

# Tech Report

## **Application Architecture/ Business Logic/ Core Data Model/ Constraints**

### Introduction

Our app "Scran" will prioritise giving users a pleasant experience in browsing recipes and ingredients. It will allow users to create profiles, post content, interact with other users' content and follow other users.

### Application Architecture

The app will adopt a Monolithic Architecture design as our app is a smaller-scale application. This choice allows for faster development speed and simplified debugging as the application is created within a single code base. It is worth noting that this has been done by Jhipster's default option.

### Business Logic

Scrans' business logic will be :

1. Users must create an account
2. Users can share Post ( there are 2 types )
3. Users can interact with Posts (Add to Pantry, view post, comment, review)
4. Users can follow and chat with other users
5. Users can search for specific post or specific users

### Core Data Model

Scrans' core data model includes these following entities and relationship :

1. **User** : Includes fields for usernames, email, password, profile picture, biography, unique identifier  
*Users can have many post ,many friends ,one Pantry, one preference*
2. **Preference (settings)** : Includes fields for Dietary, Preferred shops, blacklist, user's ID  
*Belong to one User*
3. **Friends** : Includes fields for user's ID and the added other users' ID  
*Belong to one User*
4. **Post Recipe** : Includes fields for picture, Title, Description, Dietary tags, User ID of the author, unique identifier, estimate cook time, portion size  
*Belong to one User, can have many Ingredients*
5. **Post Ingredient** : Includes fields for picture, Title, Description, Dietary tags, Shop tag, price, unique identifier
6. **Post Comment** : Includes fields for Post ID, comments description, User ID, timestamp, Review
7. **Pantry** : Include fields for Recipe ID, Ingredient ID  
*Belong to one User*

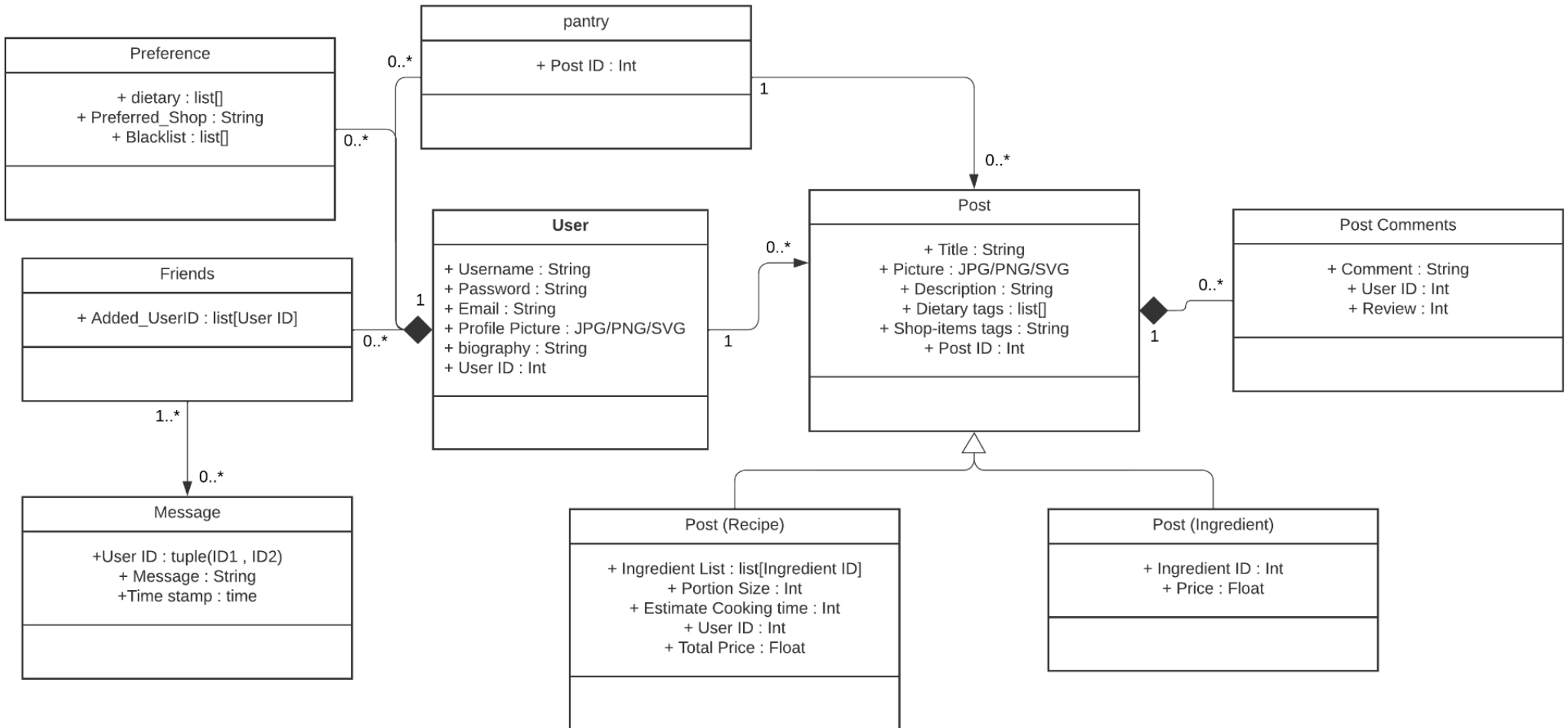
### Constraints

To ensure the app's integrity and security, the following constraints are implemented :

1. User email must be unique to prevent duplication accounts
2. Post, Comments, or reviews can't be edited once it is created to ensure integrity
3. All user are allowed to view posts, but are required to log in for other functionality such as commenting, review, messaging.

# Tech Report

## Application Architecture/ Business Logic/ Core Data Model/ Constraints



## **Tech Report**

**Application Architecture/ Business Logic/ Core Data Model/ Constraints**