5. The Use Case Approach

Ivar Jacobson & others introduced Use Case approach for elicitation & modeling.

Use Case – give functional view

Use Cases are structured outline or template for the description of user requirements modeled in a structured language like English.

Use case diagrams are graphical representations that

may be decomposed into further levels of abstraction.

Components of Use Case approach

Actor:

An actor or external agent, lies outside the system model, but interacts with it in some way. It may represent roles played by human users, external hardware, or other subjects.

Actor

→ Person, machine, information System

Requirements Elicitation

- Cockburn distinguishes between Primary and secondary actors.

- A Primary actor is one having a goal and require the assistance of the system to achieve the goal. They initiate the use cases of the system

- A Secondary actor is one from which System needs assistance. The system, and its use cases, often rely on other people, business processes, or applications in order to obtain a result or specific information required to achieve its goal. A secondary actor never initiates the use case. It is invoked by the system's use cases in order to obtain information or a result.

Use Cases

A use case is initiated by a user with a particular goal in mind, and completes successfully when that goal is satisfied.

* It describes the sequence of interactions between actors and the system necessary to deliver the services that satisfies the goal.

* System is treated as black box.

Thus , Use Case captures who (actor) does what (interaction) with the system, for what purpose (goal), without dealing with system internals.

Requirements Elicitation

*defines all behavior required of the system, bounding the scope of the system.

Jacobson & others proposed a template for writing Use cases as shown below:

1. Introduction

Describe a quick background of the use case.

2.Actors

List the actors that interact and participate in the use cases.

3.Pre Conditions

Pre conditions that need to be satisfied for the use case to perform.

4. Post Conditions

Define the different states in which we expect the system to be in, after the use case executes.

5. Flow of events

5.1 Basic Flow

List the primary events that will occur when this use case is executed.

5.2 Alternative Flows

Any Subsidiary events that can occur in the use case should be separately listed. List each such event as an alternative flow.
A use case can have many alternative flows as required.

6.Special Requirements

Business rules should be listed for basic & information flows as special requirements in the use case narration .These rules will also be used for writing test cases. Both success and failures scenarios should be described.

7.Use Case relationships

For Complex systems it is recommended to document the relationships between use cases. Listing the relationships between use cases also provides a mechanism for traceability

Use Case Template.

Use Case Guidelines

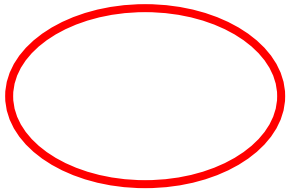
1. Identify all users
2. Create a user profile for each category of users including all roles of the users play that are relevant to the system.
3. Create a use case for each goal, following the use case template maintain the same level of abstraction throughout the use case. Steps in higher level use cases may be treated as goals for lower level (i.e. more detailed), sub-use cases.
4. Structure the use case
5. Review and validate with users.

Use case Diagrams

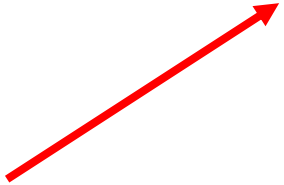
-- represents what happens when actor interacts with a system.

-- captures functional aspect of the system.

Actor



Use Case



Relationship between actors and use case and/or between the use cases.

-- Actors appear outside the rectangle.

--Use cases within rectangle providing functionality.

--Relationship association is a solid line between actor & use cases.

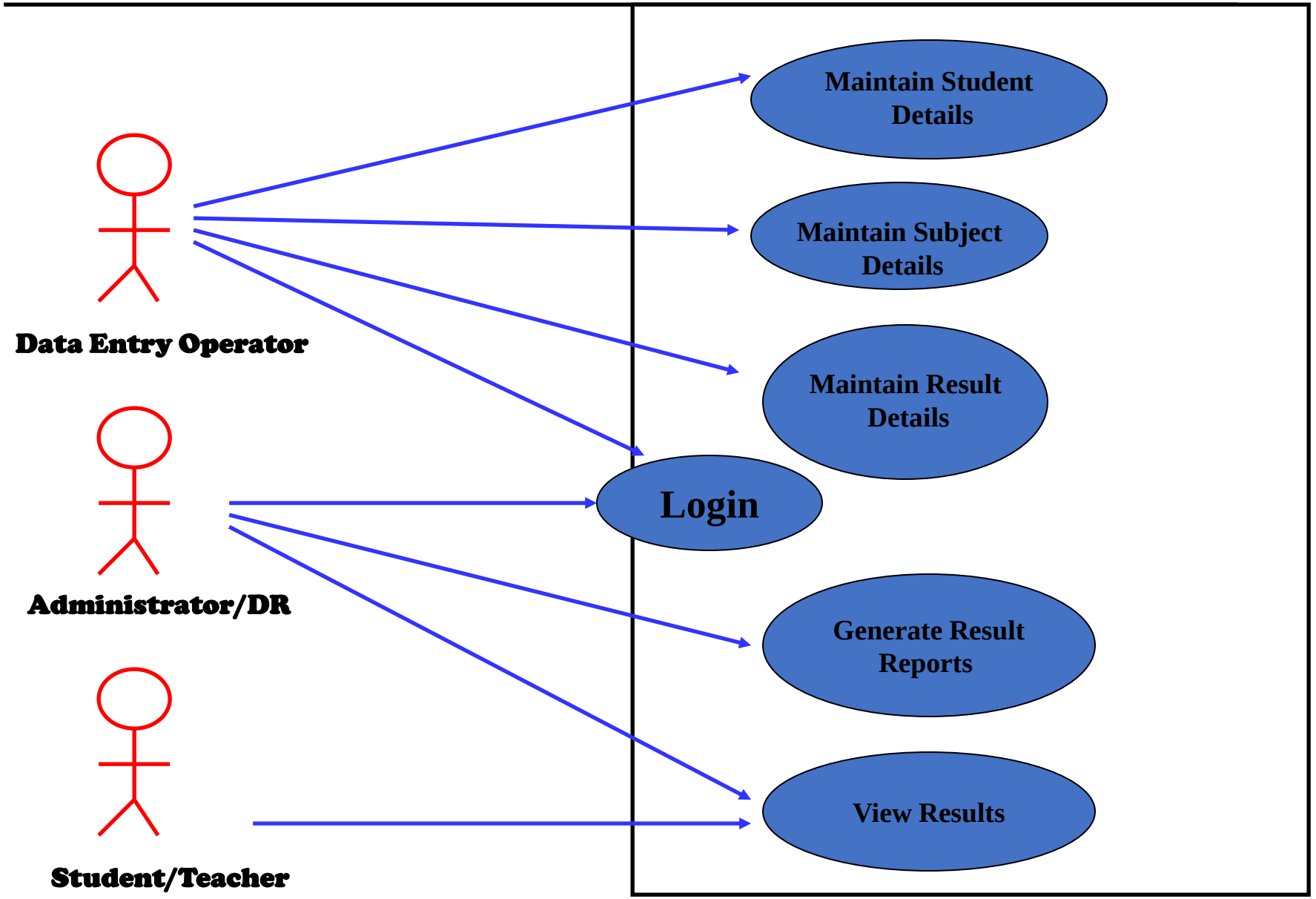
Requirements Elicitation

*Use cases should not be used to capture all the details of the system.

*Only significant aspects of the required functionality

*Use Cases are for “what” the system is , not “how” the system will be designed

Use case diagram for Result Management System



Requirements Elicitation

1. Maintain student Details

   Add/Modify/update students details like name, address.

2.Maintain subject Details

   Add/Modify/Update Subject information semester wise

3.Maintain Result Details

   Include entry of marks and assignment of credit points for each paper.

4. Login

   Use to Provide way to enter through user id & password.

5. Generate Result Report

   Use to print various reports

6. View Result
   (i) According to course code
   (ii) According to Enrollment number/roll number

Login

1.1 Introduction : This use case describes how a user logs into the Result Management System.

1.2 Actors : (i) Data Entry Operator
(ii) Administrator/Deputy Registrar

1.3 Pre Conditions : None

1.4 Post Conditions : If the use case is successful, the actor is logged into the system. If not, the system state is unchanged.

Requirements Elicitation (Use Case)

1.5 Basic Flow : This use case starts when the actor wishes to login to the Result Management system.

(i) System requests that the actor enter his/her name and password.

(ii) The actor enters his/her name & password.

(iii) System validates name & password, and if finds correct allow the actor to logs into the system.

1.6 Alternate Flows

1.6.1 Invalid name & password

If in the basic flow, the actor enters an invalid name and/or password, the system displays an error message. The actor can choose to either return to the beginning of the basic flow or cancel the login, at that point, the use case ends.

1.7 Special Requirements:

None

1.8 Use case Relationships:

None

2.Maintain student details

2.1 Introduction : Allow DEO to maintain student details. This includes adding, changing and deleting student information

2.2 Actors : DEO

2.3 Pre-Conditions: DEO must be logged onto the system before this use case begins.

Use Cases

2.4 Post-conditions : If use case is successful, student information is added/updated/deleted from the system. Otherwise, the system state is unchanged.

2.5 Basic Flow : Starts when DEO wishes to add/modify/update/delete Student information.

(i) The system requests the DEO to specify the function, he/she would like to perform (Add/update/delete)

(ii) One of the sub flow will execute after getting the information.

- If DEO selects "Add a student", "Add a student" sub flow will be executed.
- If DEO selects "update a student", "update a student" sub flow will be executed.
- If DEO selects "delete a student", "delete a student" sub flow will be executed.

2.5.1 Add a student

(i) The system requests the DEO to enter:

Name
Address
Roll No
Phone No
Date of admission

(ii) System generates unique id

2.5.2 Update a student

(i) System requires the DEO to enter student-id.

(ii) DEO enters the student_id. The system retrieves and display the student information.

(iii) DEO makes the desired changes to the student information.

(iv) After changes, the system updates the student record with changed information.

2.5.3 Delete a student

(i) The system requests the DEO to specify the student-id.

(ii) DEO enters the student-id. The system retrieves and displays the student information.

(iii) The system prompts the DEO to confirm the deletion of the student.

(iv) The DEO confirms the deletion.

(v) The system marks the student record for deletion.

2.6 Alternative flows

2.6.1 Student not found

If in the update a student or delete a student sub flows, a student with specified_id does not exist, the system displays an error message .The DEO may enter a different id or cancel the operation. At this point ,Use case ends.

2.6.2 Update Cancelled

If in the update a student sub-flow, the data entry operator decides not to update the student information, the update is cancelled and the basic flow is restarted at the begin.

2.6.3 Delete cancelled

If in the delete a student sub flows, DEO decides not to delete student record ,the delete is cancelled and the basic flow is re-started at the beginning.

2.7 Special requirements

None

2.8 Use case relationships

None

3. Maintain Subject Details

3.1 Introduction

The DEO to maintain subject information. This includes adding, changing, deleting subject information from the system

3.2 Actors : DEO

3.3 Preconditions : DEO must be logged onto the system before the use case begins.

3.4 Post conditions :

If the use case is successful, the subject information is added, updated, or deleted from the system, otherwise the system state is unchanged.

3.5 Basic flows :

The use case starts when DEO wishes to add, change, and/or delete subject information from the system.

(i) The system requests DEO to specify the function he/she would like to perform i.e.

- Add a subject
- Update a subject
- Delete a subject.

(ii) Once the DEO provides the required information, one of the sub flows is executed.

- If DEO selected “Add a subject” the “Add-a subject sub flow is executed.
- If DEO selected “Update-a subject” the “update-a- subject” sub flow is executed
- If DEO selected “Delete- a- subject”, the “Delete-a-subject” sub flow is executed.

3.5.1 Add a Subject

(i) The System requests the DEO to enter the subject information. This includes :

* Name of the subject

* Subject Code
* Semester
* Credit points

(ii) Once DEO provides the requested information, the system generates and assigns a unique subject-id to the subject. The subject is added to the system.

(iii) The system provides the DEO with new subject-id.

3.5.2 Update a Subject

(i) The system requests the DEO to enter subject_id.

(ii) DEO enters the subject_id. The system retrieves and displays the subject information.

(iii) DEO makes the changes.

(iv) Record is updated.

3.5.3 Delete a Subject

(i) Entry of subject_id.

(ii) After this, system retrieves & displays subject information.

* System prompts the DEO to confirm the deletion.

* DEO verifies the deletion.

* The system marks the subject record for deletion.

3.6 Alternative Flow

3.6.1 Subject not found

If in any sub flows, subject-id not found, error message is displayed. The DEO may enter a different id or cancel the case ends here.

3.6.2 Update Cancelled

If in the update a subject sub-flow, the data entry operator decides not to update the subject information, the update is cancelled and the basic flow is restarted at the begin.

3.6.3 Delete Cancellation

If in delete-a-subject sub flow, the DEO decides not to delete subject, the delete is cancelled, and the basic flow is restarted from the beginning.

3.7 Special Requirements:

None

3.8 Use Case-relationships

None

4. Maintain Result Details

4.1 Introduction

This use case allows the DEO to maintain subject & marks information of each student. This includes adding and/or deleting subject and marks information from the system.

4.2 Actor

DEO

4.3 Pre Conditions

DEO must be logged onto the system.

4.4 Post Conditions

If use case is successful ,marks information is added or deleted from the system. Otherwise, the system state is unchanged.

4.5 Basic Flow

This use case starts, when the DEO wishes to add, update and/or delete marks from the system.

(i) DEO to specify the function

(ii) Once DEO provides the information one of the subflow is executed.

* If DEO selected “Add Marks “, the Add marks subflow is executed.
* If DEO selected “Update Marks”, the update marks subflow is executed.
* If DEO selected “Delete Marks”, the delete marks subflow is executed.

4.5.1 Add Marks Records

Add marks information .This includes:

a. Selecting a subject code.

b. Selecting the student enrollment number.

c. Entering internal/external marks for that subject code & enrollment number.

(ii) If DEO tries to enter marks for the same combination of subject and enrollment number,the system gives a message that the marks have already been entered.

(iii) Each record is assigned a unique result_id.

4.5.2 Delete Marks records

1. DEO makes the following entries:

   a. Selecting subject for which marks have to be deleted.

   b. Selecting student enrollment number.

   c. Displays the record with id number.

   d. Verify the deletion.

   e. Delete the record.

4.5.2 Update Marks records

1. The System requests DEO to enter the record_id.

2. DEO enters record_id. The system retrieves & displays the information.

3. DEO makes changes.

4. Record is updated.

4.5.3 Compute Result

(i) Once the marks are entered, result is computed for each student.

(ii) If a student has scored more than 50% in a subject, the associated credit points are allotted to that student.

(iii) The result is displayed with subject-code, marks & credit points.

4.6 Alternative Flow

4.6.1 Record not found

If in update or delete marks sub flows, marks with specified id number do not exist, the system displays an error message. DEO can enter another id or cancel the operation.

4.6.2 Delete Cancelled

If in Delete Marks, DEO decides not to delete marks, the delete is cancelled and basic flow is re-started at the beginning.

4.7 Special Requirements

None

4.8 Use case relationships

None

5 View/Display result

5.1 Introduction

This use case allows the student/Teacher or anyone to view the result. The result can be viewed on the basis of course code and/or enrollment number.

5.2 Actors

Administrator/DR, Teacher/Student

5.3 Pre Conditions

None

5.4 Post Conditions

If use case is successful, the marks information is displayed by the system. Otherwise, state is unchanged.

5.5 Basic Flow

Use case begins when student, teacher or any other person wish to view the result.

Two ways

-- Enrollment no.

-- Course code

(ii) After selection, one of the sub flow is executed.

Course code ⟶ Sub flow is executed

Enrollment no. ⟶ Sub flow is executed

5.5.1 View result enrollment number wise

(i) User to enter enrollment number

(ii) System retrieves the marks of all subjects with credit points

(iii) Result is displayed.

5.6 Alternative Flow

5.6.1 Record not found

Error message should be displayed.

5.7 Special Requirements

None

5.8 Use Case relationships

None

6. Generate Report

6.1 Introduction

This use case allows the DR to generate result reports. Options are

a. Course code wise
b. Semester wise
c. Enrollment Number wise

6.2 Actors

DR

6.3 Pre-Conditions

DR must logged on to the system

6.4 Post conditions

If use case is successful, desired report is generated. Otherwise, the system state is unchanged.

6.5 Basic Flow

The use case starts, when DR wish to generate reports.

(i) DR selects option.

(ii) System retrieves the information displays.

(iii) DR takes printed reports.

6.6 Alternative Flows

6.6.1 Record not found

If not found, system generates appropriate message. The DR can select another option or cancel the operation. At this point, the use case ends.

6.7 Special Requirements

None

6.8 Use case relationships

None

7. Maintain User Accounts

7.1 Introduction

This use case allows the administrator to maintain user account. This includes adding, changing and deleting user account information from the system.

7.2 Actors

Administrator

7.3 Pre-Conditions

The administrator must be logged on to the system before the use case begins.

7.4 Post-Conditions

If the use case was successful, the user account information is added, updated, or deleted from the system. Otherwise, the system state is unchanged.

7.5 Basic Flow

This use case starts when the Administrator wishes to add, change, and/or delete use account information from the system.

(i) The system requests that the Administrator specify the function he/she would like to perform (either Add a User Account, Update a User Account, or Delete a User Account).

(ii) Once the Administrator provides the requested information, one of the sub-flows is executed

* If the Administrator selected “Add a User Account”, the **Add a User Account** sub flow is executed.

* If the Administrator selected “Update a User Account”, the **Update a User Account** sub-flow is executed.

* If the Administrator selected “Delete a User Account”, the **Delete a User Account** sub-flow is executed.22

7.5.1 Add a User Account

1. The system requests that the Administrator enters the user information. This includes:

(*a*) User Name

(*b*) User ID-should be unique for each user account

(*c*) Password

(*d*) Role

2. Once the Administrator provides the requested information, the user account information is added.

7.5.2 Update a User Account

1. The system requests that the Administrator enters the User ID.
2. The Administrator enters the User ID. The system retrieves and displays the user account information.
3. The Administrator makes the desired changes to the user account information. This includes any of the information specified in the **Add a User Account** sub-flow.
4. Once the Administrator updates the necessary information, the system updates the user account record with the updated information.

7.5.3 Delete a User Account

1. The system requests that the Administrator enters the User ID.
2. The Administrator enters the User ID. The system retrieves and displays the user account information.
3. The system prompts the Administrator to confirm the deletion of the user account.
4. The Administrator confirms the deletion.
5. The system deletes the user account record.

7.6 Alternative Flows

7.6.1 User Not Found

If in the **Update a User Account** or **Delete a User Account** sub-flows, a user account with the specified User ID does not exist, the system displays an error message. The Administrator can then enter a different User ID or cancel the operation, at which point the use case ends.

7.6.2 Update Cancelled

If in the **Update a User Account** sub-flow, the Administrator decides not to update the user account information, the update is cancelled and the **Basic Flow** is re-started at the beginning.

7.6.3 Delete Cancelled

If in the **Delete a User Account** sub-flow, the Administrator decides not to delete the user account information, the delete is cancelled and the **Basic Flow** is re-started at the beginning.

7.7 Special Requirements

None

7.8 Use case relationships

None

8. Reset System

8.1 Introduction

This use case allows the allows the administrator to reset the system by deleting all existing information from the system .

8.2 Actors

Administrator

8.3 Pre-Conditions

The administrator must be logged on to the system before the use case begins.

8.4 Post-Conditions

If the use case was successful, all the existing information is deleted from the backend database of the system. Otherwise, the system state is unchanged.

8.5 Basic Flow

This use case starts when the Administrator wishes to reset the system.

i. The system requests the Administrator to confirm if he/she wants to delete all the existing information from the system.

ii. Once the Administrator provides confirmation, the system deletes all the existing information from the backend database and displays an appropriate message.

8.6 Alternative Flows

8.6.1 Reset Cancelled

If in the Basic Flow, the Administrator decides not to delete the entire existing information, the reset is cancelled and the use case ends.

8.7 Special Requirements

None

8.8 Use case relationships

None