**Status Summary**

Team members: Chirag Telang, Michael Truong

Title of the project: Object Oriented Texas Hold'em

**Work Done**

In our first two weeks of code development, we have mainly focused on implementing the card game portion of our project – as that functionality is of top priority. Once that functionality is complete, we can start branching out into the other features of the project – such as Sign Up/Sign In, Join Game, Leave Game, etc. As of the creation of this document, we have finished coding the Player, Poker, and Card classes which control the actual instance of Texas Hold'em once a lobby is started. Currently, we are working on our Main.java file, which features a menu that the user can iterate through depending on what action they want to do. Features implemented include: Bet, Check, Fold, Join Game.

*Work Breakdown:*
Chirag - **User.java, TexasHoldem.java**
Michael - **Card.java**
Combined Work - **Observer.java**

**Changes or Issues**

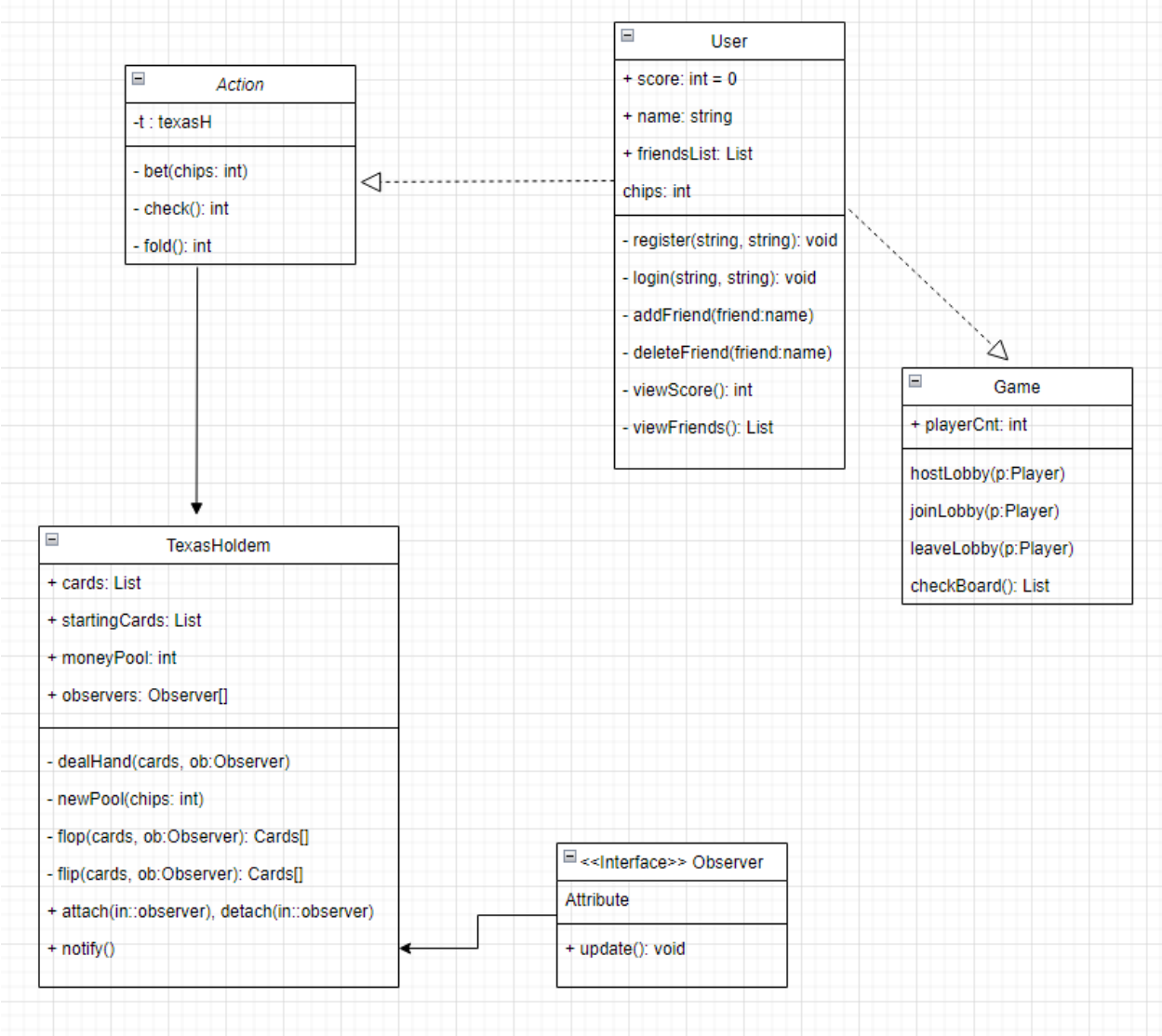
Initially, we planned on having a lobby that you could invite your friends to join. Judging by what classes we have implemented so far, we may not be able to incorporate some of the lobby features we had originally planned to include in our project design document – due to time constraints and general lack of familiarity with online applications and java as a whole. Nevertheless, though, we will certainly attempt to deliver the final project as detailed previously with all the functionalities listed.

**Pattern Use:**

*Observer* - we have implemented the observer design pattern in the Player.java class to rate each player's hand and determine the winner. Additionally, if a player chooses to fold, they will be able to see the likelihood of who will win based on the Observer's calculations. The use of an observer here is definitely well warranted – as the best hand in a particular game is always subject to change due to how the game itself is designed.

We have yet to implement the three other design patterns we plan to use (Singleton, Iterator, and Decorator). We are having trouble fitting these patterns into our current code base, but as we continue to progress through this project, we are sure we'll find suitable instances where the use of these patterns is needed to streamline the backend of our application. One idea we had was to implement the Iterator design pattern in how the cards are represented and dealt, since they will be handled sequentially.

**Class Diagram**




**Plan for Next Iteration**

We have quite a bit of features left to implement as stated above but nonetheless, we will attempt to deliver the project with the features we mentioned in the Project 5 writeup. The features we

still need to implement include: Sign Up, Sign In, Leave Game, Add Friends, Invite Friends, Score Tracking, Player Names.

We don’t believe that these features will be difficult to implement in the next coming weeks but it will take time. As we are both very busy students with finals coming our way, we plan on working over Thanksgiving break. By the date of 12/7, the project should be completed with all of the functionalities listed within Project 5 writeup.