The system contains several important components. Firstly let's dive into the model and the components that the model would need. There will be some sort of object that we can call a *board* which will contain information regarding how the board is laid out with tiles, which tiles are still available and which ones have to be removed. The board will also be responsible for keeping track of valid moves, among other verification that is needed in the game.

A *Tile* can be an object as well, this will be a single tile used to create the board. This will represent a single tile, which will store information like how many fish are on a single tile. *Penguin* is another object that needs to be accounted for in the system, on the model level of software architecture. The penguin object tracks the penguins current location of the penguin. *Player information* is another object that would need to be in the model level of the architecture as we want to store login information with their password. This would be information that the player provides in order to connect. Lastly, an object named *Tournament* is required. This object's purpose would be to contain information about the different games, standings, scoreboards, and players within the tournaments. This object would simply represent an instance of a tournament being run on the game server.

Next in the three-layered architecture there is the controller component. The controller acts as an intermediary between the model and the view. Part of the controller's responsibility would be to send events from the model to the view and vice versa. The controller is also in charge of deciding which view to present the player. As will be discussed further in the memo, there are various views that need to be implemented and the controller is in charge of figuring out which of them to show the player. Switching between the login view, and the actual game, to showing end of game results. Within the controller needs to be some timer functionality as well that decides when the signup period is over and when the game can start.

Within the View section of the architecture there are several components that need to be addressed. The first view that needs to be considered is the *Signup* view. This view's purpose is simply to welcome the user, to sign them up and collect player information that the model can store. This represents the beginning of the experience for the user.

The second View that we need to implement would be the GUI to the actual Fish game. The purpose of this view is so that the player can actively interact with the game and participate in the tournament. This view represents the actual game itself as far as the player is concerned. This view will be in charge of displaying the board, the fish, and the penguins. All moves being made will be shown in this view until the end of the game. This view will potentially show the player's score, how many penguins are left, overall view of the board, etc. This will act as the main view for the player.

The last view that needs to be discussed is the end of the game. As the game ends, the controller will manipulate this information and show a new View. This view is similar to the signup view, however it'll act as an information display for players to see their overall score, the winner, and inputs for the next tournament.