League of legends is a multiplayer strategy battle game. The game consists of over 140 unique characters, known as champions who each have four completely unique abilities, that impact the game by dealing damage to enemies, healing allies, preventing enemies from preforming actions or augmenting allies. Before the game takes place there is a very important, pick and ban phase, which is the focus of our application. During this time the players ban champions to prevent their opponents, or sometimes their teammates, from using those champions and then pick the champion they want to play. The decision of which champion to play is heavily impacted by what the opponents have already picked.

The reason we chose to create an application based around the pick and ban phase was due to the complexity of figuring out what champion would best go against the current opponents and the limited amount of time a person has to do it in. This app would simplify the decision while still giving the user a pool to choose from.

When designing the app we decided to use the waterfall model. The project was spilt up enough and straightforward enough that the we could parts to people after we had the designed the project. The project was small enough that we would not need frequent meetings to check on everyone’s progress. The first time we met we discussed the requirements for the project and what we wanted to make it do. The second time we created the design for the project and separated out who was going to do what. The third time we checked on the progress of implementation and made sure that everybody was on track. The fourth time we began testing our product and found that we would have to move away from the initial design slightly and the fifth time we generated a working app to demo.

The waterfall model was successful as we were able to rely on each member to complete their part. Each member was caught up on their tasks at the planned times and able to communicate what they had completed to the group. Because of each member’s efficiency the waterfall model was successful. If the code had been more complex we would have had difficulty without extra communication that would have been provided by an agile process, however do to the simplicity of the code the waterfall model worked.