Final Project - Initial Design

The initial design of the Court Reservation System is based on object-oriented principles and is structured to meet the functional requirements of the project. This design aims to provide an efficient, maintainable, and user-friendly system to manage court reservations for a sports club, catering to club members, coaches, and officers.

The system includes a total of eight classes, each with its own set of responsibilities and collaborators. These classes are:

1. User: This is the base class for representing users in the system. It contains basic functionality such as login() and logout(). The DataPersistence class is its collaborator for loading and saving user data.

2. ClubMember, ClubOfficer, and ClubCoach: These are subclasses of the User class, representing different user types with specific roles and permissions. They interact with the Schedule, Court, and Reservation classes to perform tasks like viewing schedules, reserving courts, canceling reservations, or managing open play sessions.

3. Court: This class represents individual courts, and it is responsible for managing reservation slots, providing available slots, and handling reservation cancellations. Its collaborators are the Schedule and Reservation classes.

4. Schedule: This class is responsible for managing the daily schedule of court reservations. It interacts with the User, Court, and Reservation classes to display the schedule, reserve courts, cancel reservations, and manage open play sessions.

5. Reservation: This class represents a reservation made by a user. It is responsible for creating and deleting reservations and collaborates with the User, Court, and Schedule classes.

6. DataPersistence: This class handles the persistence of the system's state, ensuring that data is saved and loaded correctly. It collaborates with the User, Court, and Reservation classes to manage user, court, and reservation data.

7. ReservationSystem: This class handles the system, meaning using all the classes to fulfill the functions provided to each User. It collaborates with the User, Schedule, TimeUtils, and DataPersistence class to manage user functions and saving/loading the system state.

8. TimeUtils: This utility class handles all functions related to time and dates. It collaborates with the User, Reservation and Schedule classes to help with time calculations/storing.

Here is a more in-depth look into each of the classes:

1. User
    - Member Variables/Attributes:
        - username
        - password
        - membership_type
    - Functions:
        - login()
        - logout()
2. ClubMember (subclass of User)
    - Member Variables/Attributes:
        - skill_level
    - Functions:
        - view_schedule()
        - reserve_court()
        - cancel_reservation()
3. ClubOfficer (subclass of User)
    - Functions:
        - manage_open_play()
        - modify_reservations()
4. ClubCoach (subclass of User)
    - Functions:
        - reserve_court()
        - cancel_reservation()
5. Court
    - Member Variables/Attributes:
        - court_number
        - reservation_slots
    - Functions:
        - get_available_slots()
        - reserve_slot()
        - cancel_reservation()
6. Schedule
    - Member Variables/Attributes:
        - courts
        - reservations
    - Functions:
        - view_daily_schedule()
        - reserve_court()
        - cancel_reservation()
        - manage_open_play()
7. Reservation
    - Member Variables/Attributes:
        - reservation_id
        - court_number

        - start_time
        - end_time
        - users
    - Functions:
        - create()
        - delete()
8. DataPersistence
    - Member Variables/Attributes:
        - users_data
        - courts_data
        - reservations_data
    - Functions:
        - save_data()
        - load_data()
        - save_users_data()
        - load_users_data()
        - save_courts_data()
        - load_courts_data()
        - save_reservations_data()
        - load_reservations_data()
9. ReservationSystem
    - Member Variables/Attributes:
        - users
        - next_reservation_id
        - Schedule
        - logged_in_user
    - Functions:
        - User find_user(username: String)
        - ClubOfficer find_officer()
        - ClubOfficer find_officer(username: String)
        - add_user (username: String, password: String, role: String, skill_level: String)
        - display_schedule()
        - view_requests(officer: ClubOfficer)
        - view_reservations(user: User)
        - reserve_court()
        - delete_reservation()
        - add_user_to_reservation()
        - remove_user_from_reservation()
        - request_reservation_cancellation()
        - save_data_to_file()
        - load_data_from_file()
        - login()
        - signup()

        - logout()
        - is_logged_in()

10. TimeUtils
    - Functions:
        - get_hour_from_time_point()
        - get_day_from_time_point()
        - print_time_and_date()
        - get_date_string()
        - get_time_string()
        - string_to_time_point()
        - is_future()
        - is_date_valid()
        - two_days_in_advance()
        - within_seven_days()
        - within_opening_hours()
        - within_coach_hours()
        - within_coach_hours()

Relationships:

- ClubMember, ClubOfficer, and ClubCoach classes inherit from the User class.
- ClubMember, ClubOfficer, and ClubCoach collaborate with Schedule and Reservation classes.
- Court class collaborates with Schedule and Reservation classes.
- Schedule class collaborates with User, Court, and Reservation classes.
- Reservation class collaborates with User, Court, and Schedule classes.
- DataPersistence class collaborates with User, Court, and Reservation classes.
- ReservationSystem class collaborates with User, Schedule, TimeUtils, and DataPersistence
- TimeUtils class collaborates with User, Reservation and Schedule

CRC Cards & Class Diagrams

User	
Responsibilities	Collaborators
- login() - logout()	- DataPersistence

ClubMember (subclass of User)	
Responsibilities	Collaborators
- view_schedule() - reserve_court() - cancel_reservation()	- Schedule - Reservation

ClubOfficer (subclass of User)	
Responsibilities	Collaborators
- manage_open_play() - modify_reservations()	- Schedule - Reservation

ClubCoach (subclass of User)	
Responsibilities	Collaborators
- reserve_court() - cancel_reservation()	- Schedule - Reservation

Court	
Responsibilities	Collaborators
- get_available_slots() - reserve_slot() - cancel_reservation ()	- Schedule - Reservation

Schedule	
Responsibilities	Collaborators
- view_daily_schedule() - reserve_court() - cancel_reservation() - manage_open_play()	- User - Court - Reservation

Reservation	
Responsibilities	Collaborators
- create() - delete()	- User - Court - Schedule

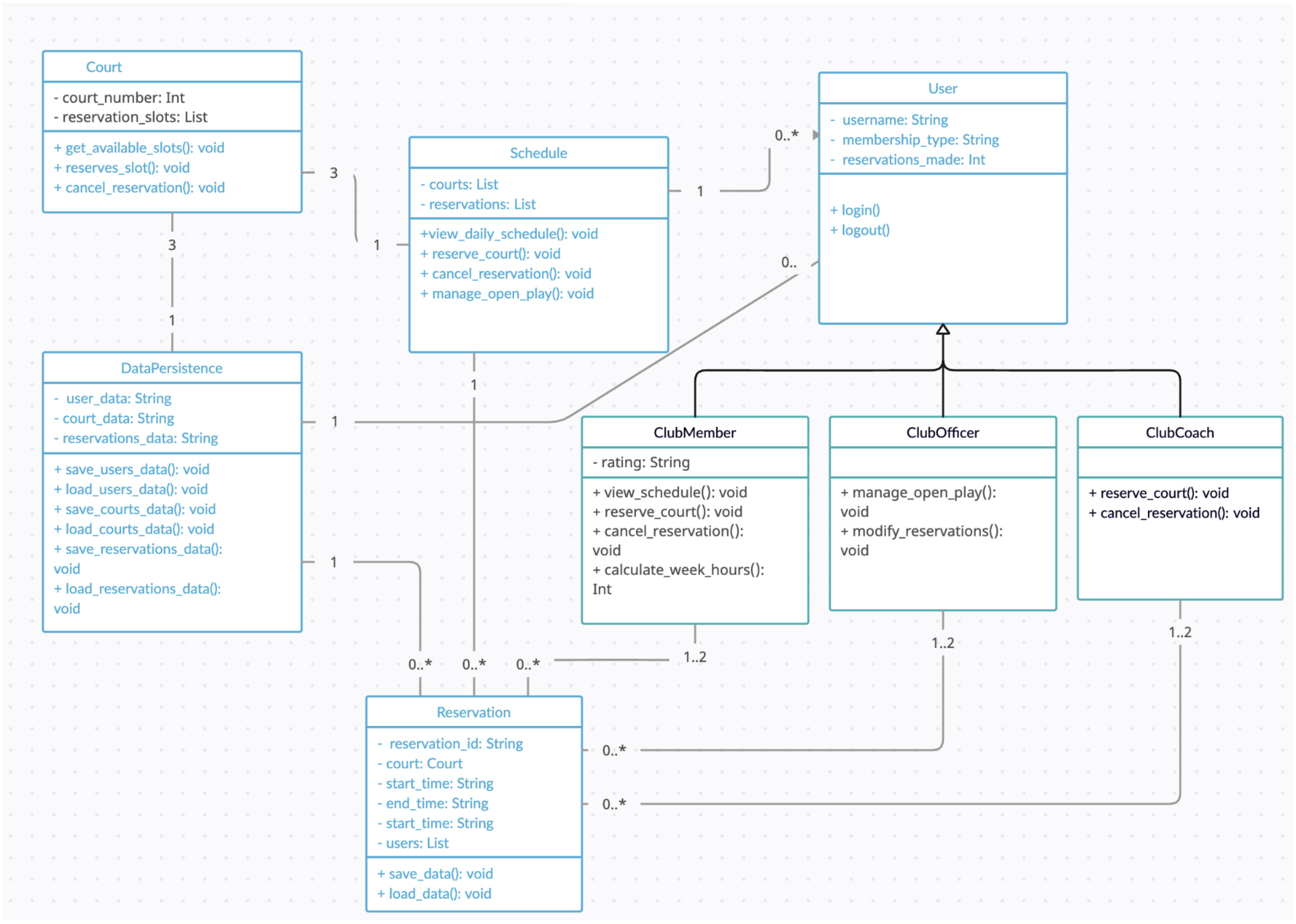
DataPersistence	
Responsibilities	Collaborators
- save_data() - load_data()	- User - Court - Reservation

ReservationSystem	
Responsibilities	Collaborators
- display_schedule() - reserve_court() - delete_reservation() - add_user_to_reservation() - save_data() - load_data()	- User - Schedule - TimeUtils - DataPersistence

TimeUtils	
Responsibilities	Collaborators
- convert_time_to_string()	- User

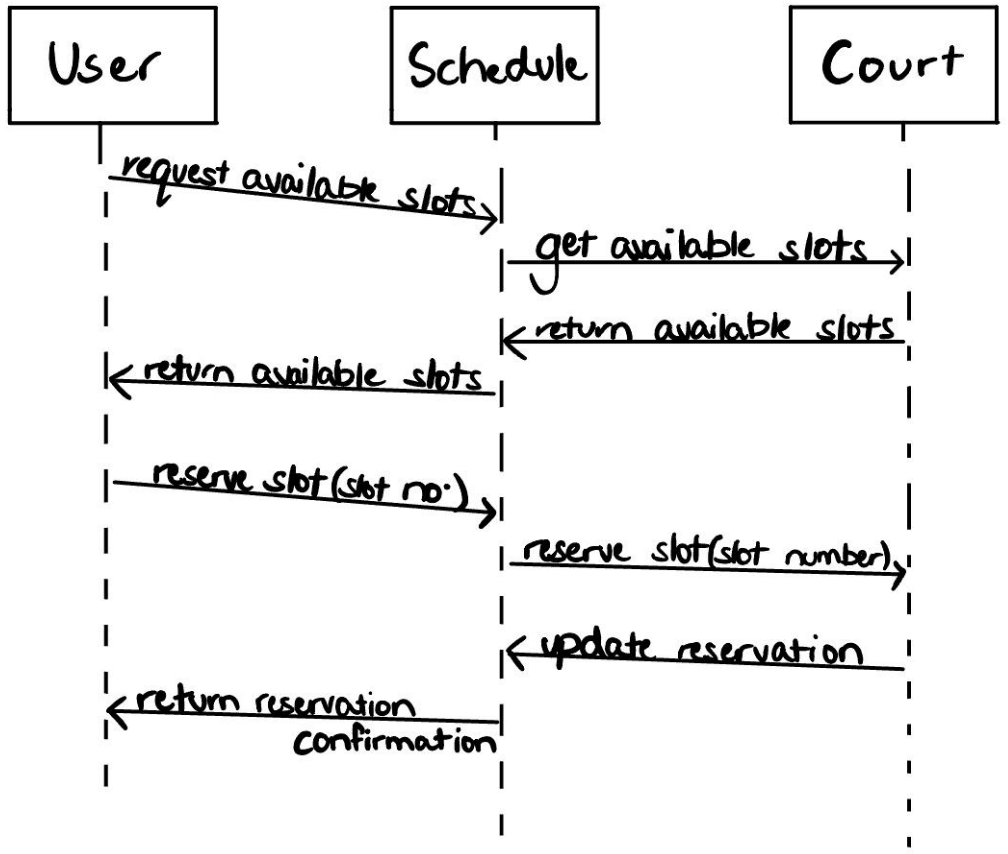
- convert_string_to_time() - is_time_within() - is_time_on_date()	- Schedule - ReservationSystem

UML Diagrams

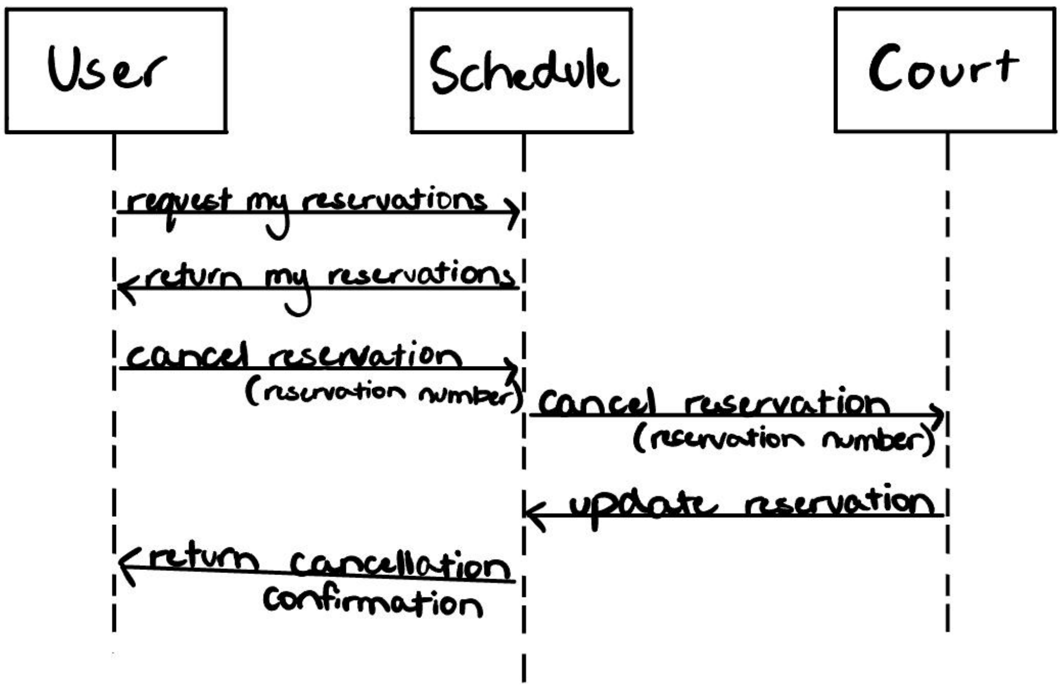


Sequence Diagrams

Reserving a Court



Canceling a Reservation


User
Schedule
Court
request my reservations
return my reservations
cancel reservation (reservation number)
cancel reservation (reservation number)
update reservation
return cancellation confirmation

Other data structures and files

Data Structures:

1. Lists/Arrays:
    - A list or array can be used to store courts in the Schedule class.
    - A list or array can be used to store reservations in the Schedule class.
    - A list or array can be used to store reservation_slots in the Court class.
    - A list or array can be used to store users in a Reservation.
2. Dictionaries/HashMaps:
    - A dictionary or hashmap can be used to store user data, court data, and reservation data in the DataPersistence class. This would facilitate quick lookups and updates.

Files:

1. users_data.txt:
    - This file will store user information, such as usernames, passwords, membership types, and skill levels (for ClubMembers).
2. reservations_data.txt:
    - This file will store reservation information, such as reservation IDs, associated court numbers, start times, end times, and the usernames of the users who made the reservations.
3. Reservation_id.txt
    - This file will store the current reservation id