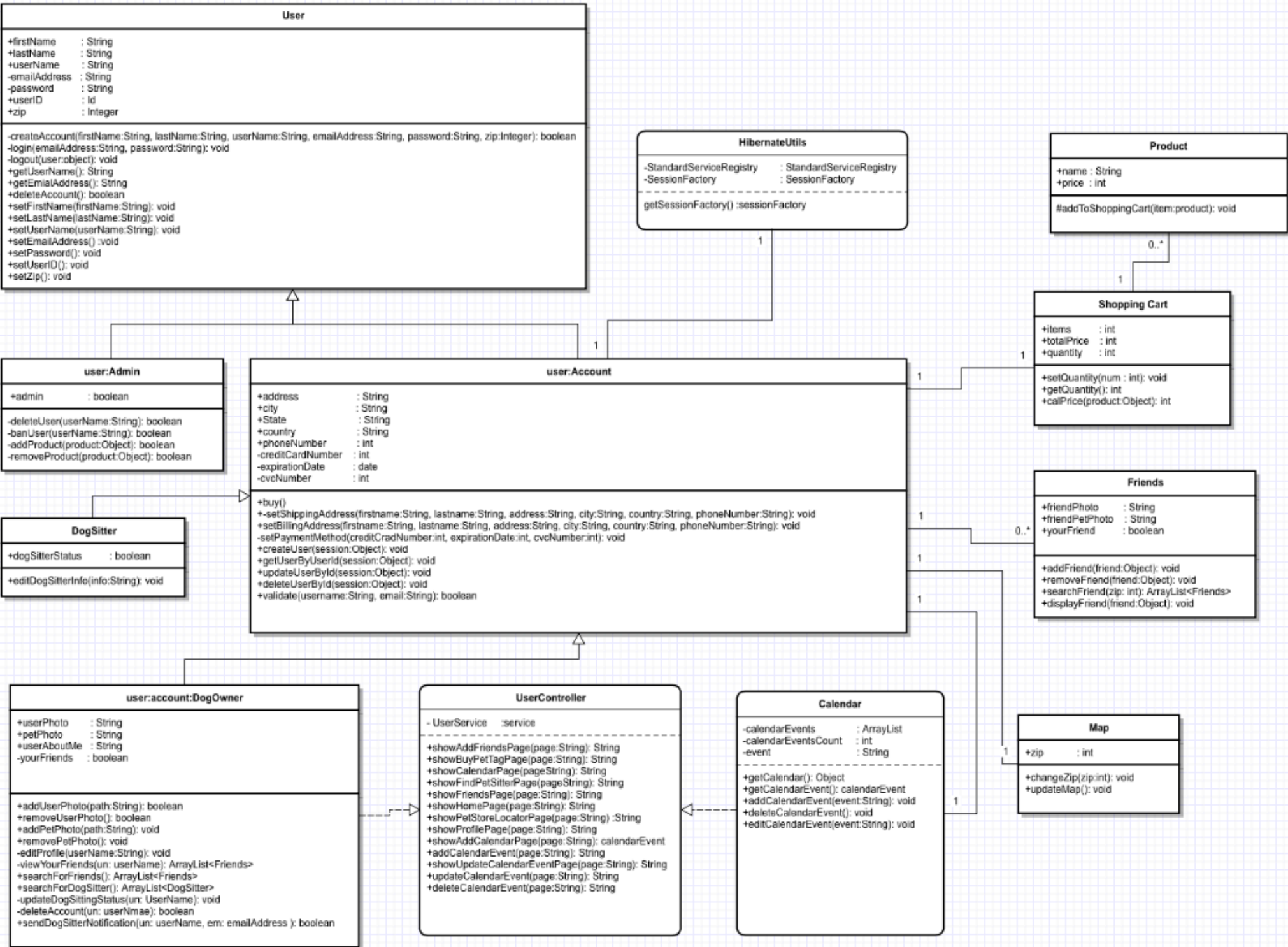


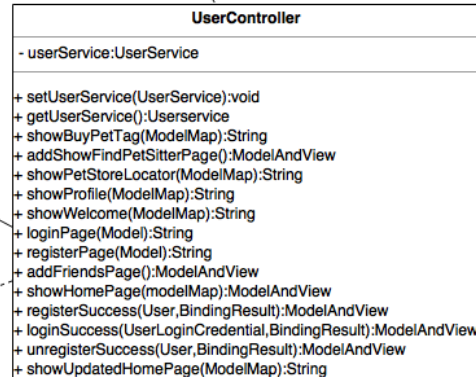
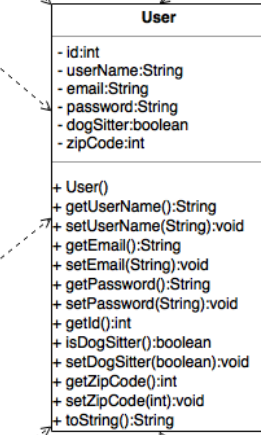
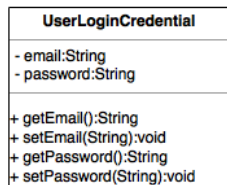
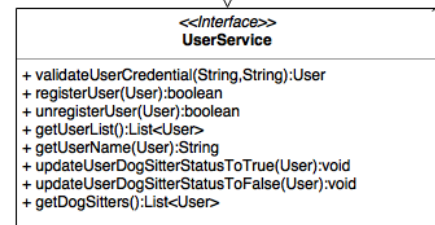
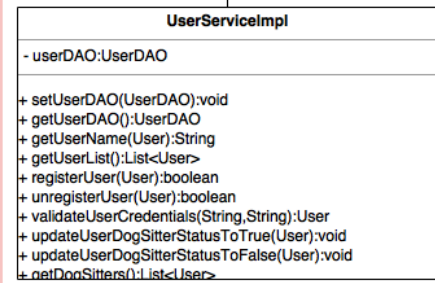
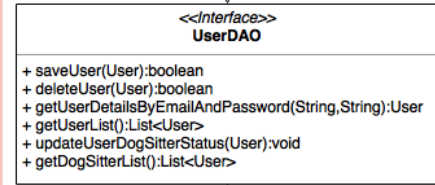
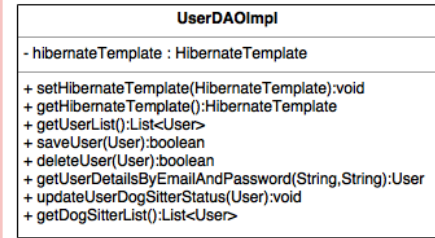
## **07\_Rover&Friends\_Part3**

**Project Part 2 class diagram on next page.**

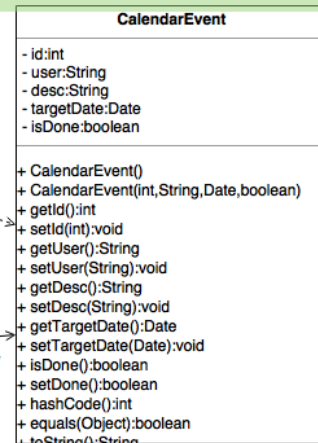
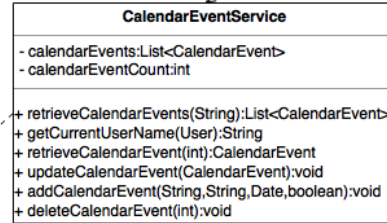
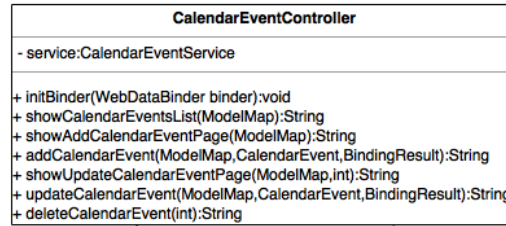


**Project Part 3 class diagram on next page.**

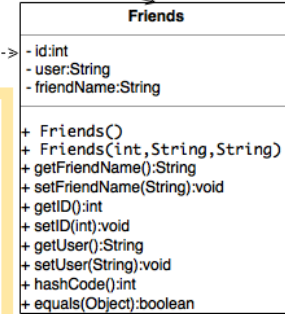
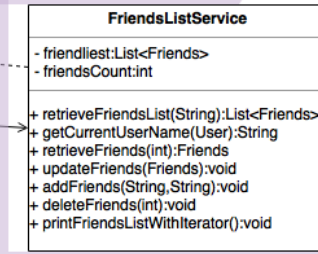
## DAO design pattern



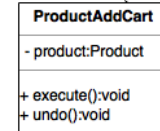
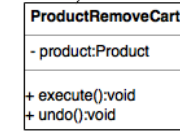
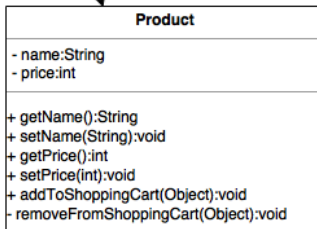
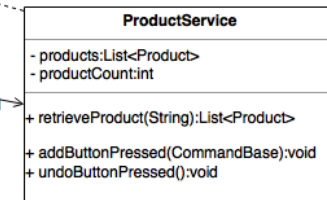
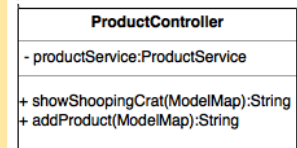
## MVC design pattern



## Iterator design pattern



## Command design pattern



For part three we refactored our code to implement the DAO(Data Access Object) design pattern due to us using Java EE with a relational database. DAO is used to separate low level data accessing from our user services. We also used MVC design pattern since we're using spring MVC for our web application to separate application concerns. From the GoF design patterns we are using the Iterator pattern and the Command pattern. However, for the Iterator pattern we simply imported Iterator to access elements in our collections. We used Command pattern to execute and undo our methods for adding products to our shopping cart. The patterns are shown in the part 3 class diagram above.