Citrus Application

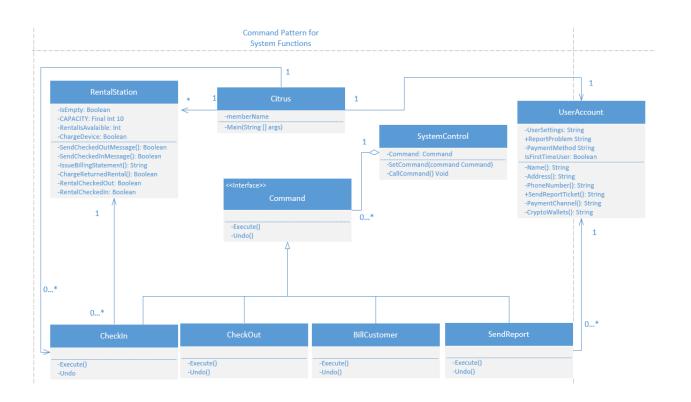
UML and Sequence Diagrams

Team: Christian Ndaye | Michael Anthony | Adam Moritz

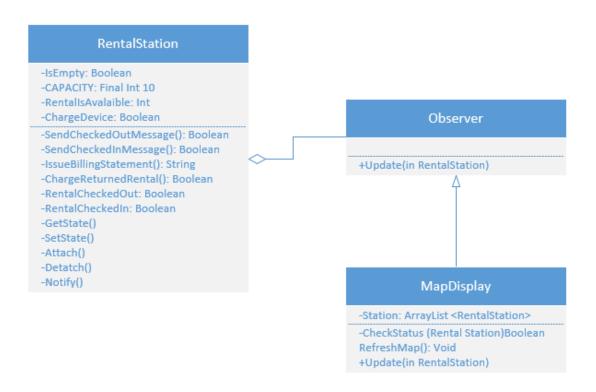
Enclosed are three pattern design: diagrams for a <u>Factory Method</u> for bike and scooter creation, <u>Command Pattern</u> for issuing billing to customer, reporting problems with rentals, checking the bike in and out along with charging the rentals. Lastly, an <u>Observer Pattern</u> to refresh the map with new relative data for the users. Along with sequence diagrams for Application processes.

Factory Method

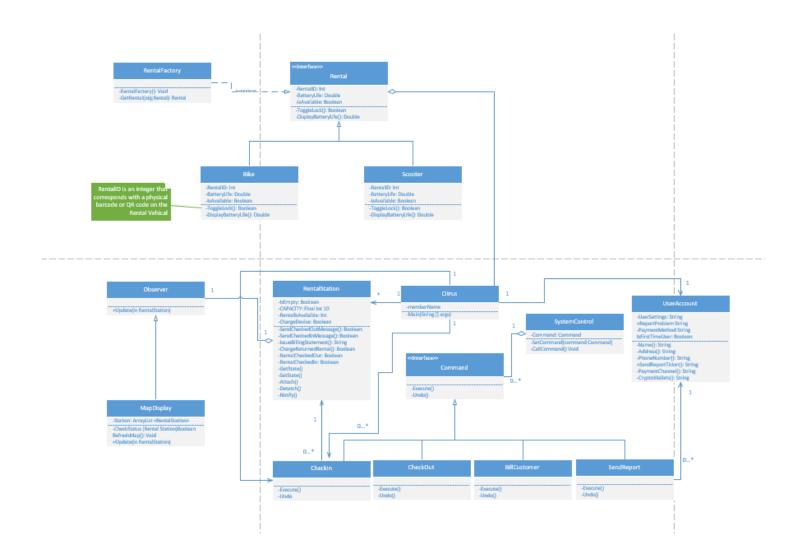
For Rentals -RentalID: Int -RentalFactory(): Void -Instantiates--BatteryLife: Double -GetRental(obj;Rental): Rental -IsAvailable: Boolean -ToggleLock(): Boolean -DisplayBatteryLife(): Double -RentalID: Int -RentalID: Int -BatteryLife: Double -BatteryLife: Double -IsAvailable: Boolean -IsAvailable: Boolean -ToggleLock(): Boolean -ToggleLock(): Boolean -DisplayBatteryLife(): Double -DisplayBatteryLife(): Double RentalID is an integer that corresponds with a physical barcode or QR code on the



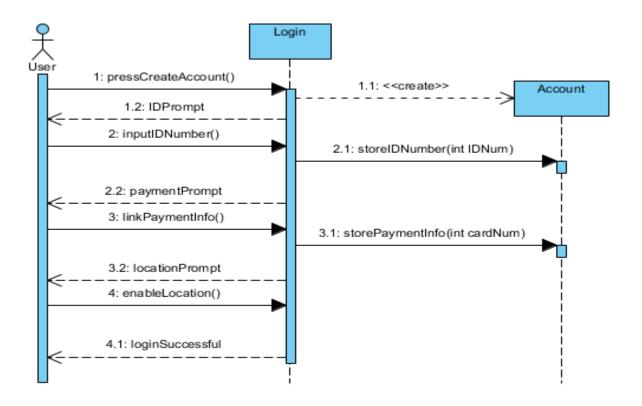
Observer Pattern for System Updates



Citrus UML Diagram



Sequence Diagram for Login Process



Sequence Diagram for Ride in Progress

