Design Group 3

ASSIGNMENT 4: DESIGN FOCUS

Blake Ruprecht, Zhibin Li, & Lauren Cochran

Architecture and System Design Process

TABLE OF CONTENTS
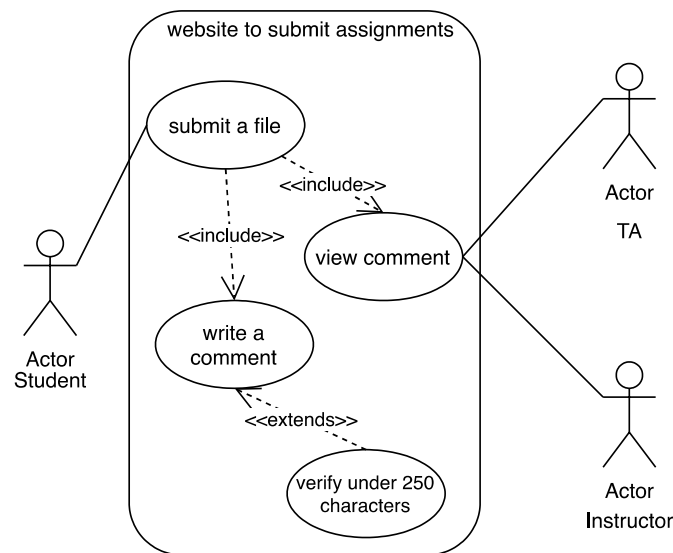
USE CASE DIAGRAM: COMMENTING

Contributor: Lauren

Case Diagram for students commenting on their submission

Student commenting on their submission Use Case Description

Description: Student goes to upload their assignment and is able to add a comment to it
Triggers: Going to the page that lets you submit an assignment, clicking on the text box for the comment
Actors: Student, TAs, Instructor
Preconditions: The student already has an account and is enrolled in the correct class. They also are signed in and are ready to submit an assignment
Post Condition: the comment is able to be read along with the file submission
Goals: The user goes to submit an assignment, clicks on the text box, writes a comment under 250 characters, and then when they're ready to submit the assignment the comment will be posted along with it.
Fail End Condition: The text box will have 0 characters and no comment will be attached to the file
Extensions:
1. Computer crashes
1a. The information in the text box is not saved and has to be retyped
2.Exceeds the character limit
2a. Error message appears

This diagram shows the process of adding a comment to a file submission. It details that when you go to submit a file, a text box for a comment will be there where you can add up to 250 characters. You can view the comment and so can the TAs and Instructors.

ENTITY-RELATIONSHIP DIAGRAM

Contributors: Blake and Lauren during class
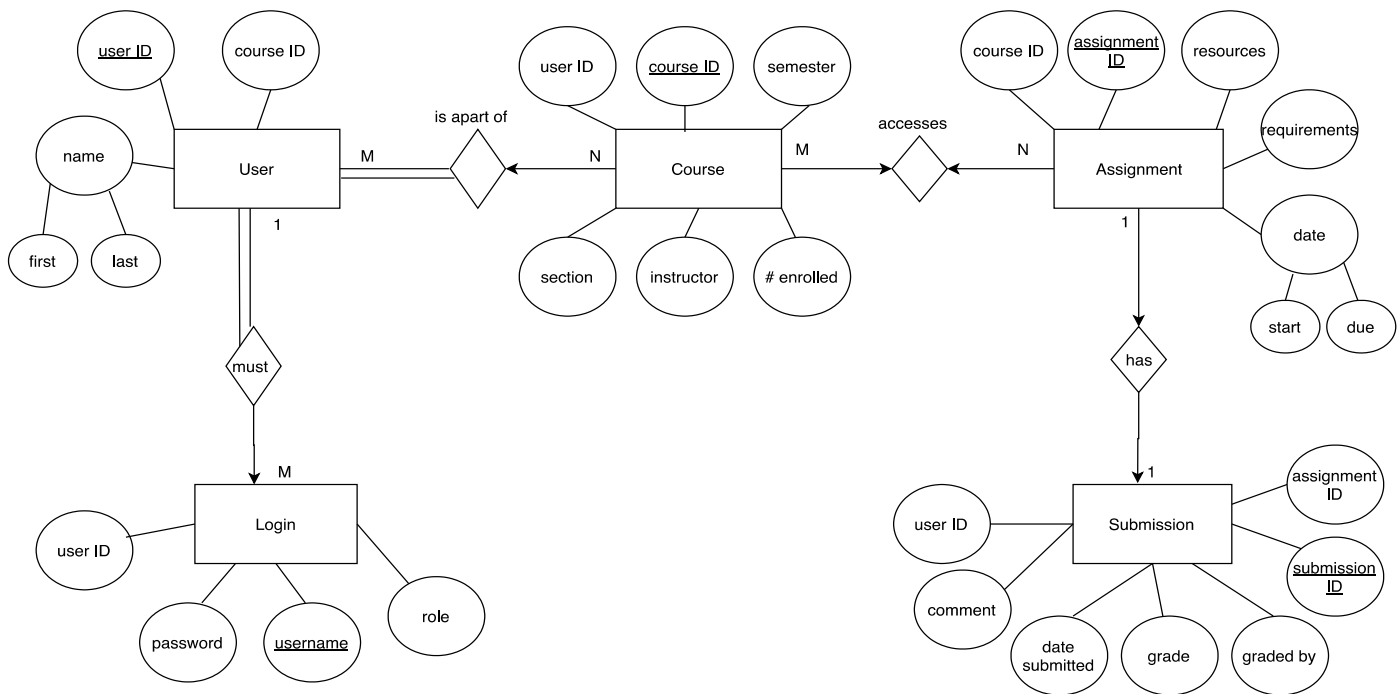
This diagram just details the relationship among the different entities for the homework submitting program. The user entity is for all the different types of users (TA, students, Instructors). The user ID would be able to tell you what type of user they are. The user must Login. The Login entity stores the username and password. The user is also a part of a course, where they are linked through the user ID and course ID. The course has an ID that corresponds to a specific class that can be broken down into sections. It keeps some basic data about each course stored. Then, each course accessed an assignment. They are linked through course ID. Here, the assignment has an assignment ID where multiple assignments can be created for a course. Then the assignment has a submission, connected through assignment ID. This is where each student's actual work is kept.

CLASS DIAGRAM: SUBMITTING ASSIGNMENT

Contributor: Blake and Lauren during class

Class Diagram

Contributors:
Blake Ruprecht, Lauren Cochran

Class diagram displaying the classes used for handling assignment submissions. Assignment has a Status: Boolean attribute that determines whether it is open of closed, as determined by the SetDeadlines() method.

```
               Course
          --------------------------
          Id: int
          Name: string
          --------------------------
          CreateAssignment()
```

```
                                          Assignment
                                   --------------------------
                                   Id: int
                                   Name: string
                                   Status: boolean
                                   --------------------------
                                   SetDeadlines()
                                   AddDescription()
```

```
               Student
          --------------------------
          Id: int
          Name: string
          --------------------------
          ChooseCourse()
          ChooseAssignment()
          UploadFile()
          Submit()
          Resubmit()
```

Contributor: Blake
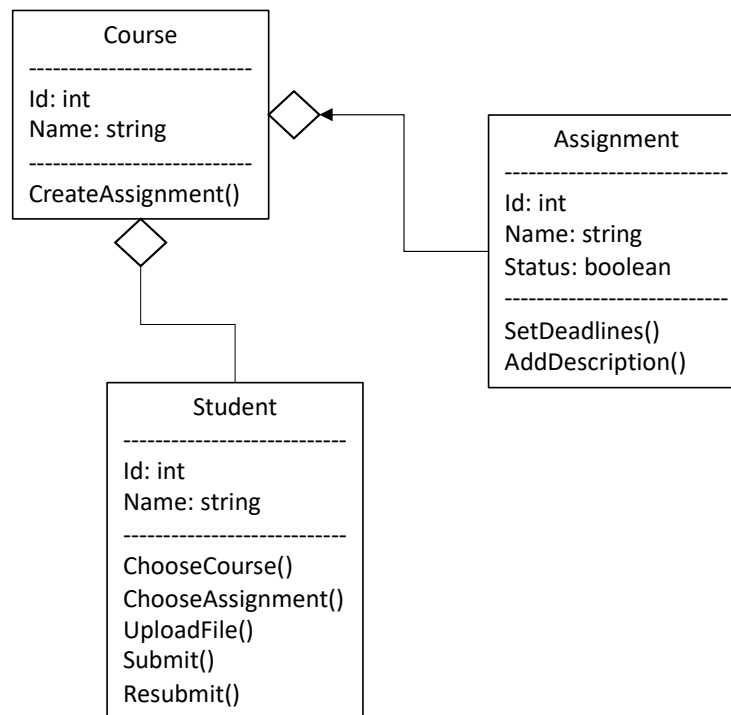
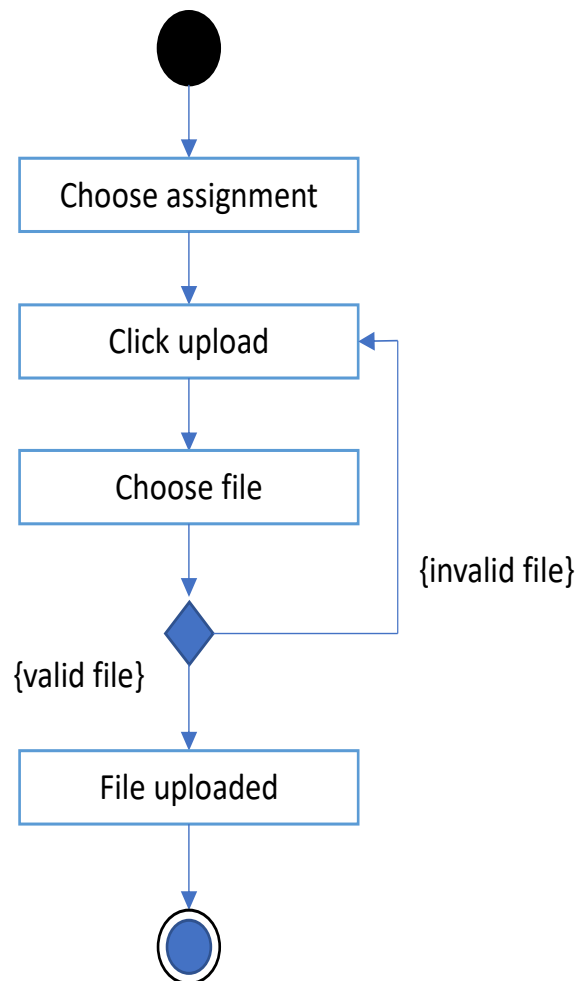## Activity Diagram for Uploading a File

Contributors: Blake Ruprecht

This is an activity diagram for the process of uploading a file, once you are already logged in

Choose assignment

Click upload

Choose file

{invalid file}

{valid file}

File uploaded

ACTIVITY DIAGRAM: ADD/REMOVE TAS

Contributor: Zhibin

select
course

manage TAs

{add}   {remove}

show
availiable
TA

display TA
list

select TA

select TA

enter TA
info

delete TA
info

finish
editing
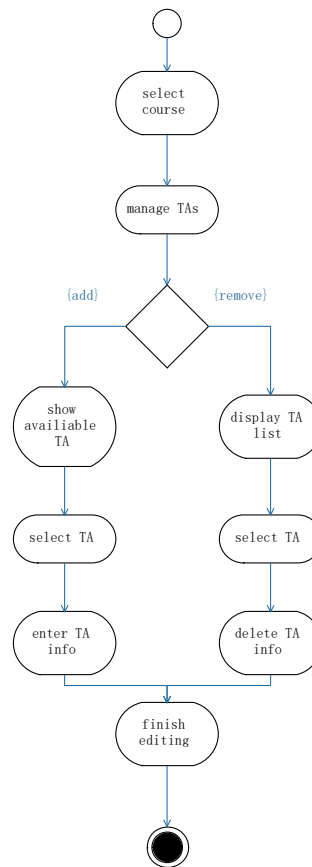
    This diagram shows how the Instructor is able to add or remove a TA to a course through an option that lets you edit/manage TAs to either delete or add them.

ACTIVITY DIAGRAM: INSTRUCTOR LOGIN

Contributor: Lauren

Instructor Login Activity Diagram

```
                    ●
                    │
                    ▼
          ┌───────────────────┐
          │  Instructor enter in │
          │    username and     │
          │      password       │
          └───────────────────┘
                    │
                    ▼
                  ◇ is the
         no    information    yes
                 correct?
        ┌─────────┘       └─────────┐
        ▼                           ▼
┌──────────────────┐      ┌──────────────────┐
│ invalid username and │      │ successfully logged in │
│      password       │      └──────────────────┘
└──────────────────┘                │
        │                           ▼
        ▼                  ┌──────────────────┐
┌──────────────────┐      │ login button changed │
│  display an error   │      │   to logout button   │
│      message        │      └──────────────────┘
└──────────────────┘                │
                                     ▼
                                     ◉
```
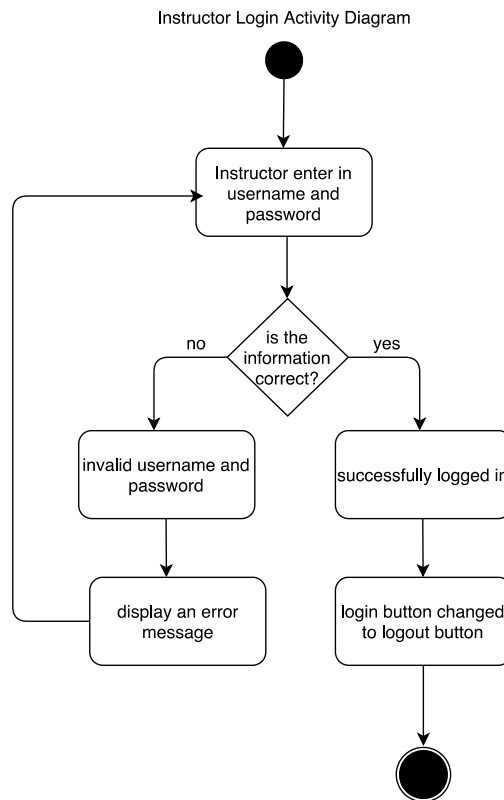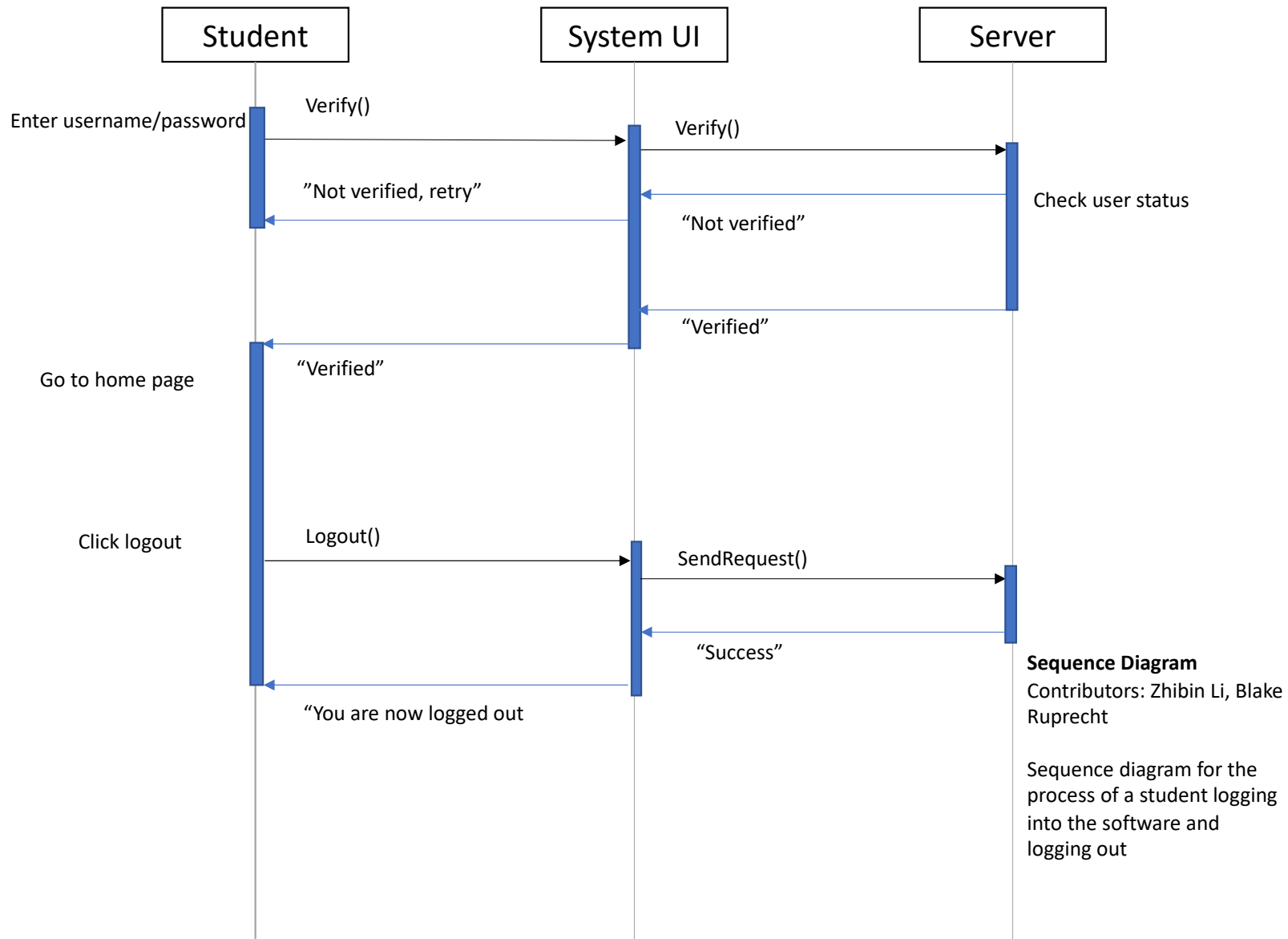
This diagram shows how an Instructor is able to login. They first enter their credentials and based on the validity of those credentials determines if their login was successful or not.
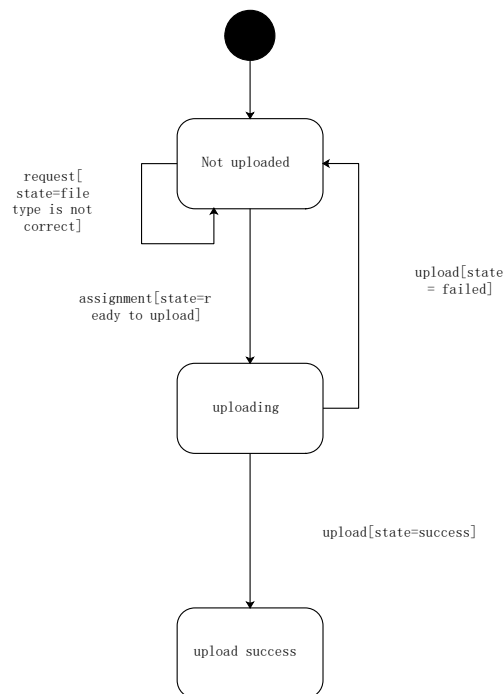
SEQUENCE DIAGRAM: STUDENT LOGIN

Contributors: Blake and Zhibin

Student | **System UI** | **Server**

Enter username/password

Verify()

Verify()

Check user status

"Not verified, retry"

"Not verified"

"Verified"

"Verified"

Go to home page

Click logout

Logout()

SendRequest()

"Success"

"You are now logged out

Sequence Diagram
Contributors: Zhibin Li, Blake Ruprecht

Sequence diagram for the process of a student logging into the software and logging out

STATE MACHINE DIAGRAM: UPLOADING ASSIGNMENT

Contributor: Zhibin

This diagram shows the process of uploading an assignment to the program. If it is not uploaded the state is failed and the assignment state is ready to upload. Once a file has been uploaded, the upload state becomes true.