Streamline Tutoring Customer Stories with Test Cases

Rohin Gilman, Evan Schwarzenbach, Stephen Watson

Login Process

Login

When the user arrives at the site, they will have to login with their university's credentials. If the user is an administrator, they will be directed to the administrator page. If the user is a tutor, they will be directed to their tutor page. If the user is a student, they will be directed to their student page. If a user fits more than one category, they will be directed to the highest level page they have access to and will have a button which allows them to switch between their profiles.

Test Case: Login

Inputs:

- Username: (user entry)
- Password: (user entry)
- Valid Credentials: true
- Rank: Admin, Tutor, Student

Expected Output:

- Login(User, Rank)

Actual Output:

- ?

Pass/Fail Criterion:

- Expected Output == Actual Output

Incorrect Credentials

If the user enters incorrect credentials, they will be redirected to the login page with a login error message.

Test Case: Incorrect Credentials

Inputs:

- Username: (user entry)
- Password: (user entry)
- Valid Credentials: false

Expected Output:

- Login Error Screen

Actual Output

- ?

Pass/Fail Criterion:

- Expected Output == Actual Output

Administrator Profile

Menu

The user will have access to a menu button which allows them to select Students, Tutors, Courses, and Settings.

Test Case: Menu

Inputs

- Login(User,Rank)

Expected Output

- If(Rank == Admin) Display menu
  - Display Students

    - Display Tutors
    - Display Courses
    - Display Settings

Actual Output

- ?

Pass/Fail Criterion

- Expected Output == Actual Output

Students

The user will see a list of students which they can search by ID number, login, etc. to select an individual student. That student’s schedule will then appear on the large schedule. Additionally, the administrator will be able to download that student’s appointment data.

The user will also be able to select a student and promote them to a tutor or administrator or demote tutors to students.

Test Case: Students

Inputs

- Search (ID number or Name)
- Students menu (select Student)
- Promote(Student, ID number, New Rank)
- Demote(Tutor, ID number)

Expected Output

- Student Schedule
- Student Data
- Promote()
    - (ID number, Student) => (ID number, New Rank)
- Demote()
    - (ID number, Tutor) => (ID number, Student)

Actual Output

- ?

Pass/Fail Criterion

- Expected Output == Actual Output

Tutors

The user will see a list of tutors which they can search by ID number, login, course, etc. to select an individual tutor.

The tutor's schedule will then appear on the large schedule. The user will have the ability to assign availability to the tutor within the schedule. The user can also assign courses to the tutor through this menu.

Additionally, the administrator will be able to download that tutor's appointment data.

The user will also be able to select a tutor and promote them to an administrator or demote them to a student.

Test Case: Tutors

Inputs

- Search (ID, Name. Course)
- Tutor List (Select Tutor)
- Tutor Schedule (Assign availability, Assign courses)
- Promote (Tutor, ID number, Administrator)
- Demote (Tutor, ID number, Student)

Expected Output

- Tutor list
- Tutor schedule
- Promote()
    - (ID number, Tutor) => (ID number, Administrator)
- Demote()
    - (ID number, Tutor) => (ID number, Student)

Actual Output

- ?

Pass/Fail Criterion

- Actual Output == Expected Output

Courses

The user will see a list of courses offered at the university which they can search by college, department, number, professor, etc.

The user can select a course to view the tutors assigned to that course, assign a tutor to that course, remove a tutor from that course. Additionally, the administrator will be able to download that course's appointment data.

Test Case: Courses

Inputs

- Search (College, Department, Number, Professor, Subject)
- Course List (Select course)
- Selected Course (Assign tutor, Remove tutor)

Expected Output

- Course List
- Course Options
- Assign()
    - (ID number, Tutor, Not assigned) => (ID number, Tutor, Assigned)
- Remove()
    - (ID number, Tutor, Assigned) => (ID number, Tutor, Not assigned)

Actual Output

- ?

Pass/Fail Criterion

- Actual Output == Expected Output

Main Screen

The user will have access to a large schedule which shows the current week. The schedule will show the schedule of the previous student or tutor that they selected. Otherwise, the schedule will be blank.

The user will also have access to a logout button which sends them to the login page.

Test Case: Main Screen

Inputs

- Last selected student/tutor (if one exists)
- Select Logout

Expected Output

- Student/tutor/blank schedule
- Logout()
  - Returns to login screen

Actual Output

- ?

Pass/Fail Criteria

- Actual Output == Expected Output

Tutor Profile

Main Screen

The user will have access to a large schedule which shows the current week. The schedule will show the user’s availability for that week. The schedule will show any students that have signed up for the user’s appointments and the course that they signed up for.

The user will also have access to a logout button which sends them to the login page.

Test Case: Main Screen

Inputs

- Login(User,Rank)
- Select Logout

Expected Output

- If(Rank == Tutor)
  - Display Schedule
  - Display Appointments
  - Display Availability
- Logout()
  - Returns to login screen

Actual Output

- ?

Pass/Fail Criteria

- Actual Output == Expected Output

During an Appointment

When the appointment begins, the user will select the student on the schedule. and mark them as checked in. When the appointment ends, the user will select the student in the schedule and mark them as checked out. The user can also append any notes about the appointment in a comment box on the student profile.

Test Case: Student Check-in/Checkout/Notes

Inputs:

- Students menu (select Student)
- Check-In
- Check-Out
- Note(user input)

Expected Output:

- If(Check-In)
  - Student Marked Checked-in
- If(Check-Out)
  - Student Marked Checked-out

- If(Note)
    - StudentNote = Note

Actual Output:

- ?

Pass/Fail Criterion:

- Expected Output == Actual Output

Student Profile

Main Screen

The user will have access to a large schedule which shows the current week. The schedule will show the user's scheduled appointments for that week.

The student will be able to select a course that they are enrolled in to see the available appointments for that course. The user can select one of these appointments to add to their schedule, as long as it does not conflict with another appointment.

The user will also have access to a logout button which sends them to the login page.

Test Case: Main Screen

Inputs

- Login(User,Rank)
- Select Logout

Expected Output

- If(Rank == Student)
    - Display Schedule
    - Display Appointments
    - Display Student Courses
- Logout()
    - Returns to login screen

Actual Output

- ?

Pass/Fail Criteria

- Actual Output == Expected Output

Test Case: Appointment

Inputs:

- Course menu (select Course XXXX)
- Appointment menu (select Appointment XXX)

Expected Output:

- AddSchedule(Appointment XXX)

Actual Output:

- ?

Pass/Fail Criterion:

- Expected Output == Actual Output