Project 7: Semester Project – Final submission
Texas Hold'em Poker Game Assistant

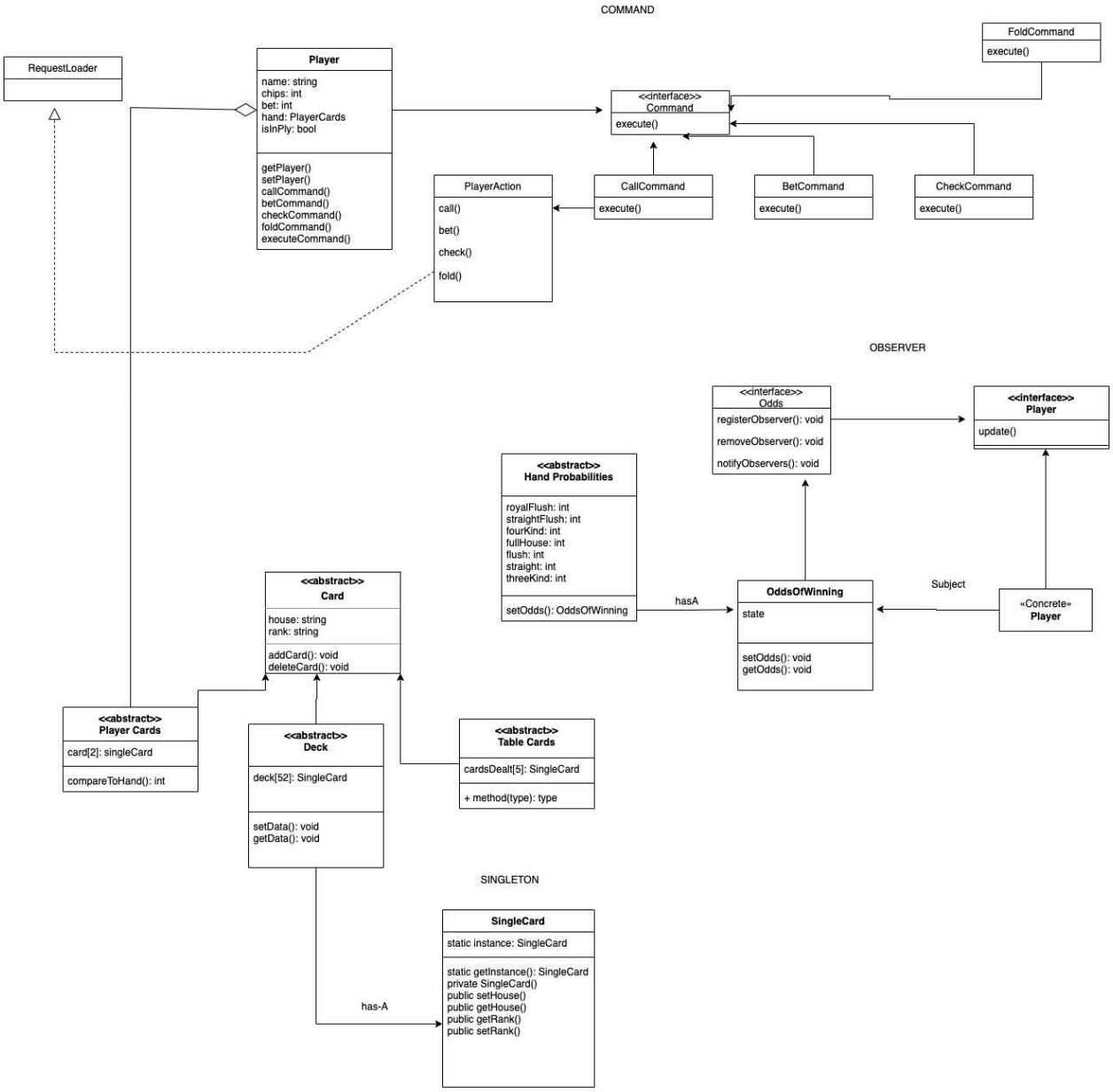
Christian Jaramillo
Michael Grodecki

Final State of System Statement

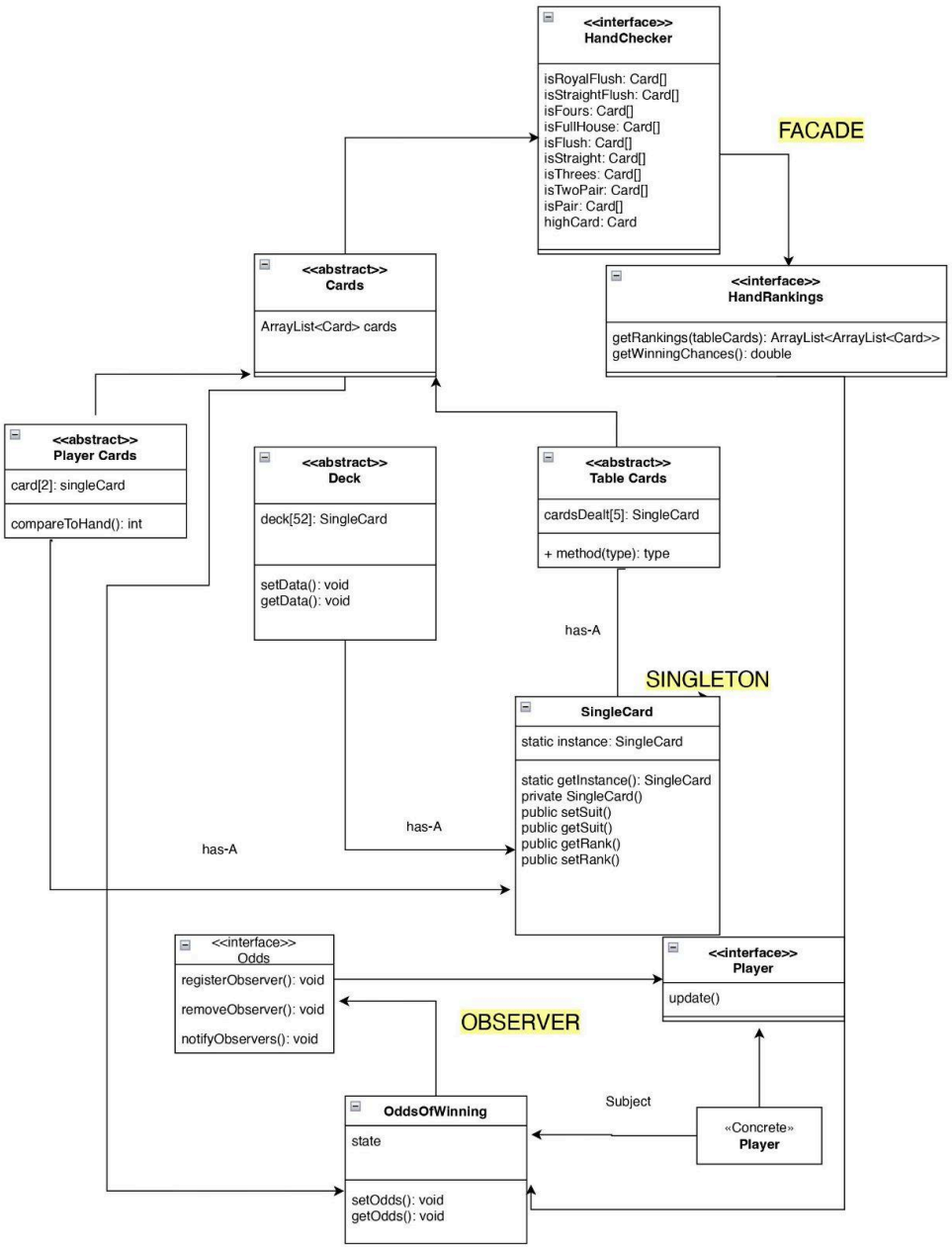
Several updates were made to the system for this final project submission. The user interface was updated and now users can add cards to the display. Based on which cards are added, the webpage will display the user's winning chances. The UI page was redesigned to improve user interaction. The command design pattern was removed and a facade pattern was implemented which wasn't in the original plan. The observer and singleton patterns remained. All card probability calculations were implemented with significant debugging and testing to ensure correctness. The system returns a probability chance but does not offer the user suggestions, such as checking, betting, calling, raising, folding, etc. as was originally planned because we no longer felt the need for this functionality. We were also unable to connect the UI with the backend java code implementing the calculations.

Final Class Diagram and Comparison Statement

Project 5:


COMMAND
RequestLoader
Player
name: string
chips: int
bet: int
hand: PlayerCards
isInPly: bool
getPlayer()
setPlayer()
callCommand()
betCommand()
checkCommand()
foldCommand()
executeCommand()
<<interface>>
Command
execute()
FoldCommand
execute()
PlayerAction
call()
bet()
check()
fold()
CallCommand
execute()
BetCommand
execute()
CheckCommand
execute()
OBSERVER
<<interface>>
Odds
registerObserver(): void
removeObserver(): void
notifyObservers(): void
<<interface>>
Player
update()
<<abstract>>
Hand Probabilities
royalFlush: int
straightFlush: int
fourKind: int
fullHouse: int
flush: int
straight: int
threeKind: int
setOdds(): OddsOfWinning
hasA
OddsOfWinning
state
setOdds(): void
getOdds(): void
Subject
«Concrete»
Player
<<abstract>>
Card
house: string
rank: string
addCard(): void
deleteCard(): void
<<abstract>>
Player Cards
card[2]: singleCard
compareToHand(): int
<<abstract>>
Deck
deck[52]: SingleCard
setData(): void
getData(): void
<<abstract>>
Table Cards
cardsDealt[5]: SingleCard
+ method(type): type
SINGLETON
SingleCard
static instance: SingleCard
static getInstance(): SingleCard
private SingleCard()
public setHouse()
public getHouse()
public getRank()
public setRank()
has-A

Project 7:



The observer and singleton patterns remained the same throughout the course of development of this project. These patterns were essential from the beginning and the singleton pattern in

specific was crucial to the structure of the application. The facade pattern was implemented between projects 6 and 7 due to increased code stress. Because we need to perform so many calculations to generate accurate winning percentages, we needed an interface to make these calculations faster and more efficient. As previously mentioned, the command pattern was removed. As we developed this application, we no longer saw the need for the functionality which the command pattern supported.

Third-Party code vs. Original code Statement

– CSS and card templates for Cards.css and TexasHoldemPokerUI.html are adapted from open source code for displaying a 52 standard deck of cards. Source: https://github.com/selfthinker/CSS-Playing-Cards. All other contents is original.

– Not code but data : https://caniwin.com/poker/texas-holdem/pre-flop/10-player-odds/

Statement on the OOAD process for your overall Semester Project

- Structuring and implementing the design patterns in the backend of the code was challenging and the patterns had to be changed and redesigned several times from what was outlined in the original UML design. As the code became more developed it became clear that certain patterns were unnecessary and that others would have to be reimplemented.
- Originally, the frontend HTML user interface was supposed to be connected with the backend java code calculating the hand probabilities. The winning probability was supposed to be displayed to the user on the HTML interface after they input their cards. After working with java servlets and JSP, we were unable to successfully implement this. Working with new software tools that were unfamiliar to us such as javascript and JSP was especially challenging throughout the process.
- Designing and figuring out the user interface turned out to be an interesting process, and several versions were implemented and redesigned once it was clear what functionalities were needed and how they could be best displayed.
- Actually calculating the winning percentage was initially thought to not be too difficult of a task. Realizing how many different calculations and variables are in play with a 52 card deck, and all the specifications of the rules of Texas Hold ‘Em, creates countless amounts of roadblocks and edge cases. This portion of the code took forever to complete and was reworked a couple of times to implement better design patterns to allow for easier calculations.

Link to GitHub repository:
https://github.com/christianjaramillo/OOAD/tree/main/TexasHoldEmAssistant