Final Project – Screenspace

SENG 6245
Ryan Cowan

What is Screenspace?

Screenspace is a java-based web application for managing and displaying digital signage content on various devices. The application must be connected to a network and uses a Raspberry Pi to display the content on the device. It will allow the uploading, editing of meta data associated with the sign, deleting, and displaying of content.

After selecting it I found that it did not have validation of inputs. After making several attempts to find an alternative I simply forked the repo and added data validation for equivalence class partitioning and boundary value analysis.

https://github.com/cowanr/screen-space

Key Stakeholders and End-Users

Digital signage's purpose is to deliver dynamic content to inform, or educate, the intended audience. Compared to traditional print signage, it uses technology such as monitors, kiosks, and projectors to deliver the content. Digital signage can be used in any situation that has a need to keep a large audience up to date such as public safety information, internal company information, sale promotions, and map information.

Key stakeholders will be the project manager, business analyst, software engineers, and liaison from the university supporting the project.

Potential end users include everything from universities and event centers to public spaces and restaurants. We will focus on universities.

Requirements

I identified key areas of support that Screenspace must support.

- Support multiple users
- Support live preview
- Intuitive create, update, and delete interface for content management system.

Use Cases

- As a user I can sign up for an account.
- As a user I can login and see my content.
- As a user I can upload an image and assign to a display.
- As a user I can see a preview of the signage.
- As a user I can reset dashboard in one click.

Domain Partitioning

Sign Up Page

The Sign Up screen allows a user to sign up for an account. A username, password, email, and phone number are required to create an account.

Username

Requirement: Must be an 8-20 character alphanumeric string

Equivalence Class Partitioning

Invalid: Any alphanumeric string less than 8 characters.
Valid: Any alphanumeric string between 8 and 20 characters.
Invalid: Any alphanumeric string greater than 20 characters.

Boundary Value Analysis

Test 1: Validate Username by entering in a 7 character alphanumeric string: [MIN-1] Invalid Boundary Check
Test 2: Validate Username by entering in a 8 character alphanumeric string: [MIN] Valid Boundary Check
Test 3: Validate Username by entering in a 9 character alphanumeric string: [MIN+1] Valid Boundary Check
Test 4: Validate Username by entering in a 19 character alphanumeric string: [MAX-1] Valid Boundary Check
Test 5: Validate Username by entering in a 20 character alphanumeric string: [MAX] Valid Boundary Check
Test 6: Validate Username by entering in a 21 character alphanumeric string: [MAX+1] Invalid Boundary Check

Email

Requirement: Must be a valid email address.

Equivalence Class Partitioning

Valid: A valid email address.
Invalid: An invalid email address.

Boundary Value Analysis

Test 1: Validate Email by an valid email.
Test 2: Validate Email by entering an invalid email.

Password

Requirement: Must be an 8-20 character alphanumeric string

Equivalence Class Partitioning

Invalid: Any string less than 8 characters.
Valid: Any alphanumeric string between 8 and 20 characters.
Invalid: Any string greater than 20 characters.

Boundary Value Analysis

Test 1: Validate Password by entering in a 7 character alphanumeric string: [MIN-1] Invalid Boundary Check
Test 2: Validate Password by entering in a 8 character alphanumeric string: [MIN] Valid Boundary Check
Test 3: Validate Password by entering in a 9 character alphanumeric string: [MIN+1] Valid Boundary Check
Test 4: Validate Password by entering in a 19 character alphanumeric string: [MAX-1] Valid Boundary Check
Test 5: Validate Password by entering in a 20 character alphanumeric string: [MAX] Valid Boundary Check
Test 6: Validate Password by entering in a 21 character alphanumeric string: [MAX+1] Invalid Boundary Check

Phone Number

Requirement: Must be a 10 digit number.

Equivalence Class Partitioning

Invalid: Number between 0 and 9 digits.
Valid: A 10 digit number.
Invalid: A number with 11 or more digits.

Boundary Value Analysis

Test 1: Validate Phone Number by entering in a 9 digit number: [MIN-1] Invalid Boundary Check
Test 2: Validate Phone Number by entering in a 10 digit number: [MIN/Max] Valid Boundary Check
Test 3: Validate Phone Number by entering in a 11 digit number: [Max+1] Invalid Boundary Check

Log In Page

The Sign In screen allows a user to sign in. A valid username and password combination is required to sign in.

Equivalence Class Partitioning

Valid: A valid username and password combination for an account.
Invalid: An invalid username and/or password combination for an account.

Boundary Value Analysis

Test 1: Validate Email by an valid email.
Test 2: Validate Email by entering an invalid email.

Dashboard Page

This is where the user can upload the image for the signage. Only BMP, GIF, JPG, and PNG files are allowed.

Upload Input

Equivalence Class Partitioning

Valid: A valid image file.
Invalid: An invalid image file.

Boundary Value Analysis

Test 1: Validate upload input by selecting an valid image file and uploading.
Test 2: Validate upload input by selecting an invalid image file and uploading.

User Profile Page

The User Profile opens a modal and displays the user account information that was entered during signup. Equivalence class partitioning and boundary value analysis is not possible since there no inputs.

ScreenPlay Page

The ScreenPlay is a separate page that previews the digital signage as it would be shown on the devices. Equivalence class partitioning and boundary value analysis is not possible since there no inputs.

Reset Page

Reset opens a modal with the option to reset or close(cancel) the modal. Clicking reset deletes all uploaded images. Equivalence class partitioning and boundary value analysis is not possible since there is only one possible value for each input.

Purchase Page

Purchase opens a modal with the option to buy or close(cancel) the modal. Clicking the buy button would normally open a store page where the item could be purchased but this wasn’t fully implemented. Equivalence class partitioning and boundary value analysis is not possible since there is only one possible value for each input.

Test Scenario Generation

1 - User Account Testing

Testing the account login component of the system by selecting the “Login” button and attempting to access the user account dashboard with valid and invalid credentials.

2 – Sign Up Testing

Testing the sign-up component of the system by selecting the “Signup” button and attempting to create an account by providing a username, password, email, and phone number.

3 – Image Upload Testing

Testing the image upload component of the system by selecting valid and invalid files to upload.

4 – Preview Testing

Testing the preview component of the system by uploading one or more images and previewing the digital signage.

5 - Reset Testing

Testing the reset component of the system by uploading 0 or more images and resetting the signage to erase all uploads.

6 - Logout Testing

Testing the logout component of the system by logging out of the system and trying to access different pages of the system.

Test Adequacy Evaluation

In attempting to reach 50% code coverage of the SUT it became apparent that it was not possible to write unit tests due to the tight coupling of the application service layers as it is written. I was able to write unit tests for the user and image models with close to 100% coverage. However, once I started to try to write tests for the servlets, I could not find a way to refactor the use of data access objects so that the unit test would not need to access the database. In researching this issue, I believe that a business layer should have been used to provide the necessary business objects. This would have allowed me to mock the controllers and DAO objects in my unit tests. I also ran into issues with how to write unit tests for the Hibernate data access objects.

In writing the unit tests for the image model I did uncover an issue with the image model getBase64Image getter function. When executed this function would also set the base64 encoded property. This made the setBase64Image function superfluous.

In conducting the testing per the testing scenarios, I found that signing up with an account that already existed resulted in unexpected behavior. A user trying to signup with a username that is already taken is logged in with the account if the password is valid or is redirected to the login page with an error if the password was not correct.

Project Screenspace
Created By Ryan Cowan

Test Scenario		Test Case									
ID	Description	ID	Description	Preconditons	Test Data	Post Conditions	Expected Result	Actual Results	Exp By	Exp Date	Status
1	User Account Testing	1	Enter valid username and valid password	Account exists and be on login page	username : TestUser01 , password : testPassword12345	User is on dashboard	User provides username and password and is authenticated.	User authenticated and redirected to dashboard	Ryan Cowan	11/28/2021	PASS
1	User Account Testing	2	Enter valid username and invalid password	Account exists and be on login page	username : TestUser01 , password : pwdTest12345	User is shown invalid username or password error.	User is on login page and shown invalid username or password error.	User authentication failed and error message was shown	Ryan Cowan	11/28/2021	PASS
1	User Account Testing	3	Enter invalid username and valid password	Account exists and be on login page	username : TestUser02 , password : testPassword12345	User is shown invalid username or password error.	User is on login page and shown invalid username or password error.	User authentication failed and error message was shown	Ryan Cowan	11/28/2021	PASS
2	Sign Up Testing	4	Sign up with valid username, password, email, and phone number.	Be on the sign up page and user Test1234 doesn't exist	username : Test1234 , password : Password , email : test@email.com , phone : 1234567890	User is redirected to the ScreenPlay dashboard	User is able to create an account	User is able to create an account	Ryan Cowan	11/28/2021	PASS
2	Sign Up Testing	5	Sign up with valid username that already exists.	Be on the sign up page and user TestUser05 does exist	username : TestUser05 , password : testPassword12345 , email : test@email.com , phone : 1234567890	User is redirected to the Signup page with error message to choose another username.	User is not able to create an account and is shown an error message.	User is logged in with the account if username and password is valid or is redirected to login page with error for invalid username and password if the account already exists.	Ryan Cowan	11/28/2021	FAIL
2	Sign Up Testing	6	Sign up with invalid username, valid password,valid email, and valid phone number.	Be on the sign up page	username : Test123 , password : testPassword12345 , email : test@email.com , phone : 1234567890	User is redirected back to the sign up page	User is redirected back to the sign up page with an error saying that the username is not valid	User is redirected back to the sign up page with an error saying that the username is not valid	Ryan Cowan	11/28/2021	PASS
2	Sign Up Testing	7	Sign up with valid username, invalid password,valid email, and valid phone number.	Be on the sign up page	username : TestUser05 , password : test , email : test@email.com , phone : 1234567890	User is redirected back to the sign up page	User is redirected back to the sign up page with an error saying that the password is not valid	User is redirected back to the sign up page with an error saying that the password is not valid	Ryan Cowan	11/28/2021	PASS

2	Sign Up Testing	8	Sign up with valid username, valid password, invalid email, and valid phone number.	Be on the sign up page	username : TestUser05 , password : testPassword12345 , email : @email.com , phone : 1234567890	User is redirected back to the sign up page	User is redirected back to the sign up page with an error saying that the email is not valid	User is redirected back to the sign up page with an error saying that the email is not valid	Ryan Cowan	11/28/2021	PASS
2	Sign Up Testing	9	Sign up with valid username, valid password,valid email, and invalid phone number.	Be on the sign up page	username : TestUser05 , password : testPassword12345 , email : test@email.com , phone : 12345678910	User is redirected back to the sign up page	User is redirected back to the sign up page with an error saying that the phone number is not valid	User is redirected back to the sign up page with an error saying that the phone number is not valid	Ryan Cowan	11/28/2021	PASS
3	Image Upload Testing	10	User can upload a valid image file	Be logged in and on the dashboard	username : TestUser01 , password : testPassword12345 , upload any BMP, GIF, JPG, or PNG file	User is redirected back to the dashboard where the upload image is show in the library grid	User is able to upload the image	User is able to upload the image	Ryan Cowan	11/28/2021	PASS
3	Image Upload Testing	11	User cannot upload a pdf file	Be logged in and on the dashboard	username : TestUser01 , password : testPassword12345 , upload a PDF, TXT, or file that is not an image file.	User is redirected back to the dashboard	User is shown an error message that the file was invalid	User is shown an error message that the file was invalid	Ryan Cowan	11/28/2021	PASS
4	Preview Testing	12	User can preview the signage they have uploaded.	Be logged in with one or more images uploaded.	username : TestUser01 , password : testPassword12345	User is taken to the preview page where the images are shown in a slider	User is show the images in a slider they can control.	User is show the images in a slider they can control.	Ryan Cowan	11/28/2021	PASS
5	Reset Testing	13	A user can reset (delete) all uploaded images.	Be logged in with one or more images uploaded.	username : TestUser01 , password : testPassword12345	User is redirected back to the dashboard	User is redirected back to the dashboard with all images removed.	User is redirected back to the dashboard with all images removed.	Ryan Cowan	11/28/2021	PASS
5	Reset Testing	14	A user can cancel the reset process.	Be logged in with one or more images uploaded.	username : TestUser01 , password : testPassword12345	User is redirect back to the dashboard	User is redirected back to the dashboard with all images are still present.	User is redirected back to the dashboard with all images are still present.	Ryan Cowan	11/28/2021	PASS
6	Logout Testing	15	A user can logout of the system.	Be logged in.	username : TestUser01 , password : testPassword12345	User is redirected to the login screen	User is redirected to the login screen	User is redirected to the login screen	Ryan Cowan	11/28/2021	PASS
6	Logout Testing	16	A logged out user cannot access the system pages.	Login then log out	username : TestUser01 , password : testPassword12345 , url : http://localhost:8090/Screenspace/previewController	An anyonmous user is redirect back to the login page	User is redirect to login page	User is redirect to login page	Ryan Cowan	11/28/2021	PASS

6	Logout Testing	17	A logged out user cannot access the system pages.	Login then log out	username : TestUser01 , password : testPassword12345 , url : http://localhost:8090/Screenspace/wdController	An anyonmous user is redirect back to the login page	User is redirect to login page	User is redirect to login page	Ryan Cowan	11/28/2021	PASS