

Principle/Pattern name (type) Ex.: “Polymorphism (GRASP)”	Classes that are involved Ex.: “Animal, Cat, Dog”	Additional comment Ex.: Cat and Dog inherit from Animal
Layer Architecture	<ul style="list-style-type: none"> - Form - ControllerManager - GeneralLogic - StudentLogic - DonorLogic 	The Form uses the controller as a gateway to other classes’ logic. For the UI to NOT modify anything in the logic engine explicitly.
Polymorphism	<ul style="list-style-type: none"> - IUser - Student - Donor - Guest 	Student, Donor, and Guest inherits from IUser interface.
PureFabrication	<ul style="list-style-type: none"> - UserState 	Keeps track of the state of the application so that the UI can Update its Interface based on user type.
Factory/Dynamic Data Allocation	<ul style="list-style-type: none"> - Project - Individual Project - Club Project - ProjectDataDictionary - ProjectFactory 	Create projects based on project type name. When attempting access, a project gets all existing projects within the assembly.
Observer	<ul style="list-style-type: none"> - Form - ControllerManager 	sends a message to the form whenever the user has been changed, while the form listens and updates based off this.
Creator	<ul style="list-style-type: none"> - StudentLogic 	Creates applications for a scholarship. Also creates the projects using the factor class.