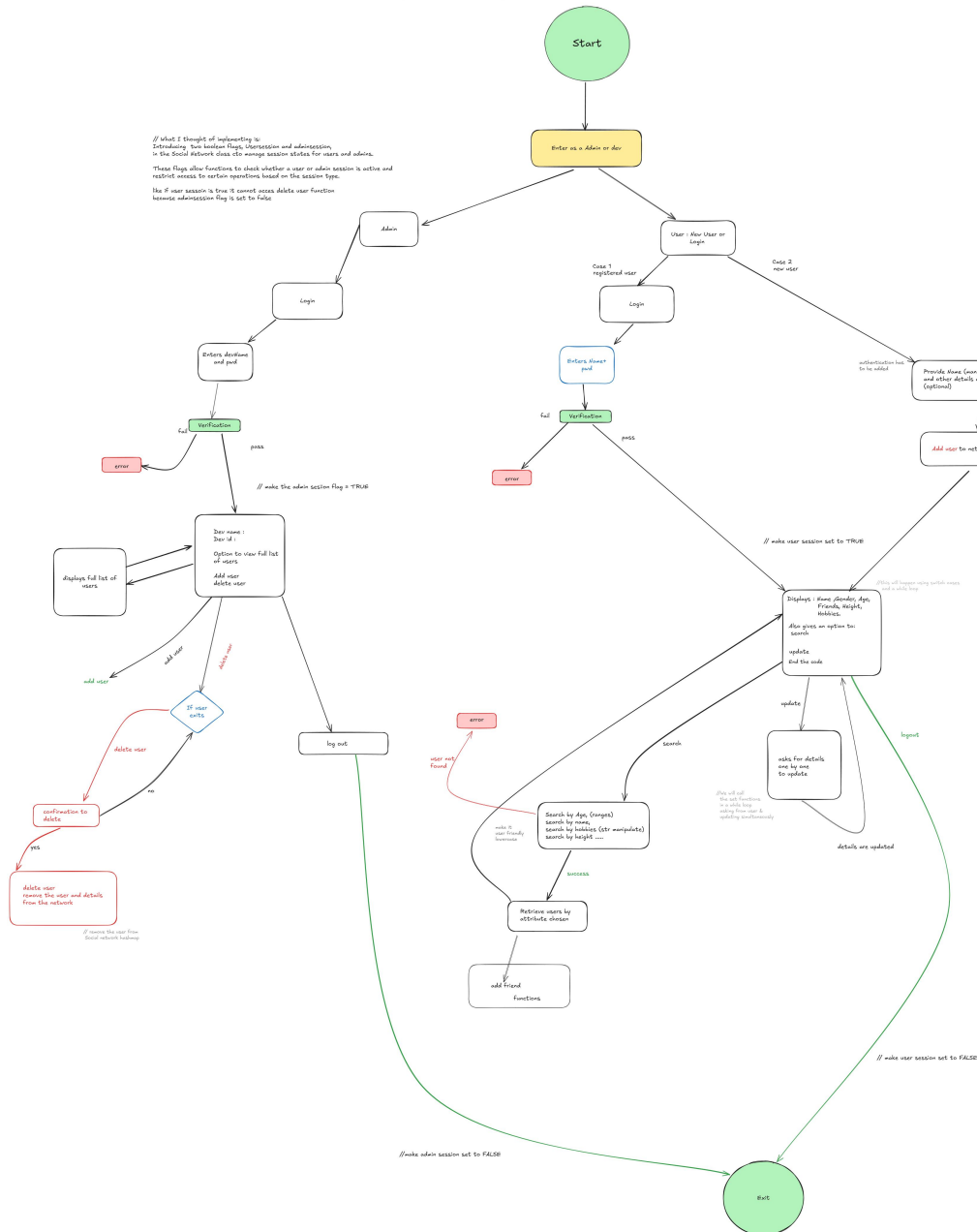


the problem statement goes like..



## Dummy Data base

name	Unique ID
Tom	UID
Lauren	U24
Patrice	U32
Stella	M32
Monica	F3

Hobbies Hash list	Stores "String"
Running	

Friends Hash list	Stores Unique ID of friends
UID	U24

HashMap of social network	
UID	Row user
M32	Stella User

**Class user**

internalId: UUID  
name: String  
age: Integer  
height: hobbies, friends

getter & setters

**Class SocialNetwork**

private Map<String, User> usersById;  
private Map<String, List<User>> usersByHobby;  
private Map<Integer, List<User>> usersByAge;  
private Map<String, List<User>> usersByHobby;

addUser(User: User): UUID  
deleteUser(userId: UUID): boolean  
searchUserByName(name: String): List<User>  
searchUserByAge(age: int): List<User>  
searchUserByHobbies(hobbies: Set<String>): List<User>

**Remaining things :-**

- Error handling
- Efficiency & details
- Add Test cases also

**Questions:-**

## Features

Unique ID Generation

Friend Relationship Management

Indexed Search (by Name, Age, Hobbies)

Efficient Friend add + remove

User Deletion

Hobby-Based Search

Java

UUID for Unique ID Generation

HashMap for Indexing (by ID, Name, Age, Hobby)

HashSet for Friend Relationships

**class SocialNetwork {**  
private Map<String, User> usersById;  
private Map<String, List<User>> usersByHobby;  
private Map<Integer, List<User>> usersByAge;  
private Map<String, List<User>> usersByHobby;  
**}**

// Stores users by their unique ID