CSE 470
Spring 2023
Final Project
Date: 4/04/2023
Name: Sishir Phuyal

Minimum Specifications:
The project must exhibit an understanding of and utilize a minimum of two concepts we covered in the course:
1. Using lists or tables
2. Navigation between several screens
3. Using maps or location services
4. Implementing persistence using databases or preferences

For the final project, I decided to make a Betting application on Andriod Studios that allows users to bet on UFC fights and see the potential payouts.

To start the project, I designed the Database I was going to use for my project. After thinking about the DB design, I created a DB with two tables. The tables I made are 'activeFights' which holds information for all the active fights, and the other table is named 'bets', which keeps track of all the user's bets. All of the code to create the database is in the DBHelper class. After making the DBHelper class, I created the DBQueries class, which serves to contain all the queries that I will use to get info to and from the DB. The first query I added was "select * from activeFights" which is the query that gets all the fightCards to display on the main page.

After all the Database code is sorted, I had to make the main page that held the recycler view to show all the fight cards that the user can bet on. To make this page, I created an ActiveFightDisplay java class that will serve to display the page. Then I made an XML file called "fight_cards_list" and designed the page so it contains a recycler view, which will display the fight cards, and I added buttons on the top, so I can use those buttons for future navigation to other pages. I labeled these buttons "View My Bets" and "Add fights (admin only). The buttons will take the user to other activity pages that I will discuss later on. But before I can run my code and look at the main page, I need to add a view adapter that will display each item in the recycler view. So I created a class called "FightCardViewAdapter" and I created "FightCardObject" which acted as the DTO, so I can display the list of FightCard on the recycler view. And again, I need to create another XML, a fight card object, that will be displayed in the recycler view. So I made a "fight_card_list_item" XML page, which is basically the container that displays each fight.

The FightCardViewAdapter grabs all the components from the “fight_card_list_item” XML and renders it to the recycler view with the correct information based on the fight card.

After the FightCardViewAdapter is set up. We can add the recycler view and set up the layout manager and viewAdapter on the onResume() method for ActiveFightDisplay. So on the onResume method, we get all the active fight cards by using the DBQueries class and set up the view adapter by passing the list of FightCardObject to the view adapter and then connecting the recycler view to the viewAdpater by using setAdapter(). We also need to add the setOnItemClickListener to the view adapter, so that when a card object is clicked on, it will take us to the main betting page for that fight card. So whenever a card object is clicked, on the recycler view, the onClick method triggers and starts the Intent, which will take the user to the specific betting page, which hasn't been created yet, I will create that after the admin page is done.

So, now the ActiveFightsDisplay has a working recycler view with a proper adapter, but we have no information to show. To allow myself to add information to the DB, I created another java file called AdminPageAddFightActivity. The purpose of this page/activity is for the admin to add fights. So in the main ActiveFightsDisplay file, I add a boolean called adminAccess, which if it is set to true, it will allow the user to click on a button called “Add fights admin only”. If it's set to false, then the visibility for that button gets set to “Gone” which prevents the user from accessing the admin page. In order to get to the AdminPage, we need to add an event listener to the button, which will trigger an Intent and take the user to the admin page.

To display the AdminPage, I created a java class called “AdminPageAddFightActivity” and an XML named “add_fight_to_db” which contains 5 editText fields: fighter one name and odds, fighter two name and odds, and the start time. And a submit button that the user can click on. And a button on the top that allows users to go back to the main display. Then on the AdminPageAddFightActivity class, we add an event listener to the submit button and the go back button. If submit is clicked, the listener is triggered, and using the DBQueries class, we insert the newly created fight card object into the DB. And if the back button is clicked, an Intent is started that takes the user to the main page. Now the

AdminPageAddFightActivity is done. And whatever fight card information is inserted into the DB, should now show up on the recycler view on the ActiveFightsDisplay page.

Now we have a way to add fights to the main page, kind of simulating an admin role. Whatever we submit to the activeFights table in the DB, will show up on the recycler view. Previously I mentioned adding onClickListeners to the individual items on the recycler view on the ActiveFightDisplay Activity. Whenever a fight card item is clicked on, the onClick gets triggered, and a new intent is activated, this intent will take us to BetPageActivity, the class that displays the betting page for the individual fight cards.

Before I make the BetPageActivity java class, I need to make an XML for the class. So I create a new XML page called “main_betting_page” which has all the fight information needed. There are fighters' names and odds, there is also a “Enter Bet Amount” and “preview winning” button that shows the possible winnings. I also added two SwitchCompat, that allows user to select the fighter they are about to bet on. And there is a submit button at the end of the page, for the user to submit their bets. After the XML page is ready, we make the BetPageActivity class. Also to mention, whenever an item from the ActiveFightsDisplay recycler view is clicked on, the fightID is passed to the BetPageActitvty, and using the fightID, the BetPageActivity fills up all of its information after querying the fightCardObject from DB. Most importantly, we need to add an event listener to the submit button, so that whenever a user submits their bet, we can insert the bet information to the bets DB table, and make the instance in the activeFights table inactive, so it doesn't show up on the active list anymore. Now the BetPageActivity is complete.

We now have a recycler view that shows all the fight card information, we have a way we can insert fight info into the DB, and we have a way where we can make bets on the fights. Now we need a way to view our bets. So to create our “View My Bets” page, we need to make a similar XML to the main page that has the recycler view. And a similar xml recycler view item, that will hold the bet information for individual bets. The XML for view my bets is called “view_my_bets_list.xml”, and it includes a recycler view and two buttons on the top(one that allows user to go back to main page and one to simulate

the payout results). And in order to display the list in the recycler view, we need to make a dto for the bets. So we make a java class “MyBetDTO”, that includes the fight card object, along with the bet amount and the user’s fighter pick. This information gets pulled from the bets table in our DB, and since the bets table has the fightID foreign key, we can also grab the matching fight card. So, when the page is rendered, the recycler view gets all the bets from db, and using the a list of MyBetsDTO, it fills up the list with all the bet information. Now we have a “View My Bets” Page that shows all of our bet information.

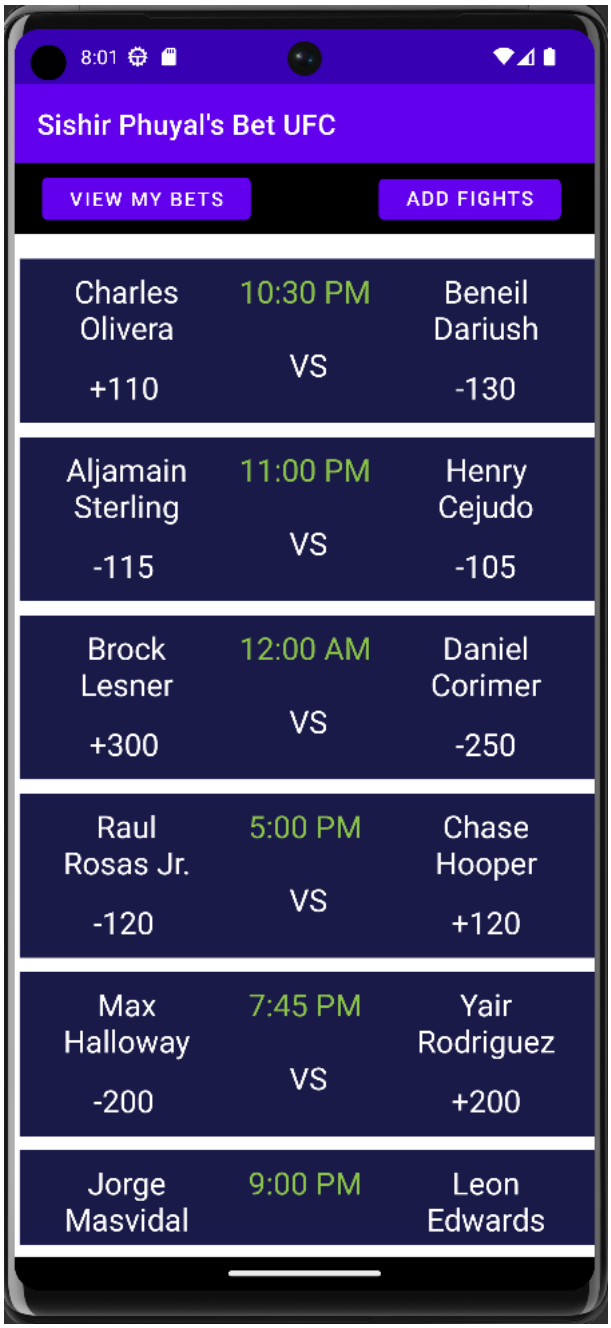
Now we need a way to simulate the bet payouts (the amount of money we have won or lost). In order to do this, I added a new text variable to the list item xml for the my_bets list, which is originally hidden, and shows if the user has won or lost the bet. I also added two new hidden text variables on the bottom of the “view_my_bets_list” xml page, so I can display the results of the payout after the “Simulate Payout” button has been pressed.

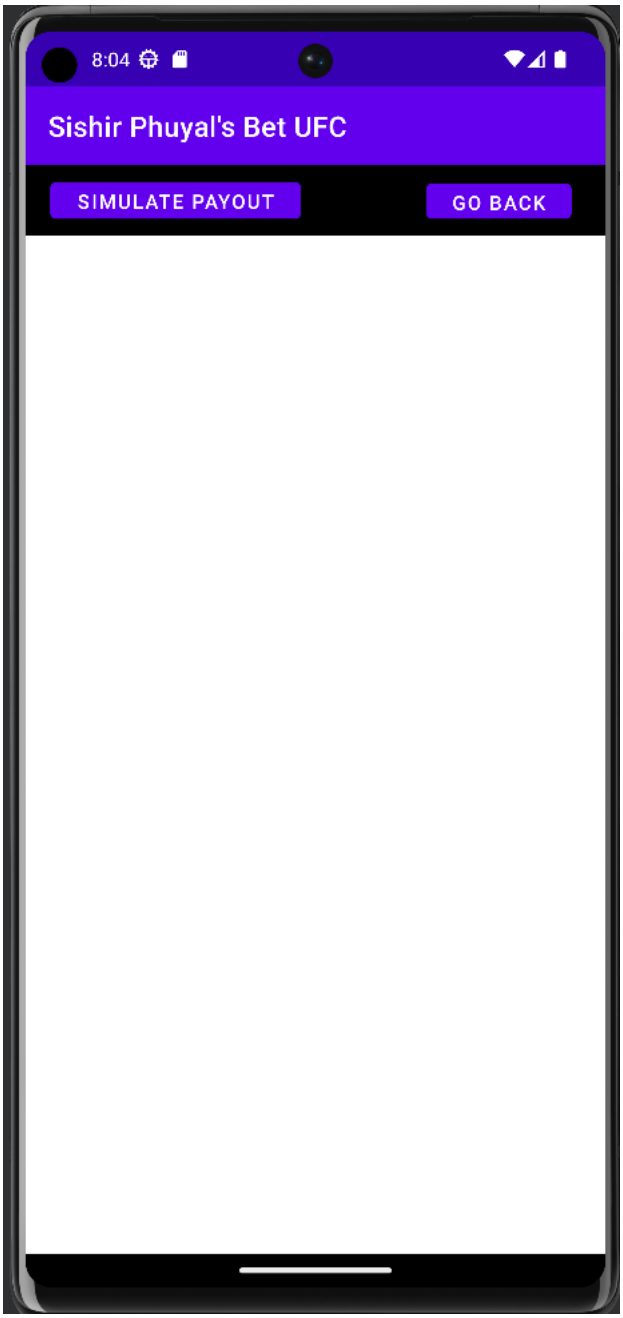
When a user clicks on simulate payout, a new intent gets called to the same page, but now it passes a variable called “simulatePayout” to the new ViewMyBetsActivity class. This variable triggers a method called simulateFights(), which simulates all the fights in the current bets DB, it randomly decides if a user has won or lost, by generating random num from 1 to 2. The variable simulatePayout also sets the boolean showPayOut to true, so now when the recycler view is getting rendered, we pass in our new modified bets list which has the results of the simulation. After the new bets list is made in the simulateFights method, we need to clear the bets table in the DB and the fightCards with the corresponding fightIDS in the activeFights table, so the bets are removed from the page.

Now the application is complete. From adding fight cards to betting on the cards, to simulating payout, the application has many features that allow it to work well and function as a betting application.

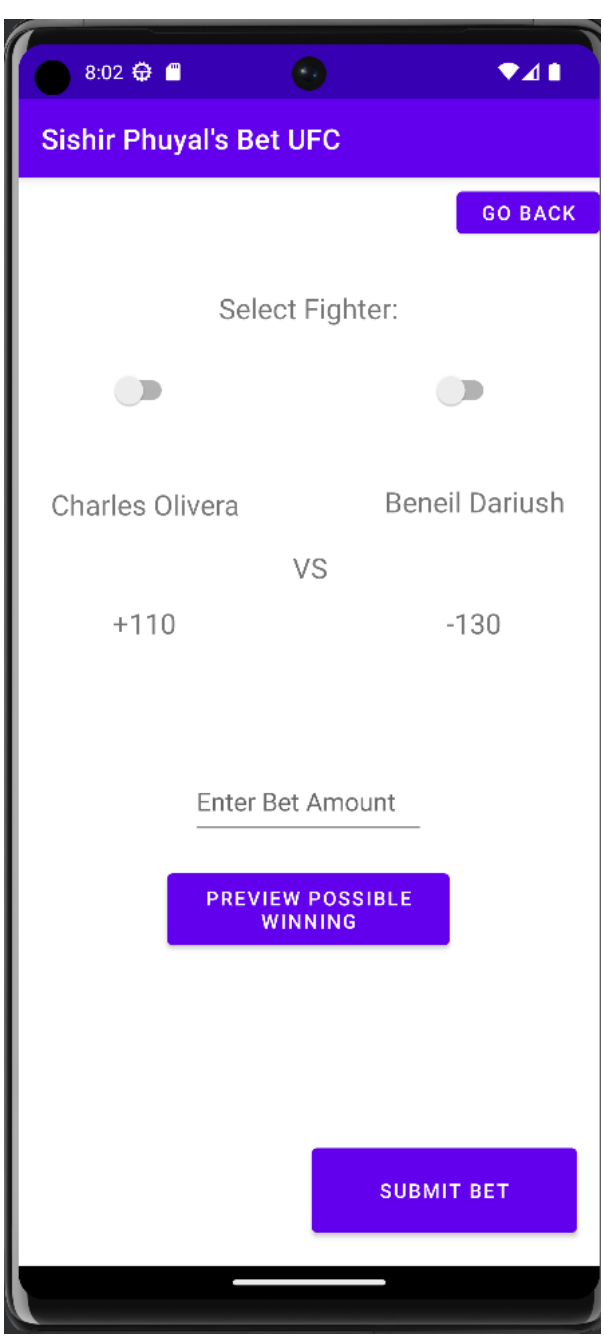
Photos showing app functionalities:

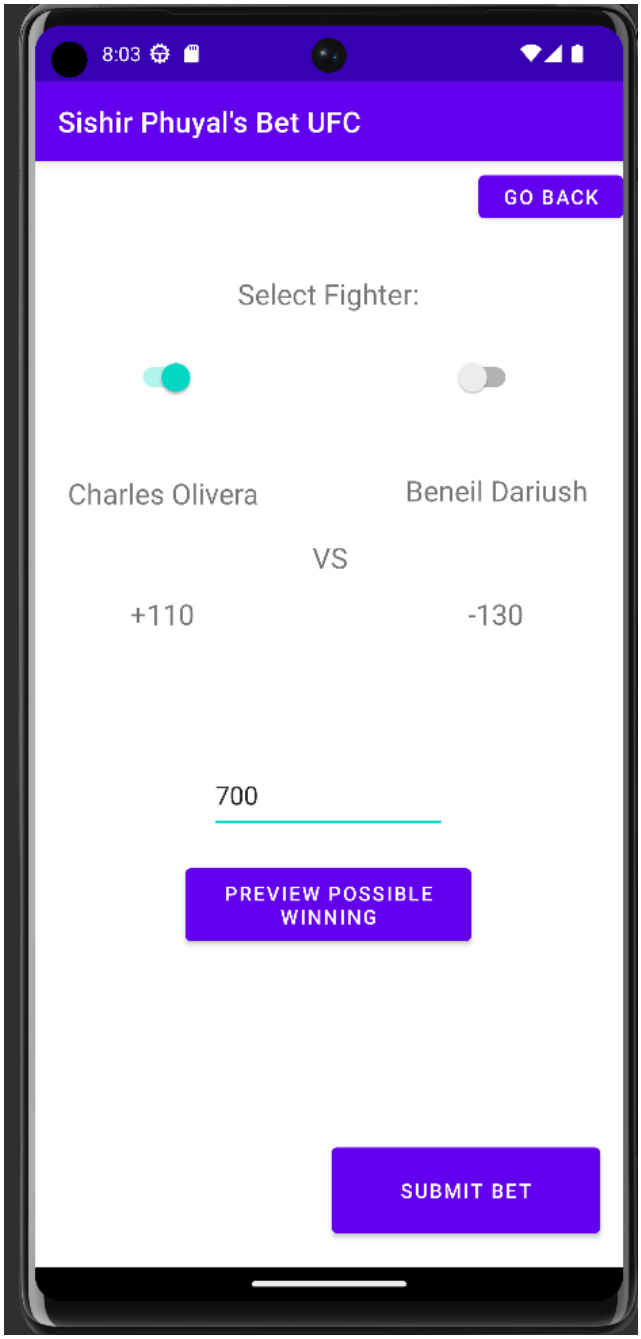
When a user opens the app it takes them to the fight cards page, which displays all the available cards to bet on. I've added a few cards already to test the recycler view. The 2nd picture is the view bets page that pops up after the user presses the view bets button, currently, there are no bets, so nothing shows up on the page.



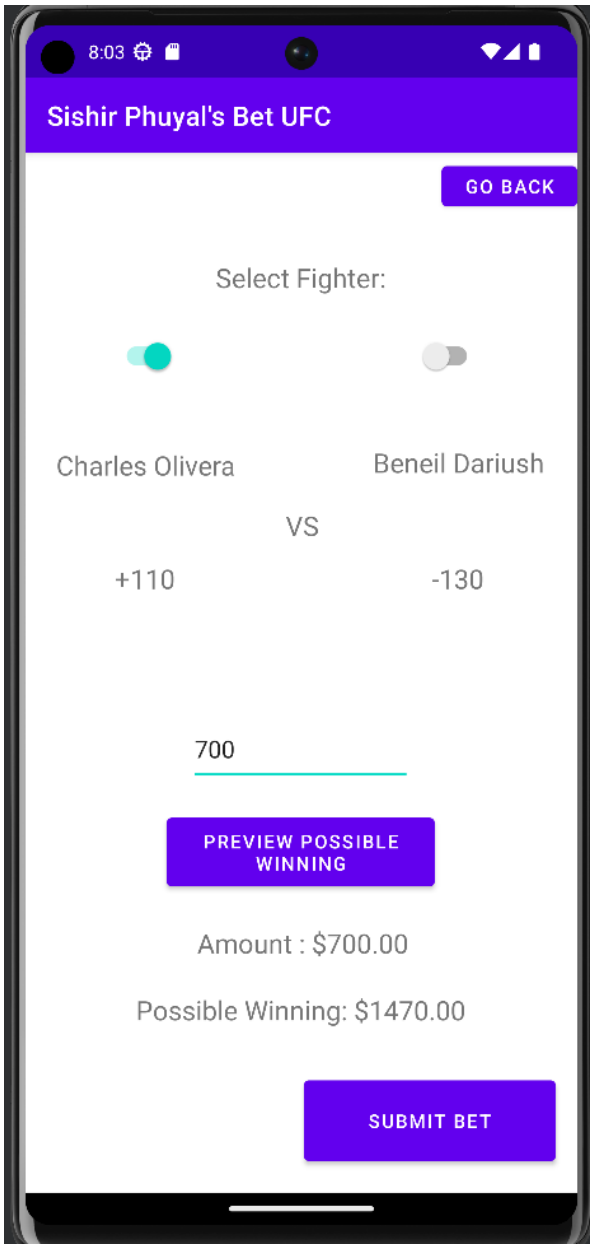


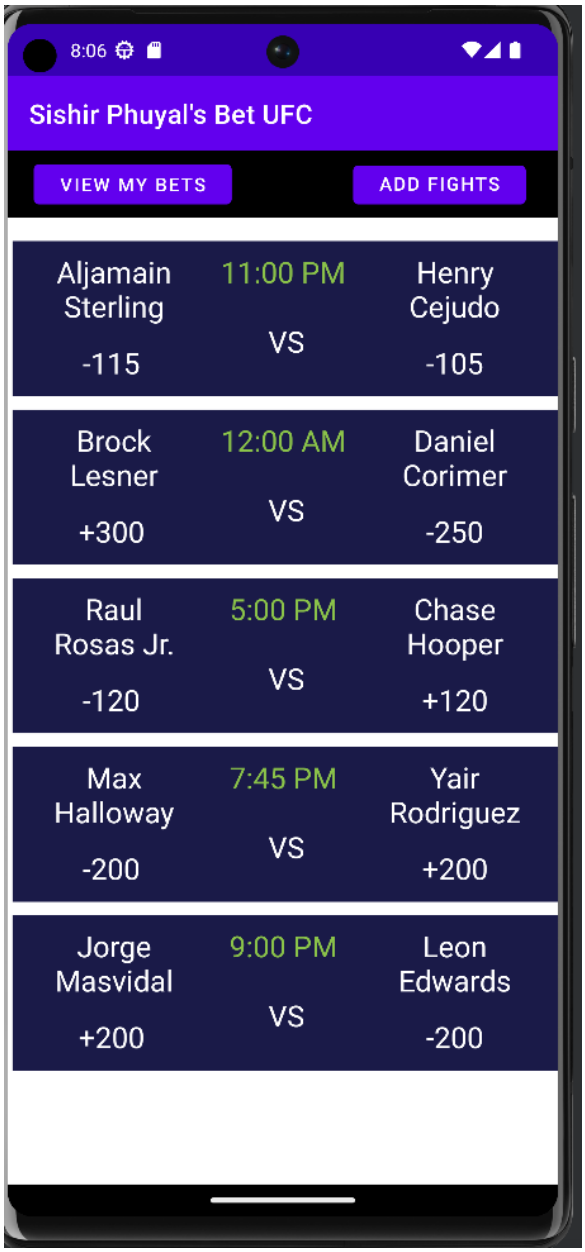
Now, let's bet on a fight. To bet on the fight, we press on the fight card objects in the recycler view and it takes us to the betting page corresponding to that fight card ID. For this example, we click on the "Charles Olivera vs Beneil Dariush Card", this takes us to the betting page (shown in the first picture). We then select Olivera as the fighter we are betting on and enter the amount of 700 for our bet amount.



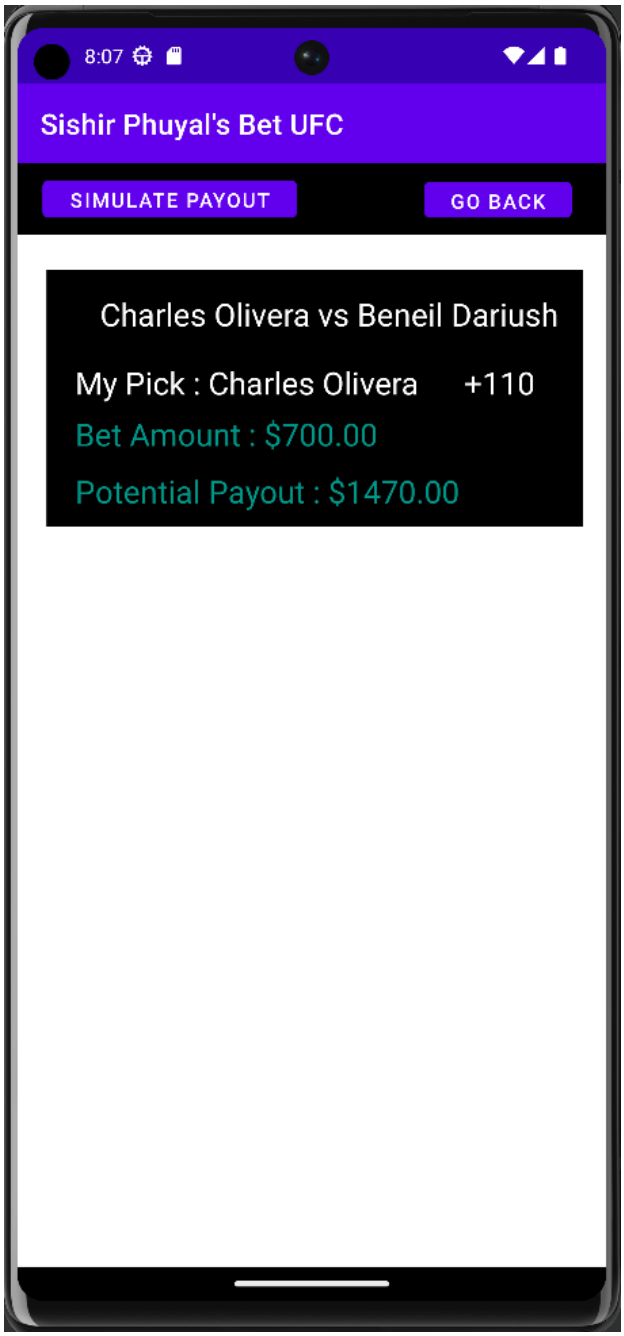


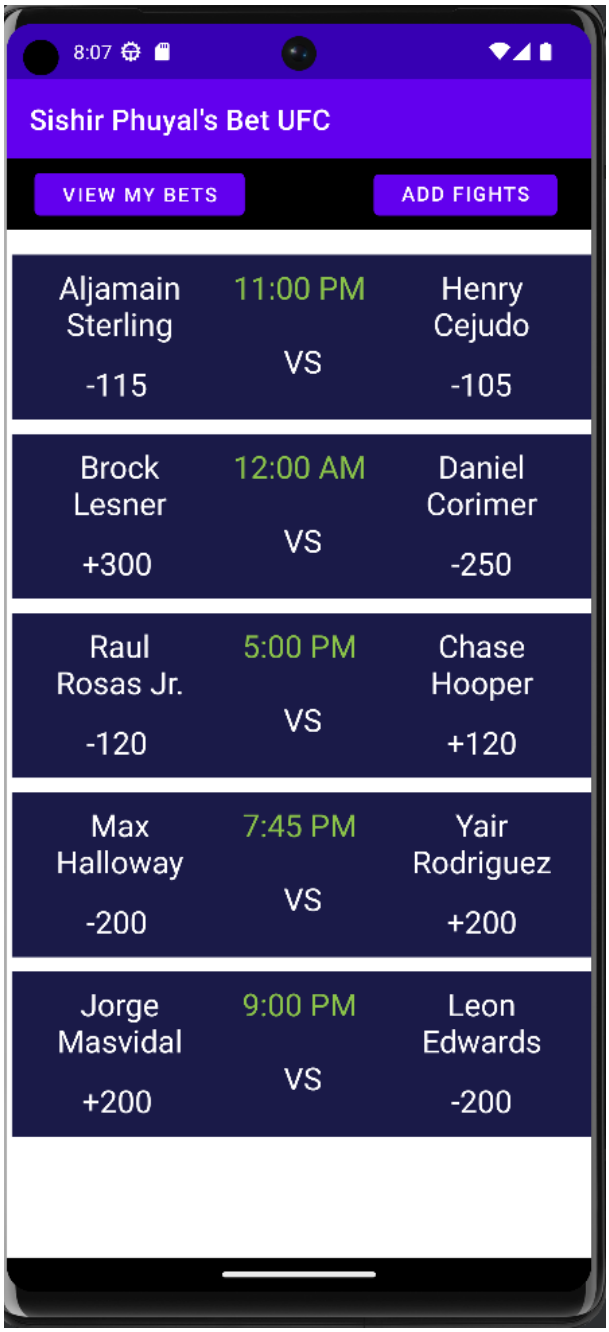
(First Picture): When we click on the preview possible winning, this will show the possible winnings of our bet based on the fighter and odds we have selected and the bet amount we have entered. (2nd picture): After we look at the amount we can possibly win, we press on the submit bet button, which adds the bet to the DB and also inactivates the fight card (preventing it from showing up on the recycler view again). Then the app takes us back to the main fight cards page, as we can see, the Olivera vs Dariush Card is no longer there.



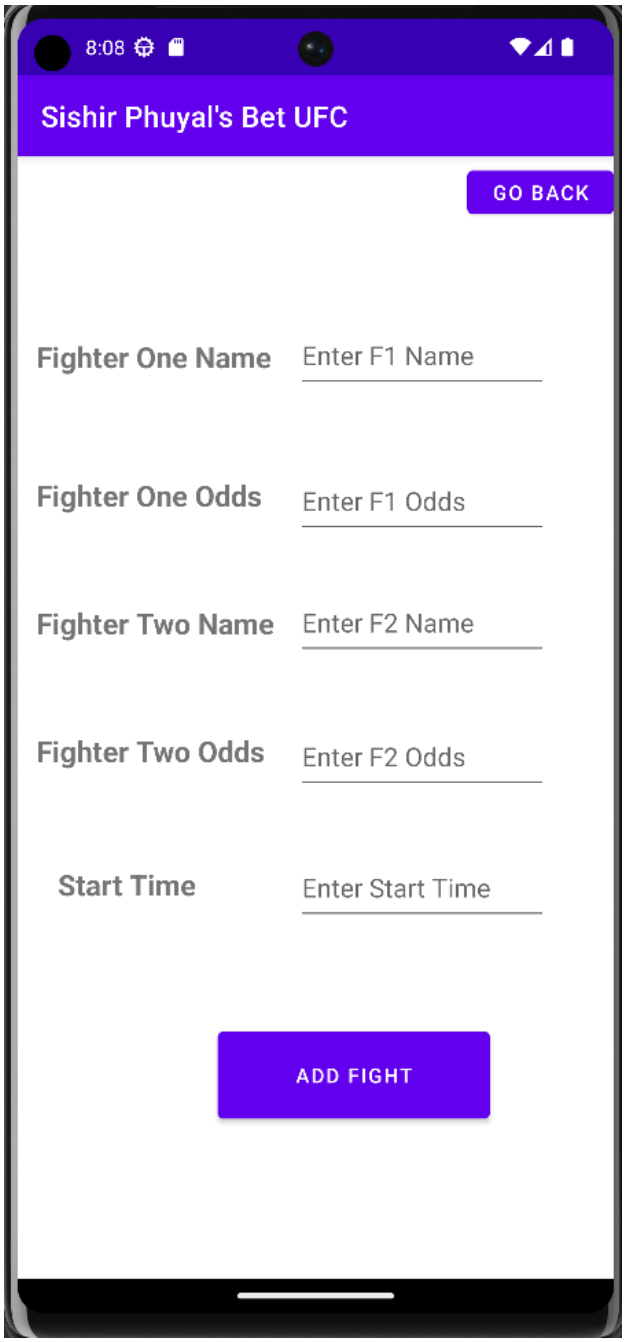


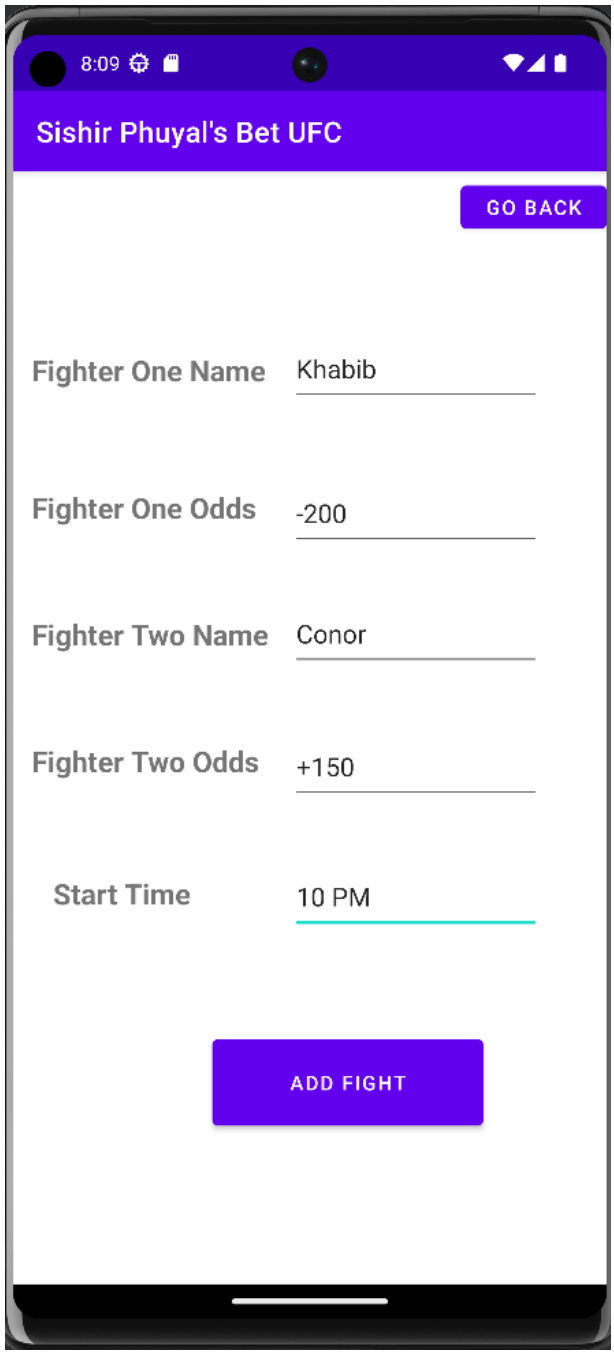
(First Picture): From the main fight cards page, we click on the “View my Bets button”, which takes us to the View bets page. Before, the view Bets pages were empty, now we can see the bet we put in for Charles Olivera. (2nd picture): we go back to the main page to show the functionality of adding a fight card.



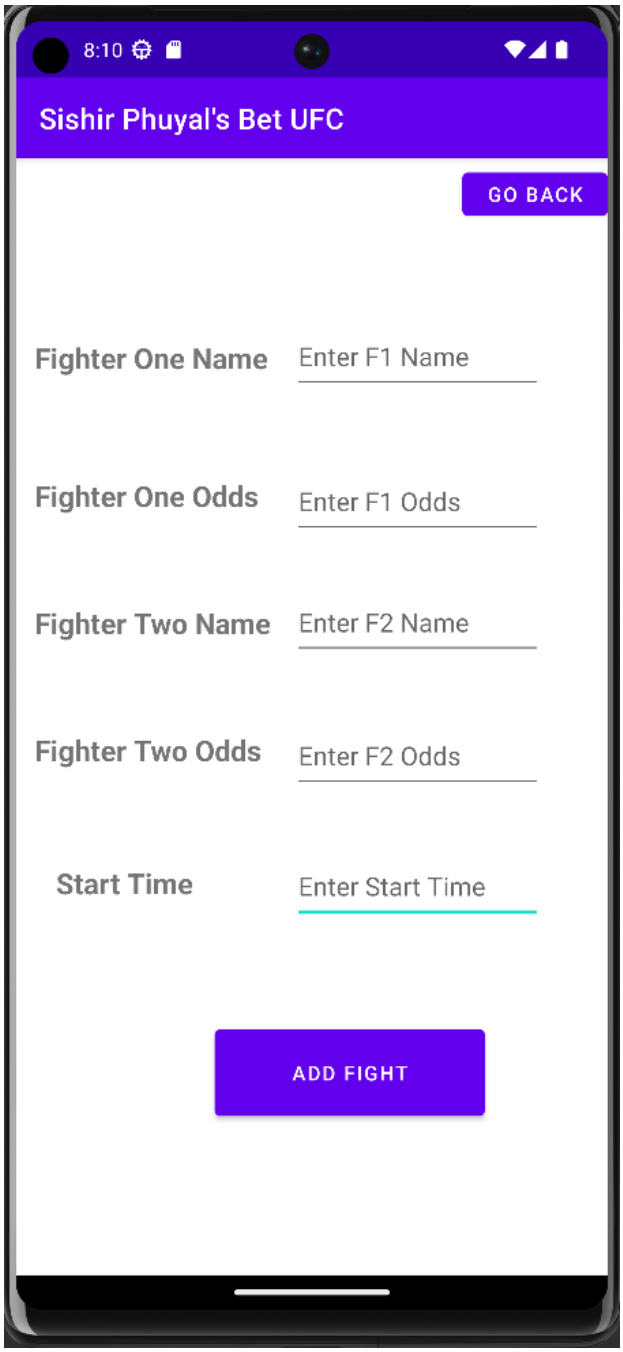


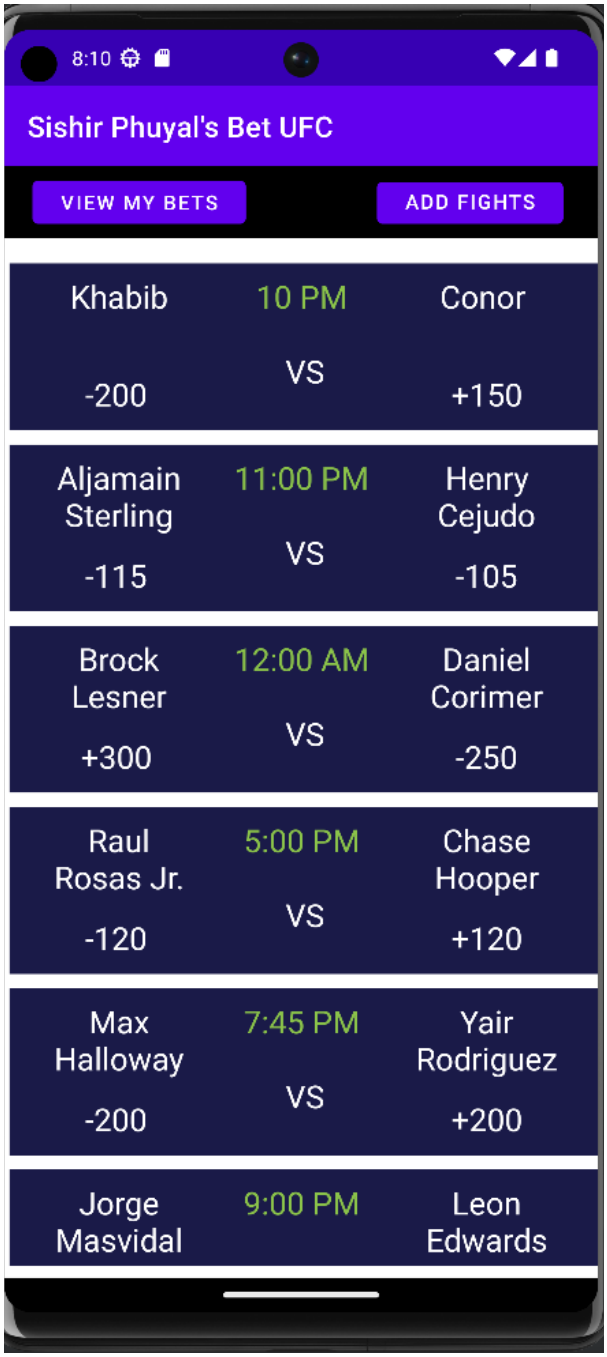
(First Picture): Clicking on add fights. This takes us to the add fights page, which only the admin has access to. (2nd picture): Adding in fight card information for Khabib vs Conor.



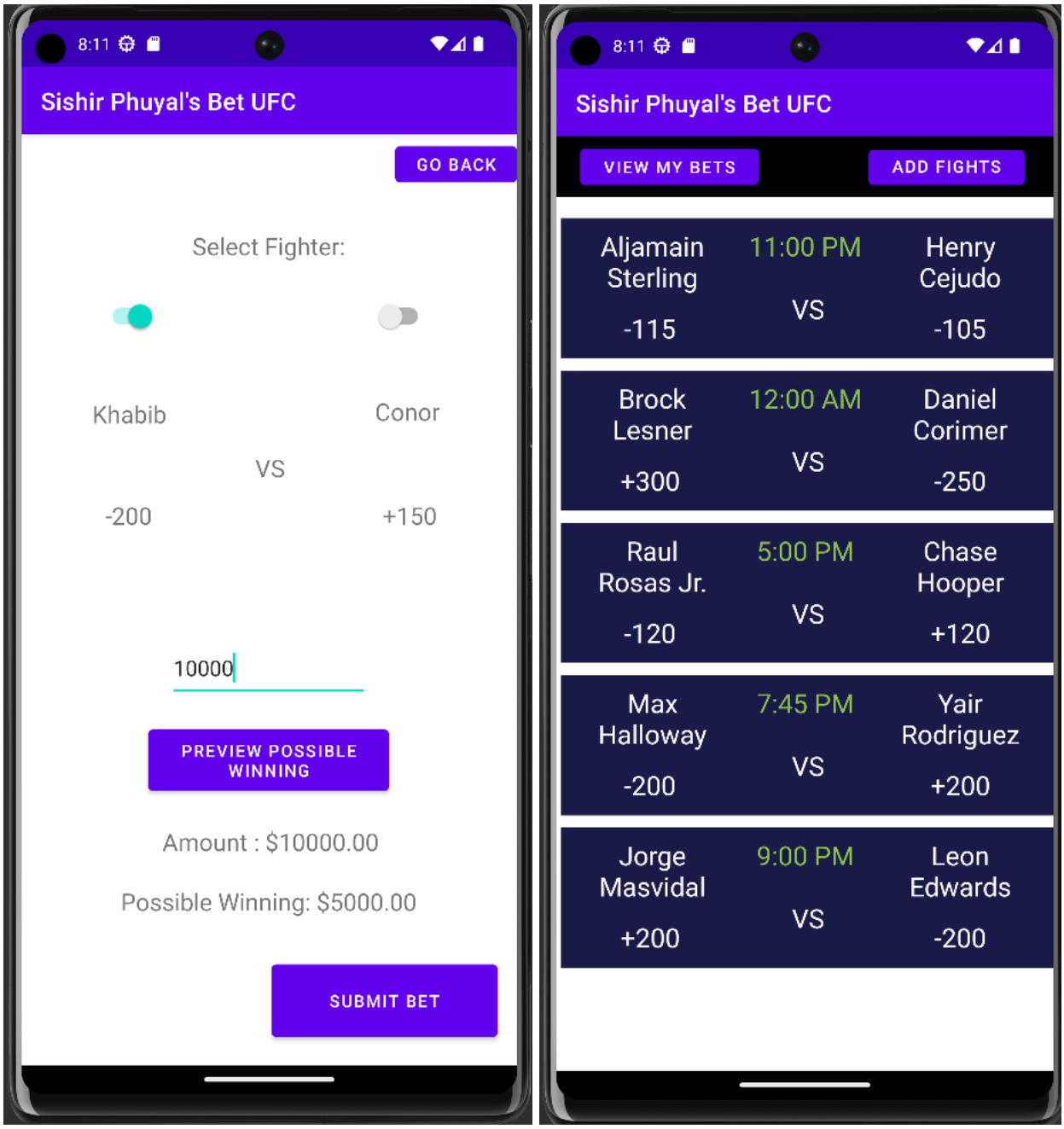


(First Picture): After we press the “Add Fight” button, it adds the fight card to the DB, and clears all the fields, so the admin can add more fights. (2nd picture): a new fight has appeared on our recycler view list, the fight we just add: Khabib vs Conor. Showing that the add fight card functionality works.



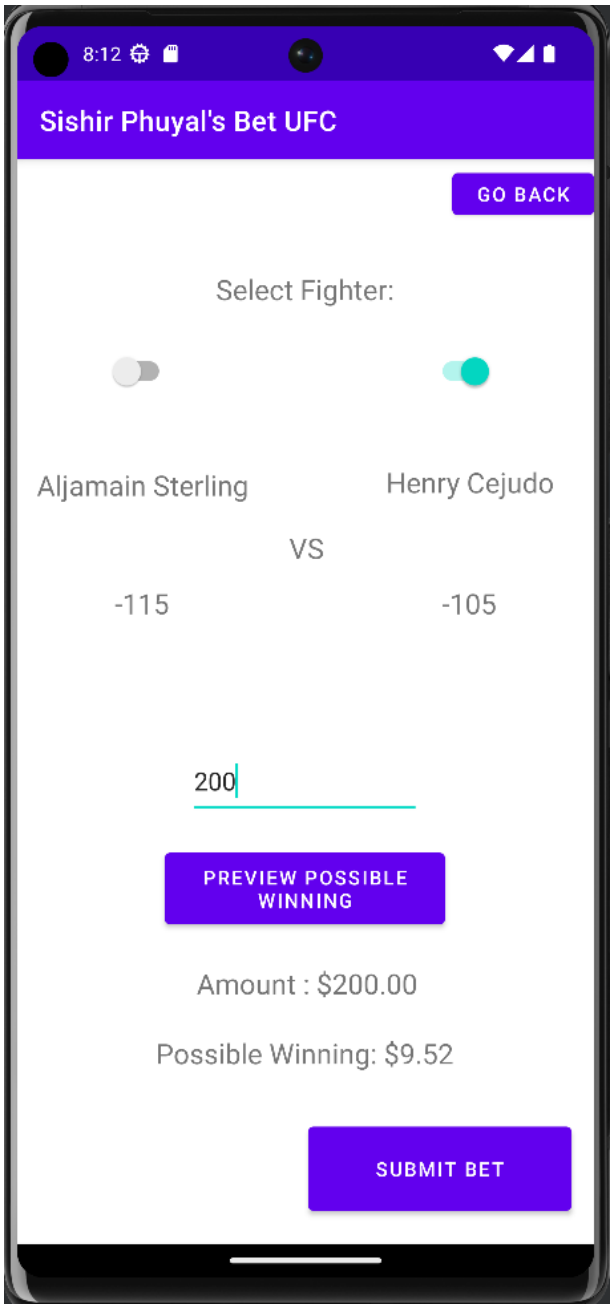


Now, let's bet on a couple of fight cards, so we can simulate a payout. A “Simulate payout” can be done with just one bet, but having more bets makes things more interesting. (First Picture): betting on Khabib, entering in bet amount, and previewing the possible winning. (2nd picture): After pressing “Submit Bet”, we can see that the “Khabib vs Conor” fight card is now deactivated and no longer there.



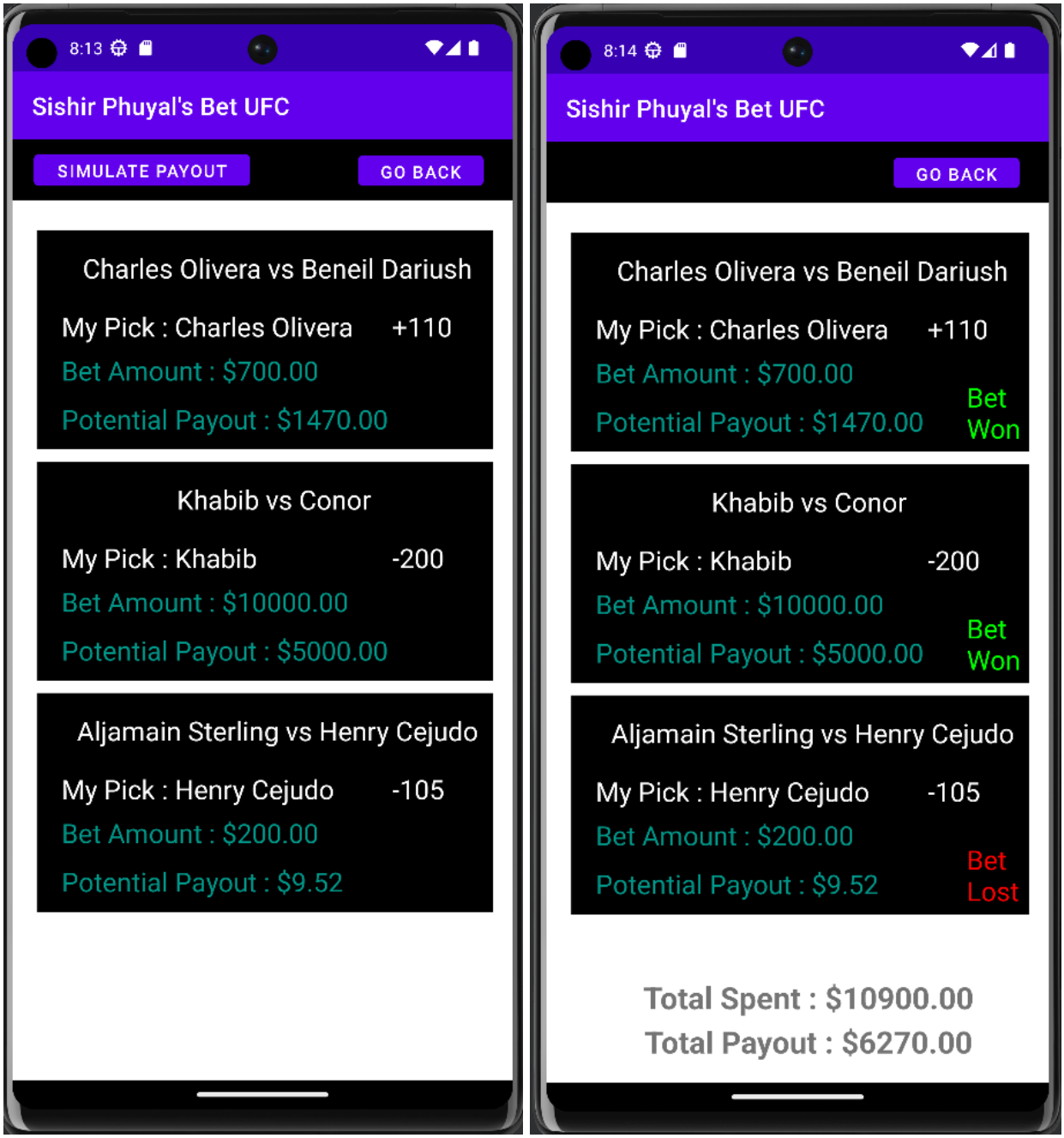
(First Picture): betting on Sterling vs Cejudo, entering in the bet amount, and previewing the possible winning.

(2nd picture): After pressing “Submit Bet”, we can see that the “Sterling vs Cejudo” fight card is now deactivated and no longer there.

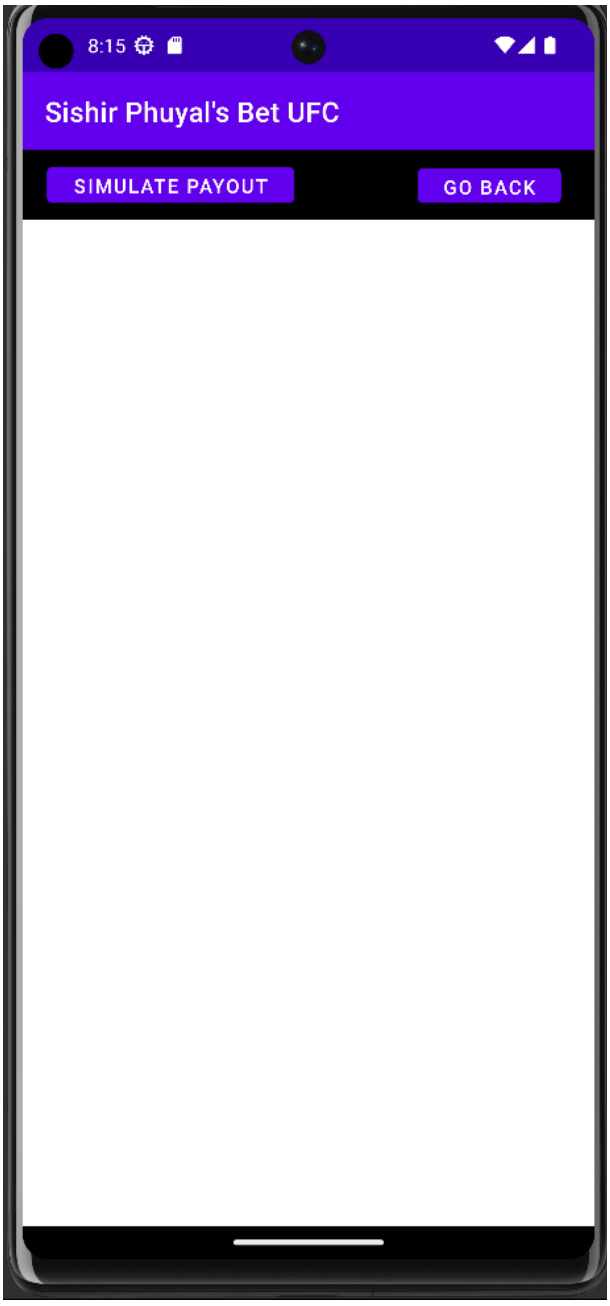


Let's check out the “View Bets” page to check if our bets were successfully added.

(First Picture): Clicking on view bets, and going to the page. As you can see the two fights we just bet on are now added to the list. (2nd picture): Clicking on the “Simulate Payout” button. As you can see, all the bets have now been simulated and the results of the bets are posted on the bet information card. And the total spent and the payout are listed at the bottom of the page.

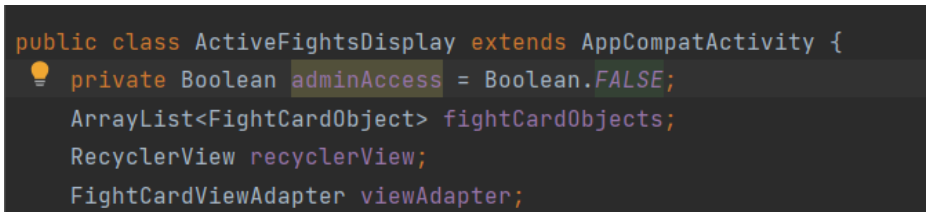


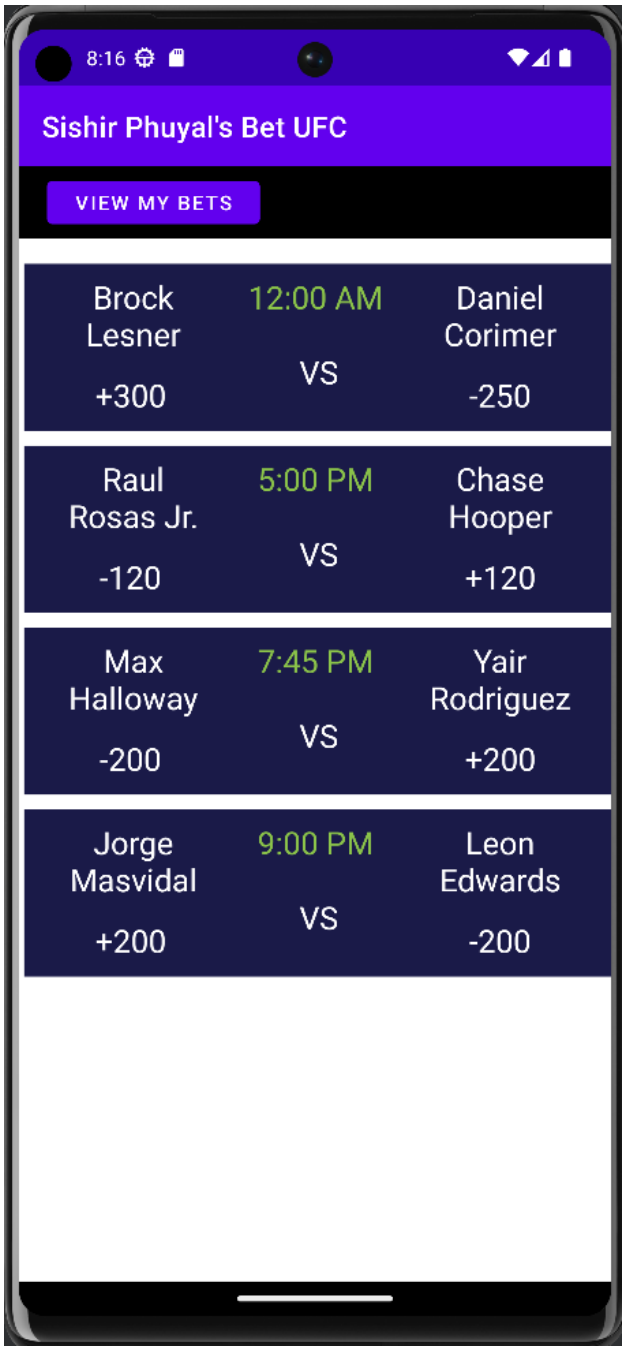
After the bets have been simulated, the bets are removed from the DB, and the corresponding fight card is also removed, (when we submitted the bet, it was just deactivated), but now the fight card gets deleted too. (First Picture): We press the “Go Back” button to go to the main page and leave the simulation results. (2nd picture): Clicking on “View My Bets” to check if the bets from earlier are still there. As you can see in the picture, all the simulated bets are now removed.



Now we have covered all the functionalities from adding fight cards to betting on the cards to simulating a bet payout. And all the functionalities are working. The last thing we need to look at is the adminAccess variable. This variable is private in the code that determines if the app is being accessed by an admin or not. If the variable is set to true, then the Add Fight button shows up, allowing the admin to add fights, If the variable is set to false, the button disappears, which will apply to most users because they shouldn't have the ability to add fights, only the admin should.

Setting adminAccess to FALSE and rerunning the application:





As you can see in the picture above, the Add fights button is no longer there. Preventing users from adding fight Cards.