Use Cases

for

<Spotter>

Version <1.1>

Prepared by <Jiatai Xie>

<BU CS 411>

<3/23/16>

Use Case List

|  |  |  |
| --- | --- | --- |
| ID | Primary Actor | Use Case Title |
| 001 | User | Initial Lost Item |
| 002 | User | Initial Found Item |
| 003 | User | Log In |
| 004 | Unregistered student | Registration |
| 005 | User | Home |
| 006 | User | Lost History |
| 007 | User | Found History |

Use Case #1

Initial Lost Item

|  |  |  |  |
| --- | --- | --- | --- |
| Use Case ID: | 001 | | |
| Use Case Name: | Initial Lost Item | | |
| Created By: | Jonathan Liu | Last Updated By: | Qi Wang |
| Date Created: | 9.28.15 | Date Last Updated: | 12. 12. 15 |

|  |  |
| --- | --- |
| Actors: | User |
| Description: | The user inputs information about such as location lost (on campus), lost item type, description of item (UID if Terrier Card). The system automatically generates the user contact information and post date based on user’s login status. The system then generates a lost item pointer based on these inputs and posts it on the map. |
| Trigger: | Click on “Do you lose something?” button on Home page |
| Preconditions: | 1. Student is a registered user of the system. 2. User identity is authenticated. |
| Postconditions: | 1. Entry is added to lost items collection 2. Marker is placed on map |
| Normal Flow: | *1.0 User navigates to the lost item page*   * 1. Page shows the map and lost form for student   2. User enters location of lost item and click “locate” button to pin a pointer on the map   3. User chooses the lost item type   4. User enters description of item   5. User clicks the “submit” button   6. System pops out the information and asks user for confirmation   7. User confirms   8. System inputs data into lost item database   9. System automatically goes to Home page which tells user already posts lost item. |
| Alternative Flows: | N/A |
| Exceptions: | *E1: Insufficient Information*  E1.1 User does not input the location of the item and not click “locate” button.  E1.2 When user clicks the “submit” button, system pops “You need to locate a location for your lost item.”  E1.3 After user clicks known button, system goes back to lost item page.  *E2: Invalid Information*  E2.1 User puts the location outside of Boston.  E2.2 When user clicks the “locate” button, system informs user that this page only works for Boston area.  E2.3 After user clicks known button, system goes back to lost item page. |
| Includes: | Home page |
| Priority: | High |
| Frequency of Use: | Varies based on user |
| Business Rules: | N/A |
| Special Requirements: | N/A |
| Assumptions: | N/A |
| Notes and Issues: | N/A |

Revision History

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Date** | **Reason For Changes** | **Version** |
| Jonathan Liu | 9.28.15 | Write up for Use Case #1 | 1.0 |
| Qi Wang | 12.12.15 | More accurate about flow and exception | 1.1 |

Use Case #2

Initial Found Item

|  |  |  |  |
| --- | --- | --- | --- |
| Use Case ID: | 002 | | |
| Use Case Name: | Initial Found Item | | |
| Created By: | Andrew Li | Last Updated By: | Qi Wang |
| Date Created: | 12/11/2015 | Date Last Updated: | 12/12/2015 |

|  |  |
| --- | --- |
| Actors: | User |
| Description: | The user inputs information about found items’ location (on campus), found item type, description of item (UID if Terrier Card). The system automatically generates the user contact information and post date based on user’s login status. The system then generates a found item pointer based on these inputs and posts it on the map. |
| Trigger: | Click on “Do you find something” button on Home page |
| Preconditions: | * + 1. User is registered     2. User’s identity is authenticated |
| Postconditions: | * + - 1. Entry is added to founditems collection       2. Marker is placed on map |
| Normal Flow: | *1.0 User navigates to the found item page*  1.1 Page shows the map and found form for student  1.2 User enters location of found item and click “locate” button to pin a pointer on the map   * 1. User chooses the found item type   2. User enters description of item   3. User clicks the “submit” button   4. System pops out the information and asks user for confirmation   5. User confirms   6. System inputs data into found item database   7. System automatically goes to Home page that informs user already posts found item. |
| Alternative Flows: | N/A |
| Exceptions: | *E1: Insufficient Information*  E1.1 User does not input the location of the item and not click “locate” button.  E1.2 When user clicks the “submit” button, system pops “You need to locate a location for your found item.”  E1.3 After user clicks known button, system goes back to found item page.  *E2: Invalid Information*  E2.1 User puts the location outside of Boston.  E2.2 When user clicks the “locate” button, system informs user that this page only works for Boston area.  E2.3 After user clicks known button, system goes back to found item page. |
| Includes: | Home page |
| Priority: | High |
| Frequency of Use: | Varying |
| Business Rules: | N/A |
| Special Requirements: | N/A |
| Assumptions: | N/A |
| Notes and Issues: | N/A |

Revision History

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Date** | **Reason For Changes** | **Version** |
| Andrew Li | 12.11.2015 | Initial Commit | 1.0 |
| Qi Wang | 12.12.2015 | More accurate about flow and exception | 1.1 |

Use Case #3

Log-in

|  |  |  |  |
| --- | --- | --- | --- |
| Use Case ID: | 003 | | |
| Use Case Name: | Log-in | | |
| Created By: | Qi Wang | Last Updated By: | Qi Wang |
| Date Created: | 12.12.15 | Date Last Updated: | 12. 12. 15 |

|  |  |
| --- | --- |
| Actors: | Anyone |
| Description: | Anyone can log in or create an account. |
| Trigger: | Go to the web page. |
| Preconditions: | N/A |
| Postconditions: | 1. User gets his or her records about lost and found information. |
| Normal Flow: | *1.0 Log in*  1.1 Student enters username.  1.2 Student enters passport.  1.3 Student clicks “Log in” button  1.4 System goes to home page, if the username and passport are correct and match. |
| Alternative Flows: | * 1. *Registration*   2. Student clicks “Create a new account”   3. System goes to registration page. |
| Exceptions: | *E1: Insufficient Information*   1. Student doesn’t input username or passport. 2. When student clicks the “Log in” button, system pops “You need to input username and passport.” 3. After student clicks known button, system goes back to log in page.   *E2: Invalid Information*   1. Student puts wrong username or passport. 2. System shows “User does not exist.” 3. After student clicks known button, system goes back to log in page. |
| Includes: | Registration page |
| Priority: | High |
| Frequency of Use: | Every time |
| Business Rules: | N/A |
| Special Requirements: | N/A |
| Assumptions: | N/A |
| Notes and Issues: | N/A |

Revision History

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Date** | **Reason For Changes** | **Version** |
| Qi Wang | 12.12.15 | Initial Use Case #3 | 1.0 |

Use Case #4

Registration

|  |  |  |  |
| --- | --- | --- | --- |
| Use Case ID: | 004 | | |
| Use Case Name: | Registration | | |
| Created By: | Conan Xiao | Last Updated By: | Qi Wang |
| Date Created: | 12.12.15 | Date Last Updated: | 12. 12. 15 |

|  |  |
| --- | --- |
| Actors: | Unregistered student |
| Description: | The unregistered student creates an account by inputting personal information: First name, last name, contact information/phone number and BUID to ensure that they are a BU student hence the name Terrier Spot It. (Site only applicable within the vicinity of the Boston University campus) |
| Trigger: | Click on “Create a new account” button from Log in page |
| Preconditions: | * + 1. Student is a BU student.     2. Student has a phone number. |
| Postconditions: | * + - 1. New user adds to user database. |
| Normal Flow: | 1. *Student creates an account*    1. Student enters in first and last name    2. Student enters in phone number    3. Student creates a username    4. Student creates a password    5. System verifies/confirms password    6. Student enters in BU email    7. Student is signed into homepage    8. Student can log out and log back in using the created username and password |
| Alternative Flows: | N/A |
| Exceptions: | E.1 *Invalid Information* E.2 System displays message “Invalid Password” when user types less in less than the minimum required amount of characters  E.3 User enters in a different email account instead of a BU email  E.4 If a user name is less than two characters or over (max #) of characters. |
| Includes: | N/A |
| Priority: | High |
| Frequency of Use: | Only in the beginning. |
| Business Rules: | N/A |
| Special Requirements: | N/A |
| Assumptions: | N/A |
| Notes and Issues: |  |

Revision History

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Date** | **Reason For Changes** | **Version** |
| Conan Xiao | 12.12.15 | Initial Use case #4 | 1.0 |
| Qi Wang | 12.12.15 | Small adjustment | 1.1 |

Use Case #5

Home

|  |  |  |  |
| --- | --- | --- | --- |
| Use Case ID: | 005 | | |
| Use Case Name: | Home | | |
| Created By: | Qi Wang | Last Updated By: | Qi Wang |
| Date Created: | 12.12.15 | Date Last Updated: | 12. 12. 15 |

|  |  |
| --- | --- |
| Actors: | User |
| Description: | This page is user’s main page. It contains links to all pages user needs: post lost item page, post found item, lost item map, found item map, user’s lost history page and user’s found history page. It also shows user’s information and indicates some behaviors from user. |
| Trigger: | * + - * 1. Click on “Log in” button on log in page after put the correct log in information.         2. Click on “Return Home” button from other pages.         3. Post lost or found item successfully.         4. Delete personal lost or found item history successfully. |
| Preconditions: | 1. Student is a registered user of the system. 2. User identity is authenticated. |
| Postconditions: | N/A |
| Normal Flow: | * 1. *User clicks on a specific link for some functionality*   1.1 System goes to the related page |
| Alternative Flows: | *1.0* *User returns to home page after some behaviors*  1.1 System shows certain result after user’s behavior. |
| Exceptions: | N/A |
| Includes: | * + 1. Initial Lost Item     2. Initial Found Item     3. Lost Item Map     4. Found Item Map     5. Lost History     6. Found History     7. Log out |
| Priority: | High |
| Frequency of Use: | Every time |
| Business Rules: | N/A |
| Special Requirements: | N/A |
| Assumptions: | N/A |
| Notes and Issues: | N/A |

Revision History

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Date** | **Reason For Changes** | **Version** |
| Qi Wang | 12.12.15 | Initial Use Case#5 | 1.0 |

Use Case #6

Lost History

|  |  |  |  |
| --- | --- | --- | --- |
| Use Case ID: | 006 | | |
| Use Case Name: | Lost History | | |
| Created By: | Qi Wang | Last Updated By: | Qi Wang |
| Date Created: | 12.12.15 | Date Last Updated: | 12. 12. 15 |

|  |  |
| --- | --- |
| Actors: | User |
| Description: | Shows all lost item that user has been posted, and user deletes own lost item that has been found. |
| Trigger: | Click on “Lost History” button on Home page |
| Preconditions: | 1. Student is a registered user of the system. 2. User identity is authenticated. |
| Postconditions: | * + 1. Remove item from lost items database.     2. Remove related pin from map. |
| Normal Flow: | *1.0 User navigates to the lost history page*  1.1 Page shows lost item list for user.  1.2 User clicks “Found It” button for item that is already been founded.  1.3 System moves the found item from lost item database.  1.4 System goes back to the Home page and tells user item has been removed from map. |
| Alternative Flows: | N/A |
| Exceptions: | N/A |
| Includes: | N/A (at the moment) |
| Priority: | High |
| Frequency of Use: | Varies based on user |
| Business Rules: | N/A |
| Special Requirements: | N/A |
| Assumptions: | N/A |
| Notes and Issues: | N/A |

Revision History

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Date** | **Reason For Changes** | **Version** |
| Qi Wang | 12.12.15 | Initial Use case # 6 | 1.0 |

Use Case #7

Found History

|  |  |  |  |
| --- | --- | --- | --- |
| Use Case ID: | 007 | | |
| Use Case Name: | Found History | | |
| Created By: | Qi Wang | Last Updated By: | Qi Wang |
| Date Created: | 12.12.15 | Date Last Updated: | 12. 12. 15 |

|  |  |
| --- | --- |
| Actors: | User |
| Description: | Shows all found item that user has been posted, and user deletes found item that has been given back to its owner. |
| Trigger: | Click on “Found History” button on Home page |
| Preconditions: | 1. Student is a registered user of the system. 2. User identity is authenticated. |
| Postconditions: | * + 1. Remove item from found item database.     2. Removes related pin from map. |
| Normal Flow: | *1.0 User navigates to the found history page*  1.1 Page shows found item list for user.  1.2 User clicks “Found It” button for item that is already been returned to its owner.  1.3 System moves the found item from found item database.  1.4 System goes back to the Home page and tells user item has been removed from map. |
| Alternative Flows: | N/A |
| Exceptions: | N/A |
| Includes: | N/A |
| Priority: | High |
| Frequency of Use: | Varies based on user |
| Business Rules: | N/A |
| Special Requirements: | N/A |
| Assumptions: | N/A |
| Notes and Issues: | N/A |

Revision History

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Date** | **Reason For Changes** | **Version** |
| Qi Wang | 12.12.15 | Initial Use case # 7 | 1.0 |