



# 4TH YEAR PROJECT

UCLA DOCUMENT

X00144631 – Ryan Deering  
Supervisor: Gary Clynych

Contents

Use Cases: .....2

Logical Architecture – Full Stack..... 10

Use Cases:

ID:	1
Title:	Follow User
Description:	A Registered User wants to 'follow' another Registered User, to keep up to date with their posts and to see their posts on their feed.
Primary Actor:	Registered User 1: Wants to follow the Registered User Registered User 2: Is being followed by Registered User 2
Preconditions:	Registered User 1 is not currently following Registered User 2.
Postconditions:	Registered User 1 has followed Registered User 2.
Main Success Scenario:	<div>1. Registered User 1 goes onto Registered User 2's profile.</div> <div>2. Registered User 1 clicks the Follow User button on Registered User 2's profile.</div> <div>3. Registered User 1 is now following Registered User 2.</div>
Extensions:	<div>1a. Registered User 1 might be blocked by Registered User 2, therefore unable to see their posts or the button to follow them.</div> <div>2a. Registered User 1 may be already following Registered User 2, and might not see the Follow User button, but an unfollow button.</div>
Frequency of Use:	Following a Registered User is a very common thing to do on a social media platform. Depending on usage, this may happen a lot on the system.

<b>ID:</b>	2
<b>Title:</b>	Register User
<b>Description:</b>	A guest User wants to become a Registered User, to have their details into the database and to be able to access the full functionality a registered user does.
<b>Primary Actor:</b>	(Guest) User
<b>Preconditions:</b>	Guest User is not a Registered User.
<b>Postconditions:</b>	Guest User has become a Registered User.
<b>Main Success Scenario:</b>	<ol style="list-style-type: none"><li>1. Guest User decides to register to the website.</li><li>2. Guest User goes onto the Registration page and inputs their details.</li><li>3. Guest User clicks register.</li><li>4. The system validates the Guest User's inputs.</li><li>5. Successfully validated, the account is created but the user must verify their account through email.</li><li>6. Guest User becomes a Registered User.</li></ol>
<b>Extensions:</b>	<p>4a. The password might not be suitable for the regex expression and ask for a stronger password to be hashed into the database.</p> <p>4b. An account with that email may already exist.</p> <p>5a. Account may never be verified. Will be deleted in the database after 3 days.</p>
<b>Frequency of Use:</b>	Registration of new users is a common and crucial component of a social media system, but only happens once per user. Typically.

<b>ID:</b>	2
<b>Title:</b>	Register User
<b>Description:</b>	A guest User wants to become a Registered User, to have their details into the database and to be able to access the full functionality a registered user does.
<b>Primary Actor:</b>	(Guest) User
<b>Preconditions:</b>	Guest User is not a Registered User.
<b>Postconditions:</b>	Guest User has become a Registered User.
<b>Main Success Scenario:</b>	<ol style="list-style-type: none"><li>1. Guest User decides to register to the website.</li><li>2. Guest User goes onto the Registration page and inputs their details.</li><li>3. Guest User clicks register.</li><li>4. The system validates the Guest User's inputs.</li><li>5. Successfully validated, the account is created but the user must verify their account through email.</li><li>6. Guest User becomes a Registered User.</li></ol>
<b>Extensions:</b>	<p>4a. The password might not be suitable for the regex expression and ask for a stronger password to be hashed into the database.</p> <p>4b. An account with that email may already exist.</p> <p>5a. Account may never be verified. Will be deleted in the database after 3 days.</p>
<b>Frequency of Use:</b>	Registration of new users is a common and crucial component of a social media system, but only happens once per user. Typically.

<b>ID:</b>	<b>3</b>
<b>Title:</b>	<b>Purchase Print</b>
<b>Description:</b>	<b>A Registered User 1 wishes to purchase a print of a post Register User 2 has uploaded. Registered User 2 has enabled the ability to purchase prints of their art or photography.</b>
<b>Primary Actor:</b>	<b>Registered User 1: Wishes to buy a print. Registered User 2: Enabled their post to be monetized.</b>
<b>Preconditions:</b>	<b>Registered User 1 has not bought a print.</b>
<b>Postconditions:</b>	<b>Registered User 1 buys and receives the print. Registered User 2 receives a payment with their take of the money paid.</b>
<b>Main Success Scenario:</b>	<b><ol style="list-style-type: none"><li>1. Registered User 1 sees Registered User 2's post on their feed.</li><li>2. Registered User 1 decides to purchase a print of Registered User 2's post.</li><li>3. They are taken to a screen where they can choose what photos of the post (one if there is only one) they want to be printed.</li><li>4. They may choose the size of the prints they want and how many prints they want. As well as the type of paper they want.</li><li>5. The system calculates an amount to be paid from what Registered User 1 inputted.</li><li>6. Proceeding to the next page, the payment engine pops up and asks the user to put in their payment details.</li><li>7. Payment is authorized and order is processed.</li></ol></b>
<b>Extensions:</b>	<b>2a. Registered User 2 may have not enabled their print to be monetized, meaning that Registered User 1 cannot buy a print. 7a. Registered User 1 may not have the funds to authorize the payment.</b>
<b>Frequency of Use:</b>	<b>Low to medium. Most social media users pay no money to the platform they use, it is safe to assume it will probably be the case here, too.</b>

<b>ID:</b>	4
<b>Title:</b>	Submit Post
<b>Description:</b>	Registered User wishes to submit a post containing photography, or art.
<b>Primary Actor:</b>	Registered User
<b>Preconditions:</b>	The post isn't in the system and hasn't been submitted.
<b>Postconditions:</b>	The post is in the system and will appear on the people who follow the Registered User's feeds.
<b>Main Success Scenario:</b>	<ol style="list-style-type: none"><li>1. Registered User clicks on the Submit Post button, leading them to the page to submit it.</li><li>2. The Registered User attaches pictures to their post. A maximum of 4 images is allowed per post.</li><li>3. The Registered User can attach a caption to their post. They can also enable monetization.</li><li>4. The Registered User submits their post.</li><li>5. The system compresses the images enough without compromising quality, and the post is uploaded.</li><li>6. The post appears on whoever is following the Registered User.</li></ol>
<b>Extensions:</b>	<p>2a. The Registered User <b>MUST</b> include an image with their post. They cannot upload without.</p> <p>2b. The images must be of a valid file type. They can't upload .RAW files for instance.</p> <p>3a. They can enable monetization but must include their payment details and be verified by phone too. If they haven't, they can't monetize.</p>
<b>Frequency of Use:</b>	High. Submitting a post is the basis of how social media works.

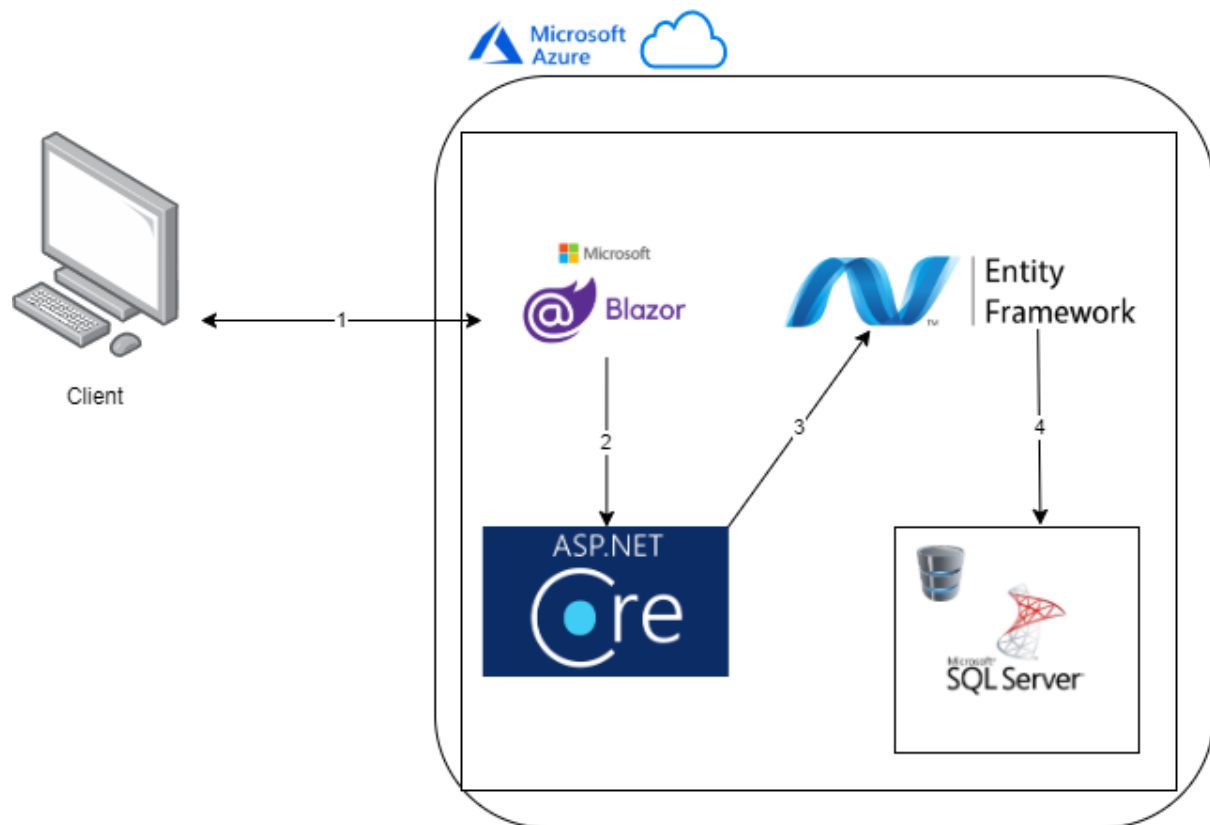
<b>ID:</b>	5
<b>Title:</b>	Submit Comment on Post
<b>Description:</b>	Registered User 1 wishes to leave a comment on Registered User 2's post.
<b>Primary Actor:</b>	Registered User 1: Commenting on post Registered User 2: Post is being commented on.
<b>Preconditions:</b>	A comment has not been submitted on the post in question.
<b>Postconditions:</b>	Registered User 2's comment appears on Registered User 1's post.
<b>Main Success Scenario:</b>	<ol style="list-style-type: none"><li>1. Registered User 1 sees Registered User 2's post on their feed.</li><li>2. Registered User 1 decides to write a comment under it.</li><li>3. The comment is submitted.</li><li>4. The comment is shown on Registered User 2's post, showing Registered User 1 that their submission was successful.</li></ol>
<b>Extensions:</b>	3a. Registered User 2 may have been suddenly blocked by Registered User 1, preventing the post from being submitted. 3b. There may be a timeout on the comments preventing submission. For example, Registered User 1 may have already submitted a comment beforehand.
<b>Frequency of Use:</b>	Very high. Comments are continuously placed on a social media platform.



<b>ID:</b>	<b>6</b>
<b>Title:</b>	<b>Send Message</b>
<b>Description:</b>	<b>Registered User 1 wishes to message Registered User 2.</b>
<b>Primary Actor:</b>	<b>Registered User 1: Messenger Registered User 2: Recipient</b>
<b>Preconditions:</b>	<b>Both users need to be following each other in order to message.</b>
<b>Postconditions:</b>	<b>Registered User 2 receives a message from Registered User 1.</b>
<b>Main Success Scenario:</b>	<b>1. Registered User 1 wants to message Registered User 2. 2. Registered User 1 writes a message and submits it. 3. Registered User 2 sees on the top right of their screen they have received it.</b>
<b>Extensions:</b>	<b>2a. Both users may not be able to follow each other. 2b. Registered User 2 may have blocked Registered User 1, not seeing the message.</b>
<b>Frequency of Use:</b>	<b>High. It is common to communicate on social media.</b>

<b>ID:</b>	7
<b>Title:</b>	Like Post
<b>Description:</b>	Registered User 1 wants to 'like' Registered User 2's post.
<b>Primary Actor:</b>	Registered User 1: Wishes to like Registered User 2's post Registered User 2: Owner of post
<b>Preconditions:</b>	Registered User 1 has not liked Registered User 2's post.
<b>Postconditions:</b>	Registered User 1 has liked Registered User 2's post.
<b>Main Success Scenario:</b>	<ol style="list-style-type: none"><li>1. Registered User 1 sees Registered User 2's post on their feed.</li><li>2. Registered User 1 likes Registered User 2's post.</li><li>3. Registered User 2 gets a notification saying their post has been liked by Registered User 1.</li></ol>
<b>Extensions:</b>	2a. Registered User 2 might have blocked Registered User 1, prohibiting them from liking their post. 2b. Registered User 1 might have already liked Registered User 2's post. 2c. Registered User 2 might have deleted the post.
<b>Frequency of Use:</b>	High. It is common to like each other's posts on social media.

## Logical Architecture – Full Stack



1. Blazor - Why Blazor? Blazor is an interesting new framework for building client-side web UI. It can interoperate with JavaScript if need be, but you can use it to write C# for both the front-end and the back end.
2. ASP.NET Core Web API – As Blazor is a component of ASP.NET, we can use .NET's Web API to easily communicate with the Blazor front-end code and Entity Framework. Allows for simple serialization of JSON and authentication, authorization. Allows us to create our API controllers.
3. Entity Framework – Provides us our data access layer, to serve data to the Blazor client. Allows us to define our models.
4. SQL Server – Hosted on our Azure instance, allows us to store our data in a performant and secure way. It also keeps with the Microsoft stack and could be easy to use with Entity Framework.