

**MicroLink Information Technology College**

**Department of Computer Science**

**Student Teaching Performance Evaluation and Information System [STPE-IS]**

**Requirement Analysis Document (RAD)**

**For Microlink Information Technology College**

**Prepared by:**

1. AsnakechSeleshi
2. FikreBerhanu
3. DegalemZeleke
4. Fatuma Mohammed
5. Marta Abebe

**Advisor:**

EheremTilahune

July 24, 2021

Revision History

|  |  |  |  |
| --- | --- | --- | --- |
| **Date** | **Version** | **Description** | **Author** |
| 24/07/2021 | 1.0 | First RAD document | 1. Asnakechseleshi 2. FikreBerhanu 3. DegalemZeleke 4. Fatuma Mohammed 5. Marta Abebe |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

Table of Contents

1. Introduction 3

1.1 Purpose 3

1.2 Scope 3

1.3 Definitions, Acronyms and Abbreviations 4

1.4 References 5

1.5 Overview 5

2. Current system 6

3. Proposed system 6

3.1 Overview 6

3.2 Functional requirements 6

3.3 Nonfunctional requirements 9

3.3.1 User interface and human factors 10

3.3.2 Documentation 10

3.3.3 Hardware considerations 10

3.3.4 Performance characteristics 11

3.3.5 Error handling and extreme conditions 11

3.3.6 System interfacing 12

3.3.7 Quality issues 12

3.3.8 System modifications 12

3.3.9 Physical environment 12

3.3.10 Security issues 13

3.3.11 Resources and management issues 13

3.3.12 Reliability…………………………………………………………………………………………….12

3.4 Constraints (“Pseudo requirements”) 13

3.5 System models 14

3.5.1 Scenarios 14

3.5.2 Use case model 20

3.5.3 Object model 27

3.5.4 Data dictionary 27

3.5.5 Class diagrams 31

3.5.6 Dynamic models 33

4. Glossary 37

5. Supporting Information 38

# Introduction

In this section, we provide an overview of what the entire document will contain. This Requirement Analysis Document (RAD) is the documented result of the requirements elicitation activity. It aims to describe the system in terms of functional and non-functional requirement and serves as a contractual basis between the client and the developers. RAD on its subject’s wills practices different textual symbolic or diagrammatic representations. Based on the system model, shows that the major functional activities, states and object interactions within the design. This document briefly lists the functional and non-functional requirements of the system, security of the system, Hardware and software requirement of the system, the risk, cost of the system, how the user uses the system, what kind of training is required for the user, errors handling, Backup method and power stability.

## Purpose

The purpose of this RAD is to describe the external behavior of the application or subsystem identified. It also describes nonfunctional requirements, design constraints and other factors necessary to provide a complete and comprehensive description of the requirements for the software. This RAD document of software Architectural provides an overview of the proposed system, using a number of different architectural views to show different aspects of the system. It is intended to capture and convey the significant architectural decisions which have been made on the system.

## Scope

The proposed system Student Teaching Performance Evaluation and Information System(STPEIS) mainly intended to provide online functionalitieswhich support students and Academic staffs.

1. STPEIS creates a platform for students, a student dean, an instructor, department head, an academic dean andSystem administrator.
2. The users of the system are;

* Students: are responsible for using the system to fill evaluation form, send their complaint, application letter and view information and announcement.
* Student Dean: is responsible to view student complaint letter and post information for students.
* Instructor: is responsible to view announcement, information and then view his / her evaluation result.
* Department Head: is responsible to add all the necessary information on the system, review the evaluation rate and post an announcement and information.
* Academic dean: is responsible for reviewing evaluation result, posting announcement and other information.
* System admin: is responsible to manage the system and also register users and create username with default password, create student username with default password check error logs and report to developers.

1. The system uses a cross-platform tool, i.e., the application can run on a smart phone, tablet as well as a desktop.
2. Definitions, Acronyms and Abbreviations

|  |  |
| --- | --- |
| **Students** | Aperson who evaluates the instructor’s effectiveness, sends complaints and application letter request. |
| **Student Dean** | A person who is responsible to review a complaint letter. |
| **Department Head** | A person who feeds all the necessary information on the system, reviews evaluation rate, posts information and other announcements. |
| **Academic Dean** | A person who reviews evaluation rate, post announcement and information. |
| **System Administrator** | A person who is responsible for managing the entire system. |
| **Users** | Student, Student Dean, Department Head, Academic Dean and Instructors. |
| **IT** | Information Technology |
| **Android** | Is mobile operating system. |
| **CSS** | Cascading Style Sheet is used to format the layout of web page. |
| **Database** | Is an organized collection of data |
| **Javascript** | Is a programming language that is used to create application for android |
| **MySQL** | Is a relational database system |
| **PHP** | Is a scripting language and used to develop dynamic web application together with MySQL, HTML, CSS and Java Script. |
| **WAMP** | Server |
| **STPEIS** | Student Teaching Performance Evaluation and Information System. |
| **FR** | Functional Requirements |
| **NFR** | Non- Functional Requirements |
| **RAD** | Requirement Analysis Document |
| **GUI** | Graphical User Interface |

## References

* UML Distilled: A brief guide to the standard object modeling language, Martin Flower- Forewords by CrisKobryn, Grady Booch,Ivar Jacobson, and Jim Rumbaugh, 3rdEdition.
* System Analysis and Design Methods: Professor Jeffery L Whitten & Professor LonnineD.Bently: Published by McGraw-Hill Education Limited, 7th Edition.
* Patterns and Java (Third Edition). Prentice Hall, 2009. ISBN 978-0136061250.
* Object Oriented Software Engineering Instructor Course Handout Unpublished.
* Senior Final Project, RAD template Unpublished.

## Overview

This RAD is organized to clearly define the proposed system Student Teaching Performance Evaluation and Information System (STPEIS) using object oriented software engineering components and techniques that briefly explain the proposed system definition, functional requirements, non-functional requirements, the interfaces of the software, what the software will do, the constraints under which it must operate and how the software will react to external stimuli. This document is intended for both the end users and the developers of the software.

# Current system

Microlink Information Technology College currently has no automated system to monitor and follow the performance of their instructor and to display announcement and information for students.

# Proposed system

## Overview

This proposed system is to create Android based student teaching performance evaluation and information system that is used to enhance the evaluation process of the college between students and Instructors under the custody of the college. Android mobile evaluation for students and Instructors with android support presents a new aspect of developing the study including on the different comprehensive performance evaluation criteria of evaluating teachers and students.

In this subsection of the RAD, we will describe the proposedsystem Functional Requirements, Non-Functional Requirements and Constraints

## Functional requirements

In this sub-section we will explain about the functional requirements of the newly proposed system that define, the fundamental actions and description of activities that the system must perform and services a system must provide and accomplish. In other words, it describes the service provided for the users. The functional requirements categorized as the following:

### **Students**

* The student can access the Student Teaching Performance Evaluation and Information System.
* The student inserts his/her username and password at log in area in the login Page.
* If the student inserts a valid username and password, he or she will see the main menu that contains the following items:
* Access His/her Profile.
* Personal information: personal number, name, email and password.
* Course information: course name, Credit Hours, and Semester.
* Instructor information: Instructors name, Course name, e-mail address.
* Evaluation, Application and Compliant form.

**Functional requirements:**

* Student can evaluate his/her instructors.
* Student can give compliant to the particular instructor or teaching environment.
* Student can see the information and announcement
* Student can apply to the department depend on application form.

### **Student Dean**

* The Student Dean can access the Student Teaching Performance Evaluation and Information System.
* The Student Dean inserts his/her username and password at login area in the login Page.
* After login user can manage all the operations on;
  + - * Access His/her Profile.
    - Basic Student information and Department Information.

### **Functional requirements:**

* Student Dean can view any compliant given by their students.
* Student Dean can post information or announcement for the students.

### **Department Heads**

* The Department Heads can access the Student Teaching Performance Evaluation and Information System.
* The Department Heads insert his/her username and password at login area in the login Page.
* After login user can manage all the operations on;
  + - Access his/her profile.
    - Manage Student information
    - Manage Instructor information
    - Manage Course detail
    - Manage Evaluation

### **Functional requirements:**

* Department Heads can view their evaluation result based on instructor.
* Department Heads can review evaluation result and forward for instructor.
* Department Head scan view any complain given by their students.
* Department Heads can view application letter given by their students.
* Department Heads can post any information or announcements for the students and instructors.
* Department heads can add the list of questions to decide the performance of instructor.

### **Academic dean**

* The Department Heads can access the Student Teaching Performance Evaluation and Information System.
* The Academic dean inserts his/her username and password at login area in the login Page.
* After login user can manage all the operations on;
  + - Access his/her Profile.
    - Basic Student information.
    - Basic Instructor information.
    - Manage Department Heads.
    - Manage Departments.

### **Functional requirements:**

* Academic dean can view the evaluation result based on instructor.
* Academic dean can view any complain given by their students.
* Academic dean can post announcements and information.

### **3.2.5** **System Administration**

* The Admin Can Access the Student Teaching Performance Evaluation and Information System.
* Admin login to the system with valid username and password
* After login user can manage all the operations on;
  + Accesses His/her profile.
  + Basic Student information.
  + Basic Instructor information.
  + Academic dean information.
  + Student deaninformation.
  + Department heads information.

**Functional requirements:**

* Admin can add or delete academic dean, department head, instructor, student dean and students.
* Create users account.

### **3.2.6 Instructor**

* The instructor can access the Student Teaching Performance Evaluation and Information System.
* Instructor login to the system with valid username and password
* After login user can manage all the operations on;
  + Accesses His/her profile.
  + Basic Course information.

**Functional requirements:**

* Access evaluation result from department head.
* Access information from academic dean and department.

## Nonfunctional Requirements

In this sub-section we explain about the non-functional requirements of the newly proposed system of STPEIS, and it will provide a description of features like, characteristics, and constraints that define a satisfactory system.

### **User interface and human factors**

The system will have six types of users that interact with the system. These users are students, Student Dean, Instructor, Department Head, Academic Dean and System Admin. The GUI of the system will be designed to have a good impression and look to be user-friendly, so that the users can easily use it and learn. Error handling is done using massages which are popped up to the user.

### **Documentation**

In the student teaching performance evaluation and information system there will be four kinds of documentation.

1. **Functional Description:**  describes the general overview of the system.
2. **Install Manual:** describe how to install the system and configure the program.
3. **Introductory Manual:** describe how to get started with the program.
4. **Reference Manual:** describe all the program functions available for the user.

### **Hardware considerations**

The Student Teaching Performance Evaluation and Information System will be executed and run in a computer and a smart phone with the following hardware considerations.

The hardware considerations listed in the below table were considered the system to provide the best of its performance.

Table -Project Hardware Consideration

|  |  |  |  |
| --- | --- | --- | --- |
| *No.* | *Hardware Type* | *Minimum Requirement*  *Specification* | *Recommended Requirement*  *Specification* |
| *1* | Desktop Computer | ***Processor:*** *Dual Core, 2.66GHz*  ***RAM:*** *2GB, DDR3*  ***HDD:*** *500 GB, 12,000rpm, WD or Seagate* | ***Processor:****5 Core, 2.80GHz or above speed*  ***RAM:*** *8GB, DDR3 or above capacity*  ***HDD:*** *1TB, 12,000rpm, WD or Seagate* |
| *2* | Server Computer | ***Processor:****5 Core, 2.10GHz, 20MB Cache Memory per processor*  ***RAM:*** *8GB, DDR4 SDRAM,1866MHz*  ***HDD:*** *1TB, SATA 6GB/s, 12,000rpm, SCSI* | ***Processor:*** *8 Core, 3.20GHz and above speed with 20MB Cache Memory per processor*  ***RAM:*** *12GB and expandable to 16GB, DDR4 SDRAM, 1866MHz or above capacity*  ***HDD:*** *2TB and expandable, SATA 6GB/s, 12,000rpm, SCSI* |
| *3* | Smart phone | *Android OS based smartPhone having version API 21 (Lollipop)*  *2 Gigabytes of RAM*  *Storage 16GB* | *Android OS based smartPhone having version API 27 (Oreo)*  *4 Gigabytes of RAM*  *Storage 32 Gb* |
| *4* | Printer | *HP LaserJet 2010* | *HP LaserJet 2010* |
| *5* | Switch | 2 CISCO Small Business 200 Series, 10/100/1000 Base-TX | 5 CISCO Small Business 200 Series, 10/100/1000 Base-TX |
| *6* | Network Cable | *CAT-5* | *CAT-6* |
| *7* | Internet Connection | *1mbps bandwidth* | *5mbps bandwidth* |
| *8* | UPS | *As required* | *As required* |

### **Performance characteristics**

The STPEIS will allow users to access the system every 24/7 days of the year.

* The system should respond to a user’s request for information in less than 0.1 sec during peak time and 0.01 sec during normal time.
* The system supports more than 500 users per hour.

### **Error handling and extreme conditions**

* System handles input errors and exceptions by pop-up message to notifying the users.
* System Administrator should continuously check flaws and maintain it.
* Hardware failure switch, Router, network cable, printer, Storage, Database problem, Network problems informing the user about the problem.
* Power failures handled by Stabilizers and UPS.

### **System interfacing**

* There will be a GUI that will be used to interact with the desktop application and a smart phone.
* The system will send the evaluation result to the Instructors via e-mail and also show to the system Administrator, Academic Dean and Department Heads.
* The system interface has user-friendly features.

### **Quality issues**

* The main things for the system to be reliable to run it have backup server and storage.
* The system will trap all possible faults runtime and user invalid input error with its error handling method pops-up a message box to the user what is happened, why is happened and what should be done and take care of about the fault.
* The downtime when the system is unavailable, the time that a system fails the system will restart in less than 20 minutes all services must be activated.
* The acceptable system downtime per 24-hour period will be the total of 45 minutes
* The system will be portable in all android based devices.

### **System modifications**

As the number of users increases from time to time, technology changes and improvements will have an effect on the different hardware’s and software’s of the system, some features might require to be changed. So that the system will face performance problems, and the user also needs a user-friendly interface for the system to used, and then a change has to be made on some of the hardware, forms, user interface and additional features.

### **Physical environment**

The target equipment operates in a personal computer and server. On the first deployment of the system is in the organization single place and the system accessed from different sides of its user. The server will be located in a safe, appropriate room in which it may not be affected by dusts, unconditional room temperature, any person unless it is required maintenance.

### **Security issues**

* Access to the system will be protected by user login screen that requires a user name and password.
* Unverified users can’t access the system and view evaluation result.
* The system’s back-end server shall only be accessible to authenticated person that is the system admin.
* The system shall automatically log out all customers after a period of inactivity.
* The system will be secure and maintain physically.
* The back-end database shall be encrypted.

### **Resources and management issues**

* The system admin should ensure adequate back-up as may be required by their operation.
* The system administration is responsible for the installation of the system and maintenance.

### **Reliability**

* The software will not be able to connect to the centralized database in the event that the internet network fails or in the event of the server being down due to a hardware or software failure.
* Components of the project code shall be tested alongside the implementation phase to ensure that they are functional.
* Final, integrated project Code shall be tested with any testing tool to ensure that greater than or equal to 80% of the integrated code is covered at run-time, and is functioning properly. The remaining 20% will be inspected through manual testing to ensure the highest chance of being quality code.

## Constraints (“Pseudo requirements”)

The pseudo requirement of the proposed system of STPEI is defined and listed below:

* The front-end system should be implemented using only native JavaScript for android.
* The backend system should be implemented using only JavaScript also.
* The database should be implemented using MySQL database and also web application must be use MySQL, and PHP dynamic page for server side scripting language.
* The web application should use JavaScript scripting language for client side validation.
* The web application should use JavaScript scripting language for client side validation.

## System models

### **Scenarios**

**Table 1– Login/ Logout Scenario**

|  |  |
| --- | --- |
| Scenario Name | User Log In/ Logout |
| Actors | System admin, Department Heads, Academic Dean, Students, Student Dean and Instructors |
| Flow of events | The user initiates the application.   1. The system redirects the user to the log in page. 2. The users inter the name and password. 3. The system checks the log in user and redirect to the home page. |
|  | 1. The system will Logout after use |

**Table 2 – Data Feed**

|  |  |
| --- | --- |
| Scenario Name | Data Feed |
| Actors | Department Heads |
| Flow of events | The user initiates the application.   1. The system redirects the user to the log in page. 2. The users inter the name and password. 3. The system checks the log in user and redirect to the home page. 4. The user feed Semester course and Instructors. 5. The system accepts the data. |

**Table 3 – Update Semester Detail**

|  |  |
| --- | --- |
| Scenario Name | Update Semester Detail |
| Actors | Department Heads |
| Flow of events | The user initiates the application.   1. The system redirects the user to the log in page. 2. The users inter the name and password. 3. The system checks the log in user and redirect to the home page. 4. The user updates semester courses and Instructors. 5. The system accepts the data. |

**Table 4 – View Data**

|  |  |
| --- | --- |
| Scenario Name | View Data |
| Actors | Academic dean, Department, students, System admin |
| Flow of events | The user initiates the application.   1. The system redirects the user to the log in page. 2. The users inter the name and password. 3. The system checks the log in user and redirect to the home page. 4. The user can view permissible data. 5. The system will Logout after use. |

**Table 5- View Evaluation Form**

|  |  |
| --- | --- |
| Scenario Name | View evaluation form |
| Actors | System admin, Department Heads, Students |
| Flow of events | The user initiates the application.   1. The system redirects the user to the log in page. 2. The users inter the name and password. 3. The system checks the log in user and redirect to the home page. 4. The student evaluates the instructor. 5. The system will Logout after use. |

**Table 6 -Generate Evaluation Result**

|  |  |
| --- | --- |
| Scenario Name | Generate evaluation result |
| Actors | System admin |
| Flow of events | The user initiates the application.   1. The system redirects the user to the log in page. 2. The users inter the name and password. 3. The system checks the log in user and redirect to the home page. 4. System admin generate the evaluation result from the system. 5. The system will Logout after use. |

**Table 7–View Evaluation result**

|  |  |
| --- | --- |
| Scenario Name | View evaluation result |
| Actors | System admin, Academic Dean, Department Head, Instructor |
| Flow of events | The user initiates the application.   1. The system redirects the user to the log in page. 2. The users inter the name and password. 3. The system checks the log in user and redirect to the home page. 4. Users view the evaluation result from the system. 5. The system will Logout after use. |

**Table 8 – Send Compliant and Application Letter**

|  |  |
| --- | --- |
| Scenario Name | View Compliant and Application Form |
| Actors | Students |
| Flow of events | The user initiates the application.   1. The system redirects the user to the log in page. 2. The users inter the name and password. 3. The system checks the log in user and redirect to the home page. 4. The users send his/her complaints and applications. 5. The system will Logout after use. |

**Table 9 – View Compliant and Application**

|  |  |
| --- | --- |
| Scenario Name | View Compliant and Application letter |
| Actors | Academic Dean, Department Head, Student Dean |
| Flow of events | The user initiates the application.   1. The system redirects the user to the log in page. 2. The users inter the name and password. 3. The system checks the log in user and redirect to the home page. 4. The users view complaints and applications letter. 5. The system will Logout after use. |

**Table 10 – Add Announcement and Information**

|  |  |
| --- | --- |
| Scenario Name | Add Announcements and Information |
| Actors | Academic dean, Department Heads |
| Flow of events | The user initiates the application.   1. The system redirects the user to the log in page. 2. The users inter the name and password. 3. The system checks the log in user and redirect to the home page. 4. Users give any announcement and information touser. 5. The system will Logout after use. |

**Table 11 – View Announcements and Information**

|  |  |
| --- | --- |
| Scenario Name | View Announcements and Information |
| Actors | Academic dean, Department Heads, Students , Instructor |
| Flow of events | The user initiates the application.  1. The system redirects the user to the log in page.  2. The users inter the name and password.  3. The system checks the log in user and redirect to the home page.  4. Users view the Announcements and Information.  5. The system will Logout after use. |

### **Use case model**



1. ***Use case Description for Login***

|  |  |
| --- | --- |
| Name | User Log In |
| Actors | Students, Student Dean, Instructor, Department Heads, Academic Dean, System admin. |
| Entry Conditions | The Log in Page should be Loaded. |
| Flow of events | 1. The user initiates the application. 2. The system redirects the user to the log in page. 3. The users inter the name and password. 4. The system checks the log in user and redirect to the home page. |
| Exit Condition | User Name and Password are validated |

1. ***Use case Description for Change password***

|  |  |
| --- | --- |
| Name | Change password |
| Actors | Students, Student Dean, Instructor, Department Heads, Academic Dean, System admin. |
| Entry Conditions | The user profile Page should be Loaded. |
| Flow of events | 1. The user initiates the application.  2. The system redirects the user to the user profile page.  3. The user select change password.  4. The user change old password by the new.  5. The system prompts the home page. |
| Exit Condition | The system can the new Password validated |

1. ***Use case Description for Student Registration***

|  |  |
| --- | --- |
| Name | Student Registration |
| Actors | System Admin. |
| Entry Conditions | The User Registration Page should be Loaded. |
| Flow of events | 1. The user initiates the application. 2. The system redirects the user to registration form page. 3. The user fills the registration form and submits. 4. The system prompts the home page. |
| Exit Condition | User was register with detail information. |

1. ***Use case Description for Update Student profile***

|  |  |
| --- | --- |
| Name | Update Student profile |
| Actors | System Admin. |
| Entry Conditions | The Student Registration Page should be Loaded. |
| Flow of events | 1. The user initiates the application.  2. The system redirects the user to student registration page.  3. The Department heads update student profile and submit.  4. The system prompts the home page. |
| Exit Condition | User updates the detail information. |

1. ***Use case Description for Manage Evaluation***

|  |  |
| --- | --- |
| Name | Manage Evaluation |
| Actors | Department Heads. |
| Entry Conditions | The Evaluation Page should be Loaded. |
| Flow of events | 1. The user initiates the application.  2. The system redirects the user to evaluation page.  3. The department heads manage the evaluation.  4. The system prompts the home page. |
| Exit Condition | User was added and finalized evaluation form. |

1. ***Use case Description for Fill Evaluation Form***

|  |  |
| --- | --- |
| Name | Fill evaluation Form |
| Actors | Students. |
| Entry Conditions | The Home page should be Loaded. |
| Flow of events | 1. The user initiates the application. 2. The system redirects the user to registration form page. 3. The users fill the evaluation form and submit. 4. The system prompts the home page. |
| Exit Condition | User filled evaluation form and result submitted . |

1. ***Use case Description for View Evaluation Result.***

|  |  |
| --- | --- |
| Name | View Evaluation Result |
| Actors | System Admin, Instructor, Department Head, Academic Dean. |
| Entry Conditions | The Evaluation Result page should be Loaded. |
| Flow of events | 1. The user initiates the application. 2. The system redirects the user to view evaluation result page. 3. The users view the evaluation result. 4. The system prompts the home page. |
| Exit Condition | Users see evaluation result. |

1. ***Send Compliant & Application Letter.***

|  |  |
| --- | --- |
| Name | Send Compliant & Application Letter |
| Actors | Students |
| Entry Conditions | The Send Compliant & Application Letter page should be Loaded. |
| Flow of events | 1. The user initiates the application. 2. The system redirects the user to Send Compliant & Application Letter page. 3. The users write their complaint &application letter. 4. The system prompts the home page. |
| Exit Condition | User send their complaint &application letter |

1. ***View Compliant & Application Letter***.

|  |  |
| --- | --- |
| Name | View Compliant & Application Letter |
| Actors | Student Dean, Department Heads, Academic Dean. |
| Entry Conditions | View Compliant & Application Letter page should be Loaded. |
| Flow of events | 1. The user initiates the application. 2. The system redirects the user to View Compliant & Application Letter page. 3. The users view the complaint & application letter. 4. The system prompts the home page. |
| Exit Condition | Users view complaint & application letter |

1. ***Add Announcement and Information.***

|  |  |
| --- | --- |
| Name | Add Announcement and Information. |
| Actors | Department Heads, Academic Dean. |
| Entry Conditions | Add Announcement and Information page should be Loaded. |
| Flow of events | 1. The user initiates the application. 2. The system redirects the user to Add Announcement and Information page. 3. The users Add Announcement and Information. 4. The system prompts the home page. |
| Exit Condition | The system display announcement & Information on the top of the page |

1. ***View Announcement and Information.***

|  |  |
| --- | --- |
| Name | View Add Announcement and Information. |
| Actors | Students ,Department Head, Academic Dean , Instructor |
| Entry Conditions | View Announcement and Information page should be Loaded. |
| Flow of events | 1. The user initiates the application. 2. The system shows there is an announcement & Information on the top of the system. 3. The user gets Announcement and Information. 4. The system prompts the home page. |
| Exit Condition | User view announcement & Information on the top of the page |

1. ***Manage users and Full Application.***

|  |  |
| --- | --- |
| Name | Manage users and Full Application. |
| Actors | System Admin |
| Entry Conditions | System page should be Loaded. |
| Flow of events | 1. The user initiates the application.  2. The system admin manage users.  3. The system admin manage the full application.  4. The system prompts the home page. |
| Exit Condition | User manages and save all changes and exit. |

### **Object model**

The object model is a part of the analysis model and focuses on the individual concepts that are manipulated by the system, their properties and their relationships.

|  |  |  |
| --- | --- | --- |
| **Entity Object** | **Boundary Object** | **Control Object** |
| 1. Student | 1. Login Form | 1. Login In Authentication |
| 2. Student Dean | 2.Registration Form | 2.Logout Action |
| 3. Department Heads | 3. Evaluation Form | 3. Display Action |
| 4. Academic Dean | 4. Compliant Form | 4. Add User Action |
| 5. System Admin | 5. Announcement Form | 5. Create Account Action |
| 1. Instructor | 1. Information Form | 1. View Information and announcement |

### **Data dictionary**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Student Registration** | | | | |
| Attributes | Data Type | Index | Null Value | Description |
| id | Int(11) |  | Not |  |
| studentId | Varchar(100) | Primary key | Not |  |
| studentName | VarChar(100) |  | Not |  |
| phoneNo | VarChar(100) |  | Not |  |
| Section | VarChar(100) |  | Not |  |
| username | VarChar(100) |  | Not |  |
| password | VarChar(255) |  | Not |  |
| status | VarChar(100) |  | Not |  |
| deprtId | VarChar(100) | Foreign key | Not |  |
| courseofrId | Int(11) | Foreign key | Not |  |
| register\_date | Timestamp |  | Not |  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Course** | | | | |
| Attributes | Data Type | Index | Null Value | Description |
| courseId | Int | Primary key | Not |  |
| courseCode | VarChar(50) |  | Not |  |
| courseName | VarChar(50) |  | Not |  |
| creditHour | Int(10) |  | Not |  |
| Register\_date | timestamp |  | Not |  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Department** | | | | |
| Attributes | Data Type | Index | Null Value | Description |
| Id | Int(11) |  |  |  |
| departmentName | VarChar(50) | Primary key | Not |  |
| departmentDiscription | VarChar(50) |  | Not |  |
| Phone | VarChar(50) |  | Not |  |
| Code | VarChar(50) |  | Not |  |
| DurationYear | VarChar(50) |  | Not |  |
| Status | Int(11) |  | Not |  |
| Created\_date | datetime |  | Not |  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Section** | | | | |
| Attributes | Data Type | Index | Null Value | Description |
| sectioned | VarChar(50) | Primary key | Not |  |
| sectionName | VarChar(50) |  | Not |  |
| departmentId | int(11) | Foreign key | Not |  |
| sectionProgram | VarChar(50) |  | Not |  |
| Register\_date | timestamp |  | Not |  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Course Offering** | | | | |
| Attributes | Data Type | Index | Null Value | Description |
| courseOfferingId | Int(30) | Primary key | Not |  |
| coursed | Varchar(50) | Foreign key | Not |  |
| sectioned | Varchar(50) | Foreign key | Not |  |
| instructorId | Int(11) | Foreign key | Not |  |
| Year | Int(10) |  | Not |  |
| Semester | int(11) |  | Not |  |
| register\_date | timestamp |  | Not |  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Evaluation Questions** | | | | |
| Attributes | Data Type | Index | Null Value | Description |
| Questioned | int(11) | Primary key | Not |  |
| quationContent | text |  | Not |  |
| register\_date | timestamp |  | Not |  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Evaluation Result** | | | | |
| Attributes | Data Type | Index | Null Value | Description |
| evaluationId | Int(11) | Primary key | Not |  |
| studentId | int(50) | Foreign key | Not |  |
| instructorId | Int(50) | Foreign key | Not |  |
| coursedId | Int(50) | Foreign key | Not |  |
| evaluationResult | float |  | Not |  |
| evaluationDate | date |  | Not |  |
| register\_date | timestamp |  | Not |  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Announcement &Information** | | | | |
| Attributes | Data Type | Index | Null Value | Description |
| posted | Int(11) | Primary key | Not | Its primary key |
| postRef\_no | Varchar(50) |  | Not |  |
| postedBy | Varchar(50) |  | Not | Describe who post information |
| postedFor | Varchar(50) |  | Not | Deprt, DFean, Steudent Dean, |
| postSubject | Varchar(100) |  | Not |  |
| postContent | text |  | Not |  |
| postExpireDate | Date |  | Not |  |
| Register\_date | timestamp |  | Not |  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Application & Complaint** | | | | |
| Attributes | Data Type | Index | Null Value | Description |
| appId | Int(11) | Primary key | Not | Its primary key |
| appType | Varchar(50) |  | Not | Application/Compliant |
| appContent | text |  | Not | Deptment, Student Dean |
| appFor | Varchar(100) |  | Not | Deptment, Student Dean |
| appDate | Date |  | Not |  |
| studentId | int(11) | Foreign key | Not |  |
| applcFile | text |  | Not |  |
| status | int(11) |  | Not |  |
| Register\_date | timestamp |  | Not |  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Admin and Staff Registration** | | | | |
| Attributes | Data Type | Index | Null Value | Description |
| stafId | Int(11) | Primary key | Not |  |
| userNaame | Varchar(50) |  | Not |  |
| password | VarChar(255) |  | Not |  |
| isadmin | int(3) |  | Not |  |
| fullName | Varchar(50) |  | Not |  |
| phone | Varchar(50) |  | Not |  |
| city | Varchar(50) |  | Not |  |
| workingZone | Varchar(50) |  | Not |  |
| e\_mail | Varchar(50) |  | Not |  |
| status | Int(11) |  | Not |  |
| deprtId | Int(11) | Foreign key | Not |  |
| Created\_date | date |  | Not |  |

### **Class diagrams**

classes, associations, attributes and operations

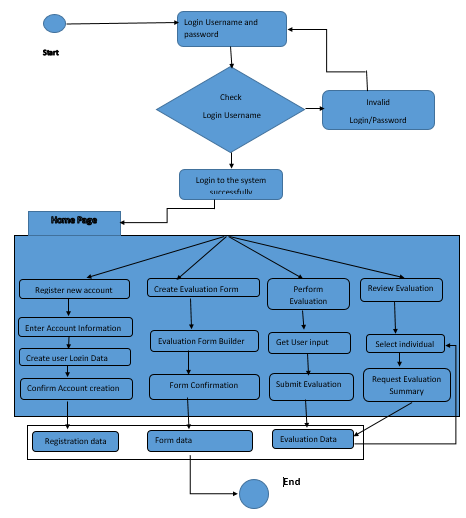


### **Dynamic models**

### **3.5.6.1 State Diagram**

State diagrams for classes with significant dynamic behavior

Sequence diagrams and collaboration diagrams for interacting objects (protocol)



#### **3.5.6.2 Sequence diagram**

* **Sequence diagram for user login**



* **Sequence Diagram For Student Evaluation System**



* **Sequence diagram of Application and Compliant system**



# Glossary

|  |  |
| --- | --- |
| ***System*** | *A set of inter-connected computer equipment and/or programs used together for a particular purpose and operating together.* |
| *System Admin* | *A person who is responsible for software installation, management, configuration, information and maintenance of a computer, network and the system, and system user’s access controls* |
| *Action* | *A fundamental unit of behavior. Actions can take a set of inputs, produce a set of outputs, and change the state of the system. Actions are fundamental in the sense that they are not decomposed further.* |
| *Actor* | *An entity that needs to interact with the system for exchange information. An external entity that needs to exchange information with the system. An actor can represent either a user role or another system* |
| *Scenario* | *A short text description on the system function.* |
| ***Use case*** | *An action implemented by the computer system.* |
| ***Diagram*** | *A simple plan which represents a machine, system or idea, etc., often drawn to explain how it works.* |
| ***Class diagram*** | *Is a type of static structure diagram that describes the structure of a system by showing the system's classes, their attributes, operations, and the relationships among objects* |
| ***Sequence Diagram*** | *A diagram which represents sequence of action that human or system represents.* |
| ***State diagram*** | *Is a type of diagram used to describe the behavior of system* |
| ***Dynamic Model*** | *The documents that; the behavior of the object model in terms of state machine diagrams and sequence diagrams. It describes the components of the system that have interesting behavior. Within Dynamic Model it contains: state machine diagrams, sequence diagrams, and activity diagrams.* |
| ***Server*** | *A computer system that provides services to other computing systems over a network.* |

# 

# Supporting Information

* Object Oriented Software Engineering -Using UML, Patterns, and Java; Bernd Bruegge& Allen H. Dutoit; Published by Pearson Education Inc., 3rd Edition.
* Object Oriented Software Engineering –Practical Software Development Using UML and Java; Timothy C. Lethbridge& Robert Laganiere; Published by McGraw-Hill Education Limited. 2nd Edition.
* UML Distilled -A Brief Guide to The Standard Object Modeling Language; Martin Fowler -Forewords by CrisKobryn, Grady Booch, Ivar Jacobson, and Jim Rumbaugh, 3rd Edition.
* System Analysis and Design Methods; Professor Jeffrey L. Whitten & Professor Lonnie D. Bentley; Published by McGraw-Hill Education Limited. 7th Edition.
* System Analysis and Design; Alan Dennis & Barbara Haley Wixom &Roberta M. Roth; Published by John Wiley & Sons Inc., 5th Edition.
* Object Oriented Software Engineering; Instructors Course Handout, Unpublished.