**Functionality** – The system must require the user to check in to their room when it is time for them to use it, otherwise the reservation will be canceled. Users should be able to reserve rooms, cancel reservations and view their reservations. The system should alert users when they have a reservation soon. There must be different user types that have different privileges.

**Usability** – The user interface is not the priority for the project because we plan on integrating it into other systems that already have user interfaces. We will, however, create a user interface to demo it. The program should be intuitive and easy to use. Users should be able to do what they need to do in a very short amount of time.

**Reliability** – The system should never have problems reserving rooms unless the room isn’t in the system, then it will need to be added by an admin user.

**Performance** – The program is not very complex, so it won’t require much processing power. There should be little wait time in using the system.

Performance is a main criterion. The program should work fast, and efficiently.

**Supportability** – The system should be created using TDD. The system will need to be updated if room numbers change, rooms are added or subtracted or information about a room changes. These things will need to be updated by an admin user.

**Design Constraints** – The system must be able to integrate with other systems. The system must use a relational database, WPF user interface, C# and TDD.

**Implementation Requirements** – The programmers must adhere to the standards set out in the group contract. TDD must be used to guarantee that everything works correctly.

**Interface Requirements** – It must interact with any system the project owner wants to use it with.

**Physical Requirements** – It should be able to run on any PC/phone/tablet.