

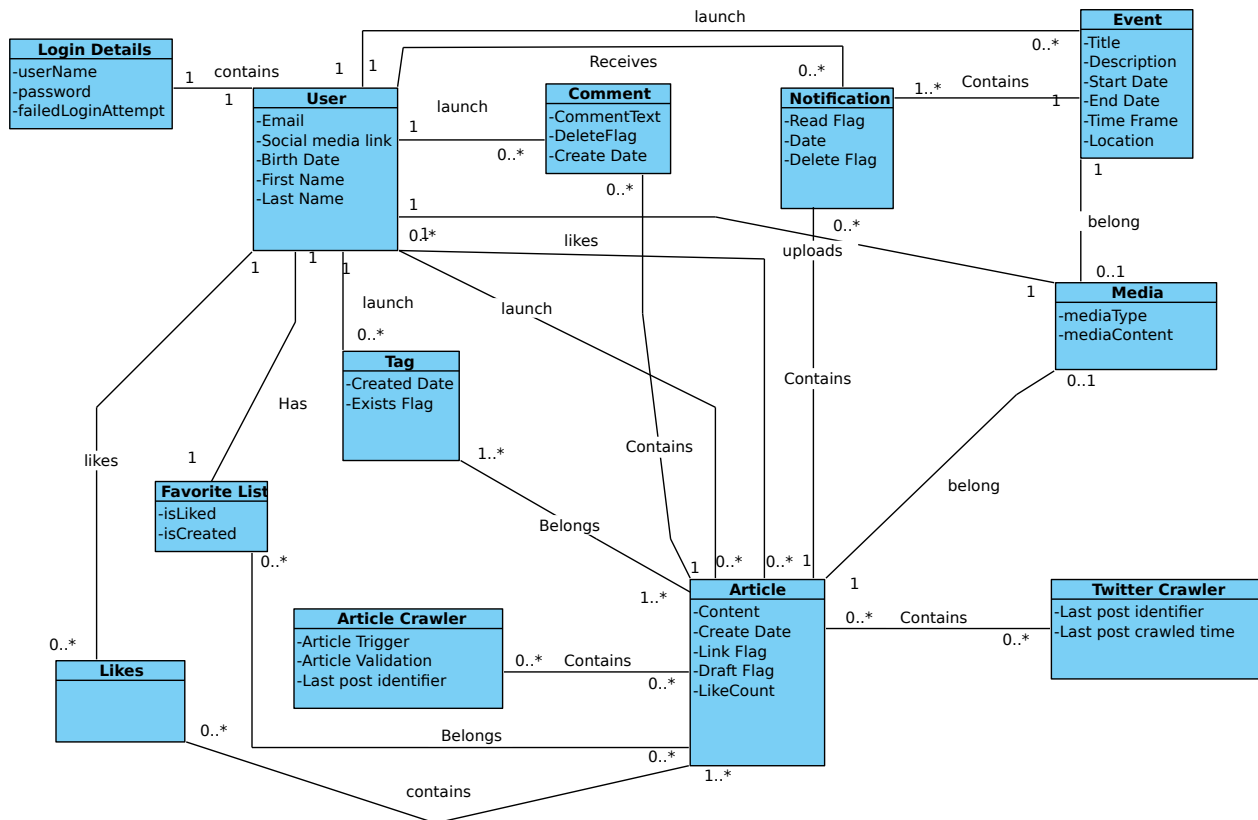
# Assignment 04(Project)

## Bear Feed

Mushfika Rahman  
Sadia Nasrin Tisha  
Muhammad Ashfakur Arju  
Tingshuo Miao  
Noorah Ashmeel

1st October 2022

## 1 Domain Model:



## 2 GRASP in Domain Model

### 2.1 GRASP pattern: Information Expert

1. *User*- For Authenticate and Login Details, it is necessary to know about the user. It is necessary to know about the user for creating Event. User is responsible to create a Tag.
2. *Twitter Crawler*- Twitter Crawler have a message subscriber which will listen to notification from twitter. It will validate the new post with predefined logic and if everything satisfied it will send the post to Article service for saving it to Database.
3. *Baylor News Crawler*- Baylor News Crawler has periodic call to Baylor news portal and it will face any new article. Baylor news crawler have a record of last fetched article from the news portal to commit duplicate article save. Article save is done asynchronously via message queue.

### 2.2 GRASP pattern: Partial Expert

1. *Article*- To create Article, we need to assign Tag and need to know about User. To fulfill the new responsibilities require information that is spread across these classes of objects Tag and User.
2. *Favorite List*- To modify the Favorite list, two responsibilities will be assigned, User and Article. To fulfill the new responsibilities require information that is spread across these classes of objects Article and User.
3. *Comment*- To create Comment, two responsibilities will be assigned, User and Article. To fulfill the new responsibilities require information that is spread across these classes of objects Article and User.
4. *Notification*- To create Comment, three responsibilities will be assigned, User and Article and Event. To fulfill the new responsibilities require information that is spread across these classes of objects Article and User and Event.

### 2.3 GRASP pattern: Creator

1. *Article creates Article*- Whenever a User creates an Article, the system will create Article instance and attaches specific tag from Tag class. Therefore, Article class is responsible for creating an article.
2. *Tag creates Tag*- Whenever a User creates a Tag, the system will create Tag instance. Therefore, Tag class is responsible for creating an tag.
3. *Event creates Event* Whenever a User creates an Event, the system will create Event instance. Therefore, Event class is responsible for creating an article.
4. *Comment creates Comment* Whenever a User creates a Comment, the system will create a Comment instance. Therefore, the Comment class is responsible for creating a comment.
5. *Media creates Media* Whenever a User uploads Media, the system will create Media instance. Therefore, Media class is responsible for creating a media.

## 2.4 GRASP pattern: Low Coupling

1. Low coupling exists between Login Detail and User via intermediate table. Because Login Detail will keep some attributes of User.
2. Low coupling exists between Crawlers and Article via intermediate table
3. Low coupling exists between Article and Comment via intermediate table. Because Comment will keep some attributes of Article.
4. Low coupling exists between Article and Tag via intermediate table.

## 2.5 GRASP pattern: High Cohesion

1. User has high cohesion because a user collaborates with other classes to fulfill the responsibilities.
2. Article has high cohesion as it has the responsibility to create the Article and Add to Favorite.
3. Event has high cohesion as it has the responsibility to create Event

## 2.6 GRASP pattern: Controller

1. *Articles Controller*- It handles a view routes request to Article entity to create article. Gives us the full view of the full article. It redirects us to a view which has the full contents of the article and all comments related.
2. *Event Controller* It handles a view routes request to Event entity to create event.
3. *Login Controller* It will handle the user's authentication process. It will handle the request to reset use password.
4. *Search Controller* It will handle the user tag search and return a view with all specific articles.
5. *User Controller* It will handle the request of updating profile from user.
6. *Tag Controller* It handles the creating tag.

## 2.7 GRASP pattern: Polymorphism

1. *User*- User class can have a updateProfile method. This method is implemented differently based on the parameter.

## 2.8 GRASP pattern: Pure Fabrication

In our whole system, we need to persist with the database. For the persistent and fetching data, we access it through interface. so our classes such as user, article, event is coupled with database interfaces.