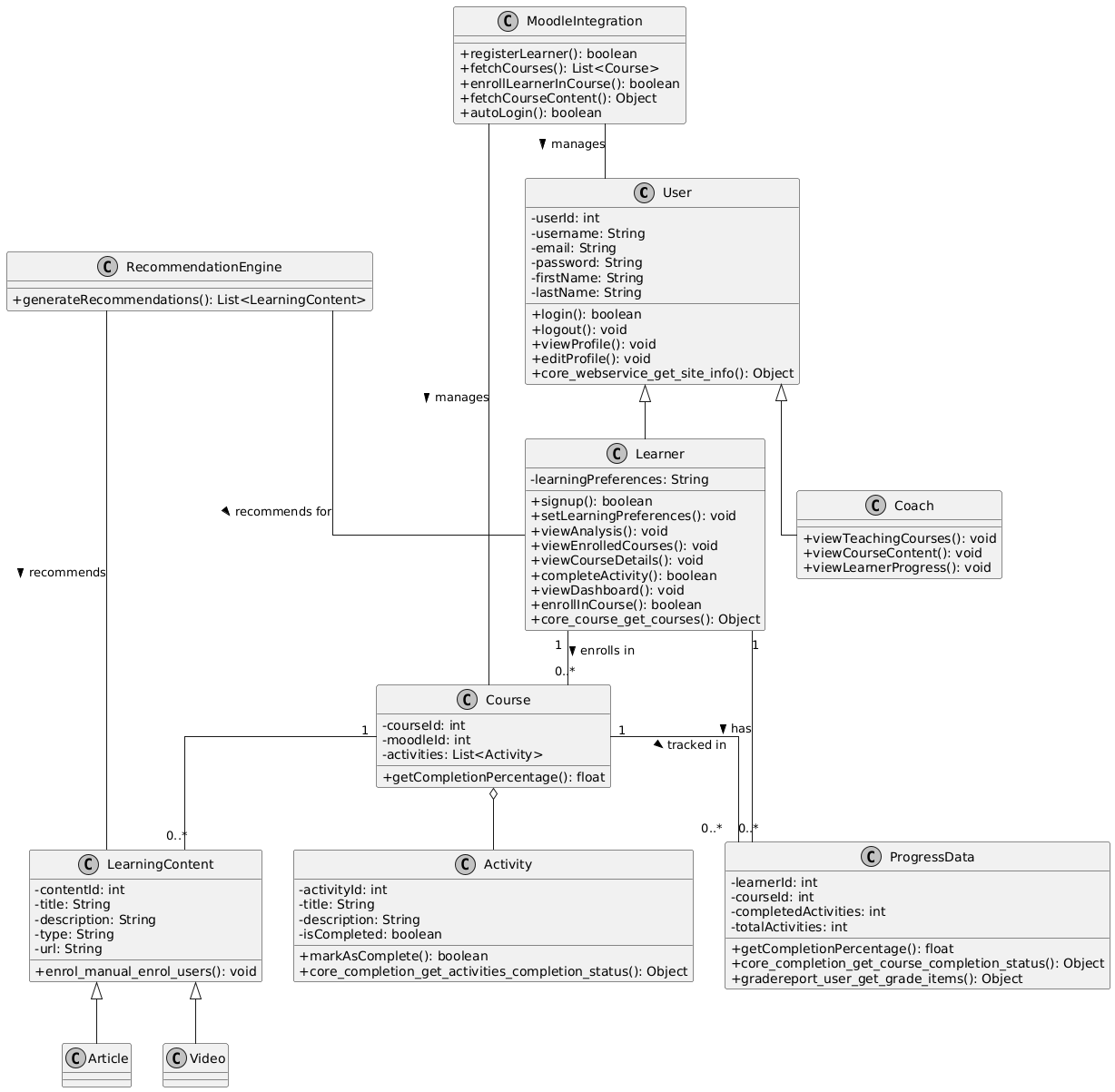
**Product Design**

**Team- 4 : Arshiya Noureen, Sanyam Agarwal, Kushal Mangla, Vishak Kashyap, Shreyas Deb**

**Design Model**

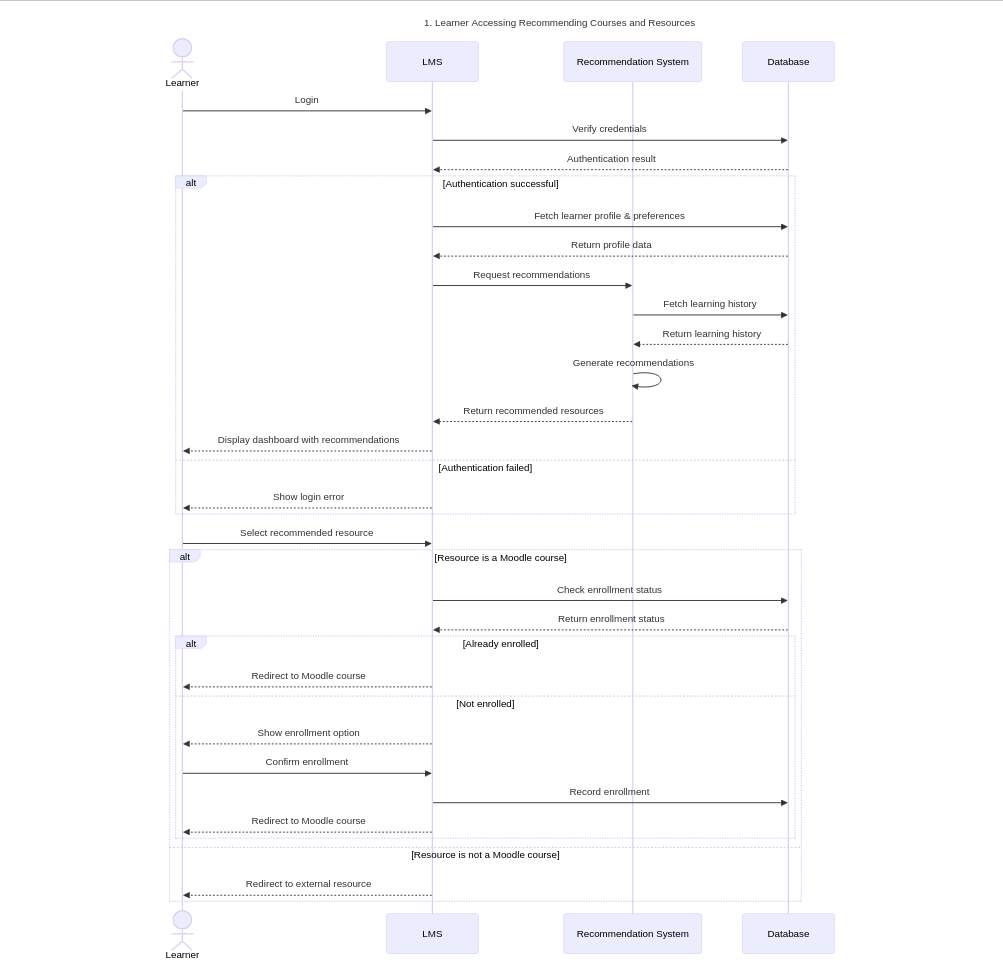
| **Class No.** | **Class Name** | **Class State(Attributes)** | **Class Behavior** |
| --- | --- | --- | --- |
| 1 | User | userId username email  password firstName lastName | **login()**: Authenticates the user and logs them in. Returns true if successful.  **logout()**: Logs the user out of the system.  **viewProfile()**: Displays the learner's profile information.  **editProfile():** Updates user profile information.  **core\_webservice\_get\_site\_info():** Retrieves user information including user ID and site details |
| 2 | Coach | Inherits from User | **viewTeachingCourses()**: Displays the list of courses taught by the coach.  **viewCourseContent()**: Displays the content for a specific course, using core\_course\_get\_contents.  **viewLearnerProgress()**: Displays the progress data of a specific learner. |
| 3 | Learner | (Inherits from User)  learningPreferences | **signup()**: Registers a new learner and returns true if successful.  **setLearningPreferences()**: Allows the learner to set or update their learning preferences.  **viewAnalysis():** Displays the learner’s progress data  **viewEnrolledCourses()**: Displays the list of courses the learner is enrolled in using core\_enrol\_get\_enrolled\_users  **viewCourseDetails()**: Displays details of a specific course.  **completeActivity()**: Marks an activity as complete for the learner. Returns true if successful.  **viewDashboard()**: Displays the learner's dashboard with an overview of their progress.  **enrollInCourse()**: Enrolls the learner in a course. Returns true if successful.  **core\_course\_get\_courses():** Fetches all available courses |
| 4 | Learning Content | contentId title description type url | **enrol\_manual\_enrol\_users():** Enrolls users into courses with specific roles |
| 5 | Course | courseId  moodleId  activities | **getCompletionPercentage()**: Calculates and returns the completion percentage of a learner in the course. |
| 6 | Article | Inherits from LearningContent | None |
| 7 | Video | Inherits from LearningContent | None |
| 8 | Activity | activityId  title  description  isCompleted | **markAsComplete()**: Marks the activity as complete for the learner. Returns true if successful as core\_completion\_mark\_course\_self\_completed.  **core\_completion\_get\_activities\_completion\_status():** Gets completion status of all activities for a user in a course |
| 9 | Progress Data | learnerId  courseId  completedActivities  totalActivities | **getCompletionPercentage()**: Calculates and returns the completion percentage for the learner in the course  **core\_completion\_get\_course\_completion\_status():** Gets overall course completion status for a user  **gradereport\_user\_get\_grade\_items():** Retrieves grade items for a user in a course |
| :10 | MoodleIntegration | None | **registerLearner()**: Registers a learner in the Moodle system. Returns true if successful  **fetchCourses()**: Fetches a list of available courses from Moodle.  **enrollLearnerInCourse()**: Enrolls a learner in a course on Moodle. Returns true if successful.  **fetchCourseContent()**: Fetches the content for a specific course.  **autoLogin()**: Automatically logs the user in using their userId. Returns true if successful. |
| 11 | Recommendation Engine | None | **generateRecommendations()**: Generates a list of recommended learning content for the learner. |

**Class Diagram:**

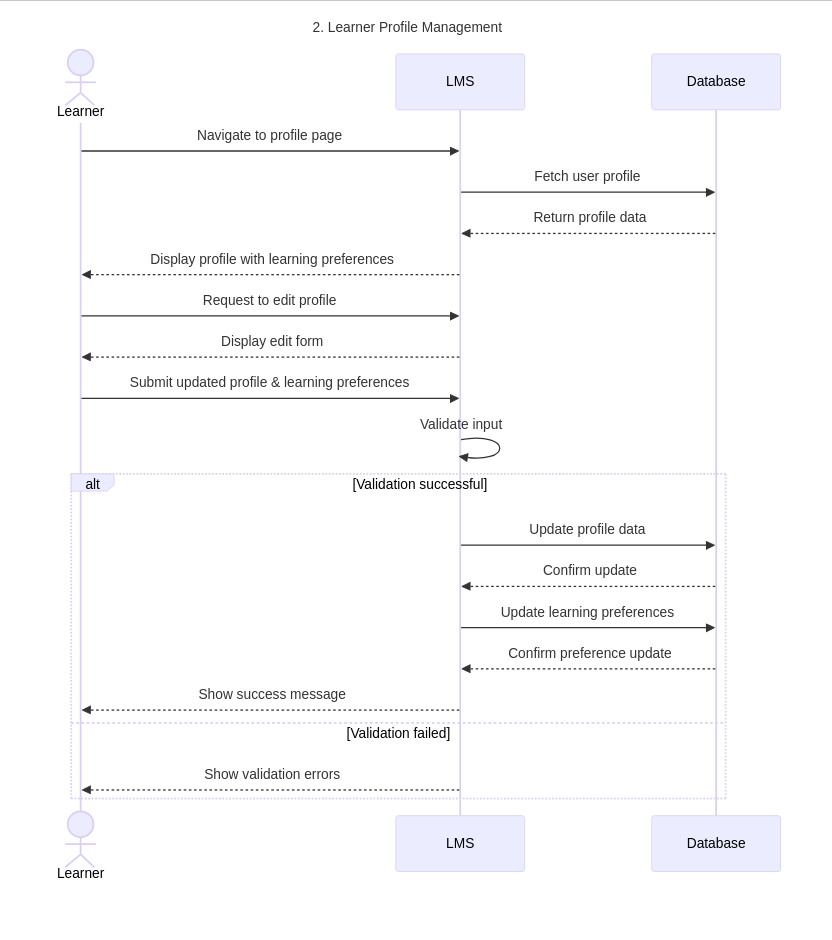
****

**Sequence Diagrams**

# **1. Learner Accessing Recommending Courses and Resources**



**2. Learner Profile Management**



# **3. Coach Managing Courses and Content**

# 

# 

# 

# 

# 

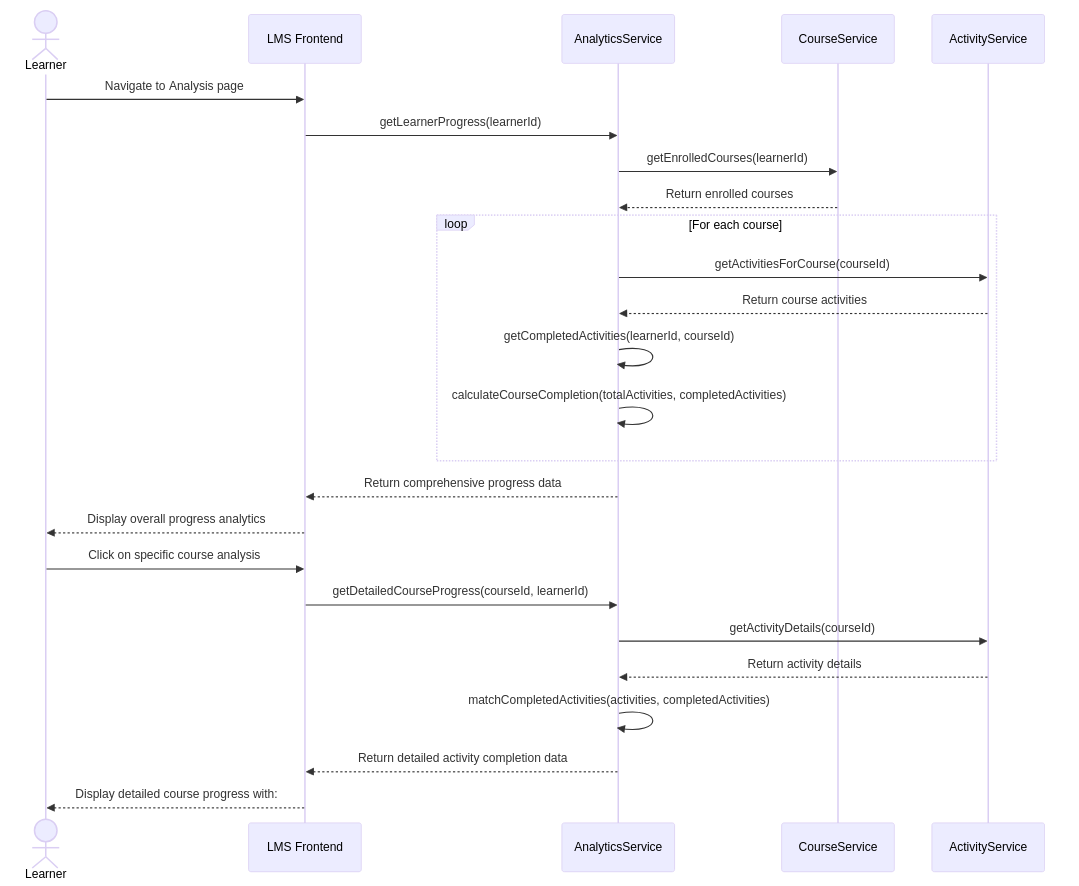
# 

# 

# 

# 

# **4. Learner Progress Tracking**



# 

# 

# 

# **Design Rationale**

**1. Decision on Custom Build vs Building on top of Moodle**

**Alternatives Considered:**

* Build a custom course management system from scratch.
* Use Moodle LMS.
* Use other LMS like Canvas.

**Final Choice:**

* **Moodle LMS** was chosen due to its open-source nature, flexibility, and support for structured course management.

**Rationale:**

* Moodle enables efficient course management without the need for custom development. While integration and customization require effort, its flexibility outweighs the costs. Custom systems demand significant resources, and alternatives like Canvas have licensing costs and limited customization.

**2. Server Allocation and Scalability**

**Alternatives Considered:**

* Single server for both backend and frontend.
* Two nodes one for Moodle & frontend, one for backend & recommendation engine.

**Final Choice:**

* Two VM nodes: One for Moodle & frontend, one for backend & recommendation engine.

**Rationale:**

* The two VM setups ensure better performance scaling and fault isolation by keeping the frontend, backend, and recommendation engine separate. The single-server approach was not considered due to potential performance bottlenecks and a higher risk of failure, which could negatively impact user experience and scalability.

**3. User Interface Design**

**Alternatives Considered:**

* Design with all features on one page.
* Focused and simplified UI with a streamlined profile, prioritized learning activities, and additional details in tabs.

**Final Choice:**

* Focused and simplified UI with a streamlined profile and prioritized learning activities.

**Rationale:**

* The chosen UI design prioritizes essential learning activities, offering a streamlined user experience. The alternative approach with all features on one page was not preferred, as it could lead to a cluttered experience, making navigation more complex for users.

### **4. Login and Redirection to Courses in Moodle**

**Alternatives Considered:**

* Manual login for Moodle.
* Auto-login using Single Sign-On (SSO).

**Final Choice:**

* Auto-login through token-based authentication.

**Rationale:**

* Token-based authentication improves user experience by automating the login process. Although backend integration with Moodle authentication is required, this approach ensures a smooth user journey without manual login steps.