



# **EX-SELL**

## **System Requirement Specification**

**Version 1.0 approved**

**Prepared by Zenith**

Yiu Hong Sum	U1621435G
Lim Kian Hock Bryan	U1620949L
Cheng Yang Zhen	U1521618K
Lim Yan Jun	U1622311B
Soong Jie Ming	U1521535B
Chen Zhenni	U1622603D
Chang Keat Lueng Aaron	U1622360F

**26<sup>th</sup> February 2018**

# Table of Contents

Table of Contents .....	ii
Revision History .....	iv
<b>1. Problem Statement .....</b>	<b>1</b>
<b>2. Overview .....</b>	<b>2</b>
2.1 Background .....	2
2.2 Overall Description .....	2
<b>3. Investigation &amp; Analysis Methodology .....</b>	<b>3</b>
3.1 System Investigation .....	3
3.2 Analysis Methodology .....	3
<b>4. Constraints .....</b>	<b>17</b>
4.1 Hardware and Software Constraint – Server .....	17
4.2 Hardware and Software Constraint – Client .....	17
4.3 Project Schedule .....	17
<b>5. Operational Requirements .....</b>	<b>18</b>
5.1 Help Desk Support .....	18
5.2 Application Services and Technical support .....	18
5.3 Administration Features .....	18
5.4 System Hardware Fail Over and Back Up .....	18
5.5 Audit Trail .....	18
<b>6. Functional Requirements .....</b>	<b>19</b>
6.1 Login .....	19
6.2 Manage Account .....	19
6.3 Transactions .....	20
6.4 Product Listing .....	20
<b>7. Input Requirements .....</b>	<b>21</b>
7.1 NTU Email .....	21
<b>8. Output Requirements .....</b>	<b>21</b>
8.1 User Account .....	21
8.2 Transactions Records .....	21
8.3 List of registered accounts .....	21
<b>9. Process Requirements .....</b>	<b>22</b>
9.1 MySQL Transaction .....	22
9.2 Data Integrity .....	22
9.3 Data Validation .....	22
9.4 Security .....	22
9.5 Performance .....	22

9.6 Data Repository .....	22
<b>10.Non-Functional Requirements.....</b>	<b>23</b>
10.1 Performance .....	23
10.2 Stability and Security .....	23
10.3 Usability .....	23
<b>11.Hardware Requirements .....</b>	<b>24</b>
11.1 Network.....	24
11.2 Client Devices .....	24
11.3 Server .....	24
<b>12. Software Requirements.....</b>	<b>24</b>
12.1 Client Operating Systems .....	24
12.2 Client Applications .....	24
12.3 Network System.....	24
12.4 Mainframe Architecture.....	24
12.5 Licenses.....	25
<b>13.Deployment Requirements.....</b>	<b>25</b>

## Revision History

Name	Date	Reason For Changes	Version
Hong Sum	16/02/2018	Added Overall Description, Prototyping, Constraints	Version 0.1
Yan Jun	17/02/2018	Added System investigation	Version 0.2
Jie Ming	20/02/2018	Added Feasibility study and Requirement Elicitation, Input Requirement, Output Requirement	Version 0.3
Bryan	20/02/2018	Added Process Requirement, Hardware & Software Requirement, and Deployment Requirement	Version 0.4
Yang Zhen	20/02/2018	Added Functional Requirement & updated Problem Statement, Background	Version 0.5
Zhen Ni	22/02/2018	Added Use Case Description	Version 0.7
Aaron	23/02/2018	Updated Overall Description, Update Use Case	Version 0.8
Hong Sum	25/02/2018	Revised Document	Version 1.0

# **1. Problem Statement**

E-commerce has transformed the landscape of the world in the way things are done on a global scale. Never before have people been able to interact in such a cost-effective and efficient way. Currently, there exist many of e-commerce platforms such as Carousell, Shopee, and Taobao targeting audiences in a larger scale both either locally or internationally.

Problems such as the ease of transaction for the buying and selling of past and current university resources, such as textbooks and previously used hall items, are some problems that NTU students commonly face. University modules often requires students to buy textbooks, which are often expensive and unused after a semester ends.

Presently, there exists no such application or website in Singapore that allow both students and staffs to make sales transactions within the NTU community itself. NTU provides a marketplace for in iNTU for students to buy old computer or computer parts that the school is no longer using. However, it does not provide a platform for students to make transactions among themselves.

An alternative application existed in the form of Carousell Campus Marketplaces, where users can join a closed group to buy/sell items. However, there are many limitations such as:

- Campus Group is not widely advertised
- There is no category filtering within the group
- Graduates are still able access the group

We believe that this idea could be developed further and better integrated to bring greater convenience to the users by allowing them to buy and sell and trade university resources via our E-commerce platform.

## **2. Overview**

### **2.1 Background**

With the rising dominance of e-commerce, retailing is undergoing a seismic shift. With e-commerce such as Alibaba, Shoppe, Amazon, etc, E-commerce is becoming the preferred shopping method for many people. This convenience for users, in turn, brings about an easy and convenient way for people to buy products at the comfort of their own home. Internet sale are increasing rapidly as consumers take advantage of lower prices offered by vendors operating with less margin.

Ex-Sell is envisioned, to integrate and provide a convenient online shop where everyone is able buy and sell product from an organisation/school. Ex-sell aims to provide a platform where students and staff can sell past and current university resources, such as textbooks and previously used hall items. Ex-sell aims to provide a platform for students who are interested in Entrepreneurship to start off a small business using the platform we provide. Ex-sell aims to provide a way for students and staff make new friends through face to face meet up session.

With these considerations in mind, Ex-Sell with be developed to accommodate every user and provide a convenient platform for them to access. In addition, Ex-sell can be moved further onward by providing the same service to different organisation and schools through different domains.

### **2.2 Overall Description**

The Web platform will be developed in ASP.NET. Our targeted users will be the students, staff, and graduates from NTU allowing them to sell past and current university resources, such as textbooks and previously used hall items.

## **3. Investigation & Analysis Methodology**

### **3.1 System Investigation**

The Ex-Sell system is based on the idea of providing ease for resale of hall items and books within campus. As such, Ex-Sell will take each user based on their unique matric number and campus email, and tagged to each transaction, where it will be unique by its own rights. Recorded transactions will be stored in JSON files, and a successful entry will return an indicator to allow users to know that their transaction has been recorded.

Users are also able to create a new post to sell their items, view list of items on sale, sort items based on category, view item's details. One unique feature of the system is every user in the system must be part of the campus community, as each of them must be registered using the campus email, which is to fulfil our objective of providing easy transaction within campus. All this is done via verification of registration email with the school database.

### **3.2 Analysis Methodology**

Before any actual development takes place, feasibility studies and requirements elicitation must be done as part of our requirements specification. The requirements elicited will be reviewed and validated to ensure consistency and correctness. After solidifying the requirements, an object-oriented approach will be taken to analyze and break down the problem as well as listing down the scope of the project and any limitations or challenges that might potentially occur. Afterwards, the requirements will be formalized through Use Case Diagrams and its corresponding description, following which development on the prototype will start proper.

#### **3.2.1 Feasibility study and requirements elicitation**

The feasibility study will be done in multiple ways. Firstly, interviews will be conducted with students of NTU. The aim is to procure and compile requirements that individuals deem important for the project, after which the list will be grouped according to similarities before the retrieving the actual requirements.

We also plan to glean insights from similar systems that were already implemented in the market such as shopee and carousell. Studies conducted on these will provide us with critical information such as the type of challenges that we may face when implementing our system.

### 3.2.2 System analysis and requirements specification

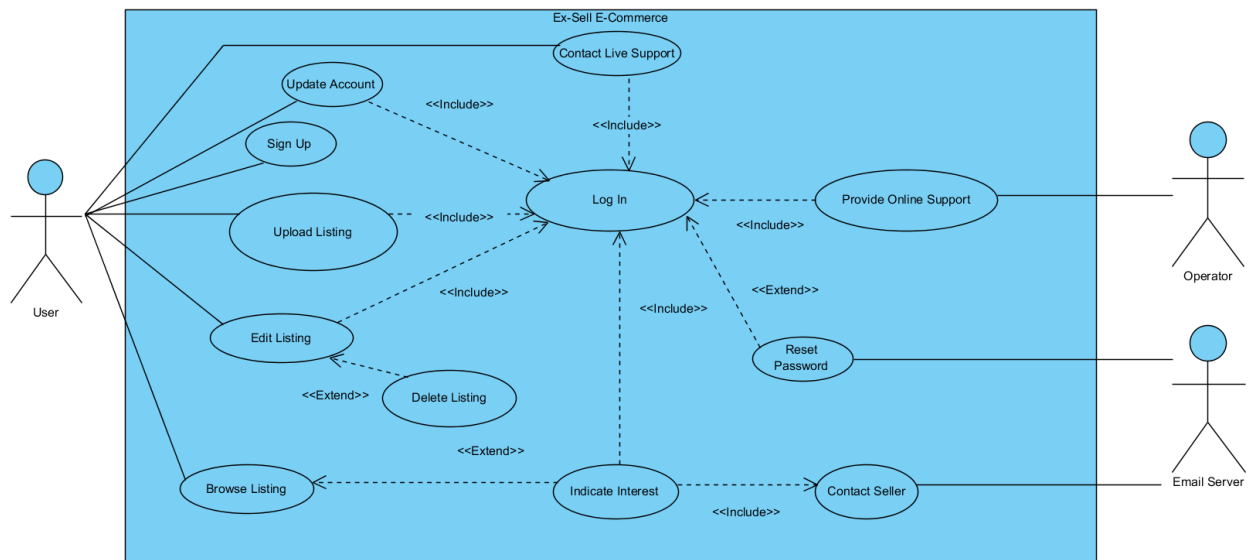
#### 3.2.2.1 Perform an analysis of the problem using object-oriented techniques

Once the requirements have been properly elicited, an external view of the enterprise model will be formed via Unified Modelling Language (UML), and some of the desired features of the new system includes:

- The ability to create and view items
- The ability to update item on sale post
- Be able to view transaction or history
- Be able to comment on items
- Be able to communicate with buyer regarding items on sale

### 3.2.3 Object-oriented design using UML

#### 3.2.3.1 Use Case Diagram





## 3.2.3.2 Use Case Descriptions

<b>Use Case ID:</b>	ES001		
<b>Use Case Name:</b>	Login		
<b>Created By:</b>	Bryan	<b>Last Updated By:</b>	Yiu Hong Sum
<b>Date Created:</b>	4 <sup>th</sup> February 2018	<b>Date Last Updated:</b>	7 <sup>th</sup> February 2018

<b>Actor:</b>	User, Administrator
<b>Description:</b>	This use case allows the Actor to log in to their accounts.
<b>Preconditions:</b>	<ol style="list-style-type: none"> <li>1. The account must already exist in the database.</li> <li>2. This use case starts when Actor selects Login</li> </ol>
<b>Postconditions:</b>	<ol style="list-style-type: none"> <li>1. The Actor is logged into the System</li> </ol>
<b>Priority:</b>	Medium
<b>Frequency of Use:</b>	Frequently
<b>Normal Flow:</b>	<ol style="list-style-type: none"> <li>1.1. Actor select login</li> <li>1.2. System prompt actor for login credentials</li> <li>1.3. Actor enters credentials</li> <li>1.4. Actor submit credentials to the system</li> <li>1.5. System validate Actor credential</li> <li>1.6. System verify the credentials with database</li> <li>1.7. System authenticates the Actor to login successfully.</li> </ol>
<b>Alternative Flows:</b>	<p>1.A1: System detects empty email address or password fields.</p> <ol style="list-style-type: none"> <li>1. System displays error message "Email address or password fields cannot be empty."</li> <li>2. Actor filled up the required field(s) for email address and password.</li> <li>3. Actor select the login button to re-attempt login again.</li> <li>4. Return to 1.2.</li> </ol> <p>1.A2: Actor enters the wrong credentials</p> <ol style="list-style-type: none"> <li>1. System displays error message "Invalid email address or password. Please re-enter."</li> <li>2. Return to 1.2.</li> </ol>
<b>Exception:</b>	-
<b>Includes:</b>	-
<b>Special Requirements:</b>	-
<b>Assumptions:</b>	-
<b>Notes and Issues:</b>	-

<b>Use Case ID:</b>	ES002		
<b>Use Case Name:</b>	Sign Up		
<b>Created By:</b>	Bryan	<b>Last Updated By:</b>	Yiu Hong Sum
<b>Date Created:</b>	4 <sup>th</sup> February 2018	<b>Date Last Updated:</b>	7 <sup>th</sup> February 2018

<b>Actor:</b>	User, Email Server
<b>Description:</b>	This use case allows the Actor to create their account.
<b>Preconditions:</b>	<ol style="list-style-type: none"> <li>1. Account must not already exist in the database.</li> <li>2. This use case starts when Actor selects Sign Up</li> </ol>
<b>Postconditions:</b>	<ol style="list-style-type: none"> <li>1. Account is created</li> </ol>
<b>Priority:</b>	Medium
<b>Frequency of Use:</b>	Frequently
<b>Normal Flow:</b>	<ol style="list-style-type: none"> <li>2.1. Actor select Sign Up</li> <li>2.2. System prompt actor for valid email address, first name, last name, password and confirm password</li> <li>2.3. Actor enter required input</li> <li>2.4. Actor submit input</li> <li>2.5. System validates the input</li> <li>2.6. System send verification code to the registered email address.</li> <li>2.7. Actor enter verification code</li> <li>2.8. System verify the input</li> <li>2.9. System create account in database and notify Actor</li> </ol>
<b>Alternative Flows:</b>	<p>2.A1: System detects mismatch between the password and confirm password.</p> <ol style="list-style-type: none"> <li>1. System displays error message "Password and Confirm Password mismatch. Please re-enter."</li> <li>2. Return to 2.1.</li> </ol> <p>2.A2: System detects the user account is already existing in the database.</p> <ol style="list-style-type: none"> <li>1. System will display error message "Account already exists."</li> <li>2. User will enter a new email address.</li> <li>3. User select the register button to re-attempt registration again.</li> <li>4. Return to 2.1.</li> </ol>
<b>Exceptions:</b>	-

<b>Includes:</b>	-
<b>Special Requirements:</b>	-
<b>Assumptions:</b>	-
<b>Notes and Issues:</b>	-

<b>Use Case ID:</b>	ES003		
<b>Use Case Name:</b>	Reset Password		
<b>Created By:</b>	Bryan	<b>Last Updated By:</b>	Yiu Hong Sum
<b>Date Created:</b>	4 <sup>th</sup> February 2018	<b>Date Last Updated:</b>	7 <sup>th</sup> February 2018

<b>Actor:</b>	User, Email Server
<b>Description:</b>	This use case allows Actor to reset their passwords
<b>Preconditions:</b>	<ol style="list-style-type: none"> <li>1. Account exists in the database.</li> <li>2. This use case starts when user select reset password</li> <li>3. This use case extends Log In (ES001)</li> </ol>
<b>Postconditions:</b>	<ol style="list-style-type: none"> <li>1. Password is reset</li> </ol>
<b>Priority:</b>	High
<b>Frequency of Use:</b>	Occasionally
<b>Flow of Events:</b>	<ol style="list-style-type: none"> <li>3.1. Actor select Reset Password</li> <li>3.2. System prompt actor for registered email address</li> <li>3.3. Actor enter their email address</li> <li>3.4. Actor submit input</li> <li>3.5. System validate input</li> <li>3.6. System verify the email address</li> <li>3.7. System generate and update new password into database</li> <li>3.8. System sends the new password to the Actor's registered email address.</li> <li>3.9. System notify user that password has been reset</li> </ol>
<b>Alternative Flows:</b>	<p>3.A1: The provided email address is not found in the database.</p> <ol style="list-style-type: none"> <li>1. System will display a message, "Invalid email entered".</li> <li>2. Return to 3.1.</li> </ol> <p>3.A2: The entered code is invalid.</p> <ol style="list-style-type: none"> <li>1. System will display a message, "Invalid verification code entered".</li> <li>2. Return to 3.4.</li> </ol> <p>3.A3: There is a mismatch between the two passwords.</p> <ol style="list-style-type: none"> <li>1. System will display a message, "Mismatch between the two passwords".</li> </ol>

	2. Return to 3.6.
<b>Exceptions:</b>	-
<b>Includes:</b>	Log In
<b>Special Requirements:</b>	-
<b>Assumptions:</b>	-
<b>Notes and Issues:</b>	-

<b>Use Case ID:</b>	ES004		
<b>Use Case Name:</b>	Delete Listing		
<b>Created By:</b>	Yiu Hong Sum	<b>Last Updated By:</b>	Yiu Hong Sum
<b>Date Created:</b>	8 <sup>th</sup> February 2018	<b>Date Last Updated:</b>	19 <sup>th</sup> March 2018

<b>Actor:</b>	User
<b>Description:</b>	To allow Actor is delete a post.
<b>Preconditions:</b>	<ol style="list-style-type: none"> <li>1. Actor must be logged in.</li> <li>2. This use case extends Edit Listing Product (ES007)</li> <li>3. This use case starts when user select Delete Post</li> </ol>
<b>Postconditions:</b>	<ol style="list-style-type: none"> <li>1. Post is deleted successfully.</li> </ol>
<b>Priority:</b>	Mid
<b>Frequency of Use:</b>	Occasionally
<b>Flow of Events:</b>	<ol style="list-style-type: none"> <li>12.1. Actor selects "Delete Post"</li> <li>12.2. System prompts for reason.</li> <li>12.3. Actor inserts reason.</li> <li>12.4. Actor selects "Submit" button.</li> <li>12.5. System prompts for confirmation.</li> <li>12.6. Actor selects "Confirm".</li> <li>12.7. System sends reason to owner of post.</li> <li>12.8. System displays "Post deleted successfully".</li> </ol>
<b>Alternative Flows:</b>	4.A1: The Actor selects Cancel. <ol style="list-style-type: none"> <li>1. Return to 3.2.</li> </ol>
<b>Exceptions:</b>	-
<b>Includes:</b>	Browse Product
<b>Special Requirements:</b>	-
<b>Assumptions:</b>	-
<b>Notes and Issues:</b>	-

<b>Use Case ID:</b>	ES005		
<b>Use Case Name:</b>	Upload Listing		
<b>Created By:</b>	Yiu Hong Sum	<b>Last Updated By:</b>	Yiu Hong Sum
<b>Date Created:</b>	7 <sup>th</sup> February 2018	<b>Date Last Updated:</b>	19 <sup>th</sup> March 2018

<b>Actor:</b>	User
<b>Description:</b>	This use case allows Actor to upload product
<b>Preconditions:</b>	<ol style="list-style-type: none"> <li>1. Actor must be logged in</li> <li>2. This use case includes Log In (ES001)</li> <li>3. This use case starts when user select Upload Product</li> </ol>
<b>Postconditions:</b>	<ol style="list-style-type: none"> <li>1. Product is posted</li> </ol>
<b>Priority:</b>	High
<b>Frequency of Use:</b>	Frequently
<b>Flow of Events:</b>	<ol style="list-style-type: none"> <li>5.1 Actor select Upload Product</li> <li>5.2 System prompt Actor for product details</li> <li>5.3 Actor input product details</li> <li>5.4 Actor submit product details</li> <li>5.5 System store product details in database</li> <li>5.6 System display product</li> </ol>
<b>Alternative Flows:</b>	<ol style="list-style-type: none"> <li>5. A1 Actor did not fill up all the inputs <ol style="list-style-type: none"> <li>1. System prompt Actor to fill up all required inputs.</li> <li>2. Return to 5.2</li> </ol> </li> </ol>
<b>Exceptions:</b>	-
<b>Includes:</b>	Log In
<b>Special Requirements:</b>	-
<b>Assumptions:</b>	-
<b>Notes and Issues:</b>	-

<b>Use Case ID:</b>	ES006		
<b>Use Case Name:</b>	Provide Online Support		
<b>Created By:</b>	Yiu Hong Sum	<b>Last Updated By:</b>	Yiu Hong Sum
<b>Date Created:</b>	7 <sup>th</sup> February 2018	<b>Date Last Updated:</b>	19 <sup>th</sup> March 2018

<b>Actor:</b>	Operator
<b>Description:</b>	This use case allows Actor to provide online support to user
<b>Preconditions:</b>	<ol style="list-style-type: none"> <li>1. Actor must be logged in</li> <li>2. Actor must be an operator</li> </ol>
<b>Postconditions:</b>	<ol style="list-style-type: none"> <li>1. Live Support is provided for user</li> </ol>
<b>Priority:</b>	High
<b>Frequency of Use:</b>	Occasionally
<b>Flow of Events:</b>	<ol style="list-style-type: none"> <li>6.1 Actor Select "Operator Login"</li> <li>6.2 System submit Actor input for verification</li> <li>6.3 System display successful log in</li> <li>6.4 Actor provide online support upon request</li> </ol>
<b>Alternative Flows:</b>	-
<b>Exceptions:</b>	-
<b>Includes:</b>	-
<b>Special Requirements:</b>	-
<b>Assumptions:</b>	-
<b>Notes and Issues:</b>	-

<b>Use Case ID:</b>	ES007		
<b>Use Case Name:</b>	Edit Listing		
<b>Created By:</b>	Yiu Hong Sum	<b>Last Updated By:</b>	Yiu Hong Sum
<b>Date Created:</b>	7 <sup>th</sup> February 2018	<b>Date Last Updated:</b>	19 <sup>th</sup> March 2018

<b>Actor:</b>	User
<b>Description:</b>	This use case allows Actor to edit product
<b>Preconditions:</b>	<ol style="list-style-type: none"> <li>1. Actor must be logged in</li> <li>2. This use case extends Upload Listing (ES005)</li> <li>3. This use case starts when user select Upload Listing</li> </ol>
<b>Postconditions:</b>	<ol style="list-style-type: none"> <li>1. Product details is updated</li> </ol>
<b>Priority:</b>	High
<b>Frequency of Use:</b>	Frequently
<b>Flow of Events:</b>	<ol style="list-style-type: none"> <li>7.1 Actor select Edit Listing</li> <li>7.2 Actor select product to edit</li> <li>7.3 System retrieve product details from database</li> <li>7.4 System prompt Actor for product details</li> <li>7.5 Actor input update product details</li> <li>7.6 Actor submit product details</li> <li>7.7 System store product details in database</li> <li>7.8 System display updated product details</li> </ol>
<b>Alternative Flows:</b>	<ol style="list-style-type: none"> <li>7. A1 Actor did not fill up all the inputs <ol style="list-style-type: none"> <li>1. System prompt Actor to fill up all required inputs.</li> <li>2. Return to 7.2</li> </ol> </li> </ol>
<b>Exceptions:</b>	-
<b>Includes:</b>	-
<b>Special Requirements:</b>	-
<b>Assumptions:</b>	-
<b>Notes and Issues:</b>	-

<b>Use Case ID:</b>	ES008		
<b>Use Case Name:</b>	Browse Listing		
<b>Created By:</b>	Aaron	<b>Last Updated By:</b>	Yiu Hong Sum
<b>Date Created:</b>	7 <sup>th</sup> February 2018	<b>Date Last Updated:</b>	19 <sup>th</sup> March 2018

<b>Actor:</b>	User
<b>Description:</b>	This use case allows Actor to search for an item by category and name
<b>Preconditions:</b>	<ol style="list-style-type: none"> <li>1. Actor must be logged in</li> <li>2. This use case includes Log In (ES001)</li> <li>3. This use case starts when user select Browse Listing</li> </ol>
<b>Postconditions:</b>	<ol style="list-style-type: none"> <li>1. Product is posted</li> </ol>
<b>Priority:</b>	High
<b>Frequency of Use:</b>	Frequent
<b>Flow of Events:</b>	<ol style="list-style-type: none"> <li>8.1 Actor select Browse Product</li> <li>8.2 Actor selects the category</li> <li>8.3 System display product of all category</li> <li>8.4 Actor selects the category of choice</li> <li>8.5 System display list of products</li> </ol>
<b>Alternative Flows:</b>	<p>8.A1. Category of choice is unavailable</p> <ol style="list-style-type: none"> <li>1. Actor will select the search box</li> <li>2. Search box comes into focus</li> <li>3. Select types in desired category</li> <li>4. System will match category based on descriptions of item</li> <li>5. Go to 8.4</li> </ol>
<b>Exceptions:</b>	-
<b>Includes:</b>	-
<b>Special Requirements:</b>	-
<b>Assumptions:</b>	-
<b>Notes and Issues:</b>	-



<b>Use Case ID:</b>	ES009		
<b>Use Case Name:</b>	Indicate Interest		
<b>Created By:</b>	Aaron	<b>Last Updated By:</b>	Yiu Hong Sum
<b>Date Created:</b>	7 <sup>th</sup> February 2018	<b>Date Last Updated:</b>	19 <sup>th</sup> March 2018

<b>Actor:</b>	User
<b>Description:</b>	Allow Actor to indicate interest for a listed item from a seller
<b>Preconditions:</b>	<ol style="list-style-type: none"> <li>1. Actor must browse a product</li> <li>2. This use case includes Browse Listing (ES008)</li> <li>3. This use case includes Contact Seller (ES010)</li> <li>4. This use case starts when user select Indicate Interest</li> </ol>
<b>Postconditions:</b>	<ol style="list-style-type: none"> <li>1. Actor bought a product</li> </ol>
<b>Priority:</b>	High
<b>Frequency of Use:</b>	Frequent
<b>Flow of Events:</b>	<ol style="list-style-type: none"> <li>9.1 Actor select Indicate Interest</li> <li>9.2 System prompt user for confirmation</li> <li>9.3 Actor confirms the prompt</li> <li>9.4 System update database of new update</li> <li>9.5 System display success message</li> </ol>
<b>Alternative Flows:</b>	-
<b>Exceptions:</b>	<ol style="list-style-type: none"> <li>9.1.E.1. Item has been removed from the listing <ol style="list-style-type: none"> <li>1. System will bring the Actor back to the page that the user came from</li> <li>2. System will display a message "The item you were looking for has been removed from the store"</li> </ol> </li> <li>9.1.E.2. Item has been bought by another user <ol style="list-style-type: none"> <li>1. System will update the "Buy" button to "Sold"</li> </ol> </li> </ol>
<b>Includes:</b>	Browse product, Contact seller
<b>Special Requirements:</b>	-
<b>Assumptions:</b>	-
<b>Notes and Issues:</b>	-

<b>Use Case ID:</b>	ES010		
<b>Use Case Name:</b>	Contact Live Support		
<b>Created By:</b>	Aaron	<b>Last Updated By:</b>	Bryan Lim
<b>Date Created:</b>	8 <sup>th</sup> February 2018	<b>Date Last Updated:</b>	19 <sup>th</sup> March 2018

<b>Actor:</b>	User
<b>Description:</b>	This use case allows Actor to contact live support operator
<b>Preconditions:</b>	1. Actor must be logged in
<b>Postconditions:</b>	1. Actor contacted live support operator
<b>Priority:</b>	Medium
<b>Frequency of Use:</b>	Occasionally
<b>Flow of Events:</b>	10.1. Actor select Live Support 10.2. System create communication page for Actor and Operator 10.3. System display communication page for the Actor and Operator
<b>Alternative Flows:</b>	-
<b>Exceptions:</b>	-
<b>Includes:</b>	-
<b>Special Requirements:</b>	-
<b>Assumptions:</b>	-
<b>Notes and Issues:</b>	-

<b>Use Case ID:</b>	ES011		
<b>Use Case Name:</b>	Update Account		
<b>Created By:</b>	Zhenni	<b>Last Updated By:</b>	Zhenni
<b>Date Created:</b>	8 <sup>th</sup> February 2018	<b>Date Last Updated:</b>	8 <sup>th</sup> February 2018

<b>Actor:</b>	User
<b>Description:</b>	To allow Actor to update their account information respectively.
<b>Preconditions:</b>	<ol style="list-style-type: none"> <li>1. Actor must be logged in.</li> <li>2. This use case includes Log In (ES001)</li> <li>3. This use case starts when user select Edit Account</li> </ol>
<b>Postconditions:</b>	<ol style="list-style-type: none"> <li>1. Actor's account information is updated.</li> </ol>
<b>Priority:</b>	Mid
<b>Frequency of Use:</b>	Occasionally
<b>Flow of Events:</b>	<ol style="list-style-type: none"> <li>11.1. Actor selects "Edit Account"</li> <li>11.2. System retrieves and displays all account details of Actor.</li> <li>11.3. Actor edits relevant information.</li> <li>11.4. Actor selects "Submit" button.</li> <li>11.5. System prompts for confirmation.</li> <li>11.6. Actor selects "OK".</li> <li>11.7. System updates database with new account information.</li> <li>11.8. System displays "Account Information Updated Successfully".</li> </ol>
<b>Alternative Flows:</b>	11.A1: The Actor selects Cancel. <ol style="list-style-type: none"> <li>1. Return to 3.3.</li> </ol>
<b>Exceptions:</b>	-
<b>Includes:</b>	Log In
<b>Special Requirements:</b>	-
<b>Assumptions:</b>	-
<b>Notes and Issues:</b>	-

<b>Use Case ID:</b>	ES012		
<b>Use Case Name:</b>	Contact Seller		
<b>Created By:</b>	Zhenni	<b>Last Updated By:</b>	Yiu Hong Sum
<b>Date Created:</b>	8 <sup>th</sup> February 2018	<b>Date Last Updated:</b>	19 <sup>th</sup> March 2018

<b>Actor:</b>	User, Email Server
<b>Description:</b>	This use case allows actor to Contact Seller
<b>Preconditions:</b>	4. Actor must be logged in. 5. This use case includes Indicate Interest (ES009) 6. This use case starts when user select Contact Seller
<b>Postconditions:</b>	1. User successfully contacted with seller
<b>Priority:</b>	Mid
<b>Frequency of Use:</b>	Occasionally
<b>Flow of Events:</b>	12.1. Actor selects "Contact Seller" 12.2. System retrieve seller contact from database 12.3. System display Actor contact number
<b>Alternative Flows:</b>	
<b>Exceptions:</b>	-
<b>Includes:</b>	Indicate Interest
<b>Special Requirements:</b>	-
<b>Assumptions:</b>	-
<b>Notes and Issues:</b>	-

### 3.2.4 Prototyping

The Object Oriented Rapid Prototyping (OORP) method will be used to implement prototype for Ex-Sell E-Commerce. The prototype will be a working example of a part of the system for demonstration and proof of concept. Once the actual requirements are understood, the prototype is discarded, and the actual system is developed with a much clear understanding of user requirements.

## **4. Constraints**

### **4.1 Hardware and Software Constraint – Server**

Ex-Sell require a dedicated Internet Service Provider with a configuration that is able to handle up to 100,000 visits per day. Uninterruptible Power Supply (UPS) is required to keep the server running in cases where a blackout or power trip happen. Air-Conditioning is required to keep the server running smoothly as overheating could cause critical information lost, downtime, and having to replace equipment due to hardware failure.

Likewise, as we require our user to register based on campus email, we would require authentication with the campus database.

### **4.2 Hardware and Software Constraint – Client**

Clients are required to have internet connection at all time. Clients are required to have an account to access all features. Clients are required to have an internet accessible device such as smartphone or computer to use Ex-Sell. Clients are required to have a personal contact number such that communications between clients can be made. Client must have some basic knowledge on using a computer and browsing websites.

### **4.3 Project Schedule**

There is a one-year timeframe to implement a production system of Ex-Sell from project commencement in time for its launch date at the end of year 2018.

## **5. Operational Requirements**

### **5.1 Help Desk Support**

Ex-sell users will have a 24x7 access to email and telephone assistance for any questions that may arise, such as, slow, or sluggish system response time, incompatible browser features, application errors, system downtime inquiries, account lock-out assistance, etc.

### **5.2 Application Services and Technical support**

Programmers and software developers will have access to source code to address bugs or system enhancements as deemed necessary. Network Administrator and Database Administrator support will also be required to maintain a 24x7 system uptime.

### **5.3 Administration Features**

Ex-sell implements different types of levels in system security and system access. Normal users do not have the rights to view other normal user's personal information such as name and contact number. Only authorized system administrator has the access to all user's personal information.

### **5.4 System Hardware Fail Over and Back Up**

Computer operations center will handle system hardware tasks such as data tape back-up, hardware maintenance, fail over, scheduled system patches and maintenance

### **5.5 Audit Trail**

System audit trails are inherent part of all student registrations. Among others, all transaction records will capture what action was taken, when (time-stamp) the transaction occurred and who made the transaction.

## **6. Functional Requirements**

### **6.1 Login**

6.1.1 New users should be able to create a new account on the webpage.

6.1.2 The NTU address, name, username, password, gender and home address of the user should be provided in order to create their account successfully.

6.1.2.1 The system must send a verification email that contains a verification link to the user, and account verification is done upon clicking the link provided in the email.

6.1.3 Users who have forgotten their password should be able to retrieve them

6.2.1.1 System should send a password reset email to the the user's NTU email address should the user selects the 'Forgot password?' tab.

6.1.4 Users will be prompted to register for an account if there is no pre-existing account present.

### **6.2 Manage Account**

#### **Settings Page**

6.2.1 Upon logging in, the user can either select the 'Your Profile' tab or their display icon to view their profile details.

6.2.1.1 Users should be able to edit their name, password, profile display picture, and home address on the profile page.

## **6.3 Transactions**

### **Buy Listing**

6.3.1 The buyer should be able purchase or enquire more about a product while it is selected.

6.3.4.1 While enquiring for a product, the buyer will enter a private chat channel with the buyer.

6.3.4.2 While selecting the purchase option, this will allow the system to inform the seller of the buyer's interest in the item.

6.3.4.3 The system should allow the seller to accept or decline the offer for the listing.

### **Sell Listing**

6.3.2 The seller should be able to upload a product's photo, and provide information about the listing based on name, description, and category.

6.3.3 The seller should be able to chat with the buyer who has indicated interest in the listings and be able to accept and decline the offer for the listing.

6.3.4 The system will send an email notification to the seller's NTU email to inform the seller of the interested listing.

6.3.4.1 The offer must be accepted within 48 hours or it will be deemed as cancelled.

## **6.4 Product Listing**

### **View Product Listings**

6.4.1 Upon logging in, the user can either select the 'Your Profile' tab or their display icon to edit their profile details.

6.4.2 Users should be able to view, edit and delete their current listings on the profile page.

6.4.3 System should be able to automatically remove item listings which have been marked as sold.

6.4.4 Profile page should have a tab which allows users to track the items which they have marked under 'My Interested Listings'.

6.4.4.1 Upon clicking the 'My Interested Listings' tab, user should be provided a list view of the products which they have bookmarked previously by the user.



## **Update Listings**

6.4.5 User should be able to delete or edit their listings on the profile page.

6.4.5.1 For the 'Edit Listings' category, user should be able to modify the description, name, image uploaded, and condition of the item.

6.4.5.2 User should be able to finalise edits made on the item and an automatic update should be shown.

## **7. Input Requirements**

### **7.1 NTU Email**

All academic students and staff should have their personal NTU email ready upon admission to the University. Users must understand that the email will be crucial to authentication when using ex-sell. The email access will be removed at the end of their tenure in NTU.

## **8. Output Requirements**

### **8.1 User Account**

All accounts created should contain the user's personal information. It should allow the user to view, post or buy products listed online. The user should be able to view and edit any personal information that they had previously listed in their profile.

### **8.2 Transactions Records**

Every account should be allowed to view a summary of all transactions made since the creation of the account. The record should contain information on the product name, timestamp and price.

### **8.3 List of registered accounts**

Accounts with administrator privileges should be entitled to view a report on the list of accounts created since the implementation of the system.

## **9. Process Requirements**

### **9.1 MySQL Transaction**

1. The system must be able to send, trigger and receive updates from the database.

### **9.2 Data Integrity**

1. All data passed to the database must be accurate and consistent.
2. The system must be able to verify the integrity of the data sent and received from the database.
3. The system must be able to only commit transactions that are completed and any unfinished or time-out transactions must not be updated to database.

### **9.3 Data Validation**

1. The system must be able to validate the inputs given to the database.
2. The system must be able to prevent data error from the frontend and backend.
3. The system must be able to perform data validation and error-handling routines in the various layers of the system architecture.

### **9.4 Security**

1. The system must be able to prevent SQL injection into the system database.

### **9.5 Performance**

1. The system must be able to handle concurrent use of the server on a 24/7 basis.
2. The system must be able to send, receive and display relevant information to improve the overall user experience.

### **9.6 Data Repository**

1. The system must maintain the existing MySQL database as the main repository of data.

## **10. Non-Functional Requirements**

### **10.1 Performance**

1. When the user accesses the website, it must load in less than 10 seconds.
2. When the user clicks on any of the link in the website, the system should respond and produce the link in less than 3 seconds.
3. Listings must be updated periodically after every listing addition, modification, and deletion.

### **10.2 Stability and Security**

1. Ex-Sell will require login information and user authentication to access the web page.
2. The username should belong to an NTU email domain
3. The password should contain 8-16 alphanumeric characters with at least one uppercase and lowercase character
4. The Database and Webpage package files will be backed up at the end of each week (Friday) to be able to fully restore the system.
5. The system database will be backed up every day at 11pm.
6. External monitoring of website every 10 minutes using third party software such as GotSiteMonitor to detect downtime.

### **10.3 Usability**

1. Listings will be displayed with the most recent result by default.
2. Advanced search filters such as category filtering (Clothes and Shoes, Used textbooks, etc) should be present as well.
3. Category headers, links of any kind should only contain at most three words for simplicity.
4. Making use of familiar icons such as a “star” icon to denote the marking a buyer’s interested listing. 90% of the users should intuitively use those buttons to aid them when they are selecting products.

## **11. Hardware Requirements**

### **11.1 Network**

1. Mobile users must have a cellular or Wifi connection.
2. Desktop users must have an ethernet or Wifi connection.

### **11.2 Client Devices**

1. Client computers must be running Mac, Unix or Windows.
2. Client mobile devices such as smartphones or tablets must be running iOS or Android or Windows.

### **11.3 Server**

1. 24-hour uptime server with a backend database.
2. Digital Ocean platform that will be hosting the Windows web-server.

## **12. Software Requirements**

### **12.1 Client Operating Systems**

1. Any operating systems that supports web browsing: MacOS, Windows, Android, etc

### **12.2 Client Applications**

1. Any web browsers that are compatible with HTML5 and Javascript: IE, Chrome, FireFox, etc

### **12.3 Network System**

1. Network software and protocols for system communication: TCP/IP, HTTP, FTP, SFTP, HTTPS, SSH, etc

### **12.4 Mainframe Architecture**

1. Windows machine running MySQL server version 14.14 & IIS v7.5.

## 12.5 Licenses

Valid licenses are required to run software from third-party vendors:

1. To use application development tools
2. To use web server, application server and database software in development, testing and production

## 13. Deployment Requirements

