**Design Decisions:**

These are the key decisions that our project was based on followed by reasons why we thought that it would be best to implement our project in that way.

We followed the Single Responsibility Principal which allowed us to delegate features of our system into their own class

* This prevented repetition of code and kept the code as unambiguous as possible.

**For example**

The schedule class focuses on its own features such as producing a schedule and that’s it.

We followed an open closed principal for our classes and strongly encourage extension over modification.

* This allows our code to be more flexible and easy to understand for whenever we wish to grow the project.

**For example**

If we have a new teacher all we need to do is extend our user class for this new user.

We also included an abstract class for User hence all basic properties of the user (eg view schedule) can be held by the system

* It the also means certain properties can be extended into the several subclasses of user . Hence we can effectively limit the access of the different users

**For example**

A general user(staff) should not be able to create a schedule as this would be something only the principal can do.