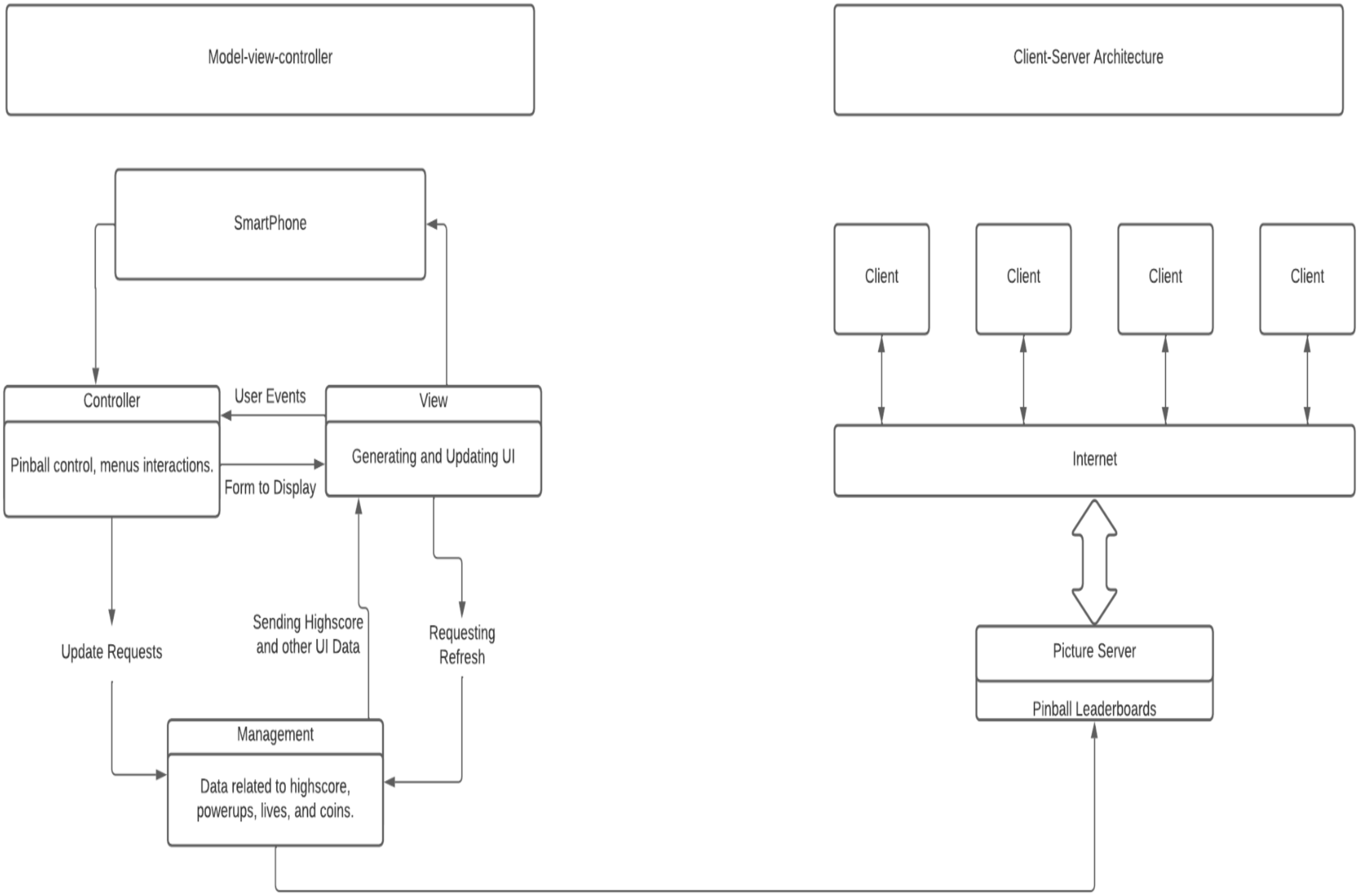
Architecture Design



For the application, we are planning on creating it with an MVC architecture design. This is because we want the controller to handle any interactable scenarios from the player. For example, the controller would handle when the user would control the pinball, select the pause menu, and more. The View would display the data, such as the score the user currently has, where the pinball is, or what power up they currently have. The Model would manage all the incoming data, and then the View would gather the data to display it. This would be calculating the high score, total coins etc. We chose this design because there is a lot of interaction and data being provided from the user, and having the controller manage that while the other two focus on the display and managing the overall data would make it an optimal design for our application.

Our website, which displays the overall high scores from the entire playerbase, we chose Client-Server Architecture. This is because the website needs to be constantly receiving POST requests from the game. The site will be a picture server, as it will display the text for the top high scores, along with a profile picture which was determined by the user within the game. It also has no controller interactions so there is no point in making it an MVC. This will allow all of the users to visit the site, as the site will just be updating the data when a new high score is received from the game.