

UWA Assessment 1

Table of Contents

Assessment	2
User Web Application (UWA) Case Study	5
UWA Development	10
System Design Components	11
Program Design.....	11
Main Module.....	11
Functions:.....	11
Use Case: Login User Scenario	12
Database Design:	12
Application Pages.....	12
Grey Box Testing Insight	15
Assessment Instructions (Part 1)	15
Assessment Instructions (Part 2)	16

UWA Assessment 1

Assessment

In this lab session, we'll dive into the Bug Life Cycle and explore each stage in detail. We'll start by reviewing the various stages of the Bug Life Cycle, including report, reproduction, analysis, fix, test, verification, and closure.

Next, we'll look at the bug handling process in more detail, including how bugs are reported, confirmed, and analyzed. We'll also explore how fixes are created, tested, and verified.

Finally, we'll walk through a case study to see the Bug Life Cycle in action. We'll work through the process of reporting a bug, confirming, and reproducing it, analyzing the cause, creating a fix, testing the fix, verifying that the bug has been resolved, and closing the bug.

By the end of this assessment session, you'll have a solid understanding of the Bug Life Cycle and the bug handling process, and you'll be well-equipped to effectively manage and resolve software defects in your own projects.

The SBLC Process Flow

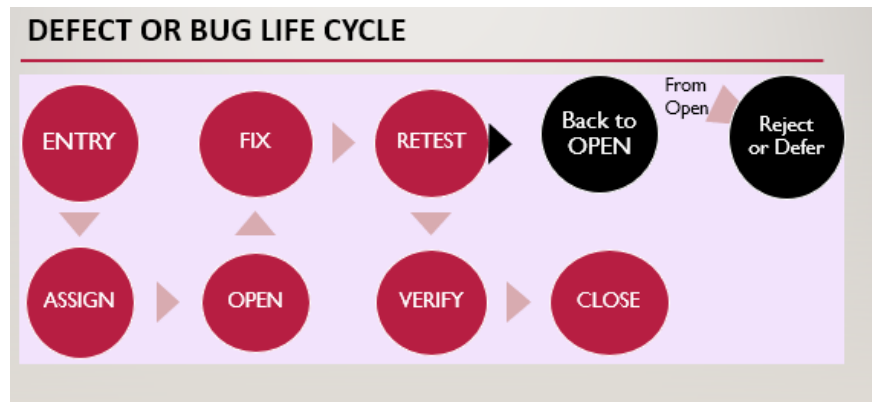
A normal software bug life cycle flow is summarized in the statement below.

Find a bug → report it → reproduce it → analyze it → fix it → retest it → verify it → close it.

Let's look at the cycle in more detail.

UWA Assessment 1

SBLC Flow Diagram



1. When the **tester** finds a new problem, a bug is created, and the status of the bug is set as "New".
2. Once the bug is reported by the tester, the tester or a lead assigns the bug to a developer. The bug status can then be set to ASSIGN or ASSIGNED if those options are available.
3. When the developer starts analyzing and works on fixing the bug, an available status might be OPEN. That helps let everyone know someone is working on the bug.
4. When the developer makes necessary changes to fix the problem, the bug status can be changed to FIX or "Fixed".
5. **Then there is the retest.** The norm is to wait for the next triage. At that time the bug status will be discussed and changed to RETEST. The tester who reported the bug usually does the retesting or issue check to confirm the status. And the bug is reassigned to the tester.
6. When the tester completes retest the bug status can change to REOPEN or VERIFIED or CLOSE. If the same bug is not detected, then the bug is considered fixed, and the status is set as "Verified". On some projects the tester can go ahead and close the bug. But some teams like to review the VERIFIED status in the triage one more time.
7. If the same bug persists even after the fixing, the tester sets the status as "REOPEN" or "REOPENED". Then the previous steps are applied to the bug once again.
8. If the bug no longer exists, then the tester sets the status as "CLOSED" or the next triage session will review and change the status.

UWA Assessment 1

SBLC Exception Handling

- If the bug is repeated twice or corresponds to some other bugs, the status is set as **"DUPLICATE"** by a developer.
- If a developer decides the bug is not a valid defect or it does not affect the functionality of the application, then it can be set to **"REJECTED"**.
- If the bug is not high priority, or it is decided to get fixed in the next release, or the fixing of the bug depends on something, then the status **"DEFERRED"** is set.
- The above diagram and steps only explain the logic of SBLC. In practice, these robotic steps can cause communication issues sometimes. It is up to the developers to avail their time to

UWA Assessment 1

User Web Application (UWA) Case Study

I am introducing to you starting from this point a new application from a case study done by a website called Code Shack. The application was built to demonstrate using Python, Flask, and MySQL as web app tools. I selected it to demonstrate the processes involved with development and test projects. And part of the discoveries will be learning how SBLC fit in with SDLC and STLC. This is an acronym reminder.

The case study did not come with adequate documentation at first. However, I took time to produce some. The Login SRS and following test requirements cover the functional requirements of the Login User use case. The test requirements I defined will help ensure that the system functions as intended and will help to identify any defects in the software. You should find some defects yourself.

The main objective is for you to document defects and send a report to me. You can use a tool to get used to that exercise (manual or automated). It is my plan to be the developer who will try to fix the defects and provide you with a new release of the application for your review and defect closure. **See the [final section](#) to get the remaining instructions for the assessment exercise due after training module 6.**

The next sections of this document provide you with what you need for this assessment exercise. I refer to it with the number 1 because there is a separate assessment document labelled number 2. The difference between the two documents are as follows:

1. **UWA Assessment 1** provides some development and test project information that supports your test exercise. The application is ready for your implementation into your test environment.
2. **UWA Assessment 2** provides an application build approach that is more suited for an Agile environment. The instructions in this document build the UWA application by page. With this approach you gain internal code knowledge that supports white-box test design for your testing.

To achieve the main objective of this assessment, there are some setup requirements. It may seem as if it is more than you signed up for. But I want you to understand that the setup activities are some things to appreciate as a test analyst. I remember when I started testing applications after being a developer, I thought, "Yeah, no more programming." I was kidding myself. A test analyst often must be involved together with a team or alone to establish one or more test environments. That is necessary to conduct testing activities. Developers and test analysts both require an environment with database and project management tools. So, you will need to prepare a test environment on your computer to complete the test project assessments.

There is a separate document that will lead you into the setup for the test environment. The remainder of this document will give you the tools to conduct your test project for both assessment exercises. Please proceed with reading and when you are ready, peruse the setup document to get your computer ready for testing.

SRS for Login User Scenario

Use Case Name:	Login User Scenario
Summary:	To get personalized or restricted information, a user must click the login button on the application Login page .

UWA Assessment 1

Basic Flow:	<ol style="list-style-type: none">1. The use case starts when a user indicates that he/she wants to login.2. The system requests the username and password on the login page.3. The user enters his username and password.4. The system verifies the username and password against all registered users.5. The system starts a login session and displays a welcome message based on the user's preferences.
Alternative Flows:	<ul style="list-style-type: none">• Step 4: if username is invalid, the use case goes back to step 2.• Step 4: if the password is invalid the system requests that the user re-enter the password. When the user enters another password the use case continues with step 4 using the original username and new password.
Extension Points:	none
Preconditions:	The user is registered.
Postconditions:	The user can now obtain data and perform functions according to his registered access level or role.
Business Rules:	<ol style="list-style-type: none">1. Some data and functions are restricted to certain types of users or users with a particular access level.2. Application pages must show the Application Name prominently.

Test Requirements for Login User Scenario

Two important words to use here are: Verify and Validate. Verify ensures that the software is designed and developed according to the specified requirements. Verification finds bugs early in the development cycle. Validation ensures that the software meets the client's true needs and expectations. Validation finds bugs that verification can't catch.

It's possible for a product to pass verification but fail validation. For example, if a product is built according to the specifications but the specifications don't address the user's needs, it can pass verification but fail validation. Just because you met the requirement, if the requirement misinterprets the business need you failed validation. Thus, the reason for user acceptance testing or what some call end-to-end testing.

Test Requirement ID	Test Requirement Description
TR_LoginUser_001	Verify that the system requests the user's username and password when the user requests to login.
TR_LoginUser_002	Verify that the system checks the user's username and password against all registered users.
TR_LoginUser_003	Verify that the system starts a login session when the user enters valid credentials and displays a welcome message based on the user's preferences.
TR_LoginUser_004	Verify that the system requests the user to re-enter the password when an invalid password is entered.
TR_LoginUser_005	Verify that the system goes back to step 2 when an invalid username is entered.

UWA Assessment 1

Test Cases for the Login User Scenario

Test Case ID	Test Case Name	Test Case Steps	Expected Results
TC_LoginUser_1	Test Login Navigation	<ol style="list-style-type: none">1. Click UWA URL to display User Login.2. Verify that the system displays a form requesting the user's username and password.	/Login page displays App waits for user entry
TC_LoginUser_2	Test Valid User Login	<ol style="list-style-type: none">1. Enter a valid username and password.2. Click on the "Login" button.	No error messages seen Success message displays on home page
TC_LoginUser_3	Test Login Message Display	<ol style="list-style-type: none">1. Enter a valid username and password.2. Click on the "Login" button.3. Verify that the system starts a login session.4. Verify that the system displays a welcome message based on the user's preferences.	No error messages seen Success message displays on home page User's name displays Message displays with User's name
TC_LoginUser_4	Test Invalid Password Login	<ol style="list-style-type: none">1. Enter a valid username and an invalid password.2. Click on the "Login" button.3. Verify that the system displays a message requesting the user to re-enter the password.4. Enter a valid password.5. Verify that the system checks the user's username and password against all registered users and starts a login session.	No change Error message about password Message displayed Nothing happens yet User logged in
TC_LoginUser_5	Test Invalid User Login	<ol style="list-style-type: none">1. Enter an invalid username but valid password.2. Click on the "Login" button.3. Verify that the system displays a message indicating that the username is invalid.4. Enter a valid username.5. Enter a valid password.6. Verify that the system checks the user's username and password against all registered users and starts a login session.	Error message about username Message displays User logged in

UWA Assessment 1

SRS for the Register User Scenario

Use Case Name:	Register User Scenario
Summary:	To get personalized or restricted information, a new user must click the register button on the register page to add registry data.
Basic Flow:	<ol style="list-style-type: none"> 1. The use case starts when a user indicates that he wants to register. 2. The system requests a username, email address, and password on the Register page. 3. The user enters a username, email address, and password. 4. The system checks that the username does not duplicate any existing registered usernames. 5. The system checks all items are entered. All Items are required. 6. The user enters the information and clicks the Register button. 7. The system determines id and access level and stores all user information. 8. The system starts a login session and displays the home page.
Alternative Flows:	<ul style="list-style-type: none"> • Step 4: If the username duplicates an existing username the system displays a message, and the use case goes back to step 2. • Step 5: If the user does not enter a required field, a message is displayed, and the use case repeats step 4. The user can proceed with entry or click the LOGOUT button to quit.
Extension Points:	<i>Register Profile</i> <i>Register Edit Profile</i>
Preconditions:	none
Postconditions:	The user can now obtain data and perform functions according to his registered access level.
Business Rules:	<ul style="list-style-type: none"> • A registered user's location is the SBE location nearest his zip code. • Access levels are <ul style="list-style-type: none"> ○ ADMIN: A user can access all system data ○ MEMBER: The user can access limited system data <p>The default access level is Member.</p>

Register Profile

Use Case Name:	Register Profile and Edit Profile
Summary:	This use case allows a registered user to enter or change his preferences.
Basic Flow:	<ol style="list-style-type: none"> 1. The use case starts when a user indicates that he wanted to enter registration data. 2. The system accepted registration data is displayed on the Profile page 3. The system displays all current user data. 4. The user clicks the HOME button to display the home page. 5. The user can click the LOGOUT button to quit. 6. The user clicks the EDIT button to display the Edit Profile page and can make changes. 7. The system stores any change to registration data. 8. A message displays indicating the status.

UWA Assessment 1

Alternative Flows:	none
Extension Points:	none
Preconditions:	The user is already logged in.
Postconditions:	The system can customize a welcome message based on the user's revised data.
Business Rules:	User data displayed is based on user that is currently logged in.

Test Requirements for Register User Scenario

Test Requirement ID	Test Requirement Description
TR_RegisterUser_001	Verify that clicking the register button on the home page takes the user to the registration page.
TR_RegisterUser_002	Verify that the registration page requests a username and password.
TR_RegisterUser_003	Verify that the registration page requests the user's name, password, and email address.
TR_RegisterUser_004	Verify that the system checks that the username does not duplicate any existing registered usernames.
TR_RegisterUser_005	Verify that the system determines the user's id and access level based on internal data.
TR_RegisterUser_006	Verify that the system executes use case Register Edit Profile.
TR_RegisterUser_007	Verify that the system starts a login session and displays a welcome message based on the user's preferences.

Test Cases for the Register Scenario

Test Case ID	Test Case Name	Test Case Steps	Expected Results
TC_RegisterUser_1	Test Register Navigation	<ol style="list-style-type: none"> Click the Register button on home page Verify navigation Click the LOGOUT button. 	The registration page displays, and the system returns to the home page
TC_RegisterUser_2	Test Registration Prompts	<ol style="list-style-type: none"> Click the Register button on home page Verify messages Click the LOGOUT button. 	The Registration page requests a username and password
TC_RegisterUser_3	Test Registration Entry	<ol style="list-style-type: none"> Click the Register button on home page Enter new username and password Click the REGISTER button Enter email address Click the REGISTER button Click LOGOUT button 	<p>The registration page requests all required data Message prompts disappear Required field msg prompt New req field msg prompt</p> <p>Reg Profile page appears Msg indicates acceptance and proceeds to the Register Preference page</p>
TC_RegisterUser_3a	Test Registration Entry	<ol style="list-style-type: none"> Click the Register button on home page Enter new username and password Enter email address Click the REGISTER button Click LOGOUT button 	<p>Registration page displays</p> <p>Message prompts disappear</p> <p>Reg Pref page appears Returns to home page</p>

UWA Assessment 1

TC_RegisterUser_4	Test Reg Duplicate Name	1. Click the Register button on home page 2. Enter existing username and password 3. Enter email address 4. Click the REGISTER button 5. Click CANCEL button	Registration page displays Message prompts disappear Duplicate username error Returns to home page
		1.	

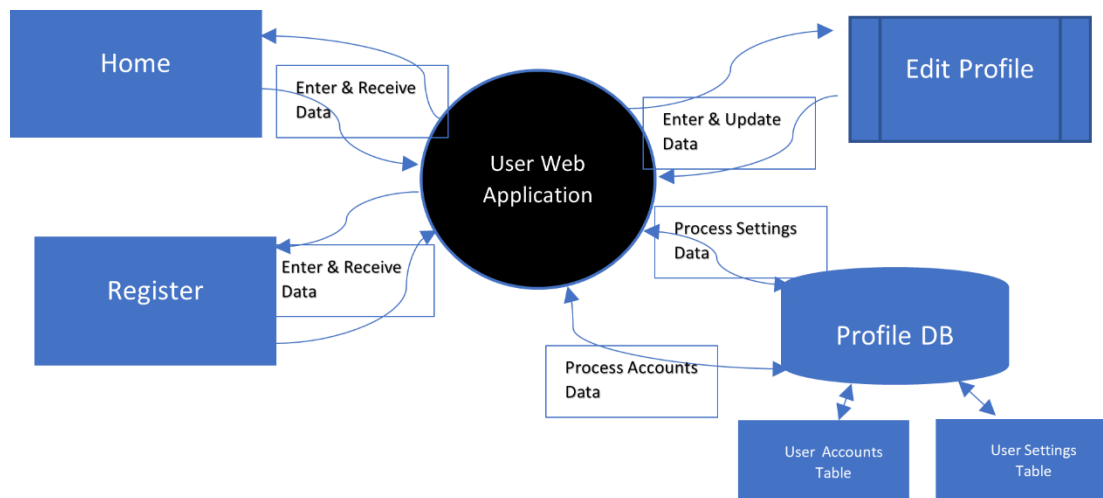
[Top of the Document](#)

[General Test Requirements](#)

Test Requirement ID	Test Requirement Description
TR_General_001	Check page flow for gaps or access without requiring login.
TR_General_002	Make sure all buttons are working as expected.
TR_General_003	Check images and titles for appropriate ALT displays.
TR_General_004	Verify no unexpected or unnecessary text exists on any page.
TR_General_005	Are any entry text boxes too small or text too big?
TR_General_006	Any text or page colors that are visual issues?
TR_General_007	Any visual issues of any kind?

User Web Application Development

UWA Development



UWA Assessment 1

System Design Components

1. UWA Pages
 - a. Home Page
 - i. User Name
 - ii. User Password
 - b. Register Page
 - i. User ID
 - ii. User Password
 - iii. Username
 - iv. Email address
 - c. Edit Profile Page
 - i. User ID
 - ii. User Password
 - iii. Username
 - iv. Email address
2. Profile Database
 - a. User Accounts Table
 - i. All fields listed on Register page
 - ii. Record Number
 - iii. Access Level
 - b. User Settings Table
 - i. id
 - ii. setting key
 - iii. setting value
 - iv. category
3. UWA Modules (Executable)
 - a. Main
 - b. HTML templates
 - c. CSS

Program Design

Main Module

The home module will display a login page to the user. The user will enter their User ID and User Password, and the system will validate the input. If the input is correct, the system will redirect the user to the Request page. If the input is incorrect, the system will display an error message.

Functions:

- Home(): Displays the home page with login and register buttons.
- Login(): Prompts the user to enter their User ID and password.
- Register(): Processes data entry for user accounts.
- Edit Profile(): Displays user data for update.

UWA Assessment 1

Use Case: Login User Scenario

1. The use case starts when a user indicates that they want to login by clicking the login button on the home page.
2. The system calls the Login() function, which displays the home page with a login form.
3. The user enters their User ID and password and other user data in the login form and clicks the register button.
4. The system calls the Edit Profile() function, which prompts the user to update user data.
5. The user enters their User ID and password and clicks the login button from the login page.
6. The system calls the verify_user_credentials(user_id, password) function, which verifies the entered User ID and password against the registered users in the User Accounts table.
7. If the User ID and password are valid, the system calls the display_welcome_message function, which displays a personalized welcome message.
8. If the User ID is invalid, the system displays an error message and prompts the user to re-enter their User ID.
9. If the password is invalid, the system displays an error message and prompts the user to re-enter their password.
10. The use case ends when the user is successfully logged in and can now access data and perform functions according to their registered access level.

Database Design:

Profile Database

Tables:

- User Accounts Table: stores user registration information including User ID, User Password, Username, Email address, Record Number, and Access Level.
- User Settings Table: stores system settings including system keys & values along with categories.

Fields:

- User ID: a unique identifier for each registered user.
- User Password: the password associated with the user's User ID.
- Username: the username associated with the user's User ID.
- Email address: the user's email address.
- Record Number: a unique identifier for each record in the tables.
- Access Level: the user's access level determines what data and functions they can access.
- System Setting keys: a unique identifier for each setting.
- System Setting value: a description of the setting key/value pair.
- System Setting category: grouping value.

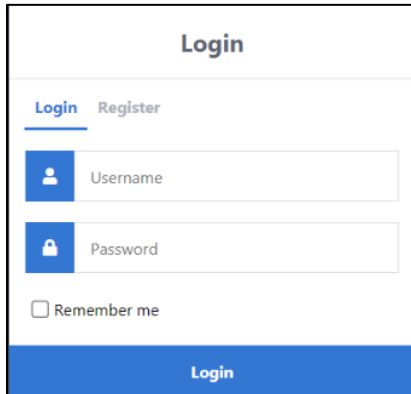
Application Pages

This section serves as a revision to an earlier section about application pages. It may also replace that section.

At present, the Login page is also the home page. Any reference to either is the same page. It can be accessed with the following URL: <http://127.0.0.1:5000/pythonlogin/>.

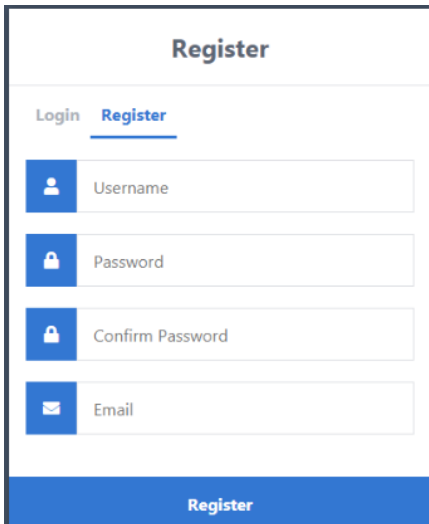
UWA Assessment 1

The page supports login entry or registration request.



The Login form is titled "Login" at the top. Below the title, there are two tabs: "Login" (which is selected and underlined) and "Register". The form contains three input fields: a "Username" field with a person icon, a "Password" field with a lock icon, and a "Remember me" checkbox. At the bottom of the form is a blue button labeled "Login".

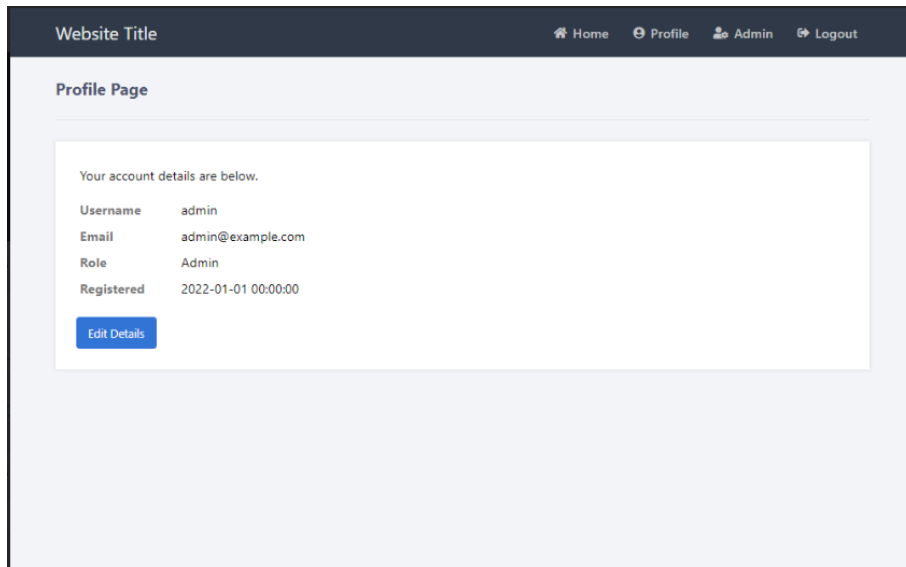
The Register page below supports account register data entry. The top two buttons can toggle between the Register or the Login page. The bottom button is used to request DB insert processing of the account register data. If the data is inserted successfully, the Profile page should display with an appropriate message.



The Register form is titled "Register" at the top. Below the title, there are two tabs: "Login" and "Register" (which is selected and underlined). The form contains four input fields: a "Username" field with a person icon, a "Password" field with a lock icon, a "Confirm Password" field with a lock icon, and an "Email" field with an envelope icon. At the bottom of the form is a blue button labeled "Register".

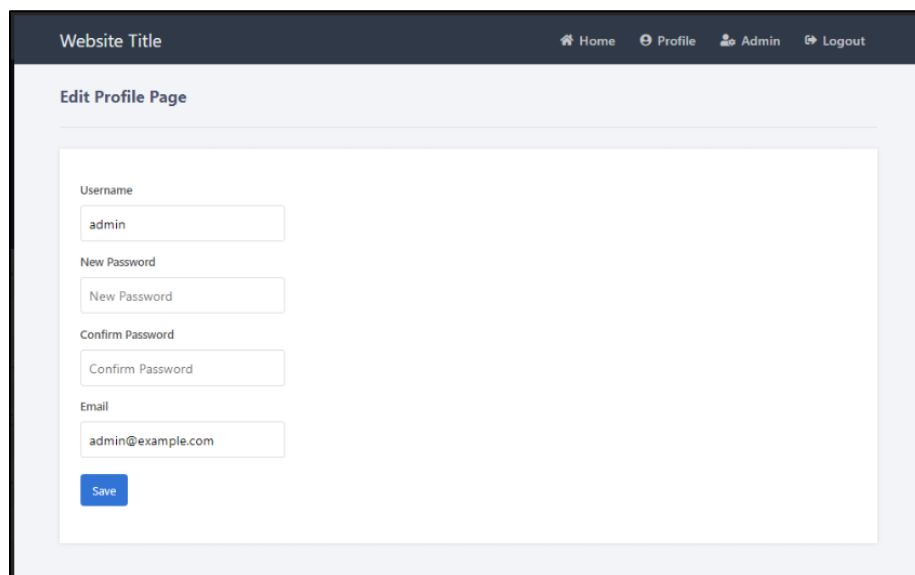
The next page in the flow is the Profile page which changes in customized view. Common buttons display at the top right location. A process button displays below the user account data. A user can choose to press any button to change the application flow. If the user does not desire to update the account data, they can choose another page.

UWA Assessment 1



The screenshot shows a web application interface. At the top, there is a dark blue header with the text "Website Title" on the left and navigation links "Home", "Profile", "Admin", and "Logout" on the right. Below the header, the main content area has a light blue background. A section titled "Profile Page" is displayed. Inside this section, there is a white box containing the text "Your account details are below." followed by a list of details: Username: admin, Email: admin@example.com, Role: Admin, and Registered: 2022-01-01 00:00:00. At the bottom of this white box is a blue button labeled "Edit Details".

The Edit Profile page is like the Register page, but it is intended for updating the account data if required. Any changes or corrections are processed from this page. The SAVE button is used to request account data updates. A message should display indicating the status of the request.



The screenshot shows the "Edit Profile Page" of the web application. It features the same dark blue header with "Website Title" and navigation links. The main content area has a light blue background. A section titled "Edit Profile Page" is displayed. Inside this section, there is a white box containing four input fields: "Username" (with "admin" entered), "New Password", "Confirm Password", and "Email" (with "admin@example.com" entered). At the bottom of this white box is a blue button labeled "Save".

From this page, a user can navigate to the other pages of the application.

UWA Assessment 1

Grey Box Testing Insight

It is possible to see messages or processing in HTML, CSS, and JavaScript components used by the application. That information might be useful to adding more test requirements to cover or directly creating test cases to confirm application processing is stable and meeting user requirements.

Component	Message	Process
Code	Not logged in	Code check for logged in condition
Code	Account doesn't exist with that email or the activation code is incorrect!	Code account activate function
Code	Invalid email address!	Code validation check
Code	Username must contain only characters and numbers!	Code validation check
Code	Please fill out the form!	Code validation check
Code	Username already exists!	Code validation check
Code	Username must be between 5 and 20 characters long!	Code validation check
Code	Password must be between 5 and 20 characters long!	Code validation check
Code	Passwords do not match!	Code validation check
Code	Account Activation Required	Gets sent to email
Code	You have changed your email address! You need to re-activate your account! You will be automatically logged-out.	Gets sent to email
Code	Account updated successfully!	Code validation check
Code	Reset password link has been sent to your email!	Code validation check
Code	An account with that email does not exist!	Code validation check
Code	Your password has been reset, you can now	Code validation check
Code	Passwords must match and must not be empty!	Code validation check
Code	Invalid email and/or code!	Code validation check
Code	Incorrect code provided!	Code validation check
Code	No email and/or ID provided!	Code validation check
Code	Account created successfully!	Code validation check
Code	Account updated successfully!	Code validation check
Code	Account deleted successfully!	Code validation check
Code	Settings updated successfully!	Code validation check
Code	Email templates updated successfully!	Code validation check
Code	Please check your email to activate your account!	Code validation check
Code	You have registered! You can now login!	Code validation check

Assessment Instructions (Part 1)

1. You will need to set up a test environment using one of the IDE's that works for you. I like PyCharm and MS VS Code. They are free and solid. There are many others that work fine.
2. Make sure you download the application and load it in your test environment.
3. You may need to listen again to one of the videos I created that is a tutorial for the UWA application setup.
4. This will be the software for you to execute and test. As you find defects, document the defects.

UWA Assessment 1

[Assessment Instructions \(Part 2\)](#)

1. Once you have completed your first round of testing, send the bug reporting by email to me. I will proceed with applying fixes and create a new release for you to load for retest and verify activities.
2. You will receive the new release by the time we complete the Test Automation module training.
3. The Test Automation module assessment exercise includes the following:
 - a. Load the new release of UWA into your test environment
 - b. Retest using your test cases
 - c. Document any new bugs and unresolved bugs
 - d. Closeout resolved bugs
4. Then, draft a closure report when you are done and send it to me.
5. I want to be sure you get a good understanding and experience with planning to test, testing and defect management.