Receive userInput data from the view and notify the model through the controller. Controller Model LogController - log:Log - foodController:FoodController - dateObj:Calendar + getCal(): dateObj + setCal(int year, int month, int day): void + calToString(): String + getLogOnDate(): ArrayList<String> + addLogEntry(String IFood, double IServings): void + getFoods(): List<String> + logWeight(String weight): String +logCalorieLimit(String calorie): String FoodController - foodCollection: FoodCollection - foodFactory: FoodFactory View + updateLog(LoggableItem entry, action) + processException(Exception e) + changeViewDate(String date) + modifyFoodLibrary(FoodComponent food, action) + setNewBenchmark(int benchmark) + addBasicFood(String fName, String fCal, String fCarbs, String fProtein, String fFat): String + getSpecificFood(String name): FoodInterface + addRecipe(String rName, String ingredients):String + getFoods(): Map + deleteFood(String keyword): String updateLog: Converts the FoodComponent to a string so the log can use it. processException: receives an exception from the view and handles it changeViewDate: receives an update of the viewDate and notifies the model modifyFoodLibrary: takes a food and an action and tells the model library what change to do setNewBenchmark: takes an integer (either for a weight or a calorie goal) and calls updateLog after converting the value to conform to the 'LoggableItem' standards.