# 1.Team Members

Keran Wang (keranw 686976)

Xue Jiang (jiangx2 665401)

# 2.Application Overview

## 2.1 Description

## 2.2 Enterprise System Properties

### 2.2.1 Persistent Data

### 2.2.2 A lot of Data

### 2.2.3 Concurrent Access

### 2.2.4 Many UI Screens

### 2.2.5 Complex Business Logic

# 3.Feature Specification

## 3.1 Description

One of the features we want to implement is item management. If a user has registered as a seller, he or she should be able to submit information about the goods he or she wants to sell to the system.

### 3.1.1 Create

There will be a table called items to save the information about the items. As a seller publish a new item on the system, the system will create a new record and save it in the database.

### 3.1.2 Retrieve

If the seller wants to check what items he or she has put in the system, the seller should be able collect the information of the items belong to the user from the database, and show the result on the page.

### 3.1.3 Update

As the seller needs to update some information of an item, the seller should be able to save the change in the database

### 3.1.4 Delete

Once a seller decides not to sale an item he or she has put in the system any more, the seller should be able to delete that record.

## 3.2 Use Cases

### 3.2.1 ItemCreate

|  |  |
| --- | --- |
| ID | IM01 |
| Precondition | The user has already registered as a seller and logged in |
| Basic flow | 1. A user click “Add a new item” button on the management page  2. The system shows the form to collect necessary information  3. The user fills in the form and click submit button  4. The system saves the record in the database  5. The system returns to the management page |
| Alternative flow | 3a. The user abandons and clicks “Cancel” button  3a1. The system returns to the management page  4a. The system finds that critical information is missing or incorrect and refuse to save  4a1. The system stays on the information collecting page |
| Postcondition | A new record saved in the database |

### 3.2.2 ItemRetrieve

|  |  |
| --- | --- |
| ID | IM02 |
| Precondition | The user has logged in the system and created at least a piece of item record |
| Basic flow | 1. A user clicks the item exists in the list on the management page  2. The system shows the detail information about the item. |
| Alternative flow | None |
| Postcondition | None |

### 3.2.3 ItemUpdate

|  |  |
| --- | --- |
| ID | IM03 |
| Preconditioin | The user has logged in the system and created at least a piece of item record |
| Basic flow | 1. A user clicks the “Update” button of an item  2. The system shows the update page with the information of the item  3. The user update the information and clicks “Submit” button  4. The system saves the changes in the database  5. The system returns to the management page |
| Alternative flow | 3a. The user abandons the change and clicks “Cancel” button  3a1. The system returns to management page  4a. The system finds that the critical information is missing or incorrect and refuses to save the change.  4a1. The system stays on the information collecting page |
| Postcondition | None |

### 3.2.4 ItemDelete

|  |  |
| --- | --- |
| ID | IM04 |
| Precondition | The user has logged in the system and created at least a piece of item record |
| Basic flow | 1. A user clicks the “Delete” button of an item  2. The system pops up a confirm window  3. The user clicks “Yes” button  4. The system closes the pop-up and deletes that record from the database  5. The system refresh the management page |
| Alternative flow | 3a. The user clicks “No” button  3a1. The system closes the pop-up and returns to the management page |
| Postcondition | A record is deleted from the database |