Software Requirement Analysis (SRA)

**Project Title:**

Online social networking system

Team Members:

* Anish Manandhar – CS23BTKMU11001
* Ankush Niroula - CS23BTKMU11002
* Vignan Kota – CS21BTECH11029
* David Maloth – CS21BTECH11035

# a) Context Diagram:

User

Online Social Networking System

Interactions

# b) Data Flow Diagrams:

A diagram of a process

Description automatically generatedA black text on a white background

Description automatically generated

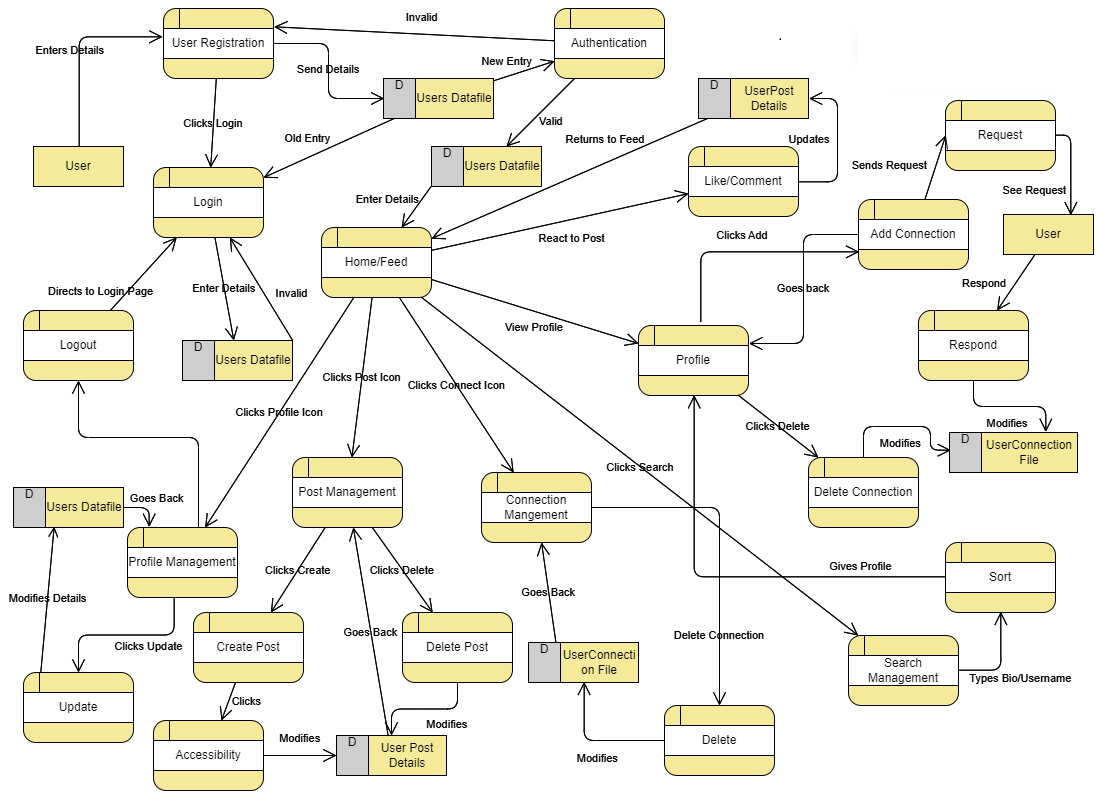
**DFD – 1:**

**A diagram of a user flow

Description automatically generated**

* This is a Level – 1 DFD.

**DFD – 2:**

****

* This is a Level – 2 DFD. Every key process mentioned in Level-1 would be examined in further detail in a Level-2 DFD. For example, within the "post management" process, we can describe the subprocesses of creating a post, deleting a post, manage accessibility, and the flow of data between these sub-processes.

***Reasons for selecting DFD - 2:***

* Level-2 DFD gives detailed insights into processes in social networking.
* Explicitly shows data flows and transformations in each sub-process.
* Crucial for understanding how user data is processed, stored, and shared.
* Enables identification of areas for optimization, process refinement, and enhancing data security.
* Better choice for a thorough understanding of the online social networking system.

# c) Function Point Analysis:

**1. Identify and Count Function Types:**

Identify and count the different types of functions in the software. The five function types are:

**External Inputs (EI): 11**

* User Registration
* User Login
* Create Post
* Like/Comment on Post
* Add Connection
* Respond to Connection Request
* Update Profile
* Delete Connection
* Create Post
* Delete Post
* Update Post

**External Outputs (EO): 5**

* View Profile
* View Feed
* View Connections
* View Post Details
* View Search Results

**External Inquiries (EQ): 1**

* Search

**Internal Logical Files (ILF): 3**

* User Database
* Post Database
* Connection Database

**External Interfaces (EIF): 1**

* Authentication System

**2. Assign Weights:**

Assign weights to each function type based on its complexity. The weights typically being classified as low, average, and high complexity. The complexity is determined by factors such as data validation, processing logic and user interactions.

|  |  |  |  |
| --- | --- | --- | --- |
| **Functional Units** | **Weighting Factor** | | |
| Low | Average | High |
| External Input (EI) | 3 | 4 | 6 |
| External Output (EO) | 4 | 5 | 7 |
| External Inquiry (EQ) | 3 | 4 | 6 |
| Internal Logical File (ILF) | 7 | 10 | 15 |
| External Interface File (EIF) | 5 | 7 | 10 |

**3. Unadjusted Function Point (UFP):**

Let us assume that EI are of low complexity, EO are of high complexity, EQ is of average complexity, ILF are of high complexity and EIF is of low complexity.

**4. Value Adjusted Factor (VAF):**

* Assess the system based on 14 general system characteristics such as data communications, performance, heavily used configuration etc.
* Assign a score from 0 to 5 for each characteristic based on its level of influence.
* Now we need to score the 14 general system characteristics:

1. Data communications - 3
2. Distributed data processing - 1
3. Performance - 4
4. Heavily used configuration - 2
5. Transaction rate - 4
6. Online data entry - 5
7. End-user efficiency - 3
8. Online update - 5
9. Complex processing - 4
10. Reusability - 2
11. Installation ease - 1
12. Operational ease - 3
13. Multiple sites - 2
14. Facilitate change - 4.

* Calculate Total Score:
* Calculate Value Adjusted Factor:
* The VAF usually ranges from 0.65 to 1.35.

**5. Adjusted Function Point (AFP):**

**6. Size of Code:**

For JavaScript, based on industry data, the average (LOC/AFP ratio) is 64 LOC/AFP

So, the estimated size of code in JavaScript would be:

Size of Code = AFP \* (LOC/AFP ratio)

= 131 \* 64

= 8,384 lines of JavaScript code

In summary, for a system with 131 Adjusted Function Points, we would expect the size of the JavaScript code to be approximately 8,384 lines.