**Name: Kaycee Salgueiro**

**Course: ELEC3225 Applied Programming Concepts**

**Assignment #2**

**Waterfall model**

1. Requirements definition
   1. Proposed timeline: a few days.
   2. System requirements are already defined and given to the programmers. The requirements detail that the system should work for 111 users that include students, instructors, and an administrator. The system will include information on school courses such as the CRNs, course name, time/dates, and instructor. The system must include courses from multiple semesters and have the ability to print schedules and have schedule preferences.
2. System and software design
   1. Proposed timeline: a week.
   2. The base class of the system must be called “User” and must have three derived classes called: “Student”, “Instructor”, and “Admin”. The derived classes will each inherit a first name, last name, and ID attribute from the base class “User”. Each derived class also includes its own set of functions, some that overlap between users and others that are only specific to that class. Functions include searching for courses, printing schedules, add/removing students from classes, removing courses from the system, etc.
3. Implementation and unit testing
   1. Proposed timeline: 1 ½ to 2 months.
   2. Each unit should be created in the following order:
      1. Base class “User” and derived classes “Student”, “Instructor”, and “Admin” that inherit first name, last name, and ID attributes.
      2. Functions for derived class “Student” which include search course, add/drop course, and print schedule.
      3. Functions for derived class “Instructor” which include search course, print schedule, and print class roster.
      4. Functions for derived class “Admin” which will include add/remove course, add/remove user, add/remove student from a course, search/print rosters and courses.
      5. Upload all users to database.
      6. Upload courses for Fall, Spring, and Summer semester to database.
      7. Create user interface and user preference options.
4. Integration and system testing
   1. Proposed timeline 1 ½ to 2 months.
   2. The units should be integrated and tested in the following order:
      1. Functions for “Student” class should be integrated and tested with test classes.
      2. Functions for “Instructor” should be integrated and tested with test classes.
      3. Functions for “Admin” should be integrated and tested with test users and classes.
      4. User information should be uploaded to database and test to ensure each class’s set of functions still works as intended with test classes.
      5. Course information for each semester should be added and tested with each derived class functions.
      6. User interface should be graphically created along with adding the option for scheduling preferences. This should be tested with users who have scheduling functions.
5. Operation and maintenance
   1. Proposed timeline: as long as the program remains active.
   2. So long as the program is active, it should be continuously tested by ensuring functions for each derived class still work as intended. Courses should also be updated each year and the old courses should be moved to archives that are still accessible to be viewed by users. Any bugs found should be addressed immediately, and users no longer attending or employed by the school should be removed.

Comments (Andy):

1. Everything looks good and about the same basically as mine, but with way more details.
2. Proposed timeline for Implementation and unit testing/integration and system testing are long but if this was a continuous service would make sense.
3. Requirements and design sections look good, I totally agree.