**Design Decisions:**

* Record the key decisions that let you explain why you have chosen to do things in a particular way  
  + We have decided to include an abstract class for Account in order to be able hold the basic properties of every account within the system, but this Account class is extended into several subclasses of each users which are Organizer, Referee and Coach. This allows us to limit access to the user, for example a Referee shouldn’t be able to create a tournament so once you log in as a referee the only option you will have is to add or edit scores.
  + We have followed SRP (Single Responsibility Principle) which has allowed us to delegate the features of the system into their own respective classes – this allows us to avoid confusion and repetition of codes. For example, the Schedule class will only need to think about its own feature like generating a schedule and getting an existing schedule.
  + We have also followed an open closed principle for our classes where we encourage extension instead of modification. An example would be if a new type of user comes up all we need to do is just extend our Account abstract class into a class for a new type of user. This enables us to be more flexible and also helps us to keep the codes easier to understand.