

Phase 2 Report

Course Management Bot



UNSW
A U S T R A L I A

COMP9323 17S2

Lei Zhao, z5047143@student.unsw.edu.au
Yijun Fang, z5061743@student.unsw.edu.au
Minhui Su, z5032791@student.unsw.edu.au
Yuanjie Yang, z5032760@student.unsw.edu.au

The aim of this report is to provide an brief introduction to our project, a course management bot with respect to the researches and investigations we have made so far for. Detailed information will include feature analysis and our design of how the system will be implemented.

Part1.

a) Requirements and features analysis:

Our project, which is a course management bot, aim to offer the following functionalities:

Lectures and Mentors will be able to:

- check/upload lecture materials and other resources
- send email to students interactively or with template of different purposes
- check/set/delete reminders for himself or students for upcoming events
- check/set/modify assignment information and mark assignments with the bot interactively

Students will be able to:

- check lecture materials and other resources
- send emails to lecturer/mentors interactively
- view/delete/set reminders for user himself for upcoming events
- check/submit assignment

b) Use-case

Feature: Sending E-mail

- Use case 1: Send email interactively

User	Teacher and student
Workflow	<ul style="list-style-type: none"> - User sends a request to send email - Bot asks who the receiver(s) is/are - User replies receiver email address - Bot asks what the subject is - User replies subject's content - Bot asks what the body is - User replies body's content
Precondition	User needs to login
Result	Email sent

- Use case 2: Send email with template

User	Teacher
Workflow	<ul style="list-style-type: none"> - User sends a request to send email to somebody with condition(ie. sick) - Bot gives a corresponding email templates for user to confirm

	- User confirms and sends or cancels
Precondition	User needs to login
Result	Email sent

Feature: Class resource

- Use case1: check class resource

User	Teacher or student
Workflow	<ul style="list-style-type: none"> - User sends a request to check resources of class with the key word(e.g. Resources name) - Bot lists all matched resources
Precondition	User needs to login
Result	Get required resources

- Use case2: upload class resource

User	Teacher
Workflow	<ul style="list-style-type: none"> - User sends a request to upload resources - Bot asks the destination - User uploads the file - Bot confirms with user if the destination is correct - User confirms or cancel
Precondition	User needs to login
Result	Resource is uploaded
Postcondition	Resource file is saved and path to it is inserted into database

Feature: Reminder

- Use case 1: List reminders

User	Teacher and student
Workflow	<ul style="list-style-type: none"> - User sends a request to check the reminder with keywords (e.g. Assignment 1) - Bot replies with the list with reminders
Precondition	User needs to login
Result	Reminder information is returned to the user

- Use case 2: delete reminders

User	Teacher and student
Workflow	<ul style="list-style-type: none"> - User sends a request to check reminder - Bot replies with the list with reminders - User can delete one of them.
Precondition	User needs to login At least one reminder
Result	Reminder info is returned to the user
Postcondition	Information of this reminder is removed from the database

- Use case 3: Set reminder

User	Teacher and student
Workflow	<ul style="list-style-type: none"> - User sends a request to set reminder - Bot asks user to input reminding time - User sends the reminding time - Bot asks user to input reminder content (email sent part)
Precondition	User needs to login
Result	Reminder set
Postcondition	The detailed information of reminder is inserted into the database

- Use case 4: Send emails to remind students who have not submitted the assignment

User	Teacher
Workflow	<ul style="list-style-type: none"> - User sends a request to send email to remind students who hasn't submitted the assignment. - Bot replies "which assignment?" - User enters the assignment name.
Precondition	The user needs to log in .
Result	Emails are sent to those who have not submitted the specific assignment.

Feature: Assignment

- Use case 1: check assignments resources

User	Teacher and student
------	---------------------

Workflow	<ul style="list-style-type: none"> - User sends a request to check assignment with keywords like the assignment title - Bot replies with the list of matched assignments resources.
Precondition	User needs to login
Result	Users get required assignment resources

- Use case 2: Set an assignment

User	Teacher
Workflow	<ul style="list-style-type: none"> - User sends a request to set assignment with the title - Bot asks the title of the assignment only if title not given last before - User replies title - Bot asks due date of the assignment - User sets the due date of the assignment - Bot asks the user to upload specification - User uploads description for the assignment - Bot ask what are the files student can upload including file name and type. - User reply a set of file names with suffix - Bot will preview all the information //confirms with the user for all assignment information - User confirms information or cancel - If user confirms, bot will ask if send email to students. - User can reply yes to inform students, no for not. - The email will be sent to students with all the information about the assignment. - The conversation ends.
Precondition	User needs to login
Result	Due date and specification of the assignment are set and could emails to all students to inform them.
Postcondition	The detailed information of assignment is inserted into the database

- Use case 3: Modify assignment information

User	Teacher
Workflow	<ul style="list-style-type: none"> - User sends a request to modify certain assignment - User replies due date or specification - Bot asks for new information - User updates information - Bot confirms with user - User confirms information

Precondition	User needs to login
Result	Due date and specification of the assignment are modified and emails will be sent to all students to inform them.
Postcondition	The detailed information of assignment in the database is modified

- Use case 4: Mark assignment

User	Teacher
Workflow	<ul style="list-style-type: none"> - User sends a request to mark certain assignment - Bot returns an unmarked assignment and ask for result mark - User inputs the mark for that assignment - Bot confirms with user - User confirms information
Precondition	User needs to login
Result	Marks of the assignment are set and emails will be sent to related students.
Postcondition	The mark of assignment is inserted into the database

- Use case 5: Submit assignment

User	Student
Workflow	<ul style="list-style-type: none"> - User sends a request to submit certain assignment - User upload a file - Bot confirms with user - User confirms information
Precondition	User needs to login
Result	Mark of the assignment is set and emails will be sent to related students.
Postcondition	Assignment file is saved and path to it is inserted into database

Feature: Invalid input handling

- Use case 1: User types the wrong information and the bot cannot understand

User	Teacher and student
Workflow	<ul style="list-style-type: none"> - User sends wrong information - The bot goes back to the beginning with the word "Hello"

Precondition	User needs to login
Result	The conversation continues in the previous state

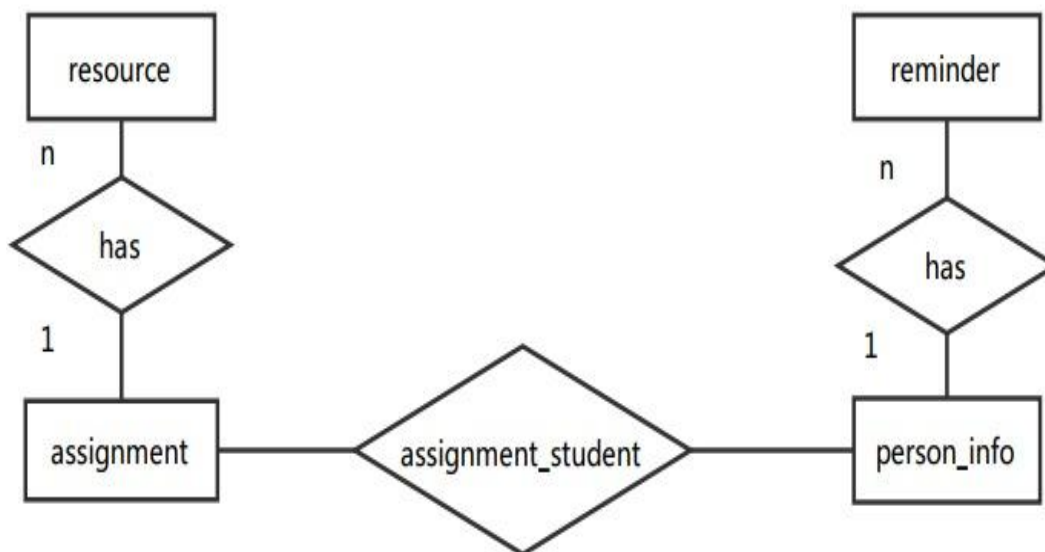
- Use case 2: User types “exit” or leaves the page without finishing the conversation

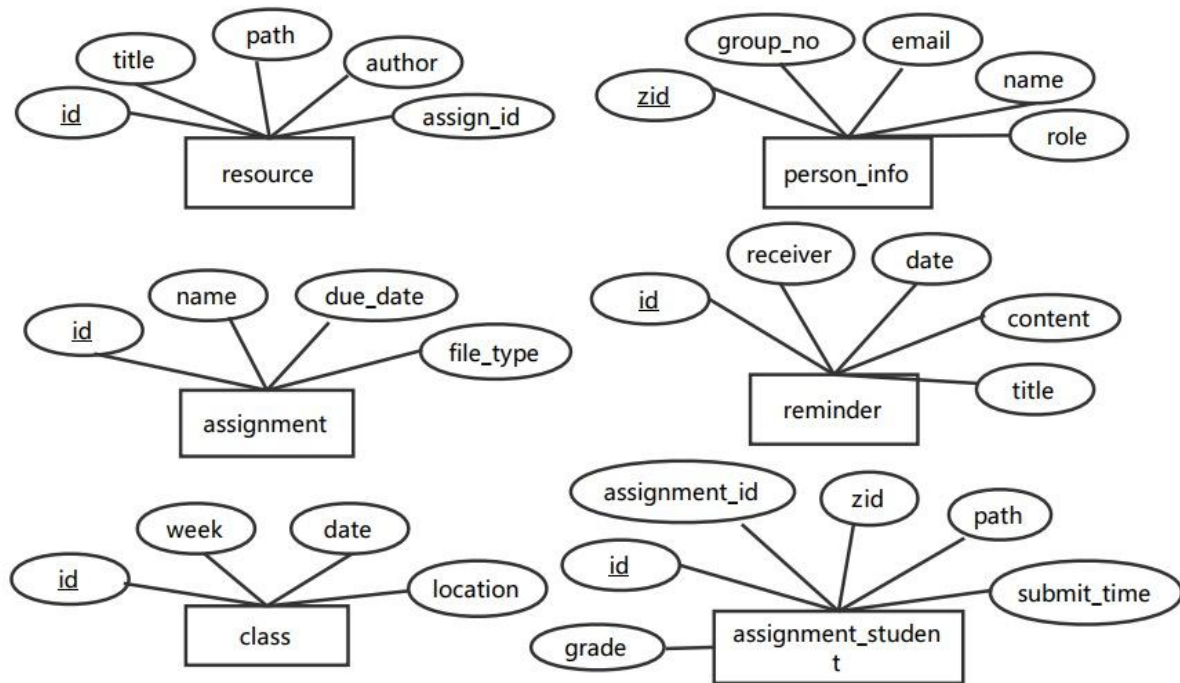
User	Teacher and student
Workflow	- The conversation finishes and no results will be returned
Precondition	User needs to login
Result	The conversation goes back to the beginning state

Part 2. System Design

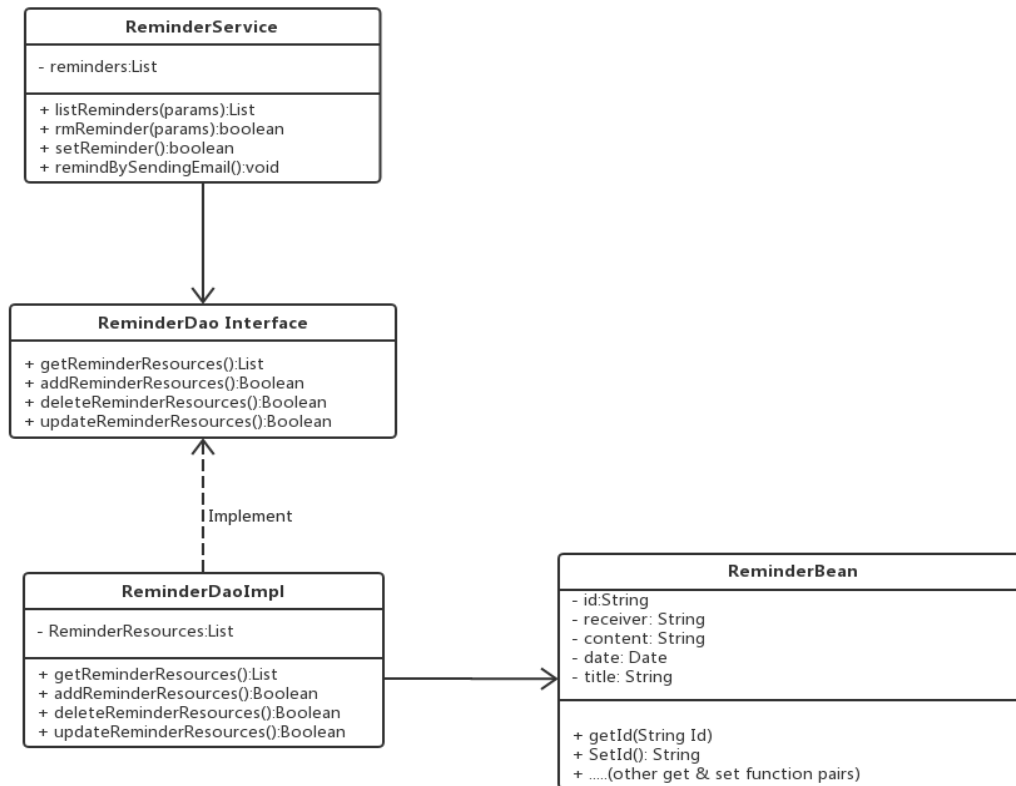
In terms of the front-end, third party instant message software such as Telegram, Facebook Messages and Slack will be used to interact with the user. For the backend, Java Spring will be used to receive request from the user and send requests to the API.AI server to retrieve the intent and parameters. After that, we will use these intent to call the responding functions with parameters and exchange data with database. For the database part, MySQL will be used. In order to interact with the database, we will use the Java MyBatis library.

For the database part, the following is the ER model





The following is the UML diagram:



A simple system architectural diagram can be found below:

