

(CS622) UML LAB

Objective of the course :

Main objective of this lab is to enable the student to practice the object-oriented analysis and design through UML on a particular application (project) so that he will apply same methodology in mini project which has to be done in final year. And also it will give exposure to tools that support UML and Object oriented software development.

1. The student should take up the case study of Unified Library application which is mentioned in the theory, and Model it in different views i.e Use case view, logical view, component view, Deployment view, Database design, forward and Reverse Engineering, and Generation of documentation of the project.
2. Student has to take up another case study of his/her own interest and do the same what ever mentioned in first problem. Some of the ideas regarding case studies are given in reference books which were mentioned in theory syllabus can be referred for some idea.

NOTE:

The analysis, design, coding, documentation, database design of mini project which will be carried out in 4th year should be done in object-oriented approach using UML and by using appropriate software which supports UML, otherwise the mini project will not be evaluated.

1. Prepare the following documents for two or three of the experiments listed below and develop the software engineering methodology.
 1. Program Analysis and Project Planning.
Thorough study of the problem – Identify project scope – Objectives – Infrastructure.
 2. Software requirement Analysis
Describe the individual Phases / Modules of the project – Identify deliverables.
 3. Data Modeling
Use work products – Data dictionary – Use diagrams and activity diagrams, build and test lass diagrams – Sequence diagrams and add interface to class diagrams.
 4. Software Developments and Debugging
 5. Software Testing
Prepare test plan – perform validation testing – Coverage analysis – memory leaks – develop test case hierarchy – Site check and Site monitor.

Suggested List of Applications:

1. Student Marks Analyzing System
2. Quiz System
3. Online Ticket Reservation System
4. Payroll System
5. Course Registration System
6. Expert Systems
7. ATM Systems
8. Stock Maintenance
9. Real-Time Scheduler
10. Remote Procedure Call Implementation

Mini-Project - I: A Point-of-Sale (POS) System

A POS system is a computerized application used to record sales and handle payments; it is typically used in a retail store, it includes hardware components such as a computer and bar code scanner, and software to run the system. It interfaces to various service applications, such as a third-party tax calculator and inventory control. These systems must be relatively fault tolerant; that is, even if remote services are temporarily unavailable they must still be of capturing sales and handling at least cash payments. A POS system must support multiple and varied client-side terminals and interfaces such as browser, PDAs, touch-screens.

Mini-Project - II: Online Bookshop Example

Following the model of amazon.com or bn.com, design and implement an online bookstore.

Mini-Project - III: A Simulated Company

Simulate a small manufacturing company. The resulting application will enable the user to take out a loan, purchase a machine, and over a series of monthly production runs, follow the performance of their company.

Mini-Project - IV: A Multi-Threaded Airport Simulation

Simulate the operations in an airport. Your application should support multiple aircrafts using several runways and gates avoiding collisions/conflicts. Landing: an aircraft uses the runway, lands, and then taxis over to the terminal. Take-Off: an aircraft taxis to the runway and then takes off.

Mini-Project -V: An Automated Community Portal

Business in the 21st Century is above all BUSY. Distractions are everywhere. The current crop of “enterprise intranet portals” are often high noise and low value, despite the large capital expenditures it takes to stand them up. Email takes up 30 - 70% of an employee’s time. Chat and Instant Messaging are either in the enterprise or just around the corner. Meanwhile, management is tasked with unforeseen and unfunded leadership and change-agent roles as well as leadership development and succession management. What is needed is a simplified, repeatable process that enhances communications within an enterprise, while allowing management and peers to self-select future leaders and easily recognize high performance team members in a dynamic way.

Additionally, the system should function as a general-purpose content management, business intelligence and peer-review application.

Glasscode’s goal is to build that system. The software is released under a proprietary license, and will have the following features: Remote, unattended moderation of discussions. However, it will have powerful discovery and business intelligence features, and be infinitely extendable, owing to a powerful API and adherence to Java platform standards. Encourages peer review and indicates for management potential leaders, strong team players and reinforces enterprise and team goals seamlessly and with zero administration.

Mini-Project -VI: A Content Management System

The goal is to enable non-technical end users to easily publish, access, and share information over the web, while giving administrators and managers complete control over the presentation, style, security, and permissions.

Features:

- | Robust Permissions System
- | Templates for easy custom site designs
- | Total control over the content
- | Search engine friendly URL’s
- | Role based publishing system
- | Versioning control ? Visitor profiling

Mini-Project-VII: An Auction Application

Several commerce models exist and are the basis for a number of companies like eBay.com, priceline.com etc. Design and implement an auction application that provides auctioning services. It should clearly model the various auctioneers, the bidding process, auctioning etc.

Mini-Project -VIII: A Notes and File Management System

In the course of one’s student years and professional career one produces a lot of personal notes and documents. All these documents are usually kept on papers or individual files on the computer. Either way the bulk of the information is often erased corrupted and eventually lost. The goal of this project is to build a distributed software application that addresses this problem. The system will provide an interface to create, organize and manage personal notes through the Internet for multiple users. The system will also allow users to collaborate by assigning permissions for multiple users to view and edit notes.

Mini-Project - IX: A Customizable Program Editor

A programmer’s editor which will be focused on an individual programmer’s particular needs and style. The editor will act according to the specific language the current source file is in, and will perform numerous features, such as auto-completion or file summarization, on the file. These features will be able to be turned on or off by the programmer, and the programming style of the user will be used to create as efficient an editing environment as possible.

Mini-Project - X: A Graphics Editor

Design and implement a Java class collection that supports the construction of graph editing applications, i.e., applications that include the ability to draw structured and unstructured diagrams.

E.g.,

The goal of the GEF project is to build a graph editing library that can be used to construct many, high-quality graph editing applications. Some of GEF’s features are:

A simple, concrete design that makes the framework easy to understand and extend.

Node-Port-Edge graph model that is powerful enough for the vast majority of connected graph applications.

Model-View-Controller design based on the Swing Java UI library makes GEF able to act as a UI to existing data structures, and also minimizing learning time for developers familiar with Swing.

High-quality user interactions for moving, resizing, reshaping, etc. GEF also supports several novel interactions such as the broom alignment tool and selection-action-buttons. Generic properties sheet based on JavaBeans introspection. XML-based file formats based on the PGML standard.

TEXT BOOK :

1. Grady Booch, James Rumbaugh, Ivar Jacobson : The Unified Modeling Language User Guide, Pearson Education.