Waterfall Model

Requirement defintion:We are gathering the information for a college to use a website call leopardweb in order for college students, instructors, and admins in order to use it to select courses, searching a course, and print the sechudle.

Main functions:

User : Student, instructors, admins

Each user will have an ID number as well as there names

students, instructors, and System software

Wentworth Secdudle system(leopardweb): Design the sechule system data structure fornd admins

Determine the system such as it includes all the semester season , sechulde preference, and printing out sechdule

Make sure that we have a class call user so that we have the main structure of Wentworth students and workers. Also derive the user to make sure we have different attributes as students are mainly picking class, instructors search their class time to teach, and admins are in charge of adding courses

Design

Student: // methods

```
void searchClass()  { void searchClass()  {
std::cout << "searching a course ." << std::endl; void addClass()  {  std::cout << "Adding a course I need " << std::endl; }
void dropClass() {
std::cout << "dropping a course I may not need " << std::endl;}
void printsechudle() {
std::cout << "Printing the schedule so I can know the time for classes " << std::endl; }
```

Implentation

For the derived class we will have student and its structure is search clas,/ ad/dropd clas, print their class time

Instructors will have search course to figure what subject topic their teaching, print the class list there teaching this semester, and print sechule to figure out the time they need to teach

Admins will have the authority to add any courses that instructors will teach and students need to take and figure out the time to atten. They can also remove a course so that they believe is no longer since it can be due to a better class. Instructors will be removed due to time delay in so another instructor will teach this class.

students may be removed due to dropping class and added for picking this class.

Design for Instructors

```
// methods
void printsechudle() {
std::cout << " print the sechudle for the class I teach " << std::endl;}
void printclass() {
std::cout << "print the class I  will be teaching" << std::endl;}
void searchclass() {
std::cout << " search the course I will be  teaching " << std::endl;}
```

Design for Admins

```
void addCourse()  {
std::cout << "Adding course..." << std::endl:   void removeCourse()  {
std::cout << "Removing course..." << std::endl;   } void addUser()  {
std::cout << "Adding user..." << std::endl:      }void removeUser()  {
std::cout << "Removing user..." << std::endl;  }
void addRemoveStudentFromCourse()  {
std::cout << "Adding/Removing student from a course..." << std::endl;  }
void searchPrintRosters()  {
std::cout << "Searching and printing rosters..." << std::endl;    }
void searchPrintCourses()  {
std::cout << "Searching and printing courses..." << std::endl;    }
```

Unit testing: use the main function for the Leopardweb to make sure Users are logged as a normal pe

```
student.name("")

student.ID("")

cout << "\Student Information:" << endl:

student print.info():

// this still works for instructors and admins it just there system as they have a different role for using L
```

Integration:

Once all the detail of the code is written that indlues make sure all the detail of the program is written for every data on the Leopardweb

in terms of degubbing shortage of a codeOperation and maintance: If there's any mistake check that all the correct format is pluged in a

Incremental model:

Outline Description:We are designing a course and sechudle systems for students, instructors , and admins

Specification

Students: look for course, add a course, drop course , and print sechudle

Instructors: searching a course, access their course, and print sechudle

Admins:Access to course, add/drop student as well as class , and print sechdule

Development:

User class attributes: Name, ID

Derived from User: Student, Instructor, admin

Intial Version for example on students as a drafr:

```
Student:  // methods  void searchClass()  }

void dropClass() {

}

void printsechudle() {

<< std::endl; }
```

```
Intermediate coding :Student:  // methods  void searchClass()  { void
searchClass()  {

std::cout << "searching a course ." << std::endl; void addClass()
{  std::cout << "Adding a course I need " << std::endl; }

void dropClass() {

std::cout << "dropping a course I may not need " << std::endl;}

void printsechudle() {
```

Valadation: Ensure that once all the code has no errors so that the result button will pop up once the code is debugged otherwise there will be some mistake that can cause a bigger task.

Final version:

```
student.name("")
student.ID("")
cout << "\Student Information:" << endl:
student print.info():
// this still works for instructors and admins it just there system as they have a different role for using Leopardweb
```

Intergration and Configure model:

When using leopard Web, we are ensuring that this platform is for students, instructors, and administrators as they are the main targets for a college website to access classes, and mostly the scheduled class.

MySQL is the website I've been considering using for a scheduling system. The way to configure them is to make a role for student, instructor, and admins. Then create a schedule table that will represent the semester, the type of classes to search for. Keep in mind that students, instructors, and admins will have different role as they have their own control. student and its structure are search class,/ ad/drop class, print their class time

Instructors will have search course to figure what subject topic their teaching, print the class list their teaching this semester, and print Schedule to figure out the time they need to teach

Admins will have the authority to add any courses that instructors will teach, and students need to take and figure out the time to attend. They can also remove a course so that they believe it is no longer since it can be due to a better class. Instructors will be removed due to the time delay so another instructor will teach this class. Students may be removed due to dropping class and added for picking this class.

Citation: *10 best database management systems for software developers*. GUVI Blogs. (2024, March 22). https://www.guvi.in/blog/database-management-systems-for-developers/