

Options:

*Information Expert* - knowledge of having data and who has the data

*Creator* - responsibility to create another class if aggregates, contains, records, or has initializing data of another class (expert)

*Controller* - Assigns responsibility of dealing with system events that represents the overall system or a use case scenario (non-ui)

*Low Coupling* - assign responsibilities so that coupling remains low

*High Cohesion* - keep objects appropriately focuses and manageable, used in support of low coupling

*Indirection* - supports low coupling and reuse by assigning mediation to an intermediary object

*Polymorphism* - responsibility for defining variation of behaviors based on type is assigned to the type for which this variation happens, use polymorphic options instead of branching

*Pure Fabrication* - does not represent a concept in problem domain, only made to achieve low coupling, high cohesion, and reuse potential

Class Name	StartupController
GRASP Pattern	Controller
Description	A controller for the first time startup of the application. Handles user input relating to user registration.

Class Name	RecipeController
GRASP Pattern	Controller
Description	A controller for the recipe list. This class handles input events directed at the Recipe portion of this application.

Class Name	FoodController
GRASP Pattern	Controller
Description	A controller for food types. This class handles input events directed at adding, editing and removing foods from the food database accessor.

Class Name	FoodInstanceController
GRASP Pattern	Pure Fabrication, Controller
Description	This class is an abstract class which is used as a framework for

	its children which are the NutritionController, RecipeController, ShoppingController, and PantryController. This class has a map of ids and FoodInstances which all child Controllers use to manage data.
--	---

Class Name	PantryController
GRASP Pattern	Controller
Description	A controller for the pantry. This class handles input events directed at the Pantry portion of the application.

Class Name	NutritionController
GRASP Pattern	Controller
Description	A controller for nutrition tracking. This class handles input events directed at the Nutrition portion of the application.

Class Name	ShoppingController
GRASP Pattern	Controller
Description	A controller for the shopping list. This class handles input events directed at the shopping list portion of this application.

Class Name	Services
GRASP Pattern	Creator
Description	This class handles the creation of services in the business layer.

Class Name	UserService
GRASP Pattern	High Cohesion
Description	This class handles registering the user upon first time setup. The class only has a few methods and data specifically relating to this functionality.

Class Name	User
------------	------

GRASP Pattern	High Cohesion
Description	This class groups together information relating to the user such as name, weight, and gender. Putting all this in one class keeps our program highly cohesive.

Class Name	RecipeService
GRASP Pattern	High Cohesion, Low Coupling
Description	This class handles functionality relating to the recipe portion of the application. It tries to have minimal connection with other classes interacting with the RecipeController and the database accessor.

Class Name	Recipe
GRASP Pattern	Information Expert, High Cohesion
Description	This class parallels a real life recipe. It holds data and methods relating to capturing the recipe.

Class Name	FoodService
GRASP Pattern	High Cohesion, Low Coupling
Description	This class handles functionality relating to food types within the application. It tries to have minimal connection with other classes interacting with the FoodController and the database accessor.

Class Name	Food
GRASP Pattern	Information Expert
Description	An information expert for food types. This class contains all necessary information to store a food type object.

Class Name	IdService
GRASP Pattern	Information Expert, Single Responsibility
Description	This class handles assigning unique identifiers to objects which

	need them.
--	------------

Class Name	IdObject
GRASP Pattern	Pure Fabrication, High Cohesion
Description	An abstract class which handles the aid attribute so that all the other objects which need ids can inherit from it.

Class Name	FoodInstance
GRASP Pattern	High Cohesion
Description	Parallels an actual food in real life as opposed to the abstract idea of a food. Used to keep track of what is in the pantry, recipe ingredients, and shopping list items. The nutrition log has a special version of this class which extends FoodInstance.

Class Name	PantryService
GRASP Pattern	High Cohesion, Low Coupling
Description	This class handles functionality relating to the pantry within the application. It tries to have minimal connection with other classes, interacting with the PantryController and the database.

Class Name	Goal
GRASP Pattern	High Cohesion
Description	This class holds data related to a nutrition goal a user has set. All methods and data related to a user goal are stored in this class.

Class Name	NutritionInstance
GRASP Pattern	High Cohesion
Description	This class is used in the Nutrition log to store data about what the user consumed when. The class inherits from FoodInstance and adds functionality related to time of consumption.

Class Name	FoodInstanceService
GRASP Pattern	High Cohesion, Low Coupling
Description	This class handles Food Instances. All methods relating to Food Instances are in one place along with the Map of Food Instances. This helps avoid coupling between the FoodInstances and the classes which use them.

Class Name	NutritionDescriptor
GRASP Pattern	Information Expert
Description	An information expert for nutrition information. This class contains all necessary information regarding nutrition.

Class Name	NutritionService
GRASP Pattern	High Cohesion, Low Coupling
Description	Inherits from FoodInstanceService. This class handles functionality relating to the Nutrition Log within the application. It tries to have minimal connection with other classes, interacting with the NutritionController and the database accessor.

Class Name	ShoppingService
GRASP Pattern	High Cohesion, Low Coupling
Description	This class handles functionality relating to the Shopping List within the application. It tries to have minimal connection with other classes, interacting with the ShoppingController and the database accessor.

Class Name	UserDBAccessor
GRASP Pattern	Low Coupling
Description	This database accessor helps separate the service from the database and keeps the services separated from each other.

Class Name	FileDBSingleAccessor
GRASP Pattern	Pure Fabrication
Description	This abstract class implements methods for database accessors which do single accesses.

Class Name	RecipeDBAccessor
GRASP Pattern	Low Coupling
Description	This database accessor helps separate the service from the database and keeps the services separated from each other.

Class Name	FoodDBAccessor
GRASP Pattern	Low Coupling
Description	This database accessor helps separate the service from the database and keeps the services separated from each other.

Class Name	NutritionInstanceDBAccessor
GRASP Pattern	Low Coupling
Description	This database accessor helps separate the service from the database and keeps the services separated from each other.

Class Name	FileDBListAccessor
GRASP Pattern	Pure Fabrication
Description	This abstract class implements methods for database accessors which access multiple elements at a time.