Benson Chen

Prof. Rawlins

Assignment 2

5/29/22

**Integration Model Diagram**

**Integration and Configuration**

**Requirement Specifications:**

The specifications and goals for the project are reviewed.

**Component Analysis:**

Find a similar project that aimed to achieve something similar to the current project and note what can be used.

**Requirement Modification:**

Based off the previous step, revise the original project’s specifications to the current project’s goals.

**System Design with Reuse:**

Use the components and modify and add what is needed.

**Development and Integration:**

We create the code and integrate what could be referenced from the existing program.

**System Validation:**

The program is finalized after testing and determined whether the specifications are satisfied.

**Requirement Specifications**

The LeopardWeb project has the goal of creating a scheduling system that will allow students, faculty, and admin to add courses, search for courses, print schedules and more.

• Database of users: the system should work for 100 students, 10 instructors, and 1 admin, however, we will test with fewer.

• Database of courses: this will contain information such as the CRN, course name, times, and instructor.

• Three types of users:

o student – can register, can see available courses and their own schedule.

o instructor – can see available courses and their own course roster.

o admin – can see everything, can edit courses/users/schedules.

• The system should include multiple semesters, print-out of schedule, scheduling preferences.

• The system as a whole and all components must be tested thoroughly.

The base class of the system is user with:

• Attributes: first name, last name, ID.

• Methods: set function for each attribute, and a function to print all info for the object.

**Component Analysis:**

* With the common usage of school and course databases, finding a site that we can model our LeopardWeb system would be beneficial. Gathering information from multiple sources can broaden our options on what we are able to add and how we would like to implement our own system.

**Requirement Modifications:**

* After finding another site to model our system after, we then make the necessary edits to our requirements to match the requirements of the already existing system. This will help make our model more efficient.
* As every school portal is unique in their own way, we must focus on which specifications we need and how each site runs differently.

**System Design with Reuse:**

After gathering ideas and components we decide on reusing, we integrate our own system design.

There will be three derived classes:

• All derived classes must contain any additional attributes and appropriate set/get functions.

• student – the student class will have functions that allow them to search courses, add/drop courses, print their schedule.

• instructor – the instructor class will have functions that allow them to print their schedule, print their class list, and search for courses.

• admin – the admin class will have functions that allow them to add courses to the system, remove courses from the system

**Development and Integration:**

This phase requires the creation and build of the code and databases. Three derived classes from user with unique functions would be created for each. After initial completion of all components of the code, these components can be combined and tested. These components include specific user classes and functions.

**System Validation:**

This stage involves the latest versions of the system to resolve any issues and bugs that would occur and finalize and update the system for final usage.