1. Overall Description

1.1 Product Perspective

The UNIVERSITY MANAGEMENT SYSTEM Allows Authorized Members to Access the records of Academic all syllabus and registered students, Counselling's and admission Procedure. It can be used in various Educational Institutes across the globe and simplifies working of institutes affiliated with this University.

2.2. Product Perspective

The proposed system shall be developed using client/server architecture and be compatible with Microsoft Windows Operating System. The front end of the system will be developed using Visual Basic 6.0 and backend will be developed using MS SQL Server 2000.

2.2.1. System Interfaces

None

2.2.2. User Interfaces

The OUMS will have following user-friendly and menu driven interfaces

a) **Login:** to allow the entry of only authorized users through valid login Id and password.

b) **College Details**: to maintain college details.

c) **Programmes Details**: to maintain programmes details.

d) **Streams Details**: to maintain scheme details of a programmes.

e) **Paper Details**: to maintain paper details of a scheme for a particular programmes

g) **Faculty Details**: to maintain the faculty details.

2.2.3. Hardware Interfaces

a) Screen resolution of at least 640 x 480 or above.

b) Support for printer (dot matrix, Deskjet, Inkjet, LaserJet)

c) Computer systems will be in the networked environment as it is a multi-user system.

2.2.4. Software Interfaces

a) MS-Windows Operating System

b) Microsoft Visual Basic 6.0 for designing front-end

c) MS SQL Server 2000 for backend

d) PLATEFORM: JAVA LANGUAGE

e) INTEGRATED DEVELOPMENT ENVIRONMENT(IDE): ECLIPSE

2.2.5. Communication Interfaces

None

2.2.6. Memory Constraints

At least 512 MB RAM and 500 MB space of hard disk will be required to run the software.

2.2.7. Operations

None

2.2.8. Site Adaptation Requirements

The terminal at client site will have to support the hardware and software interfaces specified in the section 2.1.3 and 2.1.4 respectively.

2.3. Product Functions

The OUMS will allow access only to authorized users with specific roles (System administrator, Faculty and Student). Depending upon the user's role, he/she will be able to access only specific modules of the system.

A summary of major functions that the URS will perform

- A login facility for enabling only authorized access to the system.

- System administrator will be able to add, modify or delete programmes, college, scheme, paper and login information.
- System administrator/Faculty will be able to generate reports.
- Students will be able to see notifications, Academic Details, Download their results.

2. Operating Environment

 This project will be operating in windows environment. Also compatible with internet explorer.

 The only requirement for using java on the back hand for database we are using Sql server. The product is accomplished with the login facility for user.

3.1 Design and Implementation Constraints

● Each college must keep their password as confidential. More over the user must have individual ID for creating a login.

● Only Administrator can control user addition and deletion in the system. Also this group has the access to all the official activities.

● The College must be logged in to place an the order.

3.2 Assumptions and Dependencies

● Each College must have a User ID and password.

● 80 Kbps Internet connection is a must.

● Internet connection is must.

● Proper browser should be installed in the user system.

4.1 Communication Interfaces

None

4.2 Functional Requirements

4.2.1 LOGIN

A. Use Case Description

1 *Introduction*

This use case documents the steps that must be followed in order to log into the University Registration System.

2 Actors

Administrator

Student

Faculty

Registrar

3 Pre-Condition

The user must have valid login Id and password.

4 Post Condition

If the use case is successful, the actor is logged into the system. if not, the system state remains unchanged.

4.3

4.3.1 Starts when actor wishes to login onto the UMS

i. The system requests that the actor specify the function he/she would like to perform (either Login, Change Password)

ii. Once the actor provides the requested information, one of the sub flows is executed.

- If the actor selects “Login”, the Login sub flow is executed.

- If the actor selects “Change Password”, the Change Password sub flow is executed.

1.b) A survey of various UML tools

Unified Modeling Language (UML) abstracts and visualizes systems of object-oriented programming. This makes the modeling language a practical tool for developers: On the one hand, it makes it possible to create clear blueprints for software projects; on the other hand, complex software systems can also be presented in an understandable way for people not familiar with the subject. For example, if you want to introduce new software for the latest company app to the Head of Marketing, do not use code – instead, you can use UML to show them the most important features of the app.

1. **Gliffy: an online UML tool for beginners**

The online application Gliffy is a cloud-based UML tool for the browser. First released in 2006, the modeling tool creates all types of diagrams such as flowcharts, Venn diagrams, and of course, UML diagrams. The online tool was written in HTML5 and scores points with its fast reaction time. Even before Gliffy went through the beta phase in 2007, the company (under the same name) cooperated with the Australian software group Atlassian. As early on as 2006, its collaboration software, Confluence, integrated a Gliffy plug-in. Later, the Gliffy team developed a plug-in for Jira. Google’s G Suite and Drive also contain the UML tool.

Whether as a plug-in or standalone browser version, the software is ideal for teamwork. You can share read-only versions as well as actively work on diagrams simultaneously and exchange information using the integrated comment function.

Before you can use Gliffy, sign in with your **e-mail address** or a Google or Facebook account. After the **free trial period** of 14 days, it can only continue to be used for free with limited functionality. However, you can upgrade to a premium account with a monthly subscription.

Gliffy has a large, well-assorted shape library. In addition to UML, you can use it to model simple flowcharts or business process modeling and notations (BPMN), among other things. At the beginning, an orientation window allows you to **define your desired modeling language**. If you click on UML/ERD, the program shows you the **templates** of all UML diagram types in the sidebar next to a blank page. Above it, you will find the menu bar with **tools**. If you drag a shape onto the worksheet using the mouse, the tools automatically adapt to edit the respective shape. If you select groups, the corresponding tools are highlighted in the menu bar. Grid and guidelines allow you to place diagram elements precisely where you want them.

The **theme tool** should determine the color family for the diagram but is limited to the arrow elements in the test. By clicking on classes, components, and the like, you can still quickly adjust the color and font of the symbols. You can also create different **layers** in no time by clicking on the corresponding menu item (far right in the tool bar). In addition, you do not have to plan the diagram construction step by step in advance. You can simply undo errors with the delete function. If you want to insert an element, Gliffy detects the change and suggests appropriate insert options.

As a free UML tool, Gliffy imports diagrams in GON, Gliffy, and gXML formats. For the Microsoft Visio format, VDX, you need a business account. The same applies when **exporting** common image formats such as JPG, PNG, and SVG and the connection to Google Drive. Free accounts also have the native "Gliffy" format. Save a document in this format, share it easily with others via a download link, or embed it on various platforms such as HipChat, Slack, Confluence, or WordPress.

A free account also provides **2 MB of cloud space** for your diagrams and allows you to **share five models publicly** at the same time. With a paid account from about $5 per month, Gliffy offers you even more possibilities: unlimited memory, diagram import from Visio, and export to the mentioned image formats. An enterprise account is required for significantly more performance. For such high demands, however, there is better software.

Advantages	**Disadvantages**
✓ Suitable for all current browsers	✗ Features such as image export and Google Drive interface can only be used in group business subscriptions onwards
✓ Not a download	✗ No debugger
✓ Integrated team communication	
✓ Fast operation thanks to HTML5, drag-and-drop, and clear user interface	
✓ UML 2 compatible	
✓ Comprehensive support	

2. ArgoUML: popular freeware for simple diagrams

ArgoUML has long been one of the most popular open source free UML tools for desktop. Although it is no longer maintained, many modelers continue to use the program for smaller tasks. The software is **platform-independent**. And the minimum requirement is Java 5. ArgoUML supports all diagram types of **UML version 1.4** and UML profiles. The program also offers some decorative shapes that are not part of the UML standard. If you use these shapes, you will of course deviate from the UML standard. So, make sure that this does not cause any comprehension problems. You can use OCL (Object Constraint Language) to assign restrictive information to a model.

Although the UML tool is available to download for free, ArgoUML supports a considerable range of programming languages whose **code can be generated from a diagram**. **Reverse engineering** is also possible for Java, C++, PHP, C#, and SQL. The program also recognizes other languages such as Delphi or Ruby when you add them to the ArgoUML file folder as extensions.

The UML program can be quickly installed via **click-and-go**. The user interface has certain 1990s flair. Nevertheless, the panel layout is **clearly** arranged and the menu above the digital worksheet provides the required shapes after you click on the desired diagram type.

The top left panel contains the explorer, which you can arrange according to your own or existing rules by adjusting the **perspective**. Below you will find the troubleshooter, known in Argo as the **edit panel**. Any possible incorrect modeling will appear in this field. If you click on one of the folders, the program marks faulty places or missing assignments in the diagram in red. You can export finished diagrams as code or graphics (supported image formats are PNG, PGML, SVG, EPS, GIF, PS) in a .zip file (".zargo"). Use XMI (XML metadata exchange) to transfer diagrams to other programs.

Advantages	Disadvantages
✓ Open source and adjustable	✗ Non-compliant with UML 2
✓ Freeware	✗ Limited diagram selection
✓ Debugger points out errors	✗ No undo button
✓ Round-trip engineering for C#, C++, Java, PHP, and SQL	✗ Last update 2011, support only through documentation
✓ Intuitive user interface	

3. MagicDraw: everything you need for professional UML diagrams

MagicDraw von No Magic is the first comprehensive full version for **professional modeling** in this list. A modern design and a clean layout set this desktop app apart from the rest as a proprietary software for high demands. The range of functions and user-friendliness confirm this. The modeling tool supports UML, its equivalent for operating systems, SysML, the graphical representation of business processes with BPMN (Business Process Model and Notation) and the UPDM architecture framework (United Profile for DoDAF/MODAF). In MagicDraw, you work with current diagrams according to the **UML 2.5 standard**, whose profiles you can tailor to your own needs. MagicDraw also offers **OCL**, the boundary condition notation, and **XMI**, which you can use to export diagrams to other programs without any loss.

MagicDraw offers five editions (from Personal to Enterprise) that differ in both functionality and cost. In our test, we took a closer look at the Enterprise edition because it not only offers all the features and plug-in for effective modeling, but also enables full integration into an **integrated development environment** such as Eclipse Workbench. The premium versions also facilitate **teamwork**. For example, several modelers can work on one model at the same time, and on the team server, you actively exchange information about a project. However, NoMagic offers this separately. With the Enterprise edition, you can access the **WebPortal**. The model can be viewed interactively in the browser.

The **minimum requirements** for installation are as follows:

- CPU: Intel Core TM i3
- Ram: 4 GB
- Storage space: 1 GB
- Screen resolution: 1366 x 768 pixels
- Operating system: all operating systems that are Java SE 8 capable (Windows from Vista onwards, macOS from Lion onwards)

The UML tool imports many file formats, including native XMI metadata formats from Eclipse and IBM's Rhapsody, CSV, ReqIF, DoDAF, and CA Erwin Data Modeler. You can save and print your diagrams as copies in the image formats BMP, PNG, JPG, and EMF. **Code generation**, **reverse engineering** and **round-trip engineering** are also possible. MagicDraw is based on three languages: Java, C++ in different dialects, and C#.

MagicDraw converts UML diagrams into code:

- Java
- C++
- C#
- XML schema
- Corba IDL

Reverse engineering of code into UML diagrams:

- DDL
- WSDL
- Java
- C++
- C#
- XML schema

If you model UML diagrams yourself in an empty document or add details to modeling, you'll quickly appreciate the **customizable panels**. Already at the first start of the UML diagram tool, the clear division proves to be a big plus. Add as many tabs as you consider necessary for your workflow. Despite its many functions and edition options, MagicDraw provides a **clear layout** and unobtrusive short explanations, so that even inexperienced users can quickly put together their first drafts without much familiarization time.

If you select the "diagrams" tab in the main menu bar, the UML tool opens the template for the selected type in the sidebar of the workspace. If you drag an element onto the worksheet and click it once, small **buttons appear**. Click on these (or right-click on the element) to edit the object properties, add text, or hyperlinks. The personalization toolbar above the worksheet always highlights the tools you can use to graphically customize one or more selected items. The layout tool brings order to unstructured drafts. With the zoom and perspective panel, you have your project completely in view – from small details to the whole image.

The UML program also checks your project for errors and displays them. For independent analysis, you have the option of comparing different versions of a diagram next to each other and checking different levels of abstraction for traceability. Architects particularly appreciate the free text input for creating diagrams, as it favors a smooth workflow.

Advantages	Disadvantages
✓ Contains all templates for diagram types in current UML 2.5	✗ File download only possible after completing registration
✓ Supports many programming languages and formats	✗ Comparatively high prices
✓ Intuitive and understandable user interface	✗ WebPortal only available in Enterprise edition
✓ Import/export, code generation, reverse engineering, and round-trip engineering	
✓ Short familiarization time	

4. Lucidchart: the online UML tool for teamwork

Lucidchart is a UML tool that you can access at any time in your browser. This is also possible via Android and iOS. The free account gives you a well-filled UML toolbox. It includes **7 types of UML diagrams** and business process modeling languages such as BPMN 2.0, network icon templates, mobile device mockups, and video integration. One of Lucidchart advantages is its **intuitive operation**. It also enables **team sharing and simultaneous editing of diagrams** and integrates comments directly in the tool. As a UML tool for Mac, Linux, and Windows, it is a good alternative to Microsoft Visio for Mac users.

If you already use another teamwork software, take a look at the list of **integrations**. Lucidchart can be plugged in to Google and Microsoft applications, Atlassian products, chat apps, and Amazon web services.

If you want to try out the UML tool for free, create a free no-obligation account, or use the equally free test phase of the premium accounts. With the latter, your **cloud diagram** storage grows from 25 MB to 100 MB, allowing you to export charts, integrate third-party vendors, and work with more professional forms. In the **free account**, you have the following UML diagram templates:

- Class diagram
- State diagram
- Activity diagram
- Sequence diagram
- Component diagram
- Use case diagram
- Distribution diagram

If you want to design complex diagrams with more than 60 objects, you need a premium account. The same applies if you are working on more than three active objects at the same time.

You can also import data and diagrams with the free account. The UML diagram tool supports native diagram formats from Microsoft Visio, Omnigraffle, Gliffy, and Draw.io. Drag data from CSV files, SQL, AWS architecture, and mind maps. The **data linking** feature also integrates real-time data from Excel, CSV, and Google Sheets. If you want to edit native formats of other providers, however, you will again need a paid account. It also allows you to backup and restore your data.

But how efficient is Lucidchart to work with? The user interface is characterized by a **clear layout** and plenty of space to work. The menu pops up from the right when you need it and disappears as a narrow **toolbar until** you see it. With one click, you can change the page settings, create layers, and presentations, or exchange information with teammates. You will find the templates in the bar to the left of the workspace. You can customize its content and size by adding **templates** and using the mouse to drag the width of the bar to how you want it.

The **horizontal toolbar** shows the tools you need each time you want to drag a shape or line onto the work area. You can easily adjust the size and orientation by clicking on the shape. If you hover over a shape with the mouse, small red circles will appear. You can add **arrows and lines** at these points. Use the grid and automated guides to easily draw neat diagrams. Some

developers prefer to write input commands instead, building their UML diagram. The tool offers the **UML markup** function, so you can do this.

The team edition allows you to customize user roles and protect your documents through single sign-on authentication. Lucidchart cloud storage also encrypts your data. Every time you share a document, you decide who can edit, comment, or just read it.

Advantages	Disadvantages
✓ Many teamwork features	✗ Many functions only available in Pro or Team accounts
✓ Large template library	✗ No debugger
✓ UML mark-ups accelerate the workflow	✗ Code cannot be generated from diagrams
✓ Scalable through cloud storage	
✓ Space-saving, clear design	
✓ Suitable for mobiles devices (iOS, Android)	

5. **IBM Rational Rhapsody: graphical developments environment for full process integration**

IBM Rational Rhapsody is a **graphical, integrated development environment** (IDE) for software and product development. The UML program supports object-based software development for web-based applications as well as embedded systems and real-time systems based **on C++ and Java EE**. **UML/SysML modeling** enables you to create source code quickly in the IDE for the specified languages, **C and C#, MISRA C++, and Ada**.

Rhapsody was passed on many times after its **development in 1998**. After IBM acquired the Swedish company, Telelogic AB in 2008, the American company expanded its rational product range with Telelogics Rhapsody. The Swedes only purchased the IDE in 2006 by acquiring the original developer company, I-Logix. As **one of the first graphical development environments**, Rational Rhapsody, spread in a short time.

International brands use the software, for example, for large projects that require detailed version differentiation for numerous team members in different countries. Jaguar Land Rover, for example, uses Rational Rhapsody to adapt the software to its infotainment offer to different markets.

For these and other tasks, Rhapsody offers a development environment in which you can draw diagrams, **generate and check code**, and **compare or analyze different versions**. **The UML tool supports the following**:

- UML
- SysML

- AUTOSAR
- DoDAF
- MODAF
- UPDM
- DSL

These UML-based modeling languages are suitable for **tool-based software development**, which saves programmers a lot of work. In addition, you can rely on compatibility, since UML and Rhapsody were developed in the same house – namely at Rational Software.

In Rhapsody, you either draw the models freely, **import existing code for visualizations as diagrams**, or download **requirements** from a requirements management tool such as Doors, a relative of the Rational family. The following YouTube video shows you how to create a SysML model for a real-time system from a list of requirements in Doors. In this case, IBM will present its approach to the INCOSE Vendor Challenge. The aim of the project was to create a parking system for a hotel chain.

The developers built the UML tool based on the open source software **Eclipse**. The current version still supports the Eclipse IDE platform as an integration or a plugin. It contains further integrated development environments as **workflow integration**:

- Wind River Workbench
- Green Hills MULTI
- Microsoft Visual Studio 2008

As an IDE, Rhapsody works in real time and creates source code frames from diagrams. The program is available in four different license forms. In addition, IBM offers different versions of the UML tool specifically for the different requirements of different departments. The modules are compatible with the IDE. You can choose between the Architect for Systems Engineers, the Designer for Systems Engineers, the Architect for Software, or the Developer. The programs differ mainly in functionality and price.

The Developer is the full version, which combines all features and integrates your created source code into the embedded development environment. It also optimizes the path from the first draft to the prototype and to the finished implementation. For example, you can use **round-trip engineering and reverse engineering** for this purpose. If you want to synchronize source code and design or document the process, this is also possible. Among the slim-lined versions, **Architect for Software** is the most powerful tool for working with UML, because it is the only one that offers these developer features.

Advantages	Disadvantages
✓ Modular design, scalable	✗ Longer familiarization time

✓ Supports all current UML 2 diagram types	✗ Expensive compared to other options
✓ Integrated development environments and interfaces for other development platforms and tools enable a secure, integrated workflow right from the start	
✓ Code frame generation for many programming languages	

6. Microsoft Visio: the UML drawing tool for Office users

Microsoft Visio is a popular **chart and visualization software** and belongs to the Office family. Therefore, Visio can be easily integrated into the suite. For example, if you use Office Pro 365, Microsoft offers you a subscription extension for Visio. However, the cost will be added to your Office subscription. **Visio Online Plan 2** includes a desktop app and a web-based editor for up to five PCs (volume licensing on request).

Alternatively, you can get **Visio Professional** as a permanent license. This version includes a desktop app, the browser applications, and an iPad app. **Visio Standard does not support UML diagrams** and is therefore not included here. As UML tools, the Visio versions presented are primarily aimed at corporate customers who appreciate the familiar Office environment and want to optimize their workflow by seamlessly integrating the appropriate modeling tool into their existing system. Although both versions – Profession and Online Plan 2 – are similar in functionality, they do not support UML equally:

UML diagram type	Microsoft Visio Professional 2016	Microsoft Visio Online Plan 2 (also: Visio Pro für Office 365)
Class diagram UML 2.0	✓	✓
Database notation UML 2.0	✓	✓
Sequence diagram UML 2.0	✓	✓
State machine UML 2.0	✓	✓
Use case diagram UML 2.0	✓	✓

Activity diagram UML 2.0	✓	✓
Component diagram UML 2.5	✗	✓
Communication diagram UML 2.5	✗	✓
Distribution diagram UML 2.5	✗	✓

If you need a **comprehensive shape library** with current UML notations standards (the OMG UML standard is UML 2.5.1 at present) and if you have already subscribed to Microsoft Office Pro 365, we recommend **Online Plan 2**, the minimum requirement for current Visio versions is Windows 7. The desktop app requires 1GB of memory and 3GB of disk space. By the way: If you don't need more than a compatible program to view diagrams, the free **Microsoft Visio 2016 Viewer**, a browser plugin for Internet Explorer may be enough. Due to its limited compatibility (IE 8-11, Windows 7, 8.1, and 10), the viewer is probably only attractive for a small group of customers.

Visio is largely limited to features that allow you to draw **industry-standard vector graphics and diagrams and work in a team**. If you release a diagram, authorized team members can work on it simultaneously in the document as well as exchange ideas using the built-in **Microsoft Teams plugin**. You simply share finished sketches and prototypes via the connected cloud.

If you want to create **code frames** from your diagrams, you need other **UML tools**, for example, **Visual Studio From** Microsoft. As this is also part of the product family, the integrated development environment allows Visio to be integrated into the development process. However, from Enterprise 2017 Visual Studio onwards, UML templates are no longer integrated, which is why it didn't get its own spot in the list. Save your work in the UML diagram tool in native OPC/XML metadata formats so that you can export your diagrams and continue working on them without any data loss.

Native Visio formats:

- VSD (drawing)
- VSS (stencil)
- VST (template)
- VSW (web drawing)
- VSDX (OPC/XML drawing)

- VSDM (OPC/XML drawing, macro-enabled)
- VSSX (OPC/XML stencil)
- VSSM (OPC/XML stencil, macro-enabled)
- VSTX (OPC/XML template)
- VSTM (OPC/XML template, macro-enabled)
- VSL (add-on)

Visio has neither code generation nor round-trip engineering functions. However, it enables **reverse engineering** where you create UML diagrams from imported data or source code. Therefore, the UML tool fulfils its main task, which consists of clearly displaying processes and systems. Visio also enables you to integrate real-time data into your diagrams with a single click. This so-called data linking illustrates complex business processes (with BPMN) during their entire runtime, for example. Alternatively, you can test dependencies within a system using a UML communication diagram – live with real data.

Advantages	Disadvantages
✓ Supports UML 2.0 (Online Plan 2 also supports UML 2.5), including XML metadata	✗ Steep learning curve for beginners
✓ Real-time data linking helps with live testing	✗ Expensive compared to other options
✓ Clear user interface with familiar ribbon menu	✗ No integrated code frame generation

Comparing UML tools: a summary

Whether free UML tool, online UML tool, or integrated development environment, the best UML tool is one that performs the desired tasks efficiently and uses as few resources as possible. As you can see from the previous list, there are UML tools of every color. Whether you want to make a quick draft or create source code frames for a complex system and analyze them using diagrams – one of these applications will certainly be able to help you. Since nobody wants to spend the entire monthly budget on a tool for creating drafts, we have also considered the prices for individual offers in our comparison. In the following overview, we summarize the most important key features of UML diagram tools.

UML tool	Type	Platform	Supported programming	Integrated teamwork	Supports UML version	Suitable for	Price

			languages/formats				
Gliffy	Web-based diagram software	Browser, plugin for Confluence or Jira, Google Apps, Hipchat	JavaScript, HTML5, VDX, gXML	✓	UML 2.5	Beginners, drafts	Free. Premium subscription from $5 per month
ArgoUML	Open-source UML software under Eclipse Public License 1.0	Desktop, platform-independent	Java and C++ (plugin), C#, PHP4 and 5, Ruby	✗	43191	UML beginners, drafts	Free
MagicDraw	Propriety software with single-user, floating, and mobile license	Desktop, platform-independent	Java, C++, C#, CIL, XML, CORBA, WSDL, EJB, DDL, IDL	✓ Team server in Enterprise edition	1.4 – 2.5	Large companies, complex tasks	Single payment from $300
Lucidchart	Web-based diagram software	Browser, app for iOS and Android, plugin	JavaScript, HTML5, SQL	✓	UML	Drafts, large projects, teamwork	Basic subscription from around $5 per month, Education license is free
IBM Rational Rhapsody	Graphical development environment for	Platform-independent	Java EE, C++, C#, EJB, WSDL, XSD,	With plugin	UML 2.1	Modular, full integratable IDE	On request

	model-based software development and validation		CORBA IDL, SQL, .NET				
Microsoft Visio	Proprietary diagram and vector graphics software	Windows, browser, iPad app, virtual machine	C++, C#, VSDX, VSDM	✓	UML 2.0 + 2.5	Visualizing large projects	Subscription to Visio Online Plan 2 from around $18/month, Visio Professional requires one-off payment of $590.

7. StarUML - Open Source UML Tool

StarUML is an open source software modeling tool that supports UML (Unified Modeling Language). It is based on UML version 1.4, provides eleven different types of diagram and it accepts UML 2.0 notation. It actively supports the MDA (Model Driven Architecture) approach by supporting the UML profile concept and allowing to generate code for multiple languages.

Since this evaluation, the StarUML open source software modeling tool project has been stopped and taken over by a commercial company that now sells a tool called StarUML 3!!! There is however an active open source project called WhiteStarUML that is a fork of StarUML 5.0. It provides a number of bug fixes and an improved compatibility with the modern versions of Windows.

When you start a new project, StarUML proposes which approach you want to use: 4+1 (Kruchten), Rational, UML components (from Cheesman and Daniels book), default or empty. Depending on the approach, profiles and/or frameworks may be included and loaded. If you don't follow a specific approach, the "empty" choice could be used. Although a project can be managed as one file, it may be convenient to divide it into many units and manage them separately if many developers are working on it together.

StarUML makes a clear conceptual distinction between models, views and diagrams. A Model is an element that contains information for a software model. A View is a visual expression of the information contained in a model, and a Diagram is a collection of view elements that represent the user's specific design thoughts.

StarUML is built as a modular and open tool. It provides frameworks for extending the functionality of the tool. It is designed to allow access to all functions of the model/meta-model and tool through COM Automation, and it provides extension of menu and option items. Also,

users can create their own approaches and frameworks according to their methodologies. The tool can also be integrated with any external tools.

StarUML supports the following diagram types

- Use Case Diagram
- Class Diagram
- Sequence Diagram
- Collaboration Diagram
- Statechart Diagram
- Activity Diagram
- Component Diagram
- Deployment Diagram
- Composite Structure Diagram

The user interface is intuitive. On the upper right side, a window allows to rapidly navigate between all the content of a project, adopting either a model or a diagram view. Multiple diagrams can be open at the same time and tabs allow switching rapidly between views. The lower right window allows to document the current diagram, either with plain text or attaching an external document. During diagram editing, "wizards" are located around the object that give you the quick shortcuts to main associated tasks with your current operation, like adding an attribute when you create a class for instance. A right-click on the mouse brings the full set of operations at your disposal.

StarUML has also a model verification feature. You can export diagram in different formats (jpg, bmp, wmf). It also supports a patterns approach and import of Rational Rose files.

StarUML Generator is platform module to generate various artifacts (like as Microsoft Word, Excel, PowerPoint, and Text-based artifacts) by templates depending on UML model elements in StarUML. The users can define their own templates and can apply many different kinds of templates to the same UML model, so the users can get various artifacts automatically, easily and fast. The tool supports code generation and reverse engineering for Java, C# and C++.

8. Visual Paradigm

Visual Paradigm (VP-UML) is a UML CASE Tool supporting UML 2, SysML and Business Process Modeling Notation (BPMN) from the Object Management Group (OMG). In addition to modeling support, it provides report generation and code engineering capabilities including code generation. It can reverse engineer diagrams from code and provide round-trip engineering for various programming languages.

Features of VP-UML 4.1:

- Full UML 2.0 support. VP-UML supports all aspects of UML 2.0 including latest UML diagrams and UML notations, such as Interaction Overview Diagram, Activity Diagram, Timing Diagram, Sequence Diagram, Composite Structure Diagram, State Diagram, Action, Exception Handler, Timing Frame, Port and many more.
- Support of Requirement Capturing. You can perform Use Case modeling, Textual Analysis and create CRC Card Diagram for better requirement capturing.
- Reverse Engineer for Legacy System. Reverse XML, database tables through JDBC, Hibernate format, XML schema, .NET dll or exe files, Java source/class/jar files, C++ source files, CORBA IDL source files, Ada9x into Class models instantly.
- Support of EJB Development. Design EJBs in Class Diagram and perform bi-directional code generation between UML models and EJBs.
- Diagram layout facility. Tidy up messy diagrams with the sophisticated layout facility. Various layout styles are available, and each can be fine-tuned with a set of configurable parameters.
- Ad-hoc documentation generation. Customize report in an intuitive report builder and update the report with the latest design from time to time. Report can be exported as OpenOffice format or Microsoft word format.
- Teamwork Server. Share your design with your team members through your organization network or the Internet.
- Modeling with Visio Stencil. You can model domain-specific system by using Visio stencil, beyond the standard UML notations.
- Mouse gesture support. Execute commands in UML diagrams by holding down a mouse button and moving the mouse in a certain way to from a gesture.
- Object Relational Mapping (ORM) support. Generating the persistent Java source code, you need to access database(s).
- ER Diagram (ERD) support
- Real-time Java code synchronization
- Project exporting and merging
- HTML and PDF report generation
- Ability to export UML Diagram as images (JPEG, SVG, PNG)
- Support of Rose model import
- Support of XMI import
- Template and Plugin Architecture

IDE Integration (Eclipse/IBM WebSphere, JBuilder, JDeveloper, NetBeans/Sun ONE, IntelliJ IDEA, WebLogic Workshop)

3. CREATE A USE_CASE DIAGRAM

- **Add use case diagram to the browser**
- **Add actors and use cases to the diagram**
- **Attach Use case flow of events**

Give all notations of use case diagram in this section with examples

3.1 Introduction to Use Case Diagram

A use case diagram is a dynamic or behavior diagram in UML. Use case diagrams model the functionality of a system using actors and use cases. Use cases are a set of actions, services, and functions that the system needs to perform. In this context, a "system" is something being developed or operated, such as a web site. The "actors" are people or entities operating under defined roles within the system.

3.2 Importance of Use Case Diagram

As mentioned before use case diagram are used to gather a usage requirement of a system. Depending on your requirement you can use that data in different ways. Below are few ways to use them.

1. **To identify functions and how roles interact with them** – The primary purpose of use case diagrams.
2. **For a high-level view of the system** – Especially useful when presenting to managers or stakeholders. You can highlight the roles that interact with the system and the functionality provided by the system without going deep into inner workings of the system.
3. **To identify internal and external factors** – This might sound simple but in large complex projects a system can be identified as an external role in another use case.

3.3 Basic Use Case Diagram Symbols and Notations

System

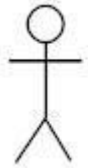
Draw your system's boundaries using a rectangle that contains use cases. Place actors outside the system's boundaries.

Use Case

Draw use cases using ovals. Label the ovals with verbs that represent the system's functions.

Actors

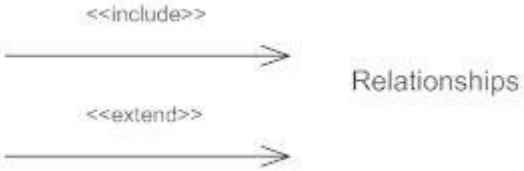
Actors are the users of a system. When one system is the actor of another system, label the actor system with the actor stereotype.



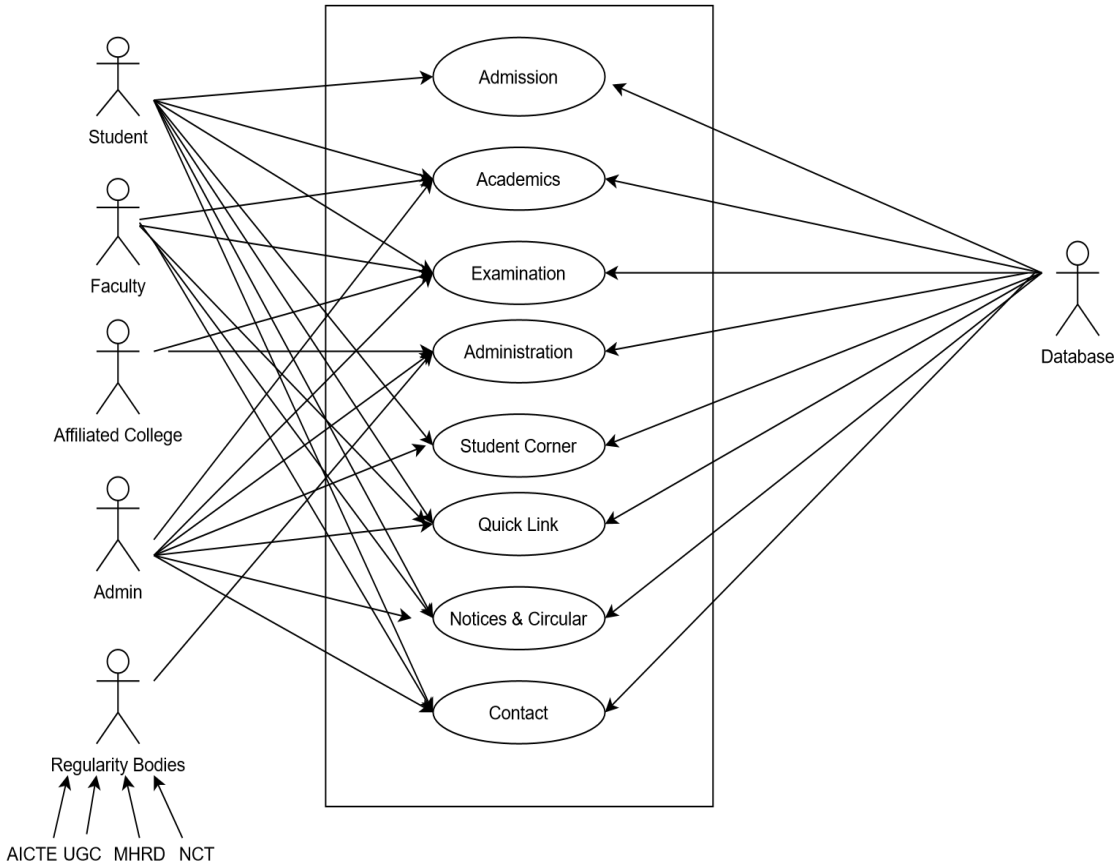
Actor

Relationships

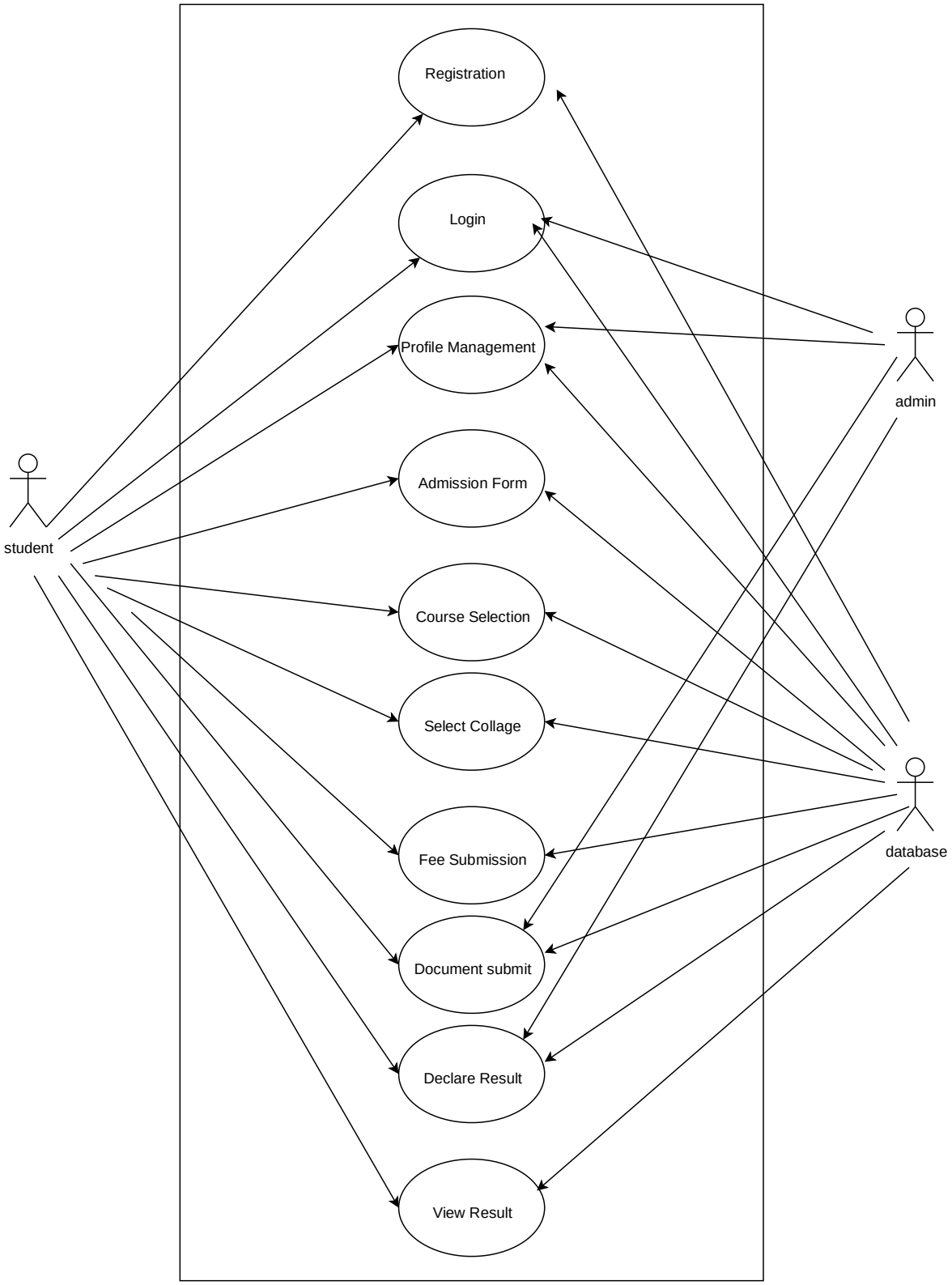
Illustrate relationships between an actor and a use case with a simple line. For relationships among use cases, use arrows labeled either "uses" or "extends." A "uses" relationship indicates that one use case is needed by another in order to perform a task. An "extends" relationship indicates alternative options under a certain use case.



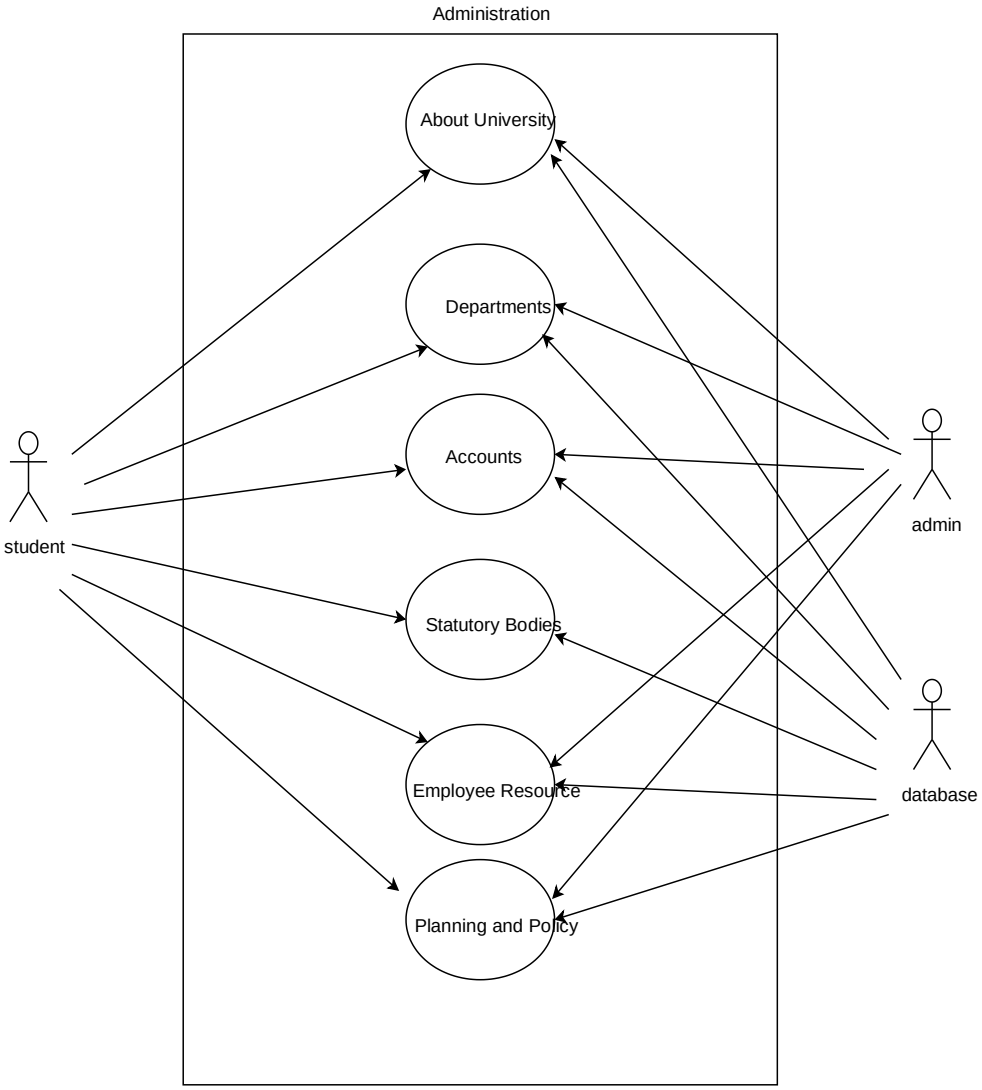
4. Use case diagram for system/subsystem "*subsystem name*" of the case study *GGSIPU*



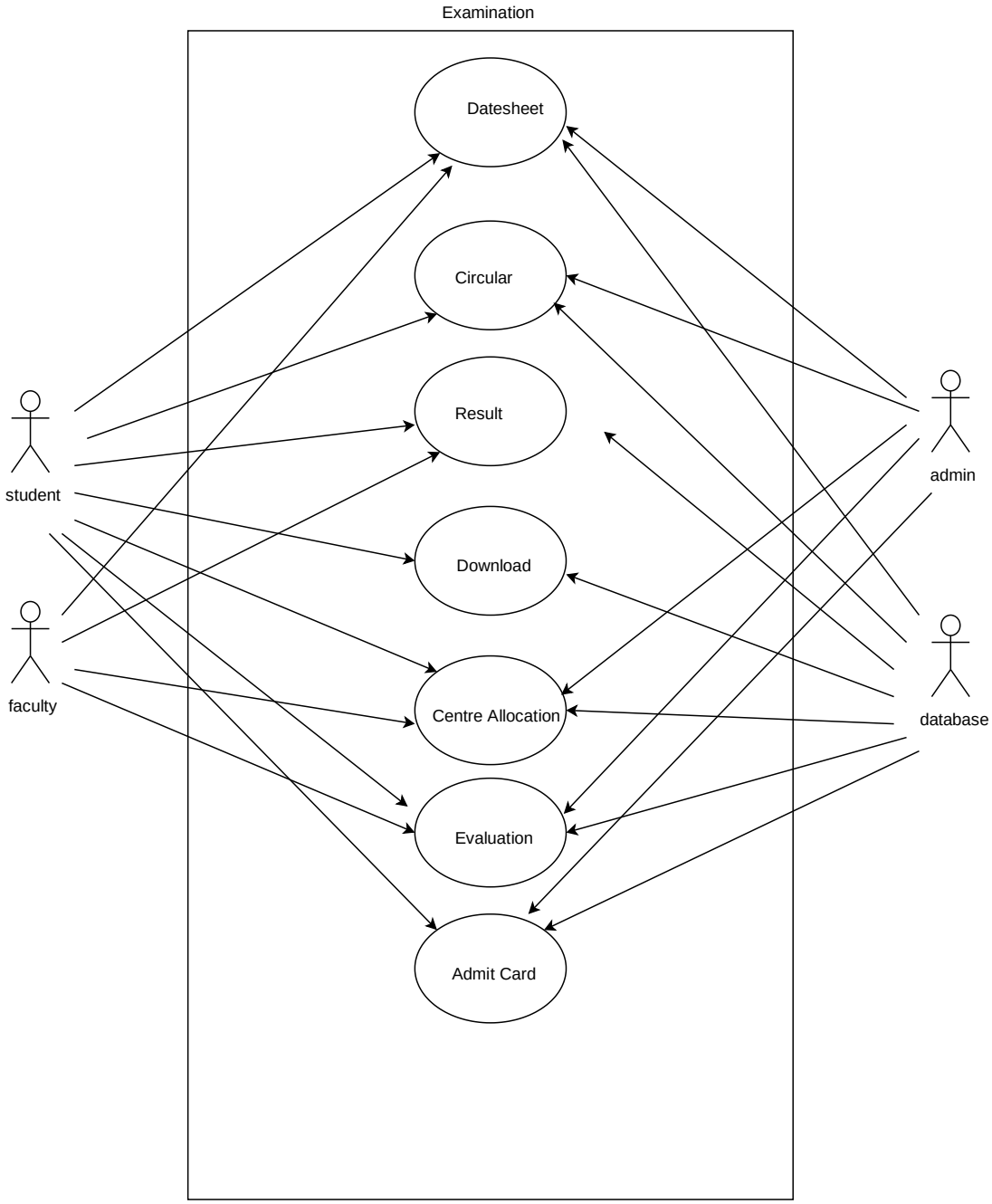
- Subsystem use case for Admission



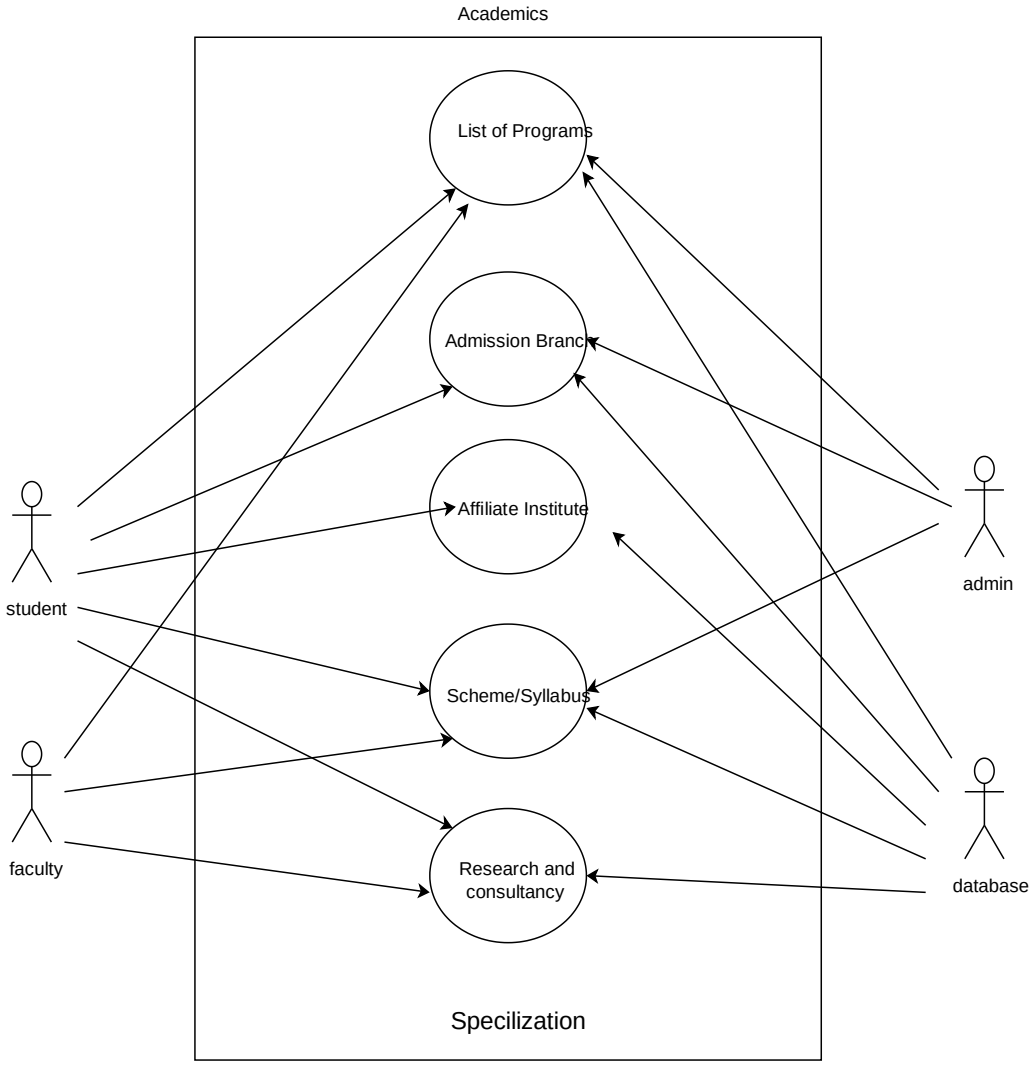
- Subsystem use case for Administration



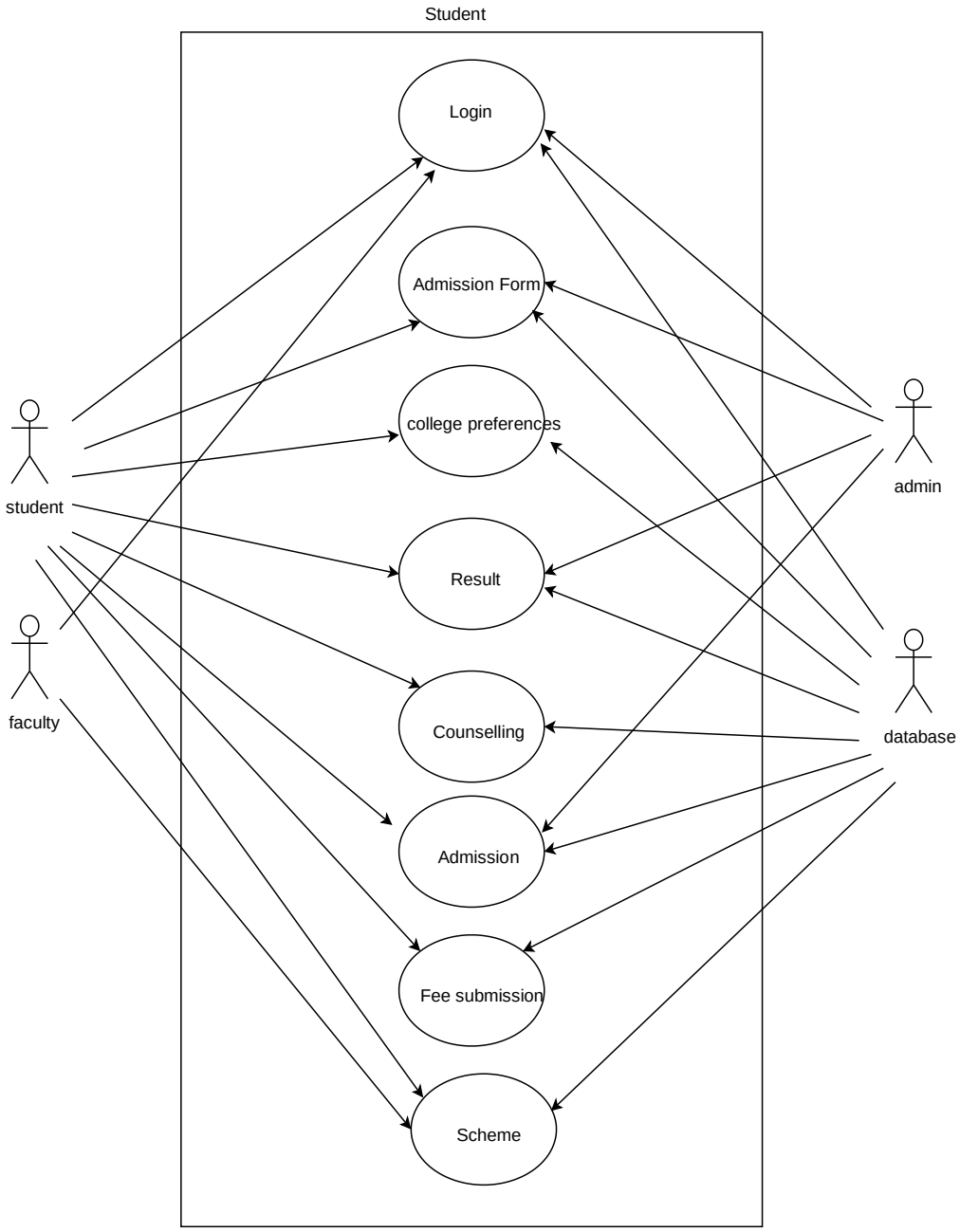
- Subsystem use case for Examination



- Subsystem use case for Academics



- Subsystem use case for student



5. CREATE A CLASS DIAGRAM

- Add class diagram to the browser
- Add classes to the diagram
- Add associations
- Add role names and multiplicity

Give all notations of class diagram in this section with examples.

6.1 Introduction to class diagram

Class Diagram gives the static view of an application. A class diagram describes the types of objects in the system and the different types of relationships that exist among them. This modeling method can run with almost all Object-Oriented Methods.
UML Class Diagram gives an overview of a software system by displaying classes, attributes, operations, and their relationships. This Diagram includes the class name, attributes, and operation in separate designated compartments.
Class Diagram helps construct the code for the software application development.

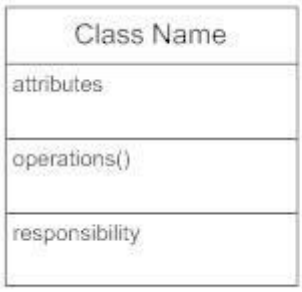
6.2 Benefits

- ❖ Class Diagram Illustrates data models for even very complex information systems
- ❖ It provides an overview of how the application is structured before studying the actual
- ❖ code. This can easily reduce the maintenance time
- ❖ It helps for better understanding of general schematics of an application.
- ❖ Allows drawing detailed charts which highlights code required to be programmed
- ❖ Helpful for developers and other stakeholders.

2.1 6.3 Basic Class Diagram Symbols and Notations

2.1.1.1 Classes

Classes represent an abstraction of entities with common characteristics. Associations represent the relationships between classes.
Illustrate classes with rectangles divided into compartments. Place the name of the class in the first partition (centered, bolded, and capitalized), list the attributes in the second partition (left-aligned, not bolded, and lowercase), and write operations into the third.



Class

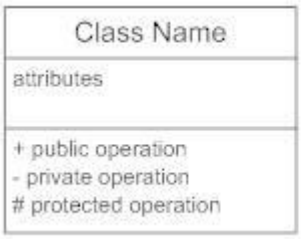
2.1.1.2 Active Classes

Active classes initiate and control the flow of activity, while passive classes store data and serve other classes. Illustrate active classes with a thicker border.

Active class

2.1.1.3 Visibility

Use visibility markers to signify who can access the information contained within a class. Private visibility, denoted with a - sign, hides information from anything outside the class partition. Public visibility, denoted with a + sign, allows all other classes to view the marked information. Protected visibility, denoted with a # sign, allows child classes to access information they inherited from a parent class.

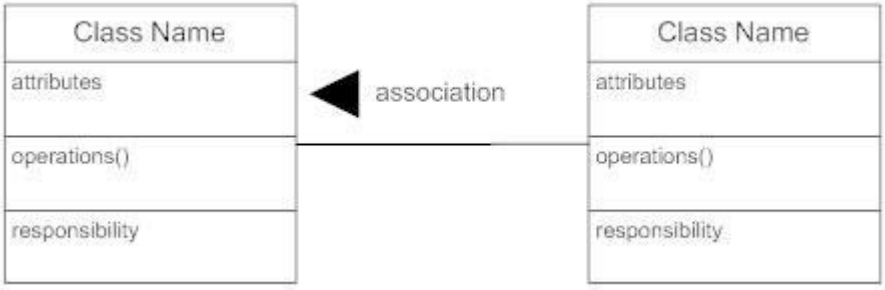


Visibility

Marker	Visibility
+	public
-	private
#	protected
~	package

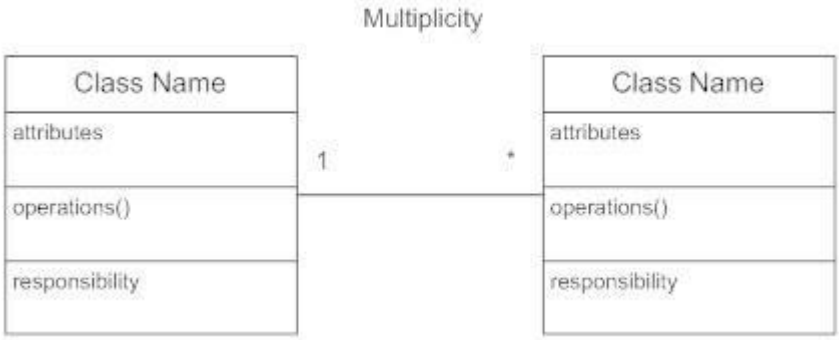
2.1.1.4 Associations

Associations represent static relationships between classes. Place association names above, on, or below the association line. Use a filled arrow to indicate the direction of the relationship. Place roles near the end of an association. Roles represent the way the two classes see each other.



2.1.1.5 Multiplicity (Cardinality)

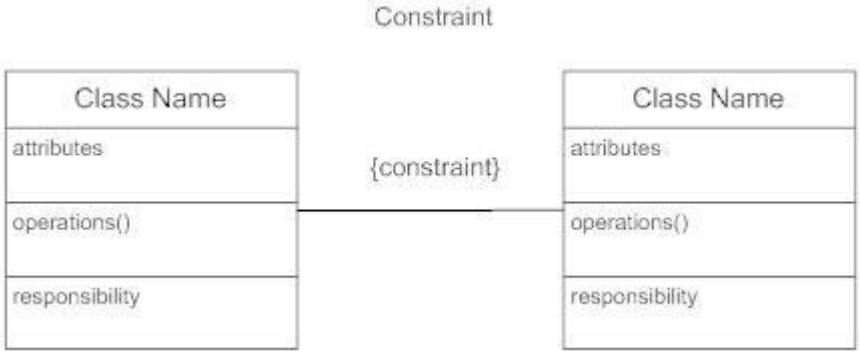
Place multiplicity notations near the ends of an association. These symbols indicate the number of instances of one class linked to one instance of the other class. For example, one company will have one or more employees, but each employee works for just one company.



Indicator	Meaning
0..1	Zero or one
1	One only
0..*	0 or more
1..* *	1 or more
n	Only n (where n > 1)
0..n	Zero to n (where n >1)
1..n	One to n (where n > 1)

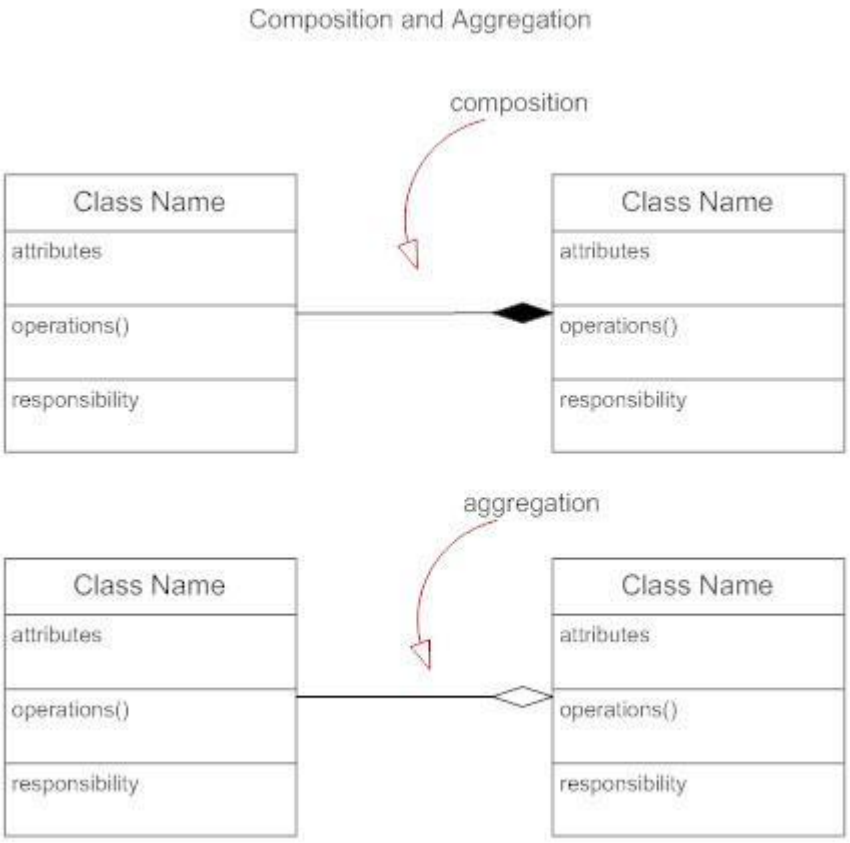
2.1.1.6 Constraint

Place constraints inside curly braces {}.



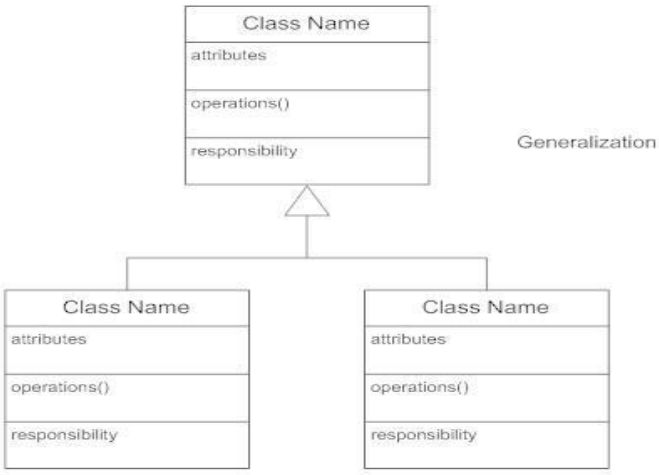
2.1.1.7 Composition and Aggregation

Composition is a special type of aggregation that denotes a strong ownership between Class A, the whole, and Class B, its part. Illustrate composition with a filled diamond. Use a hollow diamond to represent a simple aggregation relationship, in which the "whole" class plays a more important role than the "part" class, but the two classes are not dependent on each other. The diamond ends in both composition and aggregation relationships point toward the "whole" class (i.e., the aggregation).



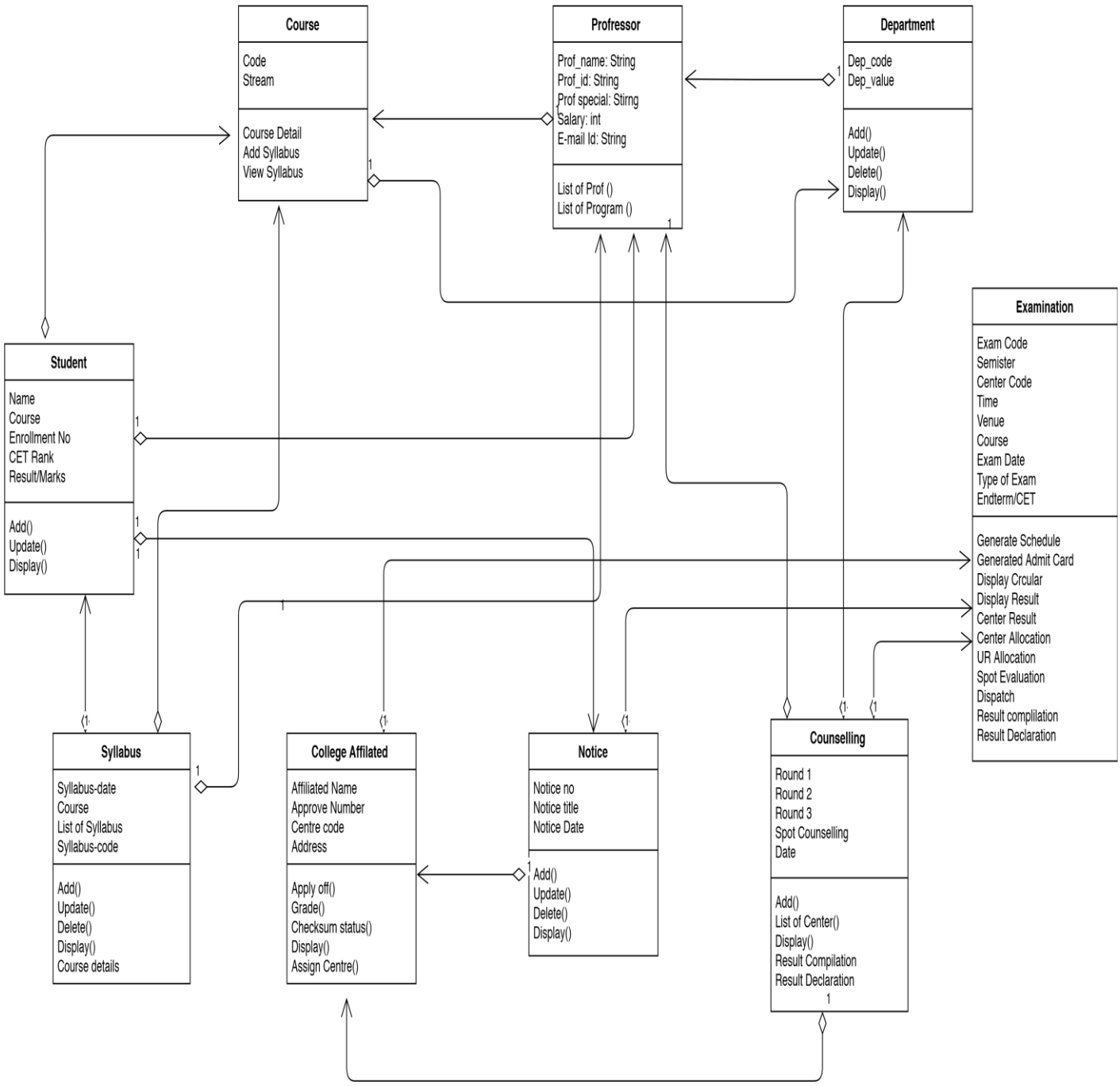
2.1.1.8 Generalization

Generalization is another name for inheritance or an "is a" relationship. It refers to a relationship between two classes where one class is a specialized version of another. For example, Honda is a type of car. So the class Honda would have a generalization relationship with the class car.



In real life coding examples, the difference between inheritance and aggregation can be confusing. If you have an aggregation relationship, the aggregate (the whole) can access only the PUBLIC functions of the part class. On the other hand, inheritance allows the inheriting class to access both the PUBLIC and PROTECTED functions of the superclass.

7. Class diagram for the case study *ipu.ac.in*



6. CREATE A SEQUENCE DIAGRAM

- Using the sequence diagram create the sequence diagram in your design model. Using
- the sequence diagram creates the sequence diagram in your design model.
- Add sequence diagram to the browser
- Add actor and classes
- Add object messages
- Add responsibilities to object messages
- Add a note

Give all notations of sequence diagram in this section with examples

8.1 Introduction to sequence diagram

From the term Interaction, it is clear that the diagram is used to describe some type of interactions among the different elements in the model. This interaction is a part of dynamic behavior of the system.

This interactive behavior is represented in UML by two diagrams known as Sequence diagram and Collaboration diagram. The basic purpose of both the diagrams is similar. Sequence diagram emphasizes on time sequence of messages and collaboration diagram emphasizes on the structural organization of the objects that send and receive messages.

8.2 Purpose of Interaction Diagrams:

The purpose of interaction diagrams is to visualize the interactive behavior of the system. Visualizing the interaction is a difficult task. Hence, the solution is to use different types of models to capture the different aspects of the interaction.

Sequence and collaboration diagrams are used to capture the dynamic nature but from a different
angle.

The purpose of interaction diagram is −

- ❖ To capture the dynamic behavior of a system.
- ❖ To describe the message flow in the system.
- ❖ To describe the structural organization of the objects.
- ❖ To describe the interaction among objects.

We have two types of interaction diagrams in UML. One is the sequence diagram and the other is the collaboration diagram. The sequence diagram captures the time sequence of the message flow from one object to another and the collaboration diagram describes the organization of objects in a system taking part in the message flow.

Following things are to be identified clearly before drawing the interaction diagram:

- ❖ Objects taking part in the interaction.
- ❖ Message flows among the objects.
- ❖ The sequence in which the messages are flowing.
- ❖ Object organization.

Following are two interaction diagrams modeling the order management system. The first diagram is a sequence diagram and the second is a collaboration diagram

8.3 Basic Sequence Diagram Notations

Class Roles or Participants

Class roles describe the way an object will behave in context. Use the UML object symbol to illustrate class roles, but don't list object attributes.

component

Activation or Execution Occurrence

Activation boxes represent the time an object needs to complete a task. When an object is busy executing a process or waiting for a reply message, use a thin gray rectangle placed vertically on its lifeline.

Activation or Execution Occurrence

Messages

Messages are arrows that represent communication between objects. Use half-arrowed lines to represent asynchronous messages. Asynchronous messages are sent from an object that will not wait for a response from the receiver before continuing its tasks.

Types of Messages in Sequence Diagrams

Synchronous Message

A synchronous message requires a response before the interaction can continue. It's usually drawn using a line with a solid arrowhead pointing from one object to another.

Synchronous

Asynchronous Message

Asynchronous messages don't need a reply for interaction to continue. Like synchronous messages, they are drawn with an arrow connecting two lifelines; however, the arrowhead is usually open and there's no return message depicted.

Simple, also used for asynchronous

Asynchronous

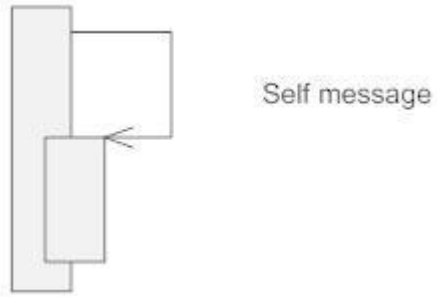
Reply or Return Message

A reply message is drawn with a dotted line and an open arrowhead pointing back to the original lifeline.

Reply or return message

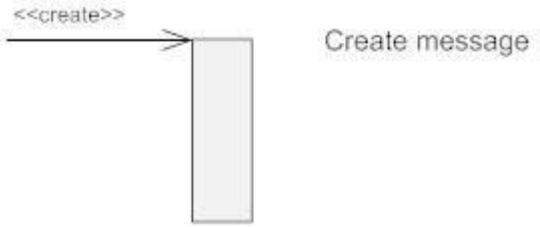
Self Message

A message an object sends to itself, usually shown as a U shaped arrow pointing back to itself.



Create Message

This is a message that creates a new object. Similar to a return message, it's depicted with a dashed line and an open arrowhead that points to the rectangle representing the object created.



Delete Message

This is a message that destroys an object. It can be shown by an arrow with an x at the end.

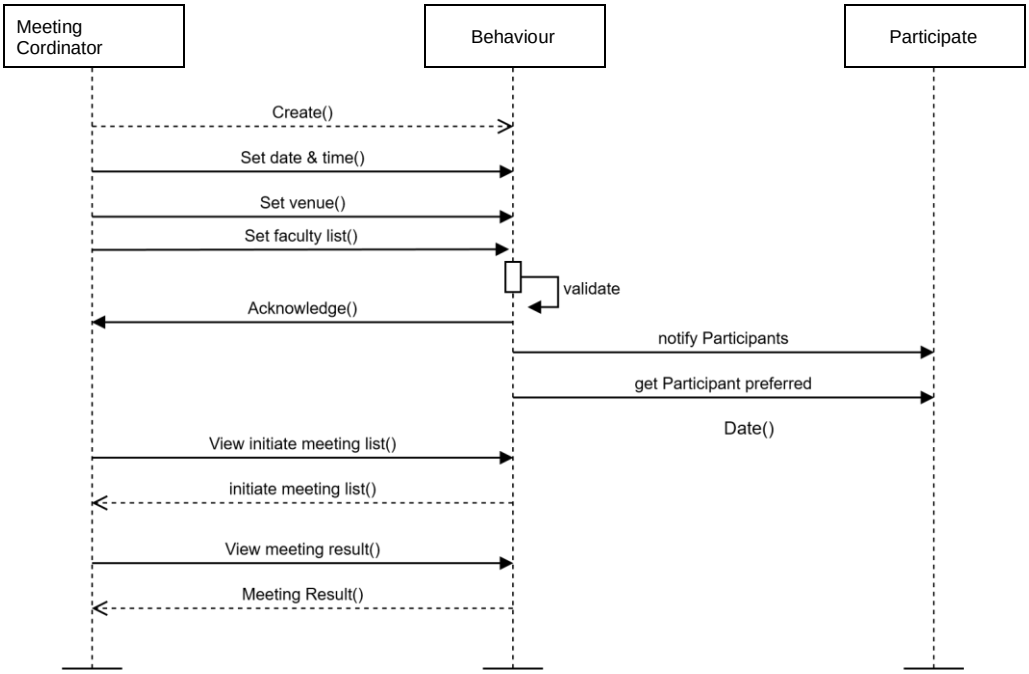
Found Message

A message sent from an unknown recipient, shown by an arrow from an endpoint to a lifeline.

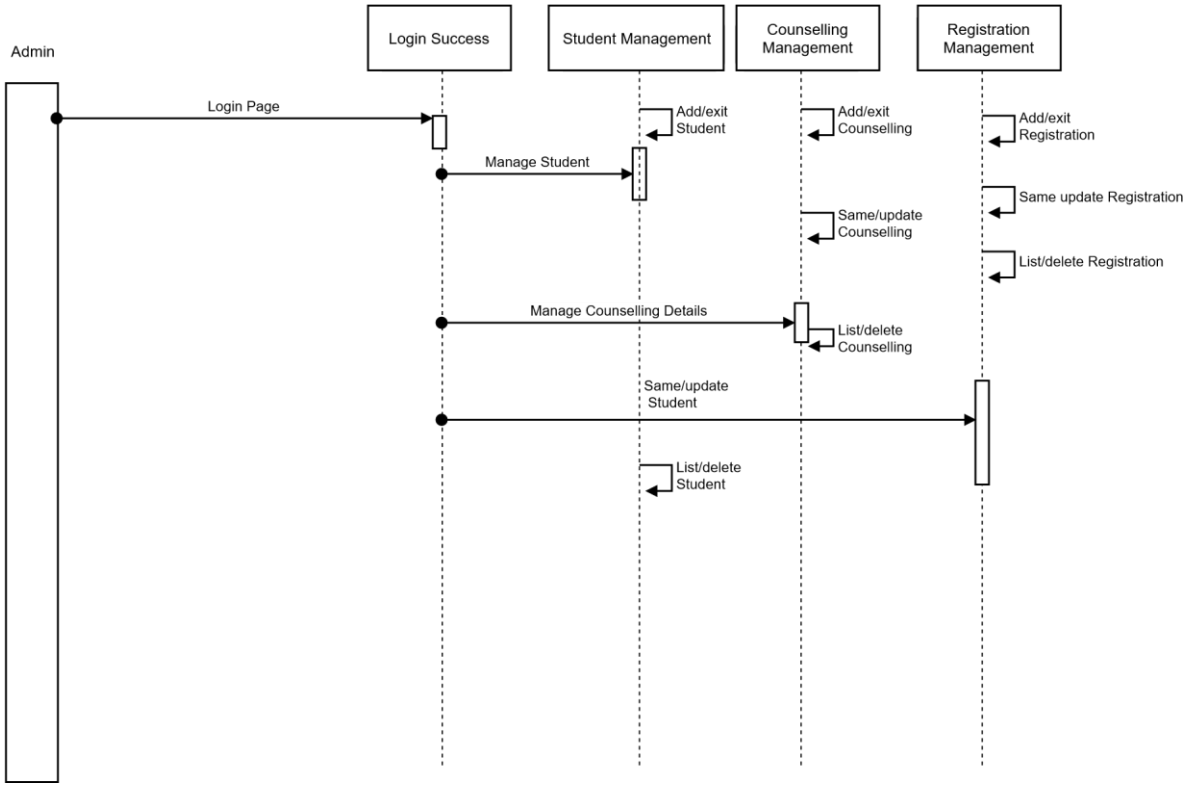
Lost Message

A message sent to an unknown recipient. It's shown by an arrow going from a lifeline to an endpoint, a filled circle or an x.

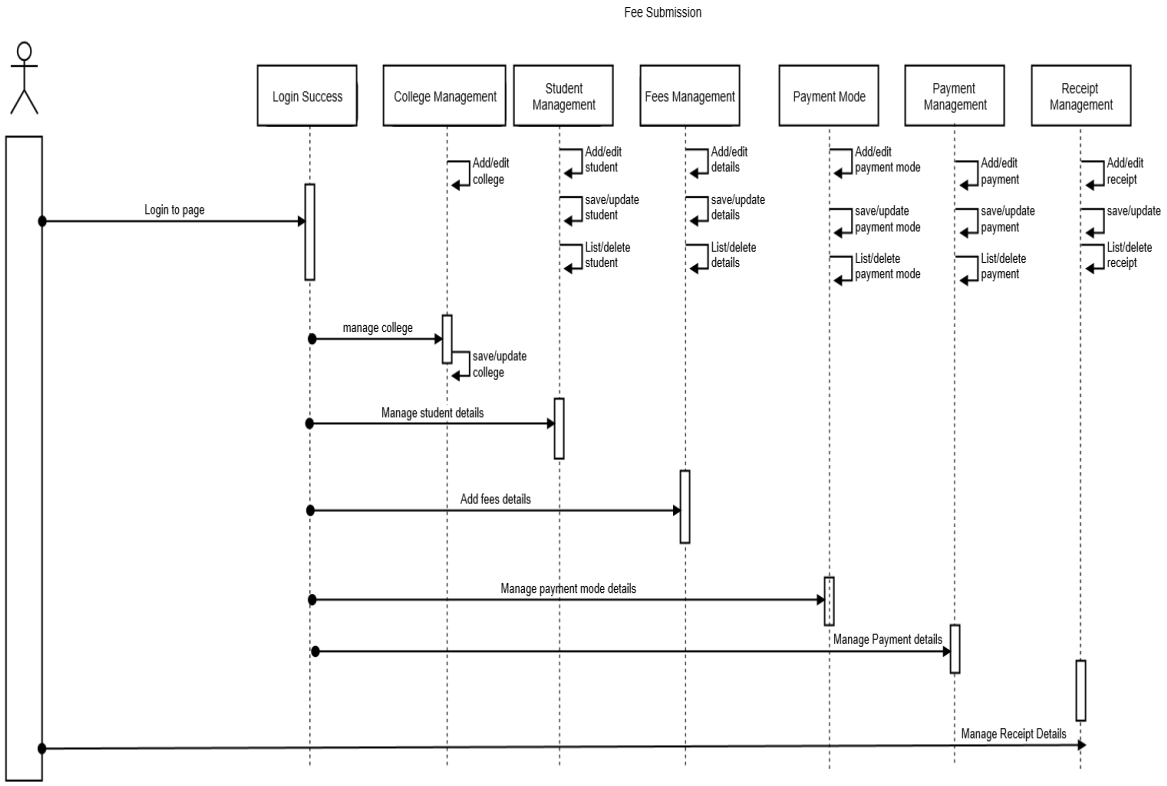
7. Sequence diagram for Program Coordinator Meeting Schedule



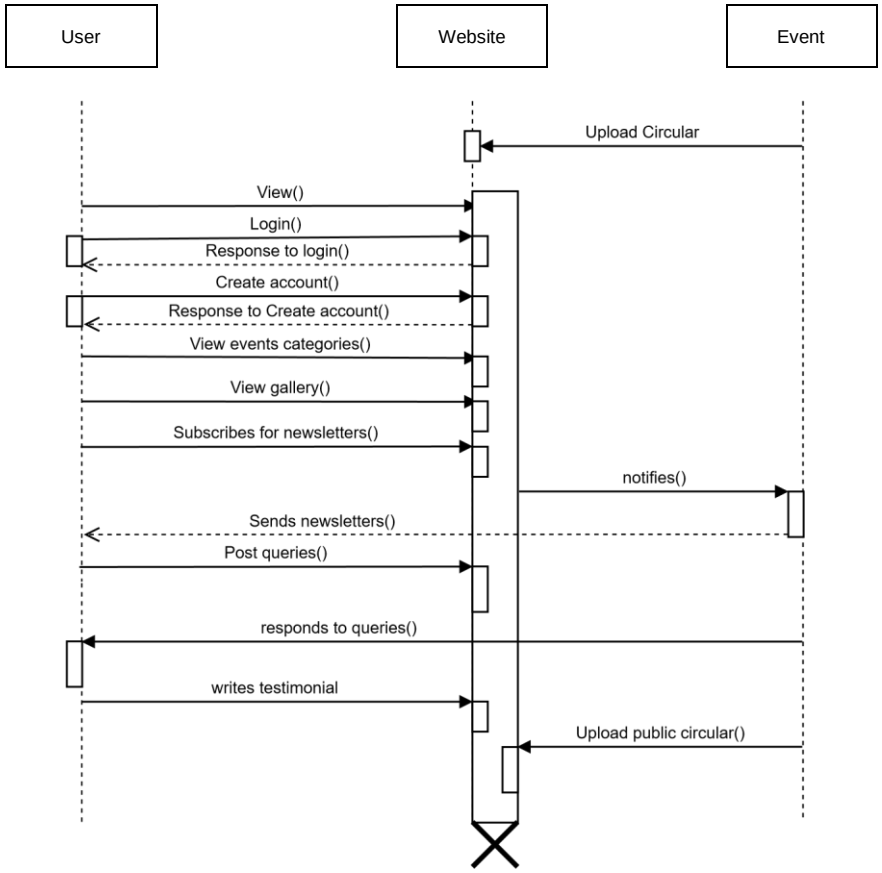
8. Sequence diagram for student counselling



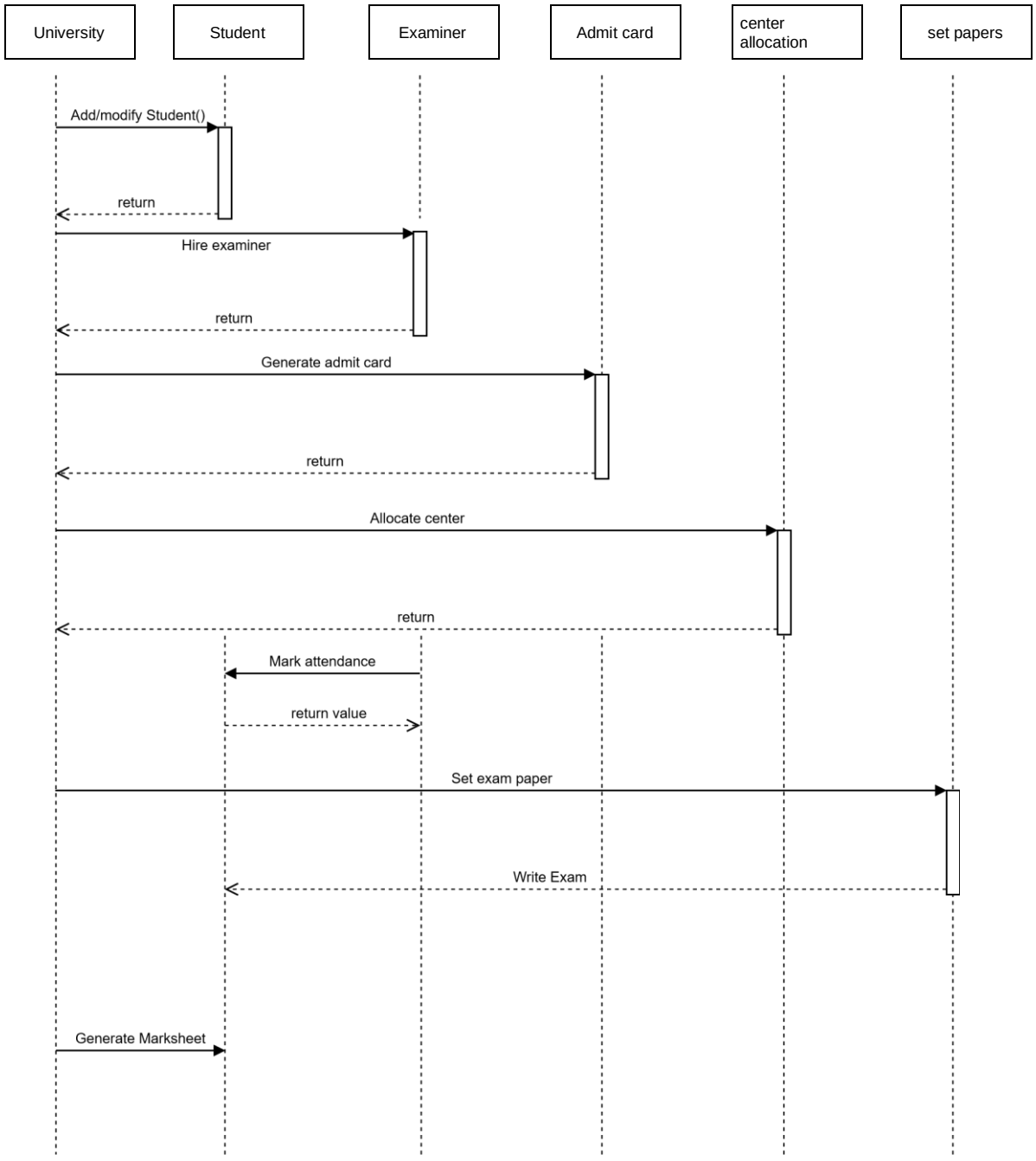
9. Sequence diagram for fee submission



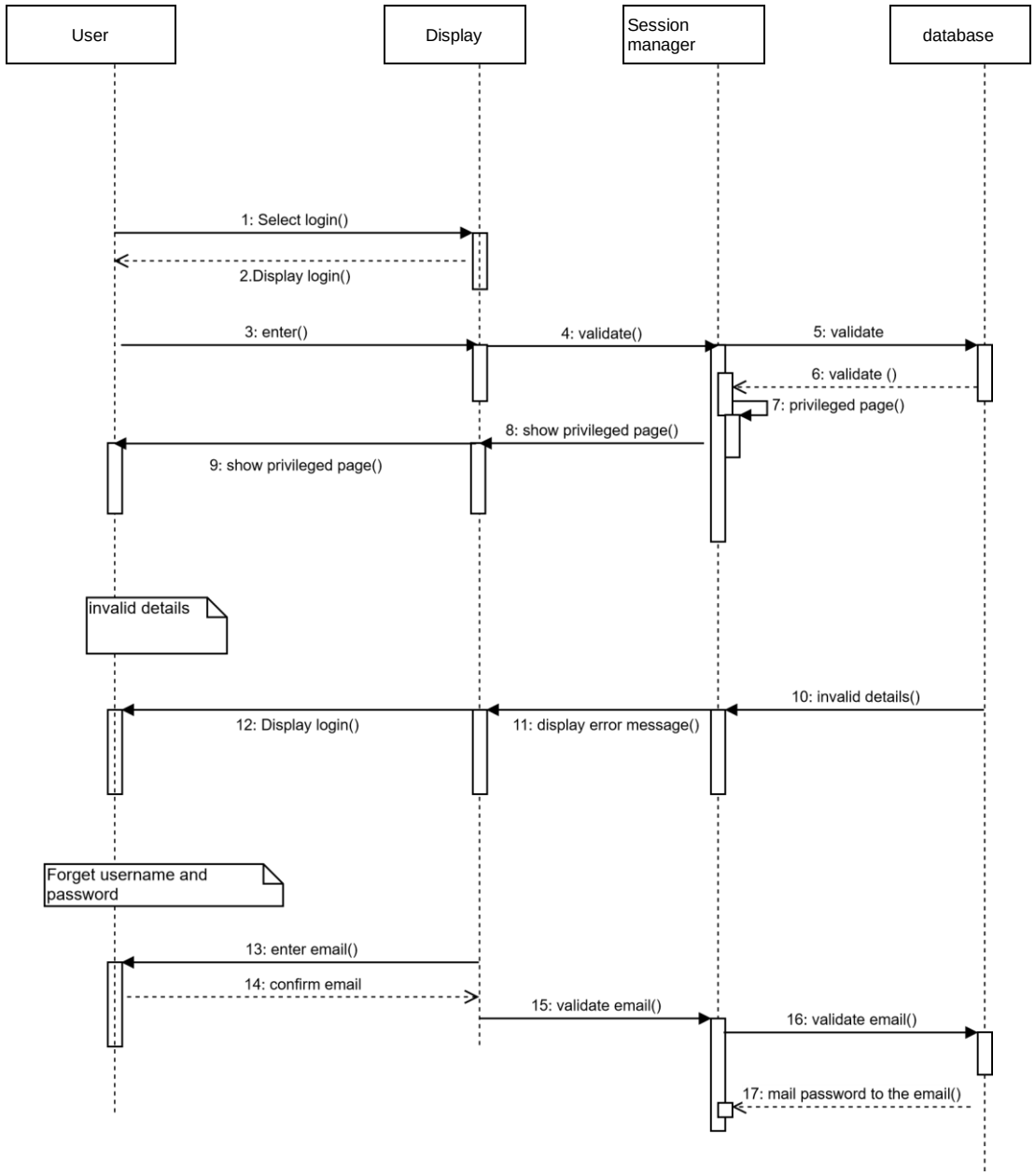
10. Sequence diagram for Upload circular for Anugoonj



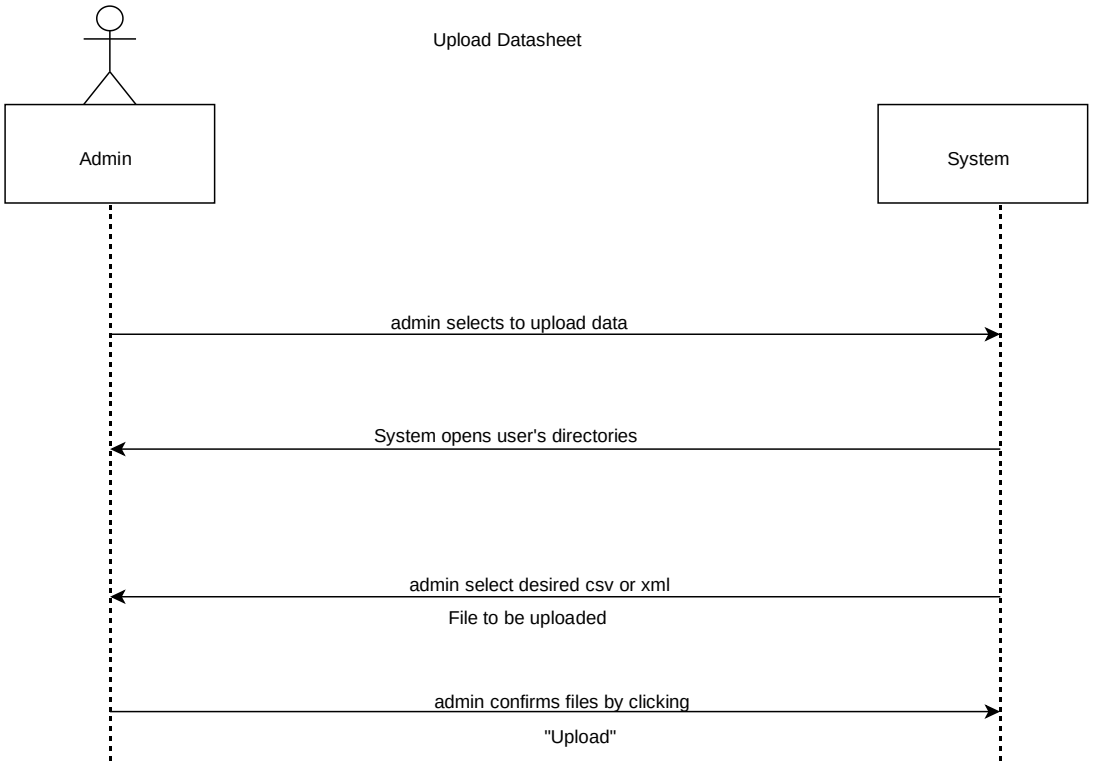
11. Sequence diagram for conduction of End-Term Examination



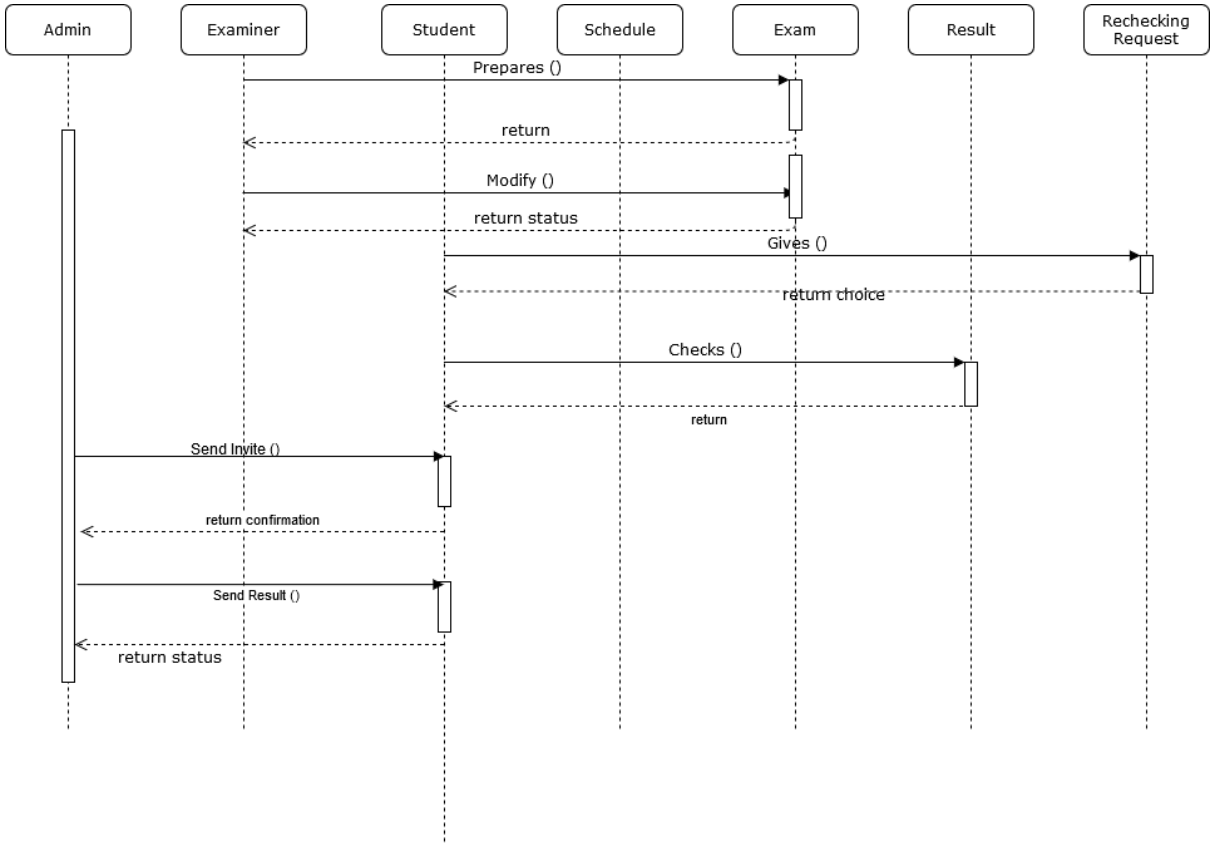
12. Sequence diagram for student/faculty Login



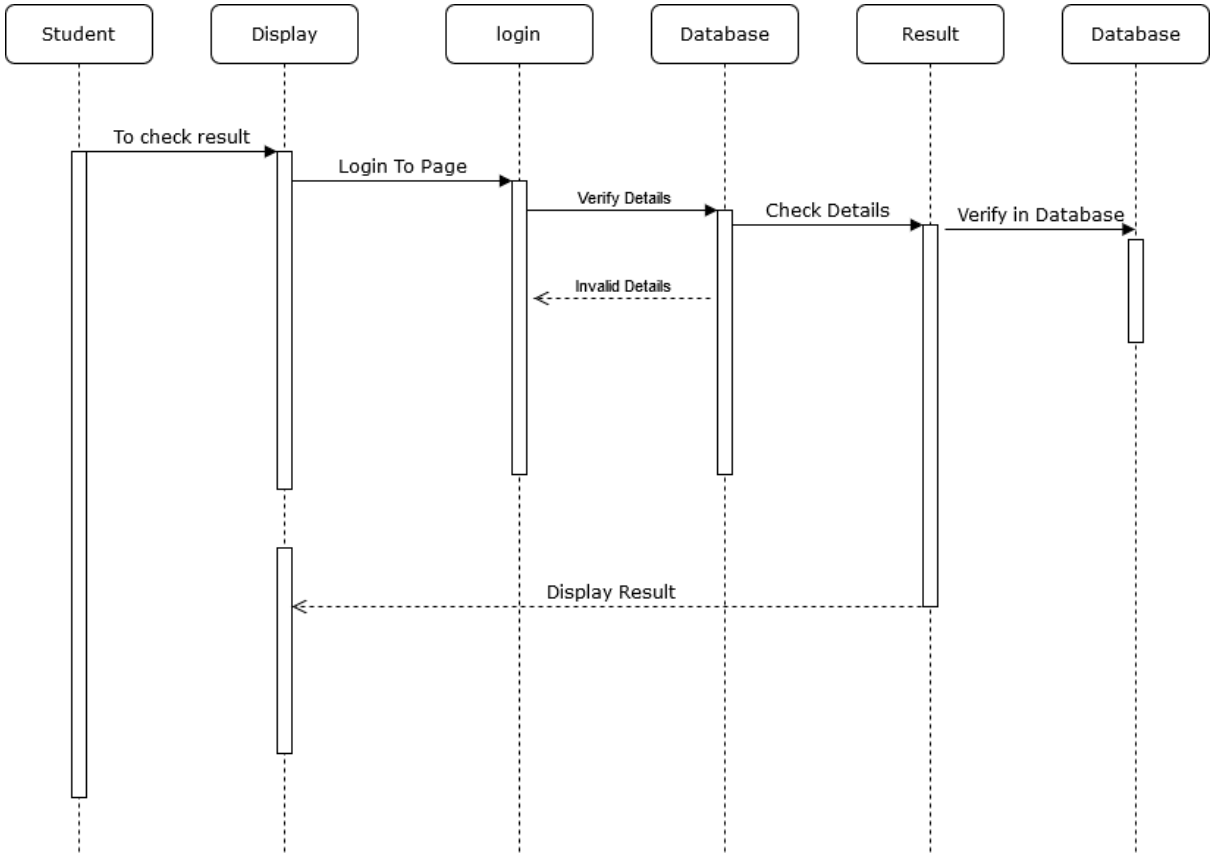
13. Sequence diagram for Upload datesheet



14. Sequence diagram for posting result



15. Sequence diagram for view result



16.Create an Activity Diagram

- Determine the activities in the basic flow of Use Case diagrams Determine the activities in the basic flow of Use Case diagrams
- Use activity diagram and follow the existing diagram to create the activity diagram in your model
- Add activity diagram to the browser
- Add activities
- Add start and end states
- Add state transitions, decisions and guard conditions
  Give all notations of Activity diagram in this section with examples

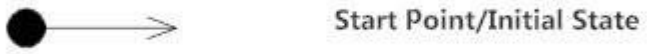
16.1 Introduction to Activity Diagram

An activity diagram visually presents a series of actions or flow of control in a system similar to a flowchart or a data flow diagram. Activity diagrams are often used in business process modeling. They can also describe the steps in a use case diagram. Activities modeled can be sequential and concurrent. In both cases an activity diagram will have a beginning (an initial state) and an end (a final state).
In between there are ways to depict activities, flows, decisions, guards, merge and time events and more. Learn about activity diagram symbols below:

16.2 Basic Activity Diagram Notations and Symbols

1.Initial State or Start Point

A small filled circle followed by an arrow represents the initial action state or the start point for any activity diagram. For activity diagram using swimlanes, make sure the start point is placed in the top left corner of the first column.



2.Activity or Action State

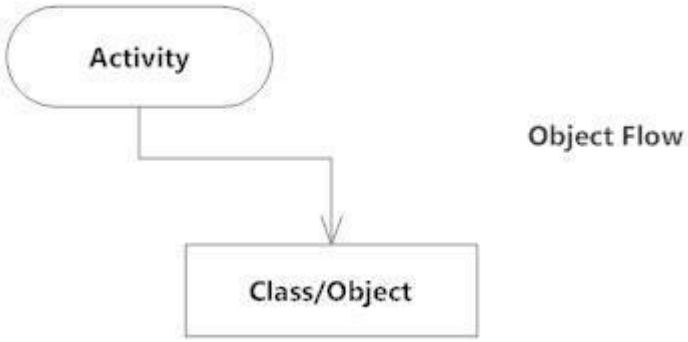
An action state represents the non-interruptible action of objects. You can draw an action state in SmartDraw using a rectangle with rounded corners.

3.Action Flow

Action flows, also called edges and paths, illustrate the transitions from one action state to another. They are usually drawn with an arrowed line.

4.Object Flow

Object flow refers to the creation and modification of objects by activities. An object flow arrow from an action to an object means that the action creates or influences the object. An object flow arrow from an object to an action indicates that the action state uses the object.



5.Decisions and Branching

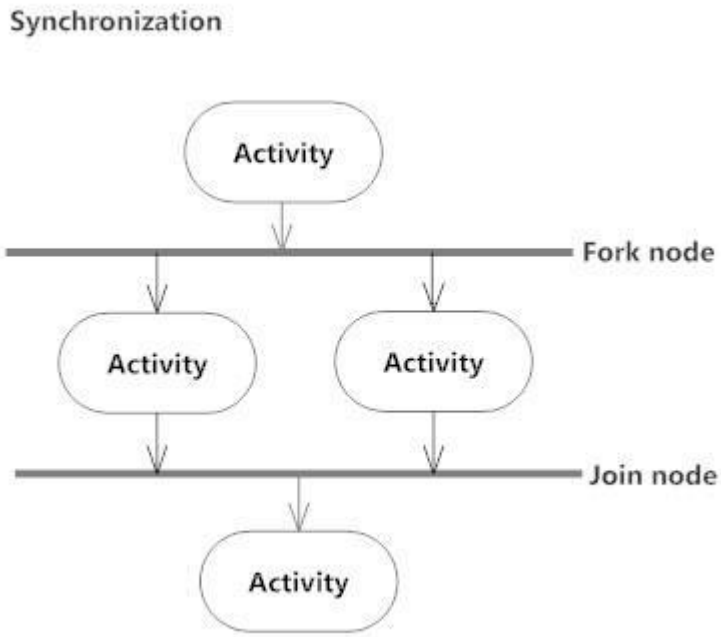
A diamond represents a decision with alternate paths. When an activity requires a decision prior to moving on to the next activity, add a diamond between the two activities. The outgoing alternates should be labeled with a condition or guard expression. You can also label one of the paths "else."

6.Guards

In UML, guards are a statement written next to a decision diamond that must be true before moving next to the next activity. These are not essential, but are useful when a specific answer, such as "Yes, three labels are printed," is needed before moving forward.

7.Synchronization

A fork node is used to split a single incoming flow into multiple concurrent flows. It is represented as a straight, slightly thicker line in an activity diagram.
A join node joins multiple concurrent flows back into a single outgoing flow.
A fork and join mode used together are often referred to as synchronization.

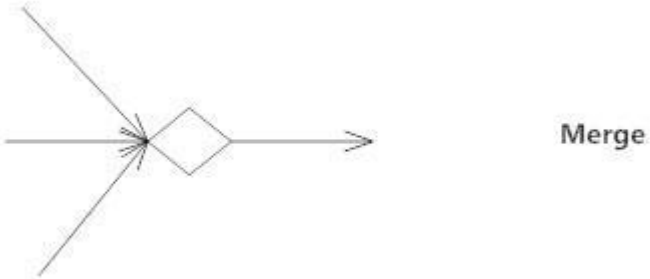


8.Time Event

This refers to an event that stops the flow for a time; an hourglass depicts it.

9.Merge Event

A merge event brings together multiple flows that are not concurrent.

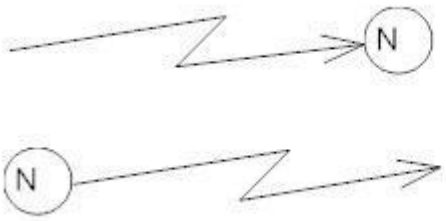


10.Sent and Received Signals

Signals represent how activities can be modified from outside the system. They usually appear in pairs of sent and received signals, because the state can't change until a response is received, much like synchronous messages in a sequence diagram. For example, an authorization of payment is needed before an order can be completed.

11.Interrupting Edge

An event, such as a cancellation, that interrupts the flow denoted with a lightning bolt.



Interrupting Edge Symbols

12.Swimlanes

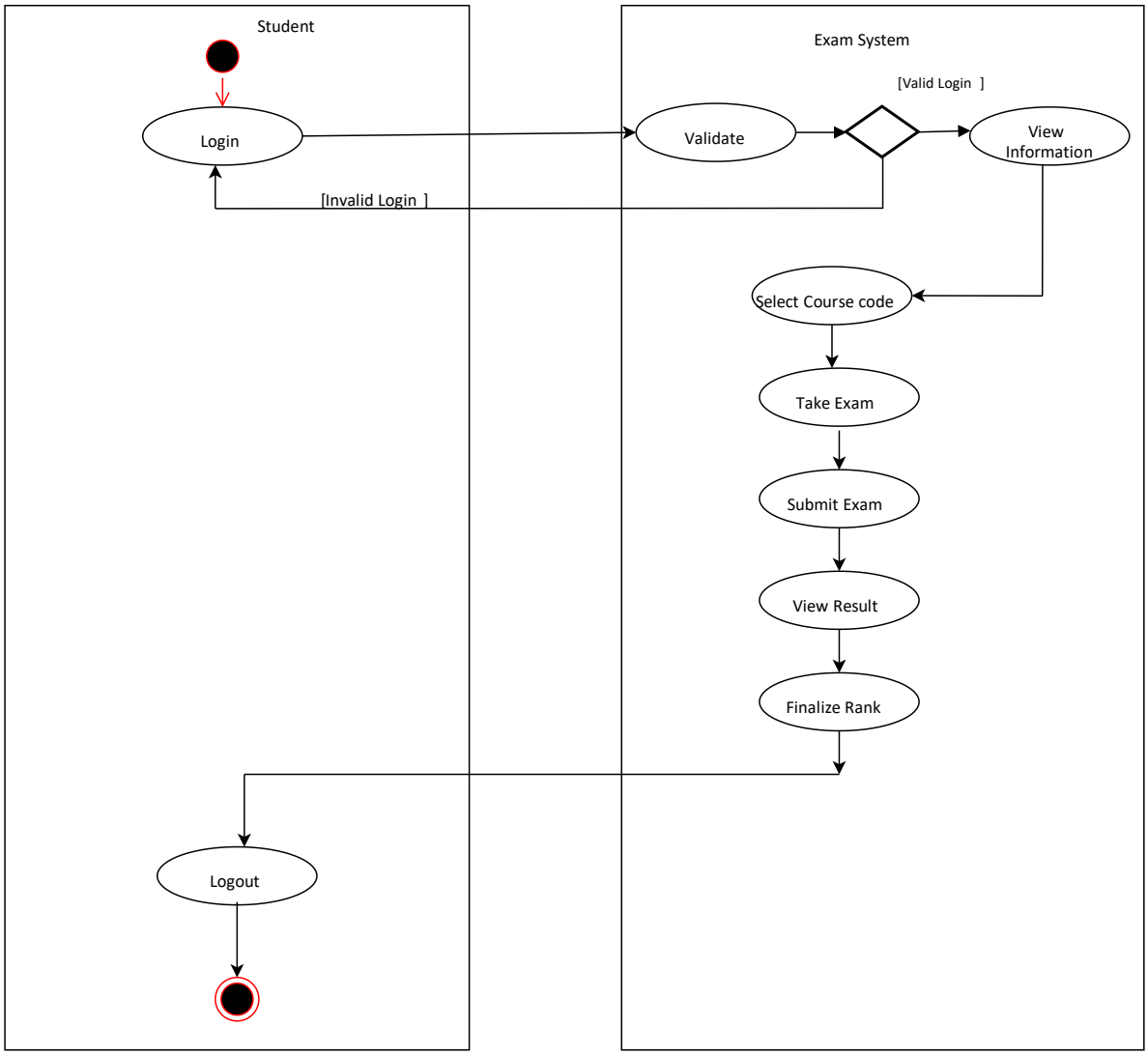
Swimlanes group related activities into one column.

13.Final State or End Point

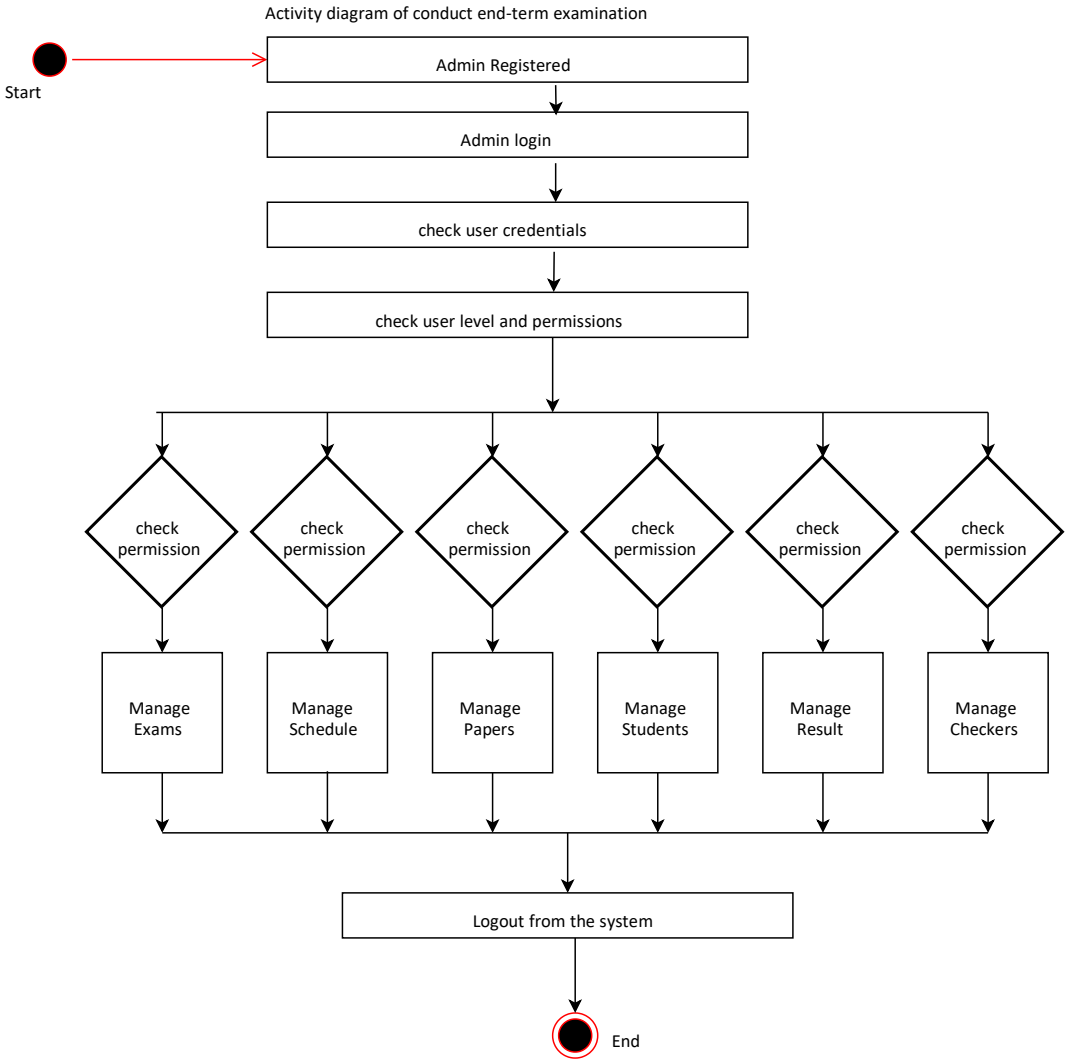
An arrow pointing to a filled circle nested inside another circle represents the final action state.

End Point Symbol

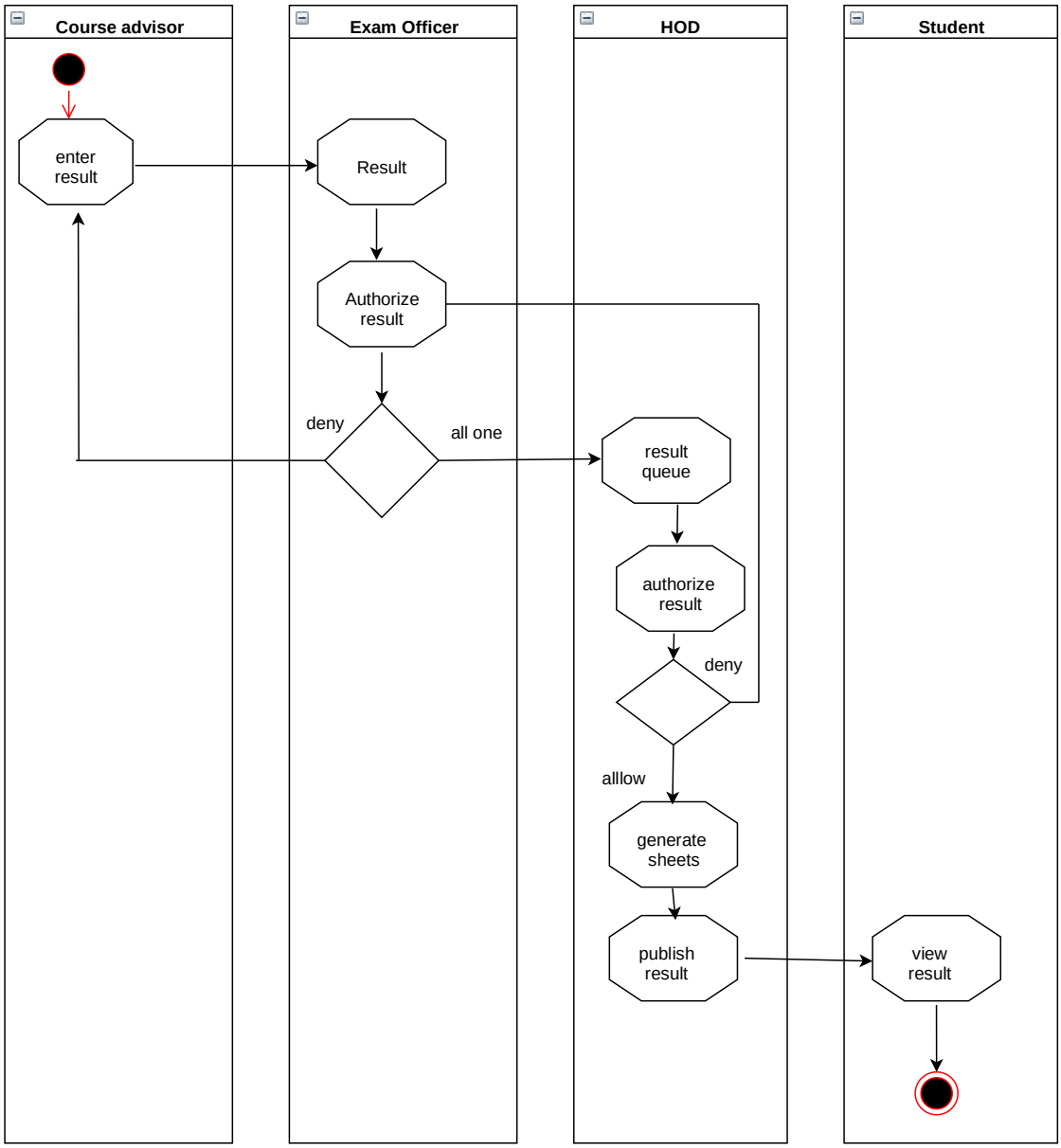
17. Activity diagram for online examination system



18. Activity diagram for conduct End-Term examination



19. Activity diagram for evaluation



20. CREATE A STATECHART DIAGRAM

- Add state chart diagram to the browser
- Add states
- Add start and end states
- Add transitions and self transitions
- Add decisions and synchronization

Give all notations of state chart diagram in this section with examples

Basic State Chart Diagram Symbols and Notations

States

States represent situations during the life of an object. You can easily illustrate a state in SmartDraw by using a rectangle with rounded corners.

A simple state

A state with internal activities

Transition

A solid arrow represents the path between different states of an object. Label the transition with the event that triggered it and the action that results from it. A state can have a transition that points back to itself.

Transition

Initial State

A filled circle followed by an arrow represents the object's initial state.

Initial state

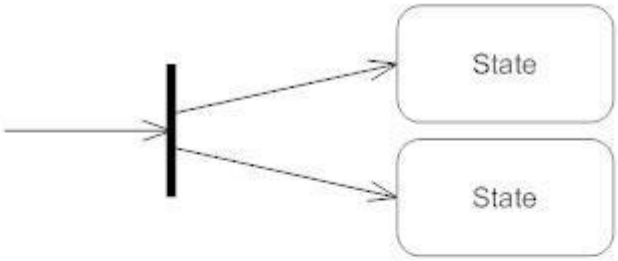
Final State

An arrow pointing to a filled circle nested inside another circle represents the object's final state.

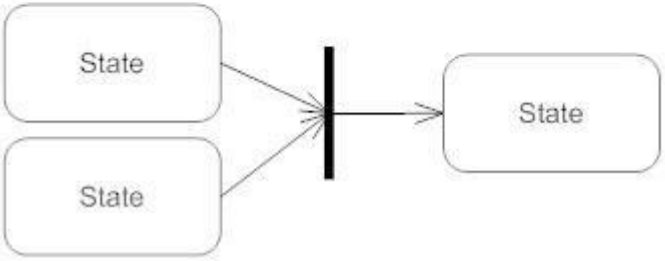
Final state

Synchronization and Splitting of Control

A short heavy bar with two transitions entering it represents a synchronization of control. The first bar is often called a fork where a single transition splits into concurrent multiple transitions. The second bar is called a join, where the concurrent transitions reduce back to one.

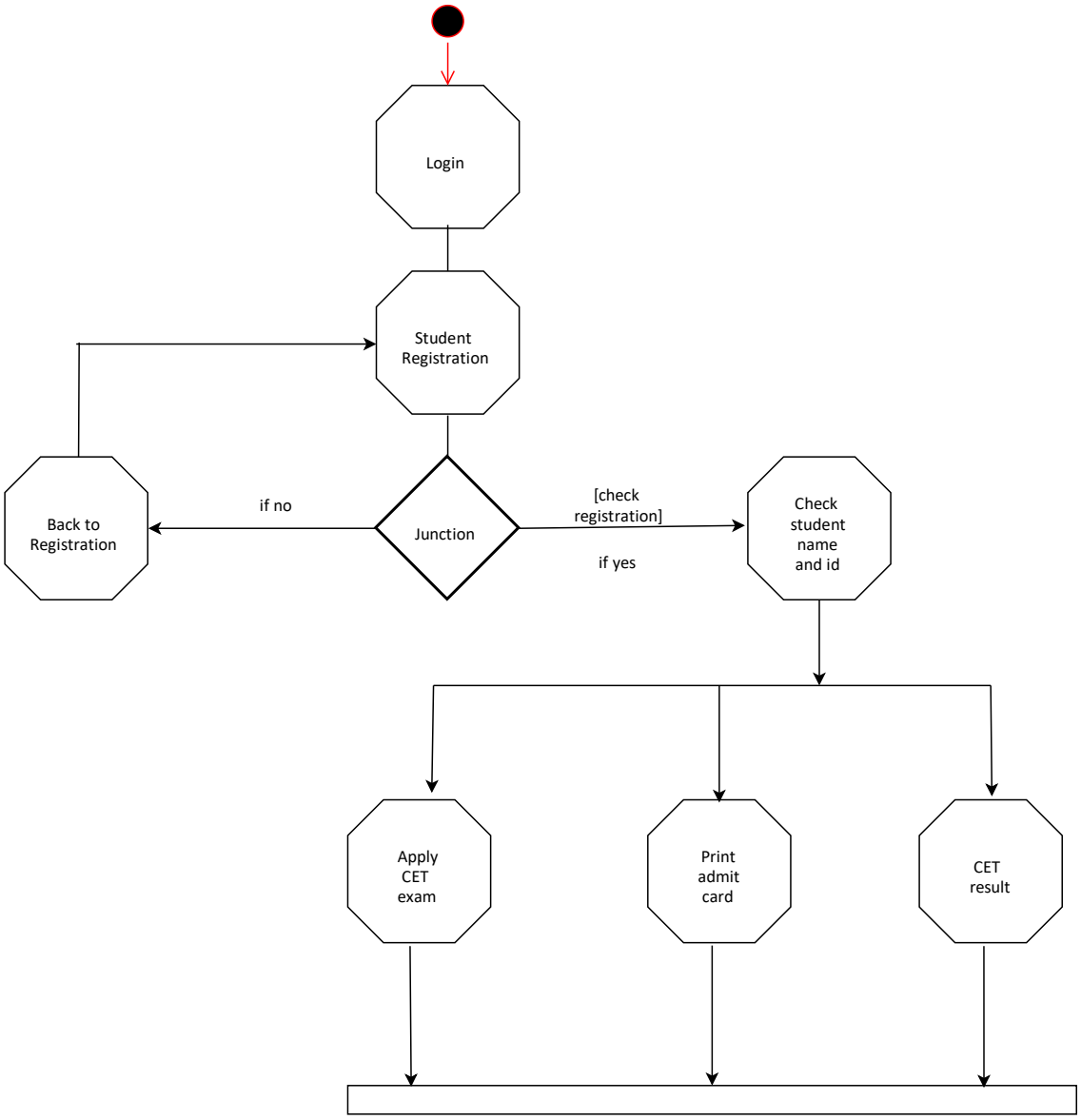


Fork

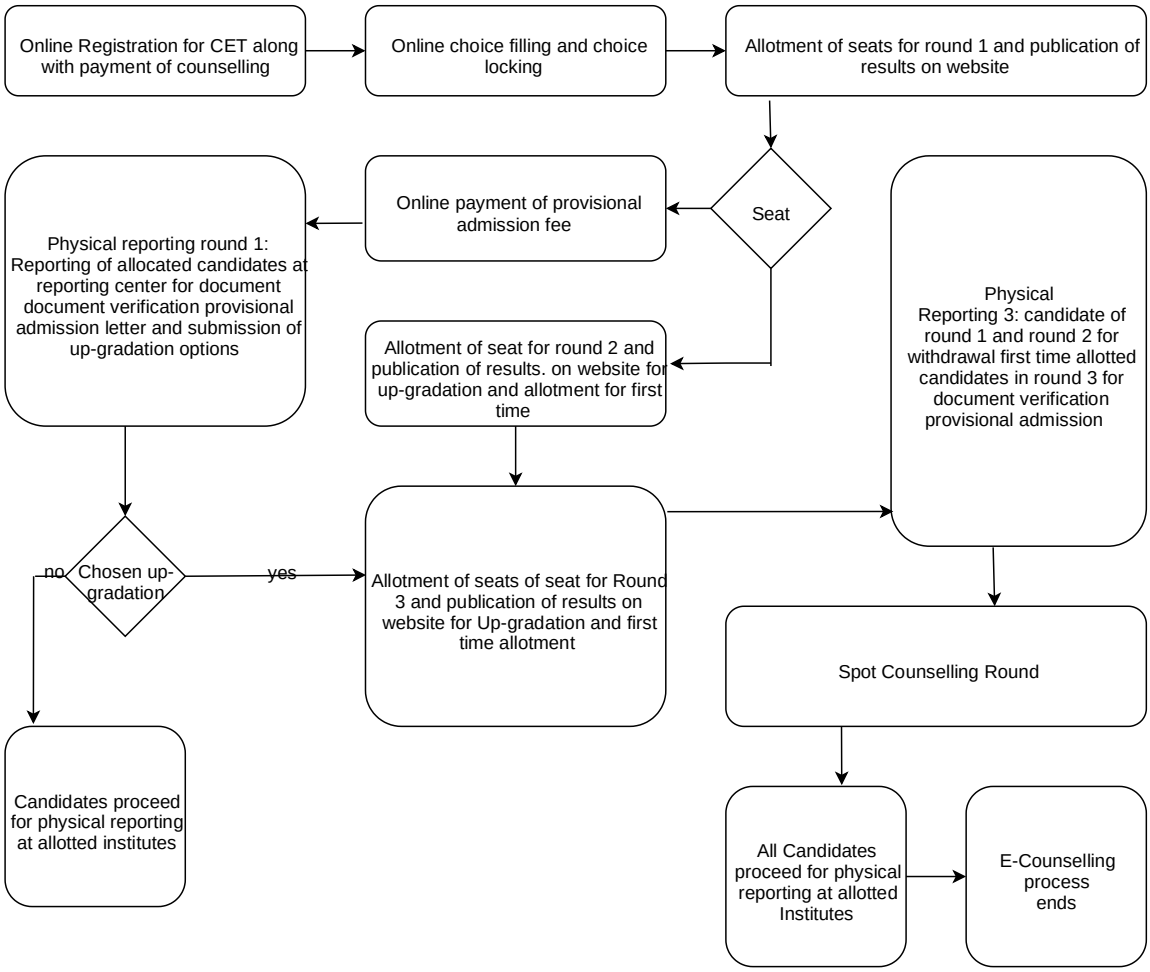


Join

21. State chart diagram of CET class with respect to conduct



22. state chart diagram of student class CET activity with respect to counselling



23. State chart diagram of student class with respect to End-Term Examination