



Respect

Excellence

Integrity

Leadership



FEASIBILITY REPORT

UNVIERSTY MANAGEMENT SYSTEM

Paschim Vihar, East
NEW DELHI



National Project Management System
Business Projects-IT-Enabled
Feasibility Phase

Authority Signatures

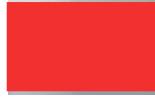
| | | | |
|---------------|---|------------------------|------------|
| Prepared by: |  | Signature | |
| Please print: |  | Project-Analyst | 20/07/2020 |
| | Name | Position | Date |
| Prepared by: |  | Signature | |
| Please print: |  | Co-founder | 20/07/2020 |
| | Name | Position | Date |
| Prepared by: |  | Signature | |
| Please print: |  | Manager | 20/07/2020 |
| | Name | Position | Date |

Table of Contents

| | |
|--|-----------|
| Authority Signatures..... | ii |
| 1 Executive Summary..... | 4 |
| 2 Problem Statement..... | 4 |
| 3 Project Business Requirement..... | 4 |
| 4 Software Description..... | 5 |
| 5 Requirements..... | 5 |
| 6 Cost Estimation for implementation..... | 6 |
| 7 Future Scope..... | 6 |

1 Executive Summary

The reason for a University Management System is to facilitate the use and measure of work at the college. It assists with keeping up a solitary record of the apparent multitude of subtleties of the subsidiary universities and their information base. This item keeps up a solitary information base for various divisions like assessment cell, confirmation cell and so forth to maintain a strategic distance from excess. It helps keeps up the consistency and a legitimate usefulness of the administration area.

- Management of such a large amount of data is impossible in handwritten manuscripts or some file formats.
- Storing such a huge data in the above mentioned way causes issues like data misplacing, time wastage in finding the data when required urgent, etc.
- For further analysis many more options of storing database, teams for implementing the whole software and low cost procedure may be added to emphasize and ease more workload.

2 Problem Statement

Management of such a lot of information is unthinkable in manually written original copies or some document designs. Storing such a gigantic information in the previously mentioned manner causes issues like information losing, time wastage in finding the information when required earnest, and so on. For further investigation a lot more alternatives of putting away information base, groups for actualizing the entire programming and minimal effort strategy might be added to underline and straightforwardness more outstanding task at hand.

3 Project Business Requirement

The Product has a colossal degree in the advanced period since these days everybody needs to get digitalized. Everybody needs to accomplish the degree of precision and needs their framework to control all the orders for them. This Product is demonstrated useful for all the colleges to keep up their information base so as to eliminate excess and botches. This single programming will be sufficiently competent to deal with all the information base of the whole college and the other partnered schools information base to keep up records in various areas like the affirmation, assessment furthermore, different undertakings.

4 Software Description

UNIVERSITY MANAGEMENT SYSTEM deals with the maintenance of University data, records, instructions, and students information within the University. UMS is an automation system, which is used to store the information, students' record, and information of courses. Starting from registration of a new student in the college, it maintains all the details regarding the students', instructors' and course details. The project collects related information from all the departments of an organization and maintains database, which are used to generate reports in various forms to measure Individual and overall performance of the students.

Features of a university management system:-

- University Administration
- University Employee Management
- College Management
- College Administration
- College Employee Management
- Department Management
- Student Management
- Courses Management
- Society Management
- Head of Department Management
- Instructor Management

5 Requirements

- **Performance:** Developer group dealt with the presentation by composing upgraded code and taking care of the information base of the students. They attempt to give flexibility, accessibility, compactness and ease of use to the product.
- **Security:** The product is made keeping the security in mind. The other individual subtleties are additionally secured by programmers. The students' information will be sheltered from the gatecrashers and programmers. On the off chance that disappointment of programming occurs, all the information will be spared and recovered upon the restart of the application.
- **Storage:** The storage of a large data of the students will require Cloud database. Here in it'll be easy to add, delete, update directly into server from anywhere by the owner. Changes will only be made with proper proofs.
- **Accessibility:** This will be provided to faculties as well as the students as viewers only. Editing access will be with the Admin only with owner tag. This will make sure not fair play between Admin and others.

6 Cost Estimation for implementation

The product sent accompanies an assortment of groups that dealt with the task for the final output software. Beginning from planning to building up the product is a joined exertion of the apparent multitude of groups.

Table -Cost Estimation Table

| S.NO | Team | Work Description | Expenditure (₹) |
|-------------------|-----------------|---|---------------------|
| 1 | Creative Team | Creating eye catching thoughts for crowds/clients. | 95,000 |
| 2 | Marketing Team | Arranging efforts to advance items and administrations to clients | 1,20,000 |
| 3 | Designing Team | Drawing and planning of the required design. | 1,00,000 |
| 4 | Developing Team | Planning the product item in view of end-clients. | 1,30,000 |
| 5 | Testing Team | Troubleshooting the product to guarantee the no disappointment | 90,000 |
| TOTAL COST | | | Rs. 5,35,000 |

7 Future Scope

With all the features we have introduced, many more features like in colleges where attendance management is strict that can be added, whoever gives e-payment there data can be stored separately. Also, more teams can be introduced and price reduction can be made possible. We can improve accessibility by introducing Android or iOS applications of this software.

Software Requirements Specification

for

University Management System

Version 1.1

Prepared by: [REDACTED]



Bharati Vidyapeeth's College of Engineering, New Delhi
Date: 27th July 2020

Table of Contents

| | |
|---|-----------|
| Table of Contents | ii |
| Revision History | ii |
| 1. Introduction..... | 1 |
| 1.1 Purpose..... | 1 |
| 1.2 Intended Audience and Reading Suggestions | 1 |
| 1.3 Product Scope | 1 |
| 2. Overall Description..... | 1 |
| 2.1 Product Perspective..... | 1 |
| 2.2 Product Functions | 1 |
| 2.3 User Classes and Characteristics | 2 |
| 2.4 Operating Environment..... | 2 |
| 2.5 Design and Implementation Constraints | 3 |
| 2.6 Assumptions and Dependencies | 3 |
| 3. External Interface Requirements | 3 |
| 3.1 User Interfaces | 3 |
| 3.2 Hardware Interfaces | 4 |
| 3.3 Software Interfaces | 4 |
| 3.4 Communications Interfaces | 4 |
| 4. Other Nonfunctional Requirements | 4 |
| 4.1 Performance Requirements | 4 |
| 4.2 Safety Requirements | 4 |
| 4.3 Security Requirements | 5 |
| Appendix A: Glossary..... | 5 |

Revision History

| Name | Date | Reason for Changes | Version |
|------------|----------------------------|--|---------|
| [REDACTED] | 27 th July 2020 | Created the first SRS document for the university management system. | 1.0 |
| [REDACTED] | 30 th July 2020 | Some minor changes existed in the document which was needed to be corrected. | 1.1 |

1. Introduction

1.1 Purpose

The purpose of this product is to ease the usage and amount of labour at the university. It helps to maintain a single record of all the details of the affiliated colleges and their database. This product maintains a single database for different sectors like examination cell, admission cell etc. to avoid redundancy. It helps maintains the consistency and a proper functionality of the management sector.

1.2 Intended Audience and Reading Suggestions

The SRS contains the entire description of the University management system. This contains all the information for the developers, project managers, marketing staffs, users, testers and documentation writers. The SRS is divided into five sections. Section – 1 consists of the introduction to the project and the documentation of the entire SRS. Section – 2 contains the overall description of the project for all the departments of the people involved followed by section – 3 which comprises of external interface requirements by users, developers etc. Lastly, Section – 4 consists of the non – functional requirements required for apart from all the other functionalities for the completion of the entire project.

1.3 Product Scope

The Product has a huge scope in the modern era because nowadays everyone wants to get digitalized. Everyone wants to attain the level of accuracy and wants their system to control all the commands for them. This Product is proven beneficial for all the universities to maintain their database in order to remove redundancy and mistakes. This single software will be capable enough to manage all the database of the entire university and the other affiliated colleges database to maintain records in different sectors like the admission, examination and other affairs.

2. Overall Description

2.1 Product Perspective

The Product comprises of three sectors – administration, students and staff members which includes all the faculty members of different branches and programs. The administration sector will be having access to all the staff information, student information of the university as well as the affiliated colleges. This sector will also be responsible to handle all the information of the examination and admission cell. It would maintain a database of all the affiliated colleges and their proper information. This sector would also keep a check on the events and societies programs happening to maintain a record of all the ongoing functionalities.

2.2 Product Functions

The Product will be capable of performing a majority of functions. Following is the list of the major functions the product will let the user perform:

1. The administration sector will be able to access to all the information of the staff, students and the university as well. They will be provided with the access of alter, delete and update any information regarding the system.

2. The staff sector which includes all the faculty members, Head of the departments of different branches and programs will be provided with an access to look the information of all the students and their ranking in the desired duration. They would also be provided with an access to record the performance of their students and compute their results after marking them.
3. The student sector will have the least freedom in this product to access a variety of stuff. The students could see their rankings both at university as well as at the college level. They will also be able to access their class information in order to see their other classmates and related information.

Figure – 1 shows the overall functioning of the software product.

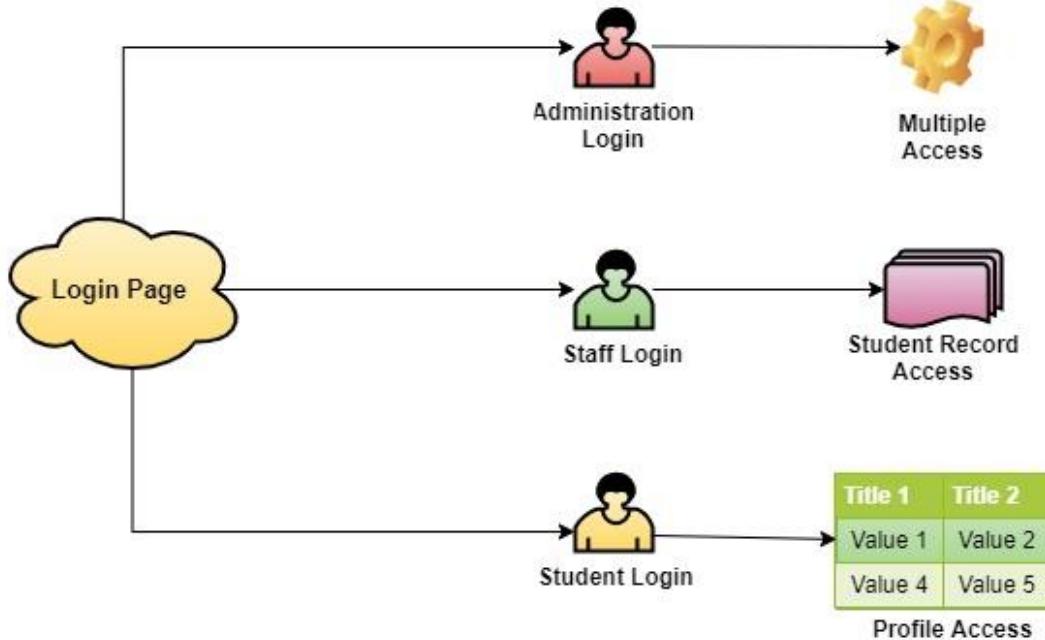


Figure 1: Overall functionality of the Product

2.3 User Classes and Characteristics

There are majorly three user classes. Following are the three user classes:

1. **Administration user class:** This sector will be operated by all the administration staff of the University. The users involved in this class are all the administrative employees, fee collectors, examination cell staff, admission staff, reception staff and all the other employees that are involved in the management sector of the university.
2. **Staff user class:** This sector involves all the faculty members of different branches and programs. All the faculty members including the principal, head of the department, professors etc. who are involved in the teaching process of the university as well as the affiliated colleges. They will have the access to all the students registered and will be capable to access their marksheets and score them based on their performance.
3. **Student user class:** This sector comprises of the least freedom to access different sectors. The users involved in this class are the students of the university and different affiliated colleges. The students will not have majority of access. They will only be able to see their ranking at university and college level including the students of his/her respective branch and program.

2.4 Operating Environment

The software will be entirely based on HTML, JavaScript and CSS. Therefore, the operating environment should be preferable a windows operating system with the latest version of chrome, Firefox or internet explorer.

2.5 Design and Implementation Constraints

Following are the design and implementation constraints in the product:

1. The complete access will be provided to the administrative staff and the least access will be granted to the registered students.
2. The time complexity of the software should be kept minimum. So, that memory management i.e., a huge amount of database could be recorded easily.
3. The maintenance of the system once made will entirely be dependent on the user.
4. In case the system crashes while working, the developer should have a backup plan to avoid the loss of the database.
5. Constraints can be modified by the administrative head user. The entire command will be handed over to the main head in chief.

2.6 Assumptions and Dependencies

Below is the list of the assumed factors that could affect the requirements stated in the SRS:

1. There is an extreme chance that the software might collapse, if proper maintenance and updates are not maintained regularly.
2. It is assumed that the software can run at a maximum of 16 hours per day without a stop.
3. The software product is entirely dependent on the user maintenance and the user is hereby responsible to maintain the software product carefully.
4. Backup of the database is herbily requested to be taken time to time in order to avoid mishappening and loss of the database. This could only happen if the window version used is outdated.

3. External Interface Requirements

3.1 User Interfaces

The user interface will be user-friendly. Though a guide tour will be given after the successful installation of the software but the GUI will be user-friendly. All the options and accessible points will be displayed right on the dashboard of the registered person. After the installation of the software, it would first ask the login details from the user, if not than the user can create his or her own account. After the login is successfully completed, the dashboard will be visible from where the user can access the specified areas which are accessible to them as shown in Figure – 2.



Figure 2: Sample of the Launching Page

3.2 Hardware Interfaces

The hardware components needed for this product is not much. This software can be installed on any windows operating system device. Laptops, PC, etc. can be used to access this software.

3.3 Software Interfaces

The software will preferably run more efficiently and precisely on windows operation system. As the entire software product is based on HTML, JavaScript and CSS. Thus, they work accurately on windows operating systems to give a better and accurate results. Otherwise the software is not constrained for any operating system. It can be accessible for any operating system like Mac, Linux, ubuntu etc.

3.4 Communications Interfaces

As this software is purely based on HTML, JavaScript and CSS therefore, the communication protocol used will be HTTP which is considered the best protocol to make requests. The communication protocol also comprises of the sending and receiving of e-mails, messages, notifications etc. which are attached within the software product. The whole application will be operated on the basis of the HTTP protocols which will act as the communication interface.

4. Other Nonfunctional Requirements

4.1 Performance Requirements

The software product must consist of a low time complexity and more memory management software because the software will be storing a huge amount of dataset. Therefore, the data must be managed properly by the system. Also, in case the system fails due to some reason then there should be a system available so that the data do not get lost. The performance of the system should be high. It should be capable enough to run on various platforms at the same time.

4.2 Safety Requirements

Following are the safety requirements which must be followed by the developer while making the software product:

1. **Responsibility:** Companies with strong safety cultures share the value of responsibility. By responsibility, I mean the shared belief that each individual is morally and ethically bound to act responsibly for the good of his fellow employees, his company, and society as a whole. Until every person in your organization believes this, you will struggle with creating a safety culture.
2. **Accountability:** Managers must be held accountable to lead by example each and every day. Managers and supervisors need to understand that their behavior influences everyone around them. Upper management needs to hold these individuals accountable to safety, and not look the other way as long as production goals are being met.
3. **Clear Expectations:** Safety expectations need to be set and communicated to everyone in the organization. The commitment to achieving these goals needs to be demonstrated from the top down. Only when the rank and file employees see this commitment will they start to change.
4. **Ethics:** Ethically driven management systems are important in developing a strong safety culture. The goal is for employees to make decisions that not only satisfy the procedures in the safety manual but

that are also ethical and moral. The individuals you hire for your company should have the ability to make ethical and rational decisions in everyday situations, and share your company's core values.

4.3 Security Requirements

The software product should be highly authenticated. The user ID and password should be protected so that there is no misuse of the information. The product consists of the database of a university therefore, security is the foremost important thing to be kept in mind. Thus, the software must consist of different factors like the "Time-Out" situation for proper functionality and authentication of the system.

Appendix A: Glossary

1. **HTML:** HTML stands for Hyper Text Markup Language. It is the standard markup language for creating Web pages. It describes the structure of a Web page. It consists of a series of elements and the elements tell the browser how to display the content on the screen. HTML elements label pieces of content such as "this is a heading", "this is a paragraph", "this is a link", etc.
2. **JavaScript:** JavaScript, often abbreviated as JS, is a programming language that conforms to the ECMAScript specification. JavaScript is high-level, often just-in-time compiled, and multi-paradigm. It has curly-bracket syntax, dynamic typing, prototype-based object-orientation, and first-class functions.
3. **CSS:** Cascading Style Sheets is a style sheet language used for describing the presentation of a document written in a markup language such as HTML. CSS is a cornerstone technology of the World Wide Web, alongside HTML and JavaScript.
4. **HTTP:** The Hypertext Transfer Protocol is an application layer protocol for distributed, collaborative, hypermedia information systems.

PRACTICAL 3

Aim: To perform the entity relationship diagram for the suggested system

Theory: An Entity Relationship (ER) diagram was introduced by P.P. Chen in 1976.

An ER model is defined as a conceptual data model that views the real world as entities & relationships.

An ER model is useful to a database designer in the following ways:

- 1) An ER model maps well to the relational model i.e., the constructs used in the ER model can be easily transformed into relational tables.
- 2) An ER model can be used by the database designer to communicate the design to the end user.
- 3) An ER model can be used to design plan by the database developers to implement a data model in specific DBMS software.

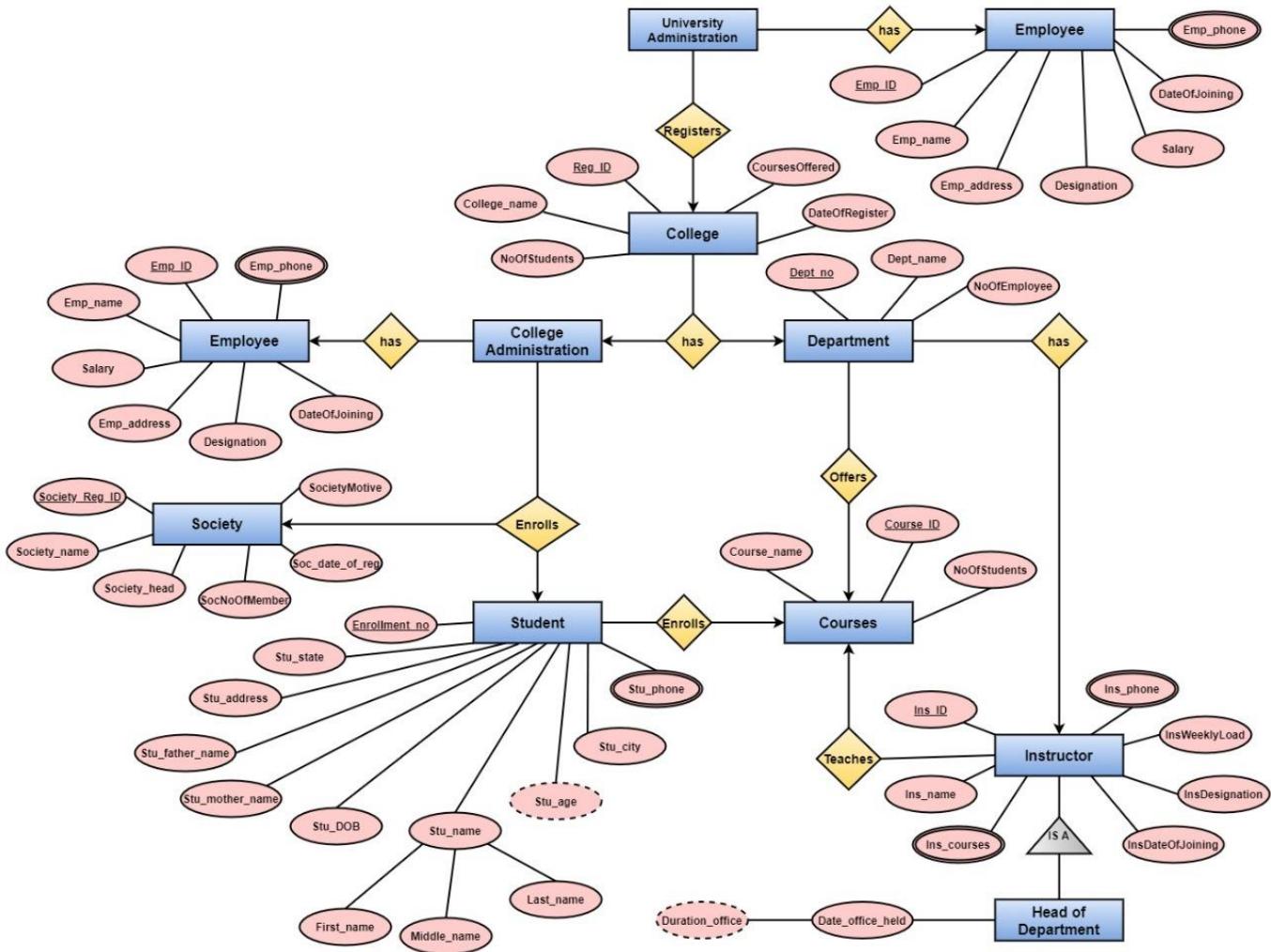
→ Basic Terminologies

- (i) Entity
- (ii) Entity type (or set)
- (iii) Relationships
- (iv) Degree of a relationship
- (v) Connectivity or cardinality
- (vi) Attributes

→ Notations

| | Symbol | Meaning |
|-----|--------|----------------------------------|
| 1. | | Entity set |
| 2. | | Weak entity |
| 3. | | Relationship |
| 4. | | Weak relationship |
| 5. | | Attributes |
| 6. | | Multi-valued attribute |
| 7. | | Primary key attribute |
| 8. | | Composite attribute |
| 9. | | Derived attribute |
| 10. | | Complete participation of entity |

ER Diagram for University Management System:



PRACTICAL – 4

Aim: To perform the user's view analysis for the suggested system: Use Case Diagram.

System: University Management System

Software Requirements: StarUML v3.2.2

Theory:

Use Case Diagram –

- To model a system, the most important aspect is to capture the dynamic behaviour. Dynamic behaviour means the behaviour of the system when it is running/operating. In UML use case diagram is one of the five diagrams which is used to capture the dynamic nature of the subsystem/system by interacting with the internal and external agents known as actors.
- Use case diagrams consist of actors, use cases and their relationships. The diagram is used to model the system/subsystem of an application. A single use case diagram captures a particular functionality of a system. Hence to model the entire system, a number of use case diagrams are used.

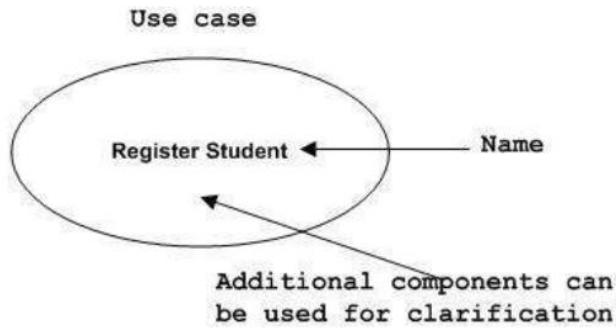
Objectives of a Use case Diagram –

1. Used to gather the requirements of a system.
2. Used to get an outside view of a system.
3. Identify the external and internal factors influencing the system.
4. Shows the interactions among the requirements are actors.

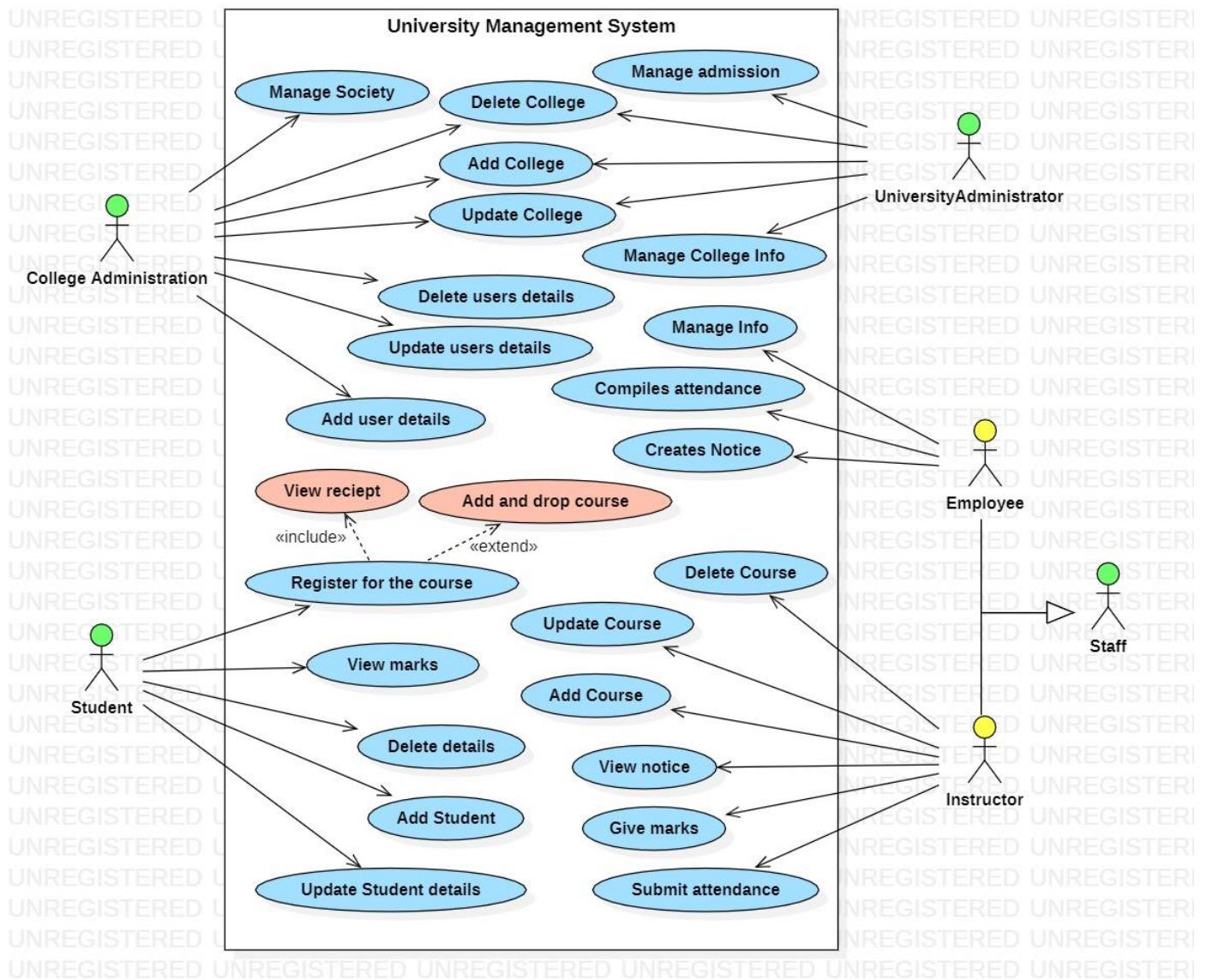
Purpose of Use case Diagram –

1. Use case diagrams are used to gather the requirements of a system including internal and external influences. These requirements are mostly design requirements. Hence, when a system is analysed to gather its functionalities, use cases are prepared and actors are identified.
2. When the initial task is complete, use case diagrams are modelled to present the outside view

Notation:



Use case Diagram of University Management System:



PRACTICAL – 5

Aim: To draw a structural view diagram for the system: Class Diagram

System: University Management System

Software Requirements: StarUML v3.2.2

Theory:

Class Diagram –

- Class diagram is a static diagram. It represents the static view of an application. Class diagram is not only used for visualizing, describing, and documenting different aspects of a system but also for constructing executable code of the software application.
- Class diagram describes the attributes and operations of a class and also the constraints imposed on the system. The class diagrams are widely used in the modelling of object-oriented systems because they are the only UML diagrams, which can be mapped directly with object-oriented languages.
- Class diagram shows a collection of classes, interfaces, associations, collaborations, and constraints. It is also known as a structural diagram.

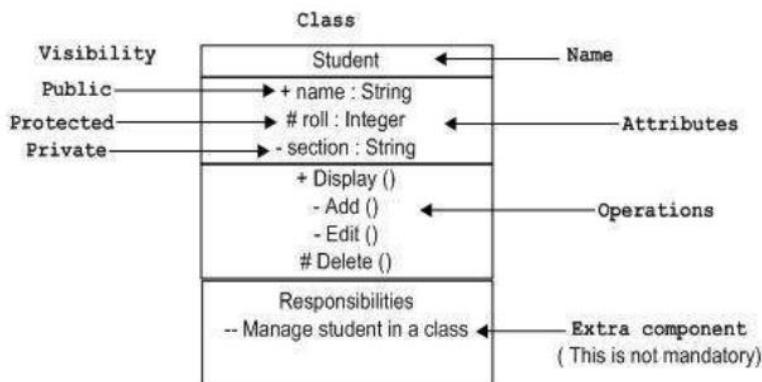
Objectives of a Class Diagram –

1. Analysis and design of the static view of an application.
2. Describe responsibilities of a system.
3. Base for component and deployment diagrams.
4. Forward and reverse engineering.

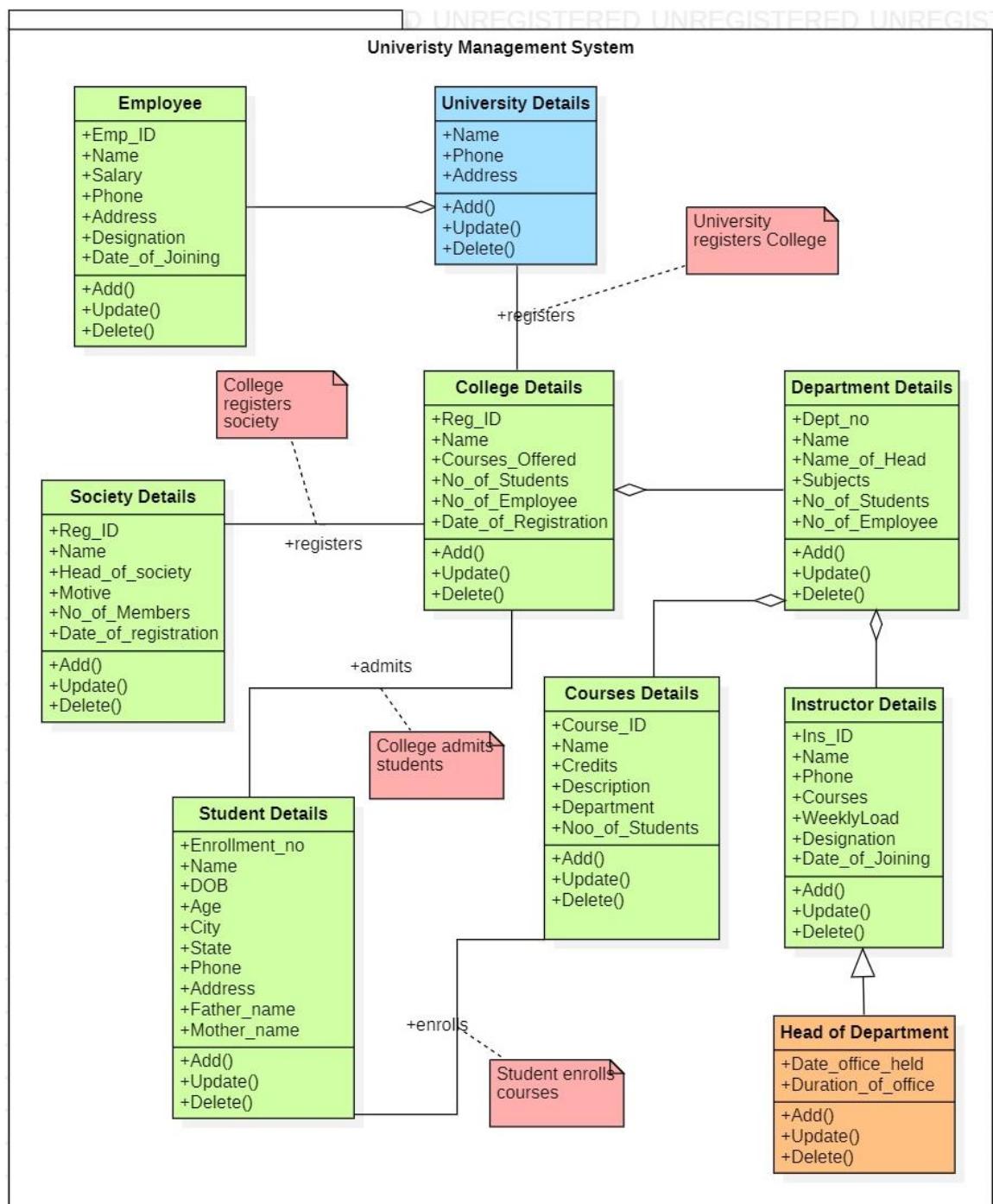
Purpose of Class Diagram –

1. The purpose of class diagram is to model the static view of an application. Class diagrams are the only diagrams which can be directly mapped with object-oriented languages and thus widely used at the time of construction.
2. UML diagrams like activity diagram, sequence diagram can only give the sequence flow of the application; however class diagram is a bit different. It is the most popular UML diagram in the coder community.

Notation:



Class Diagram of University Management System:



PRACTICAL 6

Aim: To perform the function oriented diagram: Data flow Diagram (DFD)

System → University Management System

Software used: StarUML v3.2.2.

Theory:

Data flow diagrams are versatile diagramming tools. With only 4 symbols, data flow diagrams can represent both physical & logical information systems. The four symbols used in DFD representation are data flows, data stores, processes & sources/sinks.

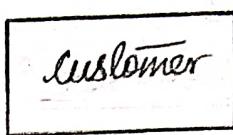
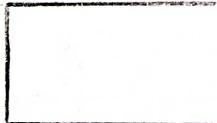
Data flowcharts can range from simple, even hand-drawn process overviews, to in-depth, multi-level DFDs that dig progressively deeper into how the data is handled. They can be used to analyze an existing system or model a new one.

DFD rules

- Each process should have atleast one input & an output.
- Each data store should have atleast one data flow in and one data flow out.
- Data stored in a system must go through a process.
- All processes in a DFD go to another process or a data store.

DFD levels

DFD can dive into progressively more detail by using levels of layers

| Name | Symbol | Description | Example |
|------------|---|---|---|
| Entity |  | Used to represent people & organizations outside the system. They either input information to the system, accept off information from the system or both. |  |
| Process |  | These are actions that are carried out with the data that flows around the system. A process accepts input data & produces data that it passes on to another part of the DFD. |  |
| Data flow |  | These represent the flow of data from or to a process. |  |
| Data Store |  | This is a place where data is stored either temporarily or permanently. |  |

going in one a particular place.

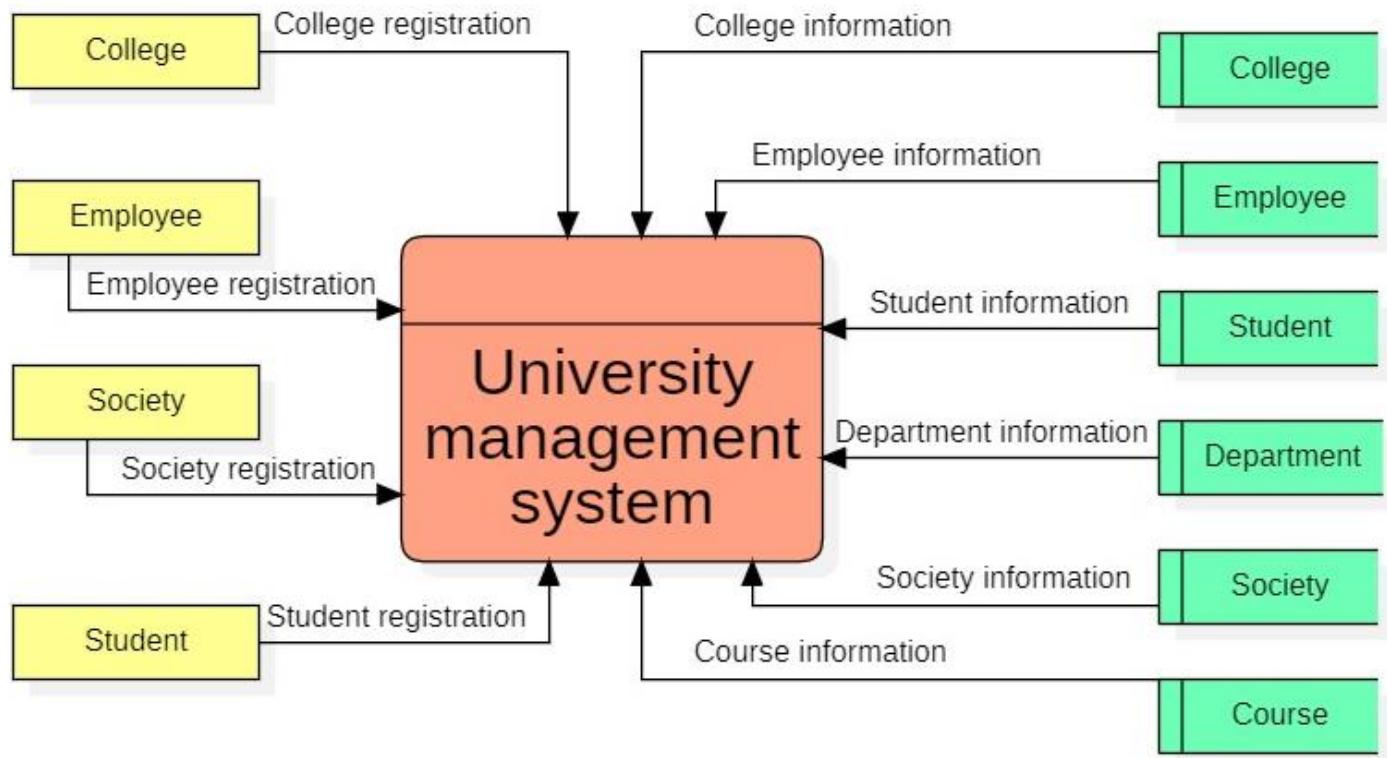
- (i) DFD level 0 is also called a Context Diagram. It's basically an overview of the whole system or process analyzed or modeled. It's designed to be an at-a-glance view, showing the system as a single high-level process, with its relationship to external entities.
- (ii) DFD level 1 provides a more detailed breakout of pieces of the Context Level Diagram. We highlight the main functions carried out by the system, as we breakdown the high-level process of the Context Diagram into its subprocesses.
- (iii) DFD level 2 then goes one step deeper into parts of level 1. It may require more text to reach the necessary level of detail about the system's functioning.

Progression to levels 3,4 and beyond is possible, but going beyond level 3 is uncommon.

Conclusion: Studied & implemented DFD successfully on University Management System.

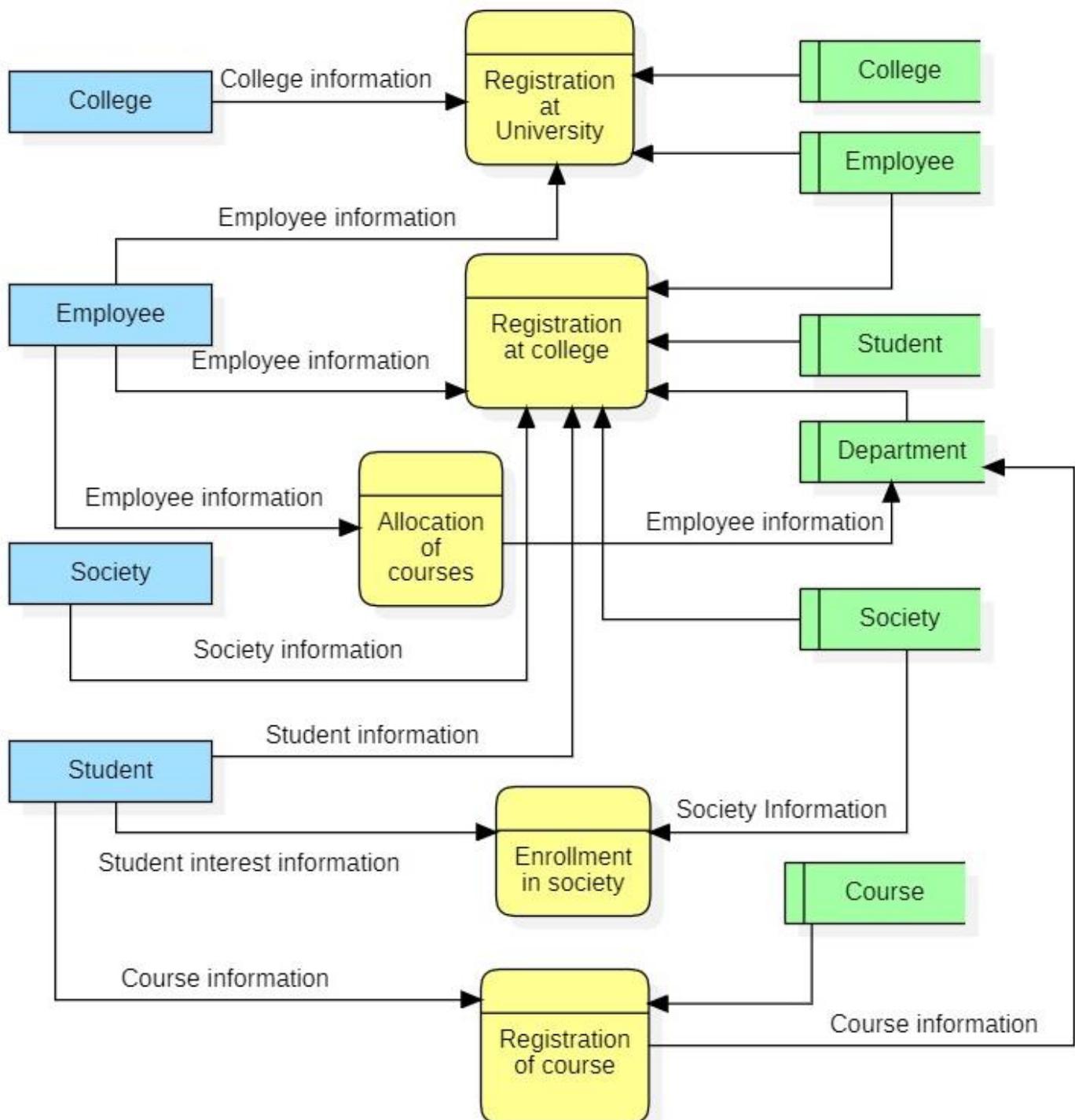
0-LEVELLED DATA FLOW DIAGRAM

0 - Level DFD



1-LEVELLED DATA FLOW DIAGRAM

1 - Level DFD

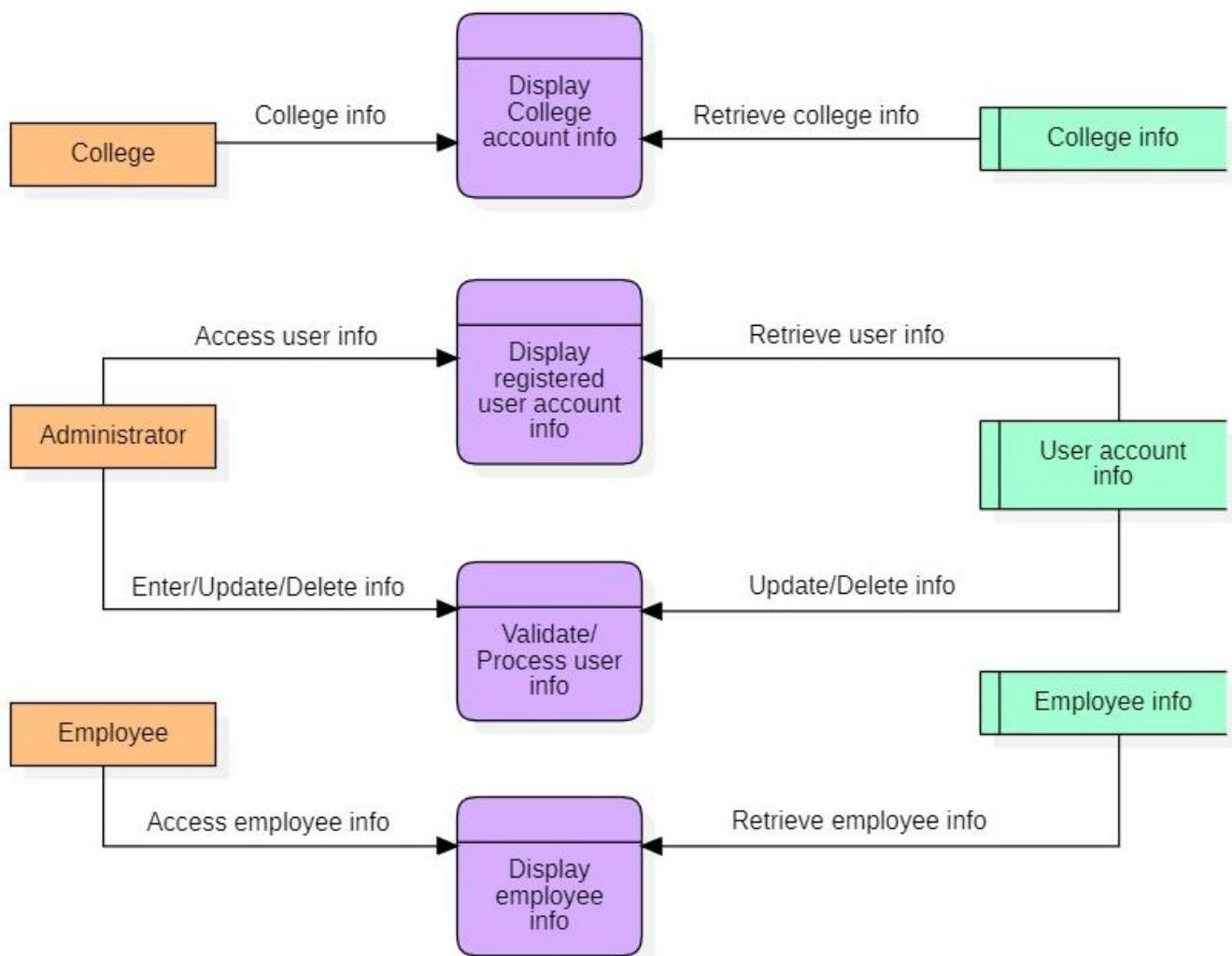


2-LEVELLED DATA FLOW DIAGRAM

1. Registration at University

2 - Level DFD

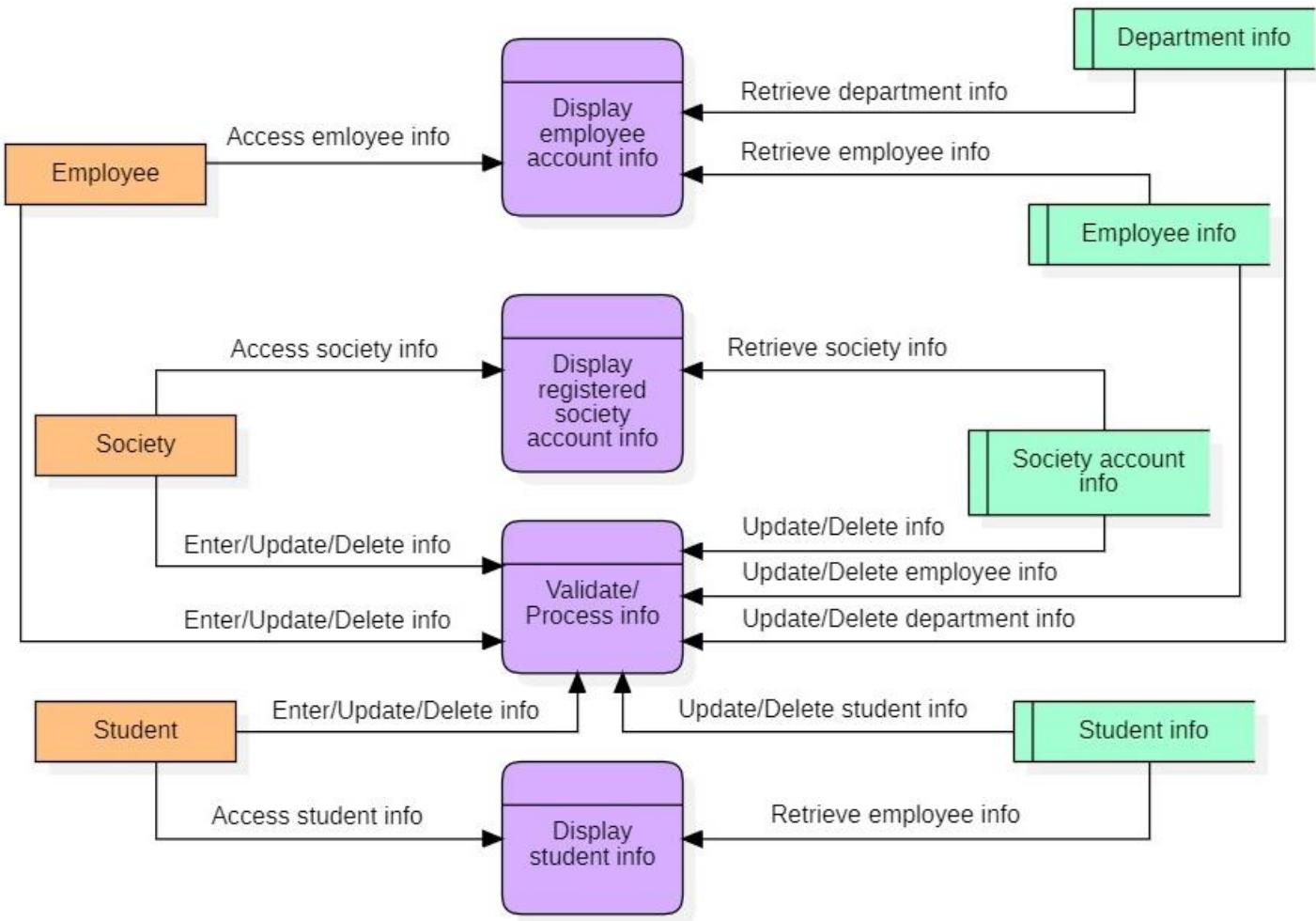
1. Registration at University



2. Registration at College

2 - Level DFD

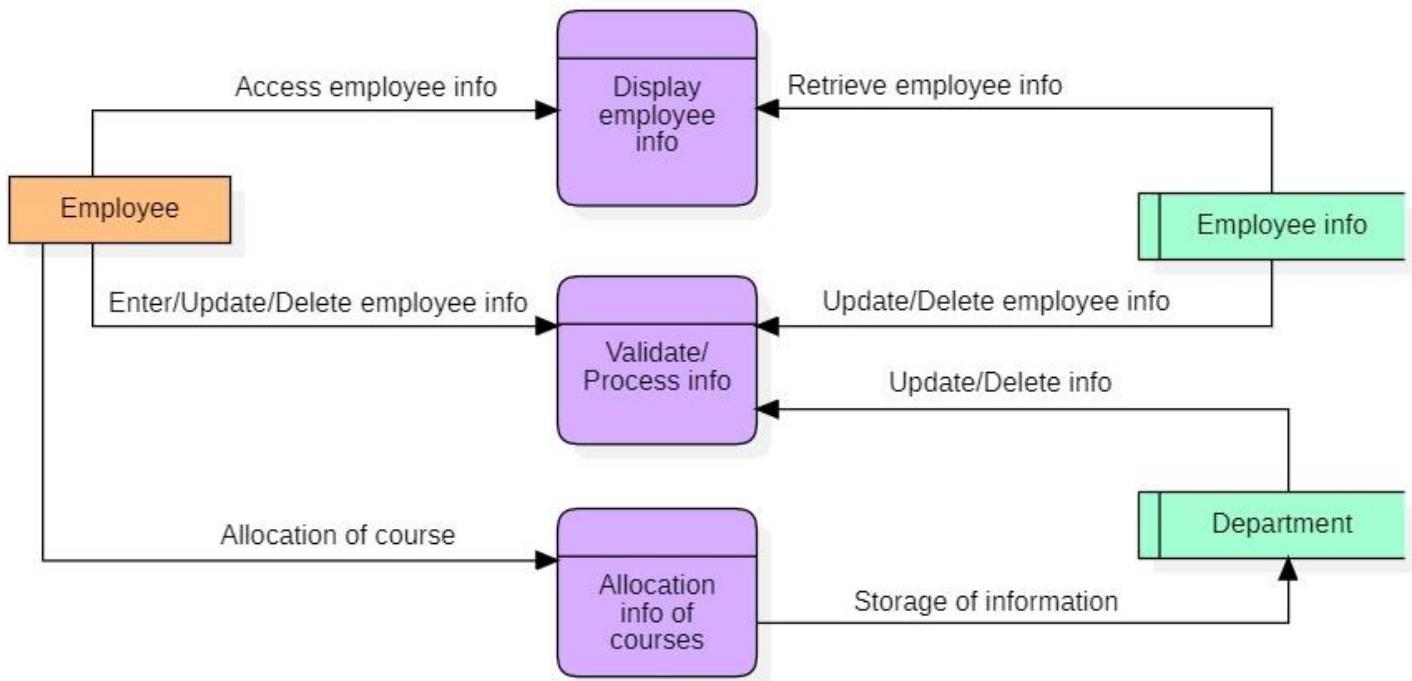
2. Registration at College



3. Allocation of courses

2 - Level DFD

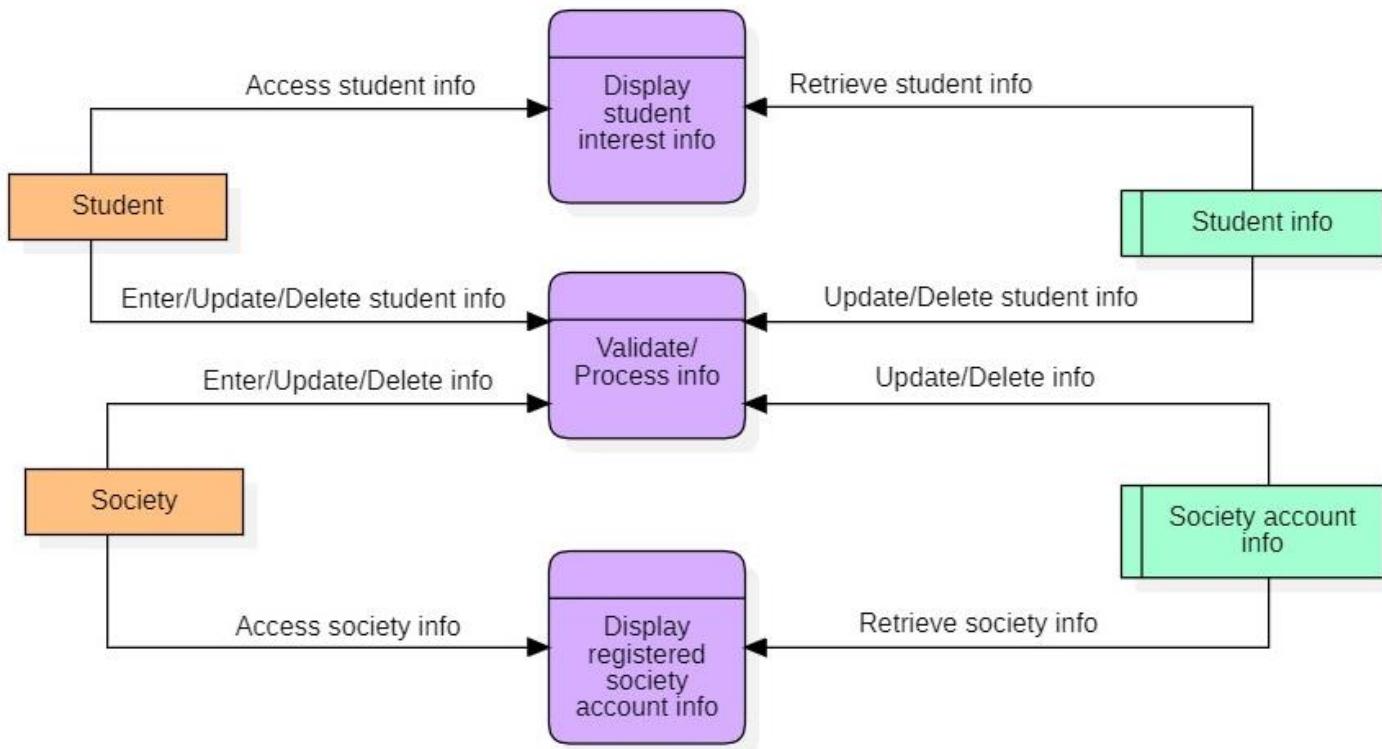
3. Allocation of courses



4. Enrolment in society

2 - Level DFD

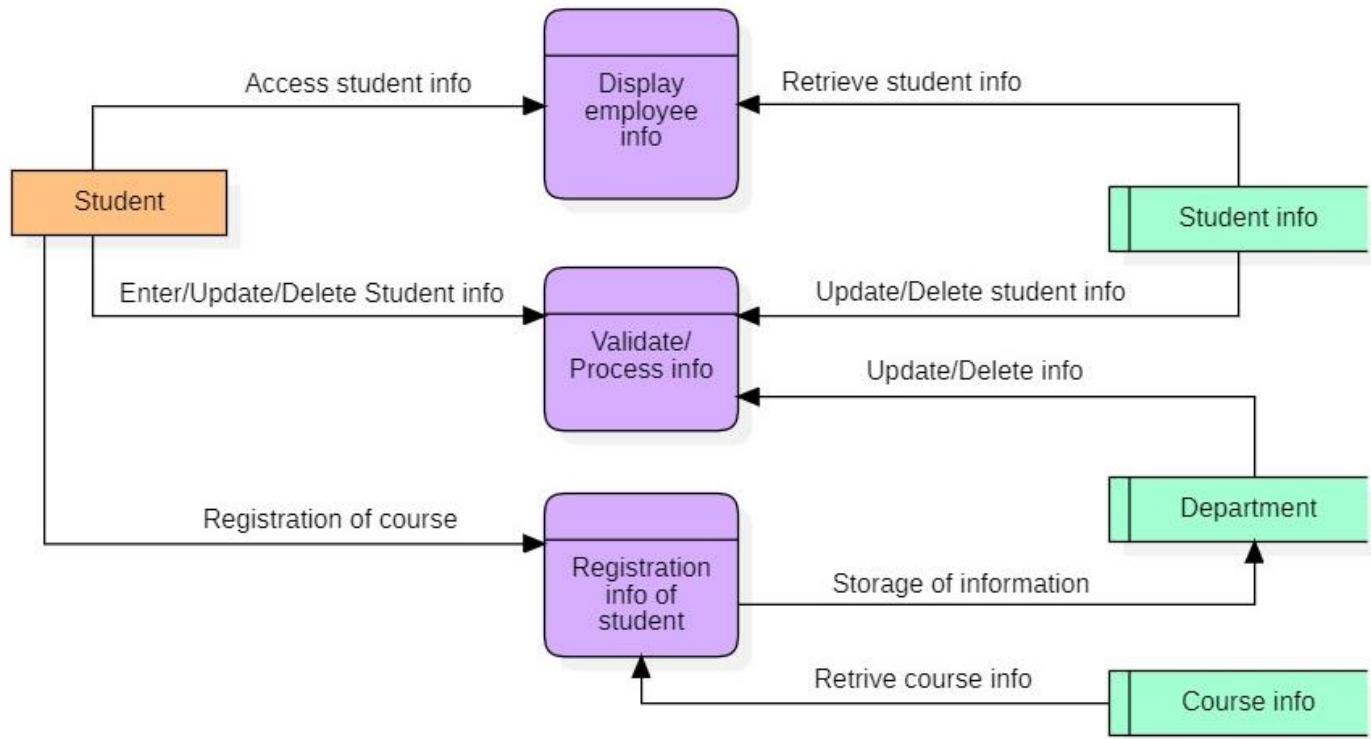
4. Enrollment in society



5. Registration of course

2 - Level DFD

5. Registration of course



PRACTICAL 7

Aim: To draw the behavioural view diagram: State diagram

System → University Management System

Software used: StarUML v3.2.2.

Theory:

A state diagram shows the behaviour of classes in response to external stimuli. Specifically a state diagram describes the behaviour of a single object in response to a series of events in a system. Sometimes it's also known as a Karel state chart machine learning diagram. This UML diagram models the flow of control from state of a particular object within a system.

Purpose of State Diagram

We use state diagrams to illustrate the dynamic view of a system. They are especially important in modeling the behaviour of an interface, class or collaboration.

State diagrams emphasize the event-ordered behaviour of an object, which is especially useful in modeling ~~systems~~ reactive systems.

Basic Notations

1) States

States represent situations during the life of an object.

| State | Notation |
|--|----------|
| Simple state | |
| State with internal activities | |
| Transition | |
| Initial state | |
| Final state | |
| Synchronization and splitting of control | |

2)

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.

3)

Initial state

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

4)

Final state

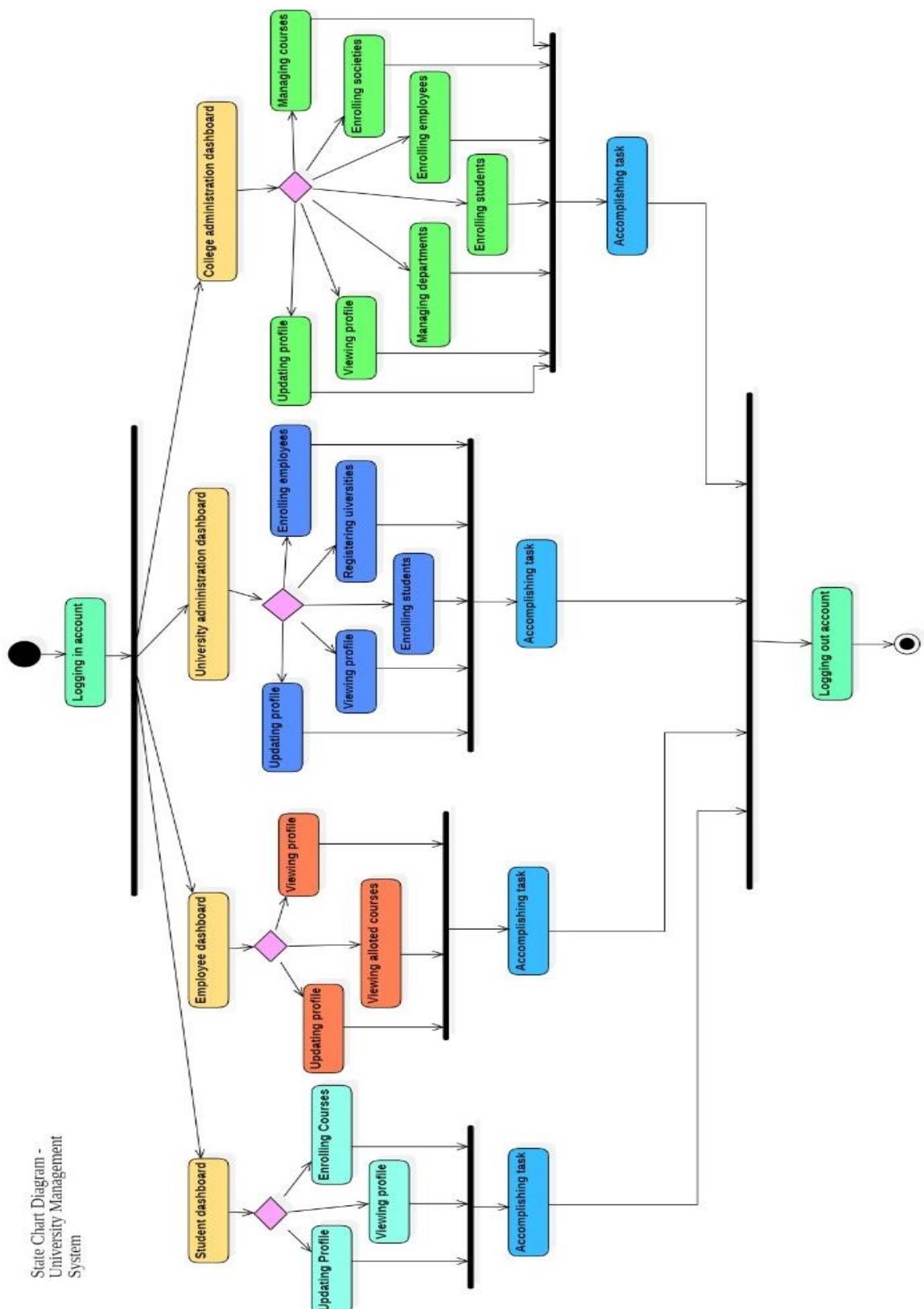
An arrow pointing to a filled circle nested inside another circle represents the object's final state.

5) Synchronization & 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.

Conclusion: Studied of implemented state diagram for university management system successfully.

STATE DIAGRAM FOR UNIVERSITY MANAGEMENT SYSTEM



PRACTICAL 8

Aim: To perform the behavioral view diagram for the suggested system : Sequence diagram.

System → University Management System

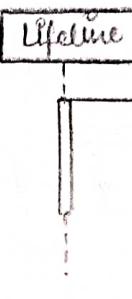
Software used: StarUML v.3.2.2.

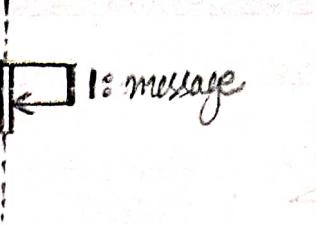
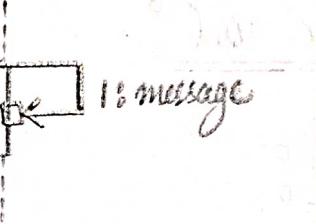
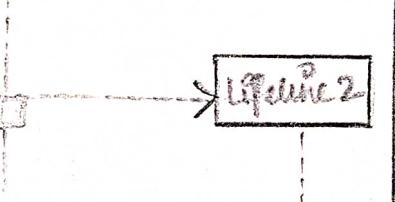
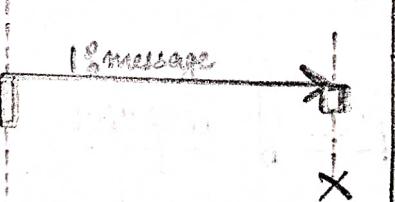
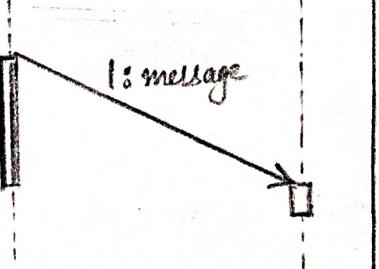
Theory: Sequence diagrams are interaction diagrams that detail how operations are carried out. They capture the interaction between objects in the context of collaboration. Sequence diagrams are time focus and they show the order of the interaction visually by using the vertical axis of the diagram to represent time when messages are sent & when.

Purpose of Sequence diagram

- Model high-level interaction between active objects in a system.
- Model interaction between object instances within a collaboration that realizes an operation or use case.
- Model the interaction between objects within a collaboration that realizes an operation.
- Either model generic interactions or specific instances of a interaction.

Sequence diagrams show elements as they interact over time & they are organized according to object (horizontally) & time (vertically).

| Notation | Representation |
|---|--|
| <u>Actor</u> : A type of role played by an entity that interacts with the subject. |  |
| <u>Lifeline</u> : It represents an individual participation in the interaction. |  |
| <u>Activations</u> : It represents the period during which an element is performing an operation. |  |
| <u>Call message</u> : It is a kind of message that represents an invocation of operation of target lifeline. |  |
| <u>Return message</u> : It is a kind of message that represents the flow of information back to the caller of a corresponding former message. |  |

| Notation | Representation |
|--|--|
| <u>Self message</u> : It is a kind of message that represents the invocation of message of the same lifeline. |  |
| <u>Recursive message</u> : It is a kind of message that represents the invocation of the activator where the message was invoked from. |  |
| <u>Create message</u> : It is a kind of message that represents the instantiation of (target) lifeline. |  |
| <u>Destroy message</u> : It is a kind of message that represents the request of destroying the lifecycle of target lifeline. |  |
| <u>Duration message</u> : It shows the distance between two time instants for a message invocation. |  |

Object dimension

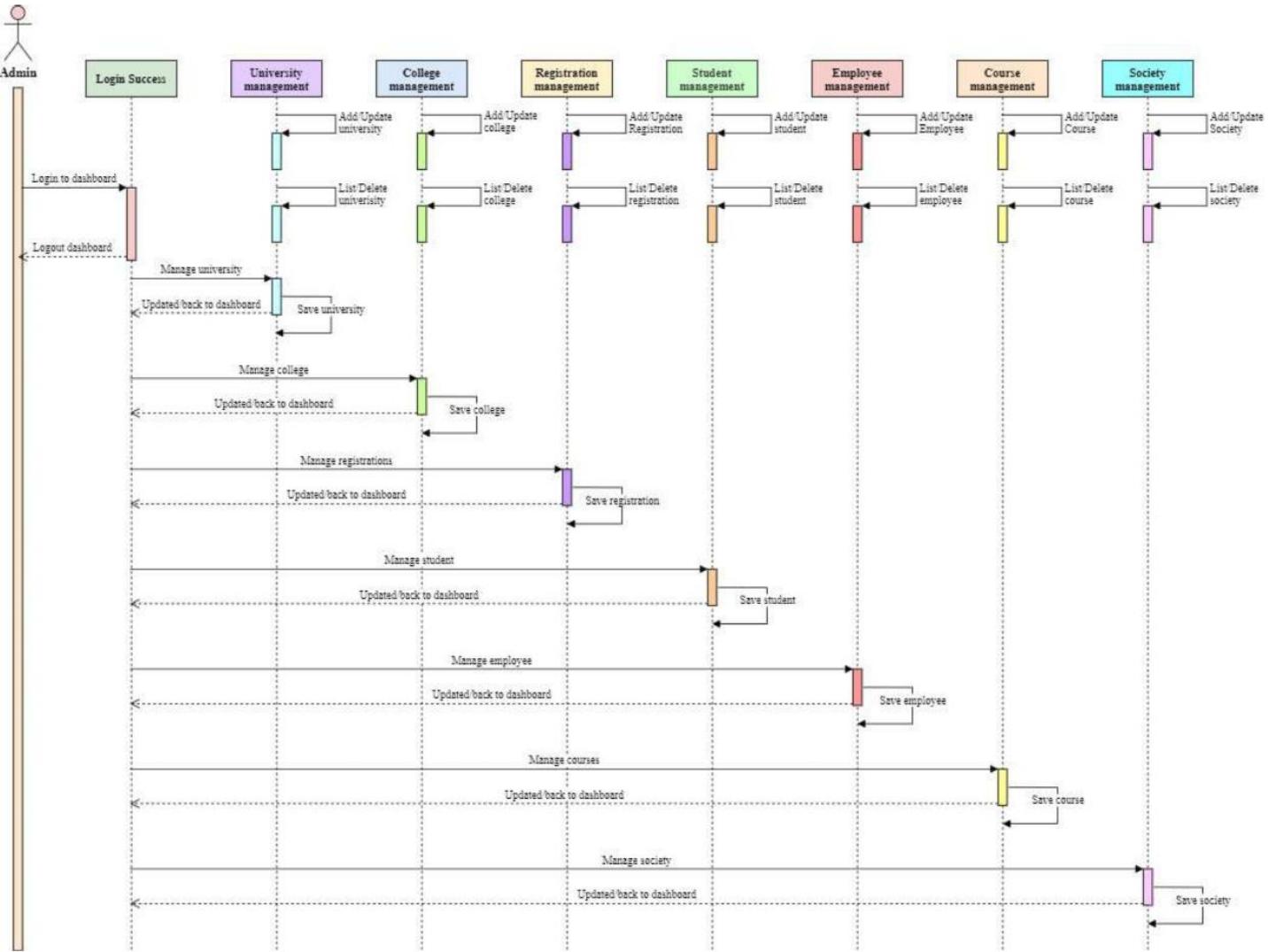
- The horizontal axis shows the elements that are involved in the interaction.
- Conventionally, the objects involved in the operation are listed from left to right according to when they take part in the message sequence.

Time dimension

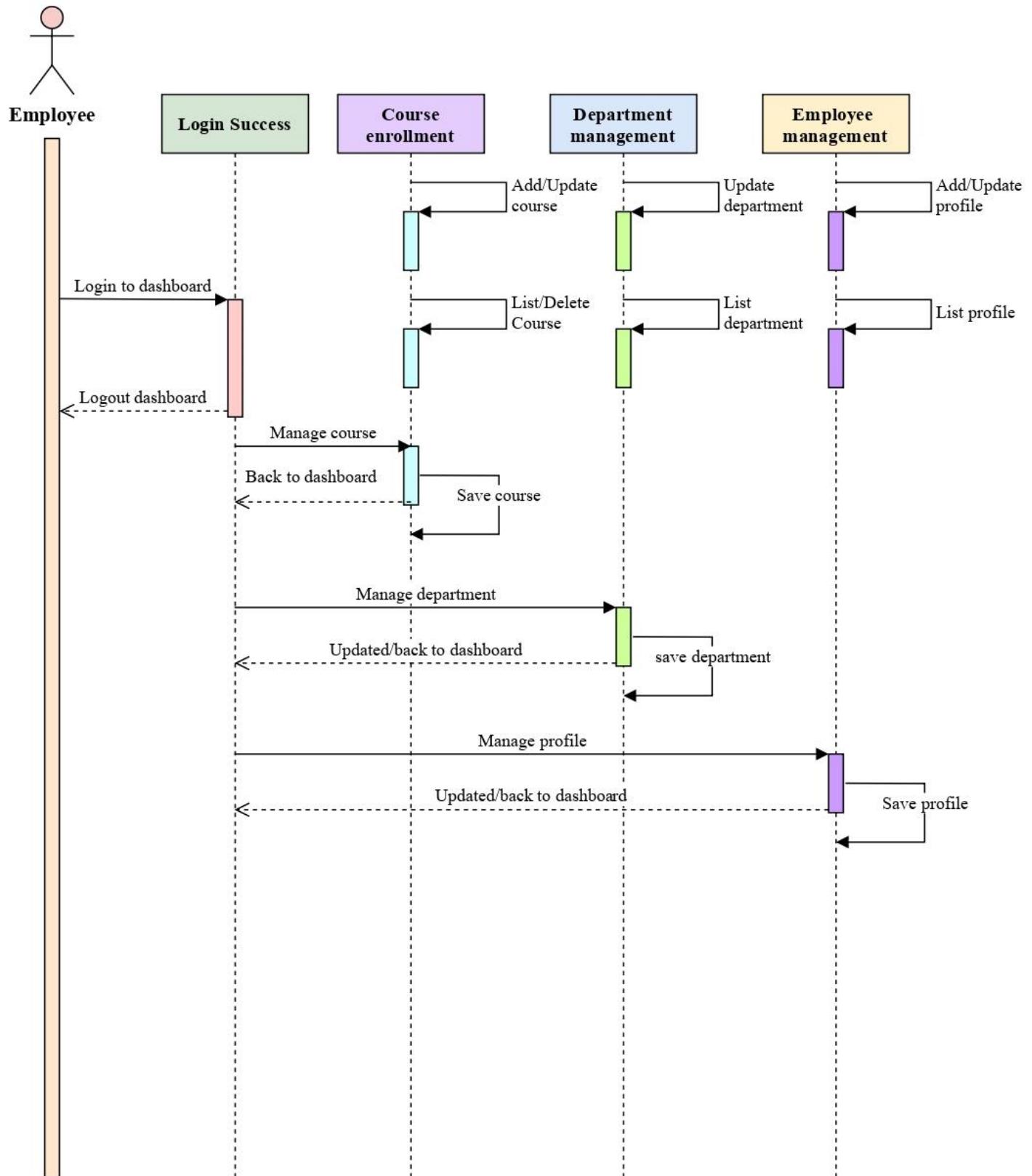
- The vertical axis represents time proceeding down the page.
- Time in a sequence diagram is all about ordering, not duration.

Conclusion: Studied and implemented sequence diagram for our system successfully.

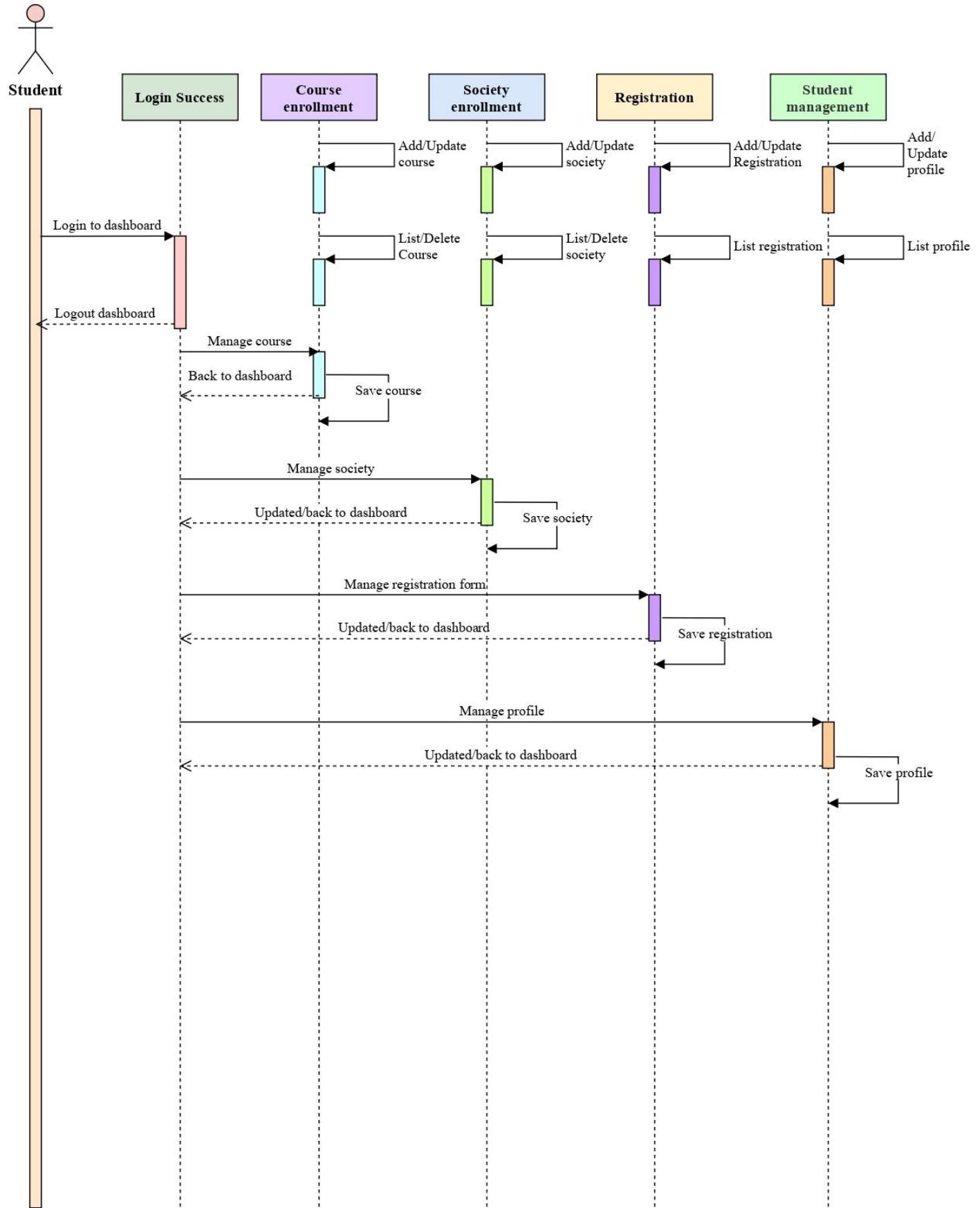
SEQUENCE DIAGRAM FOR ADMIN SERVER



SEQUENCE DIAGRAM FOR EMPLOYEE SERVER



SEQUENCE DIAGRAM FOR STUDENT SERVER



CONTENT BEYOND SYLLABUS 1

Aim : To create the data dictionary of the system using the Data flow diagram

System used → University Management System

Theory of Data Dictionary

- Data dictionary is a file or a set of files that includes a database's metadata. The data dictionary holds records about other objects in the database, such as data ownership, data relationships to other objects & other data.
- The data dictionary is an essential component of any relational database. The users of the database normally don't interact with the data dictionary, it is only handled by the database admin.
- A data dictionary provides a concise guide to understand & use the data.

Data dictionary includes information about -

- 1) Name of all the database tables & their schemas.
- 2) Details about all the tables in the database, such as their owners, their security constraints, when they were created etc.
- 3) Physical information about the tables such as where they are stored & how.
- 4) Table constraints such as primary key attributes, foreign key info, etc.
- 5) Info. about the database views that are visible.

Types of Data Dictionary -

1) Active Data Dictionary :-

If the data dictionary is automatically updated by the database management system when any changes are made in the DB then it is known as active data dictionary i.e., it's self-updating.

2) Passive Data Dictionary :-

A passive data dictionary is maintained separately to the DB whose contents are stored in the dictionary. It has to be manually updated to match the DB. Hence it needs careful handling or else the DB & Data Dictionary are out of sync.

Data Dictionary for University Management System -

| Data item name | Description of data item |
|----------------------|---|
| 1. College info. | It stores the college information such as name, ID, DOB, etc. |
| 2. Student info. | It stores the student information taking admission in colleges. |
| 3. Employee info. | It records the information of all the employees like teachers, etc. |
| 4. Society info. | Records the society information who registers under it. |
| 5. Course info. | Maintains the record of all the courses as well as subjects taught. |
| 6. Access user info. | Checks the DB for the registered user to login. |

| Data item name | Description of Data item |
|----------------------------------|---|
| 7. Student interest info. | Records the information of the student interest in opting different course of subjects. |
| 8. Enter / update / Delete info. | The data items are used to perform update, add and delete information of different users. |
| 9. Administrator | It maintains the overall data base given the complete access. |
| 10. Allocation of course | Teachers allocated different courses records are stored here. |
| 11. Validate / process info. | It helps to maintain & validate any information of the user. |
| 12. Register student | Students registration process is accomplished in college. |
| 13. Register course | Courses add on procedure is followed here. |
| 14. Register society | Registration & maintenance of new societies. |
| 15. Register employee | It maintains the registration of new employees. |
| 16. Retrieve info. | It helps to extract or can say retrieve information. |
| 17. Department info. | Maintains the record of the Departments & its courses. |
| 18. Employee | Maintains the records of the students of others activities. |
| 19. Student | Maintains the updates their courses filed & other activity. |
| 20. Society | Maintains the addition of students (enrolment). |

CONTENT BEYOND SYLLABUS :-

Aim :- To identify the functional and non-functional requirements of the system.

System used → University Management System.

Theory :- functional requirements :-

- These are the requirements that the end user specifically demands as basic facilities that the system should offer. All these functionalities need to be necessarily incorporated into the system as a part of the contract.
- These are represented or stated in the form of input to be given to the system, the operation performed & the output expected.
- They are basically the requirements stated by the user which one can see directly in the final product, unlike the non-functional requirements.

Non-functional requirements :-

- These are basically the quality constraints that the system must satisfy according to the project contract.
- The priority or extent to which these factors are implemented varies from one project to other. They are also called non-behavioural requirements.
- They basically deals with issues like :-
 1. Portability
 2. Security

3. Maintainability
4. Reliability
5. Scalability
6. Performance
7. Reusability
8. Flexibility

Functional Requirements of University Management System -

* Performance requirements

- 1) Response time :- The response of all the operation is good. This has been made possible by careful programming.
- 2) Error handling :- Response to user errors of undesired situations has been taken care of to ensure that the system operates w/o halting.
- 3) Safety & Robustness :- The system can avoid or tackle disastrous action. The system safeguards against undesired events, w/o human intervention.
- 4) Portable :- It is a cross-platform software.
- 5) User friendliness :- The system is easy to learn & understand. A native user can use the system effectively, w/o any difficulties.

* Hardware requirements

Minimum Hardware requirements :-

Processor Pentium IV

HDD 40 GB

RAM 128 MB

Recommended
Hardware
requirements

Processor Core 2 Duo E7300

HDD 80 GB

RAM 512 MB

* Software requirements

Any window-based operating system with DOS support are primary requirements for software development.

Non-functional Requirements for University Management System -

- 1) Security :- The system uses SSL (secured socket layer) in all transactions that include any confidential customer information. The system must automatically log out all customers after a period of inactivity. The system should not leave any cookies on the customer's computer containing the user's password. The system's back-end servers shall only be accessible to authenticated management.
- 2) Availability :- The system should be always available, meaning the user can access it using a web browser, only restricted by the down time of the server on which the system runs. In case of a hardware failure or database corruption, a replacement page will be shown.
- 3) Maintainability :- A commercial database is used for maintaining the database of the application server takes care of the site. In case of failure, a re-initialization of the project will be done.
- 4) Supportability :- The code of supporting modules of the system will be well documented & easy to understand.