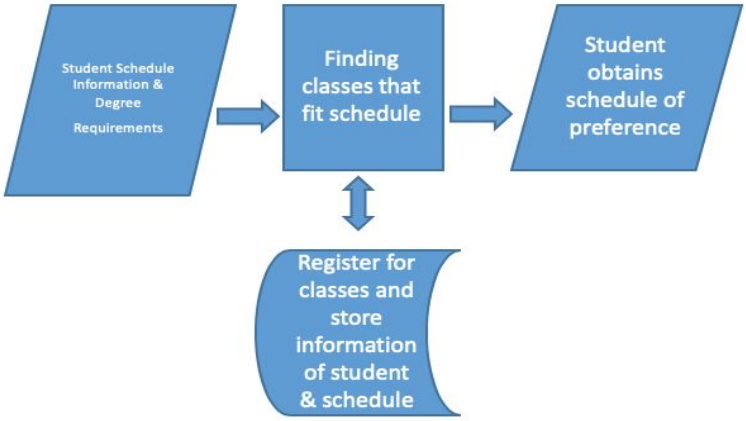
Nikhil Wora
February 20th, 2020
Professor Heidenblad

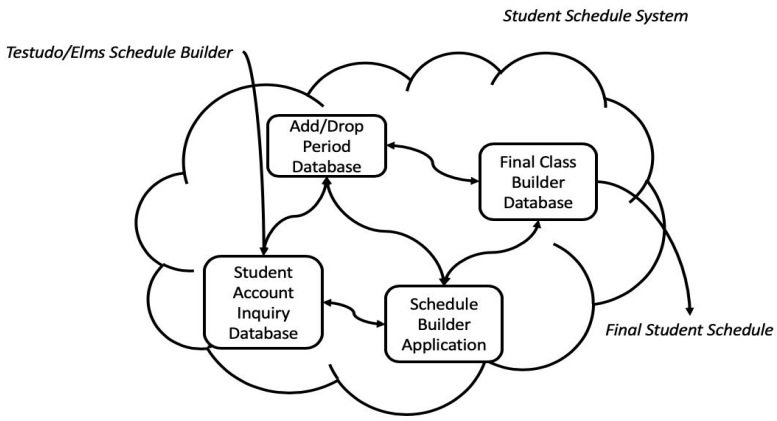
HW-00: System Architecture

Question 1:



This IPO Model reveals:
Input: student finding what classes he needs based on degree requirements.
Process: students find classes that accommodate their schedule.
Storage: stores class and student information.
Output: student obtains the schedule of their preference.

Question 2:



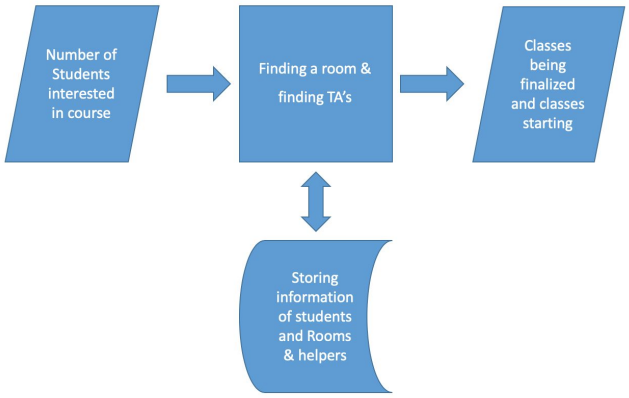
This Generic System Model reveals:

Interface: Elms schedule builder allows us to find specific classes that fit our schedule in the specific time slots we need.

4 Components: Student Account Inquiry, Add/Drop Period, Schedule Builder, Final Class Schedule.

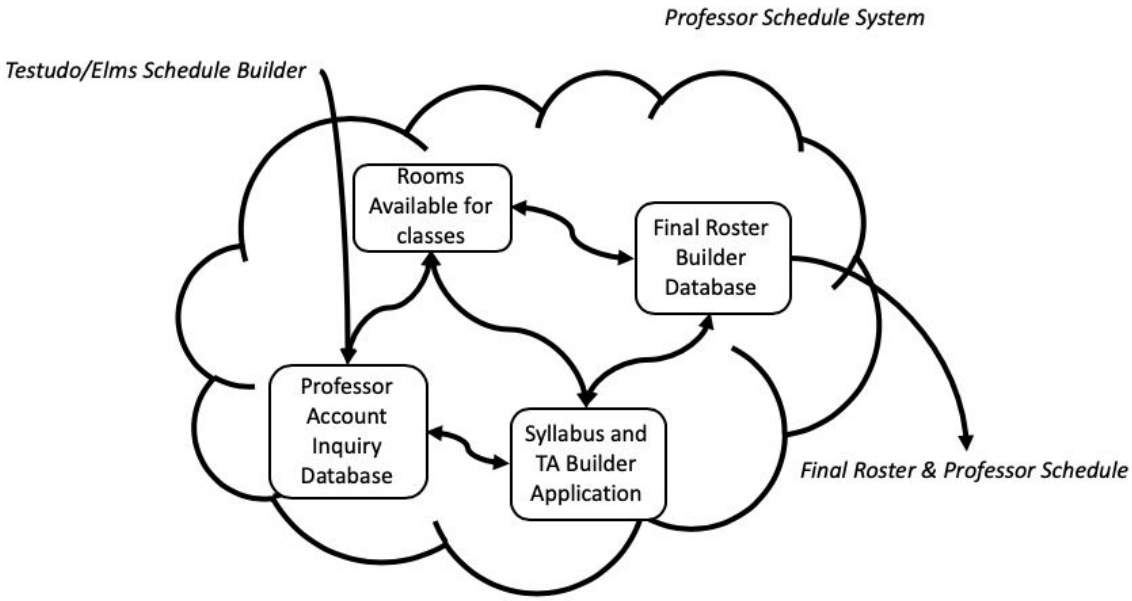
The student account inquiry is where we actually register for classes, however we have an add/drop period where we can add or remove classes. In addition, we have a schedule builder option that allows us to view what our schedule would look like if we did confirm, which leads us to our final schedule that is presented when we confirm our schedule.
The environment of our system would be the actual database we are using in order to make our schedule.

Question 3:



This IPO Model reveals:
Input: the number of students that are interested in the course.
Process: finding a specific room and finding helpers within the course.
Storage: stores information about specific students interested, rooms available, and helpers for the course.
Output: classes being finalized and classes starting.

Question 4:



This Generic System Model reveals:

Interface: Elms schedule builder allows us to find specific students that fit our schedule in the specific time slots we need.

4 Components: Profesor Account Inquiry, Rooms available for classes, Syllabus and TA Builder, Final Class Schedule & Final class roster.

The elms builder allows the professor to access their courses and times they are available to teach lectures and labs. In addition, it will allow professors to see what rooms are available for each class. This will lead to professors to build syllabuses and ta applications for help. This will lead to final rosters and final course schedules being built.

Question 5:

I would believe that there were many challenges you would face if the system incorporated professor and student databases together. However, I believe that all databases already do that in order to actually make student schedules. To be more precise, a system is a regularly interacting or independent group of items forming a unified whole which would be a class schedule. With each linked component, we are defining the boundary of the system. Everything outside of the system would be considered the environment and not a component. In order to change the two systems, we would have to store the memory in a place that is accessible for each system so it

can use the stored data to produce schedules. The CPU would be changed and would have both student and professor access which then would change the input/output mechanisms and would allow for both types of people to implement their components into the system. This would all make one schedule which both accommodates the students and the professors.