* Describe in detail all new and changed functionality your team intends to implement for project deliverable 3.
  1. **NEW:** Implementing combat checks for all skills
     1. I will add a functionality that introduces a luck-based system for all four skills that a fighter has. When battling, this means that an attack isn’t guaranteed to hit the opponent, a special attack isn’t guaranteed to hit, and vice versa. The game will now feature an RNG (Random Number Game) check to determine if the action chosen will be carried out or not. For example, Player 1 attacks Player 2, and Player 1’s RNG check comes out as ‘5’, but that isn’t equal to or greater than Player 2’s Speed stat, so the attack misses; end turn.
  2. **NEW AND CHANGED:** The “DEFEND” and “DOGDE” skills will be changed so that they give their respective Fighter stat an increase (“buff”) by an arbitrary number. Implement the “SPECIAL” attack functionality.
     1. Due to the way that the current attack system is implemented and the fact that I will be introducing a luck-based system to battling (read above), I believe increasing the stats for Defense and Speed (“DEFEND” and “DODGE”) is a good middle-ground approach.
     2. The “SPECIAL” attack functionality works almost like a regular Attack. It deals twice\* (\*subject to change) the damage, but has a much higher chance at missing the opponent.
  3. **NEW:** Implement a “isDead” state, which initiates the “FIGHT AGAIN” pop-up window to appear at the end of a battle.
     1. The “isDead” functionality will consider the Fighter’s hit points, so when a Fighter’s health is <= 0, the game announces the winner (the Fighter whose health is > 0).
     2. After a battle has determined who has won/lost, a pop-up window will appear asking the Players/Users if they would like to play again, which will send them back to the ‘Character Select’ screen and choose their Fighters again.
  4. **NEW:** Make a Map data set that will link how many Wins a Player has.
  5. **NEW:** Add sound effects when selecting a fighter, battling, and winning.
  6. **NEW AND CHANGED:** Adding more Fighters to the ‘Character Select’ screen.
* Describe how you will implement this functionality. This can be done through design documents (such as those submitted in preparation for project deliverable 1), or through detailed descriptions of changes to or new classes, methods, and data. Be especially careful to document information flow between MVC components and all changes required in all of these components.
  1. This system will be implemented in the ‘Battle’ class (I plan on moving the ‘Battle’ class into my Model class as an inner class). The system will be instantiated by the button clicks in the View, send the request to the Controller and to the Model/Battle class. If the combat check is successful, it will initiate the chosen action (Attack, Speed buff, Defense buff, and Special Attack), which is also now in the Model.
  2. The Speed and Defense stat increases will be affected by the luck-based system, but will have a very consistent success rate. If their respective buttons are clicked on in the View, the request will be handled by the Controller to pass onto the Model/Battle class. The Model will then update the fighter’s Speed or Defense stat by a fixed amount. The Special Attack will be implemented the same way, with the same View to Controller to Model/Battle path. The difference is, it will do significantly more damage but with a small success rate.
  3. “isDead” will be implemented into the Model class, as it will read the fighter’s Hit Points stat to determine if that specific fighter is dead/lost. The status of the fighter will be checked by the Model each turn. This will display the Win/Restart pop-up. A message is sent from the Model to the Controller and then to the View to display it. If the Players/Users choose to fight again, a request from the button click on the “Restart” button will travel from the View to the Controller and display a clean instance of the ‘Character Select’ screen.
  4. Instantiated and handled by the Model, it will check which Player/User is the winner based off of their Fighter status (alive or isDead). The Player/User who won will be placed into a Map data structure where the Key is the Player object and the Value will be an Int.
  5. All sound effects will be handled solely by the View class based on button clicks and the View receiving a status update from the Controller of a Fighter death.
  6. A new Fighter will be made inside of the Model class and the name and stats will be sent to the View by the Controller. The View will display the Fighter’s information.
* Provide descriptions (or drawings of) any new user interface elements.
  + The ‘Play Again’ pop-up display will be a small window that contains text (“Player # is the winner!”, “Would you like to play again?”) and a button that says “RESTART”.
  + New fighters will be displayed the same way as other fighters in the ‘Character Select’ screen.
* Describe how you will test this functionality.
  + When clicking any of the battle buttons, a message will be shown in the output console of the IDE stating “Fighter # successfully/unsuccessfully \*battle action\*”.
  + Clicking the Speed buff or Defense buff buttons will increase the Fighter’s stats, which will show that action in the output console of the IDE. The Special Attack will be shown the same as a regular attack, but the health bar of the opponent will have a more drastic change (more damage) than a regular attack.
  + “isDead” will be tested by the fully red health bar of a Fighter and by the ‘Play Again’ pop-up display appearing on the Battle screen. The “RESTART” button will be tested by the screen going back to the ‘Character Select’ screen with buttons reset.
  + Wins by each Player/User will be displayed on the top of the Battle screen.
  + Sound effects will be played upon certain button clicks on both the ‘Character Select’ screen and Battle screen.
  + More fighters will be selectable in the ‘Character Select’ Screen, and playable in the Battle screen.
* Describe the responsibilities of project team members in implementing this new functionality.
  + I am working by myself, so implementing every functionality from this document will be the sole responsibility, and work, of me.