

Software Engineering Project

By Anmol Gupta

Project Name: codeflix

Problem Statement:

There are a lot of **OpenCourseWares, Tutorials** and **Archives of Lectures** previously taught available on the Internet through sites like **YouTube, MIT OCW, University Websites** etc. These **OCWs, Tutorials or Lecture Archives** are a great way to learn but due to the current state of their scattered existence in most cases sometimes makes them hard to follow and even complete although having quality material. There is also no sense of achievement like certificate etc. after completion. Some **OCWs** have top-notch Problem Sets and Assignments that have not been solved and because of not having a community of learners are hard to follow. Although Sites like YouTube provide great way to consume courses or tutorials there is always the **distraction factor** on YouTube because of availability of non-educational content on YouTube

The goal of this SE project is to try to solve all these problems by creating a better **Educational Content Consuming and Progress Tracking System**. The project aims to provide:

- Aggregation of **OpenCourseWares, Tutorials** and **Archives of Lectures**.
- Providing better browsing and consumption of **OpenCourseWares, Tutorials** and **Archives of Lectures**.
- Creating an Interactive **Community of Learners** using the System as well as **Community for each course/tutorial**.
- Providing ability to keep track of courses/tutorials.
- Providing the ability to add User-Defined courses/tutorials.
- Providing the ability to publish/share courses/tutorials developed by the User of the System.
- Provide recommendation of courses/tutorials.

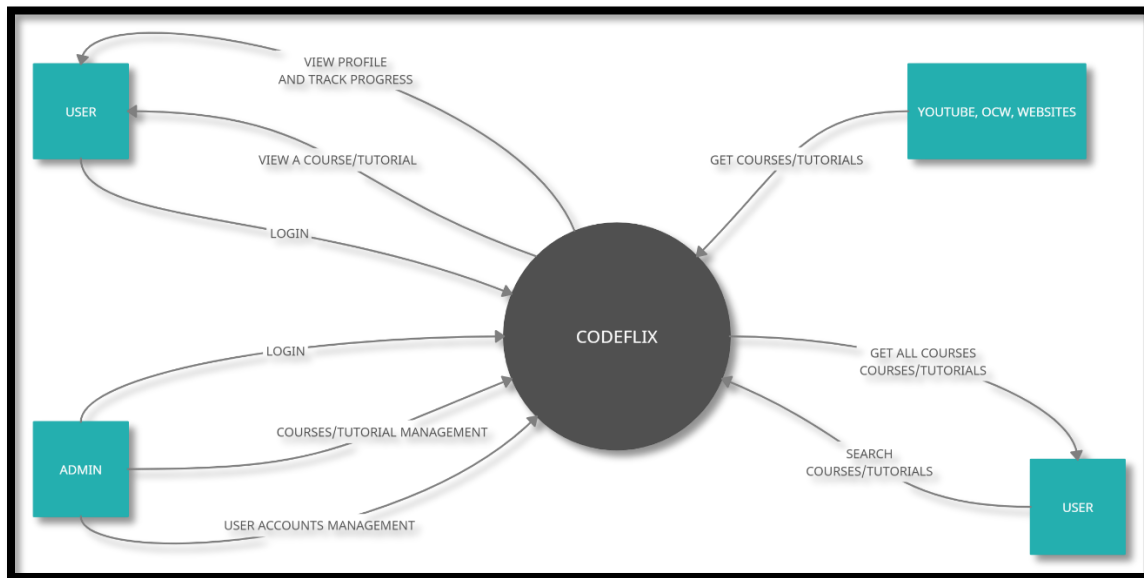
Basically, the aim is to utilize the best feature of **Netflix, Khan Academy** and **YouTube** to provide a **better and interactive self-learning environment or system** and ***“Community of Learners”***.

Process Model:

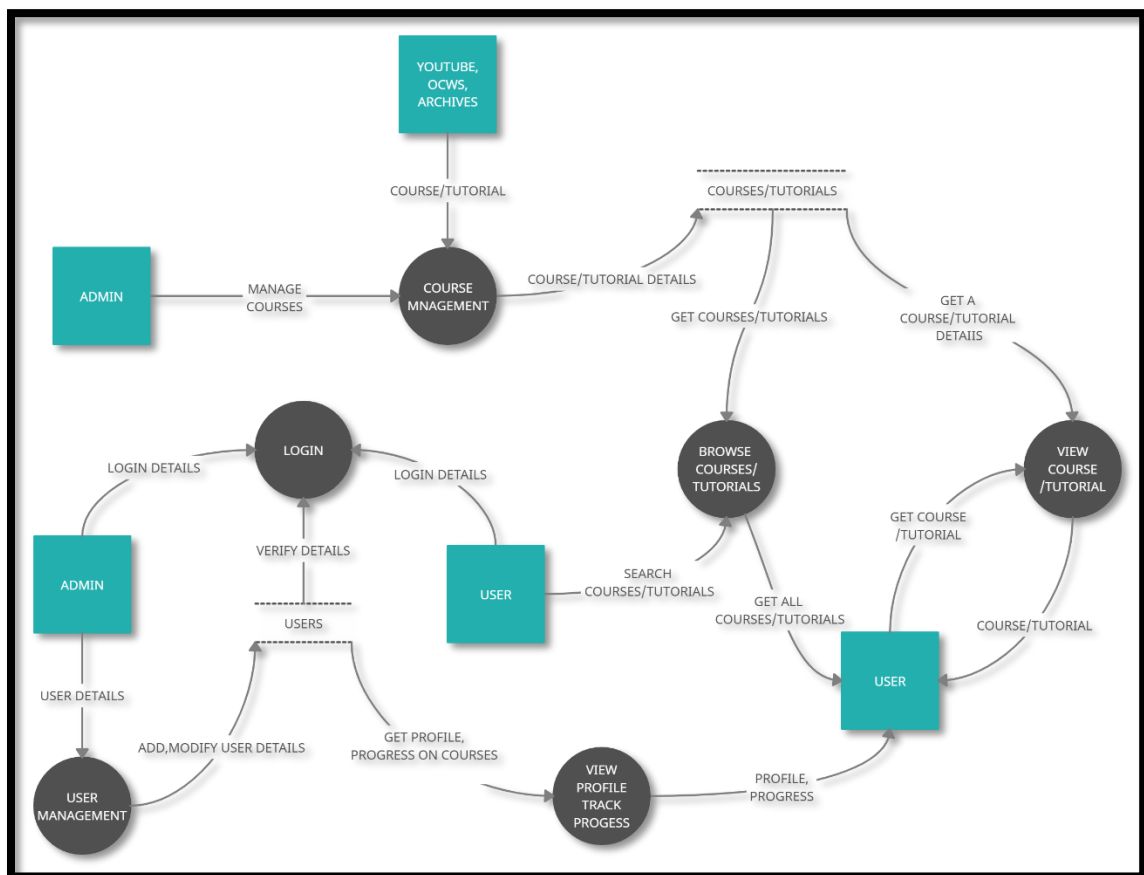
Spiral Model – because of better Risk Analysis and Incremental Development.

Data Flow Diagrams:

Level 0 (Context Diagram)

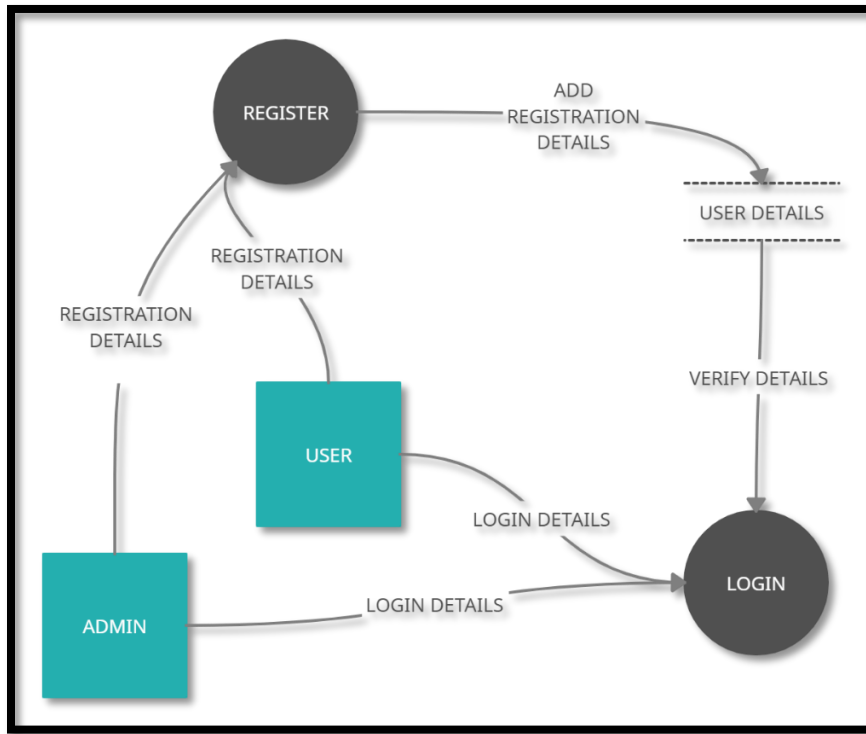


Level 1

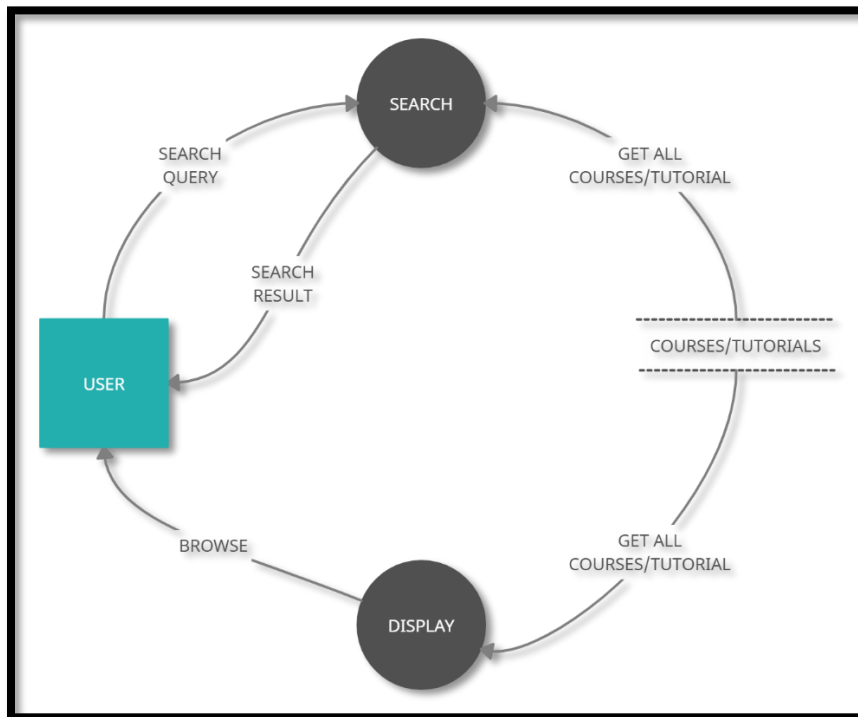


Level 2

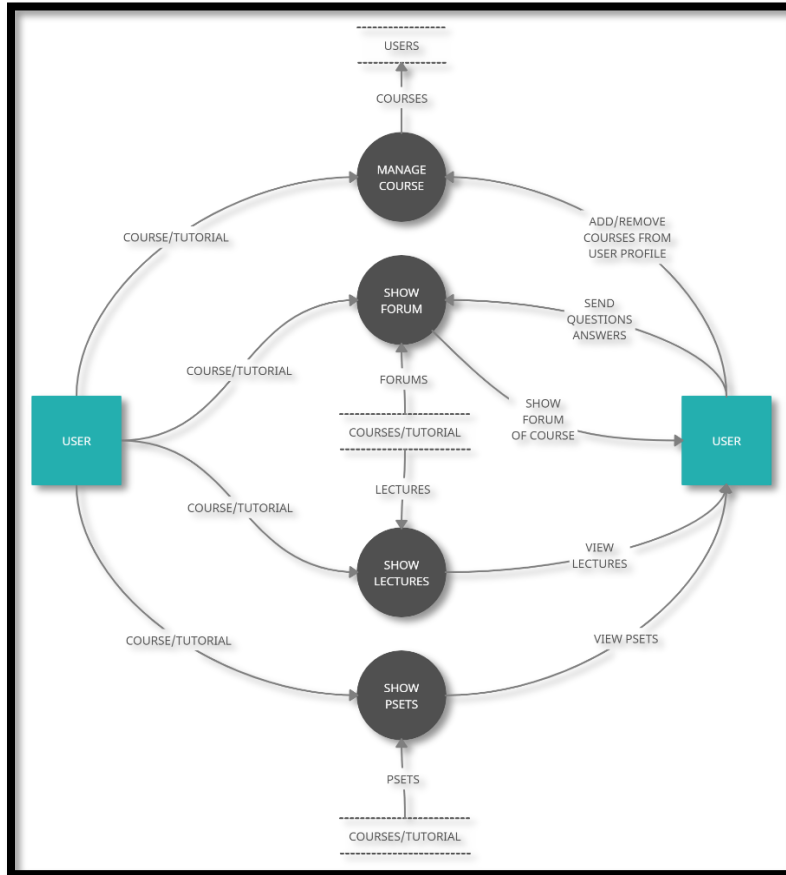
Login



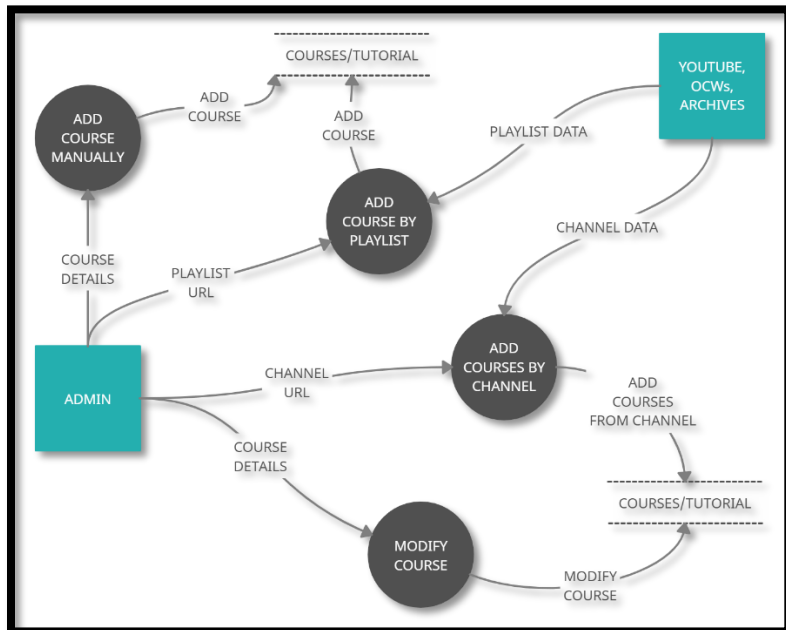
Browse Courses/Tutorials



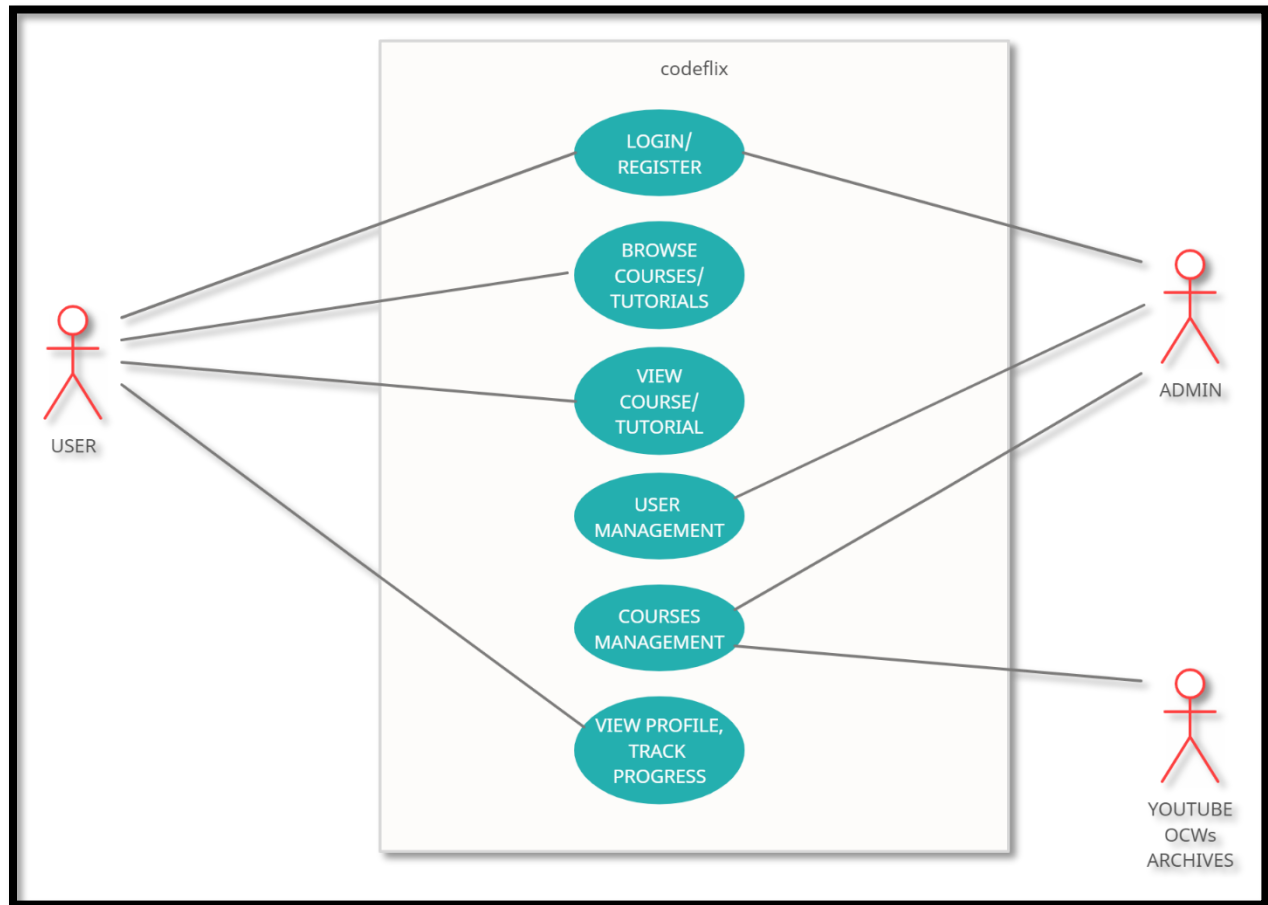
View Course/Tutorial



Course Management



Use Case Diagram:



Use Cases

Login/Register

- Description: This use case describes how the actors Logs in and Register to the System.
- Actors:
 - User
 - Administrator
- Flow of Events:
 - Basic Flow –
 - This use case starts when the actors want to Login or Register to the System.
 - The System asks the actors to select a specific function they want to perform. (Login or Register functions)
 - After the actors select a certain function the following sub-flows are executed according to the selection:
 - On Selection of “Login” function, **Login** sub-flow is executed

- On Selection of “Register” function, **Register** sub-flow is executed.
- **Login** sub-flow
 - The System request the actors to enter the following information:
 - Username
 - Password
 - The System validates the entered information by the actors.
- **Register** sub-flow
 - The System request the actors to enter the following information:
 - Name
 - Username
 - E-mail
 - Age
 - Password
 - The System also select a predefined Profile Picture for the actors and validates the entered information by the actors.
- Alternative Flow-
 - Invalid Information
 - If the actors enter invalid information in the **Register** or **Login** sub-flow, they will be displayed with an error message.
 - The actors will then go to the beginning of the Basic Flow.
- Pre-conditions:
 - None
- Post-conditions:
 - If the use case was successful the actors will have access to the system, otherwise the state of the System will be remained unchanged.
 - If the actor was the **User**, he/she will have access to the **Browse Courses/Tutorials, View a Course/Tutorial** and **View Profile, Track Progress** modules or subprocesses of the System.
 - If the actor was the **Administrator**, he/she will have access to the **User Management** and **Course Management System** modules or subprocesses of the System.

Browse Courses/Tutorials

- Description: This use case describes how the actors will browse and search for different Courses/Tutorials the System provides.
- Actors: User
- Flow of Events:
 - Basic Flow:
 - The use case starts when the actors want to browse and search Various Courses/Tutorials the System displays.

- The System asks actors to select a specific Course/Tutorial either by selecting from recommended Courses/Tutorials or through “Search” function (calls the **Search** sub-flow).
 - **Search** sub-flow:
 - The System asks the actors to enter Search Query to search for the various Courses/Tutorials provided by the System.
 - The System validates the Search Query and then take the users to the Search Result.
 - The Actor can then select from the Courses/Tutorials according to the Search Results.
 - Alternative Flow:
 - Unavailable Courses/Tutorials
 - If the actors provide invalid Course/Tutorial or a something that doesn’t exist in the **Search** sub-flow, then the user is displayed with an error message of “No Courses/Tutorials found”. The actor can now either
- Pre-conditions:
 - The actors must be logged in the System before they can use this Module or Sub-Process.
- Post-conditions:
 - If the use case is successful the user can now access the Specific Course/Tutorial he/she selected by the **View Course/Tutorial** Module or Sub-Process of the System.
- Extension:
 - **View Course/Tutorial** Use Case.

View Course/Tutorial:

- Description: This use case lets the actor access a certain Course/Tutorial provided by the System
- Actors: User
- Flow of Events:
 - Basic Flow:
 - The actors have the access to Use Case after they perform the **Browse Courses/Tutorials** successfully
 - The System shows the actors the Course/Tutorial Information and asks actors to perform a specific function according to which a sub-flow will be executed.
 - If the actors select “Show Lectures” function, **Show Lectures** sub-flow is executed.

- If the actors select “Show Problem Sets” function, **Show Problem Sets** sub-flow is executed.
 - If the actors select “Forum” function, **Forum** sub-flow is executed.
 - If the actors select “Manage Course/Tutorial”, **Manage Course/Tutorial** sub-flow is executed.
- **Show Lectures** sub-flow
 - The System displays all the Lecture of the Course/Tutorial
 - The actors can click (or select) on any Lecture to watch them.
- **Show Problem** sub-flow
 - The System displays all the Problem Sets of the Course/Tutorial if any exists.
 - The actors can click (or select) any Problem Set to view it.
- **Forum** sub-flow
 - The System lets actors perform QA.
 - The System asks actors that asks Questions the following information:
 - Question
 - Description
 - The System asks actors that Answer questions raised by other actors the following information:
 - Answer.
- **Manage Course** sub-flow
 - The System provides the ability to *add/remove* Course/Tutorial to the User Profile.
- Alternative Flow:
 - If the actor enters Invalid data in the **Forum** sub-process, he/she will get an error message and the sub-process will start from the beginning.
- Pre-conditions:
 - The actors must be logged in the System and the use case must have the **Browse Course/Tutorial** resulting data before they can use it.
- Post-conditions:
 - If the use case is successful than the actors have access to the Lectures, Problem Set provided by the System. They also have ability to add/remove a Course/Tutorial from their Profile and also have access to the Forum of the Course/Tutorial provided by the System.

Course Management

- Description: This use case describes how an actor manages Course/Tutorial Provided by the System. (Add, Remove, Modify).
- Actors: Administrator, [YouTube, OCWs, Websites]

- Flow of Events:
 - Basic Flow:
 - The System lets the actors to Add and Modify Courses/Tutorials.
 - The System asks the actor to select a specific function according to which a sub-flow is executed.
 - If the actor selects “Add Course/Tutorial by YouTube Playlist” function, **Add Course by YouTube Playlist** sub-flow is executed.
 - If the actor selects “Add Courses/Tutorials by YouTube Channel” function, **Add Course by YouTube Channel** sub-flow is executed.
 - If the actor selects “Add Course/Tutorial by Manually” function, **Add Course/Tutorial by Manually** sub-flow is executed.
 - If the actor selects “Modify Course” function, **Modify Course** sub-flow is executed.
 - **Add Course by YouTube Playlist** sub-flow
 - The System asks the actor to enter *valid YouTube Playlist URL*.
 - The System then extracts the Data from the entered URL and add the Course/Tutorial to the Course/Tutorial Database.
 - **Add Courses by YouTube Channel** sub-flow
 - The System asks the actor to enter *valid YouTube Channel URL*.
 - The System then extracts Data from the entered URL and adds all the Playlist from the Channel as Courses/Tutorials to the Course/Tutorial Database.
 - **Add Course/Tutorial by Manually** sub-flow
 - The System asks the actor the following information:
 - Course/Tutorial Name
 - Description
 - Lectures
 - Problem Set (if exists)
 - The System validates the information entered by the User and then adds Course/Tutorial to the Courses/Tutorials Database.
 - **Modify Course** sub-flow
 - The System first asks the actor to select a Course/Tutorial from the Courses/Tutorials that the System has
 - After selecting a Course/Tutorial the System asks the actor the following information:
 - Course/Tutorial Name
 - Description
 - Lectures
 - Problem Set (if exists)

- On validating the data entered by the actors the selected course by the actor is modified.
 - Alternative Flow:
 - Invalid URL
 - If the actor enters *invalid* data/information in the **Add Course by YouTube Channel, Add Courses by YouTube Channel** sub-flow, the System displays an Error Message.
 - The sub-flow is restarted
 - Cancellation
 - If the actors wish to cancel the any sub-flow the use case will be restarted.
 - Invalid Information
 - If the actor enters Invalid Information in **Add Course Manually, Modify Course** sub-flow, then use case is restarted.
 - Pre-conditions:
 - The actor must be logged in to the System and must be an **Administrator** before he/she can access/use this sub-process or module of the System.
 - Post-conditions:
 - If the use case is successful than the actor has Added or Modified the Courses/Tutorial Databases.

Data Dictionary

Data	Description
LOGIN_DETAILS	Username +Password
REGISTRATION_DETAILS	Name + Username + E-mail + Age + Password
SEARCH_QUERY	1[0[A-Z] 0[a-z] 0[0-9] 0[.]] 25
COURSE_DETAILS	CourseName + Description + Lectures + ProblemSet
PLAYLIST_URL	1[0[A-Z] 0[a-z] 0[0-9] 0[:]] 25
CHANNEL_URL	1[0[A-Z] 0[a-z] 0[0-9] 0[:]] 25
Username	5[0[A-Z] 0[a-z] 0[0-9]]25
Password	8[1[A-Z] 0[a-z] 1[0-9] 1[Special Character]]
E-mail	5[1[A-Z] 0[a-z] 1[0-9] 1[@+.]]
Age	1[0-9]
CourseName	5[0[A-Z] 0[a-z] 0[0-9]]25
Description	5[0[A-Z] 0[a-z] 0[0-9] 0[Special Character]]
Lectures	1[URL]
ProblemSet	1[URL]
URL	1[0[A-Z] 0[a-z] 0[0-9] 0[:]]

