

CSP 586 Software Modeling Dev with UML

Project Final Report

Sports Hunt

**(Sports Hub to buy sports tickets for different sports events in different cities
for different games and leagues.)**

Team 1

Lokesh, Pradeep Kumar

plokesh@hawk.iit.edu

A20449741

Wadhvekar, Ashutosh Mahesh

awadhvekar@hawk.iit.edu

A20451798

Requirements and Design

1. Brief project overview statement

Sports hunt is a sport tickets market place with tickets available for the entire sports event happening in USA. User can register and login using valid credentials to check the sports events happening in his desired location. Various sports events can be viewed based on user location, city, date range as well as type of sports. This application allows user can also check divvy stations near by the place where the sport event is happening by entering the location. User will be able to maintain his profile in which all the past sport events and the upcoming sport events to which payments are made will be displayed. In addition, the applications provide user the temperature for the sport events in his profile in a chart format. This web application also recommends users with other sport events based on user profile.

2. Requirements / Features

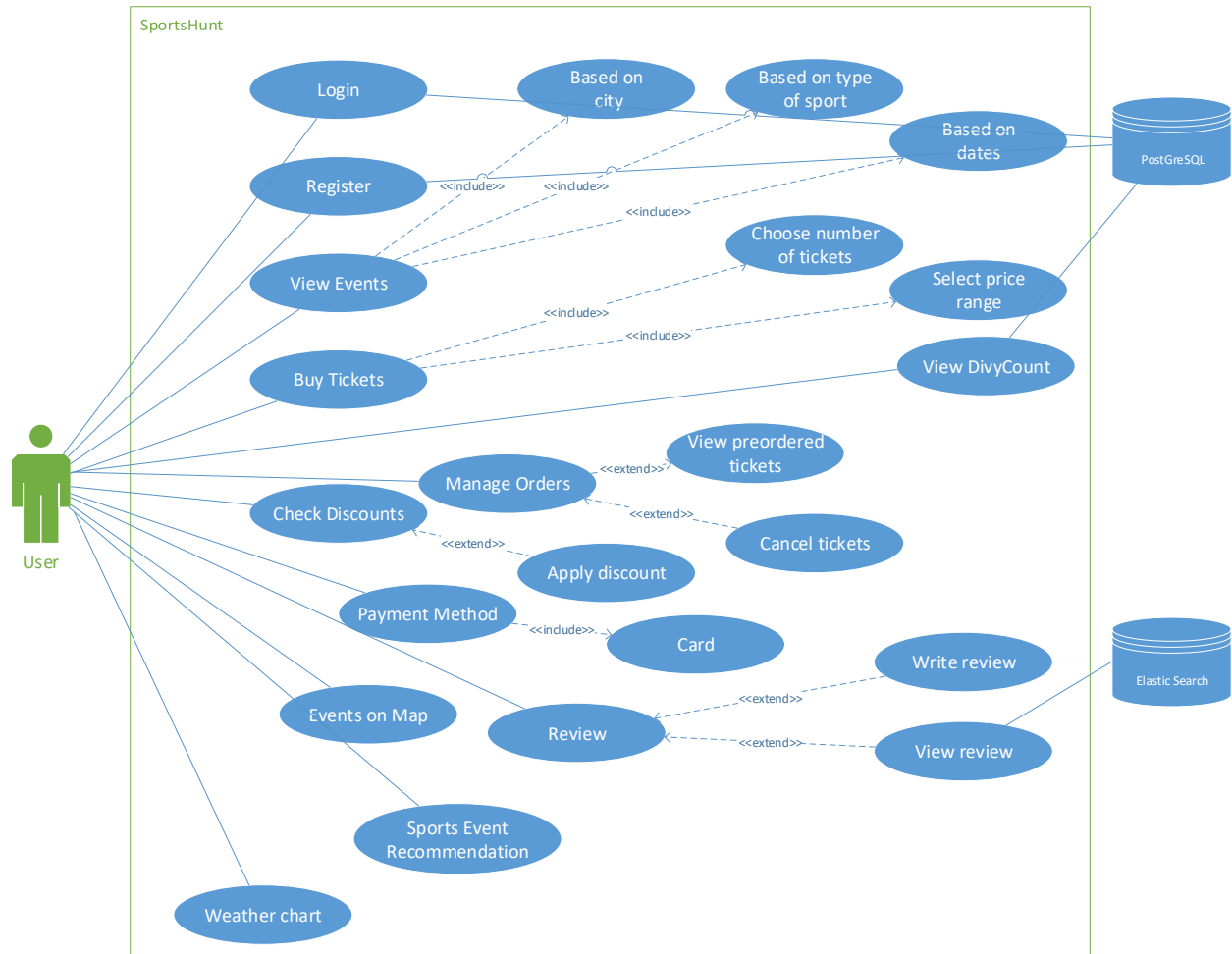
- 1) User can register and login.
- 2) User can view various sports events happening in USA.
- 3) User can view sports events based on city, type of game and date range.
- 4) User can purchase the tickets.
- 5) User can view the selected sport event details such as address, date and time of that event.
- 6) User can view the selected event on the map.
- 7) User can view reviews for the selected event.
- 8) User can select the number of tickets and also price range for the ticket while buying the tickets.
- 9) User can view the ordered or the preordered tickets in his profile.
- 10) User can give ratings and write review for the event which is present in his profile.
- 11) User can also view the event on the map and divvy stations nearby the place where the event is taking place.
- 12) User can view recommended sports events in his profile based on the ticket orders he had made earlier.
- 13) User can view a five day weather forecast on a chart for all the events at different cities he has in his profile.

3. Use cases and use case diagrams

Use Cases

Sr. No.	Use Case	Description
1	Register	➤ User has to register can register
2	Login	➤ User has to login in order to view the website
3	View Events	➤ User can view different type of sport events based on a) Type of sport b) City name c) Date range
4	Buy Tickets	➤ User can select a) Number of tickets b) Sort tickets by price
5	Manage Tickets	➤ User can view pre-ordered tickets
6	Payment method	➤ User can pay by card
7	Review	➤ User can write review ➤ User can view reviews
8	Divvy stations	➤ User can view the nearest Divvy stations on Google maps
9	Event on Map	➤ User can view the event which he is interested on Google maps
10	Sports Recommendation	➤ User will be able to receive recommendation based on the events in his profile
11	Weather Chart	➤ User will be able to view the weather forecast for 5 days in a form of line chart for all the events.

Use case diagram

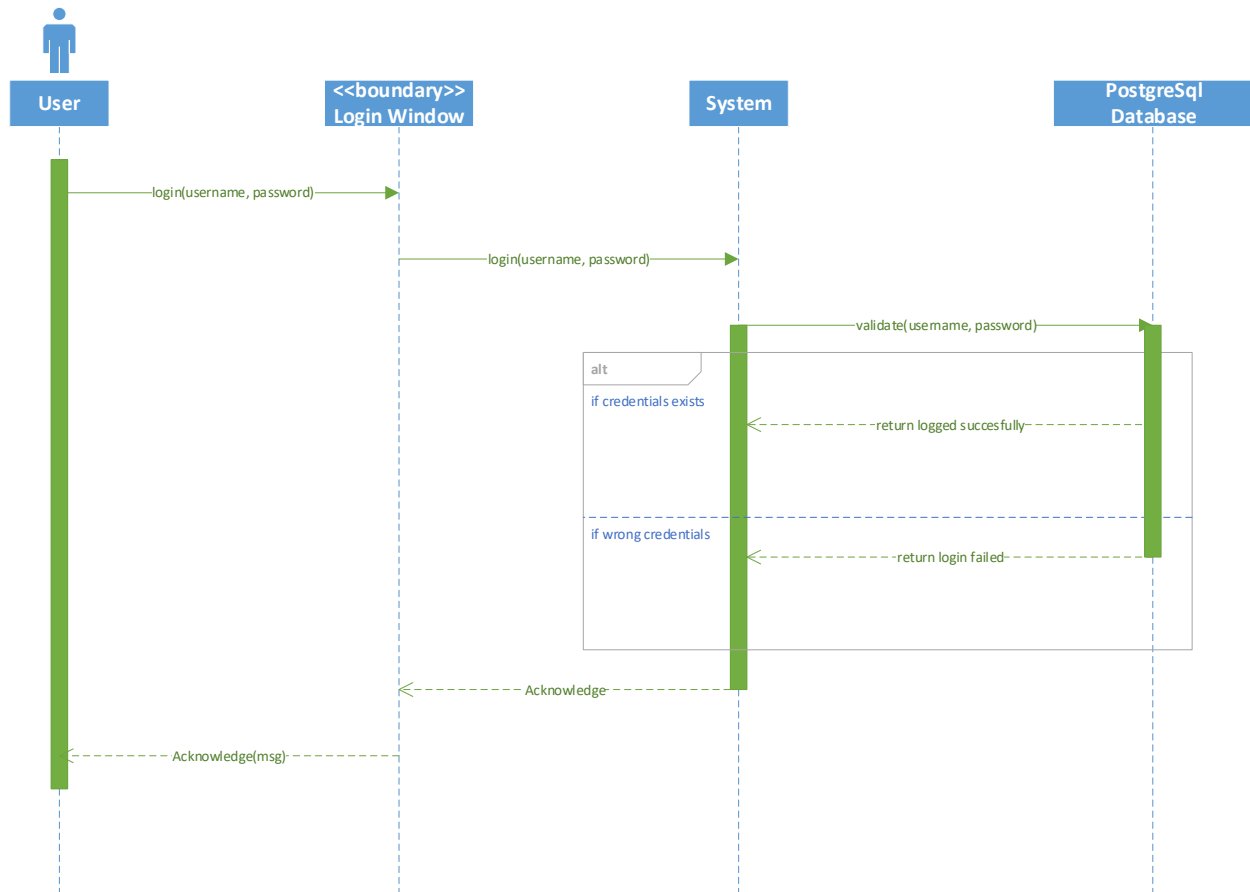


4. Activity Diagram

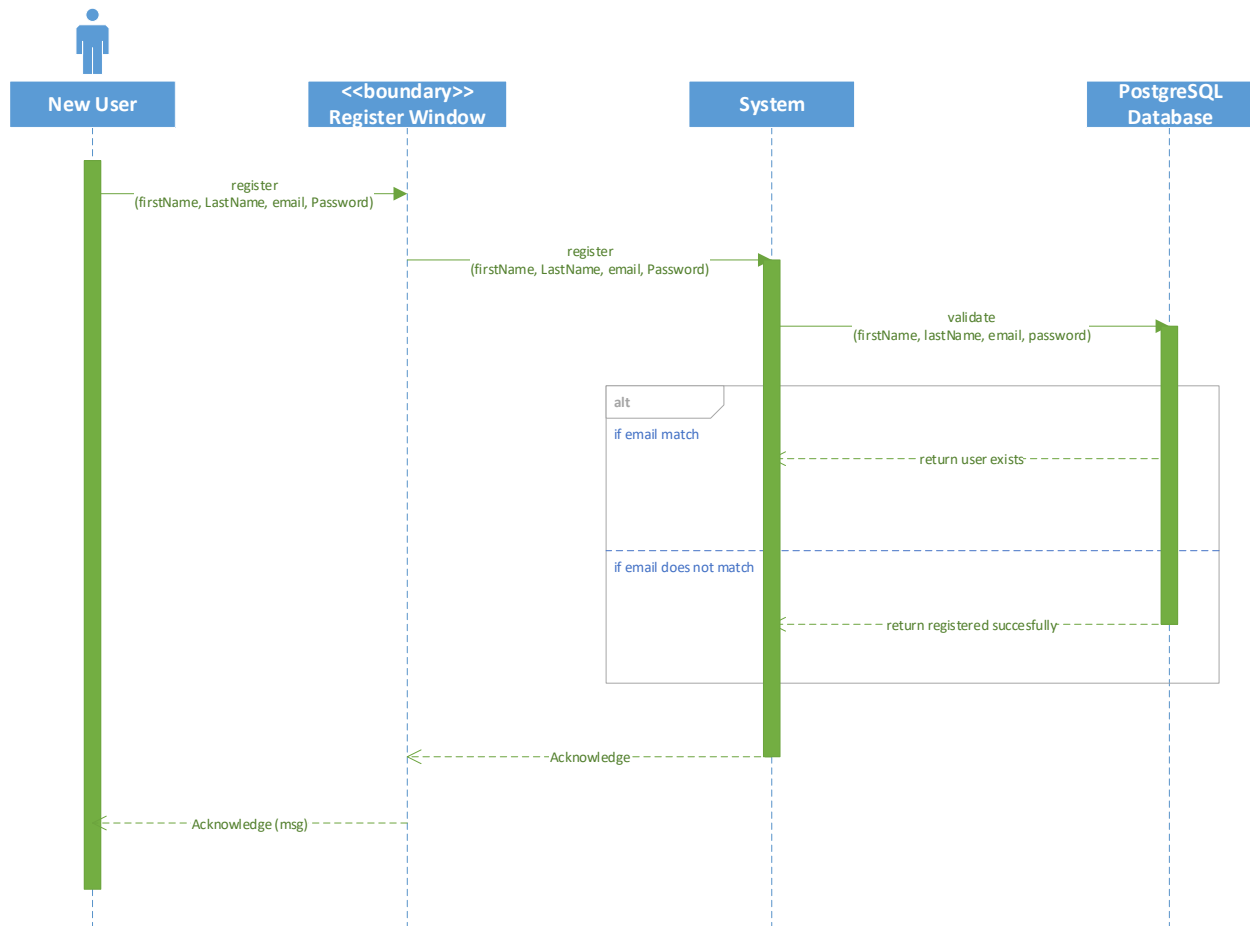


5. Sequence Diagrams

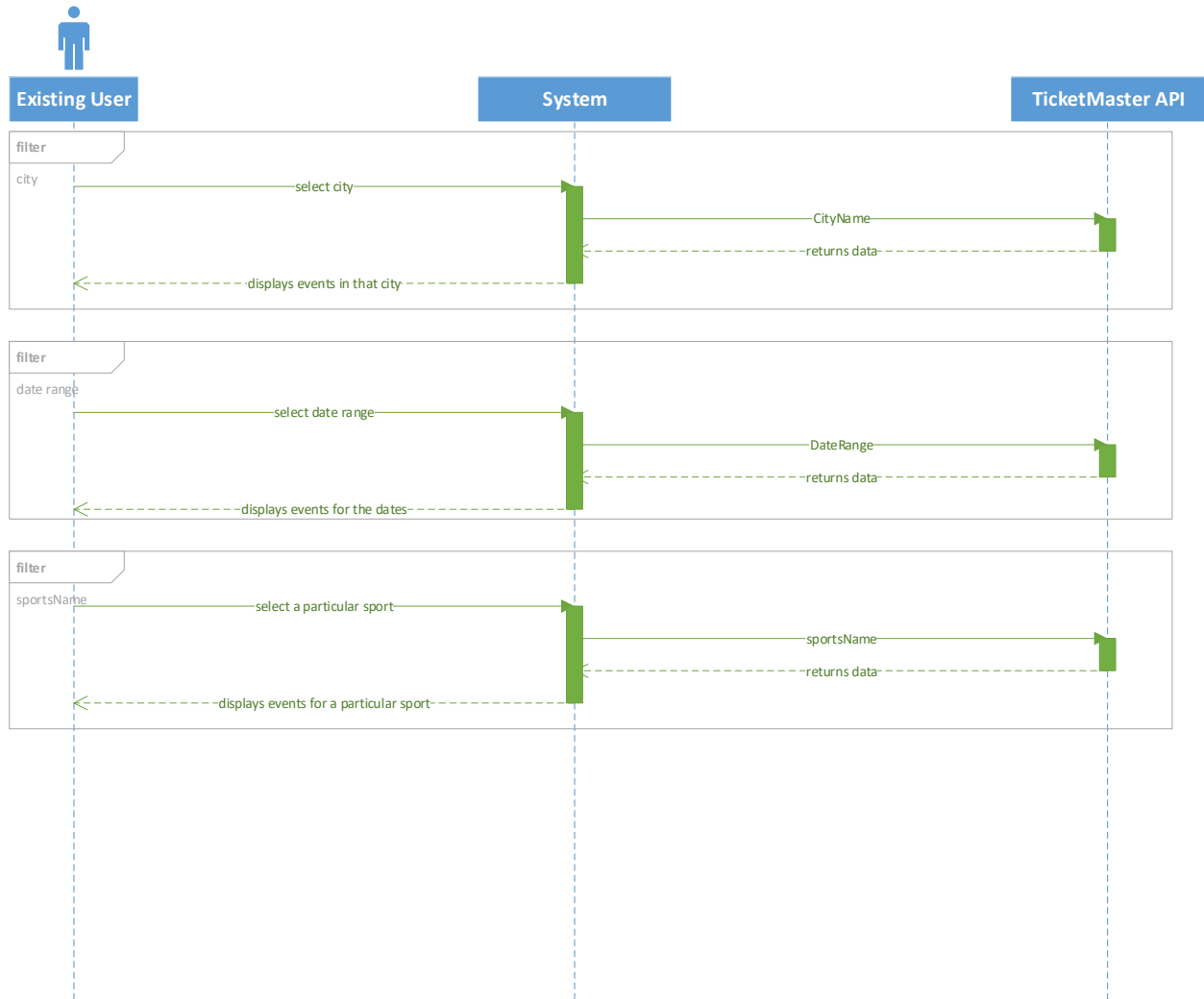
a. Sequence diagram for Login



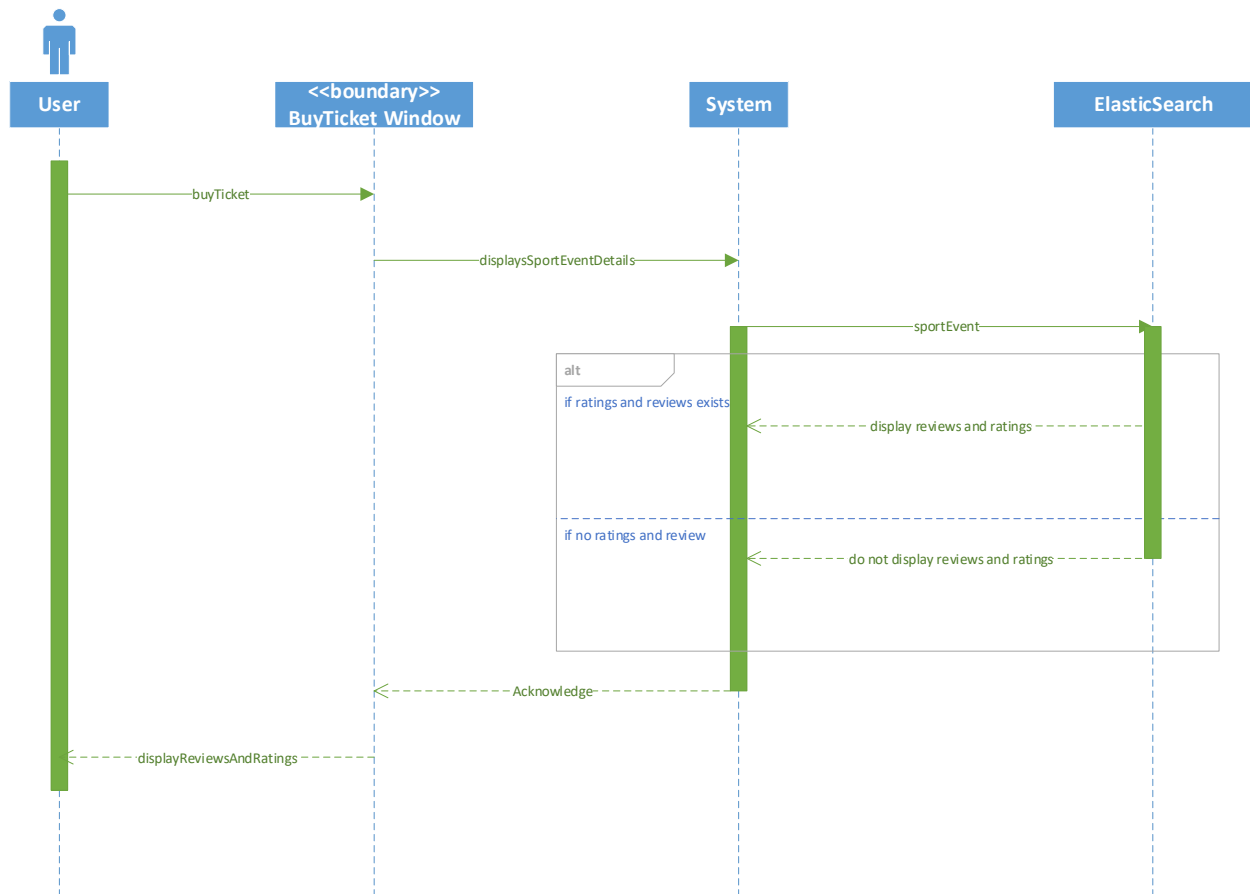
b. Sequence diagram for Registration



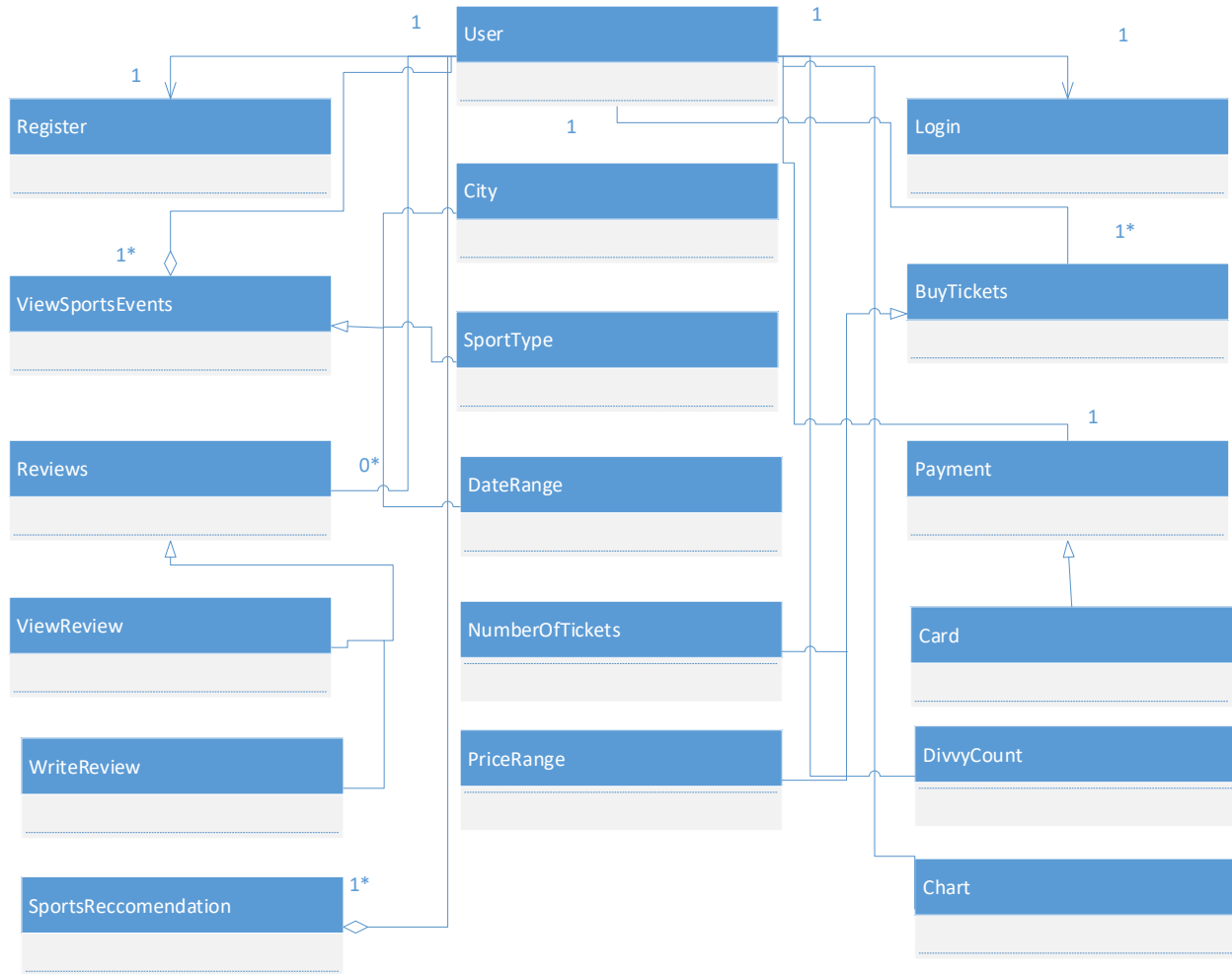
c. Sequence diagram for View Events



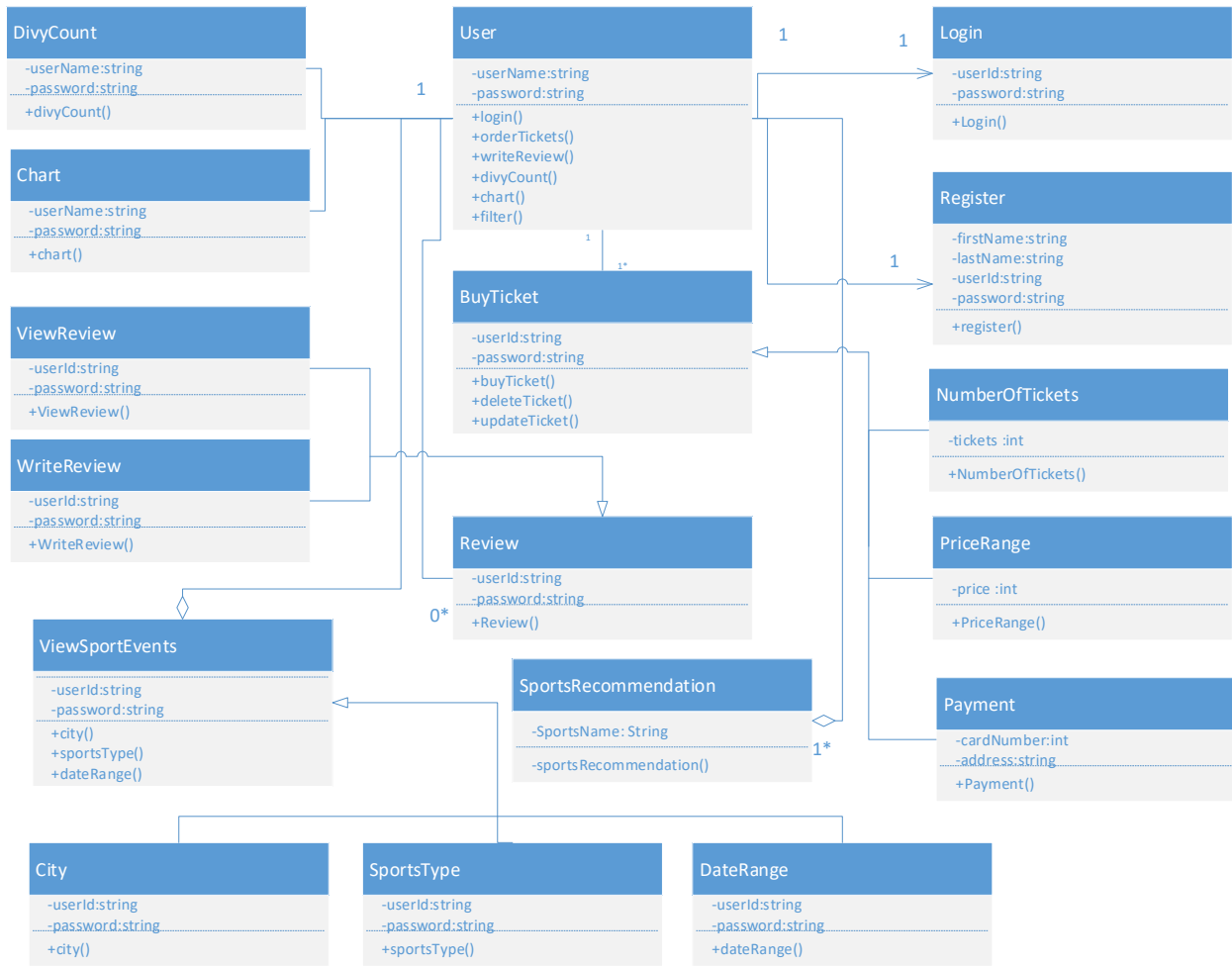
d. Sequence diagram for View Ratings



6. Domain Model Class Diagram



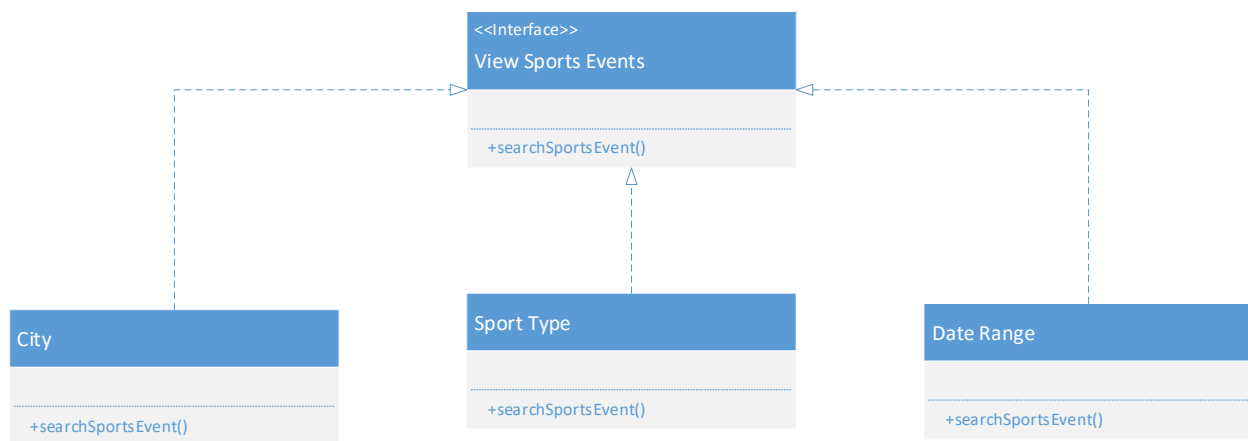
7. Design Model Class Diagram



8. Documentation and class diagrams for Design Patterns used

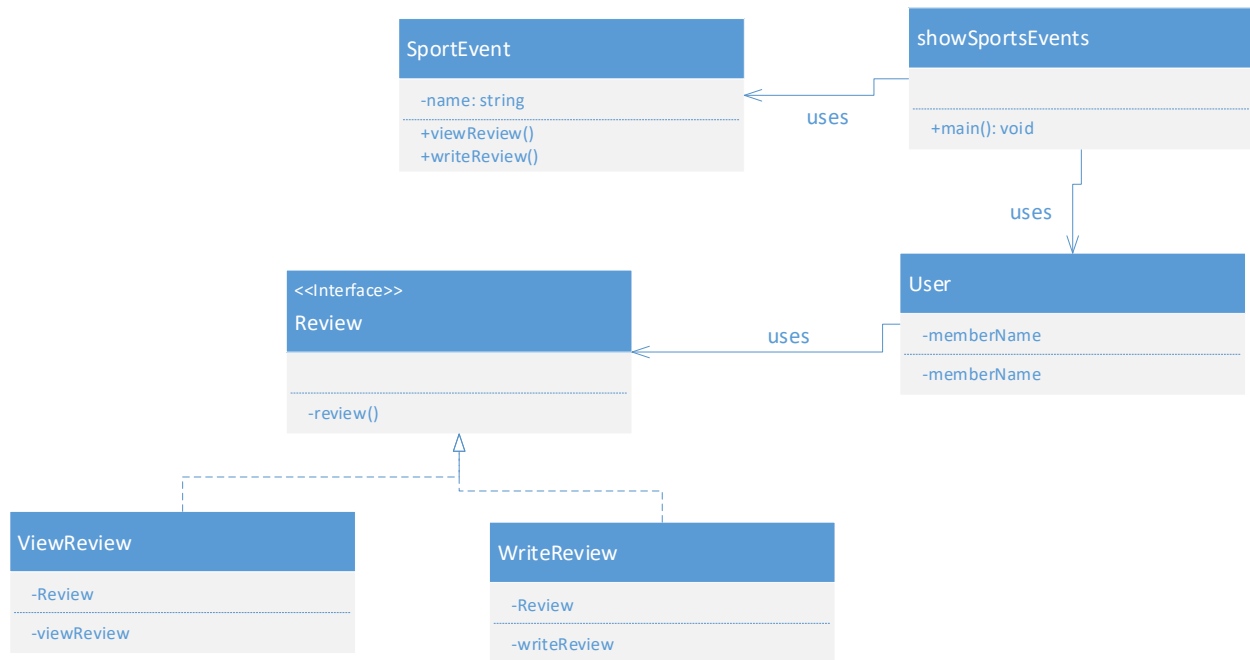
a) Strategy design pattern

In Strategy pattern, we create objects to represent various strategies and a context object whose behavior varies as per its strategy object. We have created a View Sports Events interface defining an action and concrete strategy classes implementing the View Sports Events interface.



b) Command pattern

Command pattern is a data driven design pattern where a request is wrapped under an object as command and passed to invoker object. Review is an interface which is acting as a command. SportEvent which acts as a request and concrete command classes ViewReview and WriteReview. For implementing Review interface which will do actual command processing.



c) Singleton design pattern

A singleton is a type of factory pattern where one and only one instance of the object can exist irrespective of the number of times the object is instantiated. The singleton design pattern is used for selecting Divvy docking stations of sports hunt. This is useful when exactly one object is needed to coordinate actions across the system.

