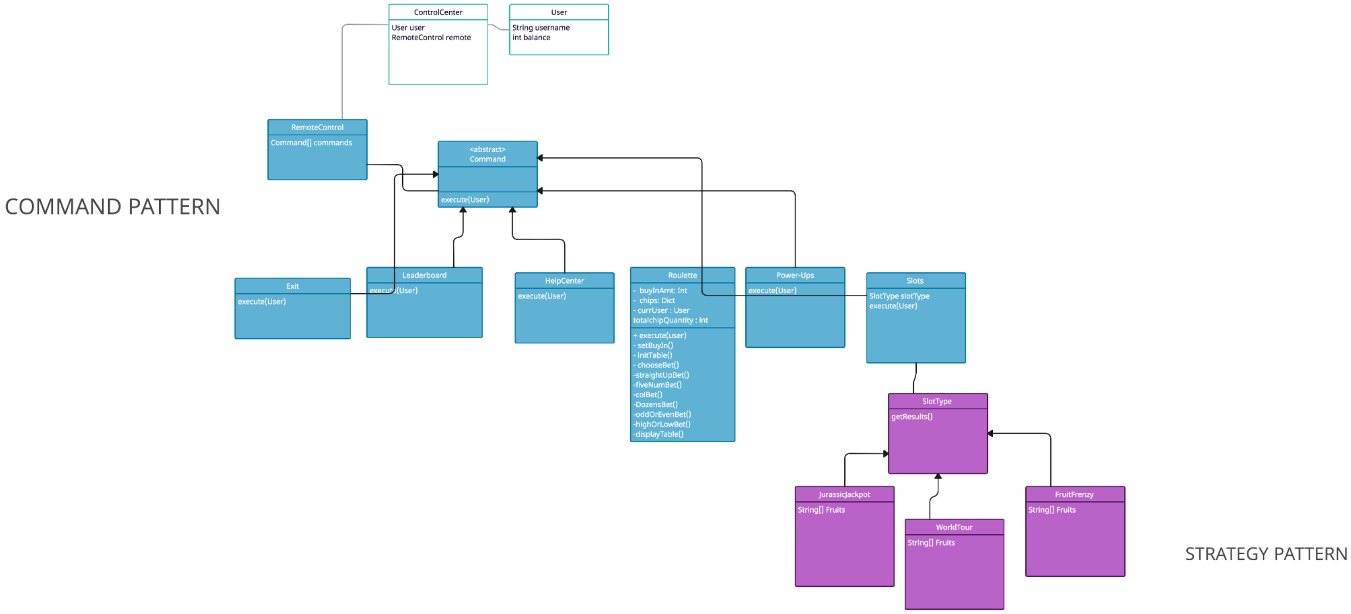
Status Summary:

- Team members: Zion Washington and Sammie Shellman
- Project Name: Lucky Bet
- Github repository
- Work done:
  - Sammie: I created a vertical slice for all the main functionality for the program besides account creation/sign-in. This included making all the relevant classes and associations/inheritances so that the program can take in user input and direct it to the proper Command using a remote control (based on the command pattern). I then went deeper into functionality for the slot machines, so all 3 types of slot machines now work fully - taking in user info, allowing the user to place a bet, running random slots based on the specific slot game, and giving proper rewards (if they win). The help center is also fully fleshed out - users can check their balance or get more information on how to play slots or roulette.
  - Zion: I created the Roulette class which will have five different roulette bets - Straight up, five number bet, column bet, dozens bet, odd or even bet, and high or low bet. I am working on creating a display for the roulette table itself, which will ideally update with an ‘x’ in place of whatever slot the user chooses to put its chip on. I currently have implemented a menu for the player to choose what kind of chips they would like, how much money they would like to bet, and what kind of bet they would like to play. I also added functions in the user class to update balances and keep track of how much money is won by the user.
- Changes/issues:
  - We realized that the program lends itself really nicely to the Command pattern, so we shifted the class organization a lot to reflect that.
  - We haven’t yet implemented the data storage, but we’re considering storing user data simply in memory, through User objects, rather than a full SQL database.
  - We decided it’s a bit confusing/unnecessary to access the leaderboard during the middle of a game, so we changed it so that the user can only access the leaderboard from the main control center.
  - We are currently working on how to best display a roulette table that updates properly as the user adds chips.
- Patterns:
  - As described above, we are using the Command pattern as the outermost pattern, which has proven useful. The program takes in user input and uses that to execute the relevant command on the remote control.
  - We have also used the strategy pattern for the differing slot machines. This is really efficient because the basics of the slot machines are the same, and the pattern allows us to easily manipulate the things that do vary.

Class Diagram:



Plan for Next Iteration:

- Most of what is left to be implemented revolves around the data storage, so that will be the main priority. After that is functional, we will finish the sign-in/sign-up processes as well as the leaderboard and power-ups.
- If that is all complete with time to spare, we can work on making the visuals more appealing and/or implementing more complex randomizers for the gambling games.
- The implementation of roulette needs to be completed with a display and all bets finished accordingly.
- Additional patterns, observer and singleton (for the leaderboard) will also be implemented.