Foundation Project: Test Case Development

Use Case ID: 1

Description: Users should be able to open a new User account with the Planetarium.

Actors: New User

1.1 First Test Data

Boundary Analysis Testing for username/password length
Requirement: usernames and passwords should not be longer than 30 characters

0 Character Username	30 Character Username	31 Character Username
	HelloMyNameIsRyanAndThisIsTest	HelloMyNameIsRyanAndThisIsATest

0 Character Password	30 Character Password	31 Character Password
	MarvelCinematicUniverseFan4Eva	MarvelCinematicUniverseFan4Eva!

Data Table

Username	Password	Account Creation Result	Redirect
(0 Characters)	(0 Characters)	User Created	User redirected to the login page
(0 Characters)	MarvelCinematicUniverseFan4Eva	User Created	User redirected to the login page
(0 Characters)	MarvelCinematicUniverseFan4Eva!	User Not Created	User remains on the creation page
HelloMyNameIsRyanAndThisIsTest	(0 Characters)	User Created	User redirected to the login page
HelloMyNameIsRyanAndThisIsTest	MarvelCinematicUniverseFan4Eva	User Created	User redirected to the login page
HelloMyNameIsRyanAndThisIsTest	MarvelCinematicUniverseFan4Eva!	User Not Created	User remains on the creation page
HelloMyNameIsRyanAndThisIsATest	(0 Characters)	User Not Created	User remains on the creation page
HelloMyNameIsRyanAndThisIsATest	MarvelCinematicUniverseFan4Eva	User Not Created	User remains on the creation page
HelloMyNameIsRyanAndThisIsATest	MarvelCinematicUniverseFan4Eva!	User Not Created	User remains on the creation page

1.2 Second Test Data

Equivalence Partition Testing for the uniqueness of usernames
Requirement: users should have unique usernames

Unique Username	Non-Unique Username
Deadpool	Batman

Data Table

Username	Password	Account Creation Result	Redirect
Deadpool	MarvelCinematicUniverseFan4Eva	User Created	User redirected to the login page
Batman	MarvelCinematicUniverseFan4Eva	User Not Created	User remains on the creation page

1.3 Third Test Data

Exploratory Testing to see if passwords are in plain text in any places we have access to in the provided software.
Error Guess Testing to see if the passwords are in plain text in the interfaces for the application a regular user would have access to.
Requirement: passwords should never be visible in plaintext

- The password that should be obfuscated: “I am the night”
    - Use for both Exploratory and Error Guess Testing

Data Table

Process/Location	Password Action	Password Obfuscated Result
Create Account	Type new password	Characters masked in the text box
Successful Account Creation	Password is confirmed in the success alert	Characters masked in the alert
Database	New password added to the user database	Characters masked within the database

1.4 Fourth Test Data

Error Guess Testing to perform Acceptance Testing for the Use Case

- Use positive and negative test data found from the First Test Data and Second Test Data depending on the Acceptance Tests being performed.

Use Case ID: 2

Description: Users should be able to securely access their account.

Actors: Existing User

2.1 First Test Data

Equivalence Partition Testing for the uniqueness of usernames
Requirement: Users should have unique usernames
(Note: users should only be able to access their account based on their username)

Data Table

Username	Password	Login Result	Redirect
Deadpool	I am the night	User not logged in	User remains on log in page
Batman	MarvelCinematicUniverseFan4Eva	User not logged in	User remains on home page

2.2 Second Test Data

Exploratory Testing to see if passwords are in plain text in any places we have access to in the provided software.
Error Guess Testing to see if the passwords are in plain text in the interfaces for the application a regular user would have access to.
Requirement: Users should have unique usernames

Data Table

Process/Location	Password Action	Password Obfuscated Result
Log In	Type current password	Characters masked in the text box

Database	Current password is checked in database	Characters masked within the database

2.3 Third Test Data

Error Guess Testing to perform Acceptance Testing for the Use Case

- Use positive and negative test data found from the First Test Data and Second Test Data depending on the Acceptance Tests being performed.

Use Case ID: 3

Description: Users should be able to see planets and moons added to the Planetarium

Actors: Existing User

3.1 First Test Data

Equivalence Partition Testing for user account validity
Requirement: Only logged in Users should be able to access the Planetarium home page

Data Table

Logged In User	Planetarium Home Access
User is logged in	User is allowed access to the home page
User is not logged in	User is not allowed access to the home page

3.2 Second Test Data

Error Guess Testing to see if planets and moons within the Planetarium are successfully displayed to the logged-in user.

Requirement: Only logged in Users should be able to access the Planetarium home page

Data Table

Username for Account	Planets and Moons on Home Page
Batman	All planets and moons should be displayed
Deadpool	All planets and moons should be displayed

3.3 Third Test Data

Error Guess Testing to perform Acceptance Testing for the Use Case

- Use positive and negative test data found from the First Test Data and Second Test Data depending on the Acceptance Tests being performed.

Use Case ID: 4

Description: Users should be able to add new Planets to the Planetarium

Actors: Existing User

4.1 First Test Data

Boundary Analysis Testing for Planet name length
Requirement: Planet and Moon names should not have more than 30 characters

0 Character Planet Name	30 Character Planet Name	31 Character Planet Name
	ThisIsAReallyLongPlanetNameLol	ThisIsWayTooLongForAPlanetName!

Data Table

Planet Name	Planet Creation Result	Refresh
(0 Characters)	Planet Created	User home page refreshed
ThisIsAReallyLongPlanetNameLol	Planet Created	User home page refreshed
ThisIsWayTooLongForAPlanetName	Planet Not Created	User home page not refreshed

4.2 Second Test Data

Equivalence Partition Testing for the uniqueness of Planets
Requirement: Planets and moons should have unique names

Unique Planet Name	Non-Unique Planet Name
Nessus	Earth

Data Table

Planet Name	Planet Creation Result	Refresh
Nessus	Planet Created	User home page refreshed
Earth	Planet Not Created	User home page not refreshed

4.3 Third Test Data

Error Guess Testing to see if when a new planet is added, it is assigned to the respective user.
Requirement: Planets should be “owned” by the user that added it to the Planetarium

Data Table

Current User Id	Planet Owner Id	Planet Creation Result
1	1	Planet Created
2	2	Planet Created

4.4 Fourth Test Data

Error Guess Testing to see if planets can be successfully added to the Planetarium with or without an associated image
Requirement: Planets and Moons should allow adding an associated image, but an image should not be required for the data to be added to the database.

Data Table

Planet Image(Y/N)	Planet Creation Result	Refresh
Yes	Planet Created	User home page refreshed
No	Planet Created	User home page refreshed

4.5 Fifth Test Data

Error Guess Testing to perform Acceptance Testing for the Use Case

- Use positive and negative test data found from the First Test Data and Second Test Data depending on the Acceptance Tests being performed.

Use Case ID: 5

Description: Users should be able to remove Planets from the Planetarium

Actors: Existing User

5.1 First Test Case

Equivalence Partition testing for the unique planet name entered for deletion
Requirement: Planets and moons should have unique names

Unique Planet Name	Non-Unique Planet Name
Nessus	Earth

Data Table

Planet Name	Planet Deletion Result	Refresh
Nessus	Planet Not Deleted	User home page not refreshed
Earth	Planet Deleted	User home page refreshed

5.2 Second Test Case

Exploratory Testing to see if the user account Id matches the Owner Id of the planet within the database
Equivalence Partition Testing to see if a user who is the Owner of a planet can remove it and a user who is not the Owner cannot remove it
Requirements:

- Users should only be able to interact with resources they have added to the Planetarium
- Planets should be “owned” by the user that added it to the Planetarium

Data Table

Current User Id	Planet Owner Id	Planet Deletion Result	Refresh
1	1	Planet Deleted	User home page refreshed
2	1	Planet Not Deleted	User home page not refreshed

5.3 Third Test Case

Error Guess Testing to perform Acceptance Testing for the Use Case

- Use positive and negative test data found from the First Test Data and Second Test Data depending on the Acceptance Tests being performed.

Use Case ID: 6

Description: Users should be able to add new Moons to the Planetarium associated with a Planet

Actors: Existing User

6.1 First Test Data

Boundary Analysis Testing for Moon name length
Requirement: Planet and Moon names should not have more than 30 characters

0 Character Moon Name	30 Character Moon Name	31 Character Moon Name
	ThisIsNotAsLongOfAMoonNameHere	ThisIsAMuchLongerNameForAMoonHa

Data Table

Planet Name	Planet Creation Result	Refresh
(0 Characters)	Planet Created	User home page refreshed
ThisIsNotAsLongOfAMoon NameHere	Planet Created	User home page refreshed
ThisIsAMuchLongerName ForAMoonHa	Planet Not Created	User home page not refreshed

6.2 Second Test Data

Equivalence Partition Testing for the uniqueness of Moons
Requirement: Planets and moons should have unique names

Unique Moon Name	Non-Unique Moon Name
Pluto	Luna

Data Table

Moon Name	Moon Creation Result	Refresh
Pluto	Moon Created	User home page refreshed
Luna	Moon Not Created	User home page not refreshed

6.3 Third Test Data

Error Guess Testing to see if when a new moon is added, it is assigned to the respective user.
Requirement: Moons should be “owned” by the Planet and the user adding the moon associated with it

Data Table

Current User Id	Moon Owner Id	Moon Creation Result
1	1	Moon Created
2	2	Moon Created

6.4 Fourth Test Data

Error Guess Testing to see if moons can be successfully added to the Planetarium with or without an associated image
Requirement: Planets and Moons should allow adding an associated image, but an image should not be required for the data to be added to the database.

Data Table

Moon Image(Y/N)	Moon Creation Result	Refresh
Yes	Moon Created	User home page refreshed
No	Moon Created	User home page refreshed

6.5 Fifth Test Data

Error Guess Testing to perform Acceptance Testing for the Use Case

- Use positive and negative test data found from the First Test Data and Second Test Data depending on the Acceptance Tests being performed.

Use Case ID: 7

Description: Users should be able to remove Moons from the Planetarium

Actors: Existing User

7.1 First Test Case

Equivalence Partition testing for the unique moon name entered for deletion
Requirement: Planets and moons should have unique names

Unique Moon Name	Non-Unique Planet Name
Pluto	Luna

Data Table

Moon Name	Moon Deletion Result	Refresh
Pluto	Moon Not Deleted	User home page not refreshed
Luna	Moon Deleted	User home page refreshed

7.2 Second Test Case

Exploratory Testing to see if the user account Id matches the Owner Id of the moon within the database
Equivalence Partition Testing to see if a user who is the Owner of a moon can remove it and a user who is not the Owner cannot remove it
Requirements:

- Users should only be able to interact with resources they have added to the Planetarium
- Moons should be “owned” by the user that added it to the Planetarium

Data Table

Current User Id	Moon Owner Id	Moon Deletion Result	Refresh
1	1	Moon Deleted	User home page refreshed
2	1	Moon Not Deleted	User home page not refreshed

7.3 Third Test Case

Error Guess Testing to perform Acceptance Testing for the Use Case

- Use positive and negative test data found from the First Test Data and Second Test Data depending on the Acceptance Tests being performed.