HW2: Waterfall Project - Requirements Assignment
CS361 Team 05

Application Name: MyDegreeTracker

Table of Contents

Team members

Pair	Name	ONID
1	Serena Tay Alexander Drath	tays dratha
2	Clint Hawkes Alessio Peterson	hawkesc peteales
3	Branden Holloway Samantha Guilbeault	hollowab guilbeas

Requirements Definition

➢ Functional Requirements Definition:

- Write and read access to school employees - student records must be created by faculty
- Read-only access to students - students are able to access their own records
- School administrators are allowed to change degree requirements
- Records are kept of any changes to the degree plans
- Final course grades must be approved by a school administrator
- Students can have multiple degrees/plans
- A student can see their current progress within their selected degree plans
- User has a valid account with appropriate level of access
- Instructor is associated with appropriate classes
- Instructors submit final grades at the end of each term for approval
- Students can view their current degree GPA or their overall GPA
- Status of grades should be reflected(pending/approved/in-progress)
- Administrators should be notified of grades that are pending
- Instructors should be notified when their submitted grades are marked for modification

➢ Non-functional Requirements Definition:

- System should be accessible from any web browser
- Instructors should be able to input grades with ease
- Students should be able to quickly ascertain their degree progress
- Changes to information in the database should be reflected in a timely manner
- Different levels of access (to edit grades, degree requirements, course additions/deletions) provided to different users
- Status of grades should be updated in a timely manner when they are changed
- Access to accounts by users are executed securely
- Sensitive data is handled securely

➢ Use Cases:

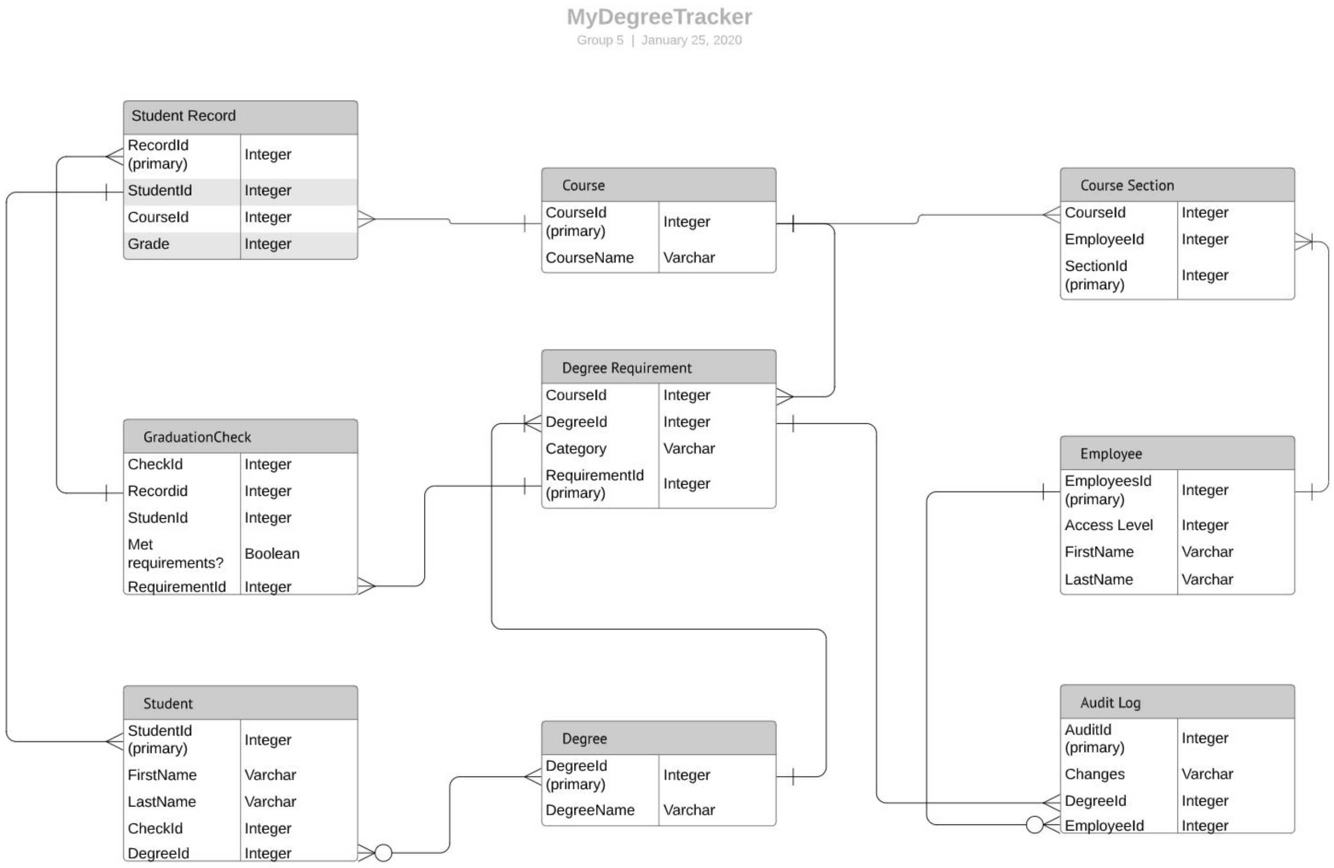
Case One - Student-view Access

- **Actors**
    - Students
- **Preconditions**
    - Student is enrolled in a valid degree plan
    - Student has a valid account
- **Postconditions**
    - None
- **Flow of Events**
    - Student is assigned an account upon enrollment
    - Student is sent notice from administration to active account
    - Student is given login credentials and temporary password
    - Student opens MyDegree tracker
    - Student logs in with temporary password and is asked to create a new one
    - Student creates password which activates the account
    - Student logs in with provided account name and created password
    - Student is able to view their current degree plans and others
    - Student selects degree they want to view
    - Degree requirements are displayed with required/completed depending upon current status
    - Completed courses display final grade
    - Student can choose to view overall GPA or their GPA for each term
    - Selected GPA is displayed
    - Student logout

- **Case Two - Institution-view Access**
  - **Actors**
    - Instructors (enter in grades)
    - Administration (Verifies/approves)
  - **Preconditions**
    - Instructor is listed as the instructor for the specific course/section
    - Instructors has valid account with appropriate level of access
    - Instructor has grades for a class to input
    - Admin has appropriate account for access
    - Admin has a class with pending approval
  - **Postconditions**
    - Instructor update grades and submit for approval
    - Class grades are queued for approval
    - Grades show pending approval status
    - Admin has verified a class's grades
    - Class grades are changed from pending to final
  - **Flow of Events**
    - Instructor opens up MyDegree Tracker app
    - Instructor Logs in to app with appropriate credentials
    - Database verifies credentials
    - List of associated classes appear for Instructor
    - Instructor chooses class to alter student grades
    - Instructor inputs desired data
    - Instructor uploads work for approval
    - A pending approval status is shown for altered class
    - Admin Logs in to app with appropriate credentials
    - Database verifies credentials
    - Admin selects appropriate class to verify grades for
    - Admin decides if grades are approved or if there are discrepancies
    - If approved grades change from pending to approved
    - If discrepancies, class is kicked back to instructor for correction

- **Case Three - Maintenance of database (degree requirements database and course database)**
    - **Actors**
        - Administration (dictate the graduation requirements)
    - **Preconditions**
        - The degree exists and is offered at the university
        - Only administration with super-user access is able to edit the program
        - Have a course list available for students to take
        - Have a list of courses that would satisfy the requirements
    - **Postconditions**
        - Create a change log for an audit trail of all changes made in the database
        - Ensure all requirements are updated and reflected on both student and institution view
    - **Flow of Events**
        - Administrator opens up MyDegreeTracker app
        - Administrator logs into app
        - Database verifies credentials
        - Administration navigates to the correct degree they are trying to edit
        - Administrator selects “Edit Graduation Requirement” button
        - Administration changes the course requirements by adding or removing courses necessary for graduation
        - User confirms that the change is reflected in the app
        - Audit trail to track changes is logged
        - Administration/System does an internal checks that the student interface correctly represents the change
        - Automated notifications to students who are pursuing a degree that the graduation requirements have changed.

➢ Class Diagram:



MyDegreeTracker

Group 5 | January 25, 2020

Student Record	
RecordId (primary)	Integer
StudentId	Integer
CourseId	Integer
Grade	Integer

Course	
CourseId (primary)	Integer
CourseName	Varchar

Course Section	
CourseId	Integer
EmployeeId	Integer
SectionId (primary)	Integer

Degree Requirement	
CourseId	Integer
DegreeId	Integer
Category	Varchar
RequirementId (primary)	Integer

GraduationCheck	
CheckId	Integer
Recordid	Integer
StudenId	Integer
Met requirements?	Boolean
RequirementId	Integer

Employee	
EmployeesId (primary)	Integer
Access Level	Integer
FirstName	Varchar
LastName	Varchar

Student	
StudentId (primary)	Integer
FirstName	Varchar
LastName	Varchar
CheckId	Integer
DegreeId	Integer

Degree	
DegreeId (primary)	Integer
DegreeName	Varchar

Audit Log	
AuditId (primary)	Integer
Changes	Varchar
DegreeId	Integer
EmployeeId	Integer

Requirements Specification

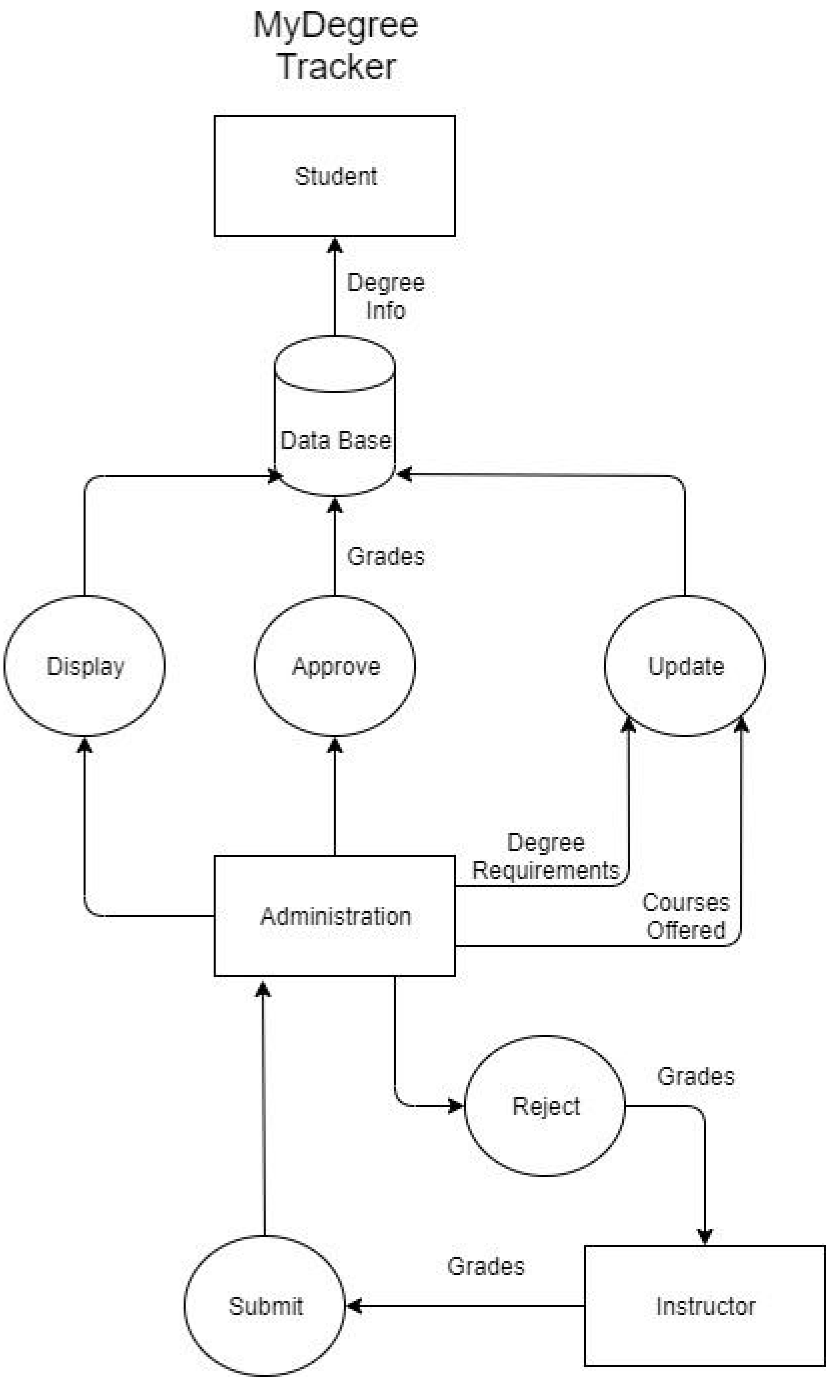
➢ Functional Requirements Specifications:

- A student is added to the database “Student” table upon enrollment in a degree plan
- A student must log in for the first time and set up a password to activate the account
- A new entry is added to the database “Student Record” table for each course a student takes
- A student can view the requirements for all current degree plans
- Administrator’s “Access Level” attribute in the “Employee” table is greater than an Instructor’s
- A student can view grades for their completed courses as well as their overall GPA
- Creating an instructor account will add a new entry to the database “Employee” table
- A new entry is added to the database “Course Section” table for every course taught by instructors each term
- Every time a “Degree Requirement” table record is modified, a new entry is created in the database “Audit Log” table

➢ Non-functional Requirements Specifications:

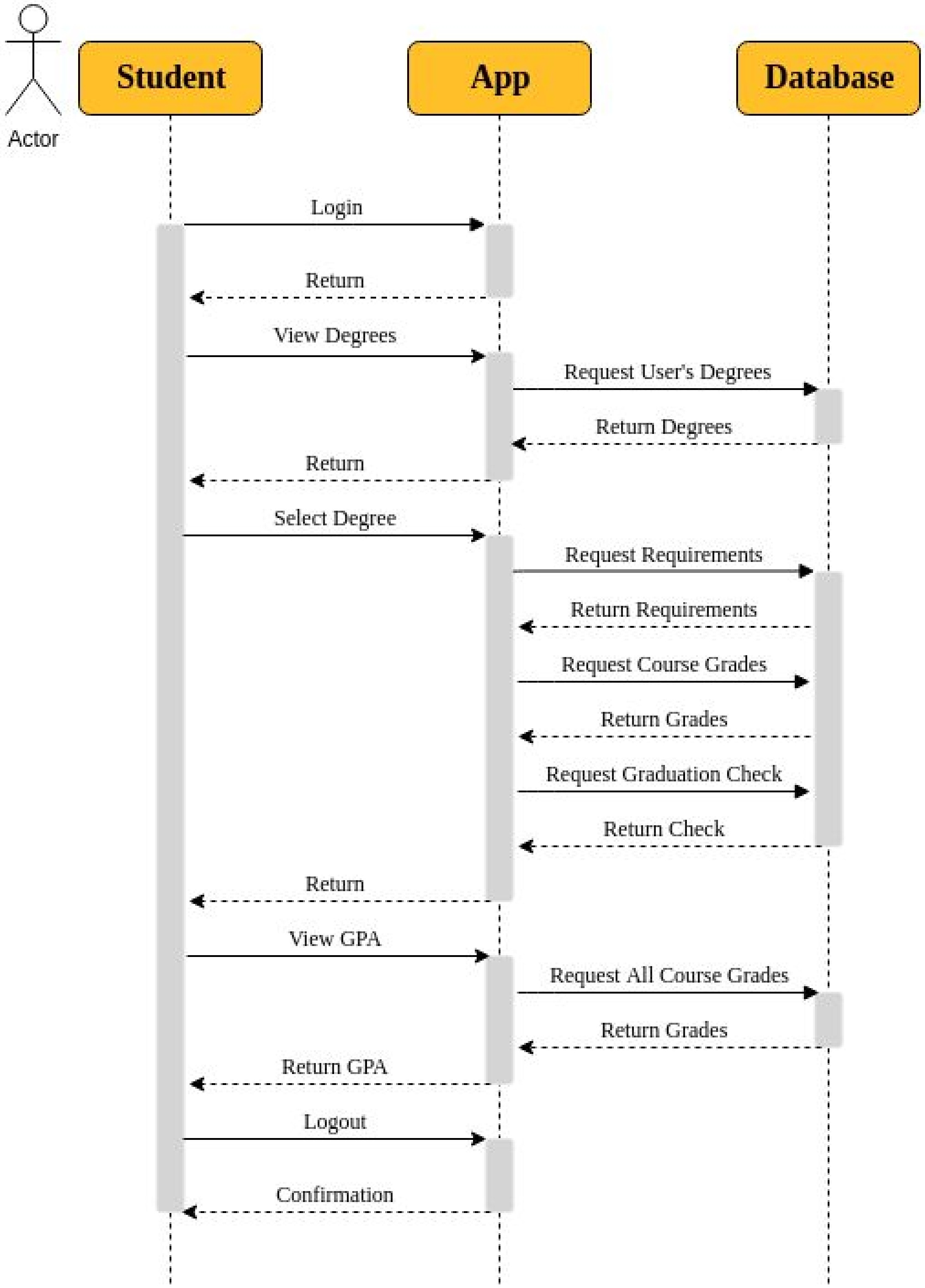
- System should be accessible from any web browser - compatible with IE, Chrome, Firefox, Chrome in Incognito mode, Safari and mobile-friendly versions
- Instructors should be able to input grades with ease - Easy UI interface, filter to courses instructors are currently teaching
- Students should be able to quickly ascertain their degree progress - pulls real-time data to their status/progress within 25 seconds
- Changes to information in the database should be reflected in a timely manner - users edits should be displayed accordingly during the need request/API call to the database
- Different level of access (to edit grades, degree requirements, course additions/deletions) provided to different users - restriction levels implemented to ensure only users with proper credentials are allowed to make edits
- App automatically log users off after 5 minutes of inactivity
- Prompts password resets every 12 months
- Privacy control - Users are not allowed to access private data (SIN, DoB, address) unless they have the appropriate credentials

➢ Dataflow Diagram:

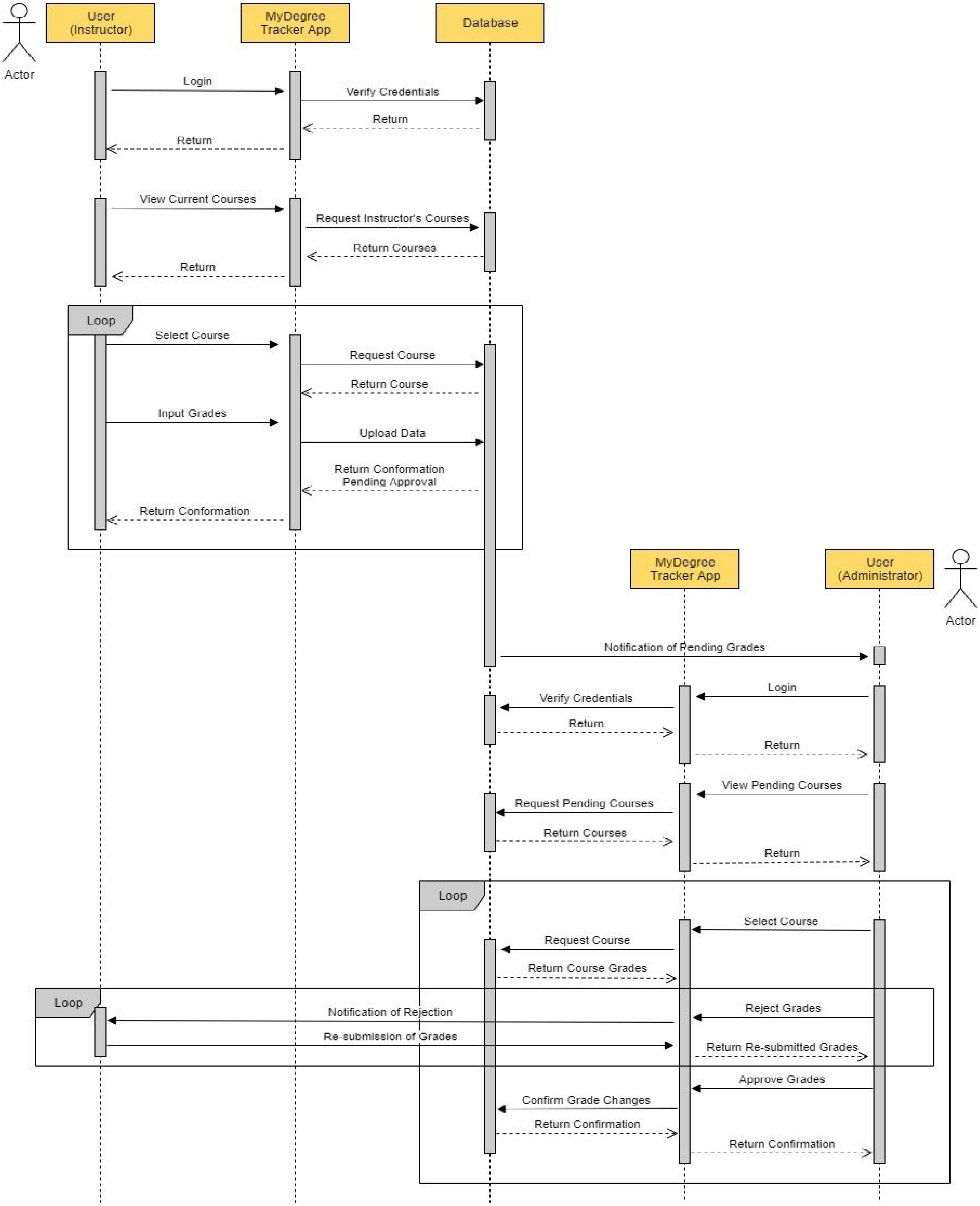


➢ Use Case Charts:

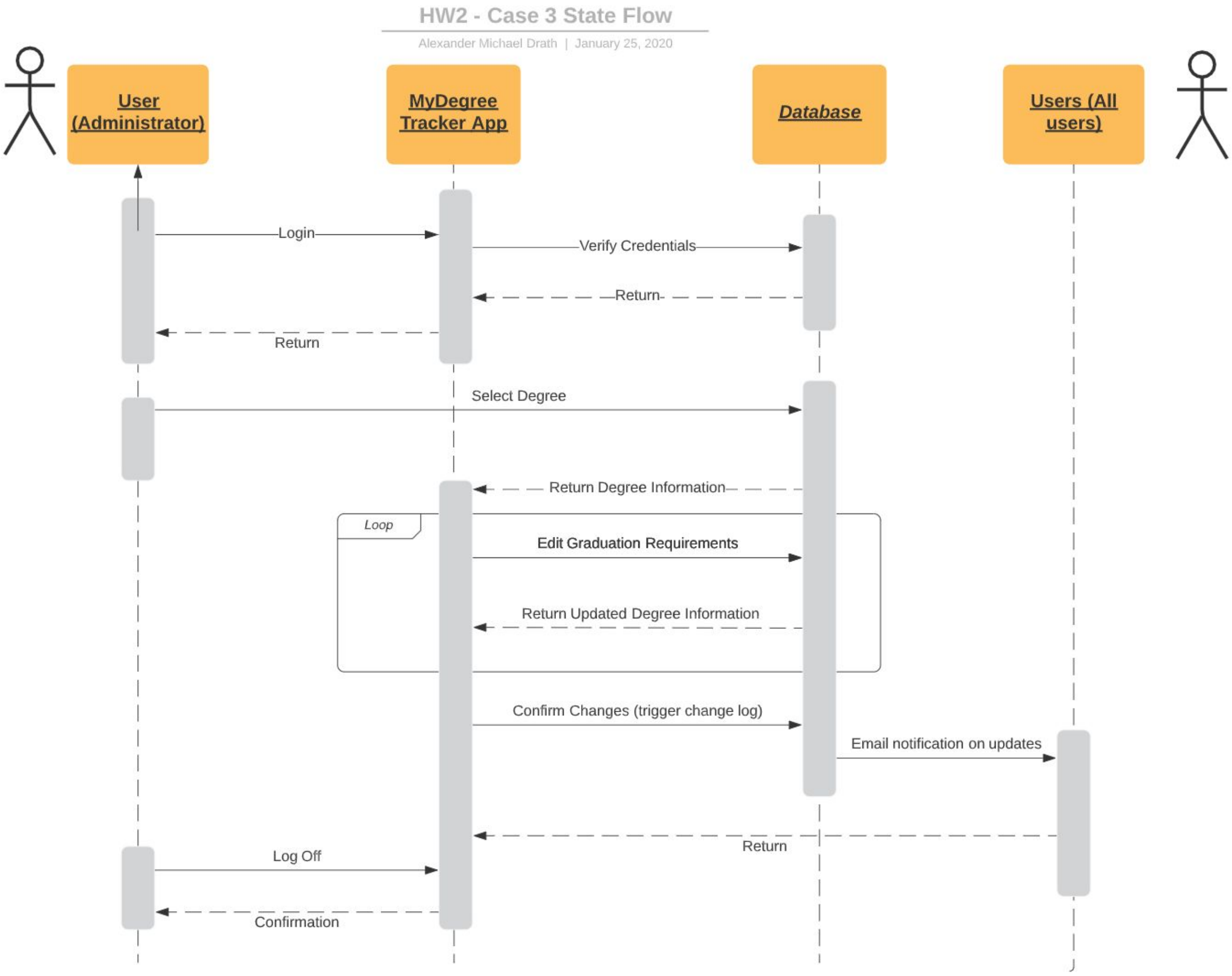
Use Case #1



Use Case #2


User (Instructor)
MyDegree Tracker App
Database
Actor
Login
Verify Credentials
Return
Return
View Current Courses
Request Instructor's Courses
Return Courses
Return
Loop
Select Course
Request Course
Return Course
Input Grades
Upload Data
Return Conformation Pending Approval
Return Conformation
MyDegree Tracker App
User (Administrator)
Actor
Notification of Pending Grades
Login
Verify Credentials
Return
Return
View Pending Courses
Request Pending Courses
Return Courses
Return
Loop
Select Course
Request Course
Return Course Grades
Loop
Notification of Rejection
Reject Grades
Re-submission of Grades
Return Re-submitted Grades
Approve Grades
Confirm Grade Changes
Return Confirmation
Return Confirmation

Use Case #3


HW2 - Case 3 State Flow
Alexander Michael Drath | January 25, 2020
User (Administrator)
MyDegree Tracker App
Database
Users (All users)
Login
Verify Credentials
Return
Return
Select Degree
Return Degree Information
Loop
Edit Graduation Requirements
Return Updated Degree Information
Confirm Changes (trigger change log)
Email notification on updates
Return
Log Off
Confirmation

Project Modifications based on TA feedback

- No changes were needed based on the feedback received.

Contribution Summary

All Members: Function/Non-Functional Requirements

Alexander Drath: Use Case 3, Use Case 3 Flow, ERD diagram

Samantha Guilbeault: Use Case 2 Flow Chart, Data Flow Diagram

Clint Hawkes: Use Case 1, Use Case 1 Chart, final doc review and submission

Branden Holloway: Use Case 2

Alessio Peterson: Use Case1

Serena Tay: Drafted template for HW2 submission on Google Doc, Use Case 3, ERD diagram, Use Case 3 State Flow

Tasks allocation:

- Requirements definition/specification (functional and non-functional)
- ERD Diagram: SERENA/ALEX
- Dataflow: BRANDEN/SAM
- CASE 3: SERENA/ALEX
- CASE 2: BRANDEN/SAM
- CASE 1: CLINT/ALESSIO