T K R COLLEGE OF ENGINEERING & TECHNOLOGY

(Autonomous)

B.TECH COMPUTER SCIENCE AND ENGINEERING

OBJECT ORIENTED ANALYSIS AND DESIGN –(B56PC9)

B.Tech VI Sem	**L/T/P/C**
	0/0/2/1

COURSE OBJECTIVE:

Write programs for solving real world problems using java language.

COURSE OUTCOMES:

After learning the contents of this course, the student must be able to

1. Write programs for solving real world problems using java collection frame work L3
2. Write programs using abstract classes.L3
3. Design and develop programs using objects and inheritance in Java language.L5
4. Write multithreaded programs.L3

List of programs:

Consider the following three case studies:

Online course reservation system, E-ticketing, Library Management System

Week 1

Familiarization with Rational Rose or Umbrello For each case study develops a problem statement.

Week 2, 3 & 4:

For each case study:

1) Identify and analyze events

2) Identify Use cases

3) Develop event table

4) Identify & analyze domain classes

5) Represent use cases and a domain class diagram using Rational Rose

6) Develop CRUD matrix to represent relationships between use cases and problem domain Classes

Week 5 & 6:

For each case study:

1) Develop Use case diagrams

2) Develop elaborate Use case descriptions & scenarios

3) Develop prototypes (without functionality)

4) Develop system sequence diagrams

Week 7, 8, 9 & 10:

For each case study:

1) Develop high-level sequence diagrams for each use case

2) Identify MVC classes / objects for each use case

3) Develop Detailed Sequence Diagrams / Communication diagrams for each use case showing Interactions among all the three-layer objects

4) Develop detailed design class model (use GRASP patterns for responsibility assignment)

5) Develop three-layer package diagrams for each case study

Week 11 & 12:

For each case study:

1) Develop Use case Packages

2) Develop component diagrams

3) Identify relationships between use cases and represent them

4) Refine domain class model by showing all the associations among classes

Week 13 onwards:

For each case study:

1) Develop sample diagrams for other UML diagrams – state chart diagrams, activity diagrams and deployment diagrams.

WEEK-1

1. Familiarization with Rational Rose or Umbrello For each case study develops a problem statement.

Umbrello UML Modeler app and code generator

Umbrello is a free and open source Unified Modeling Language modeler and code generator application for Linux, MacOS, and Windows. With UML you can create diagrams of software and other systems in a standard format to document or design the structure of your programs. It is released under the GNU General Public License v2.0 or later and a kde project. Supports XMI formats and use case, class, sequence, communication, state, activity, component, implementation, and entity relationship diagrams.

Install Umbrella on Ubuntu Linux

Umbrella is available as an out-of-the-box package for Ubuntu Linux. You can install it through the terminal app. Open the terminal app with ctrl+alt+t keys and run below umbrello instant install command.

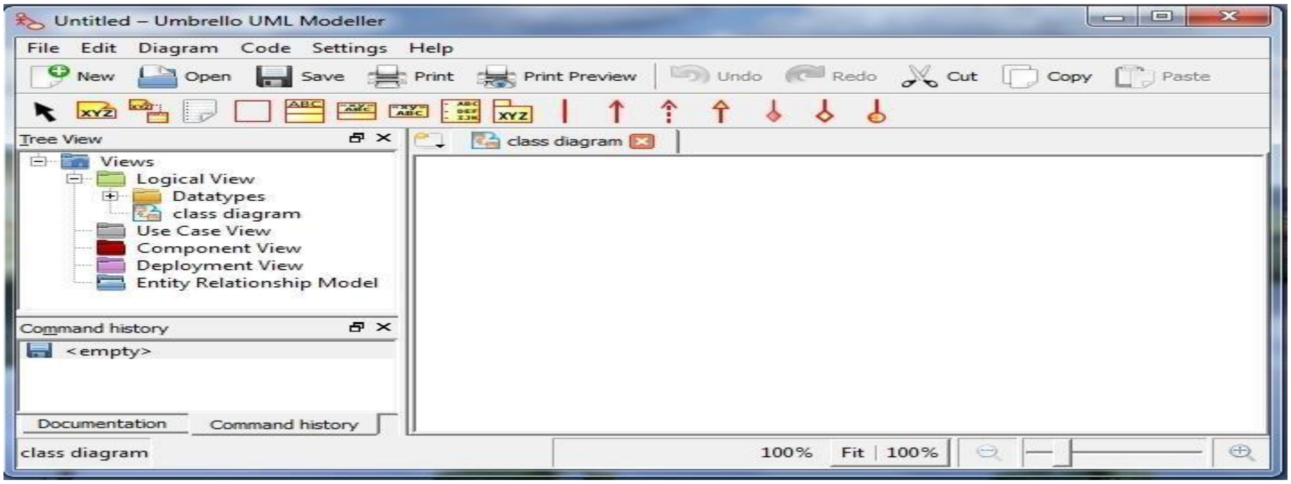
```
sudo snap install umbrello
```

After installation, you can open it through the application menu or by using the followingcommand in terminal.

```
umbrello
```

If you don't need it, uninstall it using the following command.

```
sudo snap remove umbrello
```



How to Install Umbrello (UML Modeler) On Windows 7

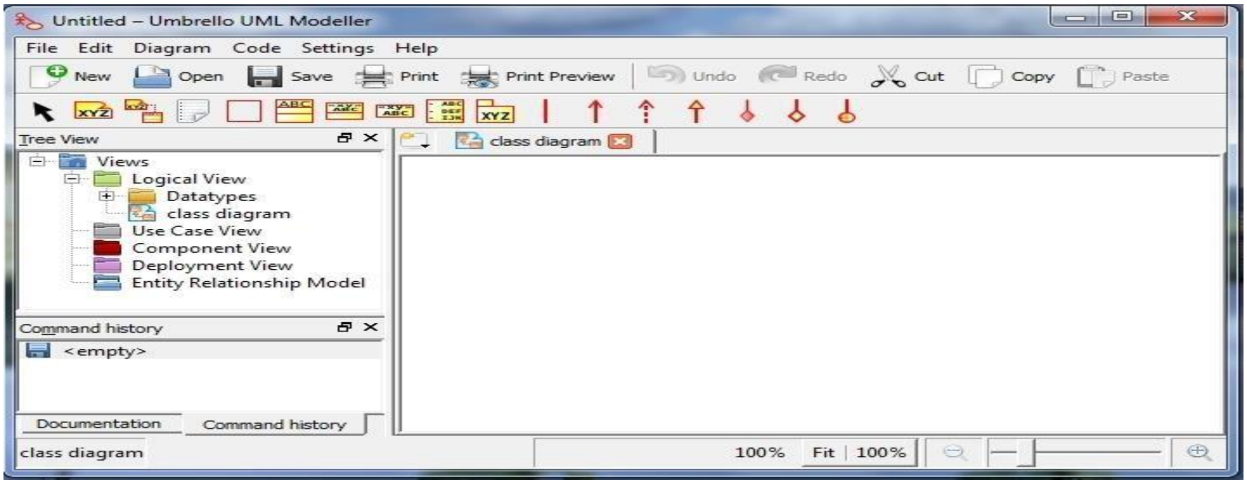
Introduction

Umbrello (UML Modeler):

Umbrello UML Modeler is a free software UML diagram tool available natively for Unix- like platforms, as well as Microsoft Windows (as part of KDE-Windows). It is part of the KDE SC 4 but works well with other desktops and programming environments. Umbrello is apretty simple UML editor though it does support a range of different views and diagrams. It makes it as easy as possible to draw a simple diagram, then lets you alter and add to it as yourdesign is refined. Everything seems to work fine under Linux with the exception of exporting to an image.

KDE For Windows

KDE for Windows is an ongoing project to port the KDE applications to MS Windows. Currently supported versions of Windows are XP, Vista and 7

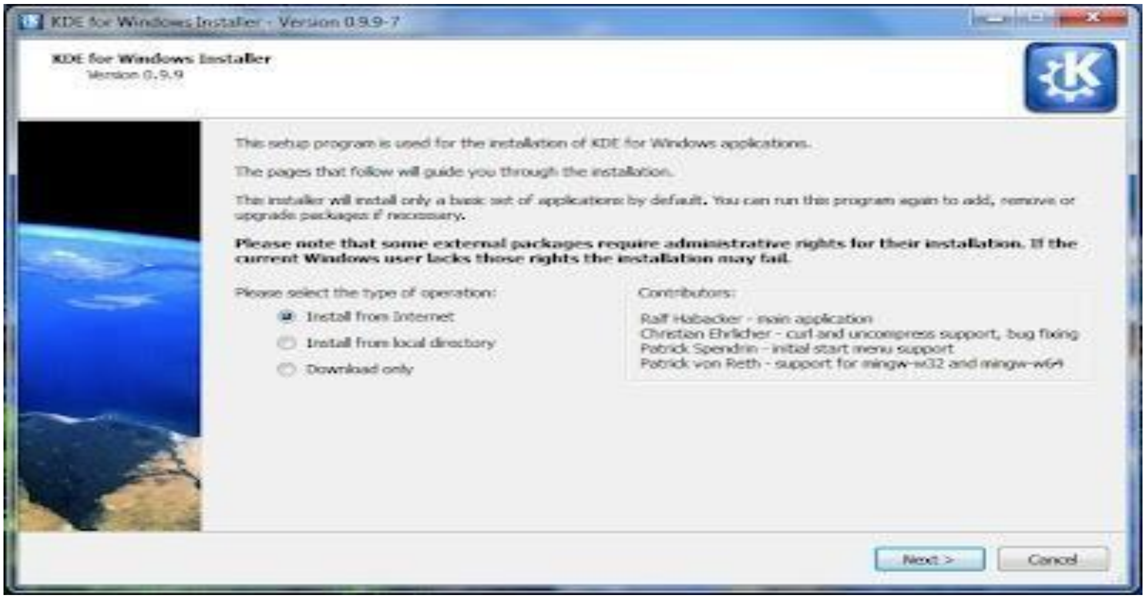


2 Requirement:

1. KDE For Windows
   a. Download : http://windows.kde.org/
2. Umbrello
   a. Download and installation with KDE For Windows

3 Installing KDE for Windows and Umbrello

1. Double click KDE installer (*.exe file, example : kdewin-installer-gui-latest.exe), Youwill see a window that looks like this



KDE for Windows Installer - Version 0.9.9-7

KDE for Windows Installer
Version 0.9.9

This setup program is used for the installation of KDE for Windows applications.

The pages that follow will guide you through the installation.

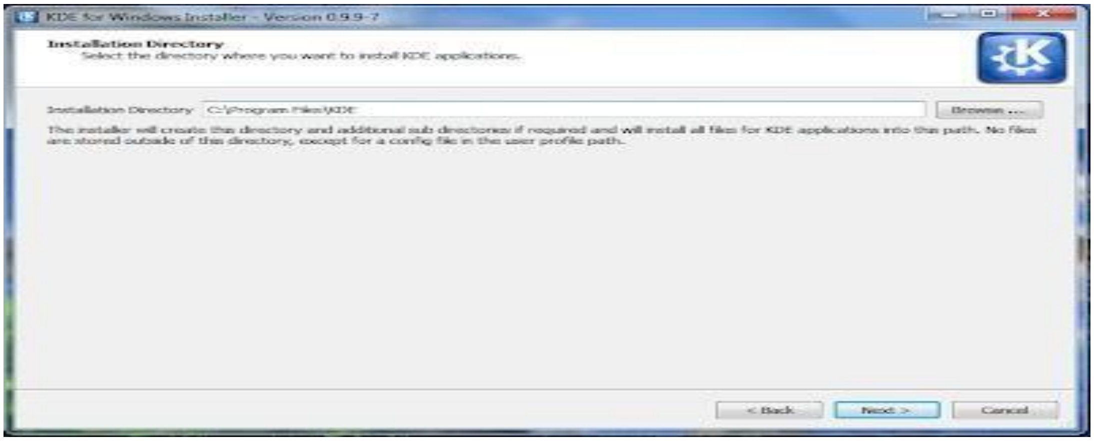
This installer will install only a basic set of applications by default. You can run this program again to add, remove or upgrade packages if necessary.

Please note that some external packages require administrative rights for their installation. If the current Windows user lacks those rights the installation may fail.

Please select the type of operation:
● Install from Internet
○ Install from local directory
○ Download only

Contributors:
Ralf Habacker - main application
Christian Ehrlicher - curl and uncompress support, bug fixing
Patrick Spendrin - initial start menu support
Patrick von Reth - support for mingw-w32 and mingw-w64

2. Choose “Install from Internet” for type operation, and then click the “Next >” button atthe bottom of the screen.
3. On the window “Installation Directory”, click the “Browse …” button to change defaultdirectory,and then click the “Next >” button.



KDE for Windows Installer - Version 0.9.9-7

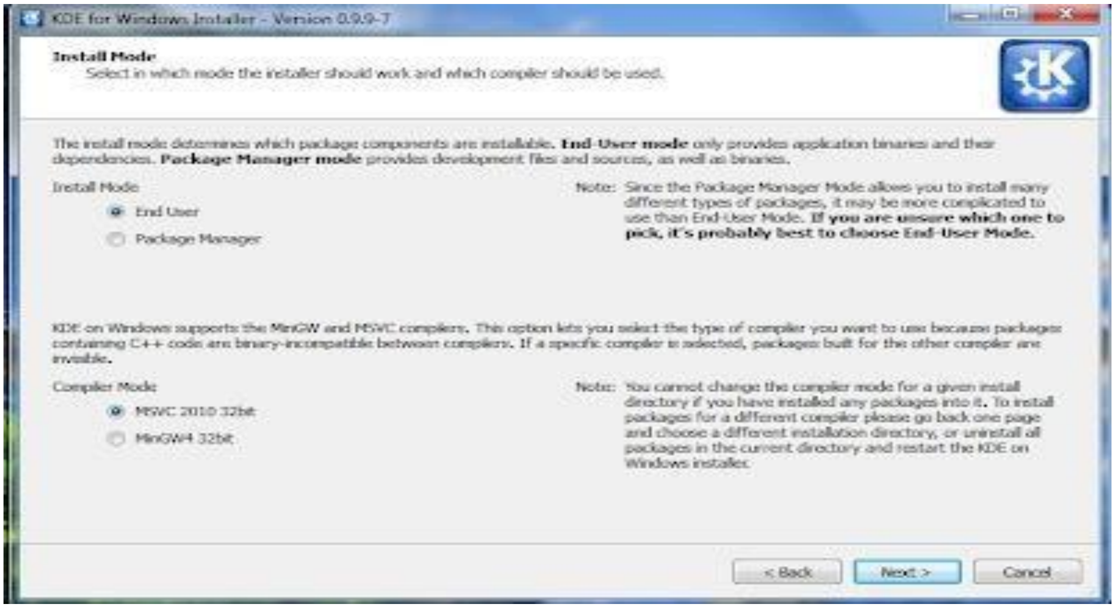
Installation Directory
Select the directory where you want to install KDE applications.

Installation Directory: C:\Program Files\KDE

The installer will create this directory and additional sub directories if required and will install all files for KDE applications into this path. No files are stored outside of this directory, except for a config file in the user profile path.

4. On the window “Install Mode” :
   1. Click the radio button “End User” on dialog Install Mode,
   2. Click the radio button “MSVC 2010 32bit” on dialog Compiler

   Mode,and then click the “Next >” button.



KDE for Windows Installer - Version 0.9.9-7

Install Mode

Select in which mode the installer should work and which compiler should be used.

The install mode determines which package components are installable. **End-User mode** only provides application binaries and their dependencies. **Package Manager mode** provides development files and sources, as well as binaries.

Install Mode

● End User

○ Package Manager

Note: Since the Package Manager Mode allows you to install many different types of packages, it may be more complicated to use than End-User Mode. **If you are unsure which one to pick, it's probably best to choose End-User Mode.**

KDE on Windows supports the MinGW and MSVC compilers. This option lets you select the type of compiler you want to use because packages containing C++ code are binary-incompatible between compilers. If a specific compiler is selected, packages built for the other compiler are invisible.

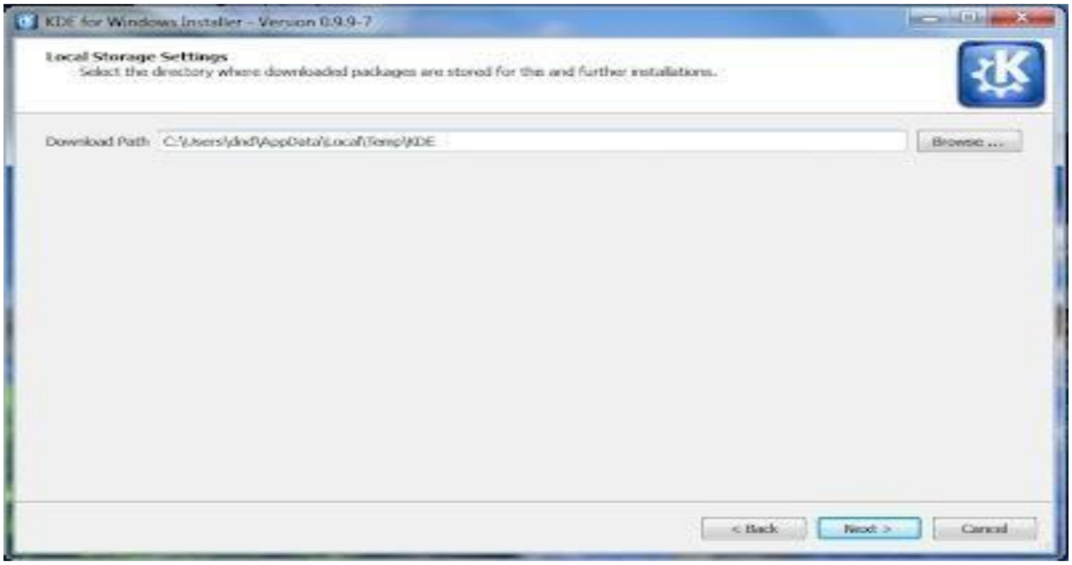
Compiler Mode

● MSVC 2010 32bit

○ MinGW4 32bit

Note: You cannot change the compiler mode for a given install directory if you have installed any packages into it. To install packages for a different compiler please go back one page and choose a different installation directory, or uninstall all packages in the current directory and restart the KDE on Windows installer.

5. On the window “Local Storage Setting”, click the “Browse …” button to change defaultdirectory,and then click the “Next >” button.



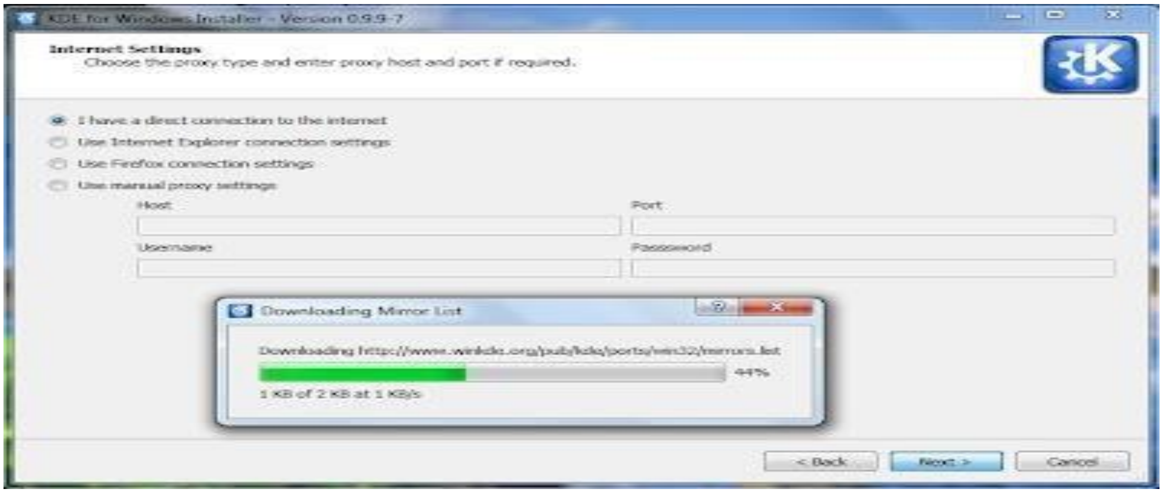
KDE for Windows Installer - Version 0.9.9-7

Local Storage Settings

Select the directory where downloaded packages are stored for this and further installations.

Download Path: C:\Users\dnd\AppData\Local\Temp\KDE

6. On the window “Internet Setting”, click the radio button “I have a direct connection to theinternet”, and then click the “Next >” button.



KDE for Windows Installer - Version 0.9.9-7

Internet Settings
Choose the proxy type and enter proxy host and port if required.

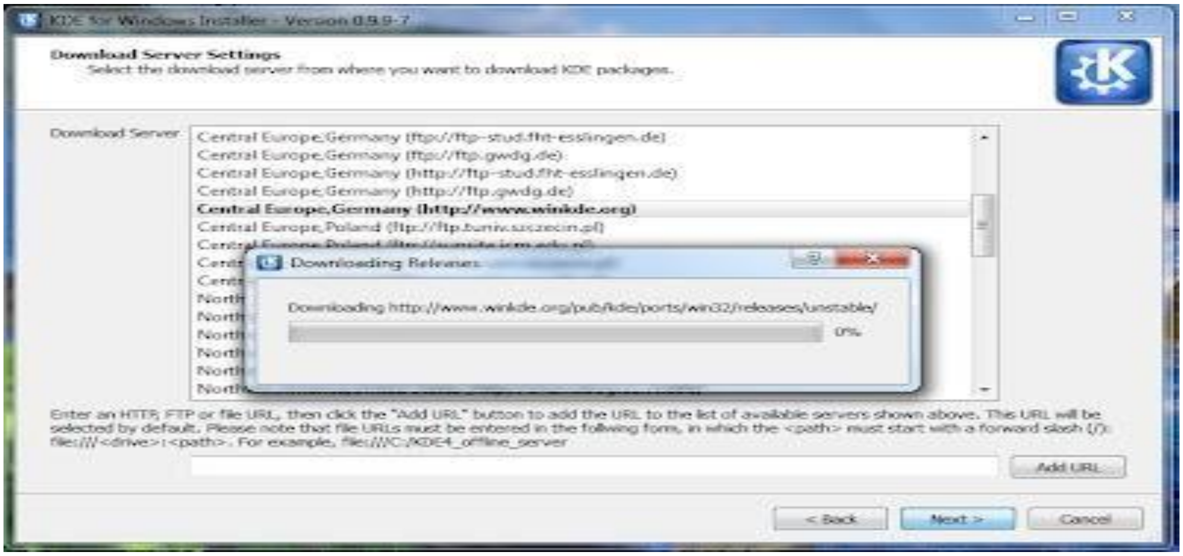
● I have a direct connection to the internet
○ Use Internet Explorer connection settings
○ Use Firefox connection settings
○ Use manual proxy settings

Host:
Port:
Username:
Password:

Downloading Mirror List

Downloading http://www.winkde.org/pub/kde/ports/win32/mirrors.list
44%
1 KB of 2 KB at 1 KB/s

7. In the window “Download Server Setting”, Choose “Central Europe, Germany (http://www.winkde.org)” to install the application,and then click the “Next” button.
8. The next step wait for download Release.



KDE for Windows Installer - Version 0.9.9-7

Download Server Settings
Select the download server from where you want to download KDE packages.

Download Server
Central Europe,Germany (ftp://ftp-stud.fht-esslingen.de)
Central Europe,Germany (ftp://ftp.gwdg.de)
Central Europe,Germany (http://ftp-stud.fht-esslingen.de)
Central Europe,Germany (http://ftp.gwdg.de)
Central Europe,Germany (http://www.winkde.org)
Central Europe,Poland (ftp://ftp.tuniv.szczecin.pl)

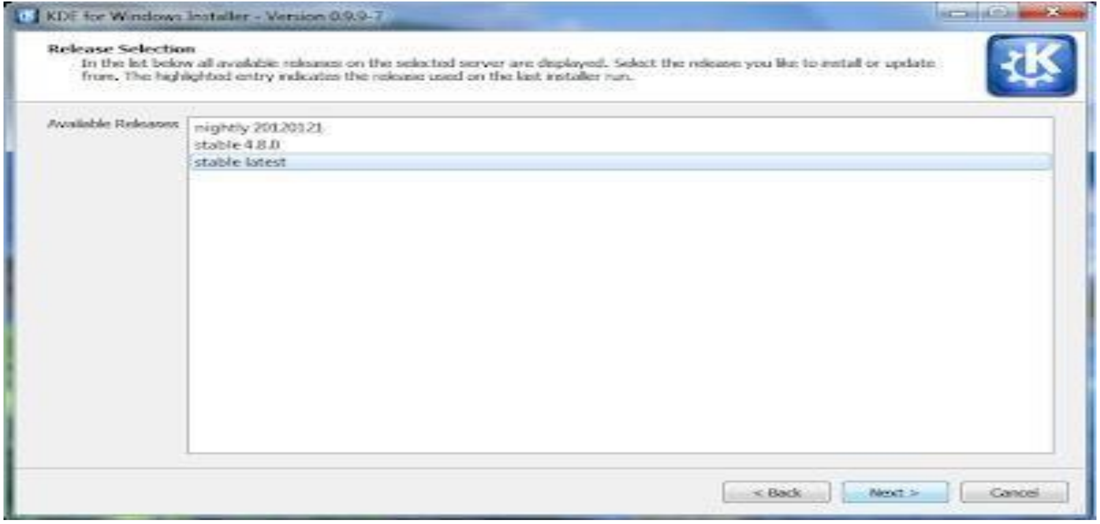
Downloading Releases

Downloading http://www.winkde.org/pub/kde/ports/win32/releases/unstable/
0%

Enter an HTTP, FTP or file URL, then click the "Add URL" button to add the URL to the list of available servers shown above. This URL will be selected by default. Please note that file URLs must be entered in the follwing form, in which the <path> must start with a forward slash (/): file:///<drive>:<path>. For example, file:///C:/KDE4_offline_server

9. The next step, On the window “Release Selection”, choose Stable Release for AvailableRelease, an then click the “Next >” button.

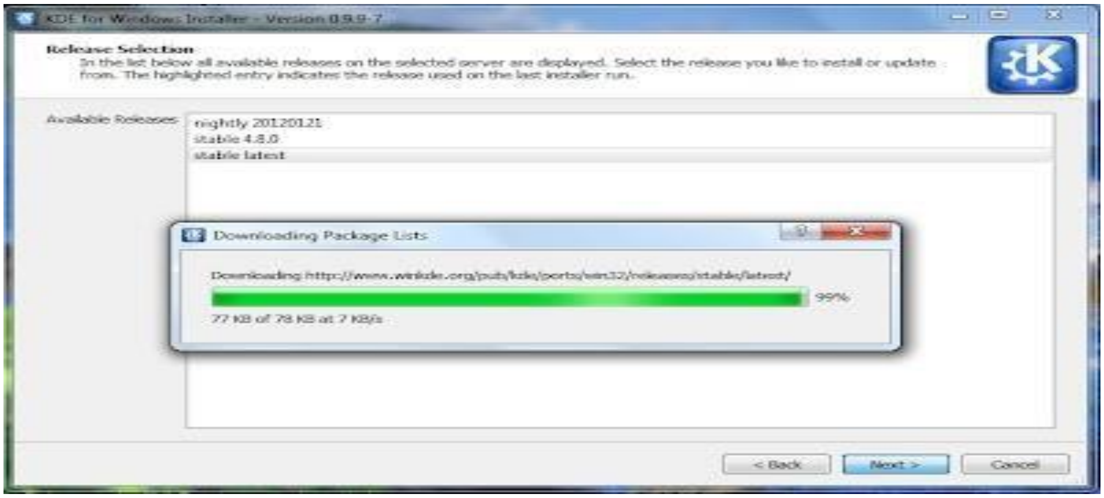
10. The next step waits for download Package List.



Release Selection
In the list below all available releases on the selected server are displayed. Select the release you like to install or update from. The highlighted entry indicates the release used on the last installer run.

Available Releases:
nightly 20120121
stable 4.8.0
stable latest

11. In the window "Package Selection", choose "Umbrello" to install the application, and then click the "Next >" button.

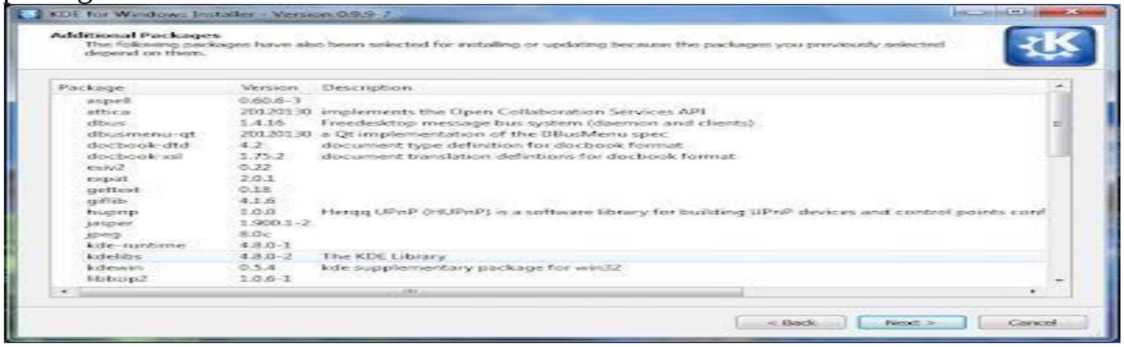

KDE for Windows Installer - Version 0.9.9-7
Downloading Package Lists
< Back
Next >
Cancel

Release Selection
In the list below all available releases on the selected server are displayed. Select the release you like to install or update from. The highlighted entry indicates the release used on the last installer run.

Available Releases:
nightly 20120121
stable 4.8.0
stable latest

Downloading http://www.winkde.org/pub/kde/ports/win32/releases/stable/latest/
99%
77 KB of 78 KB at 7 KB/s

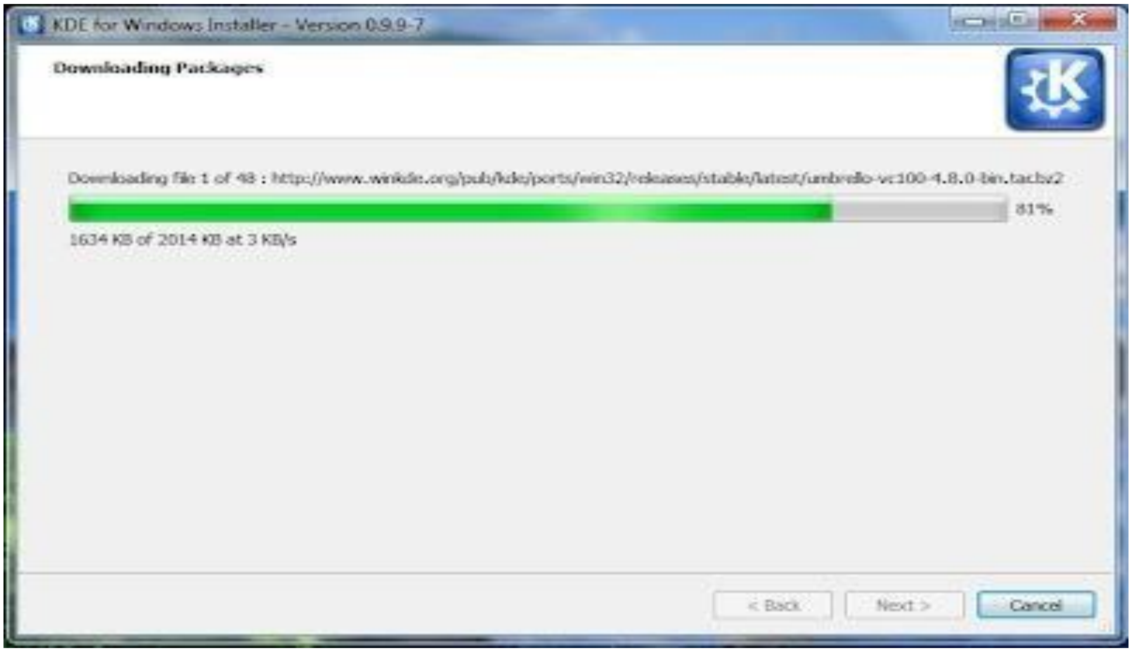
12. The next step, you will get a confirmation on the package will be downloaded, click the"Next >" button to downloadand install packages.



Additional Packages
The following packages have also been selected for installing or updating because the packages you previously selected depend on them.

Package	Version	Description
aspell	0.60.6-3	
attica	20120130	implements the Open Collaboration Services API
dbus	1.4.16	Freedesktop message bus system (daemon and clients)
dbusmenu-qt	20120130	a Qt implementation of the DBusMenu spec
docbook-dtd	4.2	document type definition for docbook format
docbook-xsl	1.75.2	document translation defintions for docbook format
exiv2	0.22	
expat	2.0.1	
gettext	0.18	
giflib	4.1.6	
hupnp	1.0.0	Herqq UPnP (HUPnP) is a software library for building UPnP devices and control points conf
jasper	1.900.1-2	
jpeg	8.0c	
kde-runtime	4.8.0-1	
kdelibs	4.8.0-2	The KDE Library
kdewin	0.5.4	kde supplementary package for win32
libbzip2	1.0.6-1	

13. The next step waits for download Packages and the finished installation.



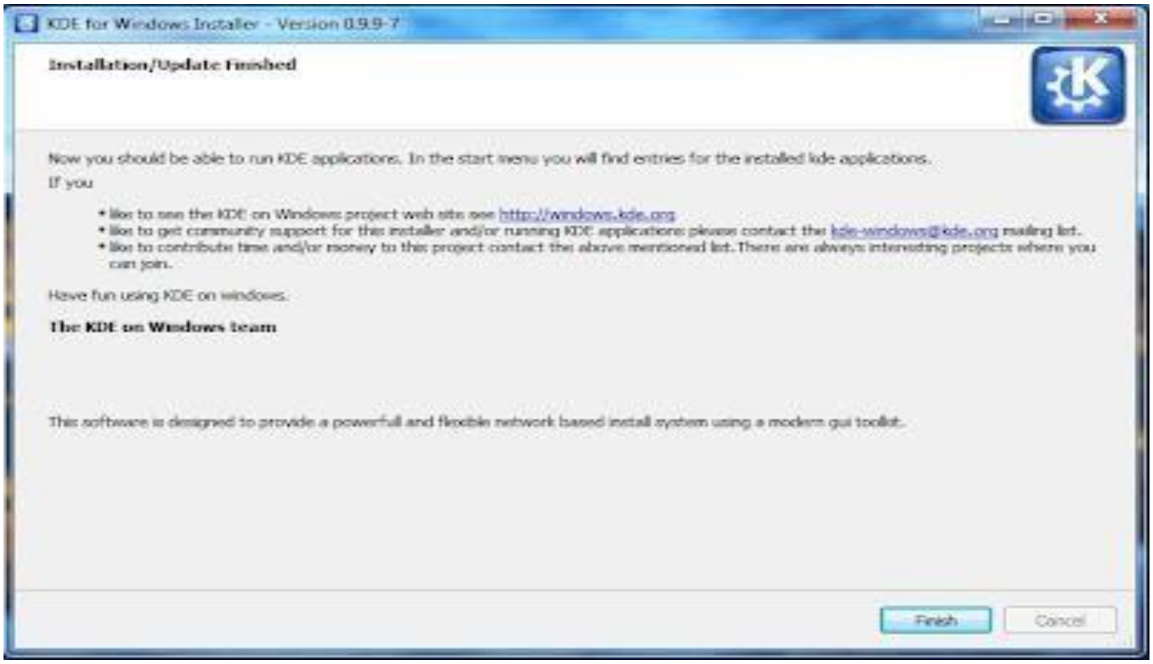
Downloading Packages

Downloading file 1 of 48 : http://www.winkde.org/pub/kde/ports/win32/releases/stable/latest/umbrello-vc100-4.8.0-bin.tar.bz2

81%

1634 KB of 2014 KB at 3 KB/s

14. To complete the installation process clicks the "Finish" button. You should see a windowthat looks like this :



Installation/Update Finished

Now you should be able to run KDE applications. In the start menu you will find entries for the installed kde applications.

If you

- like to see the KDE on Windows project web site see http://windows.kde.org
- like to get community support for this installer and/or running KDE applications please contact the kde-windows@kde.org mailing list.
- like to contribute time and/or money to this project contact the above mentioned list. There are always interesting projects where you can join.

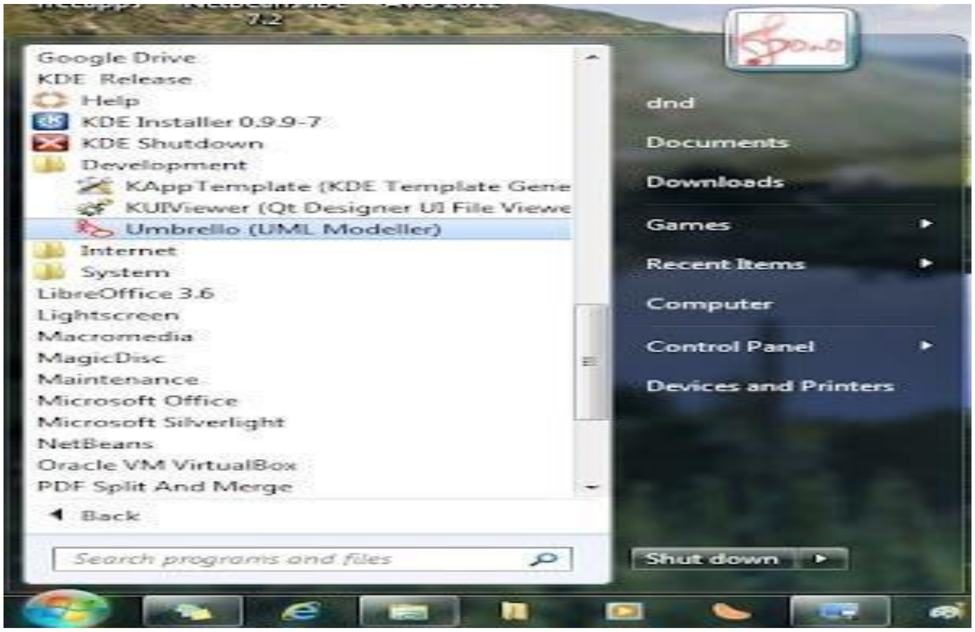
Have fun using KDE on windows.

The KDE on Windows team

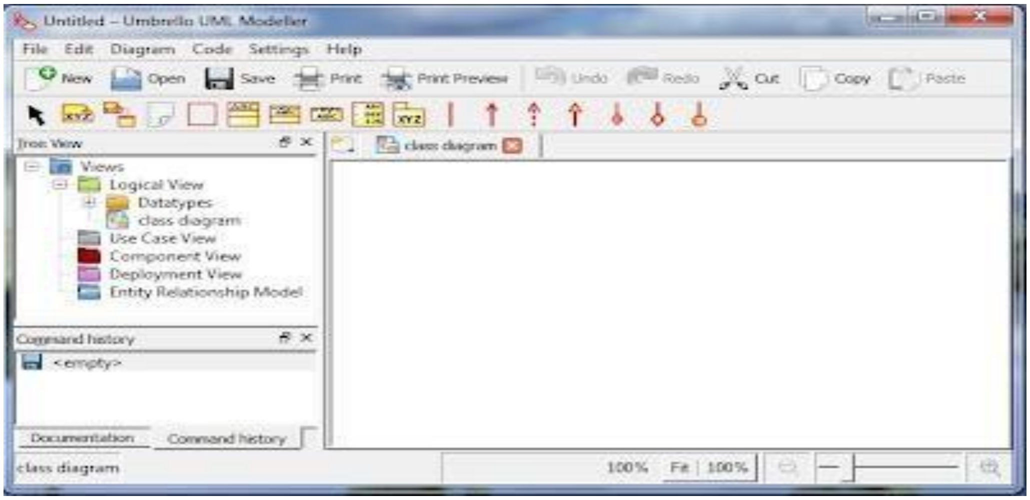
This software is designed to provide a powerfull and flexible network based install system using a modern gui toolkit.

4 Running Umbrello (UML Modeler)

1. To open Umbrello, Go to Start Menu à All Programs à KDE Release a Developmentà Umbrello (UML Modeler).



1. You will see a window that looks like this :



CASE STUDY: ONLINE COURSE RESERVATION

Week 2, 3 & 4:

For each case study:

1) Identify and analyze events

2) Identify Use cases

3) Develop event table

4) Identify & analyze domain classes

5) Represent use cases and a domain class diagram using Rational Rose

6) Develop CRUD matrix to represent relationships between use cases and problem domain Classes

Week 5 & 6:

For each case study:

1) Develop Use case diagrams

2) Develop elaborate Use case descriptions & scenarios

3) Develop prototypes (without functionality)

4) Develop system sequence diagrams

Week 7, 8, 9 & 10:

For each case study:

1) Develop high-level sequence diagrams for each use case

2) Identify MVC classes / objects for each use case

3) Develop Detailed Sequence Diagrams / Communication diagrams for each use case showing Interactions among all the three-layer objects

4) Develop detailed design class model (use GRASP patterns for responsibility assignment)

5) Develop three-layer package diagrams for each case study

Week 11 & 12:

For each case study:

1) Develop Use case Packages

2) Develop component diagrams

3) Identify relationships between use cases and represent them

4) Refine domain class model by showing all the associations among classes

Week 13 onwards:

For each case study:

1) Develop sample diagrams for other UML diagrams – state chart diagrams, activity diagrams and deployment diagrams.

Problem statements of Online course reservation system

AIM:

To create a system through which students can register to the courses desired by them.

PROBLEM STATEMENT:

- The system is built to be used by students and managed by an administrator.
- The student and employee have to login to the system before any processing can be done.
- The student can see the courses available to him/her and register to the course he/she wants.
- The administrator can maintain the course details and view all the students who have registered to any course.
-

Week 2, 3&4

Case Study 1: Online course reservation system

a) Identify the users and Analyze Events
b) Identify Use cases
c) Develop event table
d) Identify & analyze domain classes
e) Represent Use cases & Domain class diagram using Umbrello
f) Develop CRUD matrix to represent relationships between use cases and problem domain classes.
a) Identify and Analyze Events

USERS	EVENTS
Registrar/Admin	1. Refers to the super user with the privilege to manage the entire system. 2. He has the certain privileges to add the course status and to approve the issue of course. 3. He may contain a group of persons under him to verify the documents and give suggestion whether or not to approve the dispatch of course.
Applicant	1. One who wishes to register the Course 2. Submit the information to the database.
OCRS	Secure Reservation of information by the Students. • SMS and Mail updates to the students by the Registrar • Registrar can generate reports from the information and is the only authorized personnel to add the eligible application information to the database.

b) Identify Use cases

- A use case diagram shows a set of use cases and actors (a special kind of class) and their relationships. Use case diagrams address the static use case view of a system.
- These diagrams are especially important in organizing and modeling the behaviors of a system.

The following are the use cases:

1. Login

2. View course details

3. Reserve for course

4. Pay fee

5. Check status

ACTORS INVOLVED:

1. Student

2. Registrar

c) Develop event table

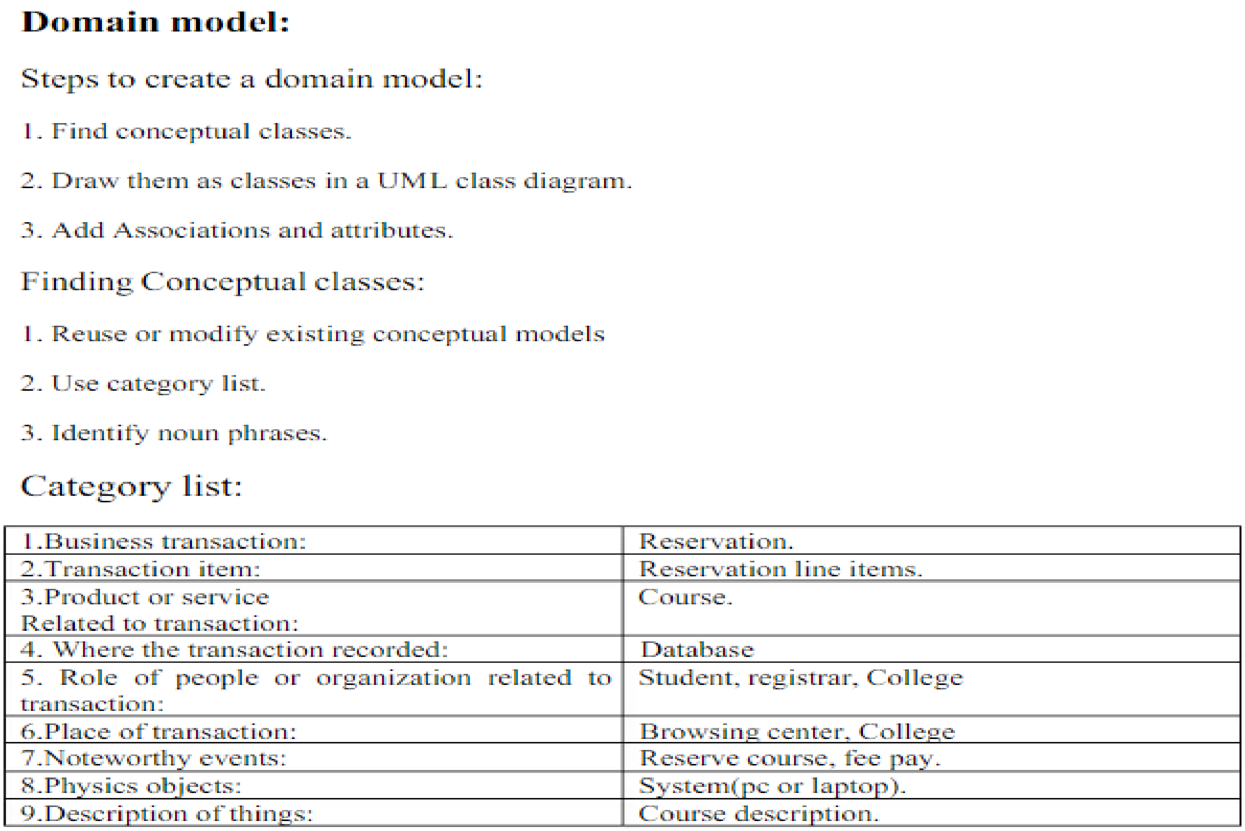
	Login	View course details	Reserve for course	Storing details in database	Pay fee	Check status
User enters username and password	*					
Checks whether the username and password entered is valid or not	*					
Checks whether user is student or registrar	*					
Students can search all the details available and choose the best course they want	*	*				
Students can view the course duration, faculty and department of the course he may choose	*	*				
Registering for the course			+			
Upon successful registration	*	*	*		+	

have to pay storing the student details in database						
after registration to course student may see the details of current course and pay the fees	*		+		*	
The student tries to check the status in which category they applied. The system displays the status information to the student.	*	*				*

* Event can occur zero to many times

+ Event can occur zero to one time

d) Identify & analyze domain classes



Domain model:

Steps to create a domain model:

1. Find conceptual classes.

2. Draw them as classes in a UML class diagram.

3. Add Associations and attributes.

Finding Conceptual classes:

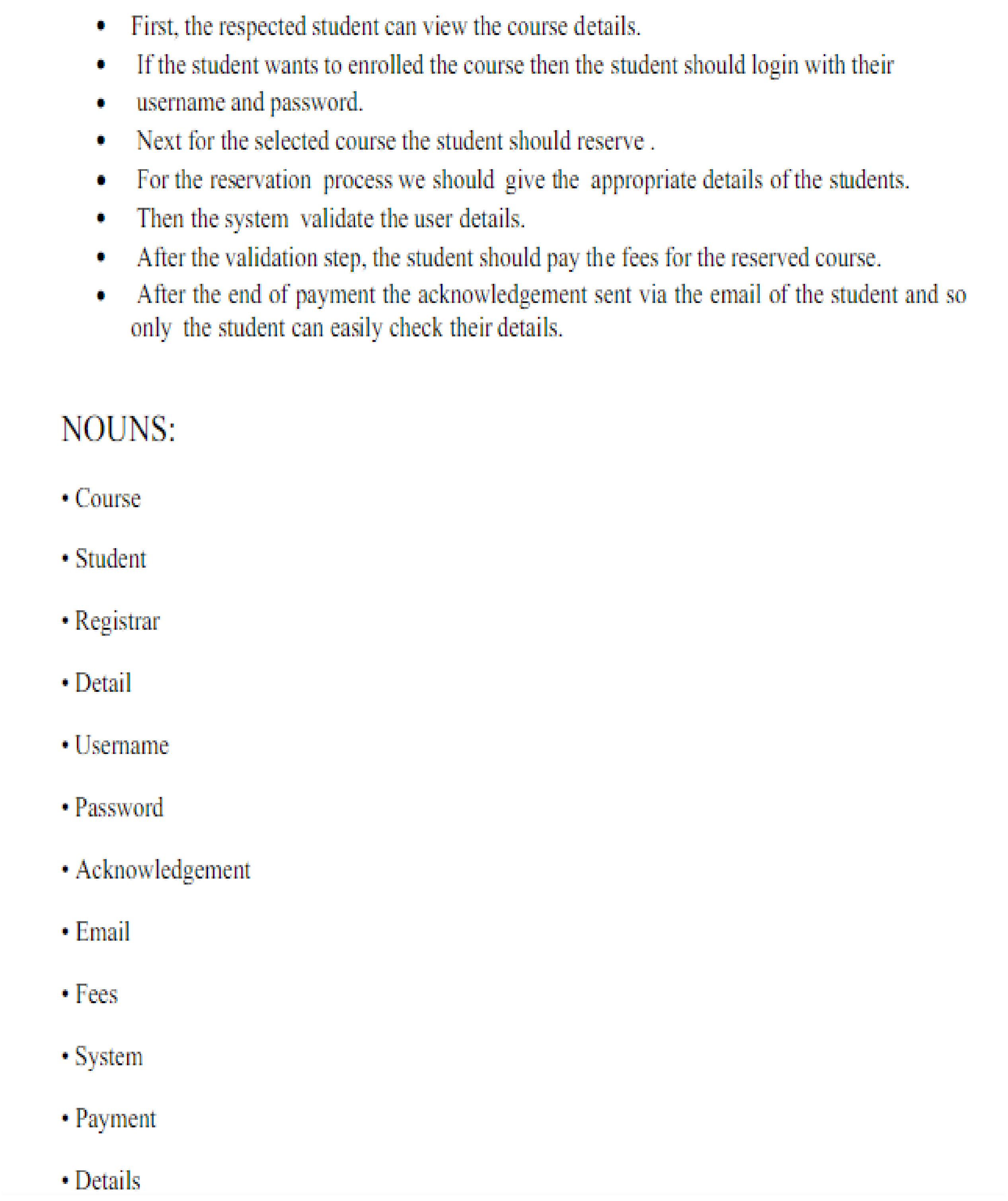
1. Reuse or modify existing conceptual models

2. Use category list.

3. Identify noun phrases.

Category list:

1.Business transaction:	Reservation.
2.Transaction item:	Reservation line items.
3.Product or service Related to transaction:	Course.
4. Where the transaction recorded:	Database
5. Role of people or organization related to transaction:	Student, registrar, College
6.Place of transaction:	Browsing center, College
7.Noteworthy events:	Reserve course, fee pay.
8.Physics objects:	System(pc or laptop).
9.Description of things:	Course description.

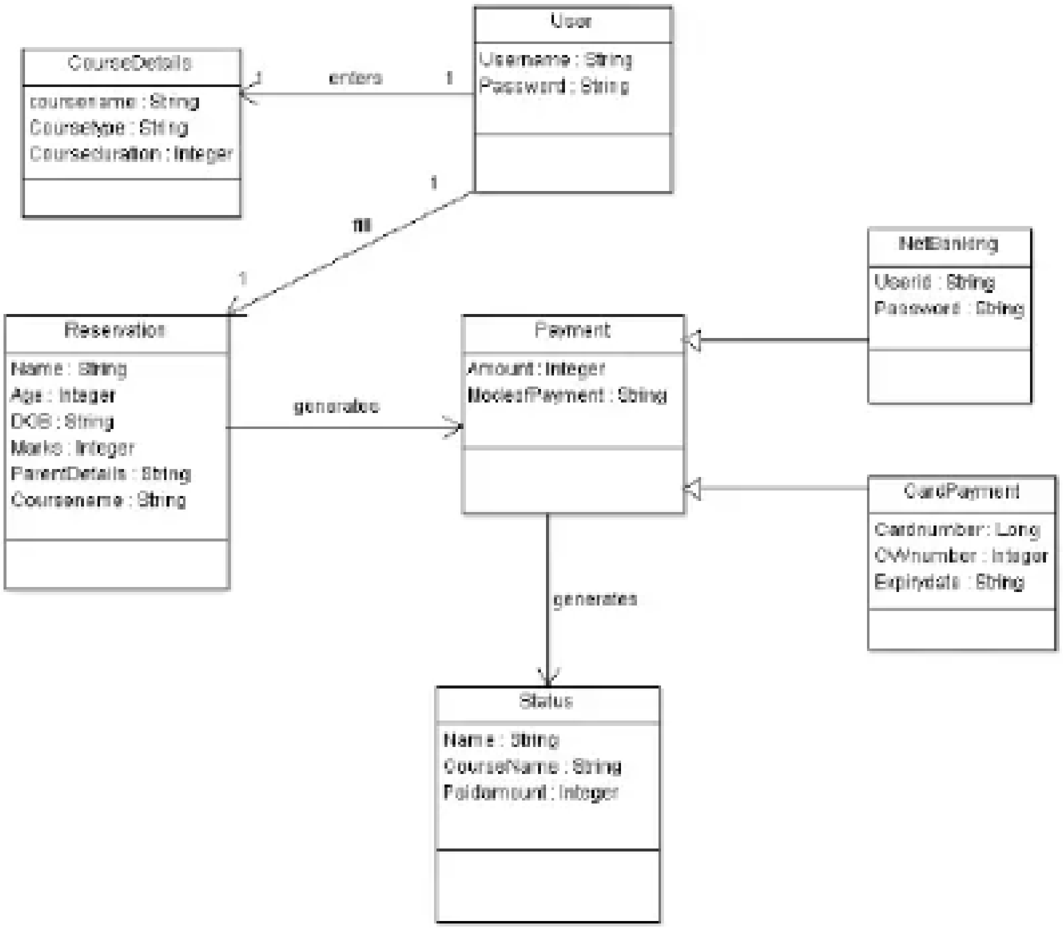


- First, the respected student can view the course details.
- If the student wants to enrolled the course then the student should login with their
- username and password.
- Next for the selected course the student should reserve .
- For the reservation process we should give the appropriate details of the students.
- Then the system validate the user details.
- After the validation step, the student should pay the fees for the reserved course.
- After the end of payment the acknowledgement sent via the email of the student and so only the student can easily check their details.

NOUNS:

- Course
- Student
- Registrar
- Detail
- Username
- Password
- Acknowledgement
- Email
- Fees
- System
- Payment
- Details

Domain Model:



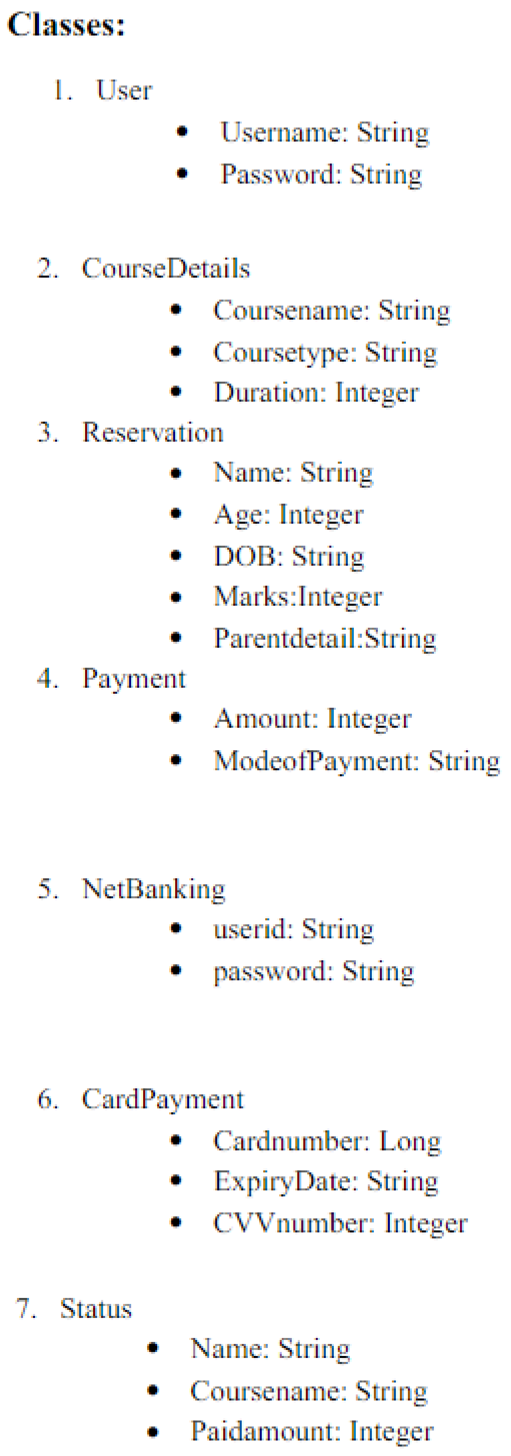
e) Represent Use cases & Domain class diagram using Umbrello

Student

They are the person who desires to obtain the course and submit the information to the database.

Registrar

He has the certain privileges to add the course status and to approve the issue of course. He may contain a group of persons under him to verify the documents and give suggestion whether or not to approve the dispatch of course.



Classes:

1. User
   - Username: String
   - Password: String

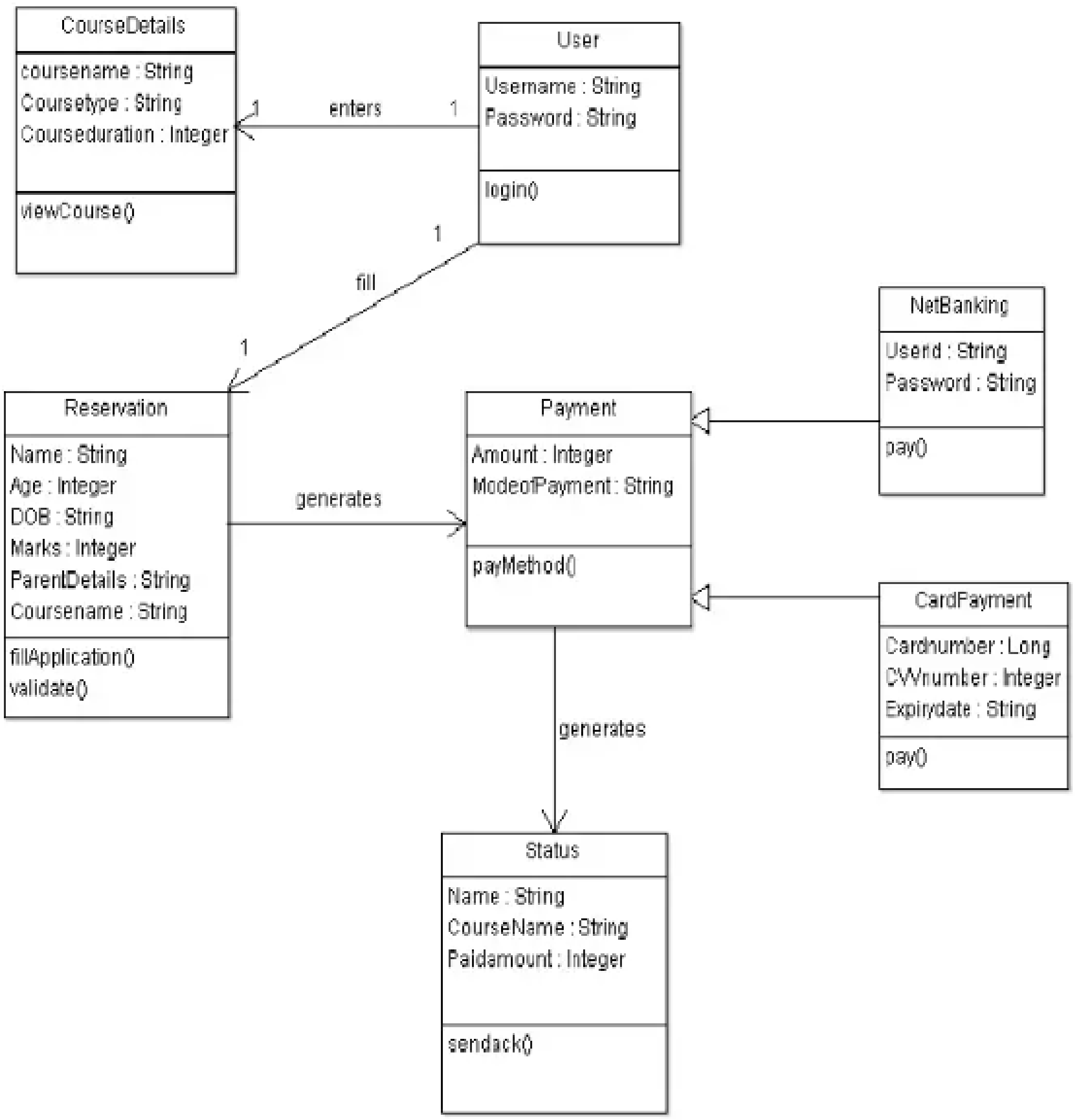
2. CourseDetails
   - Coursename: String
   - Coursetype: String
   - Duration: Integer
3. Reservation
   - Name: String
   - Age: Integer
   - DOB: String
   - Marks:Integer
   - Parentdetail:String
4. Payment
   - Amount: Integer
   - ModeofPayment: String

5. NetBanking
   - userid: String
   - password: String

6. CardPayment
   - Cardnumber: Long
   - ExpiryDate: String
   - CVVnumber: Integer

7. Status
   - Name: String
   - Coursename: String
   - Paidamount: Integer

CLASS DIAGRAM



f) Develop CRUD matrix to represent relationships between use cases and problem domain classes

CRUD technique – an acronym for Create, Read/Report, Update, and Delete; a technique to validate or refine use cases

- The CRUD Matrix is an excellent technique to identify the Tables in a Database which are used in any User interaction with a Web Site.
- CRUD means 'Create, Read, Update or Delete', and the CRUDD Matrix identifies the Tables involved in any CRUD operation.
- It is very valuable to combine a CRUD Matrix with the analysis of possible User Scenarios for the Web Site
- The analysis helps to identify any Tables which are not used, and any Tables which are used heavily, and may therefore be a performance bottleneck.

Use case/Class	**Student**	**Course**	**College**
Login/Register	**C**		
View Course Details	**R**	**C**	
Reserve for Course	**R**	**U**	**U**
Pay fee	**U**	**R**	**R**
Check Status	**R**	**R**	**R**

Week-5 & 6

a) Develop Use Case diagrams

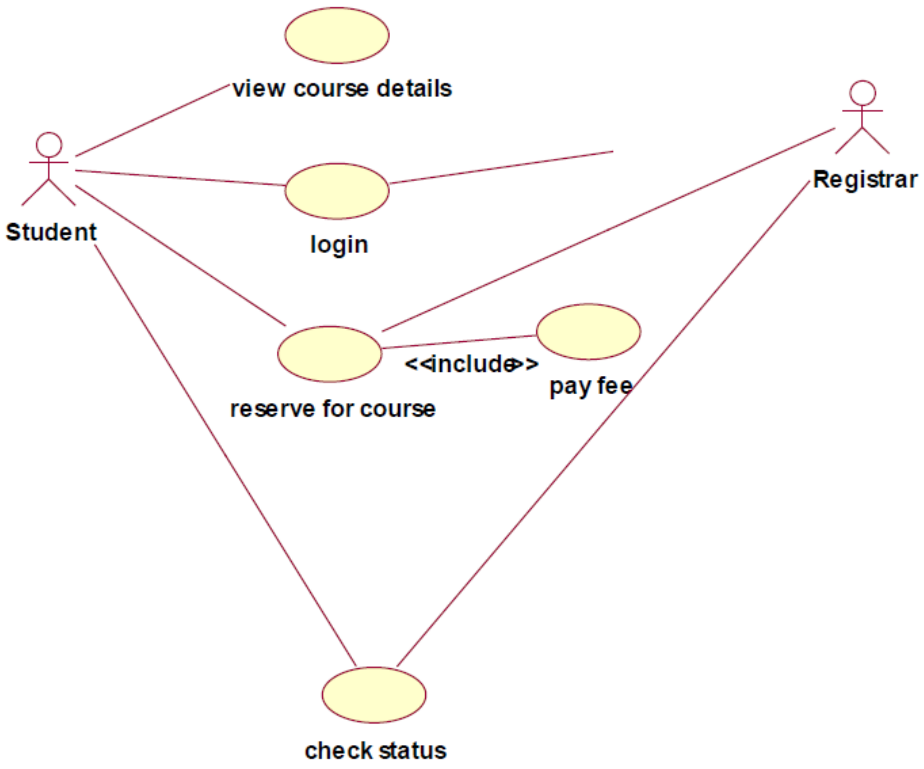
b) Develop elaborate Use case descriptions & scenarios

c) Develop Prototypes

d) Develop system sequence diagrams.

a) Use case Diagram

Use-Case Diagram:



b) Elaborated use-cases descriptions

Usecase name: Login

Brief:

The Student enters into the system's software and looks into the control panel, then he enters the login ID and password. If the password is valid then the system will display the next step or if the password is wrong then it shows a message.

Casual:

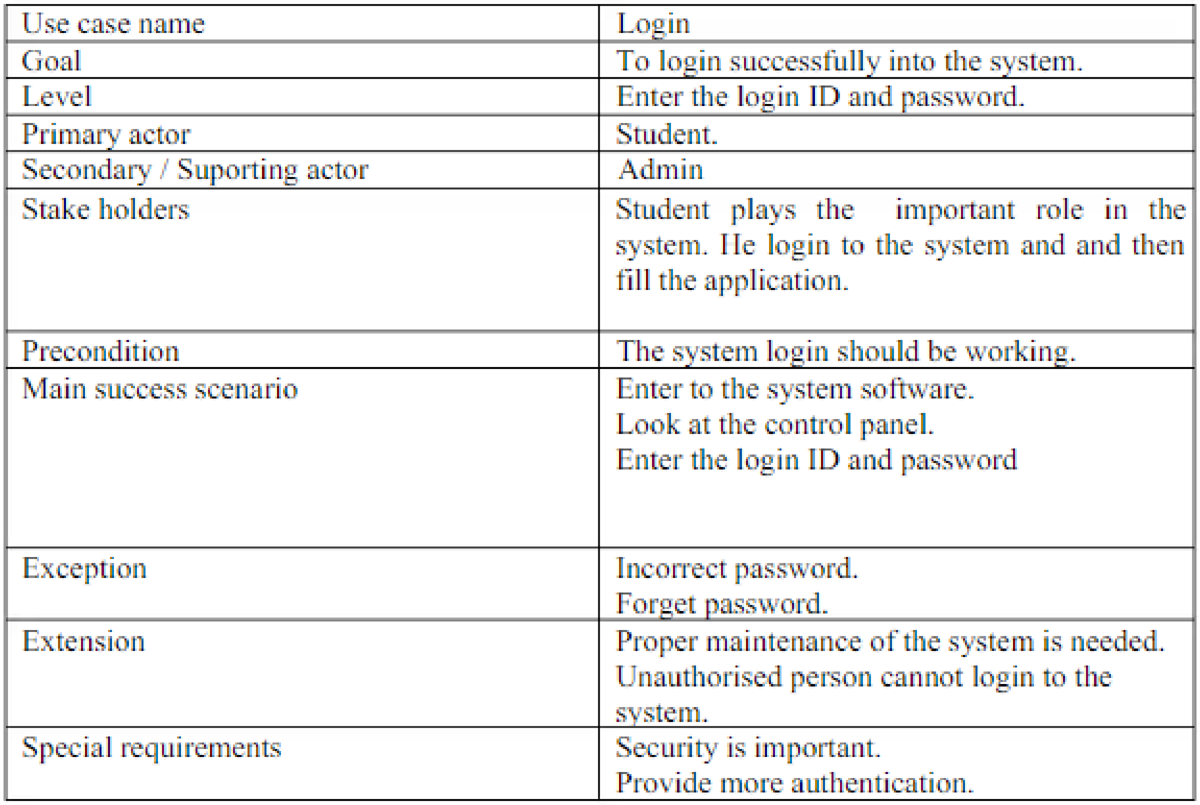
Success scenario:

The Student enters into the system's software and looks into the control panel, then he enters the login ID and password. . If the password is valid then the system will display the next step.

Alternate scenario:

If the password entered by the Student is invalid, the system cannot be opened and can't perform any actions. So he must reenter the valid password. The system allows to enter the valid password upto three attempts.

Fully Dressed:



Use case name	Login
Goal	To login successfully into the system.
Level	Enter the login ID and password.
Primary actor	Student.
Secondary / Suporting actor	Admin
Stake holders	Student plays the important role in the system. He login to the system and and then fill the application.
Precondition	The system login should be working.
Main success scenario	Enter to the system software. Look at the control panel. Enter the login ID and password
Exception	Incorrect password. Forget password.
Extension	Proper maintenance of the system is needed. Unauthorised person cannot login to the system.
Special requirements	Security is important. Provide more authentication.

USECASE NAME: View course details

- In this, a student can search all the courses available to him and choose the best course he wants.
- The student can view the course duration, faculty and department of the courses he may choose.

USECASE Name: Reserve for course

- When a student has successfully chosen a course, he can register to that course.
- Upon registration, the student’s details are stored in the database.

USECASE NAME: Pay fee

- After registration to any course, the student may see the details of his current course. He may wish to know details about fees and other information.

Usecase name: Check Status

Brief:

The Student can enter the system and select the course to check the status of seat availability for the course. If the seat is available for that course then the student will select the course else the student will select the other available course.

Casual:

Success scenario:

The student can enter the system and select the course to check the status of seat availability for the course. If the seat is available for that course then the student will select the course.

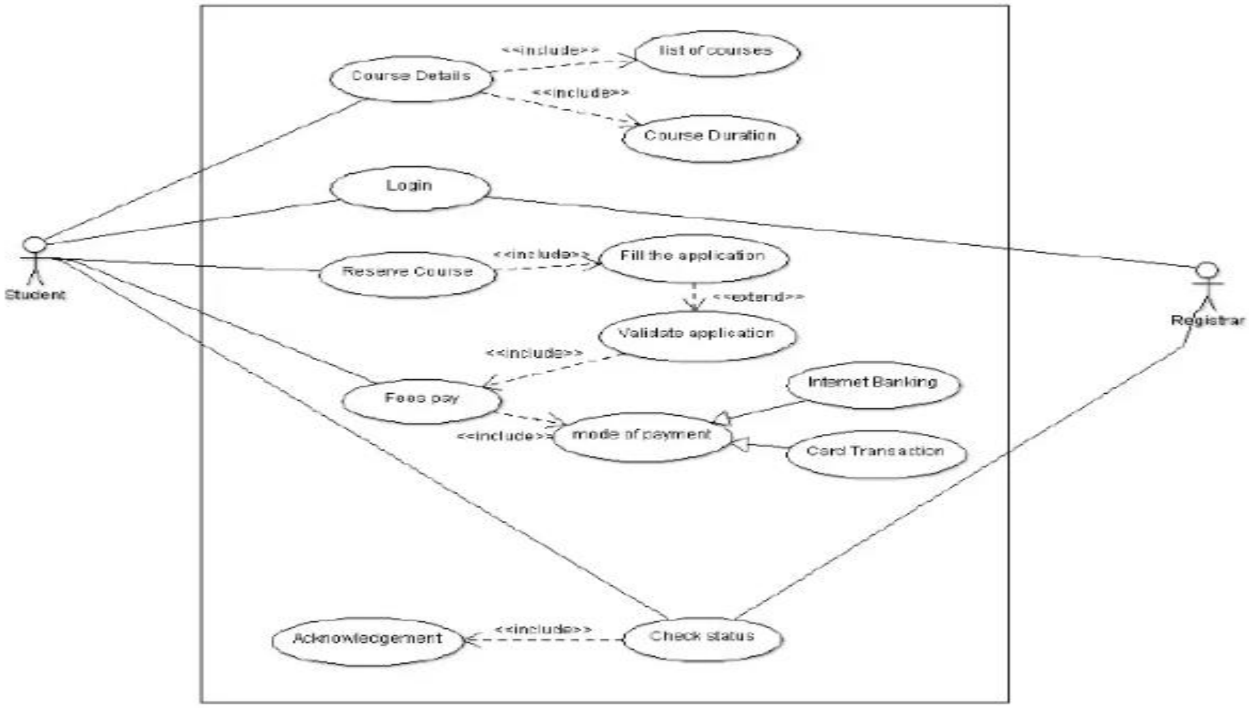
Alternate scenario:

But at some situations, if the seat is not available then the student will select the other available course.

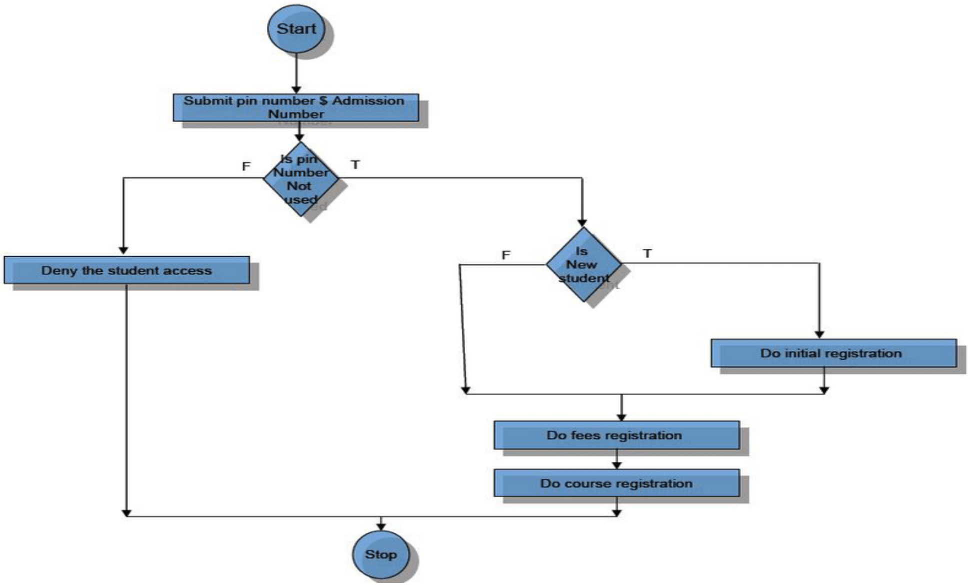
Fully Dressed:

Use casename	Check Status
Goal	To select the course from the list of course.
Level	To check the status of the course.
Primary actor	Student.
Secondary / Suporting actor	Admin , Registrar.
Stake holders	Student plays the important role in the system.
Precondition	The Student should eligible for that course
Main success scenario	The student can enter the system and select the course to check the status of seat availability for the course. If the seat is available for that course then the student will select the course.
Exception	Any other intervention of hazardous virus occurred, the whole system may be collapsed.
Extension	Proper maintenance of the system during the regular interval time.
Special requirements	Security Maintainence.

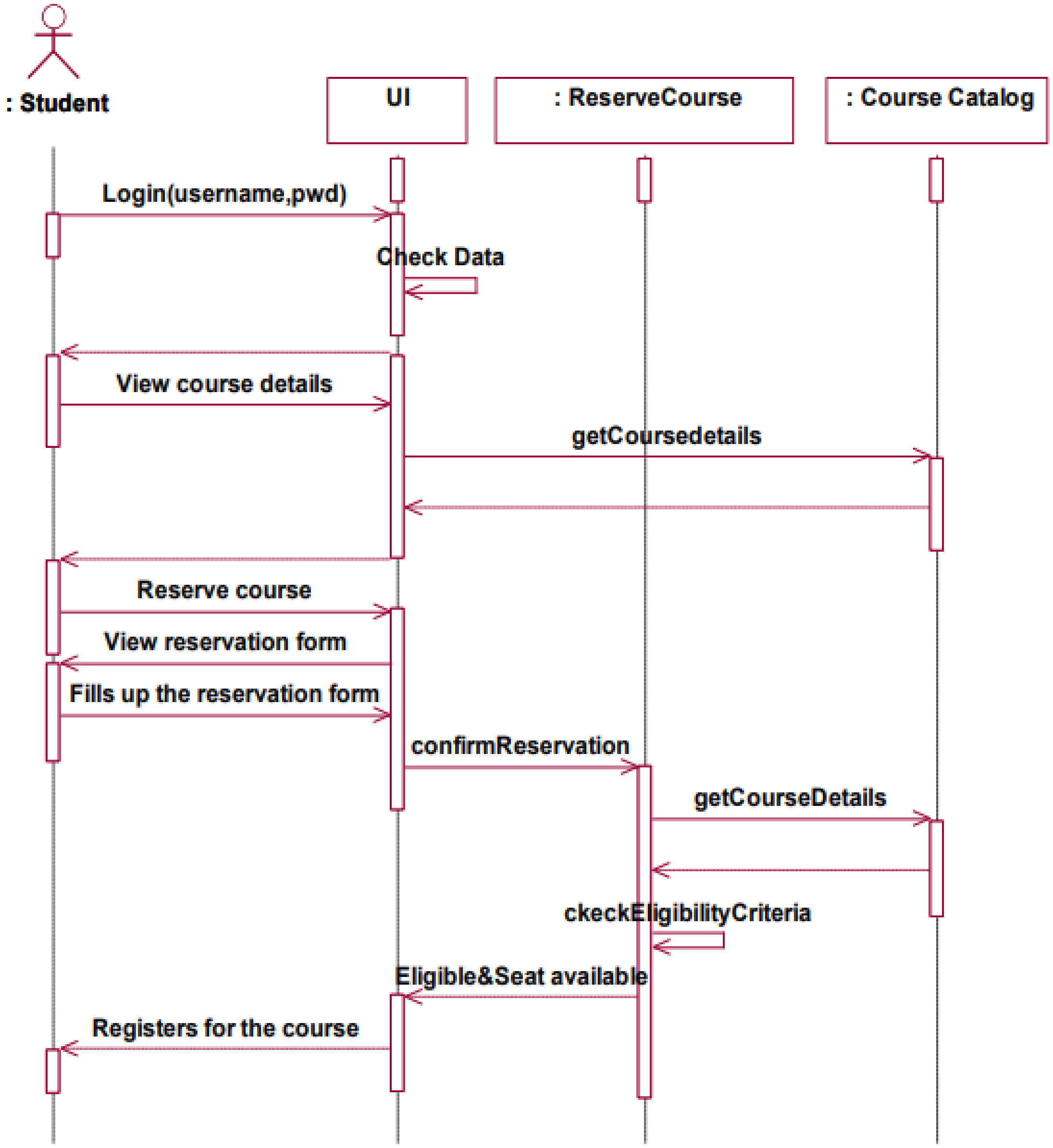
Elaborated Use Case Diagram



c) Develop prototypes



d) Develop system sequence diagrams.

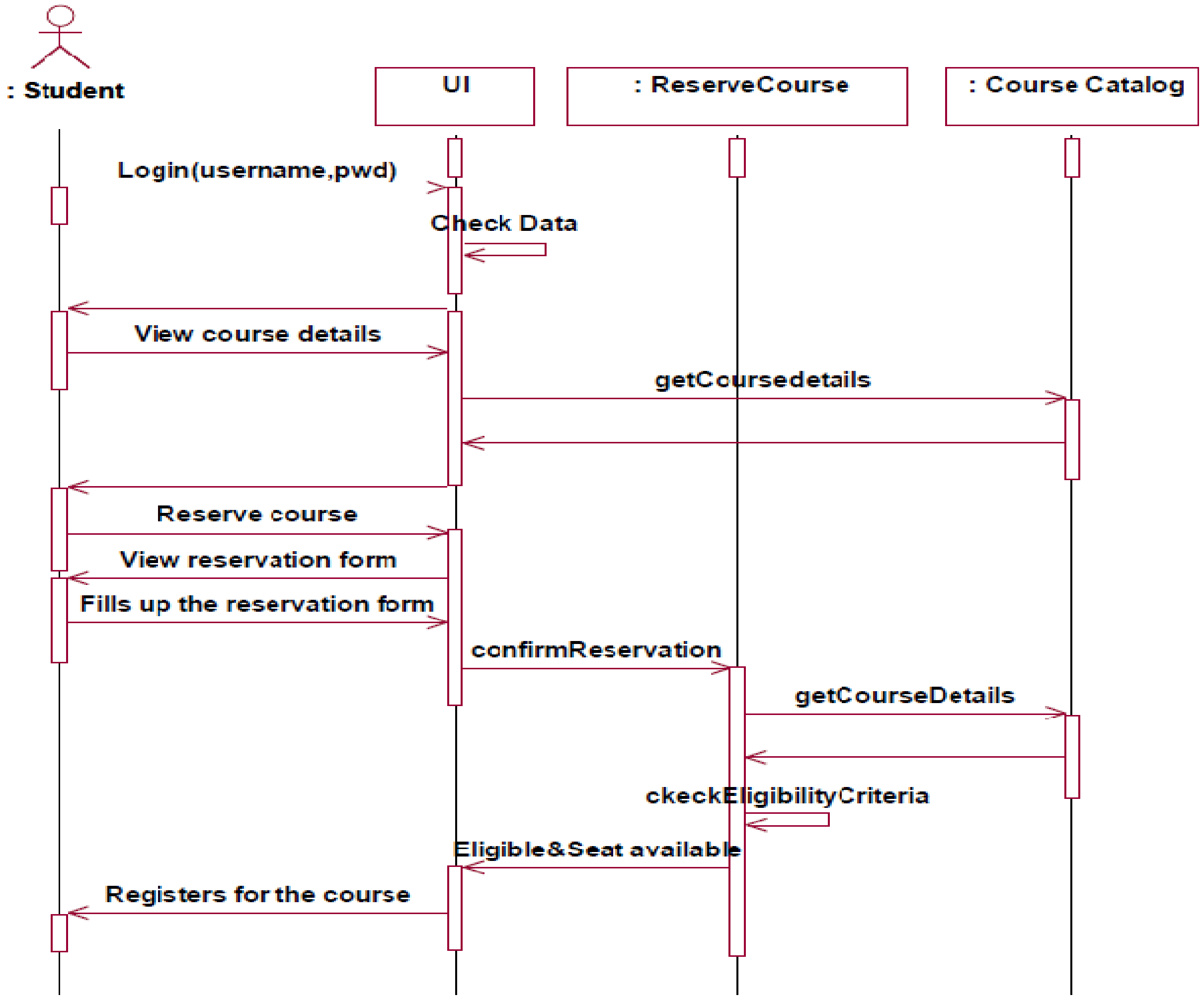


Week 7, 8, 9 & 10:

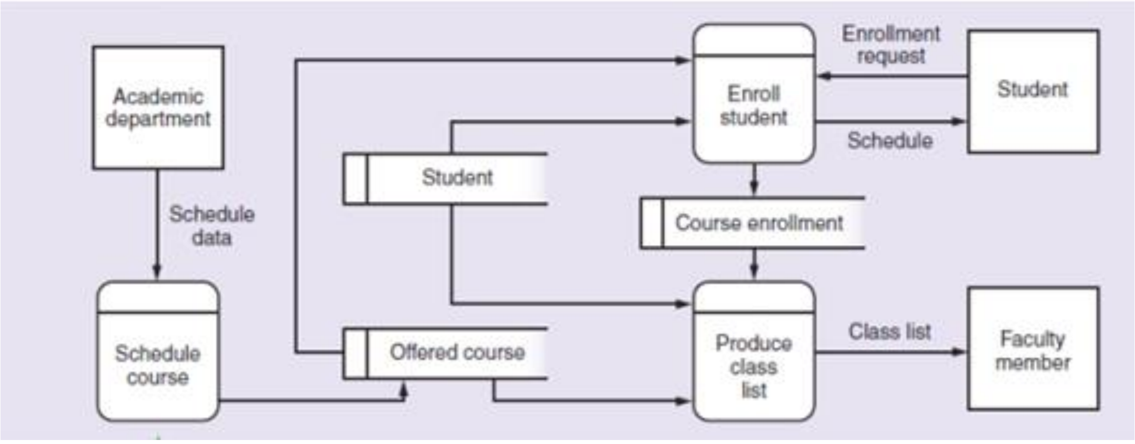
For each case study:

1) Develop high-level sequence diagrams for each use case
2) Identify MVC classes / objects for each use case
3) Develop Detailed Sequence Diagrams / Communication diagrams for each use case showing interactions among all the three-layer objects
4) Develop detailed design class model (use GRASP patterns for responsibility assignment)
5) Develop three-layer package diagrams for each case study

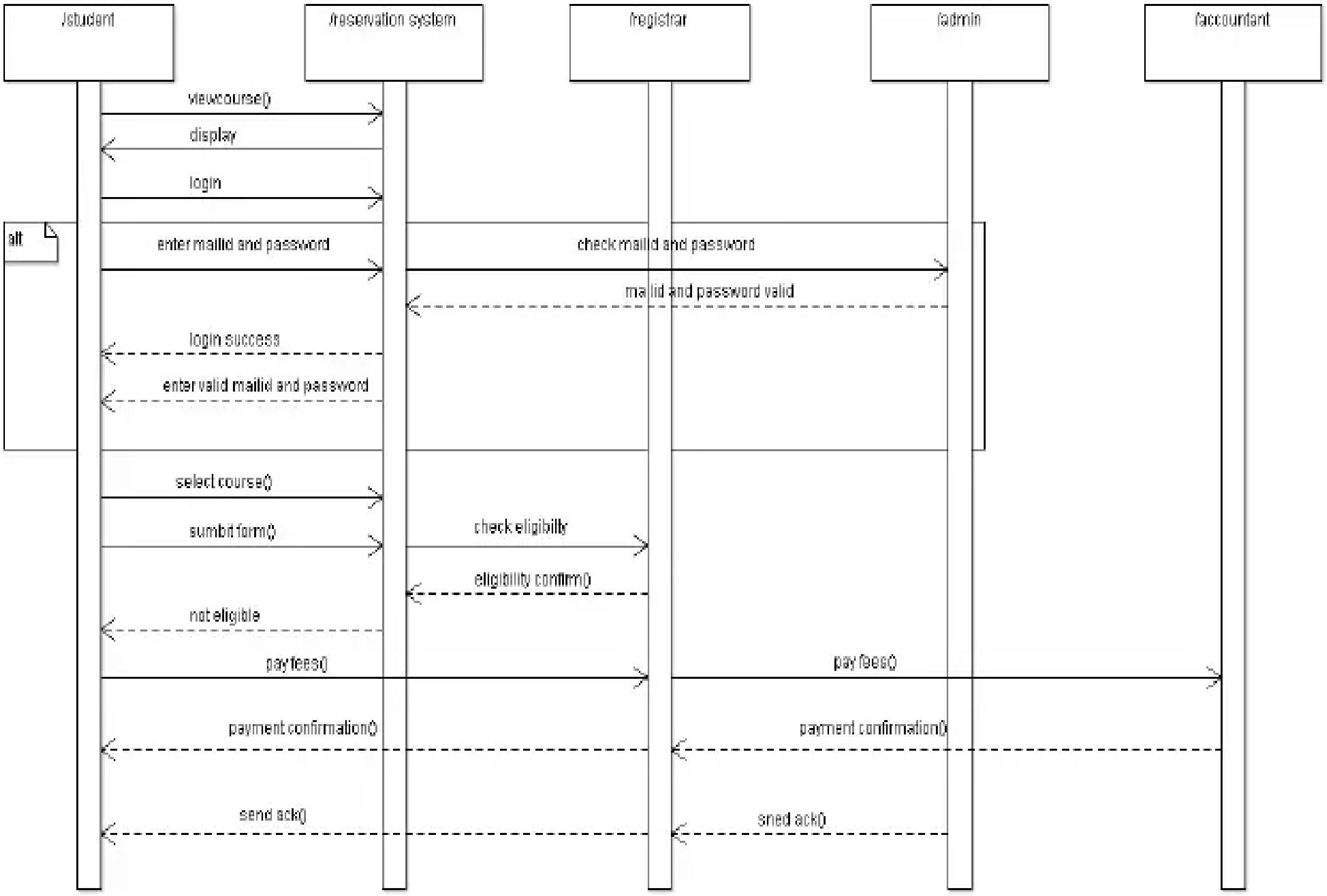
1. High-level sequence diagram for Online Course reservation system

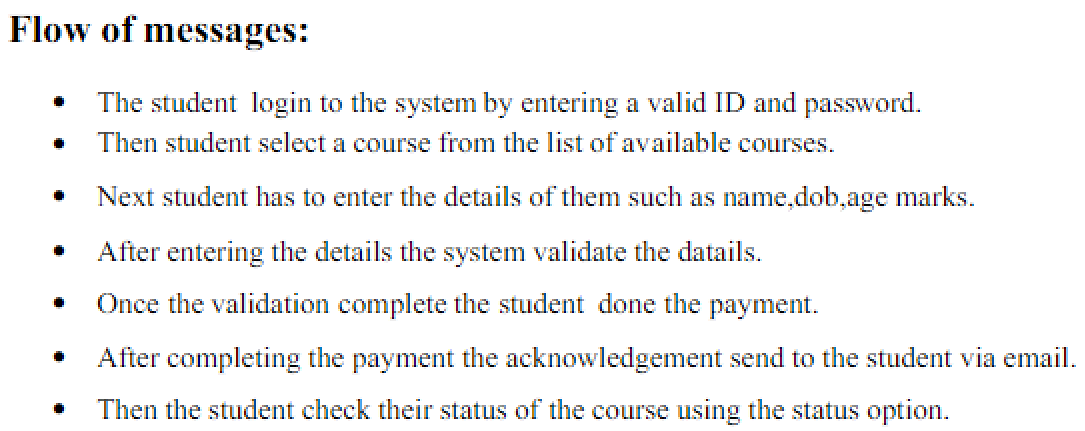


2. Identify MVC classes / objects for each use case



3. Detailed Sequence Diagrams / Communication diagrams





Flow of messages:

- The student login to the system by entering a valid ID and password.
- Then student select a course from the list of available courses.
- Next student has to enter the details of them such as name,dob,age marks.
- After entering the details the system validate the datails.
- Once the validation complete the student done the payment.
- After completing the payment the acknowledgement send to the student via email.
- Then the student check their status of the course using the status option.

4. Develop detailed design class model

CLASS DIAGRAM:

The class diagram is a graphical representation of all the classes used in the system and their operations, attributes and relationships.

The course registration system makes use of the following classes:

1. Student

2. Course Catalog

3. Reserve Course

1) STUDENT:

- It consists of the details of all the students present in the database.
- The attributes present in this class are student id, student name, student qualification, student address1, student address2, student address3, student mobile no, student emailed,, student dob, student gender.
- The object of this class is created as soon as the student registers to a course.
- The operations available to this class are add details (), modify details (), del details (), reserve course ().

2) COURSE CATALOG:

- The course catalog class consist of course id, course name, course duration course fee, course eligibility, total no of seat, course avail seat.
- The operations are add course(), update course(), del course().

3) RESERVE COURSE:

- The reserve catalog class consists of student id, course id, date, amt paid, reg id, DD no.
- The operation are get course details(), check eligibility(), confirm registration().

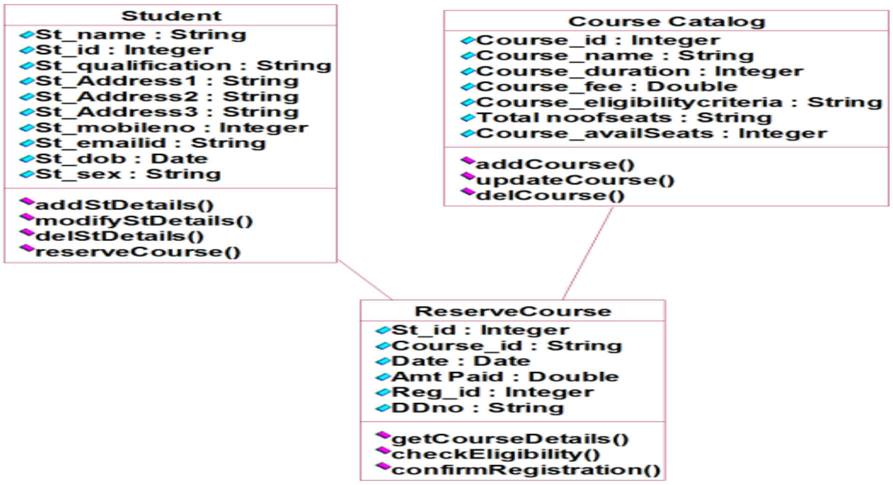


Fig: detailed design class model

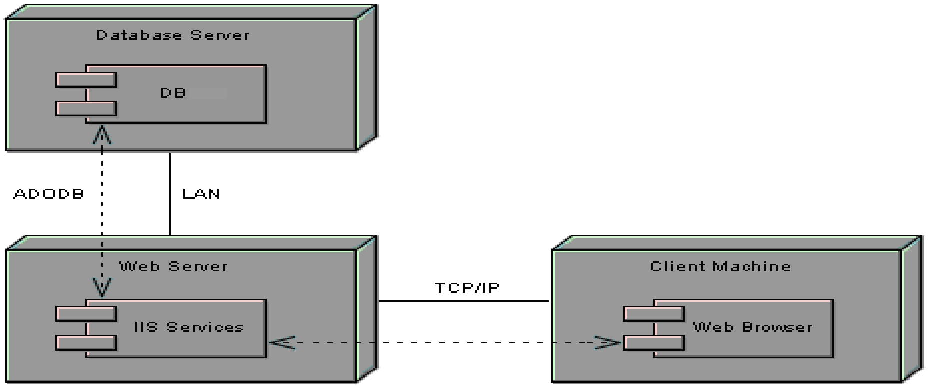
Week 11 & 12:

For each case study:

1) Develop Use case Packages
2) Develop component diagrams
3) Identify relationships between use cases and represent them
4) Refine domain class model by showing all the associations among classes

COMPONENT DIAGRAM:

Component diagrams are used to visualize the organization and relationships among components in a system.



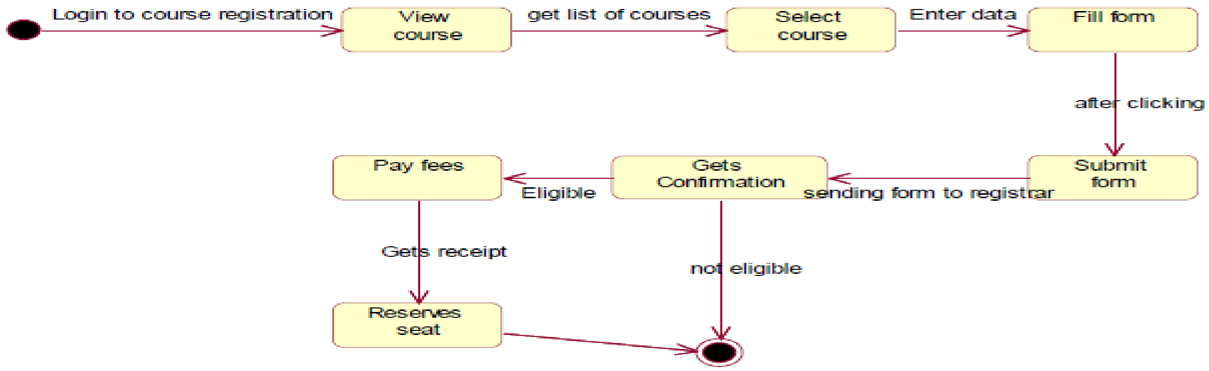
Week13onwards: For each case study:

1) Develop sample diagrams for other UML diagrams – state chart diagrams, activity diagrams and deployment diagrams.

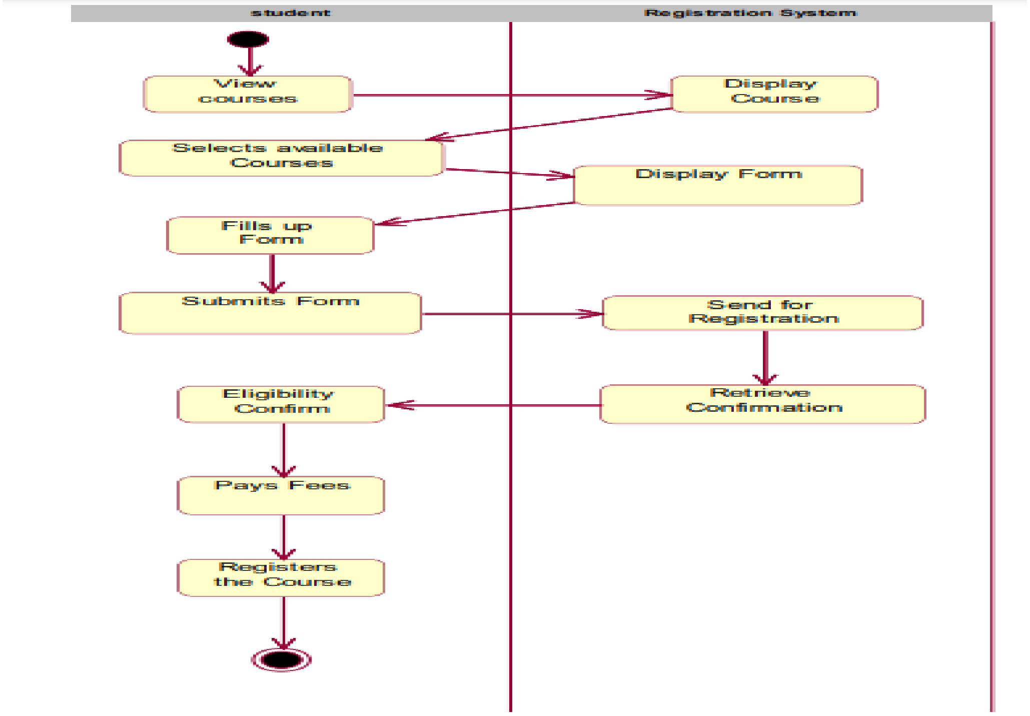
STATE CHART DIAGRAM:

Every object undergoes through some state and on receiving some event the state gets

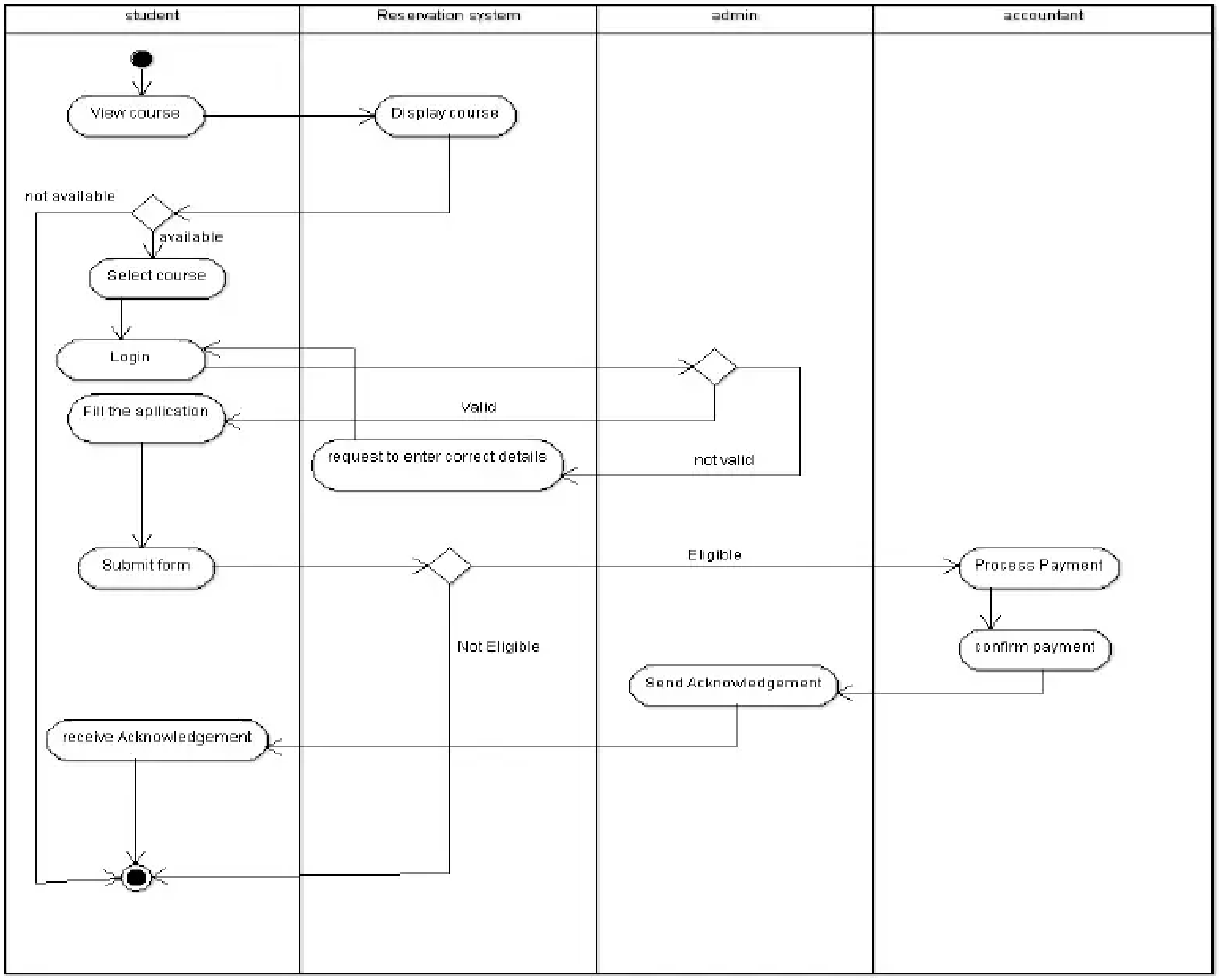
Changed. This transition of the state can be represented by the state transition diagram



Activity Diagram:

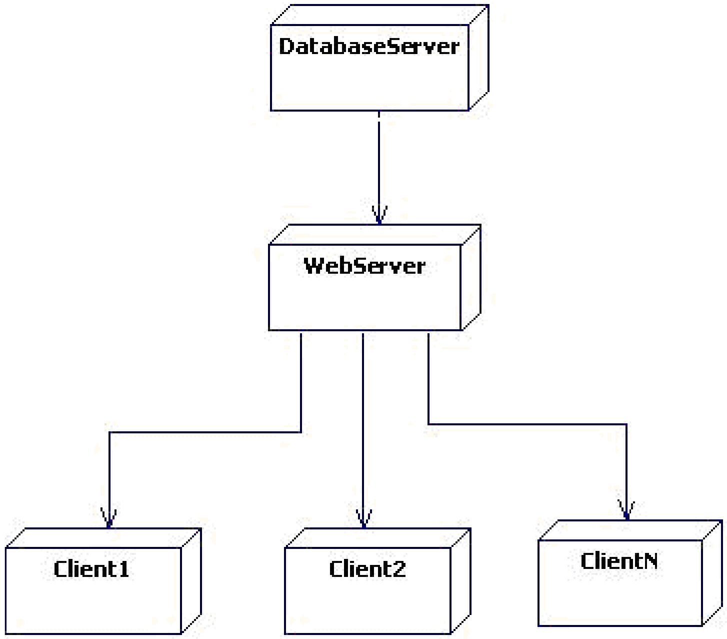


Detailed Activity Diagram

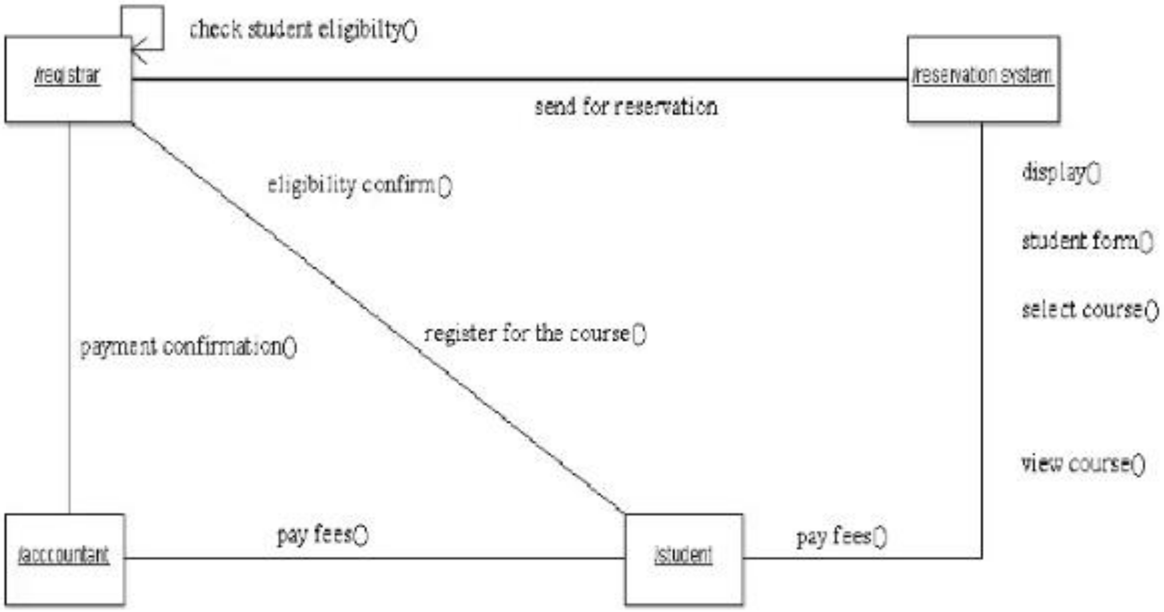


This Activity diagram consists of three swimlanes namely student,reservation system,admin,accountant.The initial process is course selection by the student.Theparallel behaviours such as branch used for selection of the course and login. The reservation then next processed by the filling the application by entering the details and get validate by the system.Then payment is done by the help of accountant.Then the acknowledgement send to the student for the resevation of a course.

DEPLOYMENT DIAGRAM

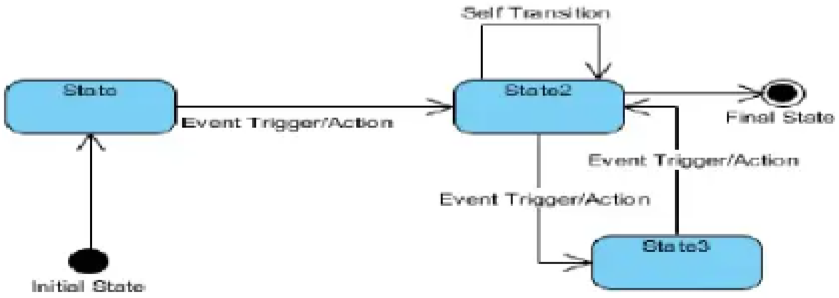


COLLABORATION DIAGRAM:

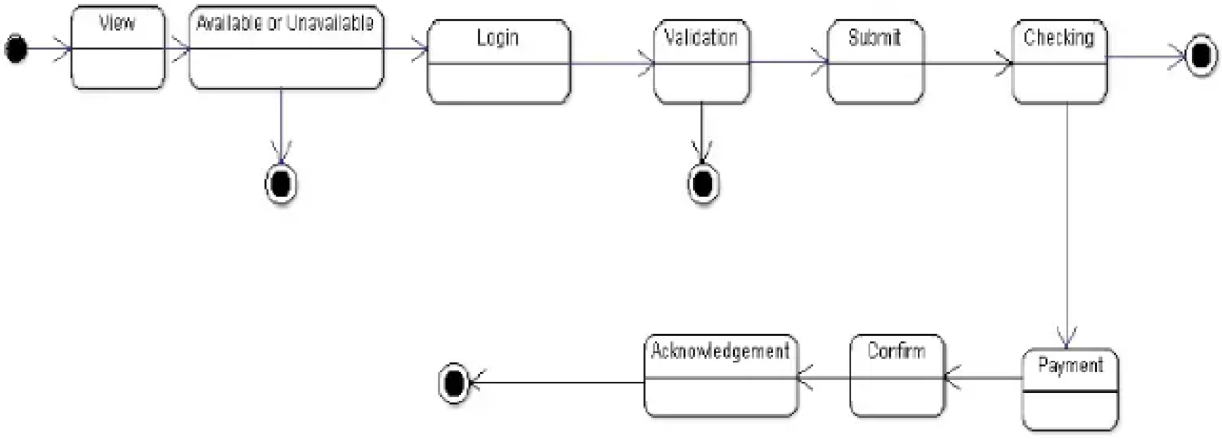


STATE CHART DIAGRAM:

A state machine Diagram (or start diagram, also called state chart of state transition diagram) is a behaviour which specifies the sequence of states an entity (or object) visits during its lifetime in response to events, together with its responses to those events.



STATE CHART DIAGRAM:



CASE STUDY: LIBRARY MANAGEMENT SYSTEM

Week 2, 3 & 4:

For each case study:

1) Identify and analyze events

2) Identify Use cases

3) Develop event table

4) Identify & analyze domain classes

5) Represent use cases and a domain class diagram using Rational Rose

6) Develop CRUD matrix to represent relationships between use cases and problem domain Classes

Week 5 & 6:

For each case study:

1) Develop Use case diagrams

2) Develop elaborate Use case descriptions & scenarios

3) Develop prototypes (without functionality)

4) Develop system sequence diagrams

Week 7, 8, 9 & 10:

For each case study:

1) Develop high-level sequence diagrams for each use case

2) Identify MVC classes / objects for each use case

3) Develop Detailed Sequence Diagrams / Communication diagrams for each use case showing Interactions among all the three-layer objects

4) Develop detailed design class model (use GRASP patterns for responsibility assignment)

5) Develop three-layer package diagrams for each case study

Week 11 & 12:

For each case study:

1) Develop Use case Packages

2) Develop component diagrams

3) Identify relationships between use cases and represent them

4) Refine domain class model by showing all the associations among classes

Week 13 onwards:

For each case study:

1) Develop sample diagrams for other UML diagrams – state chart diagrams, activity diagrams and deployment diagrams.

Problem statements of Library Management System

The library system is a web-based application which is designed for automating a library

It allows the librarian to maintain the information about

Books

Magazines

CDs

Its users

Furthermore, it should provide the following facilities to its users

Search for items

Browse

Checkout items

Return items

Make reservation

Remove reservation, etc.

For borrowing the item from the library, any user must get registered in the system initially

- The users can search for any item in the library by using the 'search option'
- If the user finds the item he/she is searching for in the library, he/she can checkout the item from the library
- If the study material is not available in the library at the moment, the user can make reservation for that item
- The moment the item is available, the user who first reserved for that item is notified first
- If the user checks out the item from the library, the reservation gets cancelled automatically.
- The reservation can also be cancelled through an explicit cancellation procedure

The librarian is an employee of the library who interacts with the borrowers whose work is supported by the system

The system allows the librarian to perform the following functions with a lot of ease

- Create
- Update
- Delete information about titles
- Borrowers
- Items and reservations in the system

Week 2, 3&4

Case Study 1: Library Management System

g) Identify the users and Analyze Events
h) Identify Use cases
i) Develop event table
j) Identify & analyze domain classes
k) Represent Use cases & Domain class diagram using Umbrello
l) Develop CRUD matrix to represent relationships between use cases and problem domain classes.

a) Identify and Analyze Events

USERS	EVENTS
New user register	1.To sign up a new user to this system
Student Login	1.So as to confirm that only an authenticated user is using the project
Search book	The user can search book based on book id, book name, or by author name.
Issue Book	To help the user get the required books issued.
Return Book	To return the book before the last date without fine, or after the specified time duration with a late fine.

Admin	It is to be operated by the admin with a unique id and password. The admin is the person who decides authentication and authorization for all the different users of the application.
Librarian	Includes all the library staff who are required to enter the records in the system and keep an eye on the various activities like the issue of the book, the return of the book, non-availability of books, etc.

b) Identify Use cases

Actor vs
Usecase:

Librarian
- Issue a book
- Update and maintain records
- Request the vendor for a book
- Track complaints

User
- Register
- Login
- Search a book
- Request for issue
- View history
- Request to the Librarian
- Unregister

Books Database
- Update records
- Show books status

Vendors
- Provide books to the library
  Payment acknowledgement

c) Develop event table

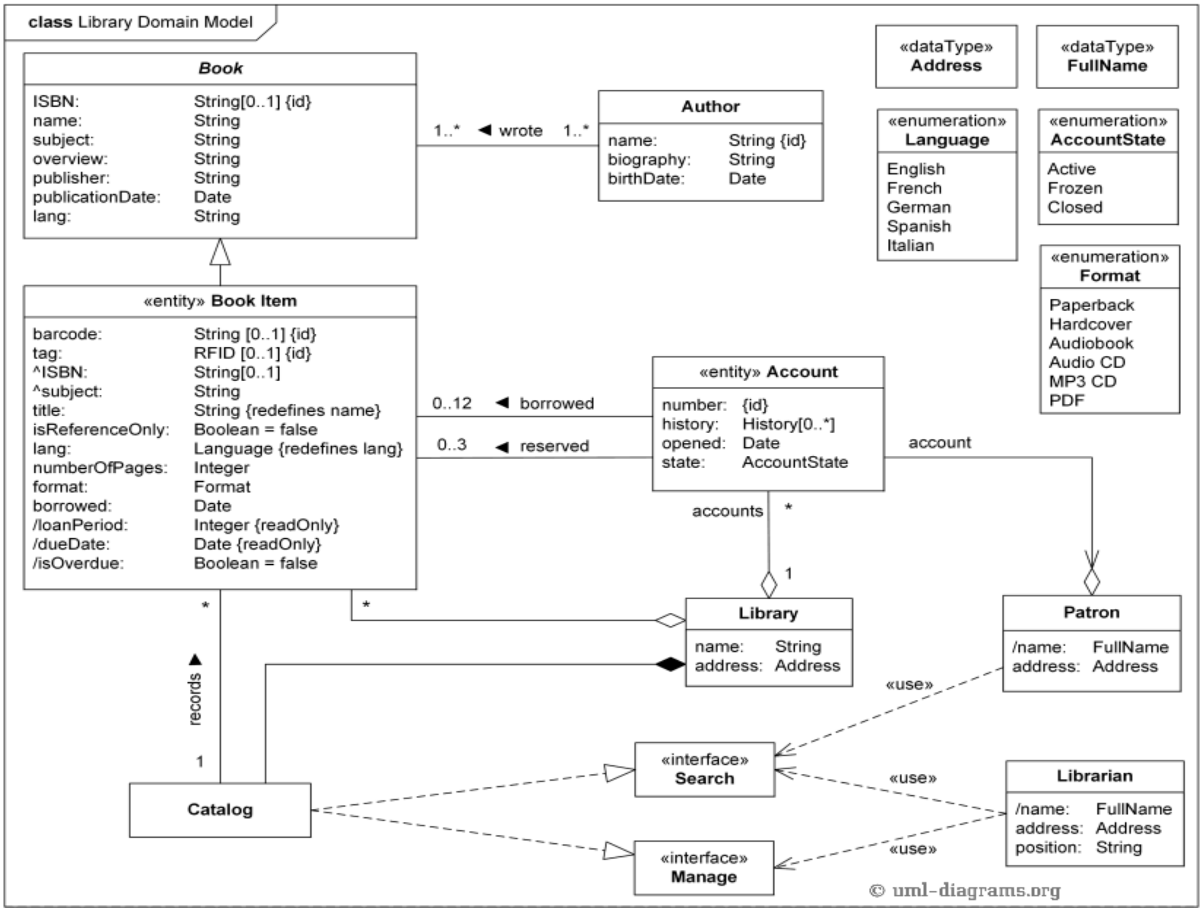
	New users			**Issue books**		**Admin**	**Librarian**
	Registers	Student login	Search for books		**Return book**		
To sign up a new user to this system	*						
So as to confirm that only an authenticated user is using the project	+	+	*	*,+	*,+		*
The user book can search name book based on book id		+	*	*	*		*
To help the user get the required books issued.		*	*		*	+	*
To return the book before the last date without fine, or after the specified time duration with a late fine.		+		+	+	*	*
It is to be operated by the admin with a unique id and password. The admin is the person who decides authentication and authorization for	+	+	+	*	*	+	+

all the different users of the application.							
Includes all the library staff who are required to enter the records in the system and keep an eye on the various activities like the issue of the book, the return of the book, non-availability of books, etc.	*	*	*	*	*	+	+

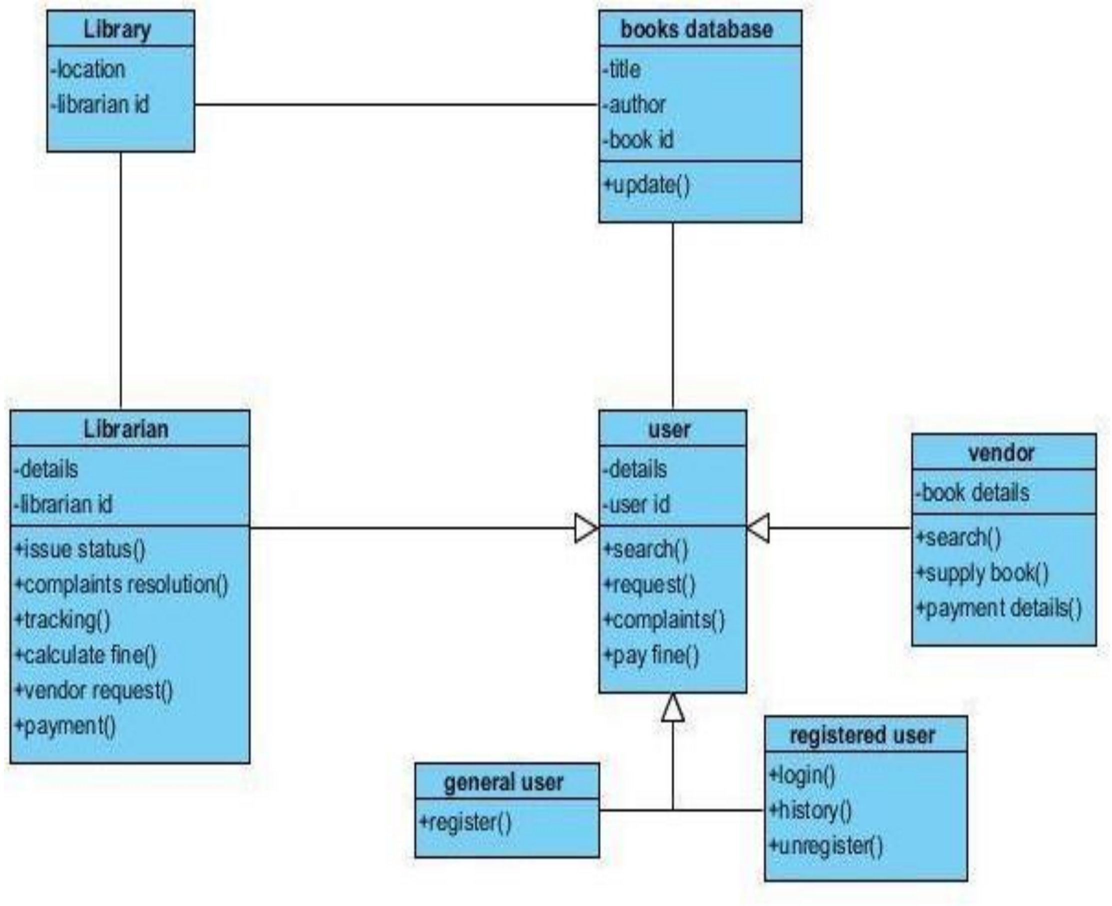
* Event can occur zero to many times

+ Event can occur zero to one time

d) Identify & analyze domain classes



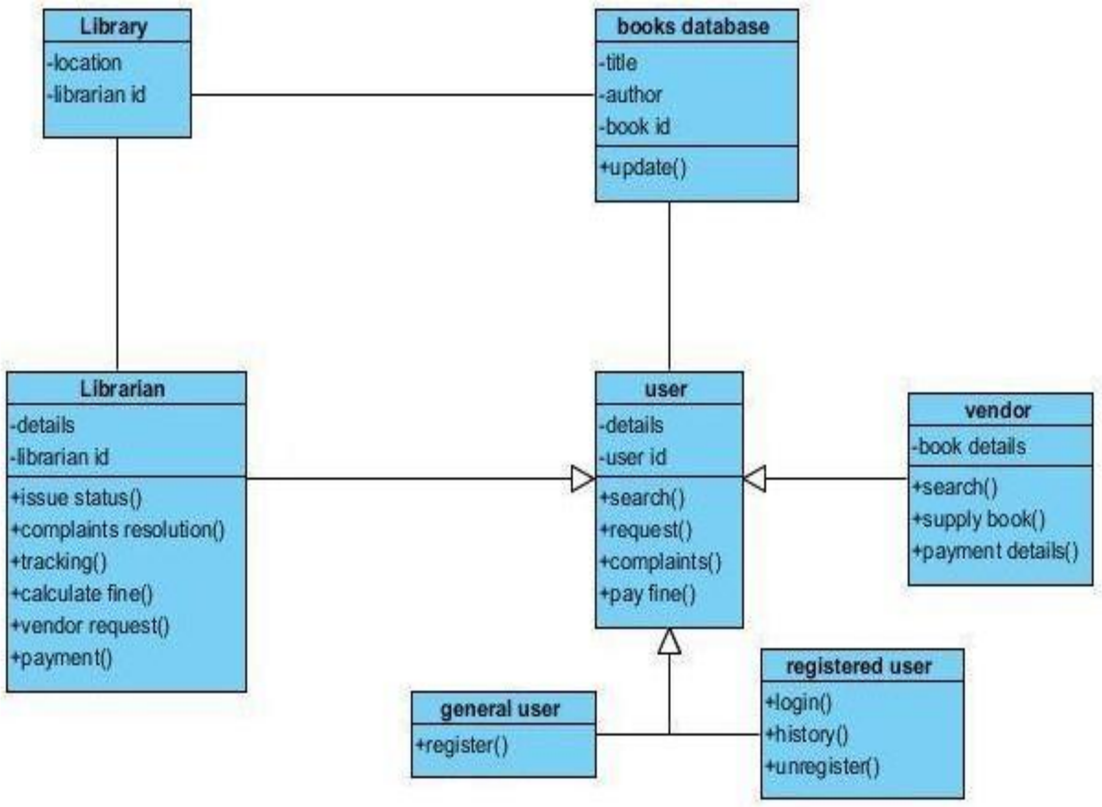
Domain Model:



e) Represent Use cases & Domain class diagram using Umbrello

•<u>Class Diagram</u>

Classes
identified:
Library
Libraria
n Books
Database
User
vendor



<u>Use-case Diagram</u>

Actors vs Use
Cases:
Librarian
•Issue a book
•Update and maintain records
•Request the vendor for a book
•Track complaints

User
•Register
•Login
•Search a book
•Request for isse
•View history
•Request to the Librarian
•Unregister

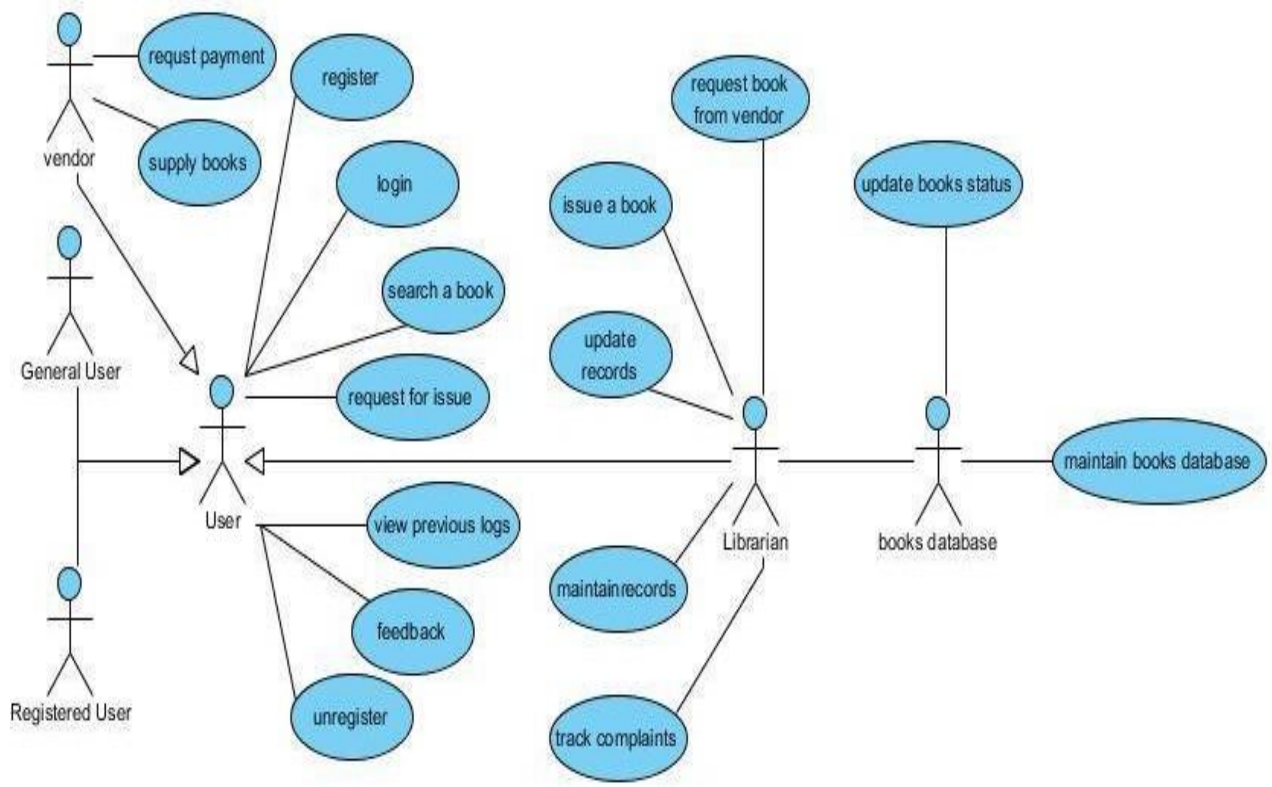
Books Database
•Update records

•Show books status

Vendors

•Provide books to the library

•Payment acknowledgement



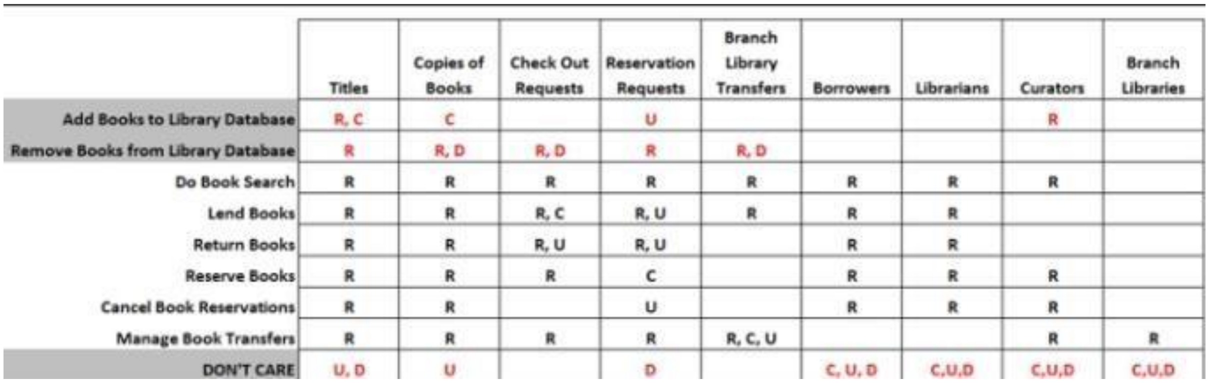
f) Develop CRUD matrix to represent relationships between use cases and problem domain classes

CRUD technique – an acronym for Create, Read/Report, Update, and Delete; a technique to

validate or refine use cases

- The CRUD Matrix is an excellent technique to identify the Tables in a Database which are used in any User interaction with a Web Site.
- CRUD means ‘Create, Read, Update or Delete’, and the CRUDD Matrix identifies the Tables involved in any CRUD operation.
- It is very valuable to combine a CRUD Matrix with the analysis of possible User Scenarios for the Web Site

- The analysis helps to identify any Tables which are not used, and any Tables which are used heavily, and may therefore be a performance bottleneck



	Titles	Copies of Books	Check Out Requests	Reservation Requests	Branch Library Transfers	Borrowers	Librarians	Curators	Branch Libraries
Add Books to Library Database	R, C	C		U				R	
Remove Books from Library Database	R	R, D	R, D	R	R, D				
Do Book Search	R	R	R	R	R	R	R	R	
Lend Books	R	R	R, C	R, U	R	R	R		
Return Books	R	R	R, U	R, U		R	R		
Reserve Books	R	R	R	C		R	R	R	
Cancel Book Reservations	R	R		U		R	R	R	
Manage Book Transfers	R	R	R	R	R, C, U			R	R
DON'T CARE	U, D	U		D		C, U, D	C,U,D	C,U,D	C,U,D

Week-5 & 6

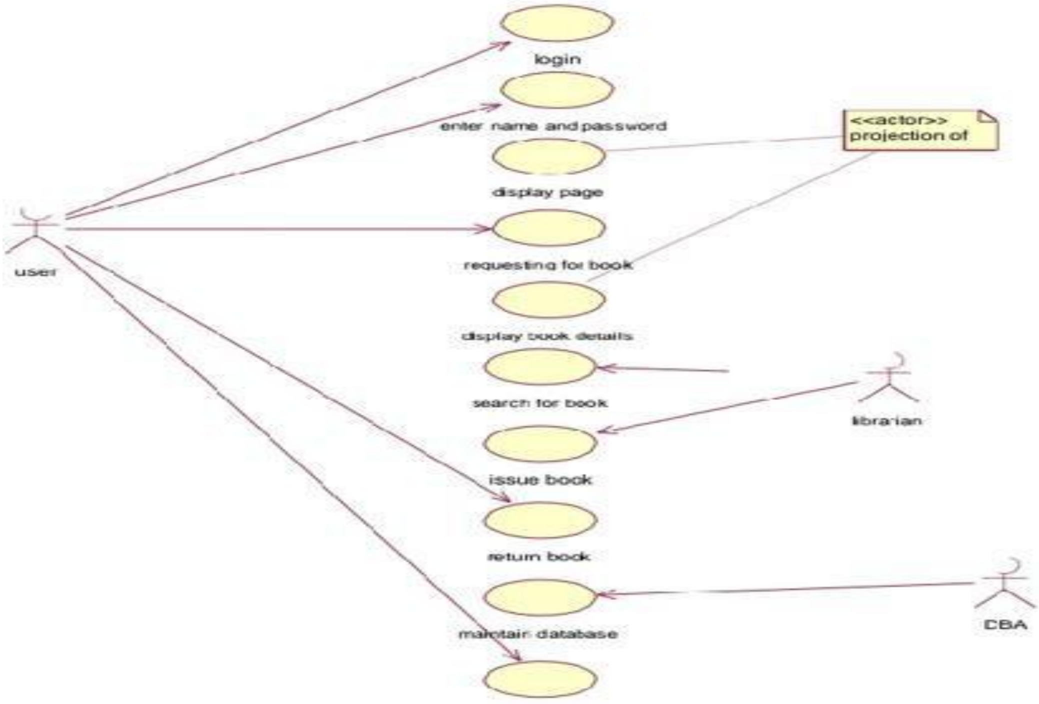
a) Develop Use Case diagrams

b) Develop elaborate Use case descriptions & scenarios

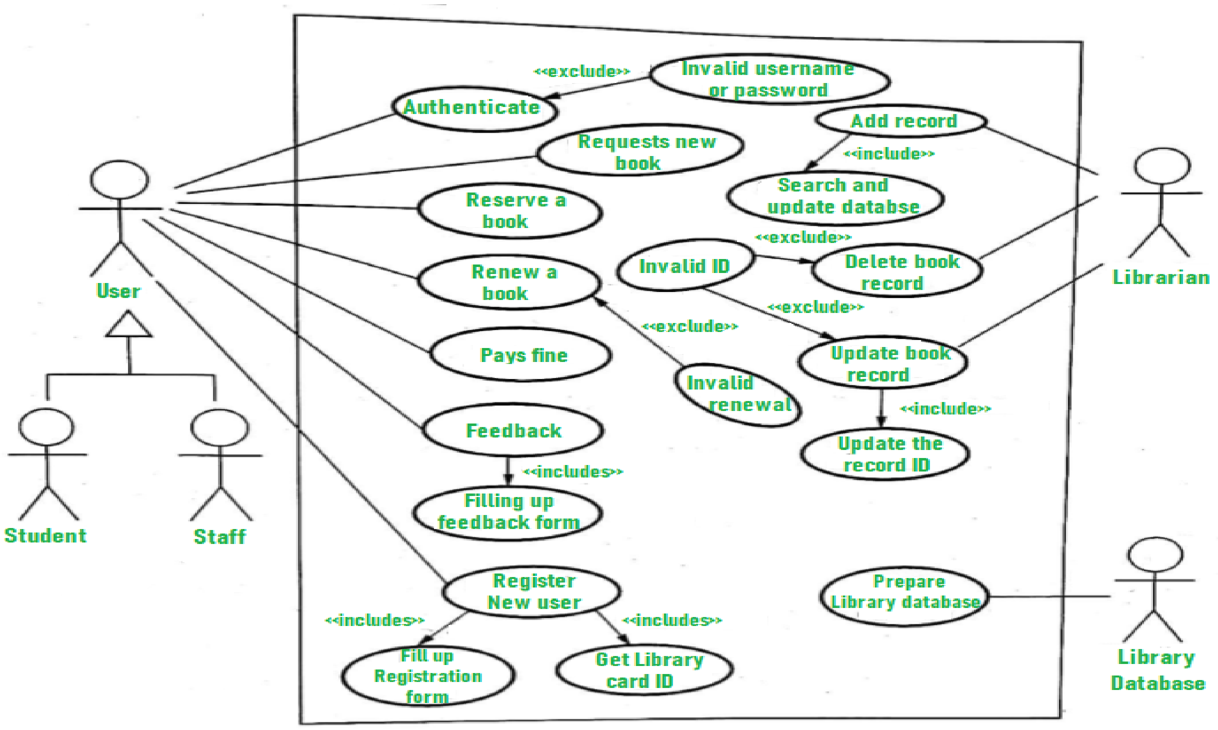
c) Develop Prototypes

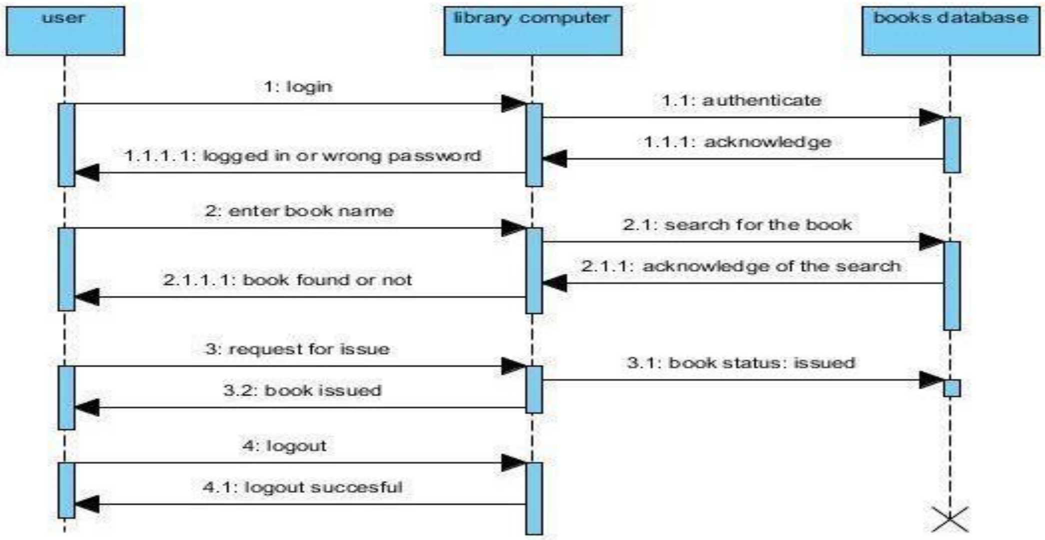
d) Develop system sequence diagrams.

a) Use case Diagram



b) Elaborated use-cases descriptions





c) Develop system sequence diagrams

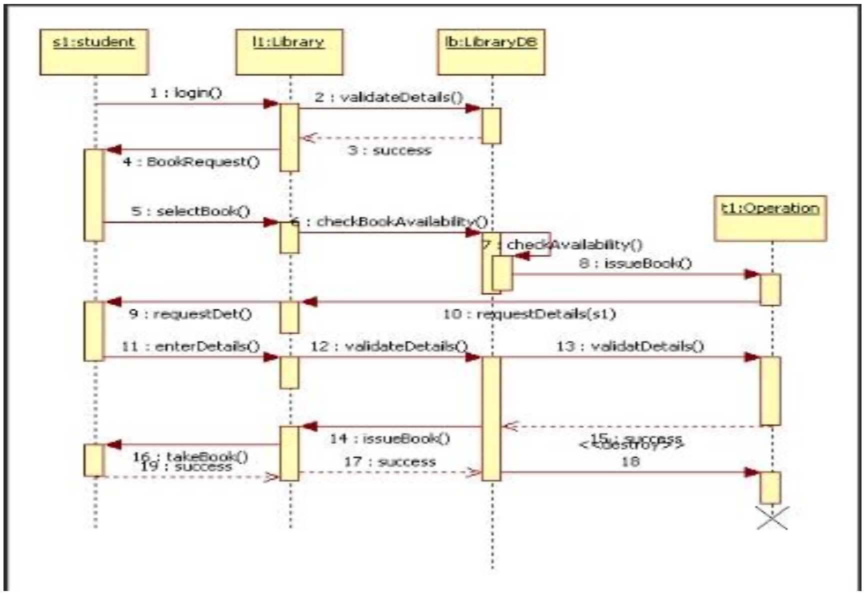
Week 7, 8, 9 & 10:

For each case study:

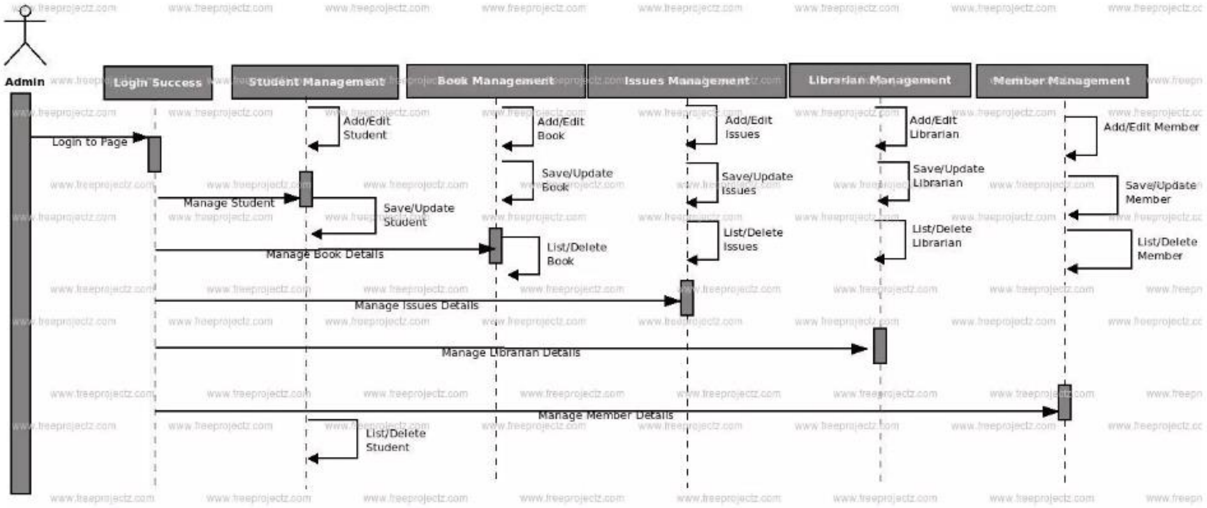
1) Develop high-level sequence diagrams for each use case
2) Identify MVC classes / objects for each use case

3) Develop Detailed Sequence Diagrams / Communication diagrams for each use case showing interactions among all the three-layer objects
4) Develop detailed design class model (use GRASP patterns for responsibility assignment)
5) Develop three-layer package diagrams for each case study

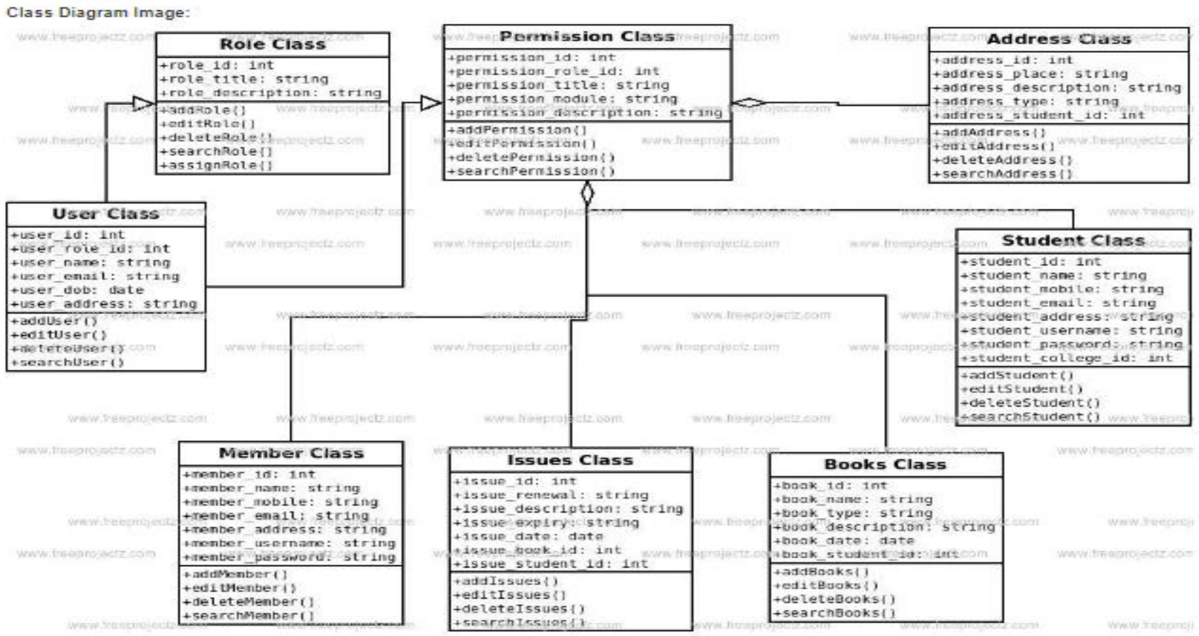
1. High-level sequence diagram for library management system:



3. Detailed Sequence Diagrams / Communication diagrams



4. Develop detailed design class model

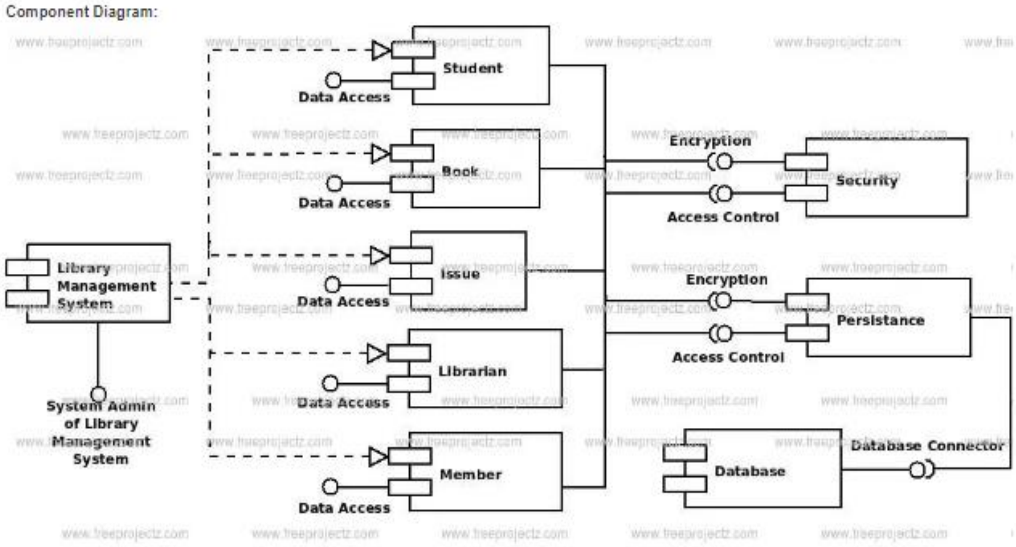


Week 11 & 12:

For each case study:

1) Develop Use case Packages
2) Develop component diagrams
3) Identify relationships between use cases and represent them
4) Refine domain class model by showing all the associations among classes

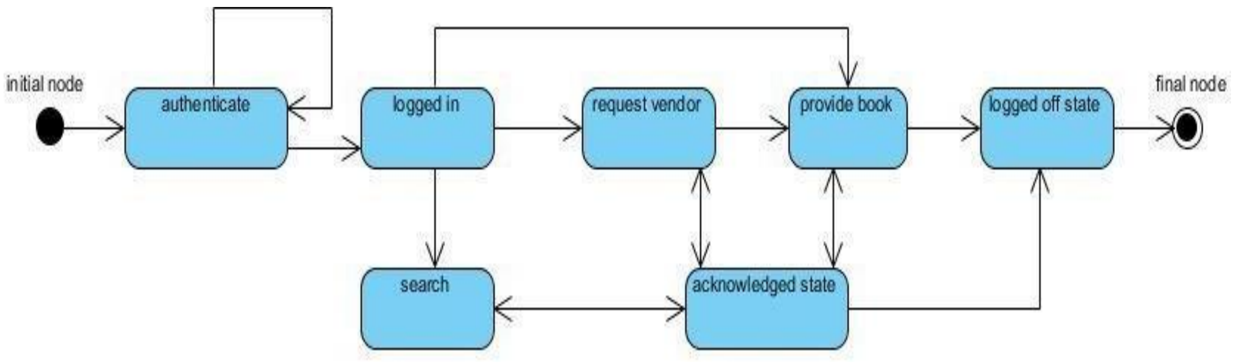
COMPONENT DIAGRAM:



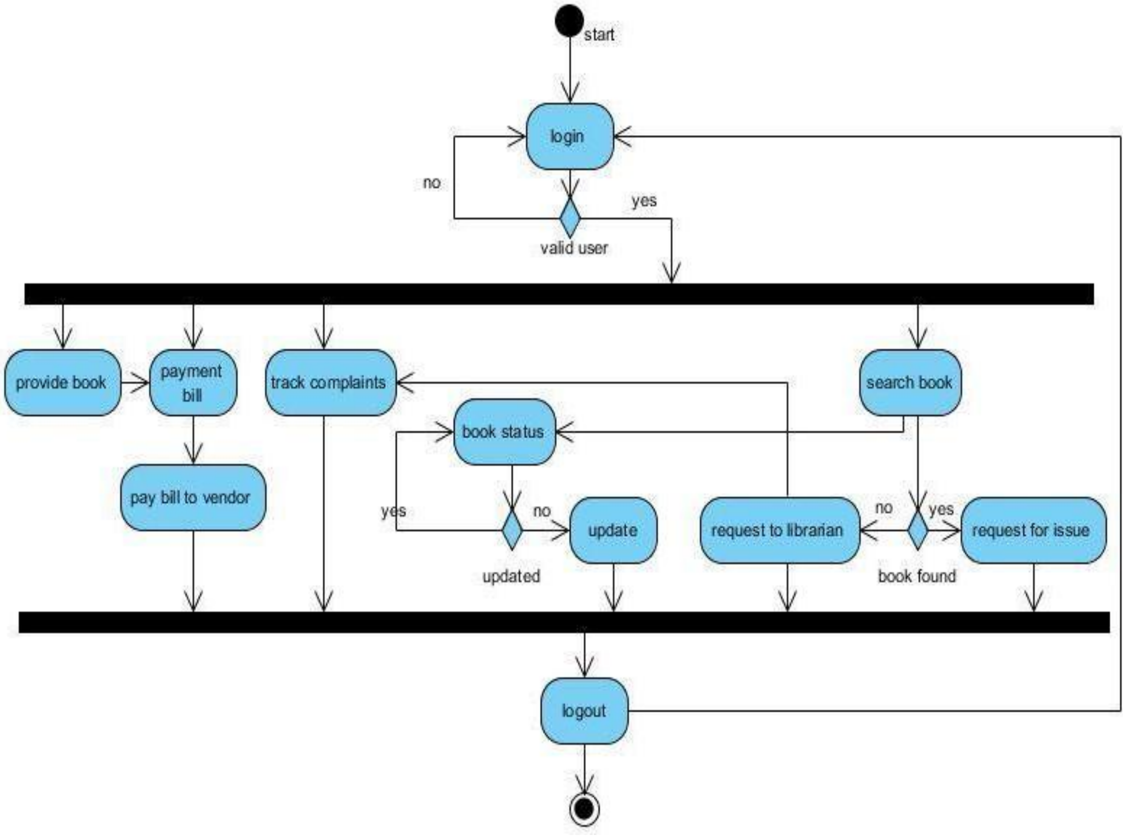
Week13onwards: For each case study:

1) Develop sample diagrams for other UML diagrams – state chart diagrams, activity diagrams and deployment diagrams.

STATE CHART DIAGRAM:



Activity Diagram:



CASE STUDY: E-TICKETING

PROBLEM STATEMENT

Our project is carried out to develop software for online Railway Reservation System. This system has various options like reservation, cancellation and to view details about available seats. Our project mainly simulates the role of a Railway ticket booking officer, in a computerized way. The reservation option enables a person to reserve for a ticket at their home itself. All he/ she has to do is to just login and enter the required details. After this the reservation database is updated with the person details, train name and also the source and destination place. The cancellation option enables the passenger to cancel the tickets that hasbeen already booked by him/her. The availability option prompts the person to enter train number, train name and date of travel. After this the availability database is accessed and available positions are produced.

Week 2, 3&4

Case Study 1: E-TICKETING

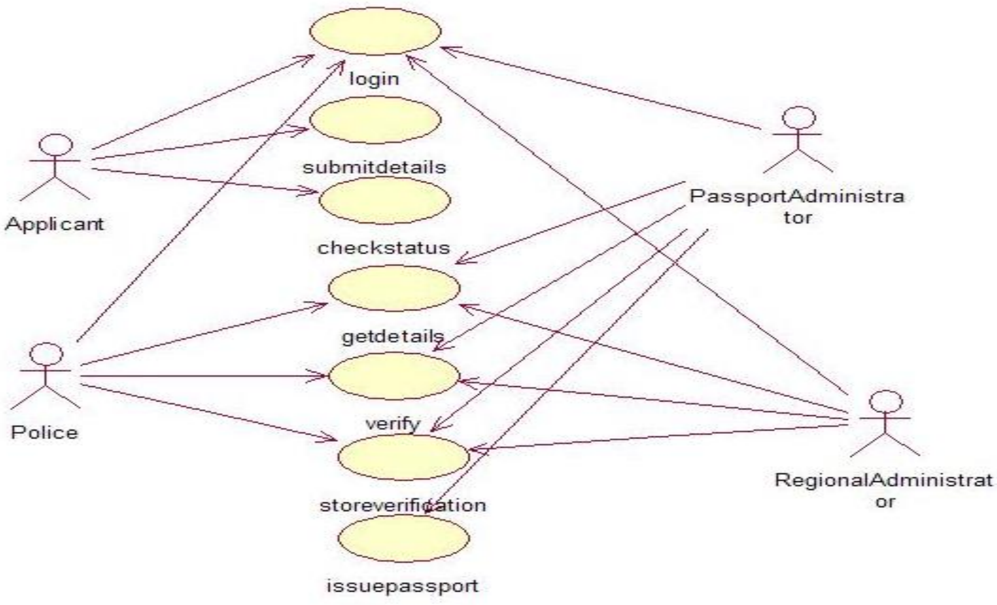
a) Identify the users and Analyze Events
b) Identify Use cases
c) Develop event table
d) Identify & analyze domain classes
e) Represent Use cases & Domain class diagram using Umbrello
f) Develop CRUD matrix to represent relationships between use cases and problem domain classes.

a) Identify and Analyze Events

USERS	EVENTS
APPLICANT	The applicant has attribute such as name and password and operations are login, give details and logout. The applicant login and fill the details that are required for applying the passport .After applying the person can view the status of the passport verification process
THE DATABASE	The database has attributed such as name and operation is store. The purpose is to store the data.

REGIONAL ADMINISTRATOR	The regional administrator has attribute such as name and operation are get details, verify details and send. The regional administrator get the details form database and verify with their database
PASSPORT ADMINISTRATOR	The passport administrator has attributed such as name and operation are get details, verify details and issue. The passport administrator get the details form database and verify with their database , update the verification and issue the passport
THE POLICE	The police has attribute such as name and operation are get details, verify details and send. The police get the details form database and verify with their database , update the verification in the database

b) Identify Use cases



c) Develop event table

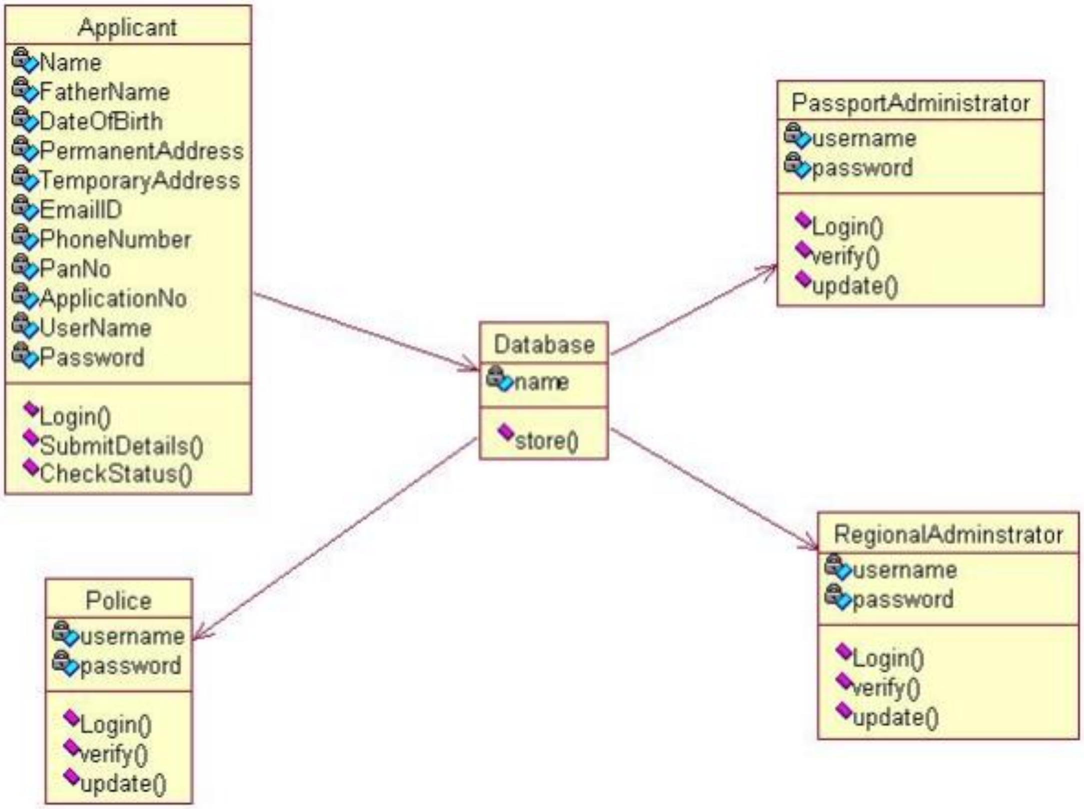
	APPLICANT	THE DATABASE	REGIONAL ADMINISTRATOR	PASSPORT ADMINISTRATOR	THE POLICE
The applicant has attribute such as name and password and operations are login, give details and logout.	+	*	+	+	+
The database has attributed such as name and operation is store. The purpose is to store the data.	+	*			
The regional administrator has attribute such as name and operation are get details Verify details and send.	*	+	+	+	+
The passport administrator has attributed such as name and operation are get details, verify details and issue.	*	*	+	+	+
The police has attribute such as name and operation are get	*	*	+	+	+

details Verify details and send. The police get the details form database and verify with their database					

*   Event can occur zero to many times

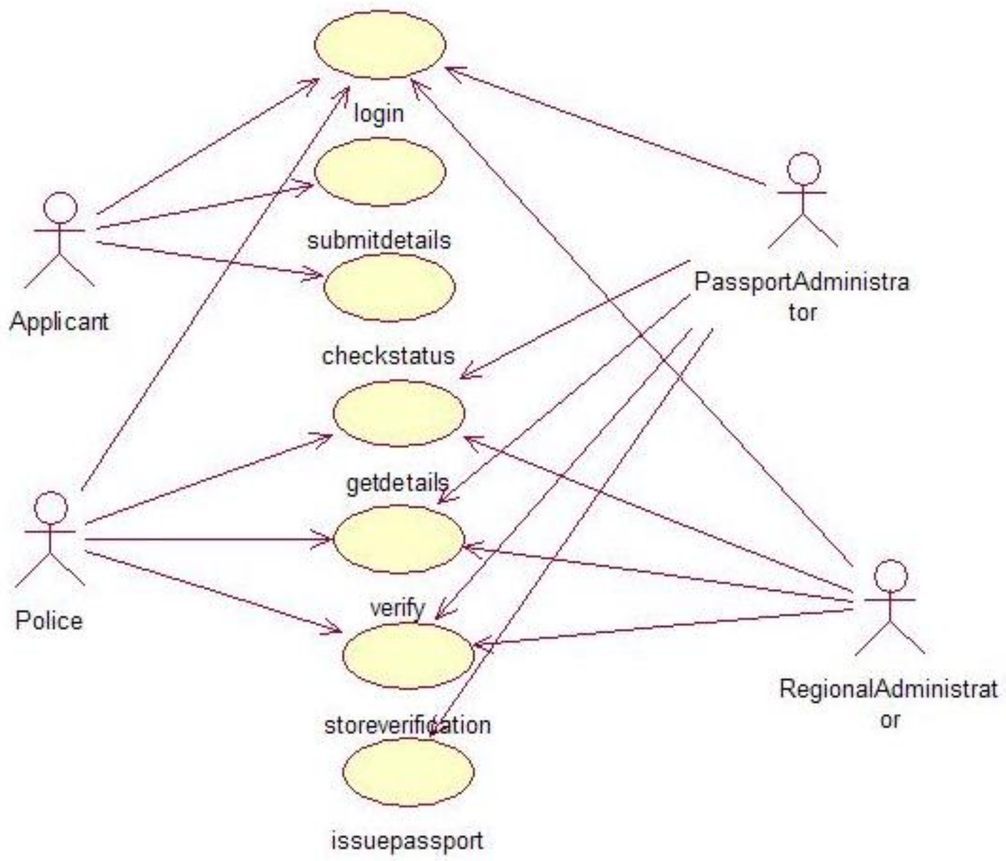
+ Event can occur zero to one time

d) Identify & analyze domain classes



e) Represent Use cases & Domain class diagram using Umbrello

DOCUMENTATION OF USECASE DIAGRAM



a. The actors in use case diagram are Applicant, regional administrator, database, passport Administrator, Police.

b. The use cases are Login, givedetails, logout, collectdetails, verification, issue.

c. The actors use the use case are denoted by the arrow

d. The login use case checks the username and password for applicant, regional administrator, passport administrator and police.

e. The submit details use case is used by the applicant for submitting his details

f. The check status use case is used by the applicant for checking the status of the application process.

g. The get details, verify and store verification use case is used by passport administrator, regional administrator, and police.

h. The details use case is used for getting the details form the database for verification

2. The verify use case is used for verifying the details by comparing the data in the database.

a. The store verification use case is to update the data in the database

b. And finally the issue passport use case is used by the passport administrator for issuing passport who's application verified successfully by all the actor .

CLASSDIAGRAM

A class is drawn as rectangle box with three compartments or components separated by horizontal lines. The top compartment holds the class name and middle compartment holds the attribute and bottom compartment holds list of operations

DOCUMENTATION OF CLASS DIAGRAM

a. APPLICANT-The applicant has attribute such as name and password and operations are login, give details and logout. The applicant login and fill the details that are required for applying the passport .After applying the person can view the status of the passport verification process

b. THE DATABASE-The database has attributed such as name and operation is store. The purpose is to store the data.

C.REGIONAL ADMINISTRATOR- The regional administrator has attribute such as name and operation are get details, verify details and send. The regional administrator gets the details form database and verify with their database

d. PASSPORT ADMINISTRATOR-The passport administrator has attributed such as name and operation are get details, verify details and issue. The passport administrator gets the details form database and verifies with their database, update the verification and issue the passport

e. THE POLICE-The police has attribute such as name and operation are get details, verify details and send. The police get the details form database and verify with their database, update the verification in the database

f) Develop CRUD matrix to represent relationships between use cases and problem domain classes

CRUD technique – an acronym for Create, Read/Report, Update, and Delete; a technique to

validate or refine use cases

- The CRUD Matrix is an excellent technique to identify the Tables in a Database which are used in any User interaction with a Web Site.

- CRUD means 'Create, Read, Update or Delete', and the CRUDD Matrix identifies the Tables involved in any CRUD operation.
- It is very valuable to combine a CRUD Matrix with the analysis of possible User Scenarios for the Web Site
- The analysis helps to identify any Tables which are not used, and any Tables which are used heavily, and may therefore be a performance bottleneck.

	Passenger	Booking	Reservation	train
Request for seat available	C	R,U	R	
Make Reservation	C		R,C	
Cancellation	D			
Check Status	R			
Don't Care	U	C,D	U,D	C,R,U,D

Week-5 & 6

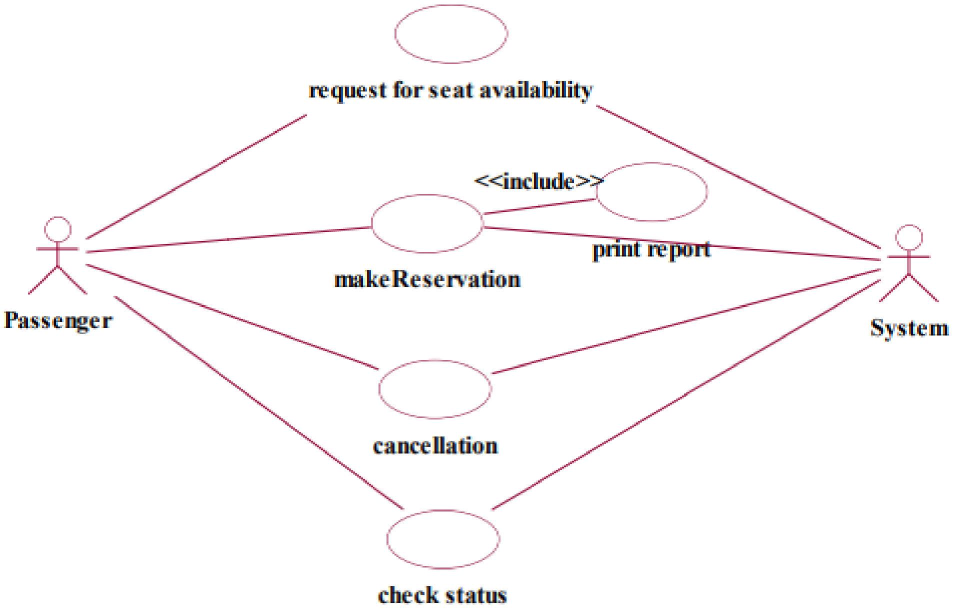
a) Develop Use Case diagrams

b) Develop elaborate Use case descriptions & scenarios

c) Develop Prototypes

d) Develop system sequence diagrams.

a) Use case Diagram



b) Elaborated use-cases descriptions

USE-CASE DIAGRAM

The online ticket reservation system uses the following use cases:

1. Request for seat availability
2. Make Reservation
3. Cancellation
4. Check status
5. Print ticket

ACTORS INVOLVED:

1) System
2) Passenger

USE-CASE NAME: REQUEST FOR SEAT AVAILABILITY

The passenger can view the train available in the database for deciding which train ticket he wishes to reserve. The passenger can search the train information based on journey date, train type and reservation type. The passenger can view the details of flightssuch as, train number, source station, destination station, arrival time, departure time, fare and number of seats available.

USE-CASE NAME: MAKERESERVATION

The user is allowed to reserve a ticket on train as he/she requires on the particular date and time. The user has to provide details such as name, train number, date of travel,source station, destination station, proof name and money transaction details.

USE-CASE NAME: PRINT TICKET

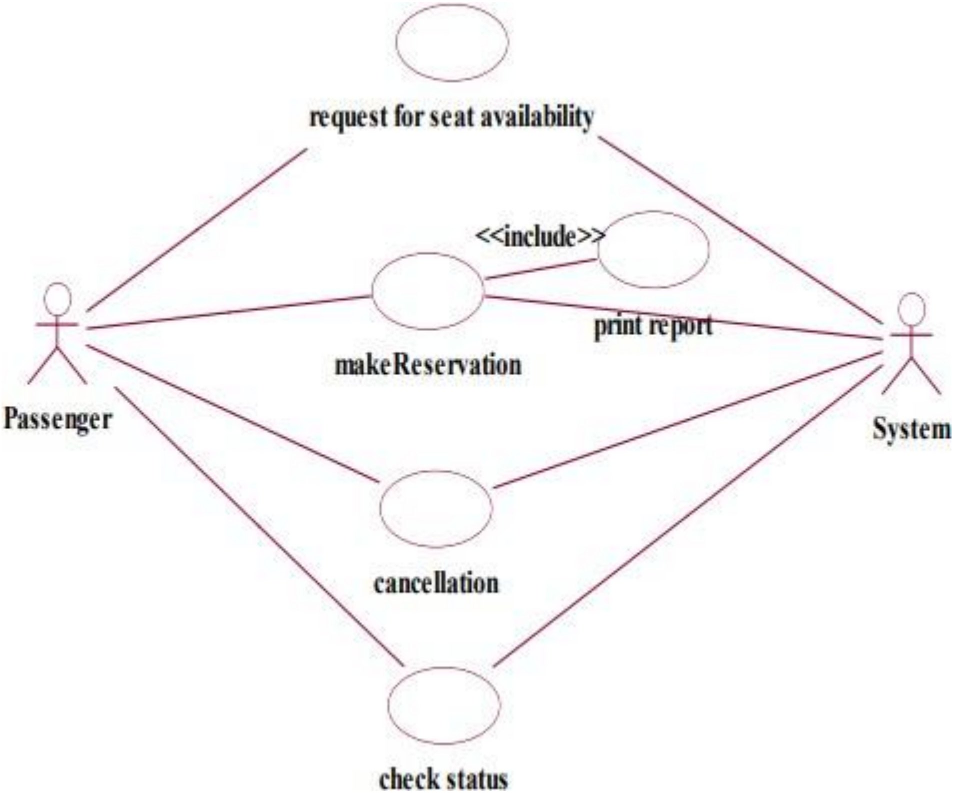
The user after booking a ticket can print a copy of the ticket reserved. The user has to provide the details about ticket number for searching in the database and passengername for confirming passenger identity.

USE-CASE NAME: CANCEL TICKET

A passenger can decide to cancel a ticket after the ticket is booked. The passenger has to provide details about ticket for searching and details about him for confirmation ofidentity.

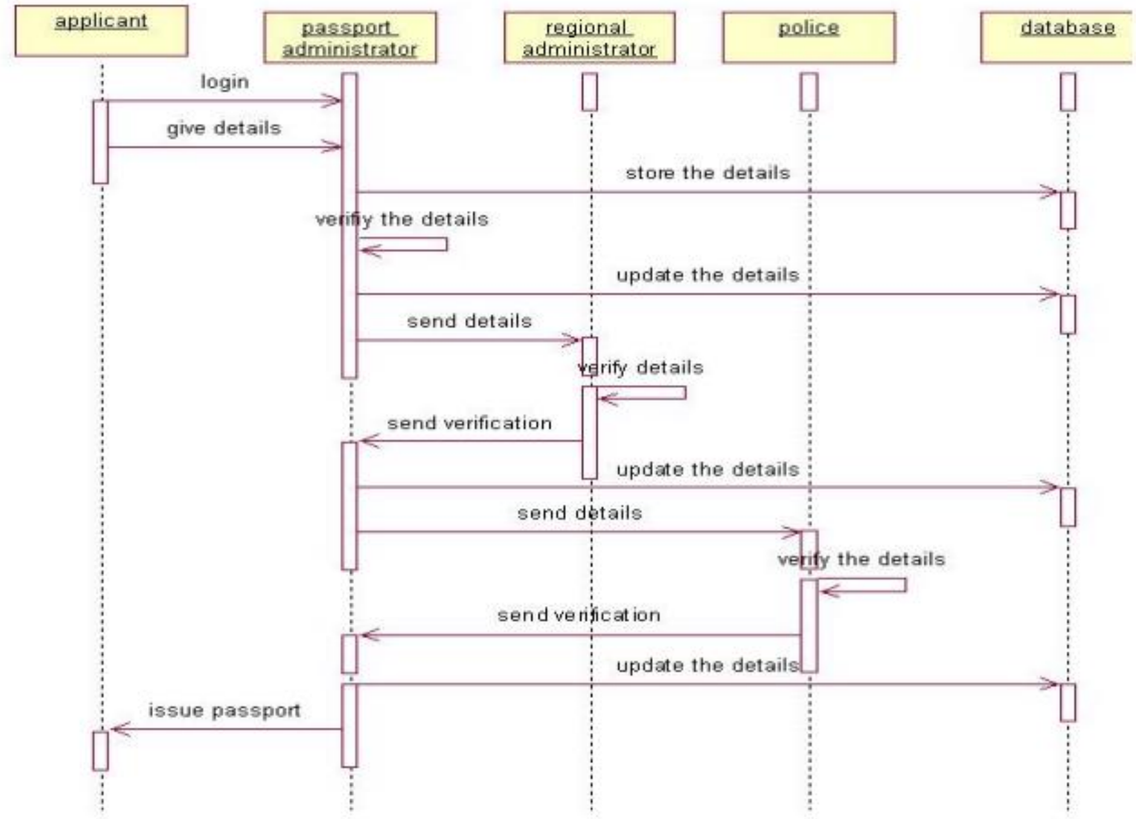
USE-CASE NAME: CHECK STATUS

The passenger can view the status of the reserved tickets. So the passenger canconfirm his/her travel.



d) Develop system sequence diagrams.

A sequence diagram shows an interaction arranged in time sequence, It shows object participating in interaction by their lifeline by the message they exchange arranged in time sequence. Vertical dimension represent time and horizontal dimension represent object.



DOCUMENTATION OF SEQUENCE DIAGRAM.

a. The applicant login the database and give his details and database store the details.

b. The passport administrator get the details from the database and do verification and the forward to regional administrator.

c. The regional administrator get details form passport administrator and perform verification and send report to passport administrator.

d. The police get the details form passport administrator and perform verification and send report to passport administrator

Week 7, 8, 9 & 10:

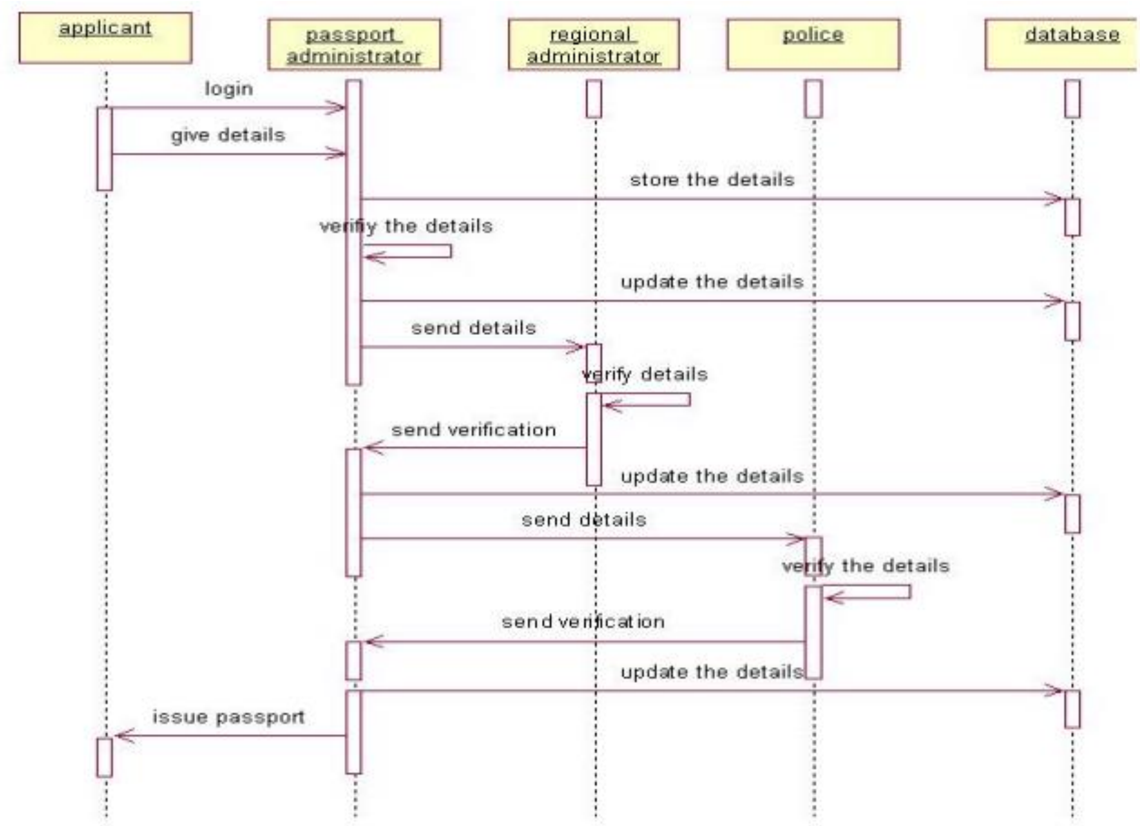
For each case study:

1) Develop high-level sequence diagrams for each use case
2) Identify MVC classes / objects for each use case
3) Develop Detailed Sequence Diagrams / Communication diagrams for each use case showing interactions among all the three-layer objects
4) Develop detailed design class model (use GRASP patterns for responsibility assignment)

5) Develop three-layer package diagrams for each case study

.

1. high-level sequence diagram for E-Ticketing



Week 11 & 12:

For each case study:

1) Develop Use case Packages
2) Develop component diagrams
3) Identify relationships between use cases and represent them
4) Refine domain class model by showing all the associations among classes

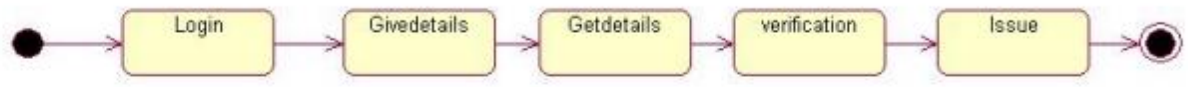
COMPONENT DIAGRAM:

Week13onwards: For each case study:

1) Develop sample diagrams for other UML diagrams – state chart diagrams, activity diagrams and deployment diagrams.

STATE CHART DIAGRAM

The state chart diagram contains the states in the rectangle boxes and starts in indicated by the dot and finish is indicated by dot encircled. The purpose of state chart diagram is to understand the algorithm in the performing method



DOCUMENTATION OF STATE CHART DIAGRAM

a. The states of the passport automation system are denoted in the state chart diagram

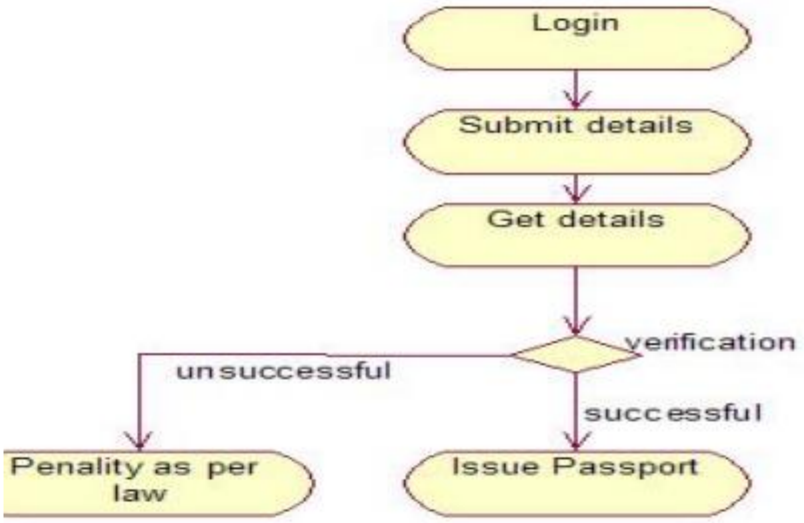
b. Login state represent authentication for login the passport automation system.

c. In this state, it checks whether the applicant has provided all the details that is required.

d. Police, regional administrator and passport administrator get necessary details and verification of the applicant are denoted from the Get detail state and verification state

ACTIVITY DIAGRAM

An activity diagram is a variation or special case of a state machine in which the states or activity representing the performance of operation and transitions are triggered by the completion of operation. The purpose is to provide view of close and what is going on inside a use case or among several classes. An activity is shown as rounded box containing the name of operation


Login
Submit details
Get details
verification
unsuccessful
successful
Penality as per law
Issue Passport