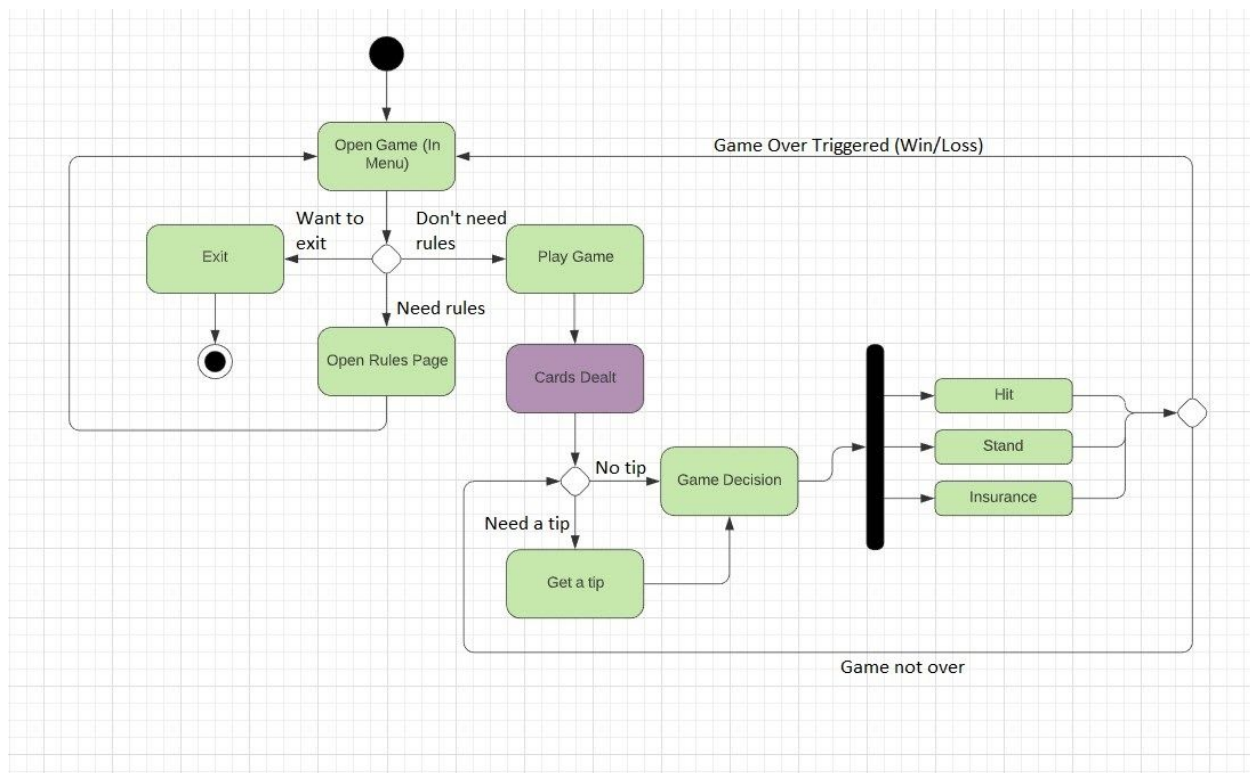


## 3.4

**Group Member:** Jacob Lussier

### Introduction

The following diagram will be based on the 3.4 requirement graphic user interface outlined in the team's software specifications document. In order to model all the interactions possible with the GUI, an activity diagram will be suitable. The main parts of the graphic user interface are the menu, the main game page, and being able to exit when desired. Also, each of these main sections has other stimulus/response sequences and decisions associated with it. These sections all act as separated "activities" which will be modeled in the diagram.



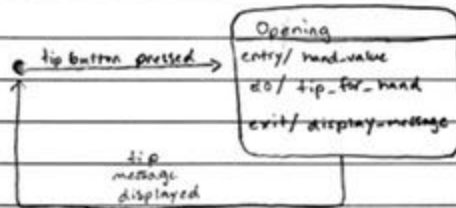
## 3.6

**Group Member:** Marcus Comolli

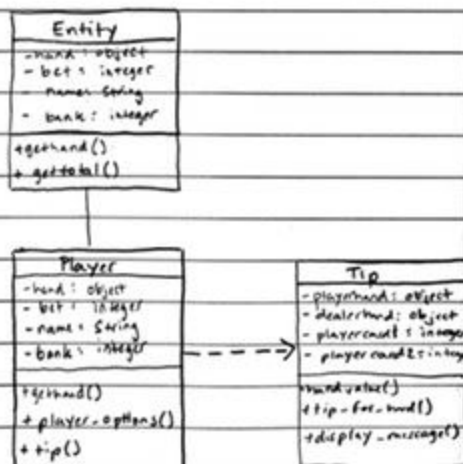
### **Introduction:**

My feature of our blackjack game will fit in as shown in my diagrams and in section 3.6 of our SRS. It is a necessary feature, but one that many blackjack games do not offer. It is a hint when the player needs it or wants to learn the best move for that situation. Maybe the player is trying to become a better blackjack player by practicing on our game and my tip class will help them with each situation in the game and give them better insight when they are playing at a casino. My class fits into our program with a button on our GUI. Just like the other player options like hit and stand, it will be displayed on the same menu. When the button is pressed, the necessary information like the players hand and dealers' face up card will be sent to my class. My class will then use a series of if statements to check the dealer's face up card to all pairings of cards the player could possibly have. When it finds the correct matching, it will print a message like "Hit" or "Stand". This will give the user a hint that may or may not help him win that hand.

## State Diagram



## Class diagram

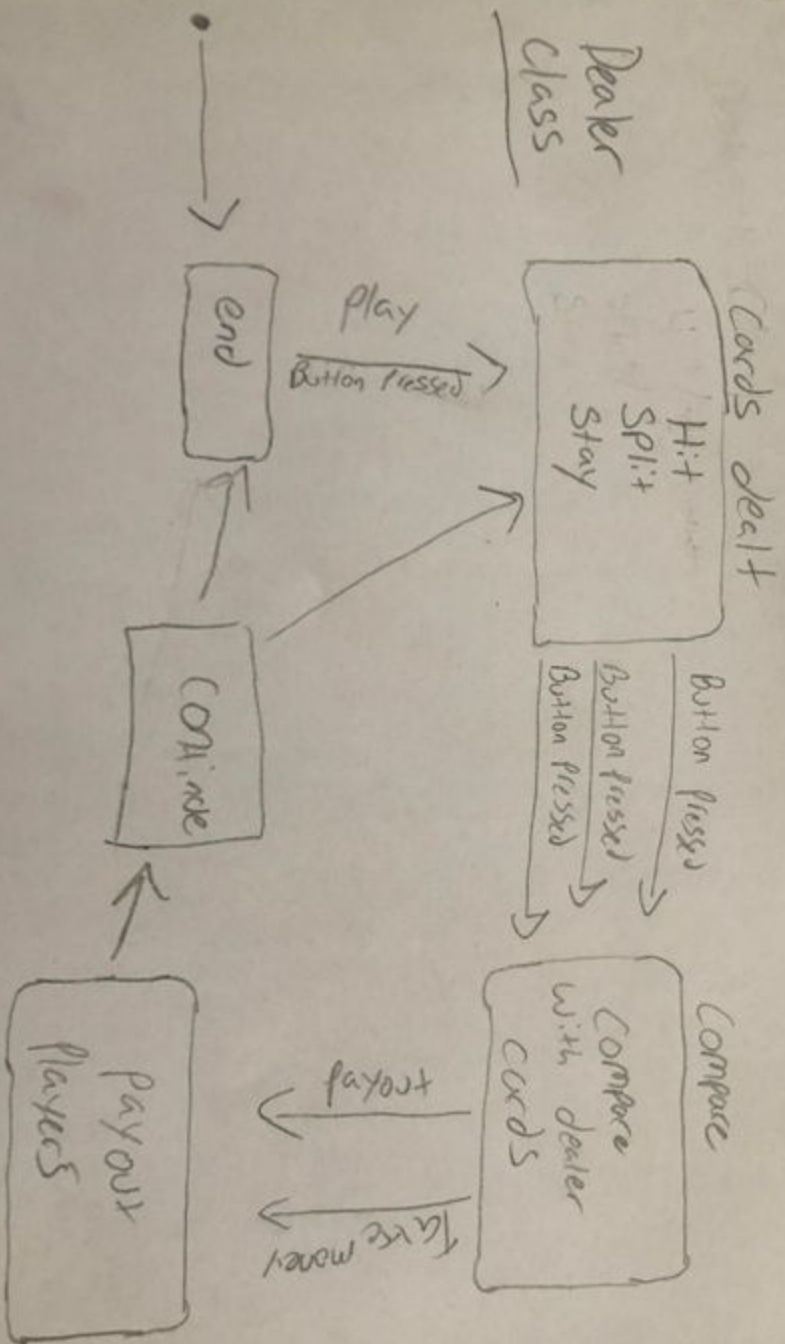


## 3.7

Group Member: Denzell Rodriguez

### **Introduction:**

The Diagram below/attached is based on the 3.7 requirement, dealer class outlined in my team's software specification document. In order to correctly represent this class and all of its interactions with all of the other classes a state diagram is necessary. The main interactions of the dealer class are the player actions, dealing a card to each player when the game starts and then giving the correct action based on the players choice, hit, split, or stay.

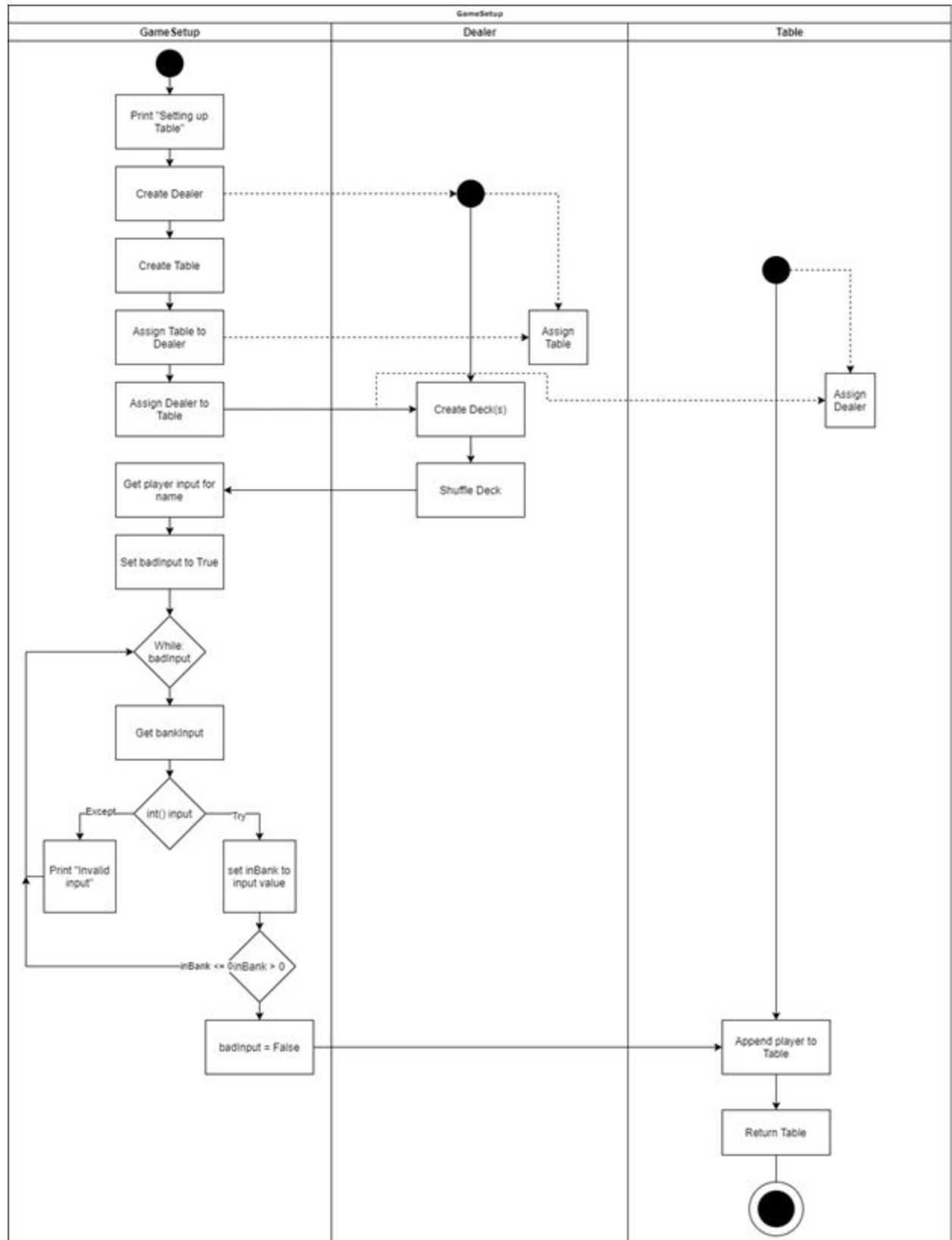


## 3.7

**Group Member:** Cory Cothrum

### **Introduction:**

My diagram will detail the initial setup of a game in our blackjack program. One of the design goals in mind was to make a game of blackjack as naturally and close to actual blackjack as possible, something which is actually very compatible with class based programming. This means that a dealer handles the deck, is assigned to a table and vice versa, and that table holds players. The deck is an actual 'deck' made from a doubly-linked list which allows for shuffling and drawing cards very analogously to real life. The players can input their information at this point, and the game can begin.



## 3.7

**Group Member:** John O'Donnell

### **Introduction:**

My design details the implementation of the NPC class detailed in Requirement 3.7. The NPC is a specification of the Player class, which is in turn a specification of the Entity class. The NPC class contains the same private members as the player class, except for the addition of a boolean Stand, which determines whether the NPC has decided to stand and can no longer hit. The NPC will periodically use the Dealer class, and more specifically the Dealer's hand, when it is determining whether it should hit or stand.



