# Use Case 1: Waitlisted Student

Use Case ID: UC-01

Use Case Name: Waitlisted Student

Relevant Requirements: Waitlist and Auto-Enrollment, Client-Server Communication

Primary Actor: Students

Pre-conditions:

* Student is already on the waitlist for a course
* Course has at least one enrolled student who drops or capacity is otherwise increased

Post-conditions:

* Student is automatically moved from waitlist to enrolled status
* Student receives a notification about the enrollment change
* Course roster and waitlist are updated accordingly

Main Flow:

* A course seat becomes available
* Server checks waitlist order.
* The server automatically enrolls the next student on the waitlist.
* Server updates the course roster and waitlist.
* The server sends an enrollment confirmation message to the client.
* Client displays notification to the student.

Extensions or Alternate Flows:

* If multiple students are on waitlist, server selects the earliest added student
* If students is already at maximum course load, enrollment is denied and the next waitlisted student is considered

Exceptions:

* Client fails to receive notification due to connection error
* Student account becomes inactive before enrollment

Related Use Cases:

* UC-03 Enroll in COurse (initial enrollment attempt)
* UC-04: Drop Course (seat becomes available when another student drops)

# Use Case 2: Login

Use Case ID: UC-02

Use Case Name: Login

Relevant Requirements: Permissions and Roles,

Primary Actor: Student or Adminstrator

Pre-conditions:

* User has a valid account in the system
* Client application is running and connected to the server

Post-conditions:

* User is authenticated and granted access to their role (student or administrator)
* User is taken to the appropriate menu

Main Flow:

* User selects “Login” from the client application
* User enters username and password
* Client sends credentials securely to the server
* Server verifies credentials and determines role (student/admin)
* Server sends confirmation and role information back to the client
* Client displays the appropriate main menu for the user’s role

Extensions or Alternate Flows:

* If credentials are invalid, server rejects login and client displays error
* If user forgets password they may contact administrator to reset it

Exceptions:

* Server unavailable then client shows connection error
* User account is locked or disabled then login is denied

Related Use Cases:

* UC-03: Browse Courses (after successful login)
* UC-05: Manage Courses (admin access after login)

# Use Case 3: Enroll in Course

Use Case ID: UC-03

Use Case Name: Enroll in Course

Relevant Requirements: Student Controls, Enrollment Verification, Waitlist and Auto-Enrollment

Primary Actor: Student

Pre-conditions:

* Student is logged into the system
* Course exists in catalog

Post-conditions:

* Student is enrolled in the course OR placed on the waitlist if the course is full
* Course roster and/or waitlist is updated in the system

Main Flow:

* Student selects “Enroll in Course” from the client menu
* Student enters the course ID
* Client sends enrollment request to the server
* Server verifies prerequisites and checks for schedule conflicts
* IF valid and seats available, server enroll student
* Server sends confirmation back to client
* Client displays enrollment success message

Extensions or Alternate Flows:

* If course is full, server places student on waitlist
* If prerequisites are not met, enrollment is denied
* If schedule conflicts exist, enrollment is denied

Exceptions:

* Invalid course ID entered then error message is displayed
* Server unavailable then client shows connection error

Related Use Cases:

* UC-02: Login (student must be logged in before enrollment)
* UC-04: Drop Course (student may later drop the course)
* UC-01: Waitlist Notification (student may be auto-enrolled later if waitlisted)

# Use Case 4: Drop Course

Use Case ID: UC-04

Use Case Name: Drop Course

Relevant Requirements: Student COntrols, Client-Server Communication

Primary Actor: Student

Pre-conditions:

* Student is logged into the system
* Student is currently enrolled in the course

Post-conditions:

* Student is removed from the course roster
* If waitlist exists, the first student on the waitlist is auto-enrolled (UC-01)

Main Flow:

* Student selects “Drop Course” from the client menu
* Student enter the course ID
* Client sends drop request to the server
* Server verifies enrollment and processes the drop
* Server updates course roster and waitlist
* Notifies first student who is waitlisted and enrolls them

Extensions or Alternate Flows:

* If student is not enrolled in the course, drop is denied

Exceptions:

* Invalid course ID entered then error is displayed
* Server unavailable then client shows connection error

Related Use Cases:

* UC-03: Enroll in Course
* UC-01: Waitlist Notification

# Use Case 5: Manage Courses

Use Case ID: UC-05

Use Case Name: Manage Courses

Relevant Requirements: Admin Controls, Store Class Data, Client-Server Communication

Primary Actor: Administrator

Pre-conditions:

* Admin is logged into the system

Post-conditions:

* Course catalog is updated with additions, modifications, or deletions

Main Flow:

* Admin selects “ Manage Course” from the client menu
* Admin chooses to add, edit, or remove a course
* Admin enters course details (course ID, department, capacity, prerequisites, schedule, location, etc.)
* Client sends request to the server
* Server updates the course database
* Client displays confirmation of changes

Extensions or Alternate Flows:

* If admin attempts to remove a course with active enrollments, system prompts for confirmation

Exceptions:

* Invalid data entry then error message is displayed
* Server unavailable then client shows connection error

Related Use Cases:

* UC-02: Login (admin access)
* UC-03 Enroll in Course (students use updated catalog)

# Use Case 6: Browse Courses

Use Case ID: UC-06

Use Case Name: Browse Courses

Relevant Requirements: Student Controls, Store Class Data

Primary Actor: Student

Pre-conditions:

* Student is logged into the system
* Course catalog exists with available data

Post-conditions:

* Student views course list and details

Main Flow:

* Student selects “Brows Courses” from the client menu
* Student views list of available courses
* Student can search or filter courses by department, name, or schedule
* Client displays course details from the server

Extensions or Alternate Flows:

* If no course matches filters, client displays “no results

Exceptions:

* Server unavailable then client shows connection error

Related Use Cases:

* UC-03: Enroll in Course (student may browse before enrolling)
* UC-04: Drop Course

# Use Case 7: Generate Reports

Use Case ID: UC-07

Use Case Name: Generate Reports

Relevant Requirements: Print Reports, Run Reports Over Time Period, Client-Server Communication

Primary Actor: Admin

Pre-conditions:

* Admin is logged into the system
* Course and enrollment data exist

Post-conditions:

* Report is generated and displayed/exported

Main Flow:

* Admin selects “Generate Reports” from the client menu
* Admin chooses report type (course list, number of students, percentage full)
* Admin optionally selects a time period or filters
* Client sends request to server
* Server compiles report data from the database
* Server sends report back to client
* Client displays report or allows export

Extensions or Alternate Flows:

* If no data exists for selected filters, system returns an empty report

Exceptions:

* Server unavailable then client shows connection error
* Invalid report parameters then error message

Related Use Cases:

* UC-05: Manage Courses (affects report data)
* UC-03: Enroll in Course (enrollment data used in reports)

# Use Case 8: Client-Server Communication

Use Case ID: UC-08

Use Case Name: Client-Server Communication

Relevant Requirements: Client and Server Applications Communicate through TCP/IP

Primary Actor: Client Application

Pre-conditions:

* Client application is running
* Server is online and accessible over TCP/IP

Post-conditions:

* Data requests are transmitted and responses are received

Main Flow:

* Client sends a request (login, enroll, drop, etc.) to the server
* Server processes the request
* Server sends back a response (success, failure, or data)
* Client displays result to the user

Extensions or Alternate Flows:

* If server takes too long, client times out and displays an error

Exceptions:

* Network failure then client shows “connection lost”
* Invalid data packet then server rejects request

Related Use Cases:

* All use cases rely on this communication layer

# Use Case 9: Search Courses

Use Case ID: UC-09

Use Case Name: Search Courses

Relevant Requirements: Student Controls, Store Class Data

Primary Actor: Student

Pre-conditions:

* Student is logged into the system
* Course catalog exists with searchable data

Post-conditions:

* Student is shown results matching their search criteria

Main Flow:

* Student selects “Search Courses from the client menu
* Student enters search criteria (course ID, name, department, time)
* Client sends query to server
* Server filters catalog data
* Server sends search results back to client
* Client displays list of matching courses

Extensions or Alternate Flows:

* If no course match then system displays “no results found”

Exceptions:

* Invalid input then error message

Related Use Cases:

* UC-06: Browse Courses
* UC-03: Enroll in Course

# Use Case 10: View Student Schedule

Use Case ID: UC-10

Use Case Name: View Student Schedule

Relevant Requirements: Student COntrols, Help Students Plan Non-Conflicting Courses

Primary Actor: Student

Pre-conditions:

* Student is logged into the system
* Student has enrolled in at least one course

Post-conditions:

* Student views current schedule of all enrolled courses

Main Flow:

* Student selects “View Schedule” from the client menu
* Client sends request to server
* Sever retrieves student’s enrolled courses
* Server sends schedule data to client
* Client displays list or calendar view of courses

Extensions or Alternate Flows:

* If student is not enrolled in any courses, system displays “no courses scheduled”

Exceptions:

* Server unavailable then the client shows a connection error

Related Use Cases:

* UC-03: Enroll in Course
* UC-04: Drop Course
* UC-08: Plan Courses