

LOGIN SCENE TESTING:
Ekaterina Kozlovsky

EECS 2311 Section X
03/23/2024

Take Home Assignment

User Stories	Main Developers	Distribution for Take Home Assignment
Login Includes: <ul style="list-style-type: none">• Login scene• User class	Isaiah	Ekaterina
Dog Profile Includes: <ul style="list-style-type: none">• DogProfileScene• Dog class	Ekaterina	Yuqian
Calendar/Appointment Includes: <ul style="list-style-type: none">• Calendar scene• BookedAppointments scene• Appointments class• AppointmentsManager class	Yuqian Edson	Zainab
Wallet Includes: <ul style="list-style-type: none">• Wallet portion in UserProfileScene• DonationScene• Wallet class• RecurringPayments class	Zainab	Isaiah
User Preferences/Settings Includes: <ul style="list-style-type: none">• UserProfileScene• Attributes class + children• Tags class	Ekaterina Isaiah	Edson
Poster Profiles/Liked Dogs Includes: <ul style="list-style-type: none">• PosterProfileScene• LikedDogsScene• Poster class	Ekaterina Zainab	Connor

Type	User Story	Main Developer - ITR3	Testing Assigned To
Login	User Login	Isaiah	Ekaterina
Dog Profile	Liking Functionality	Ekaterina	Yuqian
	Pass Functionality	Isaiah	
	Adopted Logic*	Edson	
	Dog Expanded Info	Ekaterina	
Calendar/Appointment	Book Appointment	Edson	Zainab
	Cancel Appointment	Edson	
	Reschedule Appointment	Yuqian	
	Show User Booked Appointment	Edson,Yuqian	
	Block inactive cells, show user's current appointment, show busy days for dog	Ekaterina	
	Show the calendar	Yuqian	
Wallet 1/Wallet 2	Recurring Payment	Zainab	Isaiah
	Add Money to Wallet	Zainab	
	Validate User Funds	Zainab	
	One Time Payment	Zainab	
	Sponsorship - view sponsored dogs	Ekaterina	
	Editing sponsorships	Zainab	
	Cancelling sponsorship	Zainab	
	Support multiple sponsorships	Ekaterina	
	Adding money to wallet	Zainab	
	Dog Matching Algorithm	Isaiah	
User Preferences/Settings	Tags	Ekaterina	Edson
	Attributes	Ekaterina	
	Attributes and Tag filtering	Isaiah	
Poster Profiles/Liked Dogs	Liked Dogs	Isaiah	Connor
	Poster Expanded Information	Zainab	
	Poster Dogs	Ekaterina	
* To be Implemented			

Note: Some user stories were smaller than others as well as some were related to others from the same category as such, the assignment of user stories for testing purposes was done based on their relationship to ensure proper coverage. (Example: All User preferences user stories were assigned to Edson for testing).

End to End Test Cases

Test Case ID	Test Case	Pre-condition	Test steps	Expected Result	Post condition	Actual result	Status
EmptyInputSignUp Test_0	Check error handling for empty input of username	There are no existing users in the database with the credentials the user is trying to sign up with.	Enter empty username, filled password, attempt to sign up, then attempt to log in with the same credentials.	User is blocked from signing up. Alert message alerts the user that their fields should not be empty.	No changes to users in the database.	User is successfully added to the database with an empty username and valid password. User is able to sign into account with the same credentials.	Fail ▾
EmptyInputSignUp Test_1	Check error handling for empty input of password	There are no existing users in the database with the credentials the user is trying to sign up with.	Enter a filled username, empty password, attempt to sign up, then attempt to log in with the same credentials.	User is blocked from signing up. Alert message alerts the user that their fields should not be empty.	No changes to users in the database.	User is successfully added to the database with an empty username and valid password. User is able to sign into account with the same credentials.	Fail ▾
EmptyInputSignUp Test_2	Check error handling for empty input of both username and password	There are no existing users in the database with the credentials the user is trying to sign up with.	Enter empty username, empty password, attempt to sign up, then attempt to log in with the same credentials.	User is blocked from signing up. Alert message alerts the user that their fields should not be empty.	No changes to users in the database.	User is successfully added to the database with an empty username and valid password. User is able to sign into account with the same credentials.	Fail ▾
InvalidCharacterIn putSignUpTest_0	Check error handling for invalid	There are no existing users in	Enter invalid character	User is blocked from signing up.	No changes to users in the	User is successfully added	Fail ▾

Test Case ID	Test Case	Pre-condition	Test steps	Expected Result	Post condition	Actual result	Status
	input of username	the database with the credentials the user is trying to sign up with.	username (non alphanumeric, excluding - or _ or spaces), valid password, attempt to sign up, and then attempt to log in with same credentials.	Alert message alerts the user that their fields should not contain invalid characters. Thus, account will not be created, and user will not be able to sign in.	database.	to the database, no alert message is shown. User is able to log in with invalid username and valid password.	
InvalidCharacterInputSignUpTest_1	Check error handling for invalid input of username, and empty input of password	There are no existing users in the database with the credentials the user is trying to sign up with.	Enter invalid character username (non alphanumeric, excluding - or _ or spaces), empty password, attempt to sign up, then attempt to log in with the same credentials.	User is blocked from signing up. Alert message alerts the user that their fields should not be empty.	No changes to users in the database.	The user is added successfully to the database with the empty password string and invalid username. No alert message is shown. User is able to log in with these credentials.	Fail -
EmptyInputLoginTest_0	Check error handling for empty input of username	No There are no existing users in the database with the credentials the user is trying to log in with.	Enter empty username, filled password, attempt to log in	User is blocked from logging in. Alert message alerts the user that their fields should not be empty.	No changes to users in the database.	System attempts to find user with empty username, but errors. There is no alert message. There are no changes to the user in the DB	Fail -
EmptyInputLoginTest_1	Check error handling for empty input of password	There are no existing users in the database with the credentials the user is trying to log in with.	Enter filled username, empty password, attempt to log in	User is blocked from logging in. Alert message alerts the user that their fields should not be empty.	No changes to users in the database.	System attempts to find user with username, and empty password but errors. There is no alert message. There are no changes to the user in the DB.	Fail -

Test Case ID	Test Case	Pre-condition	Test steps	Expected Result	Post condition	Actual result	Status
EmptyInputLogInTest_2	Check error handling for empty input of both username and password	There are no existing users in the database with the credentials the user is trying to log in with.	Enter empty username, empty password, attempt to log in	User is blocked from logging in. Alert message alerts the user that their fields should not be empty.	No changes to users in the database.	System attempts to find user with empty username and empty password. There is no alert message. There are no changes to the user in the DB	Fail ▾
InvalidCharacterInputLogInTest_0	Check error handling for invalid input of username	There are no existing users in the database with the credentials the user is trying to log in with.	Enter invalid character username (non alphanumeric, excluding - or _ or spaces), valid password, attempt to log in	User is blocked from logging in. Alert message alerts the user that their fields should not contain invalid characters.	No changes to users in the database.	System attempts to find user with invalid characters and fails to find them resulting in an exception. There is no alert message. There are no changes to the DB.	Fail ▾
InvalidCharacterInputLogInTest_1	Check error handling for invalid input of username, and empty input of password	There are no existing users in the database with the credentials the user is trying to log in with.	Enter invalid character username (non alphanumeric, excluding - or _ or spaces), empty password, attempt to log in	User is blocked from logging in. Alert message alerts the user that their fields should not be empty.	No changes to users in the database.	System attempts to find user with invalid characters and empty password and fails to find them resulting in an exception. There is no alert message. There are no changes to the DB.	Fail ▾
LongInputTest_0	Check if user is stopped from entering too many characters (50) in username	No precondition	Attempt to enter too many characters into the username input field.	User is not able to add any more characters after the limit is reached. User can only replace	No postcondition	User is not stopped from entering long input into username field. If they try to sign up, they are	Fail ▾

Test Case ID	Test Case	Pre-condition	Test steps	Expected Result	Post condition	Actual result	Status
				existing characters in username field, or delete existing characters in username field once limit is reached.		greeted with a "username in use" alert. No changes are made to the DB.	
LongInputTest_1	Check if user is stopped from entering too many characters (50) in password	No precondition	Attempt to enter too many characters into the password input field.	User is not able to add any more characters after the limit is reached. User can only replace existing characters in password field, or delete existing characters in password field once limit is reached.	No postcondition	User is not stopped from entering long input into password field. If they try to sign up, they are greeted with a "username in use" alert. No changes are made to the DB.	Fail ▾
SignUpTest_0	Check if user is blocked from creating account with existing username	Have an existing user with a valid username and password in the database	Attempt to create an account with the same username as the existing user in the database	User is not able to create an account. Alert message notifies the user that this username is already taken.	No changes to users in the database.	User is not able to create account with the same username as existing one. There are no changes in the DB, and the user is greeted with an alert, "username in use"	Pass ▾
LogInTest_0	Check if user is blocked from entering an account if their password is incorrect, but the	Have an existing user with a valid username and password in the database	Attempt to log in with existing username in database, but with incorrect password	User is not able to enter account. Alert message notifies the user that their password is incorrect.	No changes to users in the database.	User is not able to log into account, system sends an exception when they are not able to find the user	Fail ▾

Test Case ID	Test Case	Pre-condition	Test steps	Expected Result	Post condition	Actual result	Status
	username is correct.					with incorrect password. There is no alert shown. There are no changes to the DB.	
SuccessfulSignUp Test_0	Check that user can create an account with valid username, and password, and log into account on same session.	There is not an existing account in the DB with the credentials the user is trying to create a new account with.	Enter valid username and password into fields, sign up. Then, use the same username and password and attempt to log in.	User is able to make an account. Message alerts user that their account was created successfully. User logs in and is greeted with the dog profiles page.	New user is in the database with correct username and password.	The user is able to create a new account with valid username and password, and is able to log into the account on the same session.	Pass ▾
Successful_SignUpTest_1	Check that user can create an account with valid username, and password, and log into that account on a different session	There is an existing account in the DB with the credentials the user is trying to log in with.	Enter valid username and password into fields, sign up, close session. Then open a new session, use the same username and password as previous session and attempt to log in.	User is able to make an account. Message alerts user that their account was created successfully. User logs in and is greeted with the dog profiles page.	New user is in the database with correct username and password.	The user is able to create a new account with valid username and password, and is able to log into the account on a different session.	Pass ▾

Bug reports:

Empty Input	
Problem report number:	1
Reported by:	Ekaterina Kozlovsky
Date reported:	3/24/2024
Program (or component) name:	Login Scene
Release number:	2
Version (build) identifier:	ITR 2
Report type:	Coding
Can reproduce:	yes / no / sometimes / unknown
Severity:	Moderate
Priority:	Moderate
Problem summary:	User is allowed to sign up and/or log in with empty username and/or empty password.
Key words:	Text field, input, log in
Problem description and how to reproduce it:	Open application, attempt to sign up then log in with empty password and/or username fields. User will be saved in database, and allowed to log in (granted that there is not already an existing username in the DB)
Suggested fix:	Update the cases which check for empty fields, currently they are not correctly checking for the equality of strings. They should be using the equals method, not "==".
Status:	Open / closed / resolved
Resolution:	pending / fixed/ Cannot reproduce/ Duplicate/
Resolved by:	

Invalid Character Input	
Problem report number:	2
Reported by:	Ekaterina Kozlovsky
Date reported:	3/24/2024
Program (or component) name:	Login Scene
Release number:	2
Version (build) identifier:	ITR 2
Report type:	Coding
Can reproduce:	yes / no / sometimes / unknown
Severity:	Severe
Priority:	High
Problem summary:	User is allowed to sign up and/or log in with invalid username - program vulnerable to SQL injections.
Key words:	Text field, input, log in
Problem description and how to reproduce it:	Open application, attempt to sign up then log in with invalid characters in username (non alphanumeric, excluding - and _). User will be saved in database, and allowed to log in (granted that there is not already an existing username in the DB)
Suggested fix:	Update the cases which check for invalid characters. Use the string method, contains() to check if a text input contains invalid characters before attempting to parse the request. At the moment, there are no cases which check for invalid character input. Also, add an alert to notify the user that they should not be using invalid characters (and what those characters are).
Status:	Open / closed / resolved
Resolution:	pending / fixed/ Cannot reproduce/ Duplicate/
Resolved by:	


Long Input	
Problem report number:	3
Reported by:	Ekaterina Kozlovsky
Date reported:	3/24/2024
Program (or component) name:	Login Scene
Release number:	2
Version (build) identifier:	ITR 2
Report type:	Coding
Can reproduce:	yes / no / sometimes / unknown
Severity:	Moderate
Priority:	Moderate
Problem summary:	User is able to enter a long name and/or password, and incorrect alert is shown.
Key words:	Text field, input, sign up
Problem description and how to reproduce it:	Open application, attempt to sign with too many characters in either, or both password and username fields (long is > 50). The text input is not stopped after 50 characters, and is sent to DB on sign up. The system returns an incorrect alert, stating that the 'username is already taken'
Suggested fix:	There currently does not exist any constraints/assertions to prevent the user from entering too many characters. https://stackoverflow.com/questions/15159988/javafx-2-2-textfield-maxlength has some solutions for adding a maximum input length to the text fields in javafx (which the log in scene is using).
Status:	Open / closed / resolved
Resolution:	pending / fixed/ Cannot reproduce/ Duplicate/
Resolved by:	

Invalid Password Login	
Problem report number:	4
Reported by:	Ekaterina Kozlovsky
Date reported:	3/24/2024
Program (or component) name:	Login Scene
Release number:	2
Version (build) identifier:	ITR 2
Report type:	Coding
Can reproduce:	yes / no / sometimes / unknown
Severity:	Moderate
Priority:	Moderate
Problem summary:	User attempting to sign into account with correct username but incorrect password is shown the wrong alert.
Key words:	Text field, input, sign up
Problem description and how to reproduce it:	Open application, attempt to sign up with new valid username, and new valid password. Log into account using correct, valid username, but an incorrect password. User is not shown a message indicating them that their password is incorrect. Instead, there is a SQL exception in the console.
Suggested fix:	At the moment, the check for if the password is incorrect is incorrect. The existing if statement checks for the equality of usernames, but not for passwords. This statement needs to be reworked/reordered. In general, all of the cases should be handled by an error handler class.
Status:	Open / closed / resolved
Resolution:	pending / fixed/ Cannot reproduce/ Duplicate/
Resolved by:	

Code Review:

These are the smells I was searching for, and found:

- **Duplicated Code** 🐼💨
- **Speculative Generality** 🐼💨
- Indecent Exposure
- Comments
- Primitive Obsession
- Long method
- Large class
- Long parameter list
- Data Clumps 🐼💨
- Data Class
- Message chain
- **Feature Envy** 🐼💨
- Switch statements
- **Dead Code** 🐼💨
- **Uncommunicative Names** 🐼💨

Duplicated Code 	
Problem report number:	5
Reported by:	Ekaterina Kozlovsky
Date reported:	3/24/2024
Program (or component) name:	User Class
Release number:	2
Version (build) identifier:	ITR 2
Report type:	Coding smell
Severity:	Minimal
Priority:	Minimal
Problem summary:	<p>There are four methods which hold duplicated code. These four methods are identical, and return the same data types.</p> <ul style="list-style-type: none"> • <code>public ArrayList<Attribute> getCopyOfAgePreferences(ArrayList<Attribute> agePreferences)</code> • <code>public ArrayList<Attribute> getCopyOfSexPreferences(ArrayList<Attribute> sexPreferences)</code> • <code>public ArrayList<Attribute> getCopyOfEnergyLevelPreferences(ArrayList<Attribute> EnergyLevelPreferences)</code> • <code>public ArrayList<Attribute> getCopyOfSizePreferences(ArrayList<Attribute> sizePreferences)</code>
Key words:	Duplicate code, bloat
Suggested fix:	Since the return type of these methods is the same (children of Attribute class), and the input parameter is the same, these methods could be combined into one, <code>public ArrayList<Attribute> getCopyOfPreferences(ArrayList<Attribute> preferences)</code> ; or they could be replaced by a factory method.
Status:	Open / closed / resolved
Resolution:	pending / fixed/ Cannot reproduce/ Duplicate/
Resolved by:	

Dead Code 🐞💡	
Problem report number:	6
Reported by:	Ekaterina Kozlovsky
Date reported:	3/24/2024
Program (or component) name:	User class Login scene
Release number:	2
Version (build) identifier:	ITR 2
Report type:	Coding smell
Severity:	Minimal
Priority:	Minimal
Problem summary:	<p>The following attributes and methods are not being used for anything and should be deleted:</p> <ul style="list-style-type: none"> • User class <ul style="list-style-type: none"> ○ public ArrayList<Dog> getSponsoredDogs() ○ public String toString() ○ public String getPassword() ○ public String setPassword() • Login Scene <ul style="list-style-type: none"> ○ private User userInfo
Key words:	Dead, useless, bloat
Suggested fix:	Delete the code
Status:	Open / closed / resolved
Resolution:	pending / fixed/ Cannot reproduce/ Duplicate/
Resolved by:	

Speculative Generality 🐞💡	
Problem report number:	7
Reported by:	Ekaterina Kozlovsky
Date reported:	3/24/2024
Program (or component) name:	User class
Release number:	2
Version (build) identifier:	ITR 2
Report type:	Coding smell
Severity:	Minimal
Priority:	Minimal
Problem summary:	The email attribute is never used/not implemented.
Key words:	Dead, useless, bloat
Suggested fix:	Delete the attribute and its subsequent getters and setters
Status:	Open / closed / resolved
Resolution:	pending / fixed/ Cannot reproduce/ Duplicate/
Resolved by:	

Feature Envy 🧑🏻💡	
Problem report number:	8
Reported by:	Ekaterina Kozlovsky
Date reported:	3/24/2024
Program (or component) name:	Login Scene
Release number:	2
Version (build) identifier:	ITR 2
Report type:	Coding smell
Severity:	Minimal
Priority:	Minimal
Problem summary:	<p>There are several features that could be moved into its own class - text field sanitization/ error message class.</p> <p>This includes the following:</p> <ul style="list-style-type: none"> • Cases which check if the string in the text input fields (username, password) are valid • Error message method • Authenticating if the username/password exists
Key words:	Encapsulation
Suggested fix:	<p>Create a login helper class, called Authenticator which:</p> <ul style="list-style-type: none"> • Sanitizes text fields/checks for valid string • Checks if there is an existing user • Validates its password • Logs in/ signs up if successful • Sends error message if unsuccessful
Status:	Open / closed / resolved
Resolution:	pending / fixed/ Cannot reproduce/ Duplicate/
Resolved by:	

Uncommunicative Names 🐶🐕	
Problem report number:	9
Reported by:	Ekaterina Kozlovsky
Date reported:	3/24/2024
Program (or component) name:	User class
Release number:	2
Version (build) identifier:	ITR 2
Report type:	Coding
Severity:	Minimal
Priority:	Minimal
Problem summary:	<p>The following methods have uncommunicative names (and parameters):</p> <ul style="list-style-type: none"> • public void addPassedDogs(Dog likedDog) • public void addLikedDogs(Dog likedDog)
Key words:	Communication, names
Suggested fix:	<p>Change the names of these methods and parameters:</p> <ul style="list-style-type: none"> • public void addDogToPassedDogs(Dog passedDog) • public void addDogToLikedDogs(Dog likedDog)
Status:	Open / closed / resolved
Resolution:	pending / fixed/ Cannot reproduce/ Duplicate/
Resolved by:	