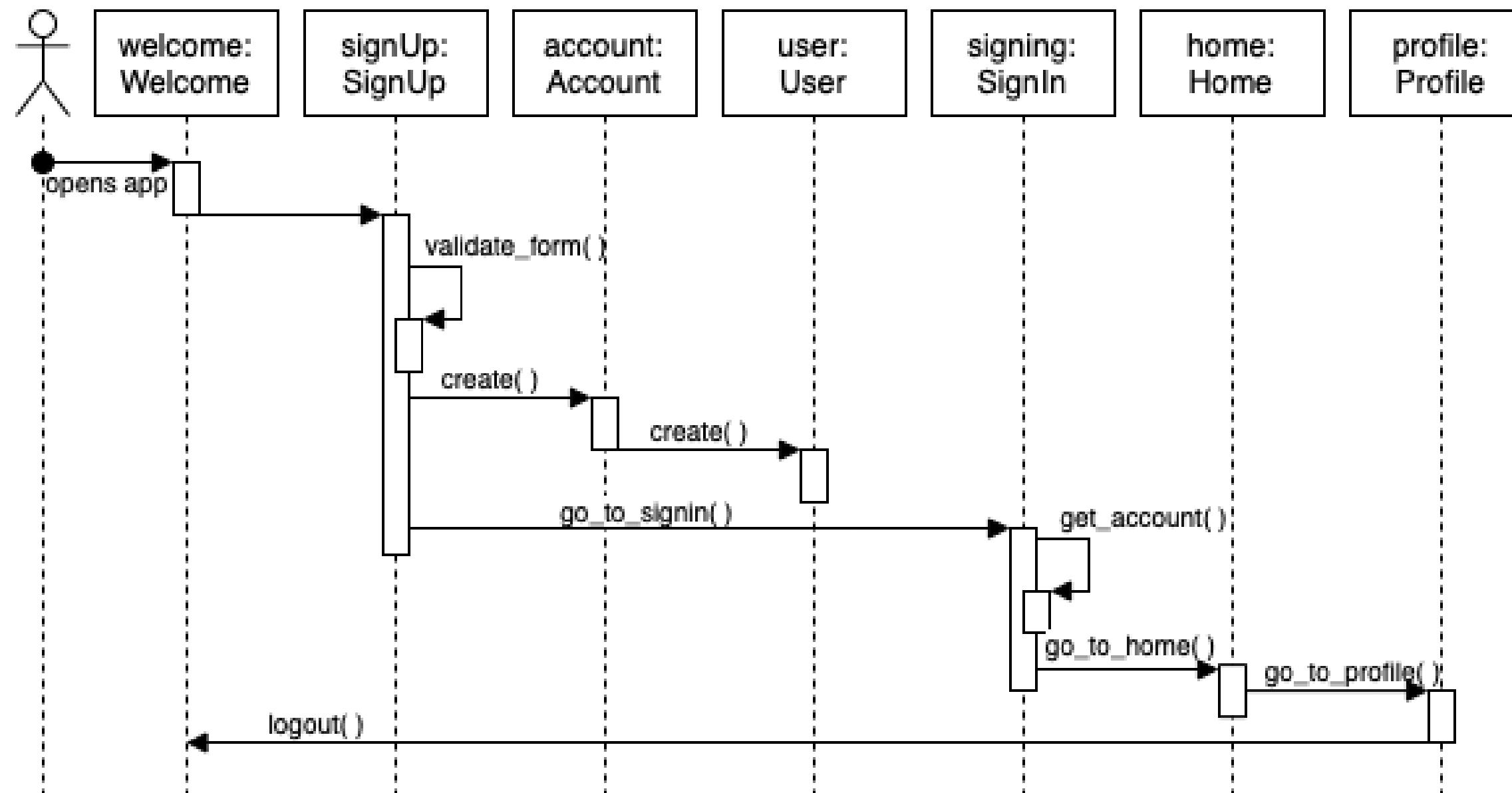


Use Case Diagram



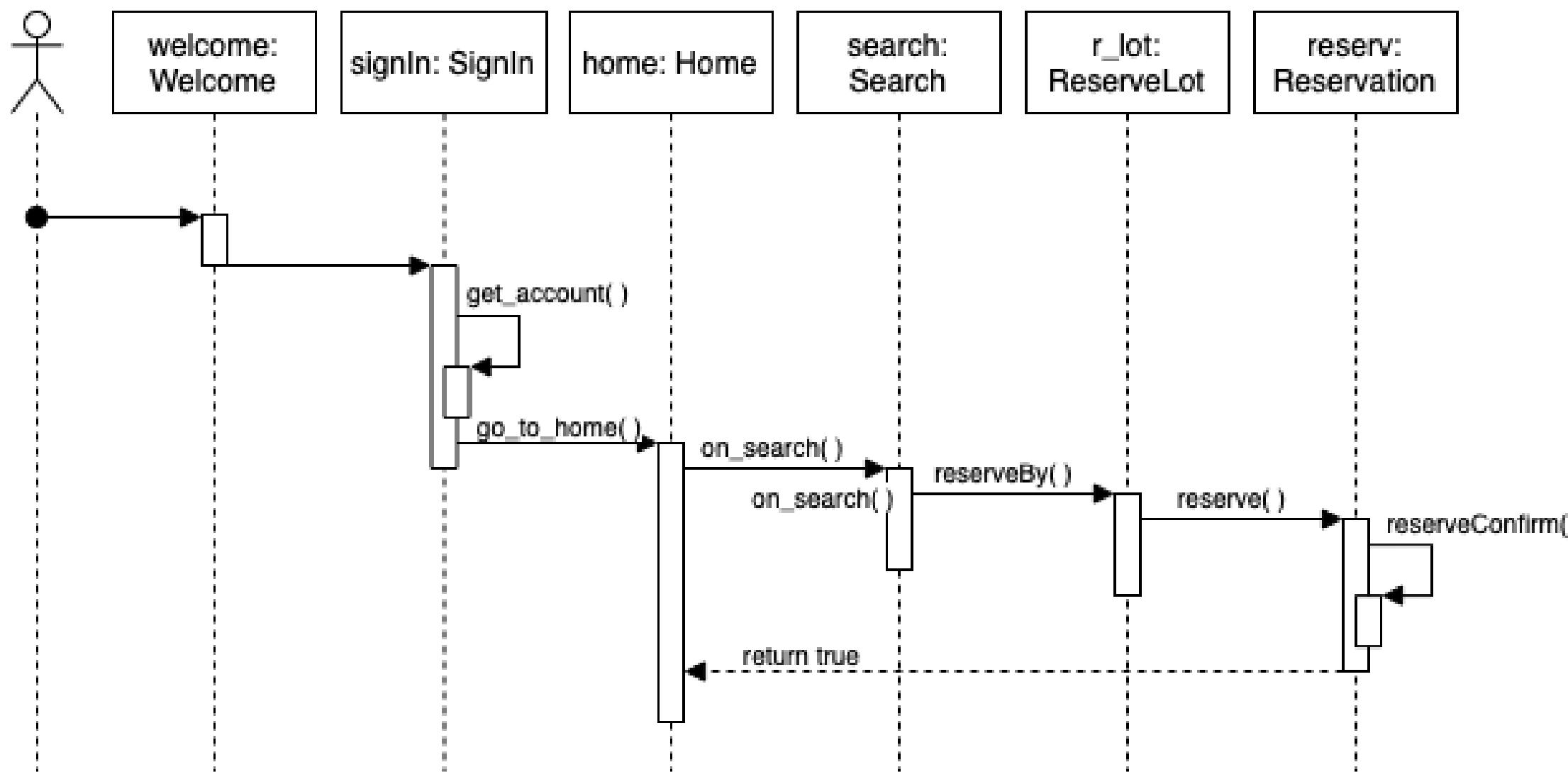
Sequence Diagram

Student wants to create an account



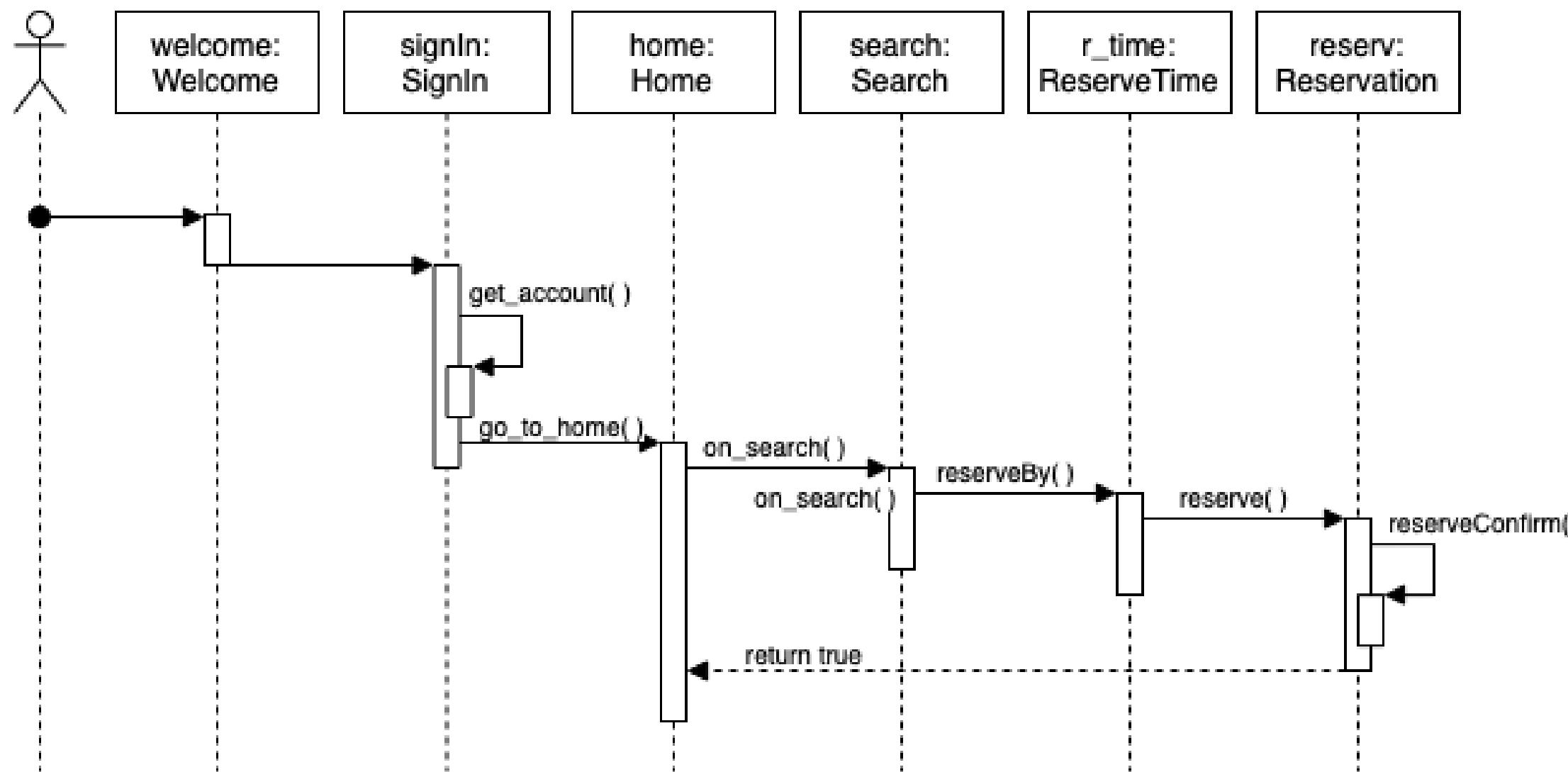
Sequence Diagram

Existing Account wants to make a reservation searching by lot



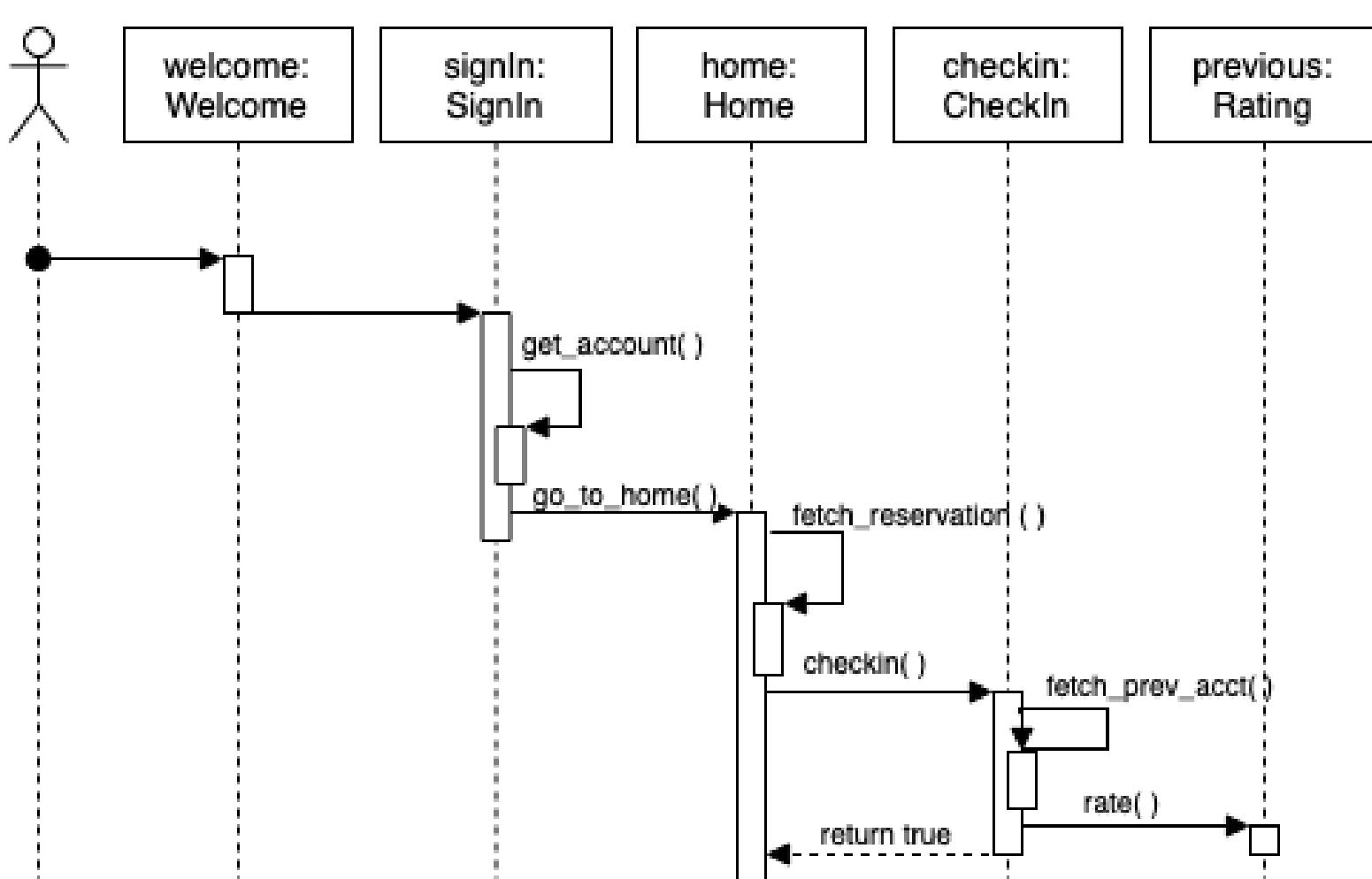
Sequence Diagram

Existing Account wants to make a reservation searching by time

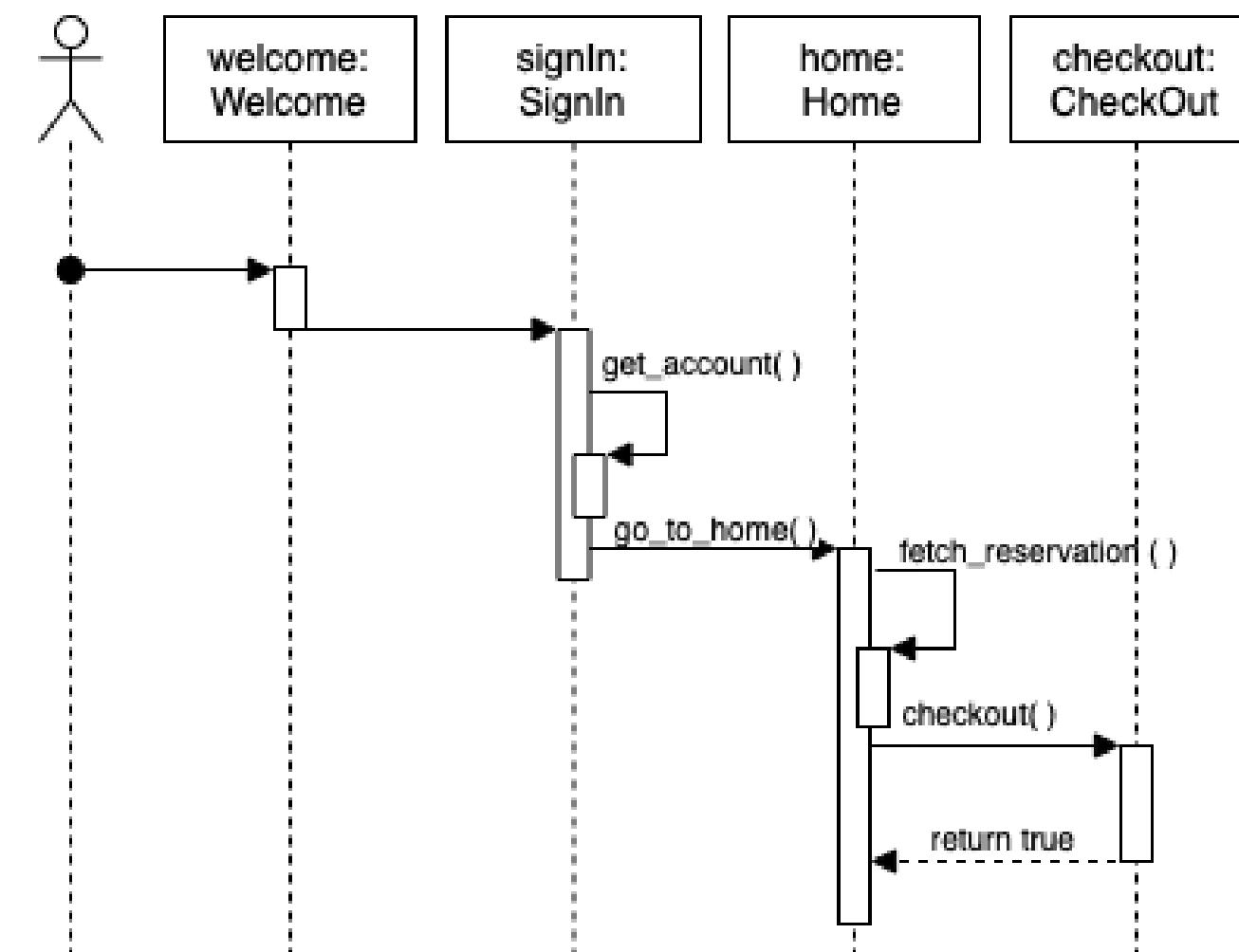


Sequence Diagram

Existing Account with an active reservation wants to check-in

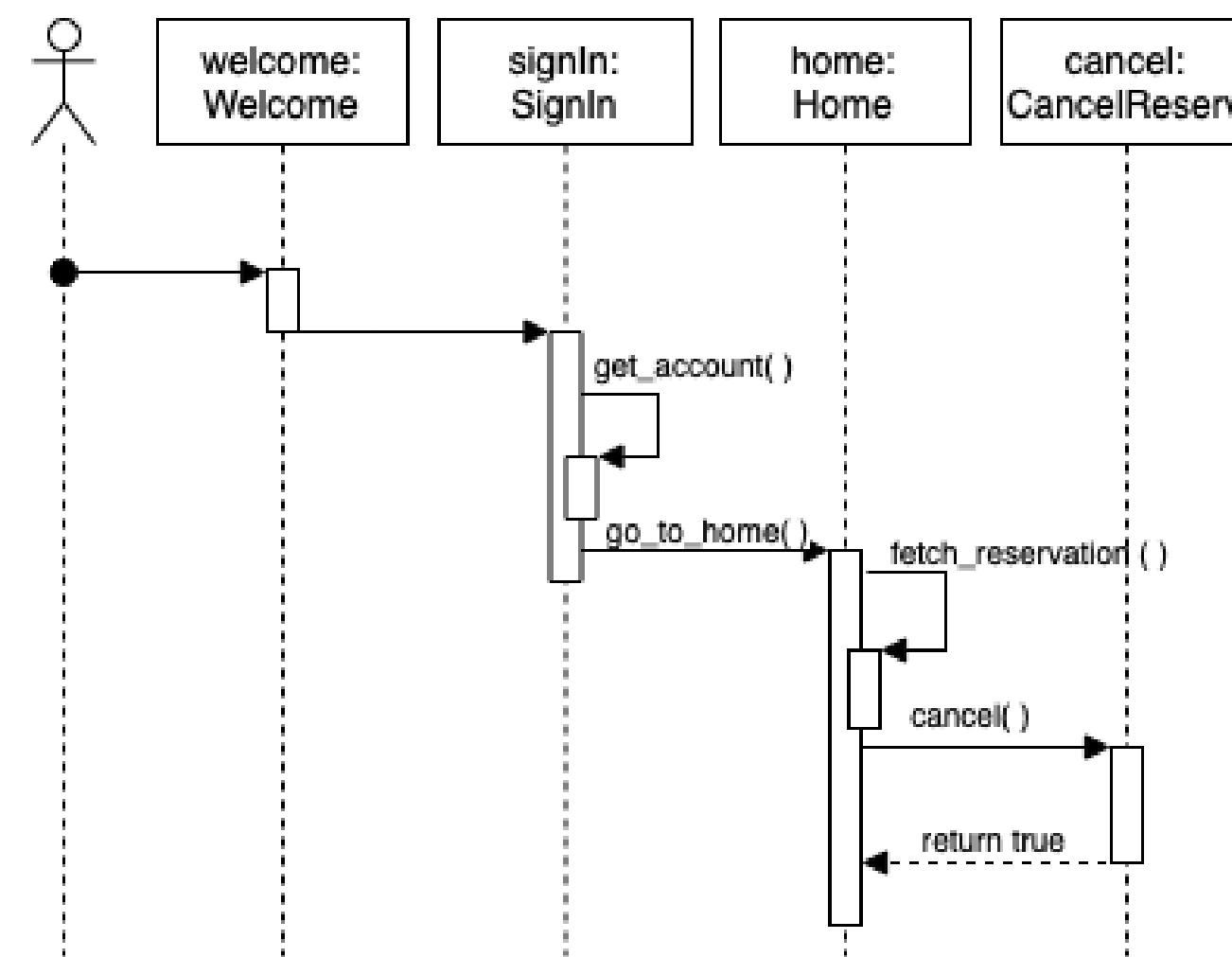


Existing Account with an active reservation wants to checkout

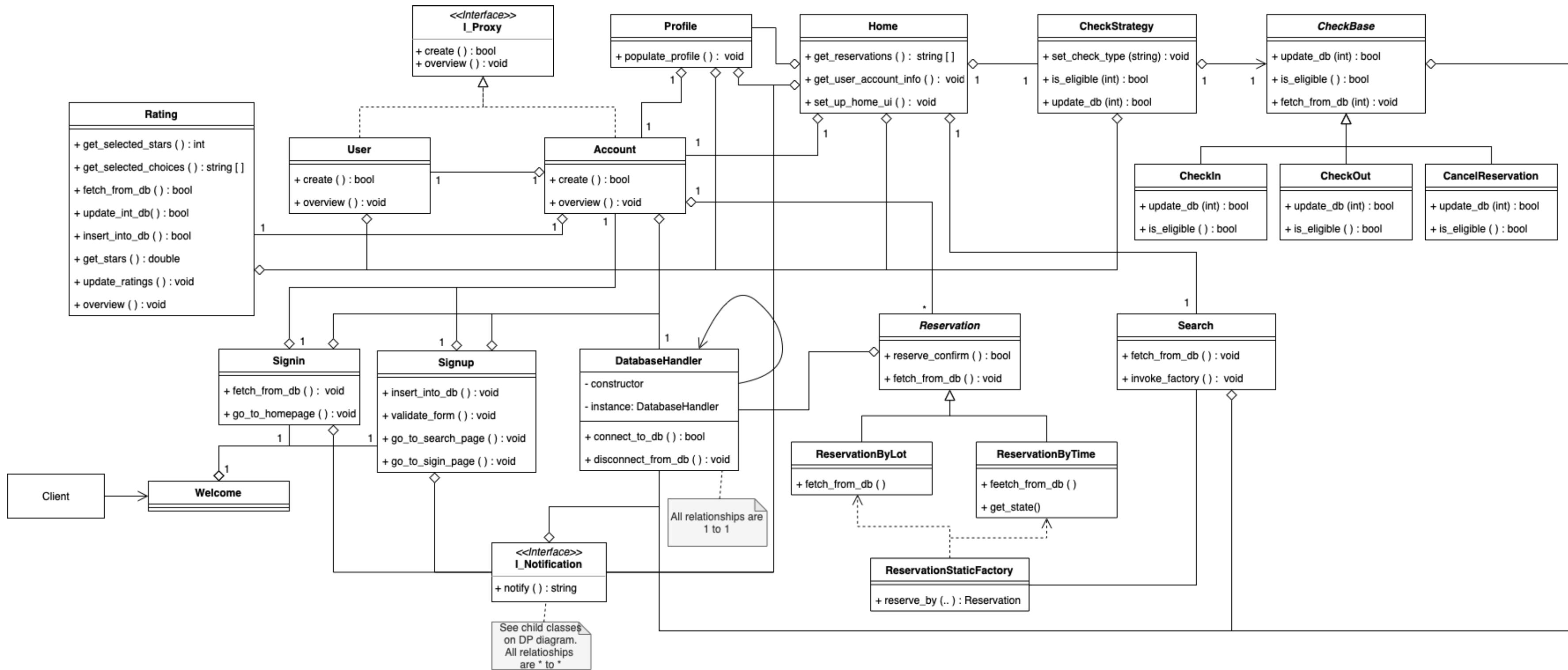


Sequence Diagram

Existing Account with an active reservation wants to cancel it



UML Class Diagram



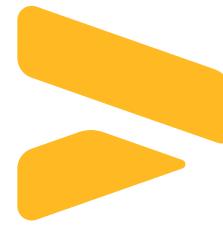
DESIGN PATTERNS

Creational Patterns: Singleton, Static Factory

Structural Patterns: Proxy, Decorator

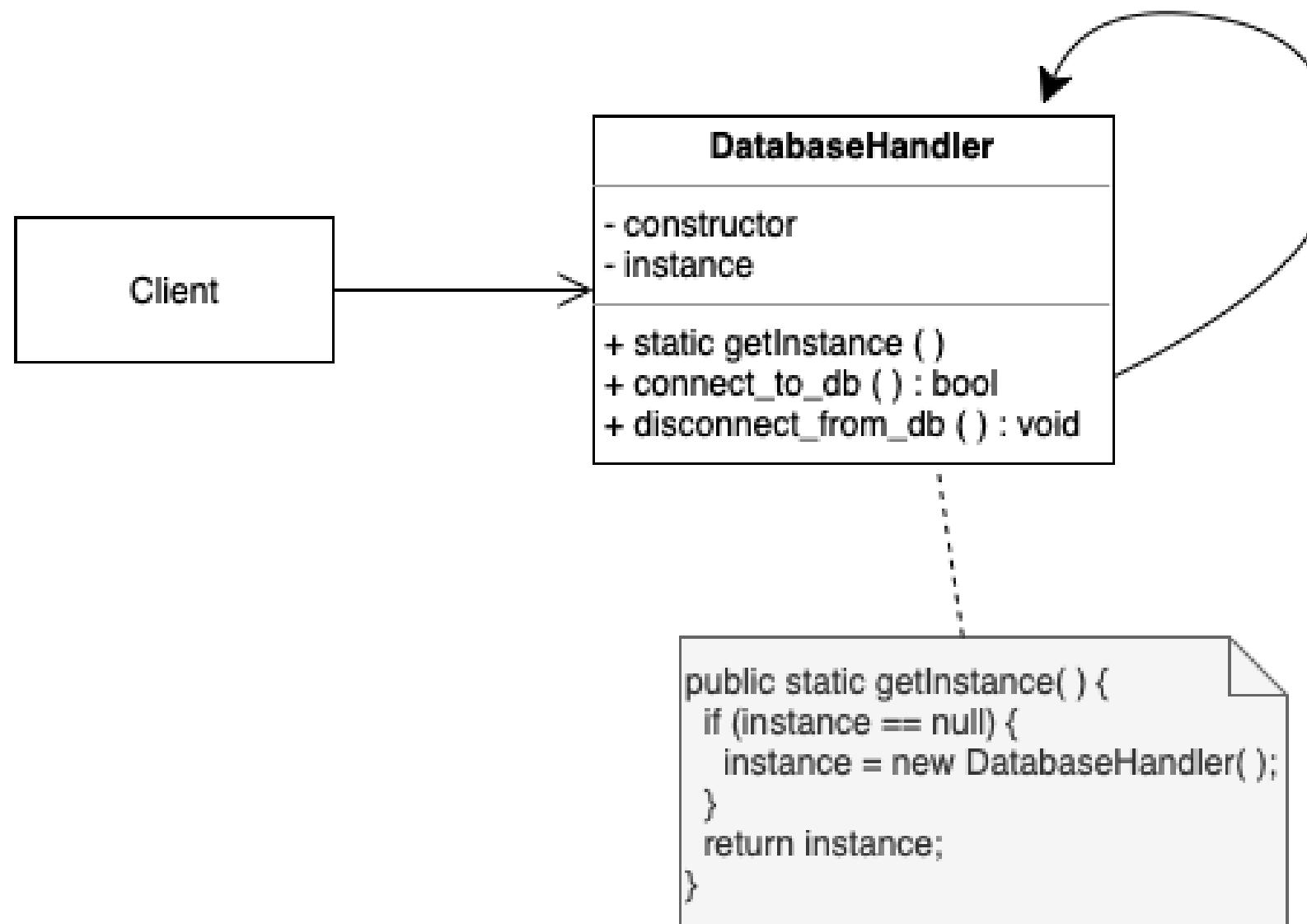
Behavioral Pattern: Strategy





Singleton

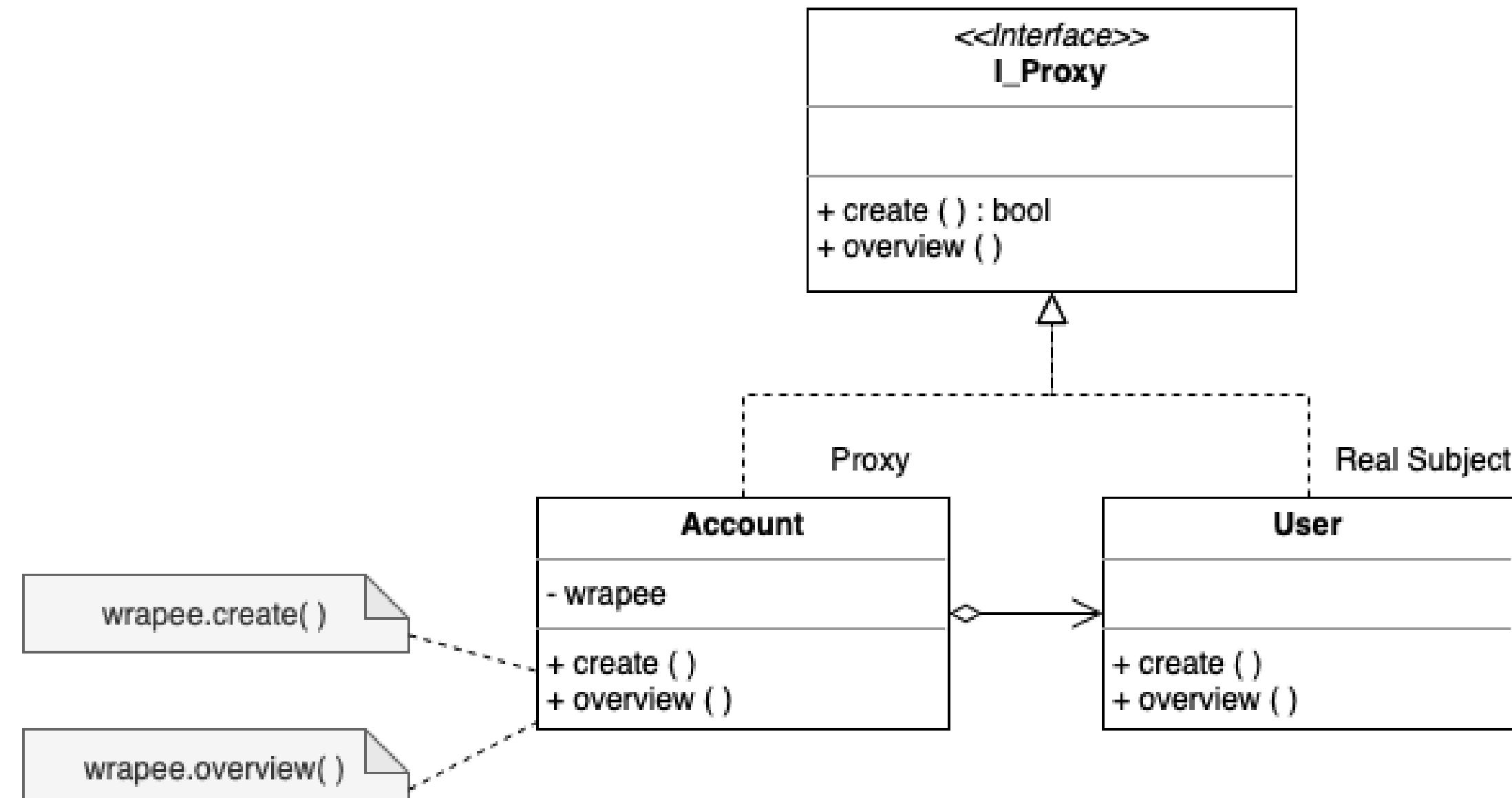
Creational Pattern





Proxy

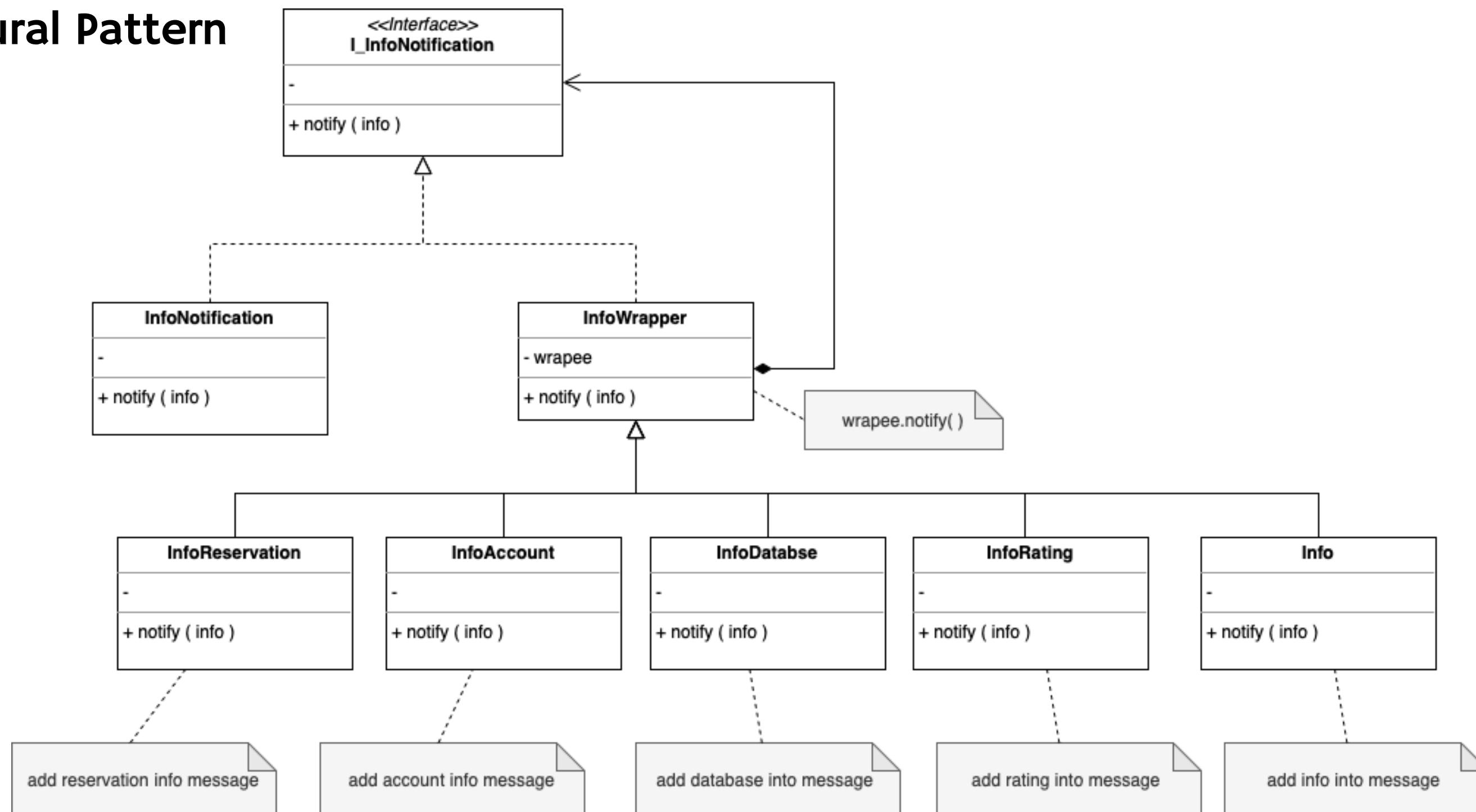
Structural Pattern





Decorator

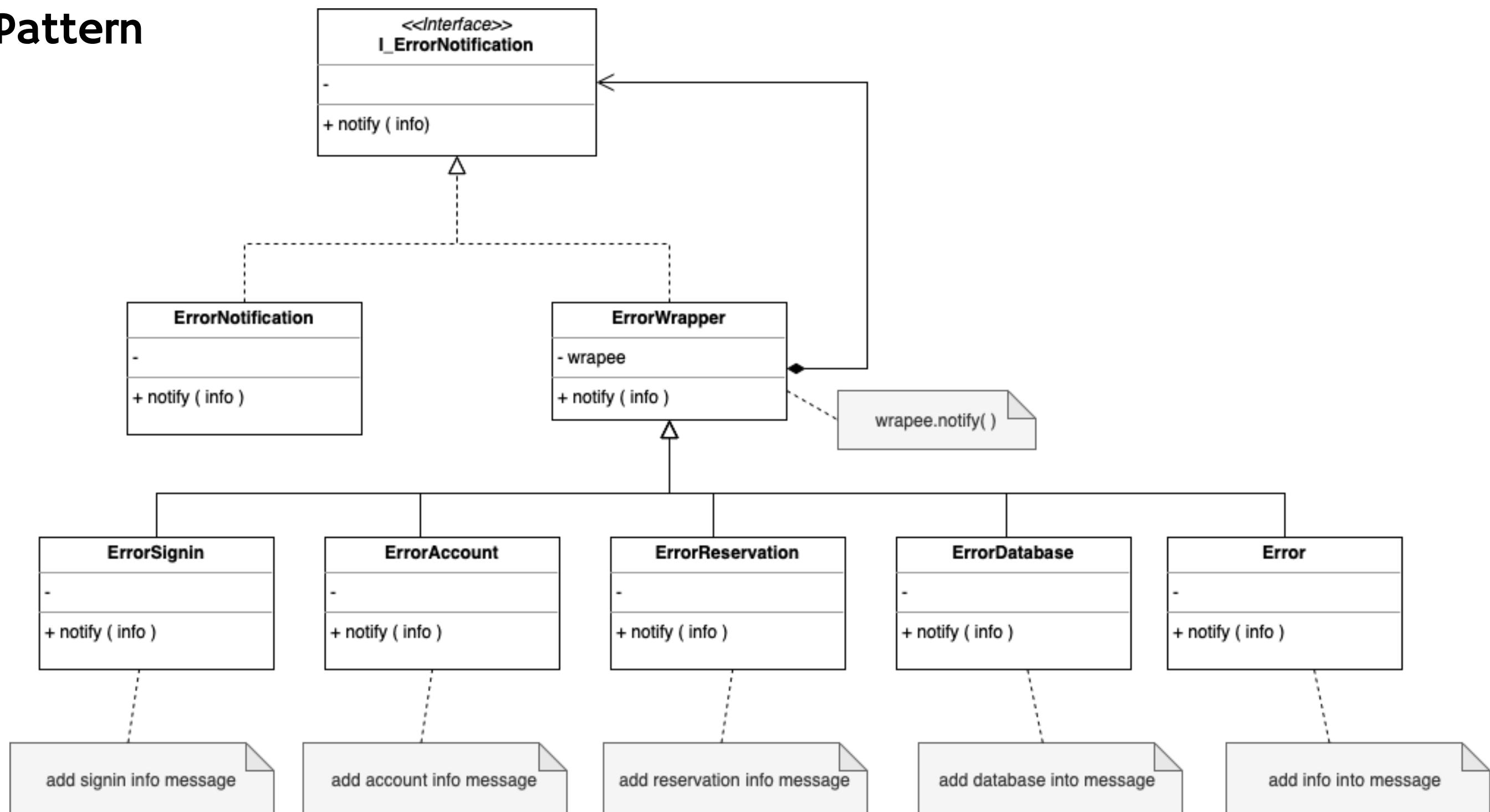
Structural Pattern





Decorator

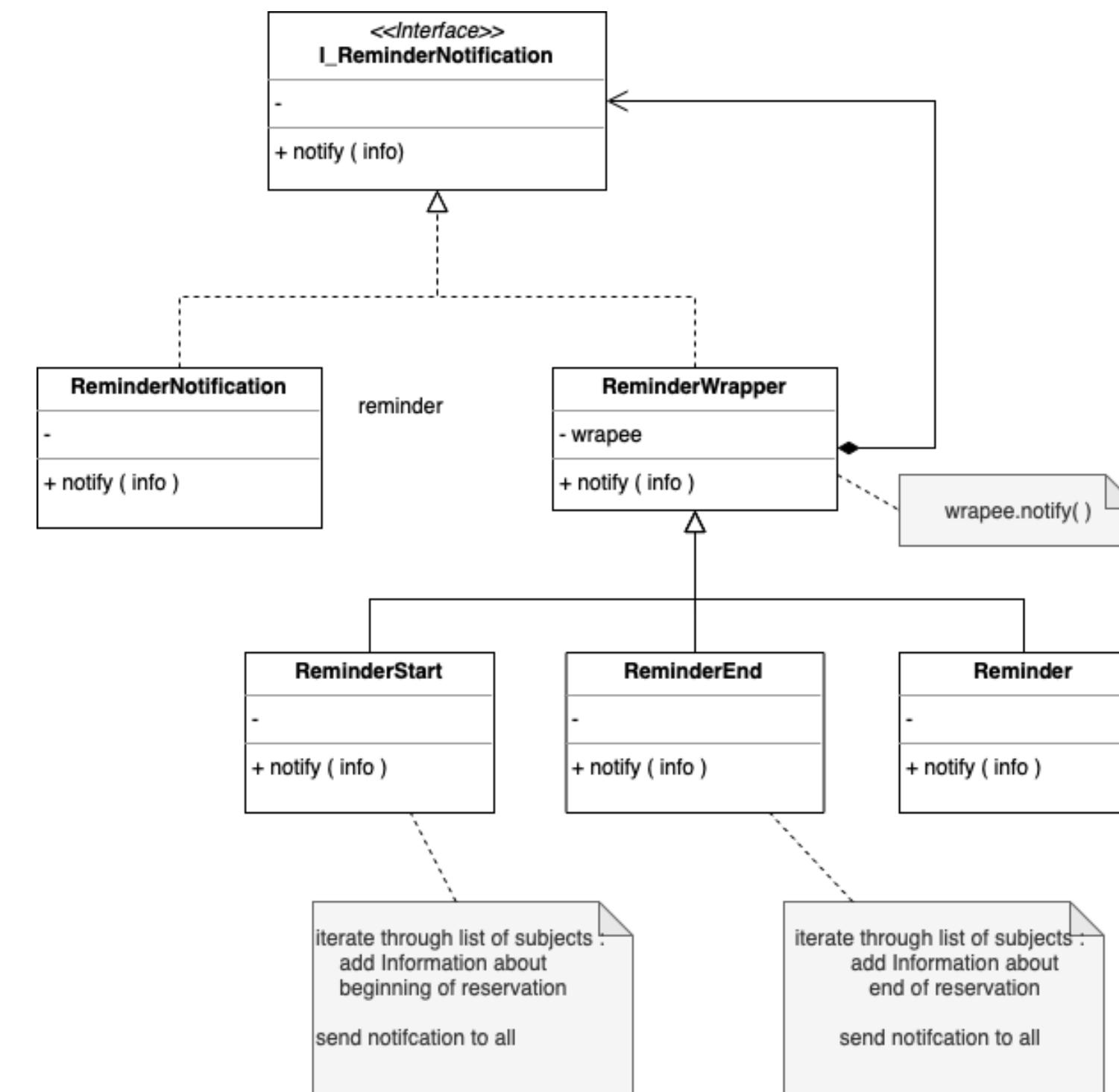
Structural Pattern

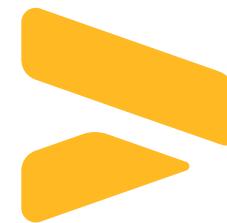




Decorator

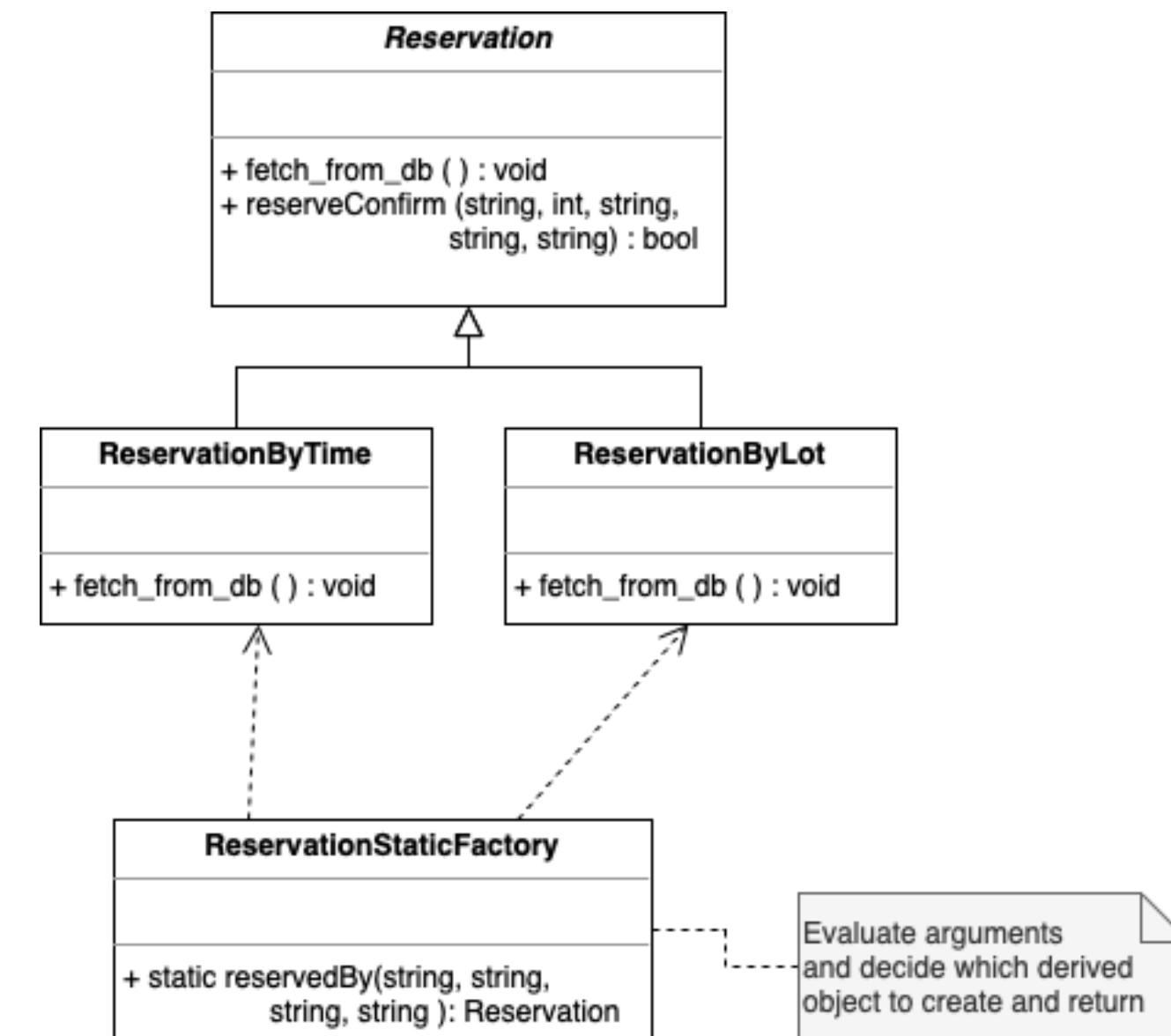
Structural Pattern





Static Factory

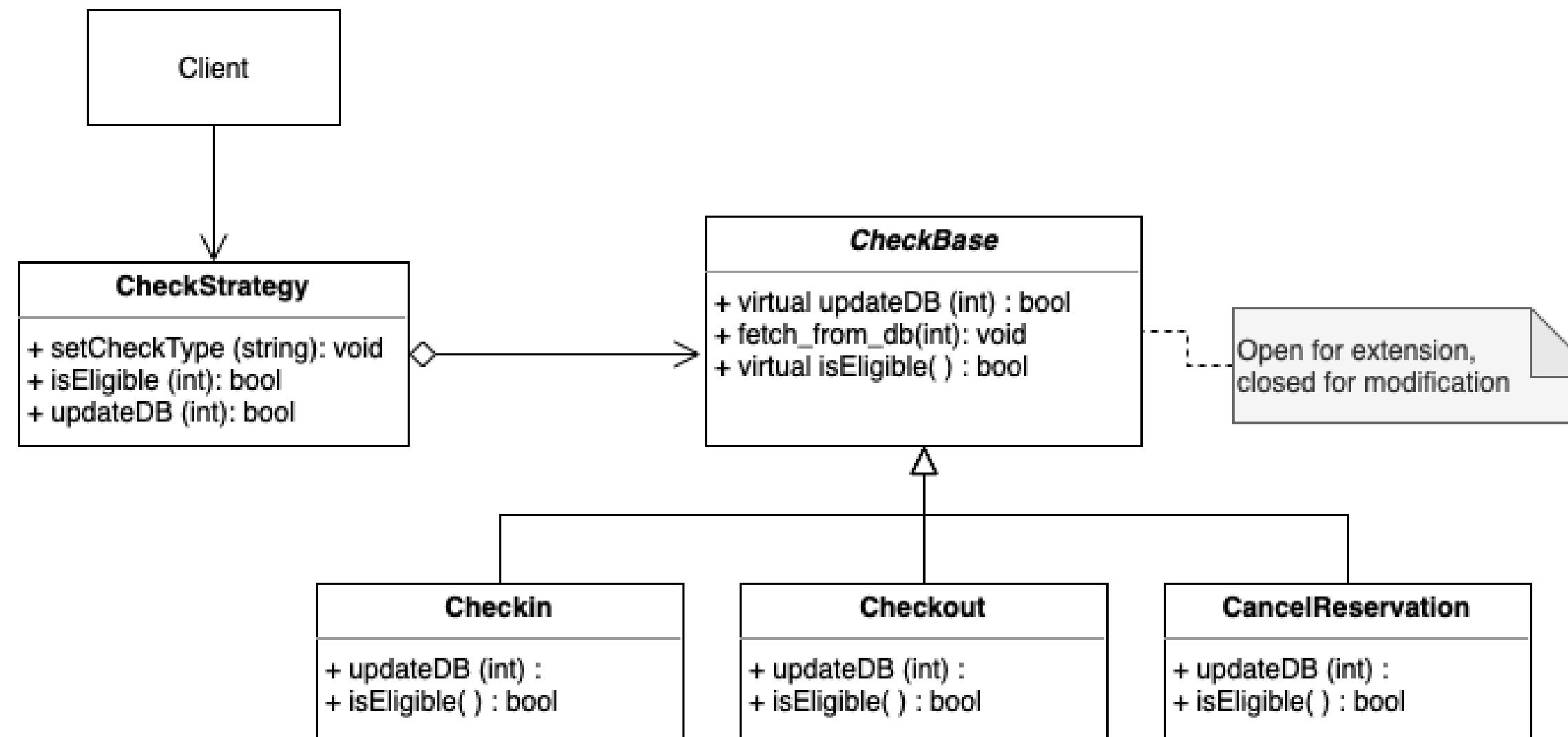
Creational Pattern





Strategy

Behavioral Pattern



Quality Assurance

REQ ID #	ACCEPTANCE CRITERION TYPE	STATUS
FR01	User Friendly UI	Success, Accepted
FR02, FR03, FR04, FR05	Account Creation, User Sign in, Homepage, My Profile Page, Information Notifications, Error Notifications	Success, Accepted
FR06, FR07, FR08, FR09, FR10, FR11, FR12	Search and Make a Reservation, Information Notifications, Error Notifications	Success, Accepted
FR13, FR14, FR15, FR16, FR17, FR18, FR19, FR20, FR21	Reservation Check-in, Check-out, Reminder Notifications, Information Notifications, Error Notifications	Success, Accepted
FR22, FR23, FR24	Reservation Cancelation, Information Notifications, Error Notifications	Success, Accepted
FR25	User Sign out	Success, Accepted
NFR01, NFR02, NFR04, NFR05, NFR06, NFR07	Achieved through Design Pattern implementation	Success, Accepted
NFR03	Achieved as part of Deployment activity	Success, Accepted

Video Demo



Thank you!

