

		MoSCoW				Justification
	FURPS Requirements	M	S	C	W	
ID	Functional	Must	Should	Could	Wont	
00F	Faculty members can request preferred classes and time slots to the chair, but the chair has the final decision.					Instructors must be able to submit their availability
01F	Faculty can request to switch sections if they are dissatisfied with their schedule.					We assume that this happen only in emergency situation because instructor preferences should already be considered at the time of schedule creation.
02F	System will not let you have conflict					Makes sense because you cannot schedule two classes or events in one location at the same time
03F	Implement a college-wide standard matrix					College scheduling policy
04F	real-time room availability tracking (if multiple users)					Too complicated for the course project
05F	Workload distribution system for Regular Faculty					The insitution must ensure time distribution among faculty members is even
06F	Hide user login information. Such as when in login page "****"					Would be nice to have, but not important to functioning of the whole program
	Usability	Must	Should	Could	Wont	
00NF	User friendly way for faculty to submit preferences and requests.					We don't have emphasis on UI design
01NF	Easy way to notify the chair about conflicts					email can be used instead, but will be helpful if it can be implemented
02NF	Easy method to contact other department chairs					It wont be needed since this schedule is for computing department
03NF	Integration with other college services					We will not have access to other college services.
04NF	Create user documentation					Users must have a way to learn how to use the application. This should be accessible
	Reliability	Must	Should	Could	Wont	
05NF	Automatic back-up					We could make a back-up system to protect data but it is unlikely that there will be data loss.
06NF	Implement authentication for logging in					We must be able to verify who the user is and what permissions they have.
	Performance	Must	Should	Could	Wont	
07NF	Scalability					We're only working in the scope of one department or faculty so its not necessary that it should scale further unless requirements changes
	Supportability	Must	Should	Could	Wont	
08NF	Create system documentation					Must have for maintainability
09NF	Error Logging into a text file					If we're using a console-based ui, then we cannot output straight to cout. So required way to check what happens logic wise in another way
10NF	Account recovering					We don't store user emails or phone number for the identity confirmation
11NF	Follows OOP paradigm					Must have for maintainability