

Course Project

**University Registration System**

**Course Code:** SWE331

**Course Title:** Object Oriented Software Development

|  |  |
| --- | --- |
| Mahmuda nasrin mukti | 151-35-1017 |
| Section | B |
| Supervisor Name | Md. Alamgir Kabir |
| Department | Software Engineering |

**Date: 06-12-17**

Table of Contents

[chapter 1 Introduction 3](#_Toc500316583)

[1. Introduction: 3](#_Toc500316584)

[1.1 About the system: 3](#_Toc500316585)

[1.2 Purpose: 4](#_Toc500316586)

[1.3 Scope: 4](#_Toc500316587)

[1.4 Vision: 4](#_Toc500316588)

[1.5 Why the system is necessary: 4](#_Toc500316589)

[1.6 Proposed solution: 5](#_Toc500316590)

[Chapter 2 System Analysis 5](#_Toc500316591)

[2. System Analysis 5](#_Toc500316592)

[2.1 Actor goal list: 5](#_Toc500316593)

[2.2 Use case model: 6](#_Toc500316594)

[2.3 Use Case Description: (Brief 6](#_Toc500316595)

[2.3.1 Pay registration fee 6](#_Toc500316596)

[2.3.2 Manage course 7](#_Toc500316597)

[2.3.3 Process registration 7](#_Toc500316598)

[2.3.4 Maintain information 8](#_Toc500316599)

[2.4 Use Case Description: (Detailed) 9](#_Toc500316600)

[2.4.1 Pay registration fee 9](#_Toc500316601)

[2.4.2 Manage Course: 10](#_Toc500316602)

[2.4.3 Process registration: 11](#_Toc500316603)

[2.4.4 Maintain information: 12](#_Toc500316604)

[2.5 System Sequence Diagram 13](#_Toc500316605)

[2.5.1 pay registration fee 13](#_Toc500316606)

[2.5.2 manage course 14](#_Toc500316607)

[2.5.3 Process registration 15](#_Toc500316608)

[2.5.4 Maintain information: 16](#_Toc500316609)

[2.6 Domain/concept model: 17](#_Toc500316610)

[2.7 Activity Diagram: 18](#_Toc500316611)

[Chapter 3 System Design 19](#_Toc500316612)

[3. System Design 19](#_Toc500316613)

[3.1 Sequence Diagrams 19](#_Toc500316614)

[3.1.1 Sequence Diagram for pay registration fee 20](#_Toc500316615)

[3.1.2 sequence diagram for manage course: 21](#_Toc500316616)

[3.1.3 Sequence diagram for process registration: 21](#_Toc500316617)

[3.1.4 Sequence diagram for maintain information: 22](#_Toc500316618)

[3.2 Class diagram: 23](#_Toc500316619)

# chapter 1 Introduction

# Introduction:

The name of the system is university registration system (URS). The university Registration System allows students, through Internet, to register, drop, or add courses within the registration period and the Advisor to do advising for the students by approving/rejecting requested courses by the student or recommending some others.The System provides for students the option to register courses, without the advisor’s prior approval, offered by their colleges during the scheduled registration periods. The students can modify their course selection by adding and/or dropping courses. However, when the registration period is over, all previously registered courses by the students will be viewable in the system.

## About the system:

A course catalog containing list of course offering. using course catalog student can easily know Information about each course, such as professor, department, and prerequisites. They can select courses or change schedule time. professors also know which course they will be teaching.The university management system. Student can select courses, cancel courses, change schedule time .The professor also select courses, de-select courses. The register offer courses, assign teacher, change courses and close registration.

The student may use the system to register, if the following conditions are fulfilled:

* The registration is done during the registration schedule, drop and add period, as published by Admission and Registration Deanship;
* The student should have registered all courses as outlined in his/her study plan and should not have had any exceptions during his/her study period
* The student status is not suspended, prohibited, canceled, or does not have an academic Block from the Admissions & Registration Deanship;
* The student has an Academic Advisor entered on the Oracle System;
* The student has regular financial status;
* The student has no academic warnings.
* The student email is active.
* The advisor or student cannot register completed courses, i.e. courses have already passed.

## Purpose:

The purpose of the registration process is to determine which students will be taking courses within the university and for administration to keep its records up-to-date.

## Scope:

University Registration System is a System which goes above a beyond other similar Software. Over the years, these modules have been developed to ensure that you get the best package for your district’s needs.

1. Student’s registration – straight forward, intuitively easy.
2. Teacher’s registration – saves precious time and efforts.
3. Classes and schedule systematization.
4. Implement changes easy as one, two, three.
5. 100 percent work-ability and safety.

## Vision:

Provides a better solution and policy for course registration. Reduces much time, cost and resources required in processing registration information of students.

Allow students to access the system during registration time to register for courses. Add or drop the registered courses. Check tuition fee and their academic history.

## Why the system is necessary:

The system is necessary for course registration, select course, de-select course ,change schedule, assign teacher ,close registration. To reduce our time and we can fulfill it easy .we can easily register my course at home. It cannot complicated. That’s why we need this system.

## Proposed solution:

1. Online database store
2. Online course registration

# Chapter 2 System Analysis

# System Analysis

## Actor goal list:

|  |  |
| --- | --- |
| Actor | Goal |
| Student | * Pay registration fee * Confirm prerequisite * Add or drop course * Confirm registration |
| Teacher | * Check prerequisite * Confirm registration * Confirm teach for course |
| Register | * Manage student information . * Manage teacher information. * Confirm registration validation |

## Use case model:

Fig-1: Use case model

## Use Case Description: (Brief

### Pay registration fee

Here students must pay registration course fee either he or she can’t get clearance. This is mandatory.

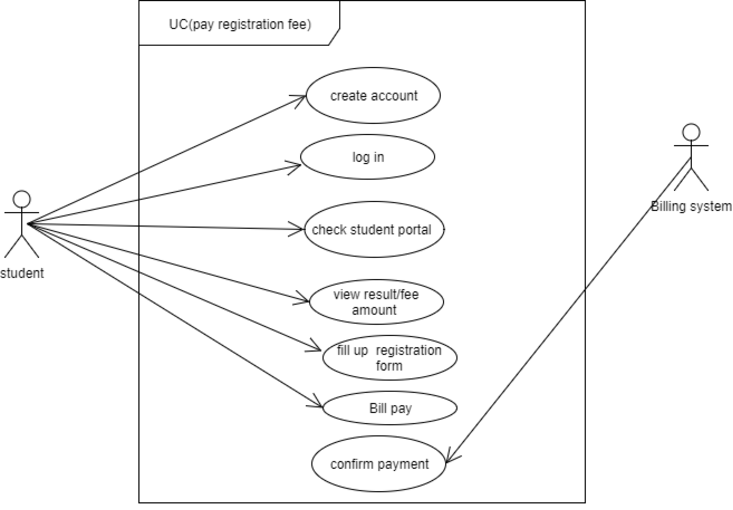


Fig-2: Use case model for pay reigstration fee

### Manage course

Here student can select or drop any course. they can select schedule time.

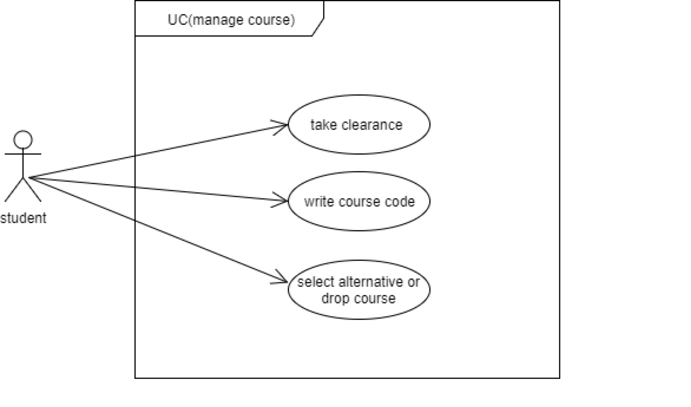


Fig-3: Use case model manage course

.

### Process registration

Here teacher can select course for teach.

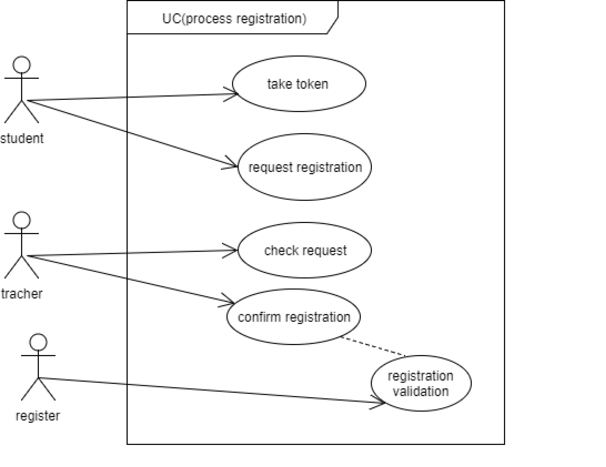


Fig-4: Use case model for proccess registration

### Maintain information

Here register will manage teacher information , student information And payment information .register will also maintain exam time and registration fee.

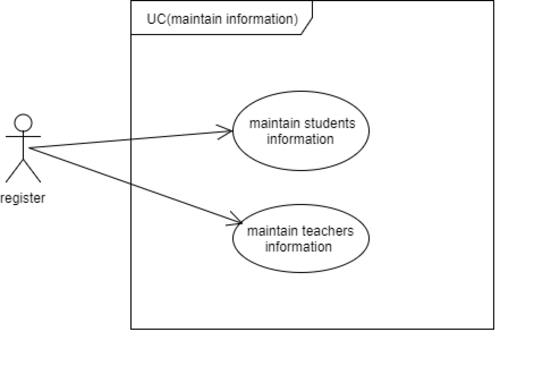


Fig-5: Use case model for maintain information

## Use Case Description: (Detailed)

### Pay registration fee

|  |  |  |
| --- | --- | --- |
| Use Case Name: | Pay registration fee | |
| Scenario : | Here student will pay registration fee. | |
| Brief Description: | Students will log in student portal. Then fill up the form and give money to the billing system .Billing system will give clearance with money receipt. | |
| Actor: | Student | |
| Precondition: | Log in the student portal and fill up the form. | |
| Post condition: | Access the system and registration course. | |
| Flow of events: | User | System |
| 1.First student log in student portal .  2.Students will check registration fee.  3.They will fill up the registration form and give Money. | * 1. Confirm log in.   2. View registration fee.   3. Confirm payment and give clearance. |
| Exception Condition: | * + 1. Without register fee students can not registration course.     2. Set not available.     3. If they can not payment successfully .     4. Registration time end. | |

### Manage Course:

|  |  |  |
| --- | --- | --- |
| Use Case Name: | Manage course. | |
| Scenario : | Here students will select , add or drop course. | |
| Brief Description: | Student will write course title and course code in clearance. They can take offering subject or alternative subject .They also can drop course. A student may drop or add new courses during the first week of the semester. To add a course, one must have the approval of the faculty advisor. | |
| Actor: | Students | |
| Precondition: | Must pay registration fee. | |
| Post condition: | Access system and registration course. | |
| Flow of events: | Actor | System |
| * 1. Students will write course code and course title.   2. They can drop ourse.   1.6 they can add alternative course. | 1.4System will confirm course.  1.5 Confirm drop course.  1.6.Confirm add course. |
| Exception Condition: | * + 1. Insert wrong course code.     2. Server problem.   1.1.7Registration time end. | |

### Process registration:

|  |  |  |
| --- | --- | --- |
| Use Case Name: | Process registration. | |
| Scenario : | Here student will request for registration and teacher will fill up the process . | |
| Brief Description: | Student will take a token and will request for registration .Teacher will confirm registration and register will check the registration validation. | |
| Actor: | Students, teacher, register. | |
| Precondition: | Must pay registration fee and select course code. | |
| Post condition: | Access the system and do registration. | |
| Flow of events: | Actor | System |
| 1.7Students will submit the token and clearance.  1.8 request for registration. | * 1. System will confirm selected course.   1.8 Confirm registration validation. |
| Exception Condition: | 1.1.8 registration information is not valid.  1.1.9 server error.  1.1.10 If student will not take token. | |

### Maintain information:

|  |  |  |
| --- | --- | --- |
| Use Case Name: | Maintain information | |
| Scenario : | Here register will manage all information of students and teachers. | |
| Brief Description: | Here register will maintain students information and teachers information. He can select teacher for teach course and also can manage schedule . | |
| Actor: | Register | |
| Precondition: | Students must pay registration fee and select course code .teacher will confirm the process of registration. | |
| Post condition: | Confirm registration validation. | |
| Flow of events: | Actor | System |
| 1.9 register will maintain student information.  1.10register will maintain teachers information,  1.11 register will confirm registration validation. | 1.9 system will view students information.  1.10 system will view teachers information.  1.11System will confirm registration validation |
| Exception Condition: | 1.1.11registration time end.  1.1.12. sever problem .  1.1.13Information is not valid. | |

## System Sequence Diagram

### pay registration fee

Use Case 1: Pay registration fee

Success Scenario

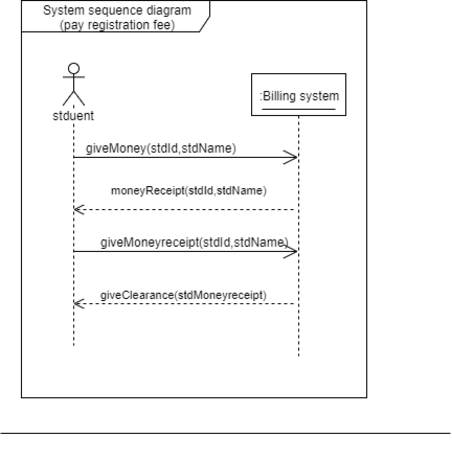


Fig-6: system sequence diagram for pay registration

fee

### manage course

Use Case 2: Manage User

Success Scenario

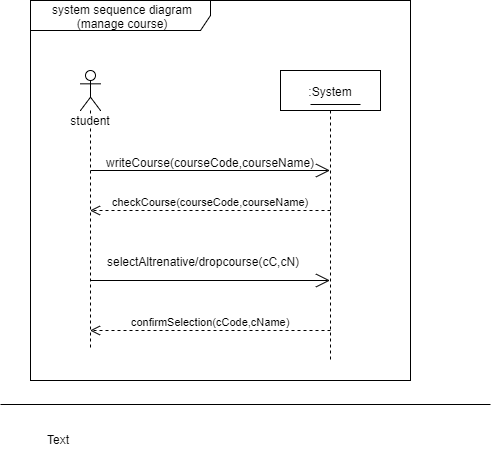


Fig-7: system sequence diagram for manage course

### Process registration

Use Case 3:Process registration

Success Scenario

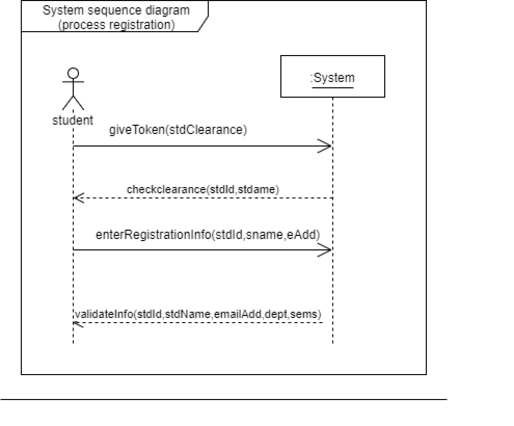


Fig-8: system sequence diagram for Process registration

### Maintain information:

Use Case 5: Maintain information

Success Scenario

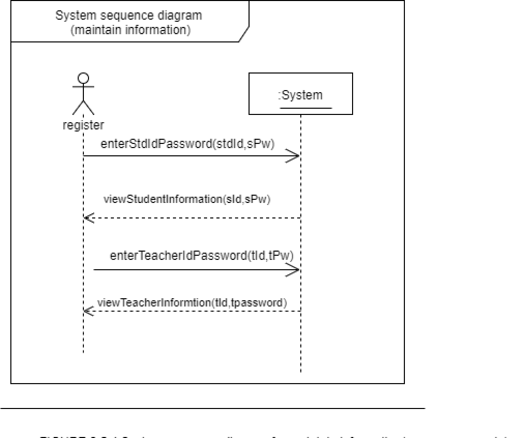


Fig-9 system sequence diagram for Maintain information

## Domain/concept model:

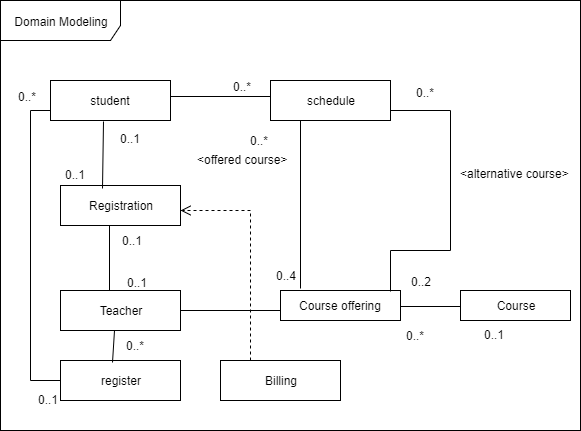


Fig-10: Domain/concept model

## Activity Diagram:

Fig-5: Use case model for maintain information

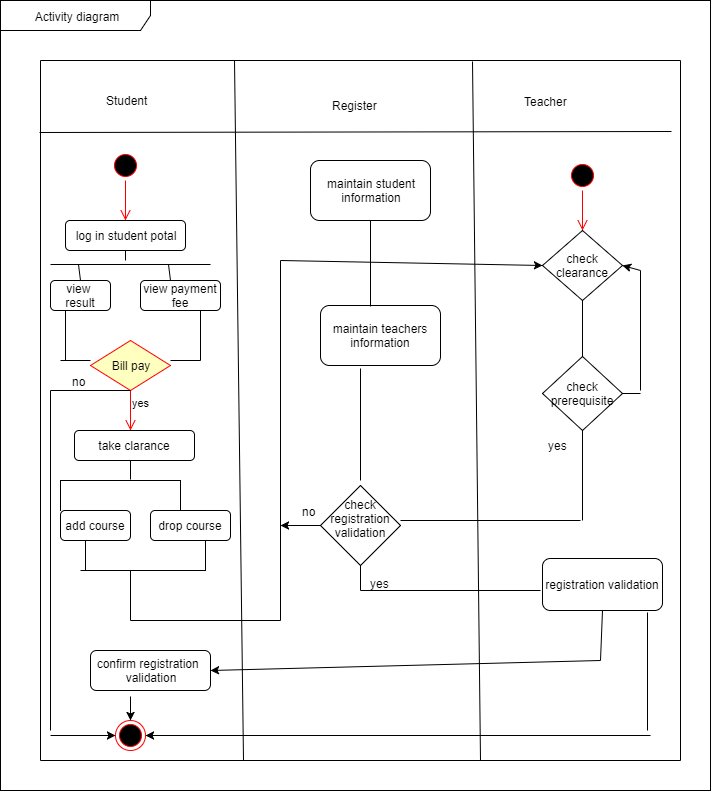


Fig-11 : Activity Diagram

# Chapter 3 System Design

# System Design

Design is a process that uses the product of analysis to produce a specification for implementing a system. Design is the logical description of how a system will work.

Design emphasizes a conceptual solution that fulfills the requirements, rather than its implementation. For example, a description of a database schema and software objects. Design ideas often exclude low-level or "obvious" details obvious to the intended consumers. Ultimately, designs can be implemented, and the implementation (such as code) expresses the true and complete realized design. The term is best qualified, as in object-oriented design or database design.

## Sequence Diagrams

The UML includes interaction diagrams to illustrate how objects interact via messages. They are used for dynamic object modeling. The term interaction diagram is a generalization of two more specialized UML diagram types:

### Sequence Diagram for pay registration fee

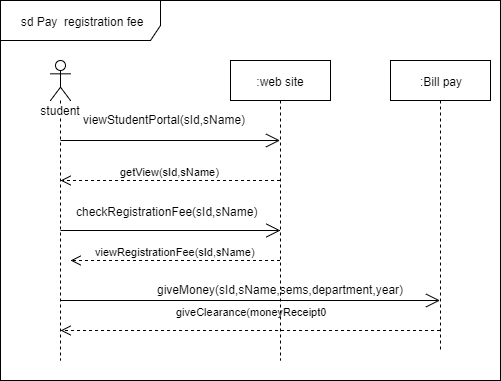


Fig-12 : sequence diagram for pay registration system

### sequence diagram for manage course:

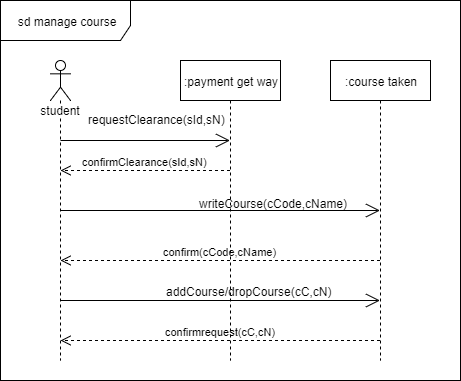


Fig-13: sequence diagram for manage course

### Sequence diagram for process registration:

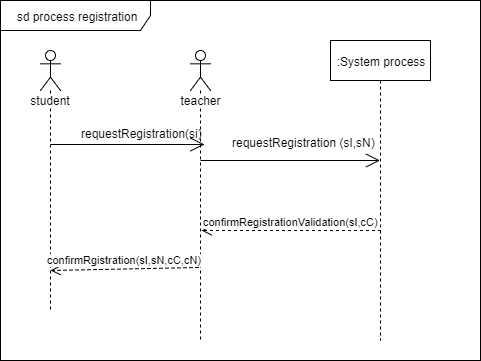


Fig-14: sequence diagram for process registration

### Sequence diagram for maintain information:

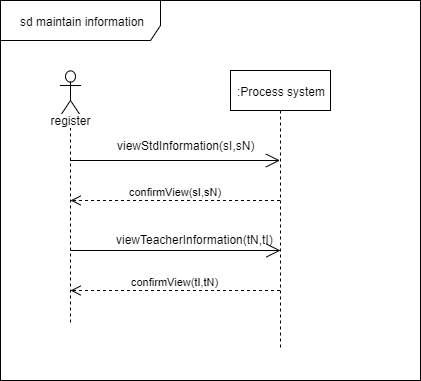


Fig-15: sequence diagram for maintain information

## Class diagram:

Class or structural diagrams define the basic building blocks of a model. They are used for static object modeling, describing what attributes and behavior it has rather than detailing the methods for achieving operations.

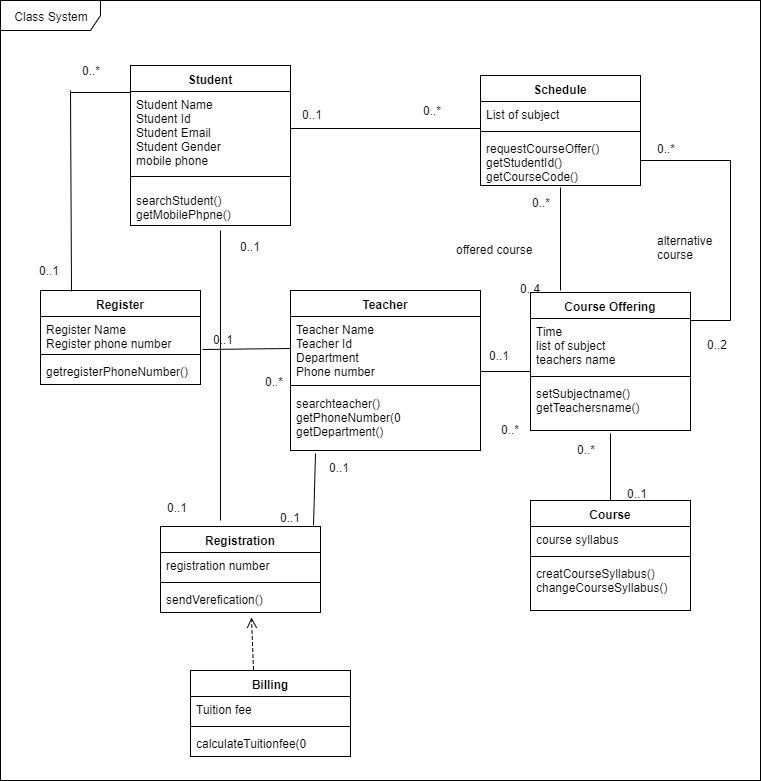


Fig-16:Class Diagram