

# **CoUPon: Check Out UP Orgs Now!**

## **Software Architecture**

Submitted to:

Asst. Prof. Ma. Rowena C. Solamo  
Faculty Member  
Department of Computer Science  
College of Engineering  
University of the Philippines, Diliman

Submitted by:  
Araya, Carl  
Comia, Camille  
Florita, Nikki

In partial fulfillment of Academic Requirements  
for the course  
CS 191 Software Engineering I  
of the  
1<sup>st</sup> Semester, AY <2017-2018>

## **Revision Control**

### **History Revision:**

<b>Revision Date</b>	<b>Person Responsible</b>	<b>Version Number</b>	<b>Modification</b>
12/1/17	Nikki Florita	1.0	Initial Document & UI classes
12/3/17	Camille Comia	1.1	Added some description to Controller classes and description of the system
12/3/17	Nikki Florita	1.2	Accomplished Data Design classes

### **Purpose:**

The document aims to provide an overview on the software architecture of CoUPon.

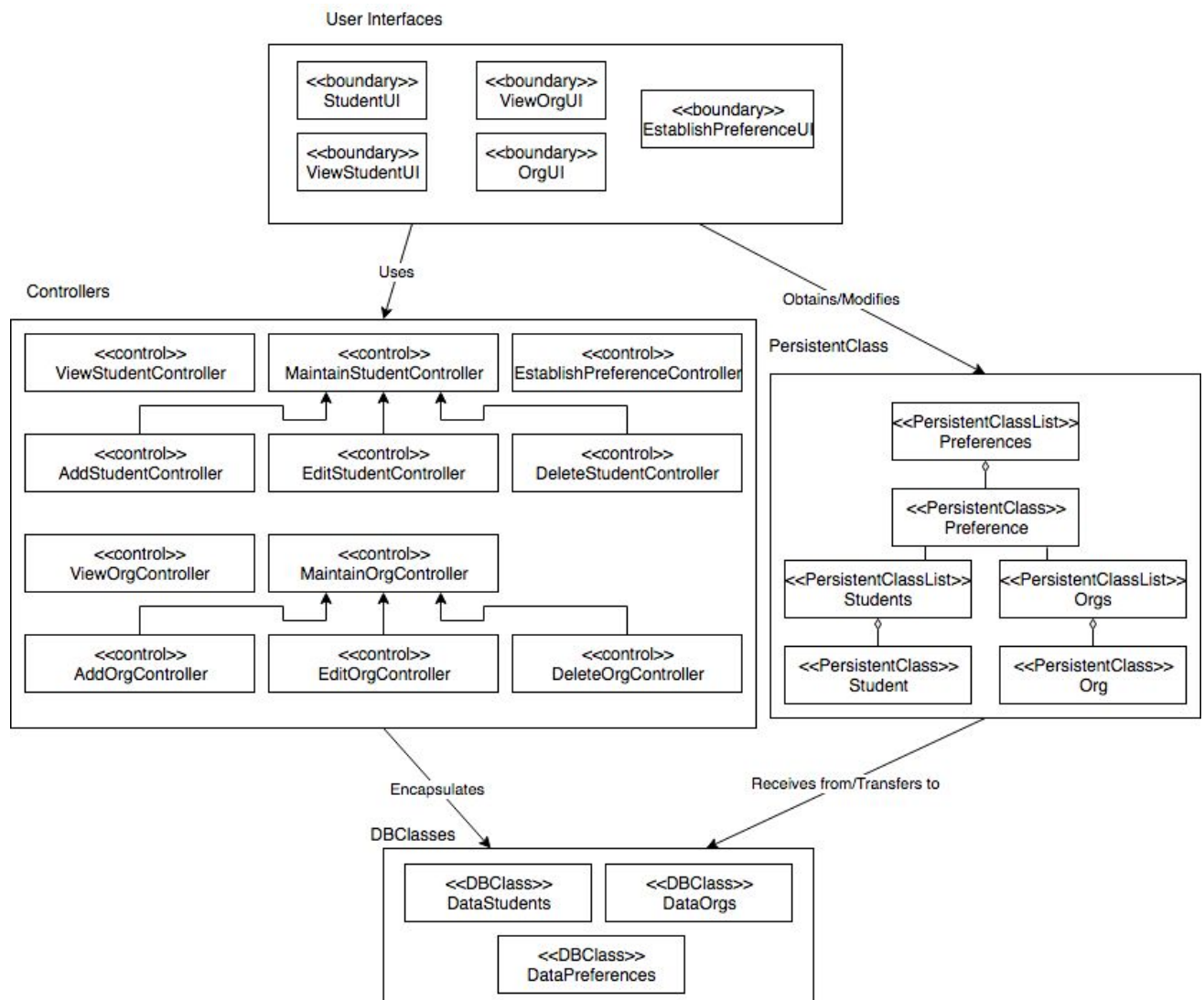
### **Audience:**

Software engineers who wish to know the software architecture of CoUPon.

**System Name:** CoUPon

**Description:** The system is an application designed for the convenience of registration for both the organizations and the students alike. It allows a student to send one application to any organization of their choice, so long as said org is a subscriber of the app. It also manages the directory of organizations affiliated to CoUPon as well as the database of student information that sent an application to their chosen organization.

**Software Architecture Diagram:**



*User Interface Design Classes:*

Screen Name	Description
StudentUI	Interface for user to input personal information for the organization
ViewStudentUI	Interface for an organization to view information about a student who wishes to apply
OrgUI	Interface for the organization to log in and to modify org data
ViewOrgUI	Interface for users to view description about different student organizations
EstablishPreferenceUI	Interface for users to select organizations they want to apply after filling out student form

*Business Logic Classes:*

Control	Description
ViewStudentController	Controller between ViewStudentUI and DataStudents class. This retrieves the Student data then sends it to the ViewStudentUI.
AddStudentController	Controller to add new Student to DataStudents. This class handles the input from the StudentUI and creates and adds a new Student entry with the input to DataStudents.
EditStudentController	Controller to edit a Student entry from DataStudents. This class handles the input from the StudentUI and proceeds to make the changes requested to an existing Student entry from DataStudents.
DeleteStudentController	Controller to delete a Student from DataStudents. This class receives request from StudentUI to delete a student and proceeds to do the task.
EstablishPreferenceController	Controller to link Student to Org by creating Preference
ViewOrgController	Controller between ViewOrgUI and DataOrgs class. This retrieves the Org data then sends it to the ViewOrgUI.
AddOrgController	Controller to add new Org to DataOrgs. This class handles the input from the OrgUI and creates and adds new Org entry with the input to DataOrgs.
EditOrgController	Controller to edit an Org entry from DataOrgs. This class handles the input from the OrgUI and proceeds to make the changes requested to an existing Org entry from DataOrgs.
DeleteOrgController	Controller to delete an Org from DataOrgs. This class receives request from OrgUI to delete an Org and proceeds to do the task.

*Data Design Classes:*

Class Name	Description
DataStudents	Contains Student entities for storage and retrieval
DataOrgs	Contains Org entities for storage and retrieval
DataPreferences	Contains Preference entities for storage and retrieval
Preferences	List containing Preference entities for UI class use
Preference	Entity that contains mapping of Student and Org entities
Students	List containing Student entities for UI class use
Student	Entity that contains student info to store in DataStudents and use for ViewStudentUI
Orgs	List containing Org entities for UI class use
Org	Entity that contains org info to store in DataStudents and use for ViewOrgUI