**UniPro**

**Jeong Eun Chong (B1002334)**

**Chong Jun Ming (B1201361)**

1. **Introduction**
   1. **Background**

Before internet developed, consumers were only able to make through offline or physical store to obtain goods, getting goods from overseas were more of a limited choice to choose and even more expensive than original price. Customers who required the item will had no other choice but to spend more efforts and methods to obtain the goods such as using a proxy to help the items while paying additional huge amount of price or with extra service charges applies on the goods. However, the condition has changed after internet formed, and transit technologies developed. Customers started to order through online from local oe international places, and receive items by parcels. Dependency on offline sellers is decreased, and advantage of them became demerit. Then sellers also expanded their business to online, and in this process, a lot of similar items get various prices as setting of sellers. In old time, market chose customer, but as customers become worldwide, customers choose market nowadays. Sellers started offer some promotions to attract customer’s attention, as the internet and e-commerce grows, too much information were spread across and the information became too broad which made consumers missed out the promotions or certainly not knowing where to find such promotions of the goods the consumers wanted.

So, we decided to develop one website, Unipro, as to let anyone to gather and share the known promotion information in one concentrated area to help each other to consume wisely.

* 1. **Objective**

This platform will provide the sharing information services between user to user in some particular categories, like Clothes, IT, Food, etc.

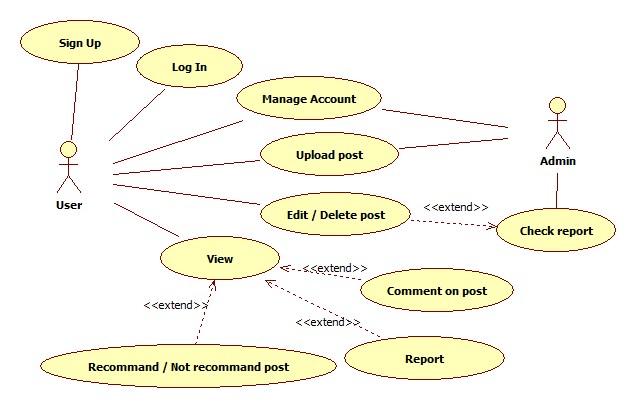
* To educate users about wise consumption for their wanted item.
* To broaden the communication between users of the specific post and information (ex: comments feature allows users to ask, answer or provide more additional information of that particular post)
* Centralize the information so users don’t need to research many sites
* To provide the location (geographical coordinate) of promotion place, if it’s offline promotion

1. **Requirement Definition**
   1. **Functional Requirement**

Our UniPro project will require such things for its function mentioned below.

* System must be connected to Internet always for every update from user and developer
* User can Sign Up and Log In to system
* The account can be managed by user and admin (Authority for user account to ban if account is being hacked or spread illegal ad, etc.)
* User and admin can upload, edit, and delete posts
* User can recommend or not recommend, and left comments on posts
* User can report bad posts, comments on post, wrong category, and user account, etc. (Incase account hacked or spread illegal ads)
* Admin can see the reports from users and do right action (edit category of post, delete post, delete comment, etc.)
  1. **Non-Functional Requirement**
* The post should be appeared in no more than few min.
* User can view/search/filter the post but are not able to place comment, likes/dislikes or create a post unless user are logged into their account.
* Users are also allowed to report inappropriate post only when logged in
* The system is capable to hold up for 100000 accounts registration without an issue
* The system will be able to sustain the performance up to 20000 accounts logged in at the same time
* The system can store up to 1millions of the post in the meantime with the current server.
* the response of the pages or function will not be more than 5second
* The system can be performed and receive submitted post into the system up to 5000 at the same time and update the post on webpage not more than 3second
* A backup system prepared in case of an event of a system failure
* Any portion of the hardware failure in storage will immediately switch to mirrored disks
* system may be able to scale up and handle wide variety of system configuration sizes by increase machines of hardware for the database server
* the system’s defect rate is to expect to be less than 1 failure per 720hours (30days) of operation
* Users and client only can access their own data and services once they’ve logged into their account
* Certain features must be authorized in descending order to access the services, an example would be in order to post on the website, and you need to create an account
* The user confidential and personal data will be kept private and will not be used in any way or leak to 3rd party

1. **Requirement Specification**
   1. **Use Case Diagram and Use Cases**



* + 1. **High level Use Cases**

|  |  |
| --- | --- |
| **Use Case 1** | Sign up |
| **Actor(s)** | User |
| **Description** | This use case begins when user want to Sign up to system. The user input user details such as userID, password, first name, last name, e-mail address, date of birth, gender for sign up to system |

|  |  |
| --- | --- |
| **Use Case 2** | Log in |
| **Actor(s)** | User |
| **Description** | This use case begins when user want to access to system. The user input userID and password to log in to system. |

|  |  |
| --- | --- |
| **Use Case 3** | Manage account |
| **Actor(s)** | User, Admin |
| **Description** | This use case begins when user want to manage their own account. The user input user details what user want to change, such as password, first name, last name, e-mail address, date of birth, gender to system. User can deactivate account their own.  The admin can ban the account based on report of other users. |

|  |  |
| --- | --- |
| **Use Case 4** | Upload post |
| **Actor(s)** | User, Admin |
| **Description** | This use case begins when user want to upload promotion information. The user input promotion information such as category, title, detail, due date.  The admin can upload post as important notice, or announcement. |

|  |  |
| --- | --- |
| **Use Case 5** | Edit / Delete post |
| **Actor(s)** | User, Admin |
| **Description** | This use case begins when user want to edit what user uploaded. The user edit category, title, detail, due date, or delete post what they uploaded.  The admin can edit or delete every posts to maintain system. |

|  |  |
| --- | --- |
| **Use Case 6** | View |
| **Actor(s)** | User |
| **Description** | This use case begins when user click on post titles. The user can view the post details. |

|  |  |
| --- | --- |
| **Use Case 7** | Comment on post |
| **Actor(s)** | User |
| **Description** | This use case begins when user want to comment on post. The user can comment on the posts as reply type. |

|  |  |
| --- | --- |
| **Use Case 8** | Recommend / Not recommend |
| **Actor(s)** | User |
| **Description** | This use case begins when user want to mark the post whether they recommend it or not. The user can express it through click recommend or not recommend button. |

|  |  |
| --- | --- |
| **Use Case 9** | Report |
| **Actor(s)** | User |
| **Description** | This use case begins when user want to report post, comment, or user. The user can report to admin if they see the post on wrong category, bad comment, user who spread illegal ad, etc. |

|  |  |
| --- | --- |
| **Use Case 10** | Check report |
| **Actor(s)** | Admin |
| **Description** | This use case begins when admin receives report from users. The admin read the report from users, and check whether the report is right or wrong, and solve the problem (edit post, delete post or comment, and ben account, etc.) as admin based on report. |

* 1. **Expanded Use Cases**

|  |  |  |
| --- | --- | --- |
| Use Case 1 | Sign Up | |
| **Goal in Context** | To create a new account and record information about the user | |
| **Primary Actor**  **Secondary Actor** | User | |
| **Trigger** | When user want to use the additional functions of the Unipro that needed an account | |
| **Typical Course of Events** | | |
| **Actor Action** | | **System Response** |
| 1. This use case begins when a user access to the sign up page and need to get a new account | |  |
| 1. The user enters the userID, password, firstName, lastName, emailAddress, dateOfBirth, gender into the system. | | The account status is set to “active”, the user number is generated and save details into system. |
| Alternative Course of Events | | |
| 2a. if the Username or E-mail Address already existed, the system will display it to user and require user to change it before reapply | | |

|  |  |  |
| --- | --- | --- |
| Use Case 2 | Log In | |
| **Goal in Context** | To access the account | |
| **Primary Actor**  **Secondary Actor** | User | |
| **Trigger** | When the user want to access to their account | |
| **Typical Course of Events** | | |
| **Actor Action** | | **System Response** |
| 1. This use case begins when a user want to access the account | |  |
| 1. The user will enter userID and password. | | The name and status logged in are displayed |
| Alternative Course of Events | | |
| 2a. if the user provided a wrong username and password, the user must  reenter the username and password into the system. | | |

|  |  |  |
| --- | --- | --- |
| Use Case 3 | Manage account | |
| **Goal in Context** | To maintain and record information about the user | |
| **Primary Actor**  **Secondary Actor** | User  Admin | |
| **Trigger** | When the user want to edit, remove or update existing information of the account | |
| **Typical Course of Events** | | |
| **Actor Action** | | **System Response** |
| 1. This use case begins when the user wants to manage information of the account | |  |
| 1. The user click into the user manage account page | | The user record of firstName, lastName, emailAddress, gender and dateOfBirth are displayed |
| 1. The user update the existing information of the user account | | The system overwrites any changes of the previous information. |
| Alternative Course of Events | | |
| 2a. if the user decided to deactivate, the user set deactivate the account into  the system. The system will remove the account.  2b. if the user account was reported by others and the admin found fault,  admin set the ban status into the system and lock the account.  3a. if the user wants to change password, the user input the new password into  the system and overwrite the previous password. | | |

|  |  |  |
| --- | --- | --- |
| Use Case 4 | Upload Post | |
| **Goal in Context** | To upload post on the news feed | |
| **Primary Actor**  **Secondary Actor** | User  Admin | |
| **Trigger** | When the user want to create a new promotion post | |
| **Typical Course of Events** | | |
| **Actor Action** | | **System Response** |
| 1. This use case begins when a user want to make a new promotion post | |  |
| 1. The user enters the category, title, detail and due date | | The post status is set to “active”, the postID is generated and datePosted is set to the system date |
| Alternative Course of Events | | |
| 2a. if the user is admin, then the admin will input title and detail. The system  will set the category as announcement. The post status is set to “special”,  the postID is generated and the dateposted is set to the system date. | | |

|  |  |  |
| --- | --- | --- |
| Use Case 5 | Edit / Delete post | |
| **Goal in Context** | To user and admin to edit or delete posts | |
| **Primary Actor**  **Secondary Actor** | User  Admin | |
| **Trigger** | When user wants to edit or delete post what they uploaded. | |
| **Typical Course of Events** | | |
| **Actor Action** | | **System Response** |
| 1. This use case begins when user want to edit what user uploaded. | |  |
| 1. The user click edit button on post. | |  |
| 1. The user update category, title, detail, due date and save it. | | The system overwrites any changes of the previous post |
| Alternative Course of Events | | |
| 2a. if user want to delete the post, user set delete to system, then system  remove post.  2b. <<extended>> if admin want to delete the post based on report system,  admin set delete to system, then system remove post. | | |

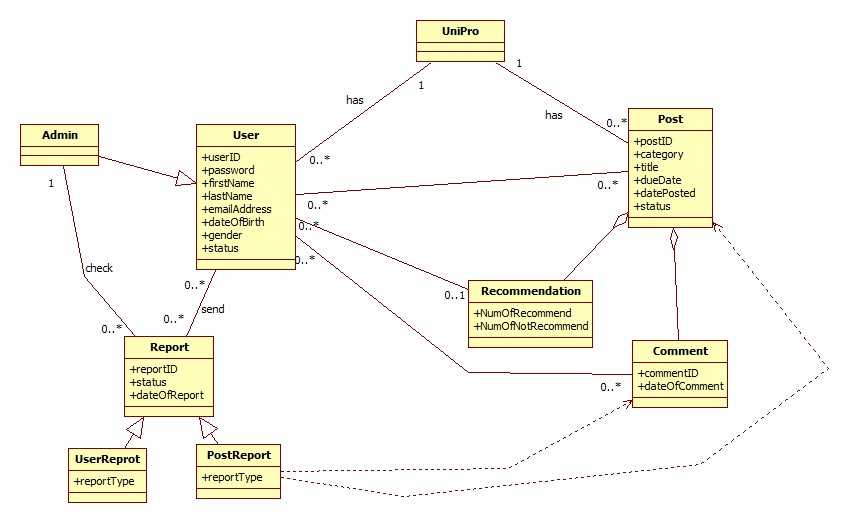
|  |  |  |
| --- | --- | --- |
| Use Case 6 | View | |
| **Goal in Context** | To view the existing posts | |
| **Primary Actor**  **Secondary Actor** | User | |
| **Trigger** | When the user want to view information of existing post | |
| **Typical Course of Events** | | |
| **Actor Action** | | **System Response** |
| 1. This use case begins when a user want to view the information of an existing post | |  |
| 1. The user will click on the post | | The post title, due date, category and details are displayed |
| Alternative Course of Events | | |
| 2a. if the user wants to find a specific post, the user can find the existing post  by entering postID or title, the post is displayed and user can click on the  post to display the information.  3b. if the user want to find posts that is based on category, the user can find  the existing posts by entering the category and the posts are displays.  The user can click the post to display the content. | | |

|  |  |  |
| --- | --- | --- |
| Use Case 7 | Comment on post | |
| **Goal in Context** | To user to left comment about posts | |
| **Primary Actor**  **Secondary Actor** | User | |
| **Trigger** | When user wants to comment on post | |
| **Typical Course of Events** | | |
| **Actor Action** | | **System Response** |
| 1. This use case begins when user want to comment on viewing posts. | |  |
| 1. The user enter comment on comment box below of post. | | The system generates commentID, save the comment and display the comment |
| Alternative Course of Events | | |
| 1. If user want to edit their comment, click edit button on comment and update comment, then System overwrites previous comment and display updated comment.   3a. If user want to delete their comment, user set delete on system, then system remove the comment. | | |

|  |  |  |
| --- | --- | --- |
| Use Case 8 | Recommend / Not recommend | |
| **Goal in Context** | To show the number of recommend and not recommend | |
| **Primary Actor**  **Secondary Actor** | User | |
| **Trigger** | When user wants to rate the post | |
| **Typical Course of Events** | | |
| **Actor Action** | | **System Response** |
| 1. This use case begins when user want to rate the post whether it’s recommendable or not. | |  |
| 1. The user click recommend / not-recommend button on bottom of post. | | The system calculates the overall number of recommend and not recommend, and display it. |
| Alternative Course of Events | | |
| 1. If user don’t want to recommend or not recommend the post anymore, click recommend or not recommend button one more time to cancel previous choice.   3a. If user already choose one of the option (recommend / not recommend),  Then the opposite option set as null. | | |
| Use Case 9 | Report | |
| **Goal in Context** | To report any inappropriate existing post, user or comments | |
| **Primary Actor**  **Secondary Actor** | User | |
| **Trigger** | When the user wants to report to the admin of any unusual content | |
| **Typical Course of Events** | | |
| **Actor Action** | | **System Response** |
| 1. This use case begins when a user want to report any malicious or inappropriate content/users | |  |
| 1. The user report on post or comment | | reportType is displayed |
| 1. The user chooses the type and enters comment. | | The report status set as “pending”, reportID is generated and date set to the system date. |
| Alternative Course of Events | | |
| 2a. if the user report on user, the system displays reportType and user can  choose the type and enter comment. The report status set as “pending”,  reportID is generated and date set to system date. | | |

|  |  |  |
| --- | --- | --- |
| Use Case 10 | Check report | |
| **Goal in Context** | To admin to check the report from users | |
| **Primary Actor**  **Secondary Actor** | Admin | |
| **Trigger** | When admin wants to check the reports | |
| **Typical Course of Events** | | |
| **Actor Action** | | **System Response** |
| 1. This use case begins when admin want to check reports from users. | |  |
| 1. The admin check report and distinguish reports which report to solve and which report to ignore. | | The system change the report status from ‘Pending’ to ‘Ignored’ for those reports that admin distinguished to ignore. |
| 1. The admin take right action to solve problems | | The system update or remove posts, comments, and ben users as admin controls |
| 1. After admin completed the solve problems, set the status of report as ‘Completed’ | | The system change the report status from ‘Pending’ to ‘Completed' |
| Alternative Course of Events | | |
|  | | |

* 1. **Analysis Class Diagram**



1. **Supplement**

