```
//------------------------------------------------------------------------
//
//-------------------------- SIMULATED_WAR---------------------
//
//------------------------------------------------------------------------
//
// Program Name: card Class Header
//
// Author        : Mosfiqur Rahman
// Date          : May, 2016
// Last Modified: 19th May, 2016
```

- **System Manual:**
  - The system uses the card class (see card.h and card.cpp files)
  - The current system uses the following methods and attribute from the card class:
    - Private [***Attribute***]:
      - **Rank myRank**
        - ✔ suitable data structure for flawless programming operation
      - **Suit mySuit**
        - ✔ suitable enum data structure for flawless programming operation
    - Default [***Constructor***]:
      - **card(Suit s, Rank r)**: it initializes the card's suit and rank
    - Other methods: gets & sets the values of specific attributes
      - **Rank getRank() const** :
        - ✔ gets the rank of a card
      - **Suit getSuit() const**:
        - ✔ gets the suit name of a card
      - **ostream & operator<<(ostream& os, const card& c)**:
        - ✔ overloads the output operator
      - **bool operator==(const card& lhs, const card& rhs)**:
        - ✔ overloads the comparison operator
      - **bool operator<(const card & my_rank, const card & my_suit)**:
        - ✔ overloads the less than operator
      - **bool operator>(const card & my_rank, const card & my_suit)**:
        - ✔ overloads the greater than operator

// Program Name: deck Class Header
//
// Author : Mosfiqur Rahman
// Date : May, 2016
// Last Modified: 19th May, 2016

- **System Manual:**
  - The system uses the card & deck class (see card.h, deck.h, deck.cpp and card.cpp files)
  - The current system uses the following methods and attribute from the card class:

    - ➢ Private [***Attribute***]:
      - ◆ **deque<card> myCards**
        - ✔ suitable data structure for flawless programming operation

    - ➢ Default [***Constructor***]:
      - ◆ **deck()**: it initializes a deck

    - ➢ Other methods: gets & sets the values of specific attributes
      - ◆ **void shuffle()** :
        - ✔ shuffles the cards of a deck

      - ◆ **card dealCard()**:
        - ✔ deals a single card from top of the deck

      - ◆ **deck dealCards(int n)**:
        - ✔ deals a stack of n cards from top of the deck

      - ◆ **size_t size() const**:
        - ✔ finds the number of cards in the deck

      - ◆ **void addCard(card c)**:
        - ✔ adds a new card to the bottom of the deck