

Matthew Xue, Tung-Chun Wu, Vevey Zhan

CS3520

Prof Bhutta

6/22/2023

Final Project: The Initial Design

Our initial design of the court reservation system for the new pickleball facility uses good object-oriented design principles such as SOLID, DRY, KISS, and YAGNI to effectively create a platform for users to easily reserve courts to play pickleball. The goal of our design is to be as simple as possible, so it will be easy for any user to navigate the application with no issues (we will include instructions for users for everything in our application). Below, we created the design of our project through crc cards, uml diagram, and sequence diagrams. We used each diagram to help produce our final product.

Our code had a total of eight unique classes that were the foundation of the court reservation application.

1. **Schedule:** This class manages the schedule for the courts at this pickleball facility. It displays the schedule, displays the occupied time slots and the open time slots, the name of the member or coach that reserved the court, etc. This class works with Reservation, Court, and User.
2. **Court:** This class initializes each individual court (one through three). It stores information about reservations and availability of the courts. This class works with the Schedule, Reservation, and ReservationSystem class.
3. **DataPersistence:** This class makes sure that all the data is being handled safely and correctly. It stores and loads the data of users, reservations, and more to text files that we created. This class works with User, Court, and Reservation.
4. **User:** This class initializes each person that creates an account. It allows the user to log in or log out of their account. This class works with its three subclasses and ReservationSystem.
5. **ClubCoach, ClubMember, ClubOfficer:** These are subclasses of User and they are used to represent different member types at the pickleball facility. Coach, member, and officers are the three different types of members that can be found at the facility. Different member types have different functions and responsibilities, such as making a reservation, deleting a reservation, modifying requests, etc. These subclasses work with User, ReservationSystem, and Court.
6. **Reservation:** This class is a court reservation initiated at the pickleball facility. It restricts courts to only two people, restricts members to only one reservation per day per week, allows players to reserve or cancel reservation, etc. It works with User, ReservationSystem, and Court.

7. TimeUtils: This class works with everything related to time, such as getting an hour or getting the day for a given time. It works with User, Schedule, and ReservationSystem
8. ReservationSystem: This class makes sure that everything is running properly and functioning. This class works with User, Schedule, and DataPersistence.

Class Diagrams

CRC Cards

Class: <User>	
Responsibilities	Collaborators
Have a unique username or ID	ClubCoach
Have a specific membership type in the system	ClubMember
	ClubOfficer
	ReservationSystem

Class: <ClubCoach>	
Responsibilities	Collaborators
Reserve courts for individual lessons or group training sessions during designated coaching hours	User
Have priority in reserving courts if the reservation is made more than 48 hours earlier	ReservationSystem
Reserve courts if they are still available during coaching hours	Court
Reserve courts if they are still available during coaching hours Not have reservation limits like other members	

--	--

Class: <ClubMember>	
Responsibilities	Collaborators
Play pickleball during open hours or during their reservations	User
Have a rating/skill level (ABC scale)	ReservationSystem
See the daily schedule for all courts	Court
Make reservations for court time	

Class: <ClubOfficer>	
Responsibilities	Collaborators
Reserve courts for open-play hours during designated time slots	User
Modify reservations from players or coaches upon their requests	ReservationSystem
Make, modify, or delete reservations	Court

Class: <Reservation>	
Responsibilities	Collaborators
Represent a court reservation slot	User
Have a time slot duration of 30 minutes	ReservationSystem
Have a maximum of two players per time	Court

slot	
Count as one reservation per player per day and week	
Can be deleted by the players on the reservation or officers upon request	

Class: <Court>	
Responsibilities	Collaborators
Represent a court in the facility	Schedule
Store information about reservations and availability	Reservation
	Reservation System

Class: <DataPersistence>	
Responsibilities	Collaborators
Handle the persistence of necessary data to files	User
Save and retrieve data related to users, reservations, etc.	Court
	Reservation

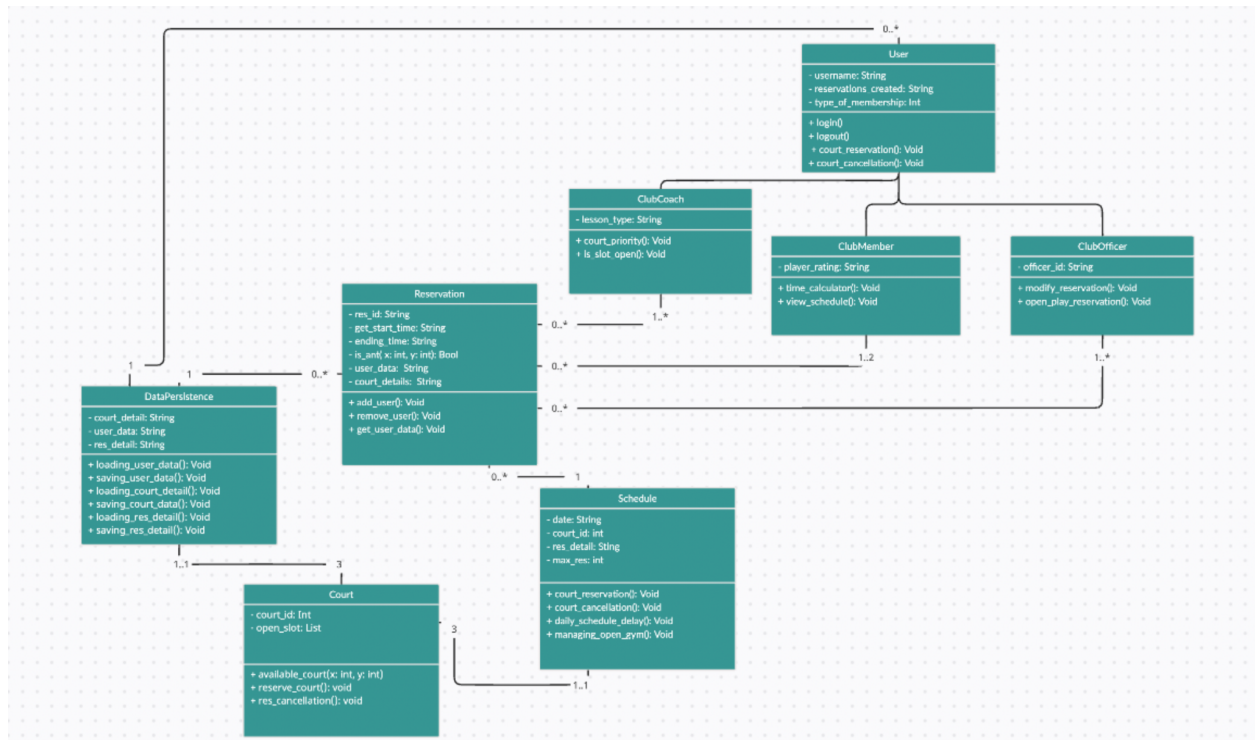
Class: <ReservationSystem>	
Responsibilities	Collaborators

Offer different features to users based on their roles	User
Enforce reservation rules (number/time of reservations)	Schedule
Allow reservations up to 7 days in advance	DataPersistence
Display the daily schedule for all courts	
Handle reservations on a first-come, first-served basis	

Class: <Schedule>	
Responsibilities	Collaborators
Display the daily schedule for all courts	User
Show the name(s) of people on each reservation slot	Court
	Reservation

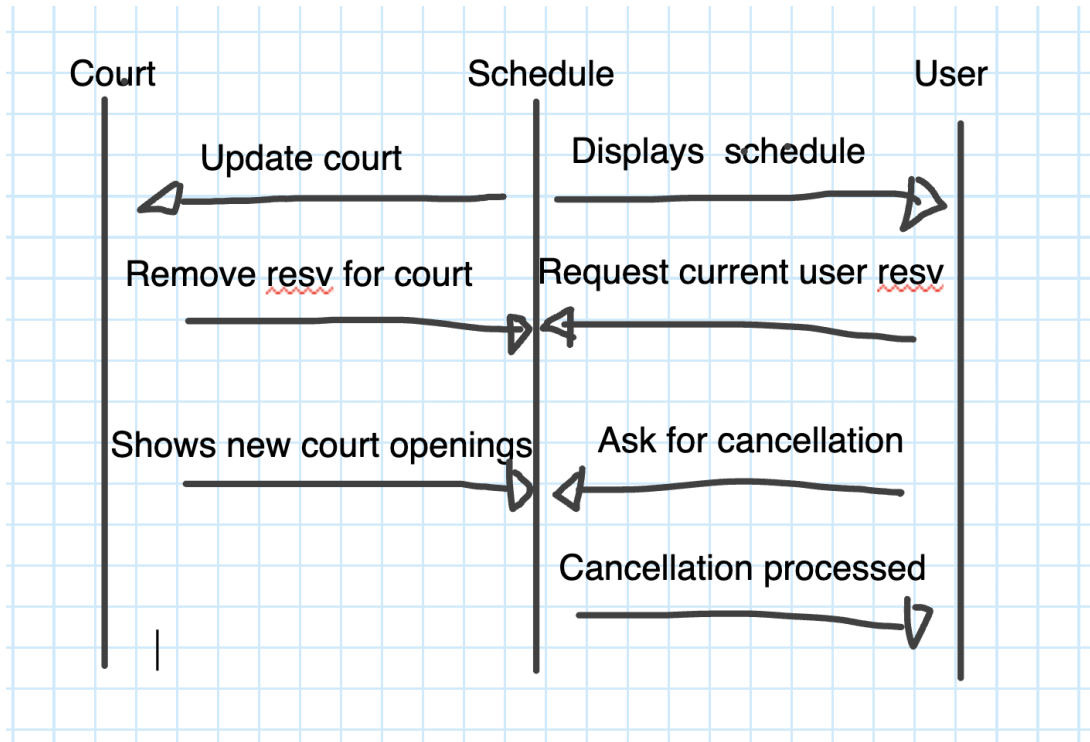
Class: <TimeUtils>	
Responsibilities	Collaborators
Converts input string dates and time to actual time and dates	User
Checks if the given data is valid	Reservation
Checks if officer or coach is available	Schedule
Retrieve information that is given by the user related to time and date	

UML Diagram

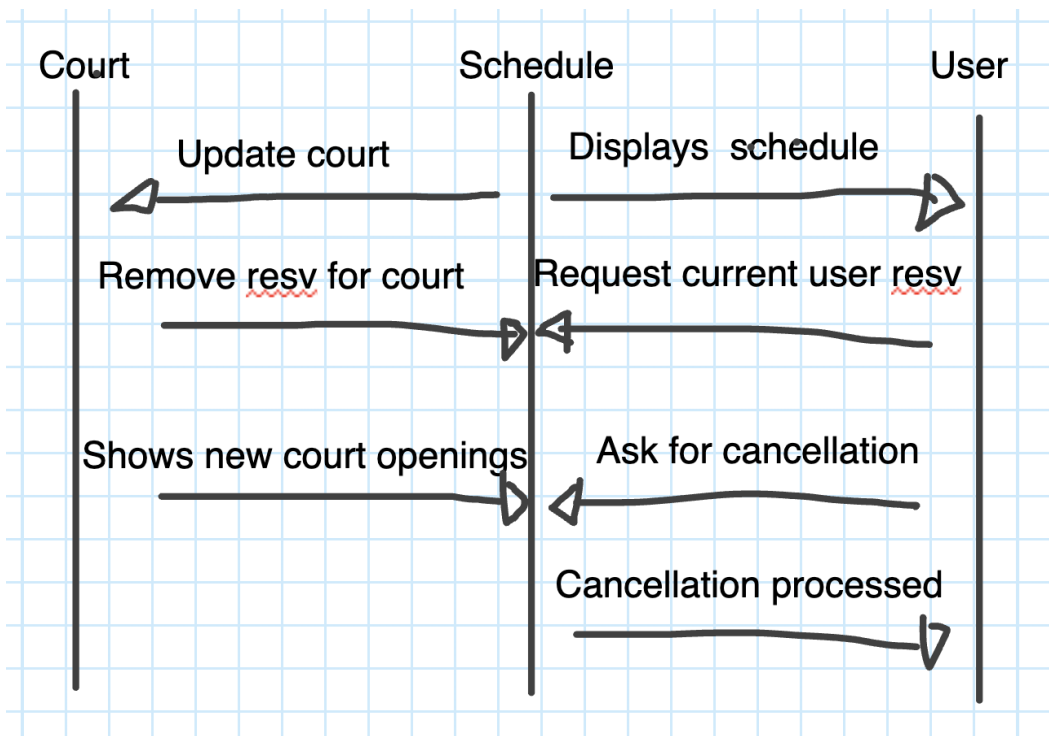


Sequence Diagram

Court Reservation



Court cancellation



Other Files For the Project

Files we included,

- Pickleball_reservation_id.txt
 - Data related to the current reservation id is stored in this file.
- Pickleball_reservation.txt
 - All data that is related to a reservation is stored in this file.
 - For example, data such as reservation time, names of users, court numbers, etc
- Pickleball_users.txt
 - All the user data that is recorded in the system is stored in this file.
 - Data such as name, member type, skill level, passwords, etc