InstructED

Software Test Plan

CSCI-P465/565 (Software Engineering I)

Project Team

Kevin Cao

Andrew Jedlicka

Blake Miller

Shaun Trimm

1. Overview (Sprint 1)
   1. Test Objectives
      * Verify a user can register an account
      * Verify a user can log into their account using their email and correct password
      * Verify a user can reset their password using a password reset link sent to the email tied to their account.
      * The password a user submits is encrypted
      * Verify a user can log using Facebook Oauth.
      * Verify Duo 2 Factor Authentication works, and is required to log in.
   2. Test Environment
      * 1: Local testing on a Linux system (Arch Linux), using the Brave browser (Chromium) and a locally hosted Postgresql database
      * 2: Non-Local testing using a Heroku based web server, and Postgresql database server
   3. Test Personnel
      * Kevin Cao
   4. Acceptance Criteria
      * A user can register and login, as well as reset their password. All links and buttons direct to where they are supposed to and no pages that can only be accessed by users can be accessed by non-users.
   5. Noted Omissions
2. Test Cases (Sprint 1)
   1. Test Case 1: Normal Login
      * Description: A user with an account would like to log into their InstructED account
      * Initial Conditions: User must already have an account, and know their password and email address
      * Input Data: Email of the account, password for the account, and phone number (for Duo Authentication)
      * Specifications: User can log into their account with the correct email and password, and Duo 2FA works
      * Procedure: Using an account created using one of our emails, a password we know, and our phone number we would test the login procedure
   2. Test Case 2: Forgot Password
      * Description: A user has forgotten their password, and wants to reset it
      * Initial Conditions: User must already have an account, know their email address and have access to the email address tied to their account.
      * Input Data: Email of the account, and the new password.
      * Specifications: A user can reset their password using a password reset link sent to the email tied to their account
      * Procedure: We would use the email associated with a test account to reset the password of the test account, then attempt to login with the new password.
   3. Test Case 3: Registration
      * Description: A new user would like to create an InstructED account.
      * Initial Conditions: New user must have an email address to use for their account, and a password to provide.
      * Input Data: Email of the new user, first name, last name, and password
      * Specifications: User can log into their account with the correct email and password.
      * Procedure: We would use one of our emails, and a password to register for an account, then verify it exists by checking the database and logging in.
   4. Unsuccessful Login
      * Description: A user would like to log in, but either types the wrong email address or wrong password
      * Initial Conditions: User must have an account
      * Input Data: Erroneous password/email address inputted
      * Specifications: User can log in with their correct email address and their correct password
      * Procedure:
   5. Login/Register With a Facebook Account
      * Description: A user would like to use their Facebook account to log into their InstructED account
      * Initial Conditions: The user must have a Facebook account, and must not have an InstructED account with the same email as their Facebook account
      * Input Data: Email/password for the Facebook account
      * Specifications: User can log into their account using Facebook Oauth
      * Procedure: Use one of our Facebook accounts to log in using Oauth
   6. Test Case 4: Testing Encryption of Passwords
      * Description: After adding a user to the database, the password should be encrypted to ensure security.
      * Initial Conditions: There must be a user in the database, and the person checking the database must know the unencrypted password for this user.
      * Input Data: A query to the database in order to see password info.
      * Specifications: The password a user submits is encrypted
      * Procedure: Add a user to the database, while knowing the password, then query the database to ensure that the password has been encrypted.
3. Overview (Sprint 2)
   1. Test Objectives
      * Verify that students can view upcoming assignments and their due dates
      * Verify students can view announcements for the classes that they are in
      * Verify that the calendar view gives access to the timetable for instructors and students
      * Verify that instructors can view the details of the courses taught by them
      * Verify that Instructors can make announcements for their courses
      * Verify that Admins can view details about all courses and their instructors
      * Verify that Admins will be able to approve and manage uploaded content
   2. Test Environment
      * 1: Local testing on a Linux system (Arch Linux), using the Brave browser (Chromium) and a locally hosted Postgresql database
      * 2: Non-Local testing using a Heroku based web server, and Postgresql database server
   3. Test Personnel
      * Kevin Cao
      * Andrew Jedlicka
   4. Acceptance Criteria
      * Every user should be able to view all information pertinent to them
   5. Noted Omissions
4. Test Cases (Sprint 2)
   1. Test Case 1: Students can view upcoming assignments
      * Description: A student who is enrolled in a class would like to view upcoming assignments
      * Initial Conditions: The account must be enrolled in a class, and at least one class that the student is in needs an assignment.
      * Input Data: None from the user except clicking on the correct widget. UserId to the database.
      * Specifications: The student can view upcoming assignments and the assignments are correctly assigned to a course they are in.
      * Procedure: Sign in using a student account which is enrolled in a course that has an assignment. Then click on the Upcoming Assignments widget and verify the output is correct.
   2. Test Case 2: Students can view announcements for their classes
      * Description: A student who is enrolled in a class would like to view class announcements
      * Initial Conditions: The account must be enrolled in a class, and the class must have at least one announcement made
      * Input Data: None from the user except clicking on the correct widget. UserId to the database.
      * Specifications: The student can view announcements for their courses
      * Procedure: Sign in using a student account which is enrolled in a course that has an announcement. Then click on the Announcements widget and verify the output is correct.
   3. Test Case 3: Calendar view gives access to the timeline for Students and Instructors
      * Description: A Student/Instructor would like to know the assignments for a certain day for all of their courses.
      * Initial Conditions: The account must be enrolled in or instructing a class.
      * Input Data: The user clicks on a day in the calendar. This and the UserId is sent to the database.
      * Specifications: The student/instructor can view assignments for a certain day
      * Procedure: Sign in using a student/instructor account that is enrolled in or instructing a class, then click on a day that is known to have an assignment and verify the output is correct.
   4. Test Case 4: Instructors can view details of their courses
      * Description: An instructor would like to view the details of a course they instruct.
      * Initial Conditions: The account must be an instructor, and be instructing a course.
      * Input Data: The courseID to the database, and the UserId to the database.
      * Specifications: The instructor can view the details of their course.
      * Procedure: Sign in using an instructor account that is instructing a class. Click on a class in the sidebar, then verify the details of the course are correct.
   5. Test Case 5: Verify Instructors can make announcements for their courses
      * Description: An instructor would like to make an announcement for a course they are instructing.
      * Initial Conditions: The account must be an instructor, and must be instructing a course.
      * Input Data: The courseId, and UserId to the database. As well as the announcement title and announcement text from the user.
      * Specifications: The instructor can make an announcement for a class they teach.
      * Procedure: Sign in using an instructor account that is instructing a class. Click on the make announcement widget and fill out all of the fields to make an announcement. Then sign in as a student for the class and verify the announcement is correct and is displayed.
   6. Test Case 6: Verify that Admins can view details about all courses and their instructors
      * Description: An admin would like to view the details of all courses and instructors.
      * Initial Conditions: The account must be an admin, and there must be at least one course in the database.
      * Input Data: None.
      * Specifications: The admin can view all courses and instructors for each course.
      * Procedure: Sign in using an admin account, then view the course list and verify the output is correct and matches the database.