

Breazy Fit

Low Level Design



BREAZY FIT

Github Link:

https://github.com/ChazArvizu/CECS491_Hexadecimators.git

Team Name: Hexadecimators

Team Leader: Chaz Arvizu

Team Members: Carlsean Claricia, Tania Adame, Tyler Kelsey,
Andrew De La Rosa, Sean Iida

Submission Date: 11/09/2022

Version History

Version 1.0 - Initially Created: 10/18/2022

Table of Contents

Page Number

Section 1: Scope of Low Level Design.....	3
Section 2: Low Level Design Diagrams.....	3
2.1: Logging.....	3
2.1.1: Logging Success Case 1.....	3
2.1.2: Logging Success Case 2.....	4
2.1.3: Logging Success Case 3.....	4
2.1.4: Logging Success Case 4.....	5
2.1.5: Logging Failure Case 1.....	5
2.1.6: Logging Failure Case 2.....	6
2.1.7: Logging Failure Case 3.....	7
2.1.8: Logging Failure Case 4.....	7
2.1.9: Logging Failure Case 5.....	8
2.2: Registration.....	8
2.2.1: Registration Success Case 1.....	8
2.2.2: Registration Failure Case 1.....	9
2.2.3: Registration Failure Case 2.....	10
2.2.4: Registration Failure Case 3.....	10
2.2.5: Registration Failure Case 4.....	11

1. Scope of Low Level Design

The scope of our Low Level Design document is to show how the success and failure cases for the registration and logging classes will work within our product. Diagrams and appropriate descriptions will be provided to help give the client a better understanding of how each of these success and failure events will occur and how they will be handled.

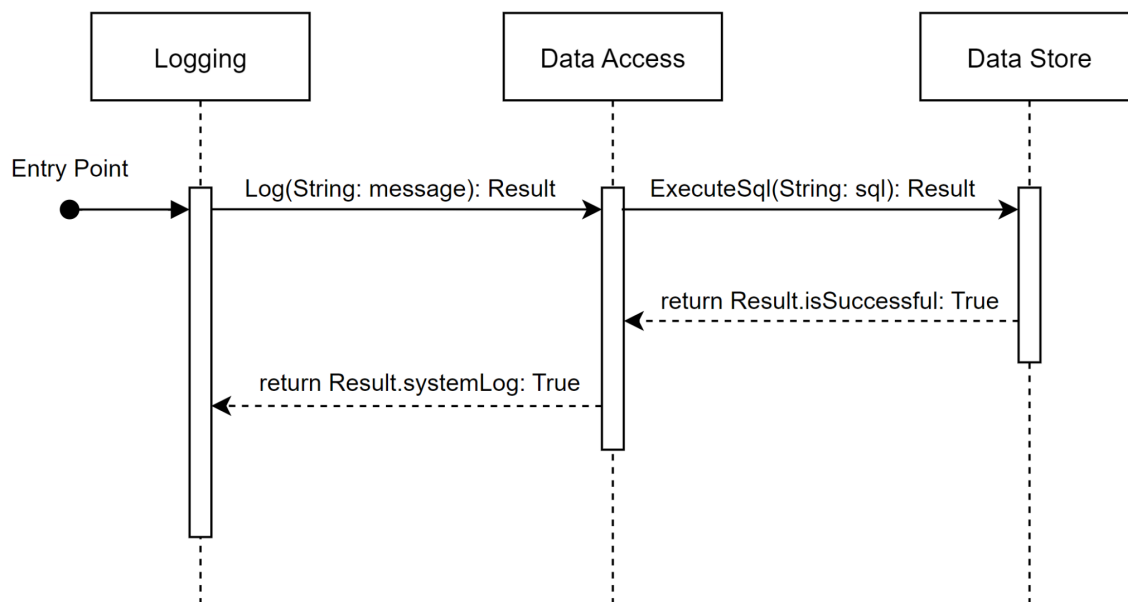
2. Low Level Design Diagrams

2.1 Logging

2.1.1 Logging Success Case 1

Case - The system logs success events

Sequence Diagram

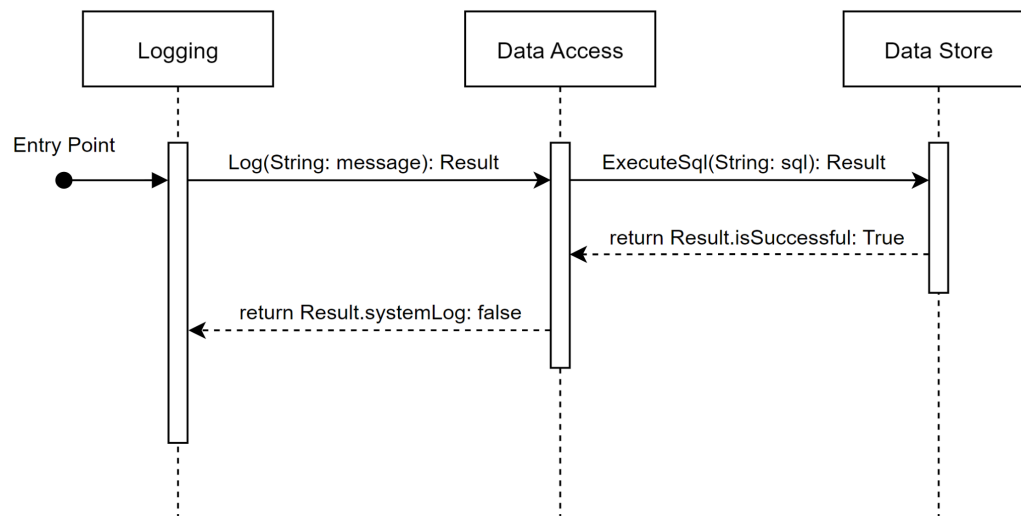


Sequence Flow Explained - At the entry point of the diagram is some system success event that has occurred. Within the business logic the Logging class will contain the method Log(string message) to log the success events in the SQL database. Within the Data Access layer the ExecuteSql(string sql) method will execute the sql query necessary to add the log entry to the SQL database. The Data Store Layer will then store the system success log entry.

2.1.2 Logging Success Case 2

Case - The system logs failure events

Sequence Diagram

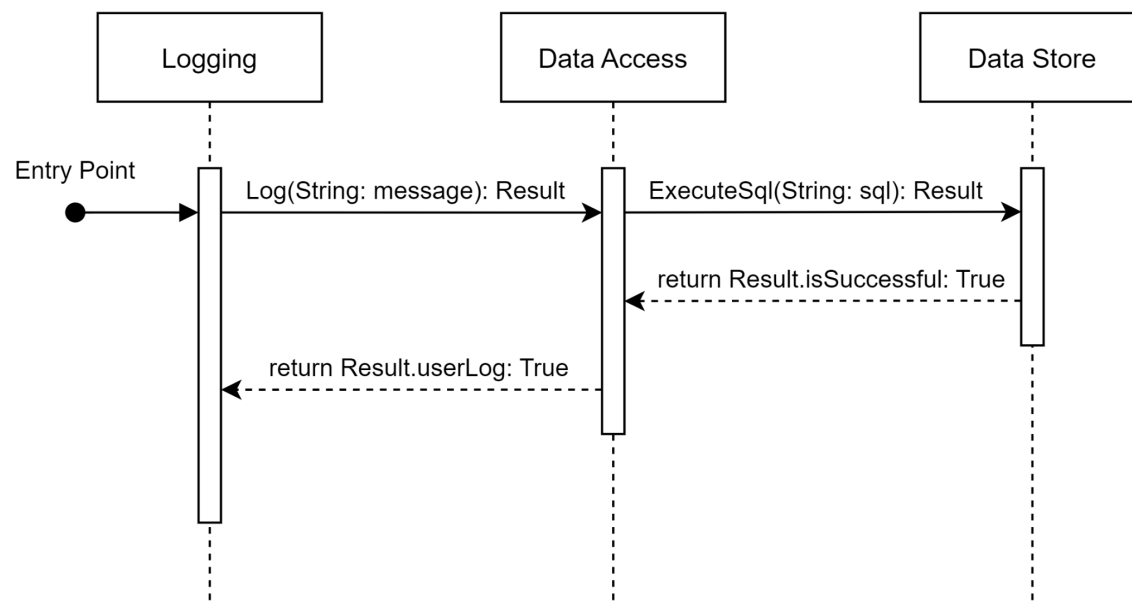


Sequence Flow Explained - At the entry point of the diagram is some system failure event that has occurred. Within the business logic the Logging class will contain the method Log(string message) to log the success events in the SQL database. Within the Data Access layer the ExecuteSql(string sql) method will execute the sql query necessary to add the log entry to the SQL database. The Data Store Layer will then store the system failure log entry.

2.1.3 Logging Success Case 3

Case - The system logs user success events

Sequence Diagram

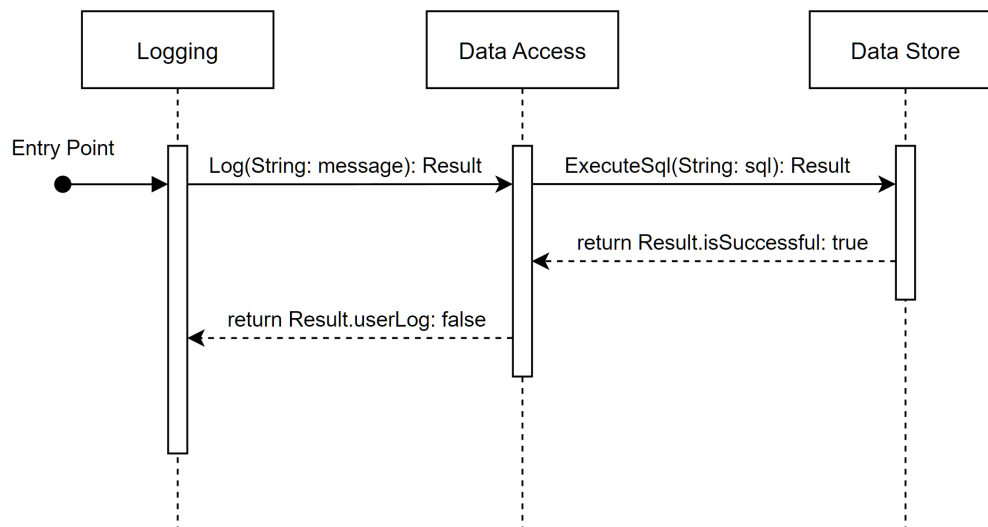


Sequence Flow Explained - At the entry point of the diagram is some user success event that has occurred. Within the business logic the Logging class will contain the method Log(string message) to log the success events in the SQL database. Within the Data Access layer the ExecuteSql(string sql) method will execute the sql query necessary to add the log entry to the SQL database. The Data Store Layer will then store the user success log entry.

2.1.4 Logging Success Case 4

Case - The system logs user failure events

Sequence Diagram

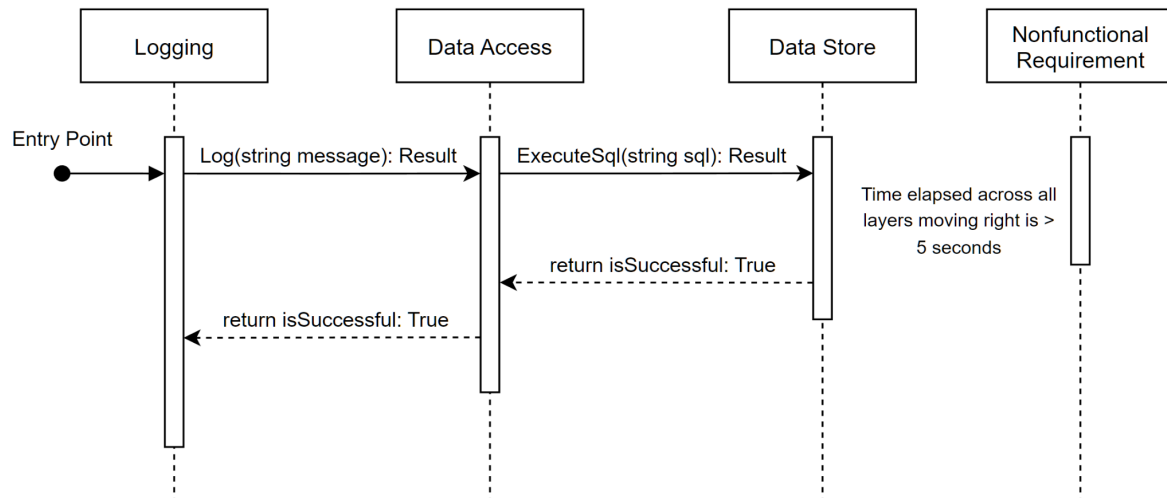


Sequence Flow Explained - At the entry point of the diagram is some user failure event that has occurred. Within the business logic the Logging class will contain the method Log(string message) to log the success events in the SQL database. Within the Data Access layer the ExecuteSql(string sql) method will execute the sql query necessary to add the log entry to the SQL database. The Data Store Layer will then store the user failure log entry.

2.1.5 Logging Failure Case 1

Case - The logging process took longer than 5 seconds to complete upon invocation

Sequence Diagram

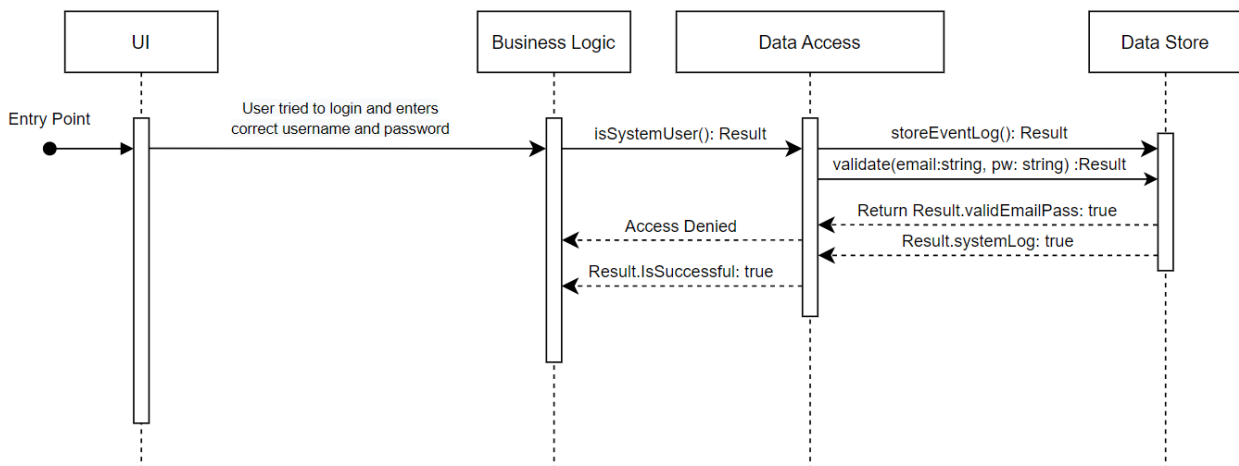


Sequence Flow Explained - Failure to meet the nonfunctional requirement of completing the process in under 5 seconds results in a logging fail. The sequence follows the same flow as a successful case until our clock internally reaches 5 seconds during the logging process. Our storeEventLog() then logs a failure case instead of a success. The process still executes normally and users are not blocked from registering.

2.1.6 Logging Failure Case 2

Case - The logging process blocks a user from interacting with the system

Sequence Diagram

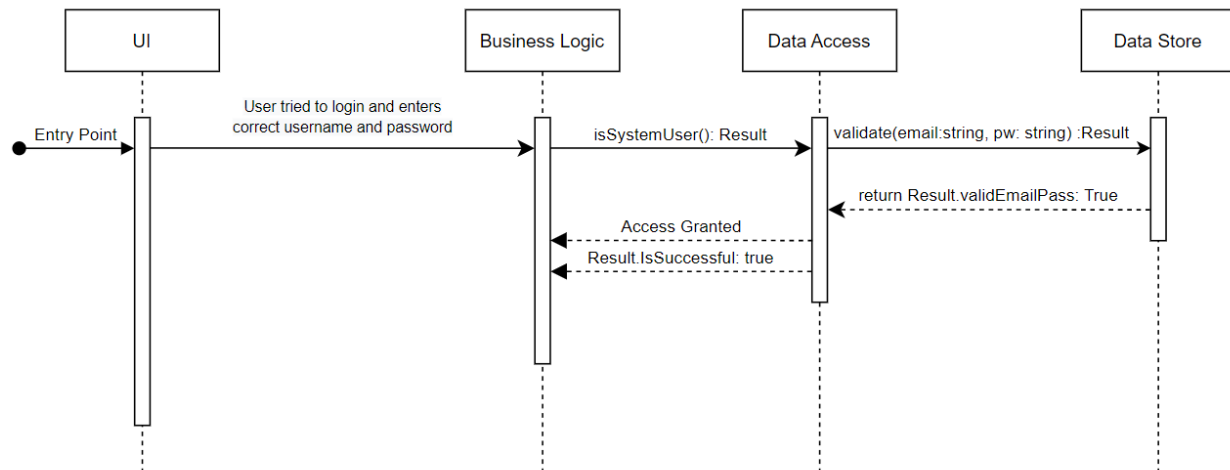


Sequence Flow Explained - Failure to allow the user access after inputting the correct logging info. The user inputs their unique username and password combination which is then passed into the data access point then into the data store through the validate() method which returns a true result and logs the interaction successfully only for the data access point to incorrectly interpret the result and deny access to the user.

2.1.7 Logging Failure Case 3

Case - The logging process completes within 5 seconds, but did not save to a persistent data store

Sequence Diagram

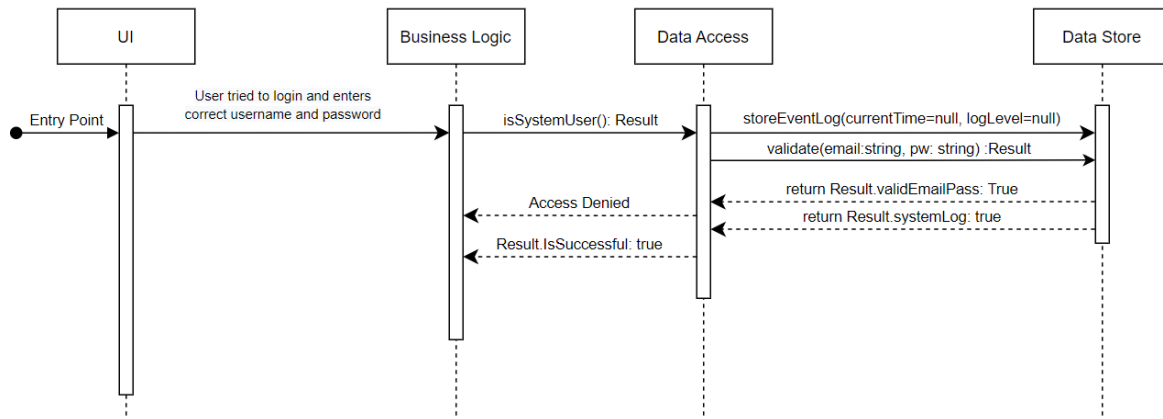


Sequence Flow Explained - The system is passed the correct unique username and password combination and allows the user access into the system but fails to initiate any of the storeEventLog() methods to pass into the data store leading to an undocumented logging attempt within the data store.

2.1.8 Logging Failure Case 4

Case - The logging process completes within 5 seconds, but did not accurately save the event to the persistent data store

Sequence Diagram



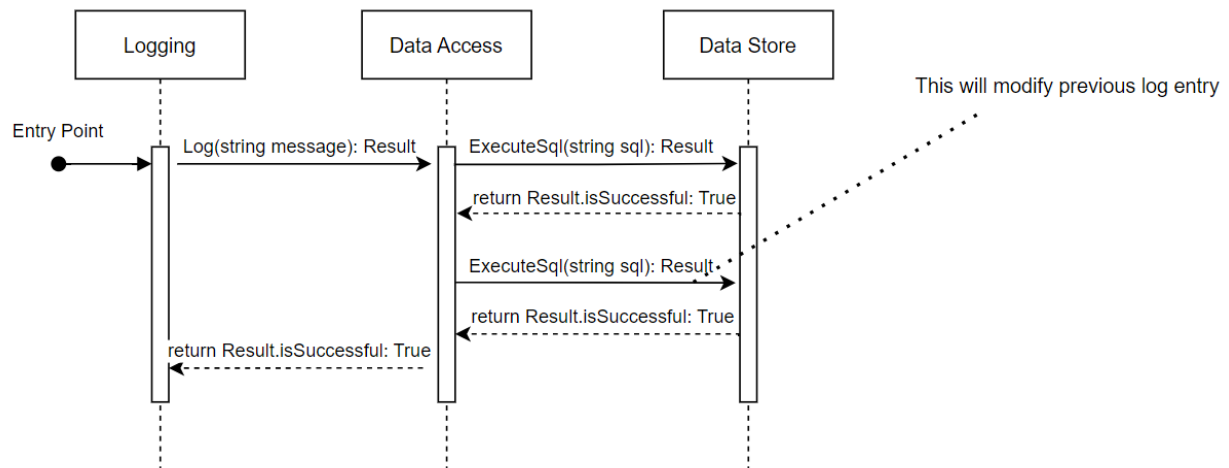
Sequence Flow Explained - The user inputs the correct data into the UI and all the data is validated through the validate() method and subsequently logged into the data

store with incorrect data for the event time or the log level which then incorrectly generates a false log with incorrect info inside of the data store

2.1.9 Logging Failure Case 5

Case - Previously saved log entries are modifiable

Sequence Diagram



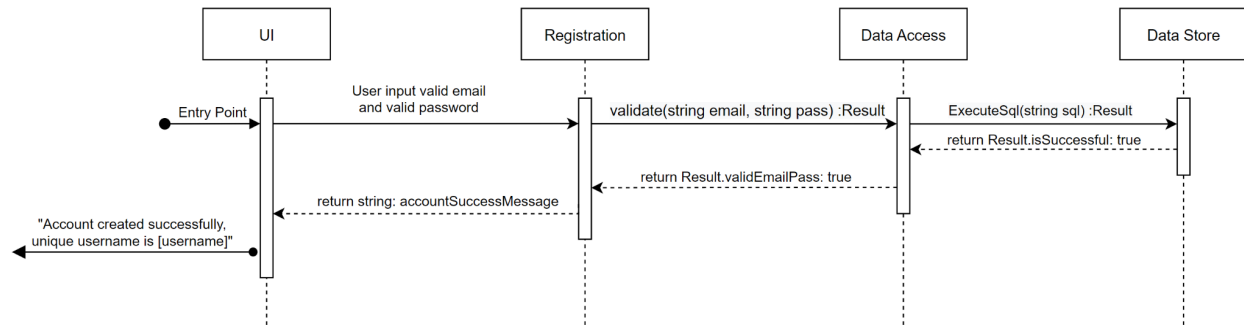
Sequence Flow Explained - The entry point is when the system tries to log any kind of log event, after logging the event using the ExecuteSql() method it calls the the same method again and overwrites the previous log entry, causing the logging to fail as it is now modifiable.

2.2 Registration

2.2.1 Registration Success Case 1

Case - Users register with a valid email and valid passphrase . They system is able to assign a system-wide unique username. A system message displays “Account created successfully” within 5 seconds of invoking the registration process. The system provides the username to the user.

Sequence Diagram

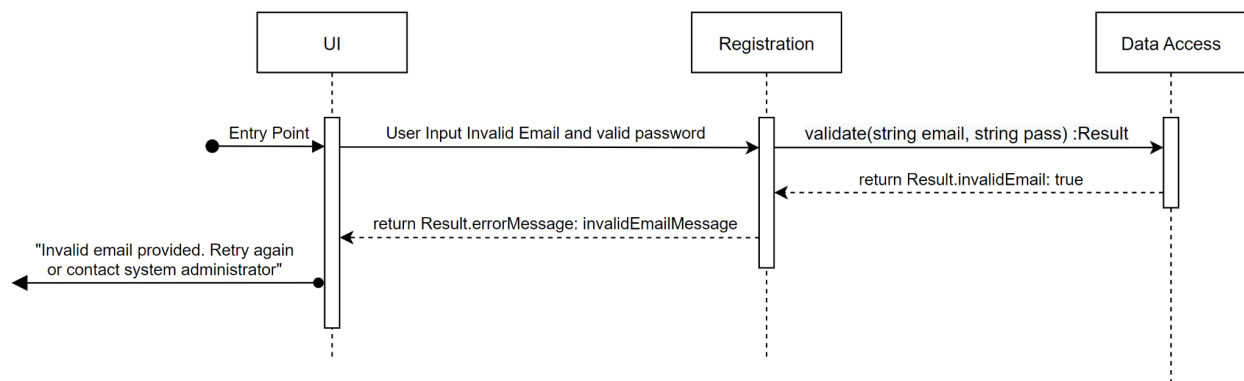


Sequence Flow Explained - At the entry point of the diagram a user will have inputted a valid email and valid password from the UI, our Registration class will then take these strings as parameters and pass it into the our Validate(string email, string pass) method to check to make sure they are valid, after confirming that both the email and password are valid it will then generate a unique username, which is then passed to the Data Access layer. The Data Access layer will then use the ExecuteSql(string sql) method to store the user's registration information into the database, after everything has been properly stored, the system will then display a message to the user stating that their account was created and it will also display their unique username.

2.2.2 Registration Failure Case 1

Case - User Registers with an invalid email. A system message displays "Invalid email provided. Retry again or contact system administrator" or no system message. Account is not created.

Sequence Diagram



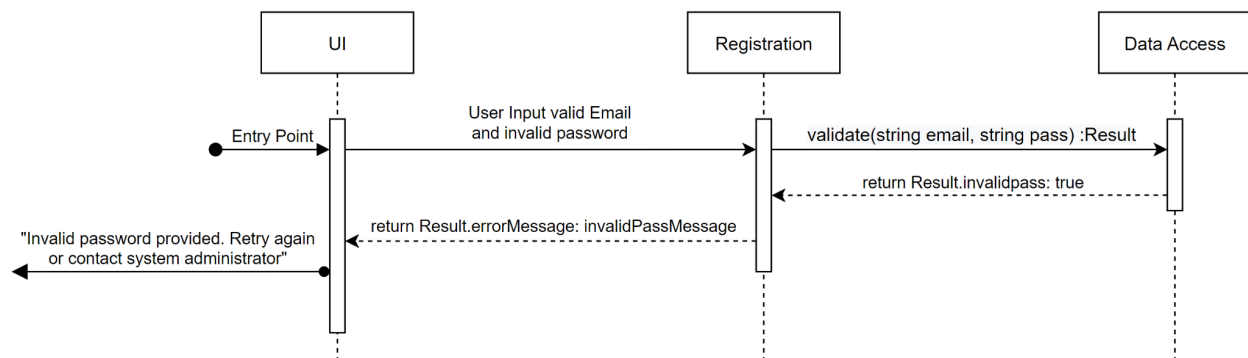
Sequence Flow Explained - At the entry point of the diagram a user will have inputted an invalid email and valid password from the UI, our Registration class will then take these strings as parameters and pass them into the Validate(string email, string pass) method, this would then fail while trying to send the information to the Data Access layer

since the email in this case is invalid. After failing the system will then display a message to the user stating that their email was invalid and to try again or contact a system administrator.

2.2.3 Registration Failure Case 2

Case - User registers with an invalid passphrase. A system message displays “Invalid passphrase provided. Retry again or contact system administrator” or no system message, Account is not created.

Sequence Diagram

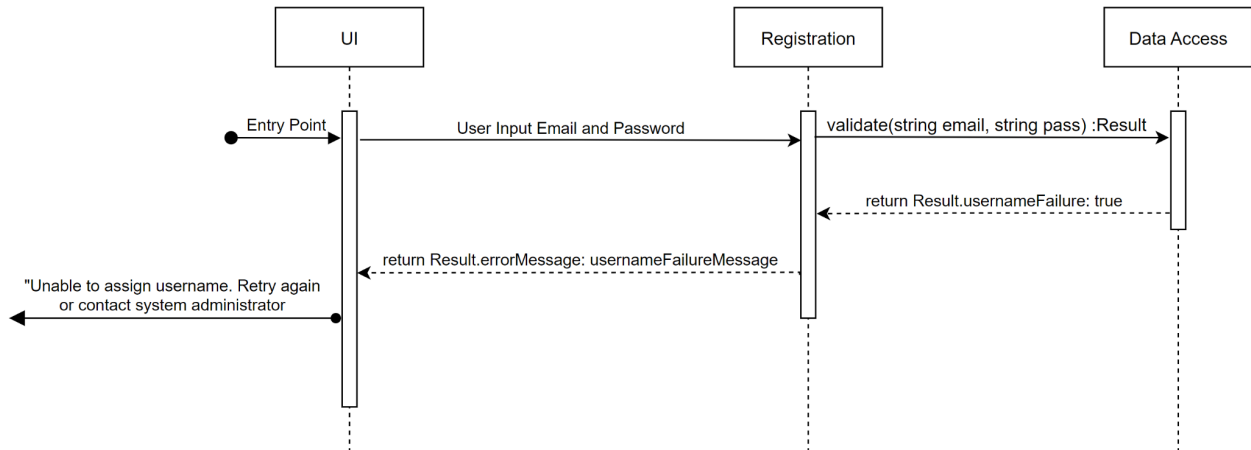


Sequence Flow Explained - At the entry point of the diagram a user will have inputted a valid email and invalid password from the UI, our Registration class will then take these strings as parameters and pass them into the Validate(string email, string pass) method, this would then fail while trying to send the information to the Data Access layer since the password in this case is invalid. After failing the system will then display a message to the user stating that their password was invalid and to try again or contact a system administrator.

2.2.4 Registration Failure Case 3

Case - User registers with a valid email and valid passphrase. The system was unable to assign a system-wide username. A system message displays “Unable to assign username. Retry again or contact system administrator”. Account is not created.

Sequence Diagram

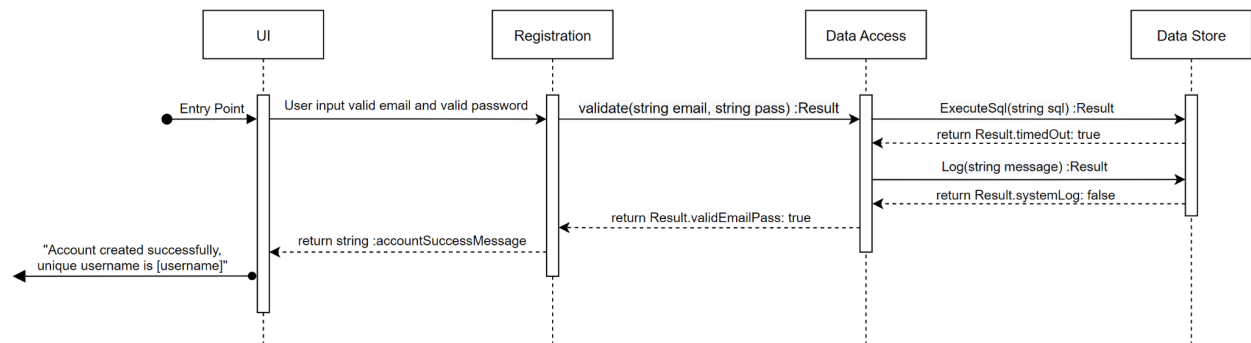


Sequence Flow Explained - At the entry point of the diagram a user will have inputted a valid email and valid password from the UI, our Registration class will then take these strings as parameters and pass it into the our Validate(string email, string pass) method to check to make sure they are valid, after confirming that both the email and password are valid it will then generate a unique username, however, in this case the system was not able to properly generate a unique username, so the process fails and an account is not created. After failing the system will then display a message to the user stating that it was unable to assign the user their username and to try again or contact a system administrator.

2.2.5 Registration Failure Case 4

Case - User registers with a valid email and valid passphrase. The system was able to assign a system-wide username. Entire process took longer than 5 seconds. A system log entry is recorded. Account is created.

Sequence Diagram



Sequence Flow Explained - At the entry point of the diagram a user will have inputted a valid email and valid password from the UI, our Registration class will then take these strings as parameters and pass it into the our Validate(string email, string pass) method to check to make sure they are valid, after confirming that both the email and password are valid it will then generate a unique username, which is then passed to the Data Access layer. The Data Access layer will then use the ExecuteSql(string sql) method to store the user's registration information into the database, however, even though the entire process was a success, in this case the entire process took longer than 5 seconds, so the system will recognize that and log a system failure event. After everything has been properly stored, the system will then display a message to the user stating that their account was created and it will also display their unique username.