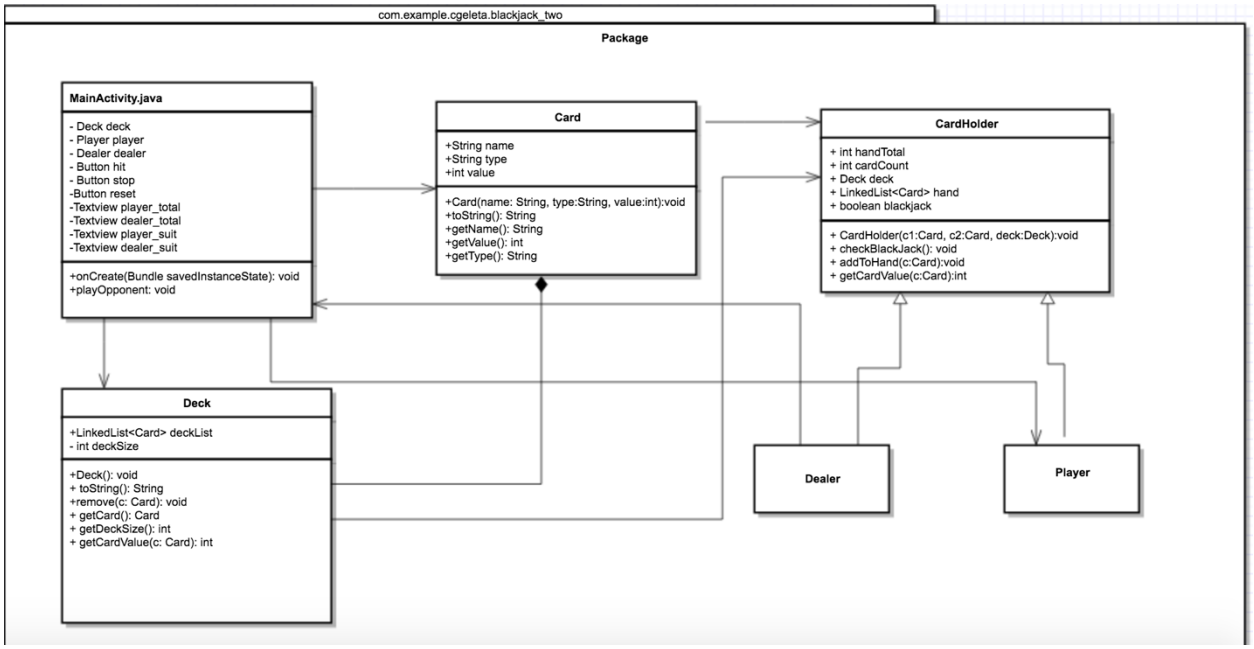
Christopher Geleta
CS482.01
Software engineering
Blackjack App

UML:



Brief design summary:

The backend of the blackjack app is a simple java OOP project using inheritance. All the main objects in this problem have a class, such as dealer, player, the deck, and a card. I needed to make 52 cards so I made a class for that and it would hold the suit, name, and value for that card. I then needed somewhere to store those cards. I thought a list would work. I made a class to represent the object that would store 52 cards i.e. a deck. So I made a deck class this deck class had to be able to randomly get cards and tell me the value of the cards. So I gave it that functionality. I then realized the dealer and the player do the same stuff so why write the code twice? So I made one class that is a cardholder and I had the Dealer and Player implement that class so they had the same functionality and I saved time coding. I then needed to create the game play. I made three buttons. One for hit one for stop and one for new game. The player could hit those buttons to play the game. But before those buttons are pressed there was some set up. Two cards draw the player and dealer each and check for blackjack. So I added that functionality. Then the user was free to play until he lost. When pressing the hit button a card is drawn and when pressing the stop button the dealer plays and tries to win. The hand of each player and the hand value is displayed to the screen in textviews and is updated as the game

plays on and cars are drawn. When a someone loses or wins the app displays an alert box and disable the hit and stop button so they must restart a new game by hitting the restart button.