level 0 use case diagram-AMS

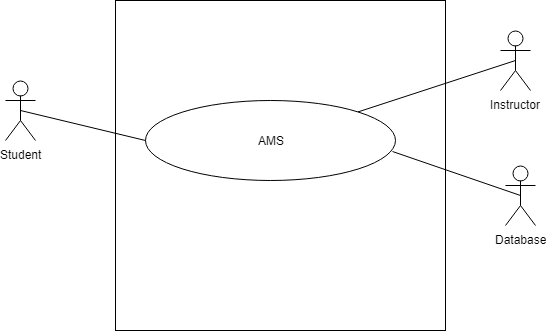


Figure-1: level 0 use case diagram- AMS

Name: Assignment Management System

Primary actor: Instructor, Student, Database

Secondary actor: Result Management System

# level 1 use case diagram-Subsystem

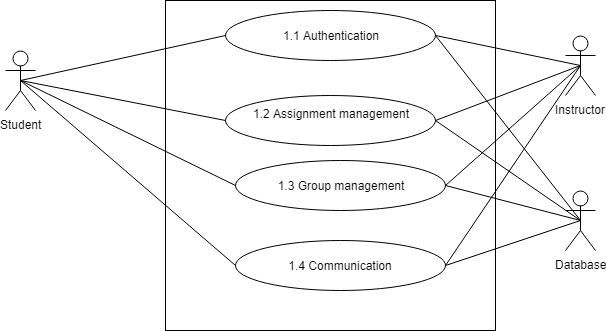


Figure-2: level 1 use case diagram - Subsystem

Name: Subsystem of AMS

Primary actor: Instructor, Student, Database

Secondary actor: N/A

There are 4 subsystems in the Assignment Management System. They are-

* Authentication
* Assignment management
* Group management
* Communication

# level 1.1 use case diagram- Authentication

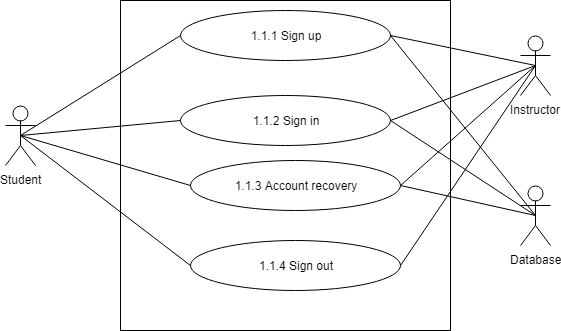


Figure-3: level 1.1 use case diagram – Authentication

Name: Authentication of AMS

Primary actor: Instructor, Student, Database

Secondary actor: N/A

# Description of level 1.1 use case diagram-

Authentication is the process of determining whether someone or something is, in fact, who or what it is declared to be. The authentication subsystem of PMS can be divided into four parts. These are:

* Sign up
* Sign in
* Account recovery
* Sign out

## sign up

* Primary actor: Instructor, Student, Database
* Secondary actor: N/A

### Students action/reply

* Action: Student enter information to sign up.
* Reply: System check validity and store information.

### Instructor action/reply

* Action: Instructor enter information to sign up.
* Reply: System check validity and store information.

Database action/reply

* Store valid data.
* Show data successfully store or not.

## Sign in

* Primary actor: Instructor, Student, Database
* Secondary actor: N/A

### Students action/reply

* Action: Student enter information to sign in.
* Reply: System check validity and store information.

### Instructor action/reply

* Action: Instructor enter information to sign in.
* Reply: System check validity and store information.

Database action/reply

* Store valid data.
* Show data successfully store or not

## Account Recovery

* Primary actor: Instructor, Student, Database
* Secondary actor: N/A

### Students action/reply

* Action: Student enter email.
* Reply: System check validity and store information.

### Instructor action/reply

* Action: Instructor enter information to sign in.
* Reply: System check validity and send pin.

Database action/reply

* Store modified data.
* Show data successfully store or not

# level 1.2 use case diagram- Assignment management

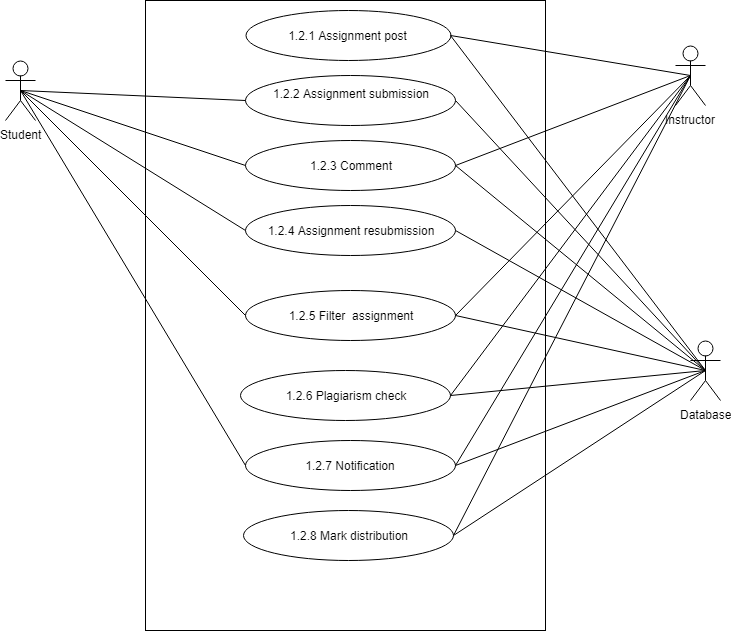


Figure-4: level 1.2 use case diagram- Assignment management

Name: Assignment management of AMS

Primary actor: Instructor, Student, Database

Secondary actor: N/A

# Description of level 1.2 use case diagram-

There are 8 subsystems in Assignment subsystem. These are-

* Assignment post
* Assignment submission
* Comment
* Assignment resubmission
* Filter assignment
* Plagiarism check
* Notification
* Mark distribution

# Level 1.2.1 use case diagram- Post Assignment

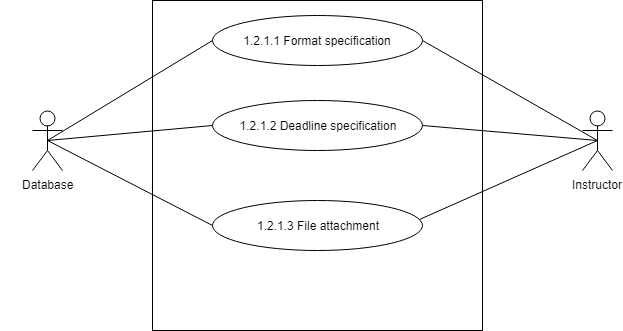


Figure-5: level 1.2.1 use case diagram- Post assignment

Name: Post Assignment of AMS

Primary actor: Instructor, Database

Secondary actor: N/A

# Description of level 1.2.1 use case diagram-

There are 3 subsystems of Post assignment subsystems. These are-

* Format specification
* Deadline specification
* File attachment

## 1.2.1.1 Format specification

* Primary actor: Instructor, Database
* Secondary actor: N/A

### Instructor action/reply

* Action: Select a format.
* Reply: Format selected.

### Database action/reply

* Action: Store format of assignment.
* Reply: Format stored.

## 1.2.1.2 Deadline specification

* Primary actor: Instructor, Database
* Secondary actor: N/A

### Instructor action/reply

* Action: Give deadline for assignment.
* Reply: Deadline confirmed.

### Database action/reply

* Action: Store deadline of assignment.
* Reply: Deadline stored.

## 1.2.1.3 File attachment

* Primary actor: Instructor, Database
* Secondary actor: N/A

### Instructor action/reply

* Action: Select a file to attach.
* Reply: File attached.

### Database action/reply

* Action: Store attach file of assignment.
* Reply: File stored.

## 1.2.2 Assignment submission

* Primary actor: Student, Database
* Secondary actor: N/A

### Student action/reply

* Action: Students submit assignment.
* Reply: System will check format and requirements of corresponding assignment and show a message whether it is submitted or not.

### Database action/reply

* Action: Store the assignment.
* Reply: Show assignment successfully store or not.

## 1.2.3 Comment

* Primary actor: Instructor, Student, Database
* Secondary actor: N/A

### Instructor action/reply

* Action: Instructor comment on any post
* Reply: Commented on post.

### Student action/reply

* Action: Student comment on any post
* Reply: Commented on post.

### Database action/reply

* Action: Store comment in the database.
* Reply: Show a message whether it successfully store or not.

## 1.2.4 Assignment resubmission

* Primary actor: Student, Database
* Secondary actor: N/A

### Student action/reply

* Action: Submit assignment.
* Reply: System will check format and requirements of corresponding assignment and show a message whether it is submitted or not.

### Database action/reply

* Action: Store the assignment.
* Reply: Show assignment successfully store or not.

## 1.2.5 Filter assignment

* Primary actor: Instructor, Student, Database
* Secondary actor: N/A

### Instructor action/reply

* Action: Instructor select option to filter assignment.
* Reply: Assignment filtered.

### Student action/reply

* Action: Student select option to filter assignment.
* Reply: Assignment filtered.

### Database action/reply

* Action: View filtered assignment.
* Reply: Assignment viewed.

## 1.2.6 Plagiarism check

* Primary actor: Instructor, Database
* Secondary actor: N/A

### Instructor action/reply

* Action: Instructor select assignment folder to check plagiarism.
* Reply: System give corresponding folder path to MOSS.

### Database action/reply

* Action: Evaluating plagiarism of file a link will store on the database.
* Reply: link stored in the database.

## 1.2.7 Notification

* Primary actor: Instructor, Student, Database
* Secondary actor: N/A

### Instructor action/reply

* Action: Instructor select notification to view.
* Reply: Notification viewed.

### Student action/reply

* Action: Student select notification to view.
* Reply: Notification viewed.

### Database action/reply

* Action: Store notification.
* Reply: Notification stored.

## 1.2.8 Mark distribution

* Primary actor: Instructor, Database
* Secondary actor: N/A

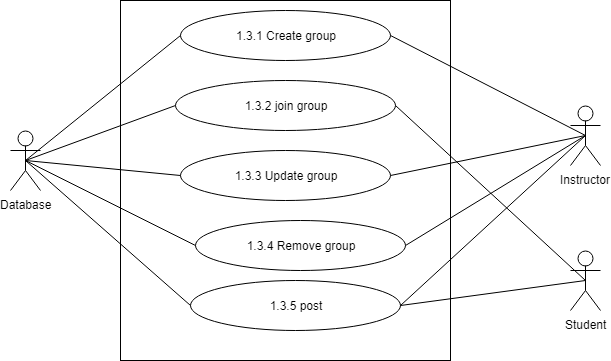
### Instructor action/reply

* Action: Instructor select student to give mark.
* Reply: Mark was given.

### Database action/reply

* Action: Store mark in the database
* Reply: Mark stored.

# Level 1.3 use diagram- Group management



. Figure-5: level 1.3 use case diagram- Group management

Name: Group management of AMS

Primary actor: Instructor, Student, Database

Secondary actor: N/A

# Description of level 1.3 use case diagram-

There are 5 subsystems in group management subsystem. These are-

* Create group
* Join group
* Update group
* Remove group
* Post

## 1.3.1 Create group

* Primary actor: Instructor, Database
* Secondary actor: N/A

### Instructor action/reply

* Action: Instructor give group name, section and subject to create a group.
* Reply: Group was created.

### Database action/reply

* Action: Store group information.
* Reply: Information stored.

## 1.3.2 Join group

* Primary actor: Student, Database
* Secondary actor: N/A

### Student action/reply

* Action: Student enter code to join group.
* Reply: System show a message whether student is entered or not.

### Database action/reply

* Action: Store student information.
* Reply: Information stored.

## 1.3.3 Update group

* Primary actor: Instructor, Database
* Secondary actor: N/A

### Instructor action/reply

* Action: Instructor edit group information to update.
* Reply: Group was updated.

### Database action/reply

* Action: Store update group information.
* Reply: Information stored.

## 1.3.4 Remove group

* Primary actor: Instructor, Database
* Secondary actor: N/A

### Instructor action/reply

* Action: Instructor edit group information to update.
* Reply: Group was updated.

### Database action/reply

* Action: Store update group information.
* Reply: Information stored.

## 1.3.5 Post

* Primary actor: Instructor, Student, Database
* Secondary actor: N/A

### Instructor action/reply

* Action: Instructor give post.
* Reply: Posted successfully.

### Student action/reply

* Action: Student give post.
* Reply: Posted Successfully.

### Database action/reply

* Action: Store post.
* Reply: Post stored.

## 1.4 Communication

* Primary actor: Instructor, Student, Database
* Secondary actor: N/A

### Instructor action/reply

* Action: Instructor comment or message to communicate.
* Reply: Communicated successfully.

### Student action/reply

* Action: Student comment or message to communicate.
* Reply: Communicated successfully.

### Database action/reply

* Action: Store comment or message.
* Reply: Comment or message stored.