# Report No. 3 Software Requirement Specification

## User Requirement Specification

### Unauthorized User Requirement

Unauthorized Users are those do not login to this system. They only have one function.

* Login.

### Authorized User Requirement

Authorized Users are those have logged into this system. They have two functions.

* Edit profile.
* Logout.
* search

### Borrower Requirement

Borrower can use some following functions: //dinh nghia borrower

* + Check in
  + Check out
* Add wish list book
* View borrowed list

### Librarian Requirement

Librarian is the user who interacts directly to borrower, they can use some following functions:

* Search borrower
* Edit borrower
* Add new books
* Search books

### Admin Requirement

Admin, who is responsible for managing accounts for the whole system, has the following function:

* Manage accounts

### Emulator Requirement

Emulator is the device which can interact with mobile phone do what?, it can use some following functions:

* Read & write RFID tags
* Verify user by NFC/QR code

### Auto Handler Requirement

Auto Handler can use some following functions: dinh nghia

* + Check in, out borrower
  + Push notification to whom?

## System Requirement Specification



### External Interface Requirement

#### User Interface

* The user interface uses Vietnamese/English language for mobile application and English language mobile application.
* The user interface displays best on 1280x768 resolutions on desktop, and 1080x1920 on mobile.

#### Hardware Interface

* Smartphone with:
  + - * BLE support (Bluetooth 4.0)
      * NFC support
* RFID Reader and tags use 125 kHz.
* Estimate Beacon.

#### Software Interface

* Mobile application: Android OS version 4.4 (or higher). limit
* Estimate cloud for manage beacons
* Estimate SDK for connect between mobile application and beacons.

#### Communication Protocol

* Use HTTP 1.1 to communicate between the web browser and the web server.
* Use HTTP 1.1 to communicate between the mobile application and the web service.
* Use BLE protocol for communication between the mobile application and the Estimate beacons.
* Use NFC protocol for communication between the mobile application and the emulator for check in.
* RFID

### System Overview Use Case



Figure 1 : <Use case> System overview



#### Web Application



Figure 2: <Use case> Web application overview

#### Mobile Application



Figure 3: <Use case> Mobile application overview



### List of Use Case

#### Common Use Case

##### Unauthorized User

###### <Unauthorized User> Overview use case

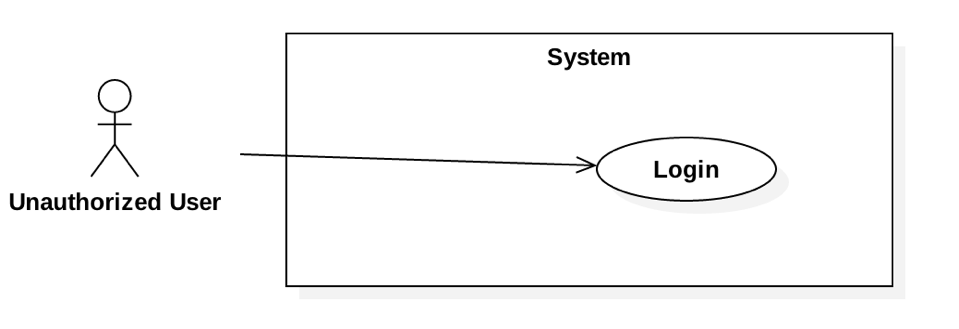


Figure 4: <Unauthorized User> Overview Use Case

###### <Unauthorized User> Login

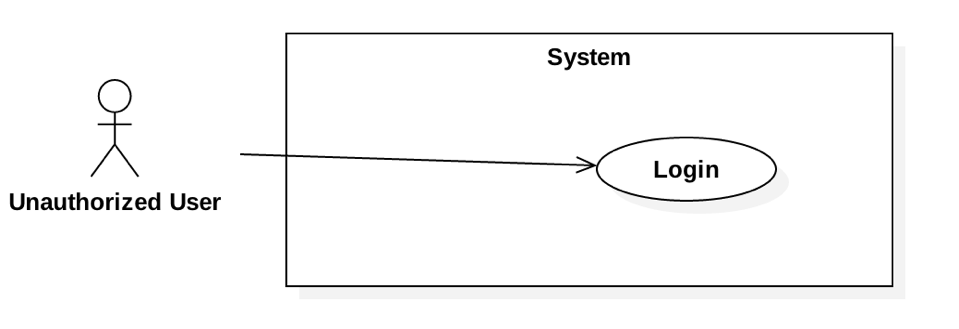


Figure 5: <Unauthorized User> Login

|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE – JWL01** | | | |
| **Use Case No.** | JWL01 | **Use Case Version** | 1.0 |
| **Use Case Name** | Login | | |
| **Author** | Nguyen Tuan Anh | | |
| **Date** | February 15, 2017 | **Priority** | Normal |
| **Actor:**   * Unauthorized User.   **Summary:**   * This use case allows Unauthorized User to log into the system with a specific role.   **Goal:**   * Authenticate user, redirect user to a specific view based on the user’s role.   **Triggers:**   * Unauthorized User sends Login command.   **Preconditions:** N/A  **Post Conditions:**   * **Success:** Unauthorized User is logged into the system as Authorized User. * **Fail:** N/A   **Main Success Scenario:**   |  |  |  | | --- | --- | --- | | Step | Actor Action | System Response | | 1 | Unauthorized User inputs user ID and password, then sends Login command.  [Alternative 1] | Unauthorized User will be logged into System with their specific role.  The system redirects to the role’s view. |   **Alternative Scenario:**   |  |  |  | | --- | --- | --- | | No. | Actor Action | System Response | | 1 | | Unauthorized user   * Leaves the user ID and password field empty, or * Inputs wrong user ID or password | System displays the error message: “Your user ID or password is invalid.” |   **Exception:** N/A  **Relationships:** This use case will be extended by all other use cases, depending on the Actor’s role in the system.  **Business Rules:**   * System authenticates Actor by checking the user ID and password. * Actor enters password in a free text field which hides the password. * Actor’s password must be encrypted before sending to server. * After logged in to the system, the Actor will be redirected to a specific view:   For web view:   * If the role is “Admin”, the system will display Admin view. * If the role is “Librarian”, the system will display Librarian view.   For mobile application view:  + If the role is “Borrower”, the system will display the mobile application’s view. | | | |

##### Authorized User

###### <Authorized User> Overview Use Case

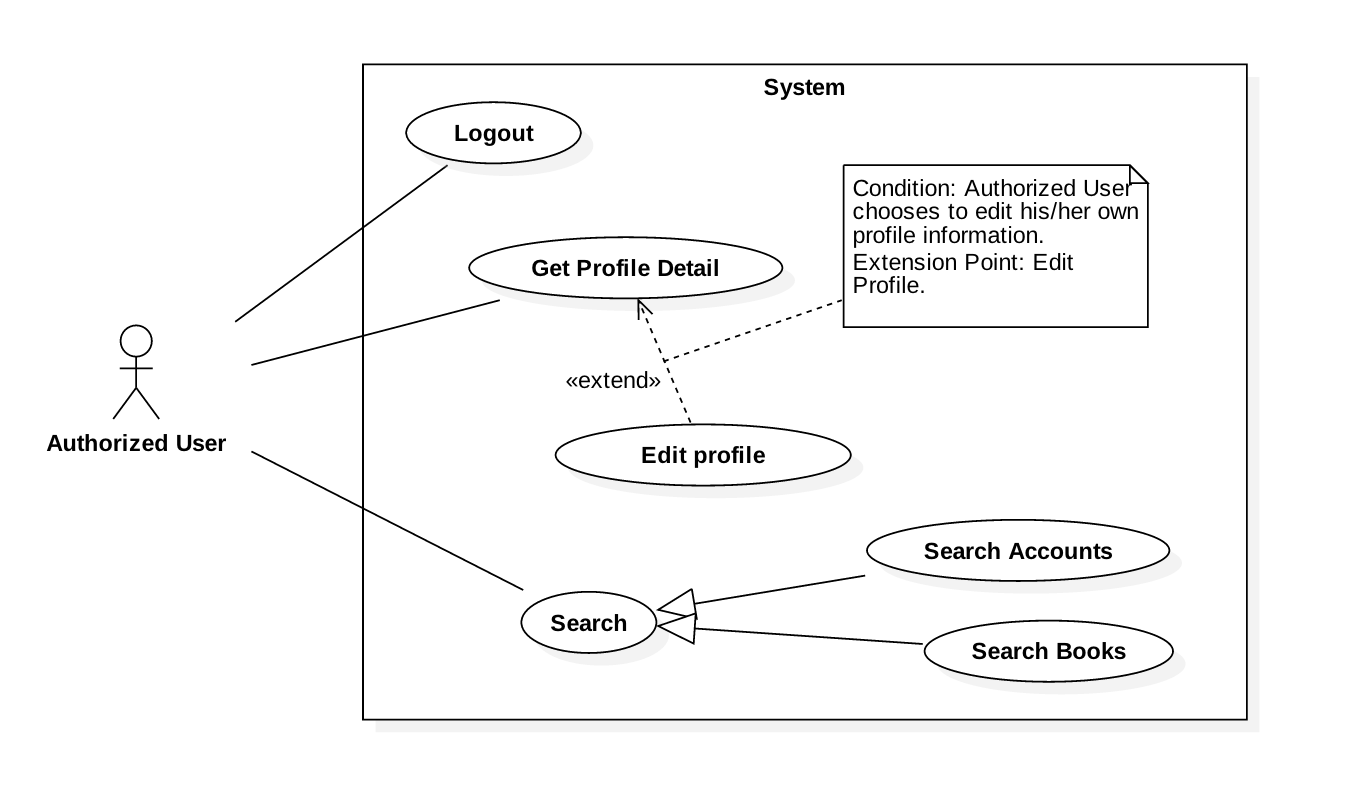


Figure 6: <Authorized User> Overview Use Case

###### <Authorized User> Get Profile Detail



Figure 7: <Authorized User> Get Profile Detail

|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE – JWL02** | | | |
| **Use Case No.** | JWL02 | **Use Case Version** | 1.0 |
| **Use Case Name** | Get Profile Detail | | |
| **Author** | Nguyen Tuan Anh | | |
| **Date** | February 20, 2017 | **Priority** | Normal |
| **Actor:**   * Authorized User.   **Summary:**  This use case allows an Actor to view his/her profile detail information, including:   * + - * User image       * User ID       * Full name       * Date of Birth       * Email       * Address       * Phone number       * Place of Work       * User Role   **Goal:**   * The System shows the Actor his/her profile detail information.   **Triggers:**   * The Actor sends Get Profile Detail command.   **Preconditions:**   * The Actor logged into the system.   **Post Conditions:**   * **Success:** The System shows the Actor his/her profile detail information. * **Fail:** N/A.   **Main Success Scenario:**   |  |  |  | | --- | --- | --- | | Step | Actor Action | System Response | | 1 | Actor sends Get Profile Detail. | System shows the Actor his/her profile detail information:   * “User image”: image * “User ID”: label * “Full name”: label * “Date of Birth”: label * “Email”: label * “Address”: label * “Phone number”: label * “Place of Work”: label * “User Role”: label |   **Alternative Scenario:** N/A  **Exceptions:** N/A  **Relationships:** From this use case, an Actor can [Edit Profile](#_<Authorized_User>_Edit).  **Business Rules:**   * This use case only allows an Actor to get his/her own detail information. | | | |

###### <Authorized User> Logout

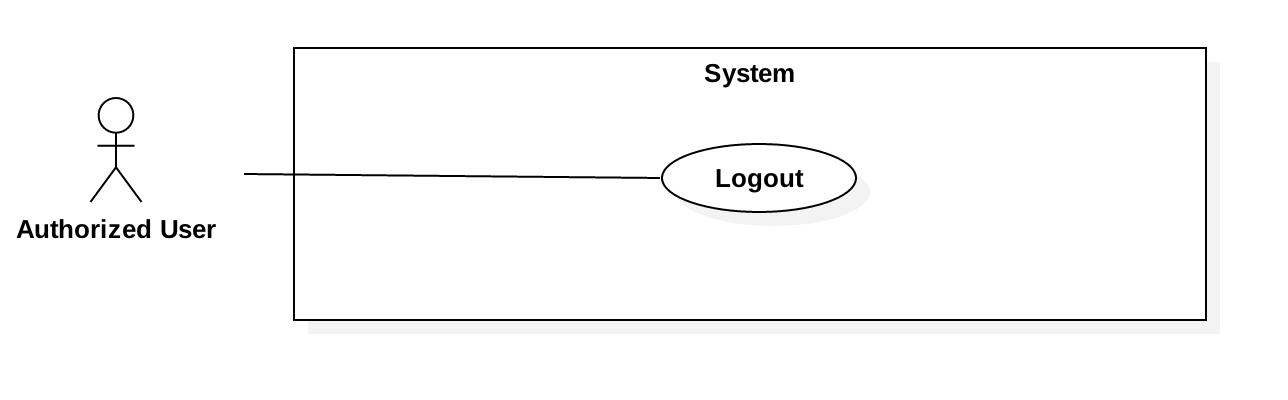


Figure 8: <Authorized User> Logout

|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE – JWL03** | | | |
| **Use Case No.** | JWL03 | **Use Case Version** | 1.0 |
| **Use Case Name** | Logout | | |
| **Author** | Nguyen Tuan Anh | | |
| **Date** | February 15, 2017 | **Priority** | Normal |
| **Actor:**   * Authorized User.   **Summary:**   * This use case allows Authorized User to log out off the system and become an Unauthorized User.   **Goal:**   * Invalidate the Authorized User, redirect the User to login view.   **Triggers:**   * Authorized User sends Logout command.   **Preconditions:**   * Actor logged in the system as an Authorized User.   **Post Conditions:**   * **Success:** Authorized User is logged out off the system and becomes Unauthorized User. * **Fail:** N/A   **Main Success Scenario:**   |  |  |  | | --- | --- | --- | | Step | Actor Action | System Response | | 1 | Authorized User sends Logout command.  [ Alternative 1] | System redirects the User to login view. |   **Alternative Scenario:**   |  |  |  | | --- | --- | --- | | No. | Cause | System Response | | 1 | | After at least one month not using the System, User sends any authorized commands. | System automatically redirects the User to login view. |   **Exceptions:** N/A  **Relationships:** N/A  **Business Rules:**   * Authorized User’s session ends after one month of inactivity. | | | |

###### <Authorized User> Edit Profile



Figure 8: <Authorized User> Edit Profile

|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE – JWL04** | | | |
| **Use Case No.** | JWL04 | **Use Case Version** | 1.0 |
| **Use Case Name** | Edit Profile | | |
| **Author** | Nguyen Tuan Anh | | |
| **Date** | February 16, 2017 | **Priority** | Normal |
| **Actor:**   * Authorized User.   **Summary:**  This use case allows an Actor to edit his/her personal information, including:   * + - * Full name       * Password       * Email       * Address       * Date of birth       * Phone number       * Place of work   **Goal:**   * Update the Actor’s personal information according to the Actor’s input.   **Triggers:**   * The Actor sends Edit Profile command.   **Preconditions:**   * The Actor must be logged into the System.   **Post Conditions:**   * **Success:** The Actor’s personal information is updated in the database. * **Fail:** System displays error messages according to fail reasons (see table **Exception**).   **Main Success Scenario:**   |  |  |  | | --- | --- | --- | | Step | Actor Action | System Response | | 1 | Actor sends Edit Profile command. | System lists out the information of the account:   * “Full name”: free text input * “Password”: free text input that hides user’s password * “Confirm Password”: free text input that hides users’ confirm password * “Date of Birth”: free text input that allows Actor to input day - month - year * “Email”: free text input * “Address”: free text input * “Phone number”: free text input * “Place of Work”: free text input | | 2 | Actor updates information. |  | | 3 | Actor sends Submit command. | * System validates the inputs then updates the account’s profile. * System redirect the Actor to the Get Profile Detail’s view. * System displays a successful message: “Update Successfully.”   [Exception 1, 2, 3, 4, 5, 6, 7] |   **Alternative Scenario:** N/A  **Exception:**   |  |  |  | | --- | --- | --- | | No | Cause | System Response | | 1 | Actor inputs Full Name field with the following wrong format:   * Blank, or all space characters * Longer than 100 characters. | System displays error messages:   * “Full name is required.” * “Full name must not be longer than 100 characters.” | | 2 | Actor inputs Password field with the following wrong format:   * Blank, or all space characters * Shorter than 6 characters * Longer than 50 characters | System displays error messages:   * “Password is required.” * “Password’s length is 6-50 characters.” | | 3 | Actor inputs Confirm Password field that does not match Password field. | System displays error message: “Confirm Password must match Password.” | | 4 | Actor inputs Email field with the following wrong format:   * Blank, or all space characters * Wrong email format (email format: [user.name]@[email.name]). | System displays error messages:   * “Email is required.” * “Wrong Email format. Please input something like your.name@mail.com” | | 5 | Actor inputs Address field with the following wrong format:   * Blank or all space characters * Longer than 250 characters | System displays error messages:   * “Address is required.” * “Address must not be longer than 250 characters.” | | 6 | Actor inputs Phone Number field with the following wrong format:   * Blank or all space characters * Wrong phone number format | System display error messages:   * “Phone Number is required.” * “Phone Number should be like +841692536224 or 01692536224.” | | 7 | Actor inputs Place of Work field with the following wrong format:   * Blank or all space characters * Longer than 100 characters | System displays error message:   * “Place of Work is required.” * “Place of Work must not be longer than 100 characters.” |   **Relationships:** Actor can only Edit Profile from [Get Profile Detail](#_<Authorized_User>_Get).  **Business Rules:**   * Actor’s information must not be blank. * Full name’s and place of work’s length must not be longer than 100 characters. * Password’s length is 6-50 characters. * Confirm password must match input password. * Email must be in format: [user.name]@[email.name]. * Phone number must be in format: +[phone no.] or [phone no.] * Address’s length must not be longer than 250 characters. * Password and Confirm Password must be encrypted before saving to the system. * Actor cannot update username, or user role. * This use case only allows an Actor to update his/her own profile information. | | | |

###### <Authorized User> Search

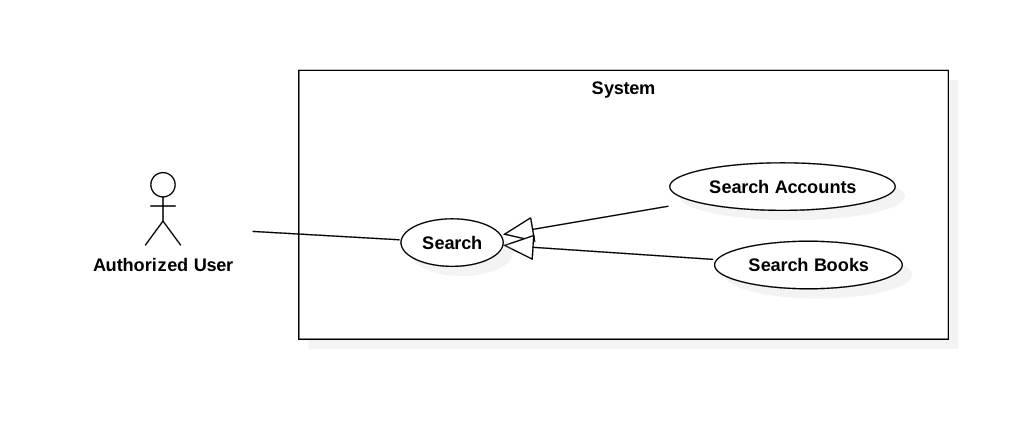


Figure 9: <Authorized User> Search

###### <Authorized User> Search Accounts

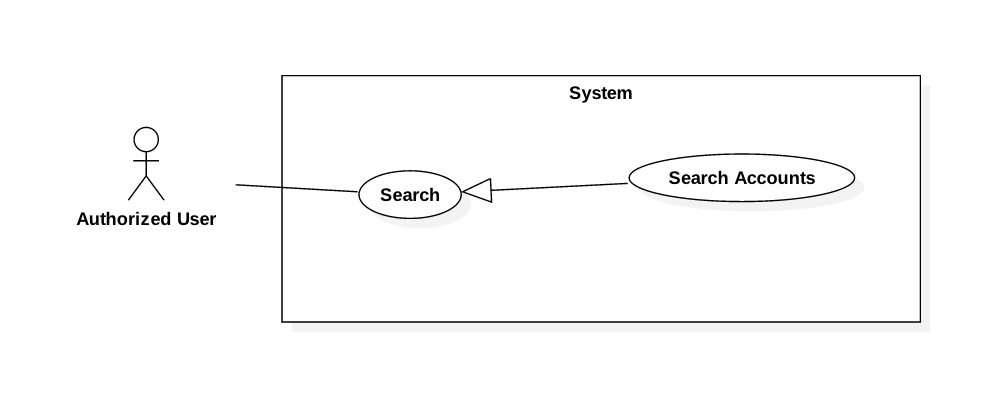


Figure 10: <Authorized User> Search Accounts

|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE – JWL05** | | | |
| **Use Case No.** | JWL05 | **Use Case Version** | 1.0 |
| **Use Case Name** | Search Accounts | | |
| **Author** | Nguyen Tuan Anh | | |
| **Date** | February 19, 2017 | **Priority** | Normal |
| **Actor:**   * Authorized User with role Admin or Librarian.   **Summary:**   * This use case allows an Actor to search Account by its user ID.   **Goal:**   * The Actor can get list of Accounts that best match his/her input user ID. The result list can vary based on the Actor’s role in the system:   + - * If the Actor’s role is Admin, the result list contains all kinds of Account (Admin, Librarian, Borrower).       * If the Actor’s role is Librarian, the result list contains only Borrowers.   **Triggers:**   * Authorized User sends Search Accounts command.   **Preconditions:**   * The Actor must be logged into the System. * The Actor must have role Admin or Librarian.   **Post Conditions:**   * **Success:**    + - * List of Accounts is showed based on the Actor’s search term.       * If there is no matched Account, the System shows message: “We couldn’t find any account with user ID like [user ID].” * **Fail:** N/A.   **Main Success Scenario:**   |  |  |  | | --- | --- | --- | | Step | Actor Action | System Response | | 1 | Actor inputs search term in the text input. |  | | 2 | Actor sends Search Accounts command. | System loads the search results with their information:  Account:   * “User ID”: label * “Full name”: label * “Phone Number”: label * “Role”: label, only visible if the Actor’s role is Admin * Is in library: label * Is activated: label   [Alternative 1] |   **Alternative Scenario:**  [Alternative 1]   |  |  |  | | --- | --- | --- | | No. | Cause | System Response | | 1 | There is no Account matches search term. | System shows message: “We couldn’t find any account with user ID like [user ID].” |   **Exception:** N/A  **Relationships:** From this Use Case, an Actor can:   * Get Account Detail if the Actor’s role is Admin. * Get Borrower Detail if the Actor’s role is Librarian.   (TODO: Update Links)  **Business Rules:**   * An Actor can only search Account by its user ID. * System lists all Accounts which user IDs contain the search term. * Only Admin and Librarian can search Accounts:   + - * Admin can search all Accounts: Admin, Librarian, and Borrower.       * Librarian can only search Borrower. | | | |

###### <Authorized User> Search Books

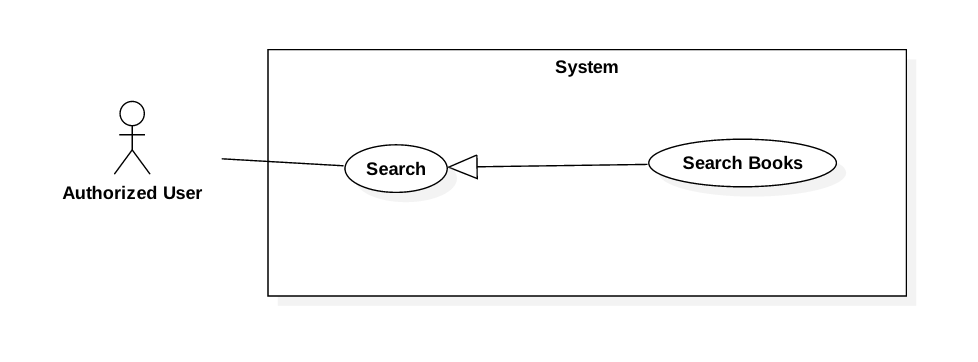


Figure 11: <Authorized User> Search Books

|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE – JWL06** | | | |
| **Use Case No.** | JWL06 | **Use Case Version** | 1.0 |
| **Use Case Name** | Search Books | | |
| **Author** | Nguyen Tuan Anh | | |
| **Date** | February 19, 2017 | **Priority** | Normal |
| **Actor:**   * Authorized User with role Librarian or Borrower.   **Summary:**   * This use case allows an Actor to search Books by:   + - * Title       * Author       * Type (textbooks, reference books, research papers, magazines, ...)       * Category (science, information technology, literature, physics, mathematics, …)       * Book copy’s ID (only available if the Actor’s role is Librarian)   **Goal:**   * The Actor can get list of Books that best match his/her input criteria.   **Triggers:**   * Authorized User sends Search Books command.   **Preconditions:**   * The Actor must be logged in the System. * The Actor must have role Librarian or Borrower.   **Post Conditions:**   * **Success:**    + - * List of Books is showed based on the Actor’s search criteria.       * If there is no matched Books, the System shows message: “We couldn’t find any book that match your description.” * **Fail:** N/A.   **Main Success Scenario:**   |  |  |  | | --- | --- | --- | | Step | Actor Action | System Response | | 1 | Actor inputs search term in the text input.  [Alternative 1] |  | | 2 | Actor sends Search Books command. | System loads the search results with their information:  Account:   * “Title”: label * “Author”: label * “Available Copies”: label   [Alternative 2] |   **Alternative Scenario:**  [Alternative 1]   |  |  |  | | --- | --- | --- | | No. | Actor Action | System Response | | 1 | Actor sends Advance Search commands.  Actor inputs wanted search criteria.  Actor send Search Books command. | System shows more search criteria:   * Title: free text input * Author: free text input * Type: list of available book types that allows the Actor to choose from * Category: list of available book categories that allows the Actor to choose from * Book copy ID: free text input, only visible if the Actor’s role is Librarian.   System loads the search results with their information:  Account:   * “Title”: label * “Author”: label * “Available Copies”: label | | 2 | There is no Book matches search term. | System shows message: “We couldn’t find any book that matches your description.” |   **Exception:** N/A  **Relationships:** From this Use Case, an Actor can   * Get Book Detail * Add Book To Wish List if the Actor’s role is Borrower, and the Book is not currently available   (TODO: Update links)  **Business Rules:**   * Only Librarian and Borrower can search Books:   + - * Both Librarian and Borrower can search Book by its title, author, type, category.       * Only Librarian can search Book by book copy ID. * System lists all Books that match all search criteria of the Actor. | | | |

#### Web Application

##### Admin

###### <Admin> Overview Use Case

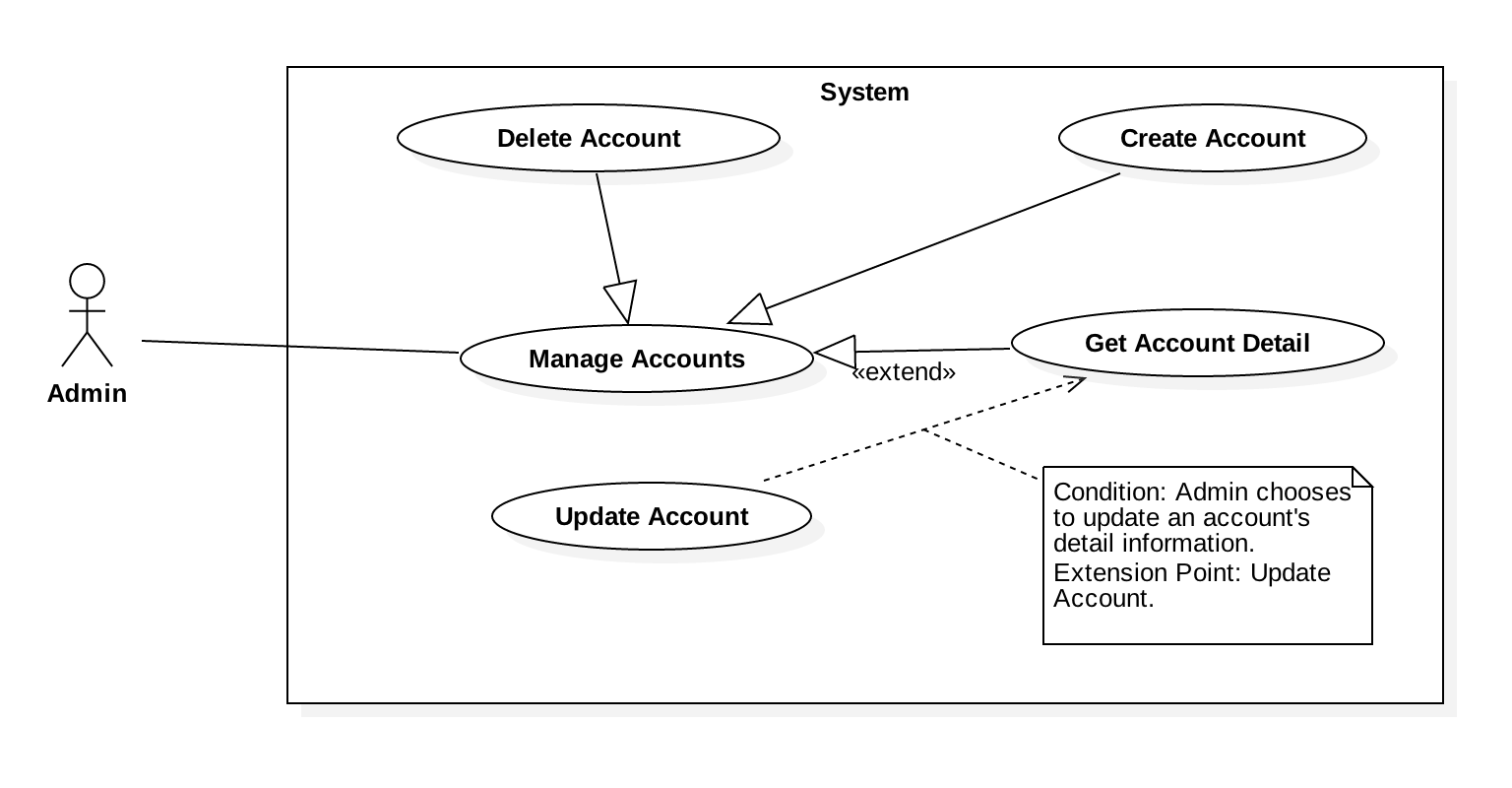


Figure 12: <Admin> Overview Use Case

###### 

###### <Admin> Create Account

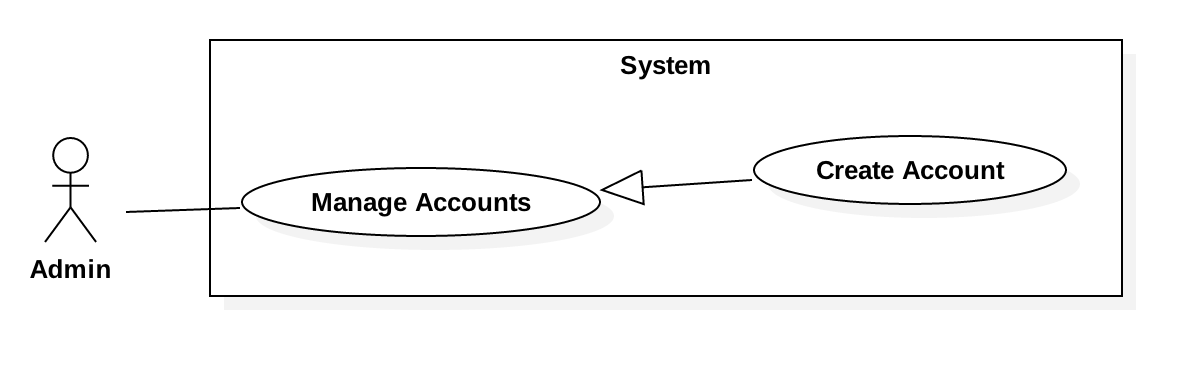


Figure 13: <Admin> Create Account

|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE – JWL07** | | | |
| **Use Case No.** | JWL07 | **Use Case Version** | 1.0 |
| **Use Case Name** | Create Account | | |
| **Author** | Nguyen Tuan Anh | | |
| **Date** | February 16, 2017 | **Priority** | Normal |
| **Actor:**   * Admin   **Summary:**   * This use case allows Admin to create an account for the System. The new account can be a Borrower, a Librarian, or an Admin.   **Goal:**   * A new account is added to the System.   **Triggers:**   * Admin sends Create Account command.   **Preconditions:**   * Actor logged into the system with role Admin.   **Post Conditions:**   * **Success:** A new account with input information is added to the System’s database. * **Fail:** System displays error messages showing fail reasons (see table **Exception**).   **Main Success Scenario:**   |  |  |  | | --- | --- | --- | | Step | Actor Action | System Response | | 1 | Admin sends Create Account command. | System lists out the information of the new account:   * “User ID”: free text input * “Full name”: free text input * “Password”: free text input that hides the input password * “Date of Birth”: free text input that allows Admin to input day - month - year * “Email”: free text input * “Address”: free text input * “Phone number”: free text input * “Place of Work”: free text input * “User Role”: list that allows Admin to choose one of the system role in the list: Admin, Librarian, Borrower | | 2 | Admin inputs the information. |  | | 3 | Admin sends Submit command. | * System validates the input then create a new account. * System displays a successful message: “New Account for [User ID, Full name] is added.” * System clears all the free text inputs in the Create Account view.   [Exception 1, 2, 3, 4, 5, 6, 7, 8, 9] |   **Alternative Scenario:** N/A  **Exceptions:**   |  |  |  | | --- | --- | --- | | No | Cause | System Response | | 1 | Admin inputs User ID field with the following wrong format:   * Blank, or all space characters * Shorter than 6 characters. * Longer than 100 characters. | System displays error messages:   * “User ID is required.” * “User ID’s length must be 6-100 characters.” | | 2 | Admin inputs Full Name field with the following wrong format:   * Blank, or all space characters * Longer than 100 characters. | System displays error messages:   * “Full name is required.” * “Full name must not be longer than 100 characters.” | | 3 | Admin inputs Password field with the following wrong format:   * Blank, or all space characters * Shorter than 6 characters * Longer than 50 characters | System displays error messages:   * “Password is required.” * “Password’s length is 6-50 characters.” | | 4 | Admin inputs Email field with the following wrong format:   * Blank, or all space characters. * Wrong email format (email format: [user.name]@[email.name]). | System displays error messages:   * “Email Address is required.” * “Wrong email format. Please input something like your.name@mail.com”. | | 5 | Admin inputs Address field with the following wrong format:   * Blank or all space characters * Longer than 250 characters | System displays error messages:   * “Address is required.” * “Address must not be longer than 250 characters.” | | 6 | Admin inputs Phone Number field with the following wrong format:   * Blank or all space characters * Wrong phone number format. | System display error messages:   * “Phone Number is required.” * “Phone Number should be like +841692536224 or 01692536224.” | | 7 | Admin inputs Place of Work field with the following wrong format:   * Blank or all space characters * Longer than 100 characters | System displays error message:   * “Place of Work is required.” * “Place of Work must not be longer than 100 characters.” | | 8 | User ID already exists. | System displays error message: “User ID already exists.” |   **Relationships:** N/A  **Business Rules:**   * System check for the Actor’s role: only role Admin can create a new account. * System check for existing User ID before trying to create a new account. * User’s information must not be blank. * User ID length is 6-100 characters. * Full name’s and place of work’s length must not be longer than 100 characters. * Password must be encrypted before saving to the system. * Password’s length is 6-50 characters. * Email must be in format: [user.name]@[mail.name]. * Phone number must be in format: +[phone no.] or [phone no.] * Address’s length must not be longer than 250 characters. * Newly created Account has ‘activated’ status. | | | |

###### <Admin> Get Account Detail

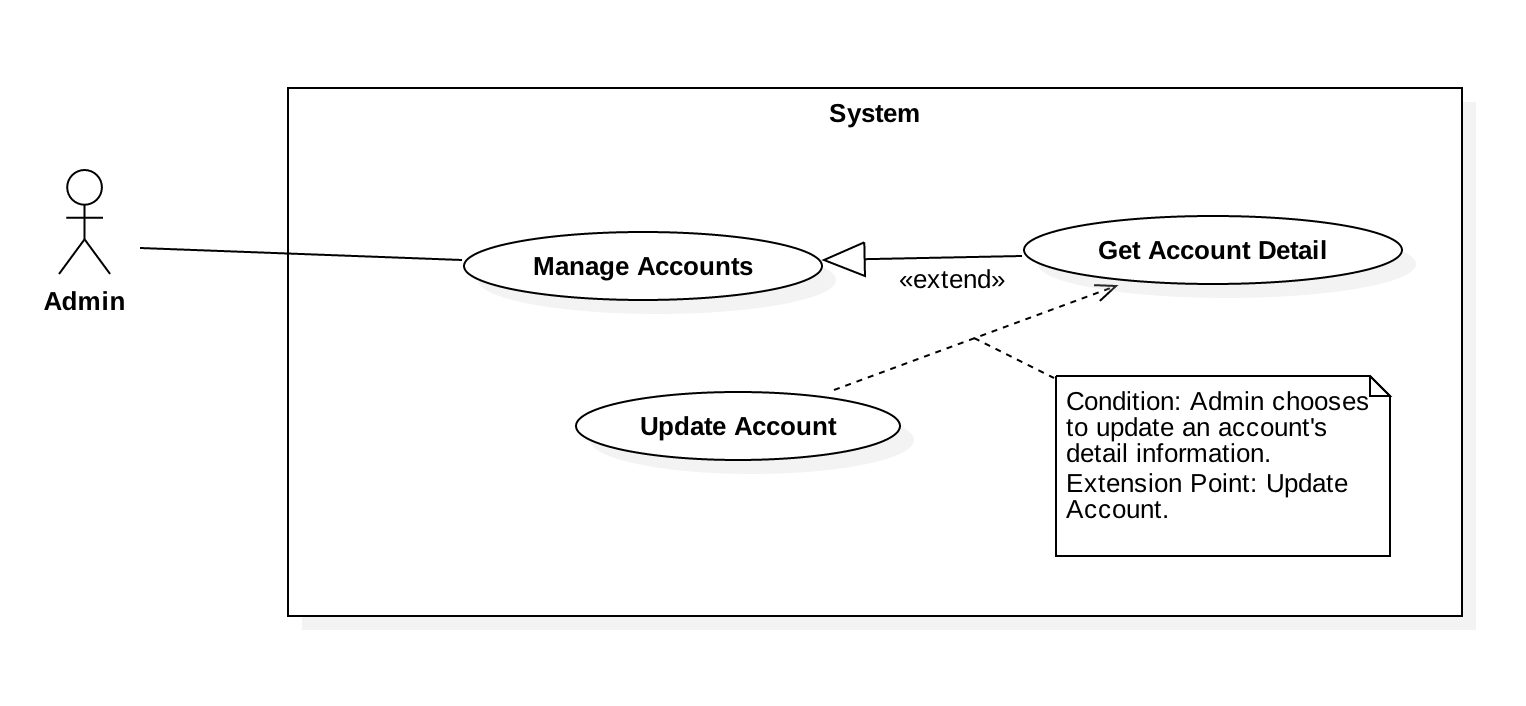


Figure 14: <Admin>Update Account

|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE – JWL08** | | | |
| **Use Case No.** | JWL08 | **Use Case Version** | 1.0 |
| **Use Case Name** | Get Account Detail | | |
| **Author** | Nguyen Tuan Anh | | |
| **Date** | February 21, 2017 | **Priority** | Normal |
| **Actor:**   * Admin   **Summary:**  This use case allows Admin to get an account’s detail information, including   * + - * User ID       * Full name       * Email       * Address       * Date of birth       * Phone number       * Place of work       * User role       * Status of user (activated/deactivated)   **Goal:**   * The System show the Admin his/her requested account’s detail information.   **Triggers:**   * Admin sends Get Account Detail command.   **Preconditions:**   * Actor logged in system with role Admin.   **Post Conditions:**   * **Success:** The account’s personal information is showed to the Admin. * **Fail:** N/A   **Main Success Scenario:**   |  |  |  | | --- | --- | --- | | Step | Actor Action | System Response | | 1 | Admin sends Get Account Detail command of a specific account. | System lists the detail information of the requested account:   * “User ID”: label * “Full name”: label * “Date of Birth”: label * “Email”: label * “Address”: label * “Phone number”: label * “Place of Work”: label * “User Role”: label * User status”: label |   **Alternative Scenario:** N/A  **Exception:** N/A  **Relationships:** From this use case, Admin can [Update Account](#_<Admin>_Update_Account).  **Business Rules:**   * Admin can get any account’s detail, including Admin, Librarian, and Borrower. | | | |

###### <Admin> Update Account

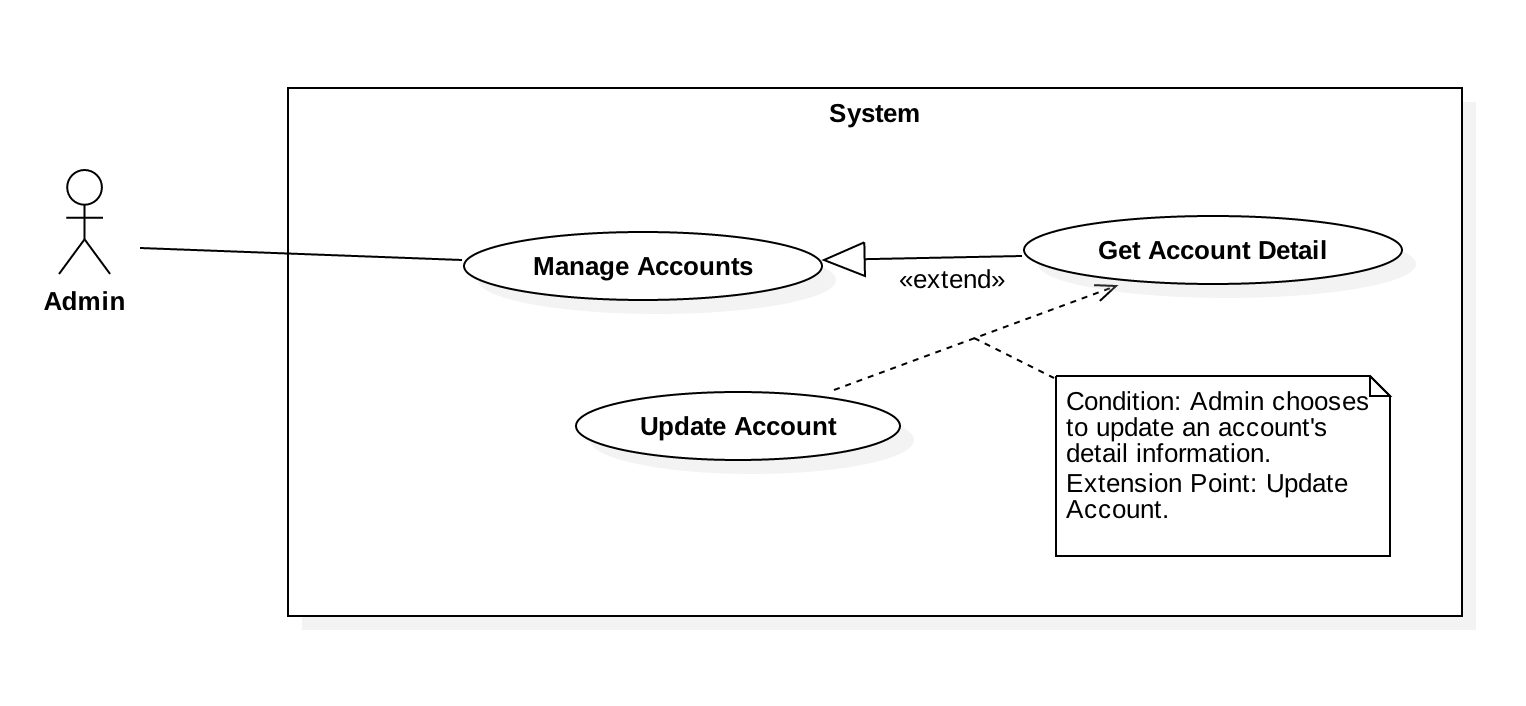


Figure 15: <Admin>Update Account

|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE – JWL09** | | | |
| **Use Case No.** | JWL09 | **Use Case Version** | 1.0 |
| **Use Case Name** | Update Account | | |
| **Author** | Nguyen Tuan Anh | | |
| **Date** | February 16, 2017 | **Priority** | Normal |
| **Actor:**   * Admin   **Summary:**  This use case allows Admin to update other accounts’ information, including   * + - * Full name       * Password       * Email       * Address       * Date of birth       * Phone number       * Place of work       * User role       * Status of user (activated/deactivated).   **Goal:**   * Update personal information of an account that Admin chooses.   **Triggers:**   * Admin sends Update Account command.   **Preconditions:**   * Actor logged in system with role Admin.   **Post Conditions:**   * **Success:** The account’s personal information is updated in the database. * **Fail:** System displays error messages showing fail reasons (see table **Exception**).   **Main Success Scenario:**   |  |  |  | | --- | --- | --- | | Step | Actor Action | System Response | | 1 | Admin sends Update Account command. | System lists out the information of the account:   * “Full name”: free text input * “Password”: free text input that hides the input password * “Confirm Password”: free text input that hides the input confirm password * “Date of Birth”: free text input that allows Admin to input day - month - year * “Email”: free text input * “Address”: free text input * “Phone number”: free text input * “Place of Work”: free text input * “User Role”: list that allows Admin to choose one of the system roles: Admin, Librarian, Borrower * User status”: checkbox, which has 2 values: checked and unchecked * Checked: activated. * Unchecked: deactivated. | | 2 | Admin input the information. |  | | 3 | Admin sends Submit command. | * System validates the input then update the account. * System redirect the Admin to Get Account Detail’s view. * System displays a successful message: “The Account of [username] is updated.”   [Exception 1, 2, 3, 4, 5, 6, 7, 8] |   **Alternative Scenario:** N/A  **Exception:**   |  |  |  | | --- | --- | --- | | No | Cause | System Response | | 1 | Admin inputs Full Name field with the following wrong format:   * Blank, or all space characters * Longer than 100 characters. | System displays error messages:   * “Full name is required.” * “Full name must not be longer than 100 characters.” | | 2 | Admin inputs Password field with the following wrong format:   * Blank, or all space characters * Shorter than 6 characters * Longer than 50 characters | System displays error messages:   * “Password is required.” * “Password’s length is 6-50 characters.” | | 3 | Admin inputs Confirm Password field that does not match Password field. | System displays error message: “Confirm Password must match Password.” | | 4 | Admin inputs Email field with the following wrong format:   * Blank, or all space characters * Wrong email format (email format: [user.name]@[email.name]). | System displays error messages:   * “Email is required.” * “Wrong Email format. Please input something like abc@somemail.com”. | | 5 | Admin inputs Address field with the following wrong format:   * Blank or all space characters * Longer than 250 characters | System displays error messages:   * “Address is required.” * “Address must not be longer than 250 characters.” | | 6 | Admin inputs Phone Number field with the following wrong format:   * Blank or all space characters * Wrong phone number format. | System displays error messages:   * “Phone Number is required.” * “Phone Number should be like +841692536224 or 01692536224.” | | 7 | Admin inputs Place of Work field with the following wrong format:   * Blank or all space characters * Longer than 100 characters | System displays error message:   * “Place of Work is required.” * “Place of Work must not be longer than 100 characters.” | | 8 | User ID does not exist. | System displays error message: “The Account [User ID, Full name] is invalid.” |   **Relationships:** Admin can only Update Account from [Get Account Detail](#_<Admin>_Get_Account).  **Business Rules:**   * System check for The Actor’s role: only Admin role can update an account. * System check for account’s existence (by User ID) before trying to update it. * User’s information must not be blank. * Full name’s and place of work’s length must not be longer than 100 characters. * Password’s length is 6-50 characters. * Confirm password must match input password. * Email must be in format: [user.name]@[email.name] * Phone number must be in format: +[phone no.] or [phone no.] * Address’s length must not be longer than 250 characters. * Password must be encrypted before saving to the system. | | | |

###### <Admin> Delete Account

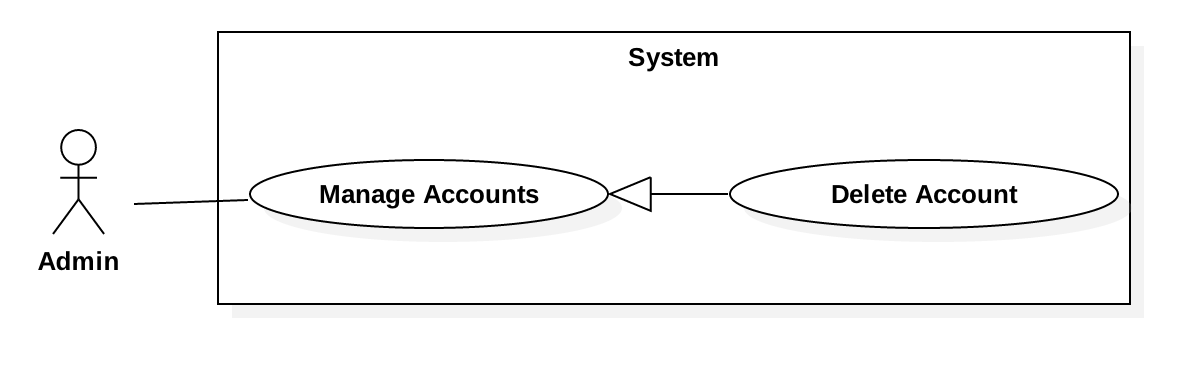


Figure 16: <Admin> Delete Account

|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE – JWL010** | | | |
| **Use Case No.** | JWL10 | **Use Case Version** | 1.0 |
| **Use Case Name** | Delete Account | | |
| **Author** | Nguyen Tuan Anh | | |
| **Date** | February 20, 2017 | **Priority** | Normal |
| **Actor:**   * Admin   **Summary:**   * This use case allows Admin to delete another account from the system.   **Goal:**   * Delete an account from the system. No Actor can have access to that account anymore.   **Triggers:**   * Admin sends Delete Account command.   **Preconditions:**   * Actor logged in system with role Admin.   **Post Conditions:**   * **Success:**   + The account’s delete date is updated in the System database.  + The account can no longer be viewed or do any functions of the System.   * **Fail:** System displays error messages showing fail reasons (see table **Exception**).   **Main Success Scenario:**   |  |  |  | | --- | --- | --- | | Step | Actor Action | System Response | | 1 | Admin sends Delete Account command on a specific account. | System shows Delete Account pop up which includes:   * “User ID”: label * “Full name”: label * “Date of Birth”: label * “Phone number”: label * “Address”: label * “Place of work”: label * “User Role”: label * System asks for confirmation. | | 2 | Admin sends Confirm command.  [Alternative 1] | * System displays a successful message: “the Account [user ID, Full name] is successfully deleted.” |   **Alternative Scenario:**   |  |  |  | | --- | --- | --- | | No | Actor Action | System Response | | 1 | Admin sends Cancel command. | System closes the Delete Account pop up. |   **Exception:** N/A  **Relationships:** N/A  **Business Rules:**   * Before deleting an account, System checks the following conditions:   + - * The account’s user ID’s existence: if the account does not exist, or it has delete date field, System stop the delete account command.       * If the account is a Borrower and having unreturned books, System will:   + Stop the delete account command.  + Warn the Admin “This Borrower is having unreturned books!”   * + - * If the account is a Borrower and is in the library, System will:   + Stop the delete account command.  + Warn the Admin “This Borrower is in the library!”   * After the deletion, the account’s delete date is updated in the System database. * Deleted accounts can no longer be view or do any functions of the System. | | | |

##### Librarian

###### <Librarian> Overview Use Case

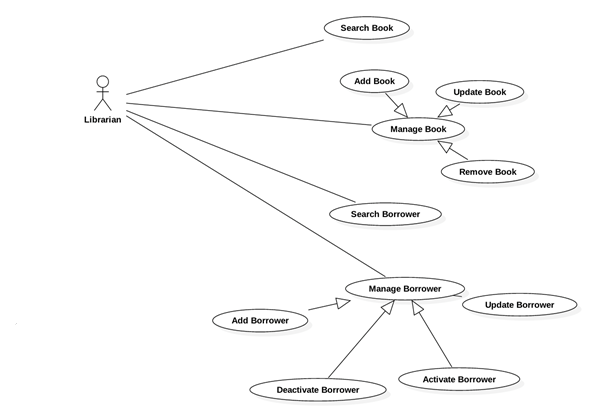


Figure 14: <Librarian> Overview Use Case

###### <Librarian> Search Book



Figure 15: <Librarian> Search Book

|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE – JWL08** | | | |
| **Use Case No.** | JWL09 | **Use Case Version** | 1.0 |
| **Use Case Name** | Search Book | | |
| **Author** | Dang Nhat Thien | | |
| **Date** | February 12, 2017 | **Priority** | Normal |
| **Actor:**   * Librarian   **Summary:**   * This use case allows Librarian to search book.   **Goal:**   * Librarian can get a list book that match input book’s id, book’s name or book’s author.   **Triggers:**   * Librarian changes the content of the search field.   **Preconditions:**   * Actor logged in system with role Librarian.   **Post Conditions:**   * **Success:** List of matching books based on search term is showed. * **Fail:** N/A   **Main Success Scenario:**   |  |  |  | | --- | --- | --- | | Step | Actor Action | System Response | | 1 | Librarian inputs wanted book’s id, book’s name or book’s author. | System loads the result each time the Librarian click on “Search” button. Information for each account:   * Book Title * Publisher * Description * Publish year * Number of pages * Number of copies * Book type * Position * ISBN   [Alternative 1] |   **Alternative Scenario:**   |  |  |  | | --- | --- | --- | | Step | Cause | System Response | | 1 | There is no book that matches the search term. | System shows message: “No book found.” |   **Exceptions:** N/A  **Relationships:** N/A  **Business Rules:**   * Each book has a unique id. | | | |

###### <Librarian> Add Book

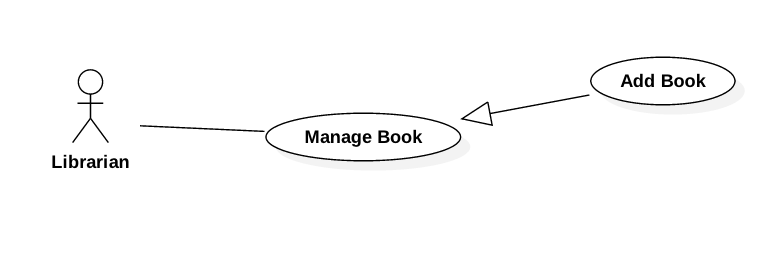


Figure 16: <Librarian> Add Book

|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE – JWL08** | | | |
| **Use Case No.** | JWL10 | **Use Case Version** | 1.0 |
| **Use Case Name** | Add Book | | |
| **Author** | Dang Nhat Thien | | |
| **Date** | February 12, 2017 | **Priority** | Normal |
| **Actor:**   * Librarian   **Summary:**   * This use case allows Librarian to Add Book.   **Goal:**   * Librarian can Add New Book and its copies.   **Triggers:**   * Librarian send Add New Book command. * Librarian click on Add New Copy. * Librarian click on Add Book.   **Preconditions:**   * Actor logged in system with role Librarian.   **Post Conditions:**   * **Success:** New Book is saved to database and system show message: “Book [title] has been added!” * **Fail:** N/A   **Main Success Scenario:**   |  |  |  | | --- | --- | --- | | Step | Actor Action | System Response | | 1 | Librarian send Add New Book command.  Librarian input information and   1. Librarian click on Add New Copy 2. Librarian click on Add Book | System lists out the information of the new Book:   * “Book title”: text input, length 1-50 * “Publisher”: text input, length 1-100 * “Publish year”: text input, length 4 * “Page Number”: text input, length 1-4 * “Description”: text input, length 1-500 * “RFID”: text input, length 10   This RFID is display in a table below Book information. [Exception 6]  New Book with all copies of it are saved to database. The system show message “Book [title] has been added!”  [Exception 1,2,3,4,5] |   **Alternative Scenario:** N/A  **Exceptions:**   |  |  |  | | --- | --- | --- | | No | Cause | System Response | | 1 | Admin inputs Title field with wrong format. | System displays error message: “Title length is 1 - 50 characters.” | | 2 | Admin inputs Publisher field with wrong format. | System displays error message: “Publisher length is 1-100 characters.” | | 3 | Admin inputs Publish Year field with wrong format. | System displays error message: “Password length is 4 characters and Password must be positive integer” | | 4 | Admin inputs Page Number field with wrong format. | System displays error message: “Page Number length is 1 - 4 characters and Page Number must be positive integer” | | 5 | Admin inputs Description field with wrong format. | System displays error message: “Description length is 1 - 500 characters and Description must be positive integer” | | 6 | Admin inputs RFID field with wrong format. | System displays error message: “RFID length is 10 characters.” |   **Relationships:** N/A  **Business Rules:** N/A | | | |

###### <Librarian> Update Book

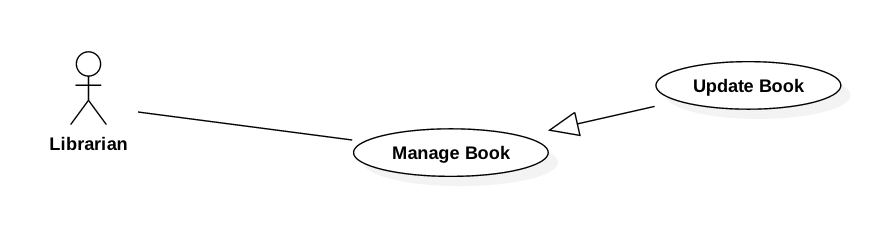


Figure 17: <Librarian> Update Book

|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE – JWL08** | | | |
| **Use Case No.** | JWL11 | **Use Case Version** | 1.0 |
| **Use Case Name** | Update Book | | |
| **Author** | Dang Nhat Thien | | |
| **Date** | February 12, 2017 | **Priority** | Normal |
| **Actor:**   * Librarian   **Summary:**   * This use case allows Librarian to Update Book.   **Goal:**   * Librarian can Update Book detail.   **Triggers:**   * Librarian send Add New Book command. * Librarian click on “Add New Copy” button. * Librarian click on “Add Book” button. * Librarian click on “Delete This Copy” button.   **Preconditions:**   * Actor logged in system with role Librarian. * Librarian find out the Book want to update via [<Librarian> Search Book](#_<Librarian>_Search_Book)   **Post Conditions:**   * **Success:** Book is saved to database and system show message: “Book [title] has been updated!” * **Fail:** N/A   **Main Success Scenario:**   |  |  |  | | --- | --- | --- | | Step | Actor Action | System Response | | 1 | Librarian send Update Book command from a Book Detail.  Librarian input information and   1. Librarian click on “Add New Copy” button. 2. Librarian click on “Delete Copy” button. 3. Librarian click on “Add Book”   button | System lists out the information of the new Book:   * “Book title”: text input, length 1-50 * “Publisher”: text input, length 1-100 * “Publish year”: text input, length 4 * “Page Number”: text input, length 1-4 * “Description”: text input, length 1-500 * “RFID”: text input, length 10   This RFID is display in a table below Book information. [Exception 6]  This RFID is deleted from view.  Book with all copies of it are saved to database. The system show message “Book [title] has been updated!”  [Exception 1,2,3,4,5] |   **Alternative Scenario:** N/A  **Exceptions:**   |  |  |  | | --- | --- | --- | | No | Cause | System Response | | 1 | Admin inputs Title field with wrong format. | System displays error message: “Title length is 1 - 50 characters.” | | 2 | Admin inputs Publisher field with wrong format. | System displays error message: “Publisher length is 1-100 characters.” | | 3 | Admin inputs Publish Year field with wrong format. | System displays error message: “Password length is 4 characters and Password must be positive integer” | | 4 | Admin inputs Page Number field with wrong format. | System displays error message: “Page Number length is 1 - 4 characters and Page Number must be positive integer” | | 5 | Admin inputs Description field with wrong format. | System displays error message: “Description length is 1 - 500 characters and Description must be positive integer” | | 6 | Admin inputs RFID field with wrong format. | System displays error message: “RFID length is 10 characters.” |   **Relationships:** [<Librarian> Search Book](#_<Librarian>_Search_Book)  **Business Rules:** N/A | | | |

###### <Librarian> Remove Book

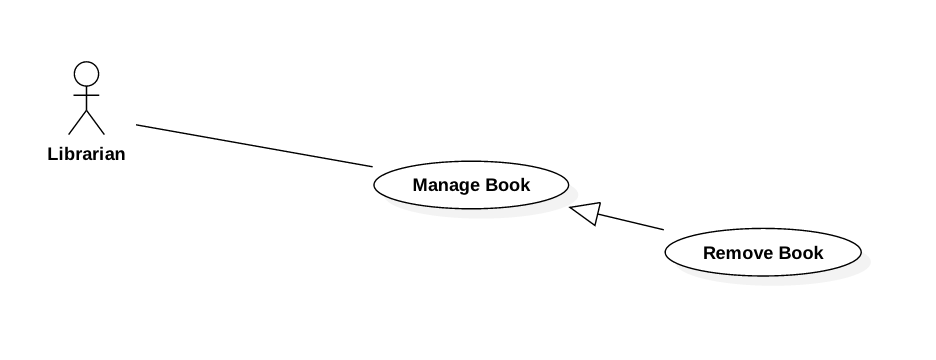


Figure 18: <Librarian> Add Book

|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE – JWL08** | | | |
| **Use Case No.** | JWL12 | **Use Case Version** | 1.0 |
| **Use Case Name** | Remove Book | | |
| **Author** | Dang Nhat Thien | | |
| **Date** | February 12, 2017 | **Priority** | Normal |
| **Actor:**   * Librarian   **Summary:**   * This use case allows Librarian to Remove Book.   **Goal:**   * Librarian can Remove Book.   **Triggers:**   * Librarian send Delete Book command.   **Preconditions:**   * Actor logged in system with role Librarian. * Librarian find out the Book want to update via [<Librarian> Search Book](#_<Librarian>_Search_Book)   **Post Conditions:**   * **Success:** Book is removed from database and system show message: “Book [title] has been removed!” * **Fail:** N/A   **Main Success Scenario:**   |  |  |  | | --- | --- | --- | | Step | Actor Action | System Response | | 1 | Librarian send Remove Book command. | System delete this book from database and show message: “Book [title] has been removed!” |   **Alternative Scenario:** N/A  **Exceptions:** N/A  **Relationships:** [<Librarian> Search Book](#_<Librarian>_Search_Book)  **Business Rules:** N/A | | | |

###### <Librarian> Add Borrower



Figure 19: <Librarian> Add Borrower

|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE – JWL09** | | | |
| **Use Case No.** | JWL13 | **Use Case Version** | 1.0 |
| **Use Case Name** | Add Borrower | | |
| **Author** | Dang Nhat Thien | | |
| **Date** | February 13, 2017 | **Priority** | Normal |
| **Actor:**   * Librarian   **Summary:**   * This use case allows Librarian to create an account for Borrower   **Goal:**   * Librarian can create a new account for Borrower.   **Triggers:**   * Librarian sends Create Borrower command.   **Preconditions:**   * Actor logged in system with role Librarian.   **Post Conditions:**   * **Success:** A new Borrower with input information is added to the System database. * **Fail:** N/A   **Main Success Scenario:**   |  |  |  | | --- | --- | --- | | Step | Actor Action | System Response | | 1 | Librarian sends Create Borrower command. | System lists out the information of the new account:   * “Username”: text input, length 6-50 * “Full name”: text input, length 6-100 * “Password”: password input, length 6-50 * “Confirm Password”: password input, length 6-50 * “Date of Birth”: datetime picker * “Email”: email input * “Address”: text input, length 6-100 * “Phone number”: text input, length 8-12 * “Place of Work”: text input: length 6-50 * “Account Type”: dropdown box: Student, Teacher, Officer, Retired Officer, Business Person * Commit command. | | 2 | Librarian input the information. |  | | 3 | Librarian sends Submit command. | * System validates the input then create a new borrower. * System displays a successful message: “New Borrower: [username] is added.”   [Exception 1, 2, 3, 4, 5, 6, 7] |   **Alternative Scenario:** N/A  **Exceptions:**   |  |  |  | | --- | --- | --- | | No | Cause | System Response | | 1 | Librarian inputs Username field with wrong format. | System displays error message: “Full name length is 6-50 characters.” | | 2 | Librarian inputs Full Name field with wrong format. | System displays error message: “Full name length is 6-100 characters.” | | 3 | Librarian inputs Password field with wrong format. | System displays error message: “Password length is 6-50 characters.” | | 4 | Librarian inputs wrong Confirm Password field. | System displays error message: “Confirm Password must match Password.” | | 5 | Librarian inputs Email field with wrong format. | System displays error message: “Wrong Email format. Please input something like abc@somemail.com” | | 6 | Librarian inputs Address field with wrong format. | System displays error message: “Address length is 6-100 characters.” | | 7 | Librarian inputs Phone Number field with wrong format. | System displays error message: “Phone Number should be all numbers at 8-12 length.” | | 8 | Librarian inputs Place of Work field with wrong format. | System displays error message: “Place of Work length is 6-100 characters.” |   **Relationships:** N/A  **Business Rules:**   * Each borrower has a unique username, such as student ID, or identity ID. * Password must be encrypted before saving to the system. | | | |

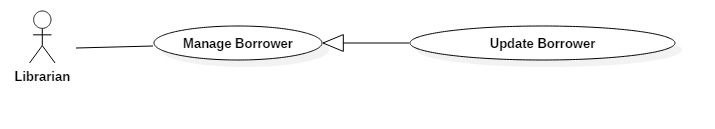
###### <Librarian> Search Borrower



Figure 22: <Librarian> Search Borrower

|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE – JWL10** | | | |
| **Use Case No.** | JWL14 | **Use Case Version** | 1.0 |
| **Use Case Name** | Search Borrower | | |
| **Author** | Dang Nhat Thien | | |
| **Date** | February 13, 2017 | **Priority** | Normal |
| **Actor:**   * Librarian   **Summary:**   * This use case allows Librarian to search for a borrower.   **Goal:**   * Librarian can get a list borrower that match input username or full name.   **Triggers:**   * Librarian changes the content of the search field.   **Preconditions:**   * Actor logged in system with role Librarian.   **Post Conditions:**   * **Success:** List of matching borrowers based on search term is showed. * **Fail:** N/A   **Main Success Scenario:**   |  |  |  | | --- | --- | --- | | Step | Actor Action | System Response | | 1 | Librarian inputs wanted username or full name. | System loads the result each time the input search field changes. Information for each Borrower:   * Username * Full name * Date of Birth * Phone No. * Status (active/deactivated)   [Alternative 1] |   **Alternative Scenario:**   |  |  |  | | --- | --- | --- | | Step | Cause | System Response | | 1 | There is no borrower that matches the search term. | System shows message: “No borrower found.” |   **Exceptions:** N/A  **Relationships:** N/A  **Business Rules:**   * Each borrower has a unique username, such as student ID, or identity ID. | | | |

###### <Librarian> Update Borrower



|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE – JWL11** | | | |
| **Use Case No.** | JWL15 | **Use Case Version** | 1.0 |
| **Use Case Name** | Update Borrower Information | | |
| **Author** | Dang Nhat Thien | | |
| **Date** | February 13, 2017 | **Priority** | Low |
| **Actor:**   * Librarian   **Summary:**   * This use case allows Librarian to update borrower’s information.   **Goal:**   * Librarian can update borrower’s information as he/she sends Submit command.   **Triggers:**   * Librarian sends Update command.   **Preconditions:**   * Actor logged in system with role Librarian. * Librarian find out the Borrower want to update via [<Librarian> Search Borrower](#_<Librarian>_Search_Borrower)   **Post Conditions:**   * **Success:** The account’s personal information is updated in the database. * **Fail:** N/A.   **Main Success Scenario:**   |  |  |  | | --- | --- | --- | | Step | Actor Action | System Response | | 1 | Librarian sends Update Account command on a specific account. | System lists out the information of the account:   * “Username”: text input, length 6-50 * “Full name”: text input, length 6-100 * “Password”: password input, length 6-50 * “Confirm Password”: password input, length 6-50 * “Date of Birth”: datetime picker * “Email”: email input * “Address”: text input, length 6-100 * “Phone number”: text input, length 8-12 * “Place of Work”: text input: length 6-50 * “Account Type”: dropdown box: Student, Teacher, Officer, Retired Officer, Business Person * “Status”: dropdown box: Activate, Deactivate * Update command. | | 2 | Librarian updates information.  [Alternative 1]: Librarian change state of dropdown box |  | | 3 | Librarian sends Submit command. | * System validates the input then updates the account’s profile. * System displays a successful message: “Update Successful”   [Exception 1, 2, 3, 4, 5, 6, 7] |   **Alternative Scenario:**   |  |  |  | | --- | --- | --- | | No | Cause | System Response | | 1 | Librarian change state of dropdown box:   * From Activate to Deactivate * From Deactivate to Activate | * System show text input below the dropdown box, so Librarian can enter the reason. * System show the fee the Borrower must pay for late returning book before. |   **Exception:**   |  |  |  | | --- | --- | --- | | No | Cause | System Response | | 1 | Librarian inputs Username field with wrong format. | System displays error message: “Full name length is 6-50 characters.” | | 2 | Librarian inputs Full Name field with wrong format. | System displays error message: “Full name length is 6-100 characters.” | | 3 | Librarian inputs Password field with wrong format. | System displays error message: “Password length is 6-50 characters.” | | 4 | Librarian inputs wrong Confirm Password field. | System displays error message: “Confirm Password must match Password.” | | 5 | Librarian inputs Email field with wrong format. | System displays error message: “Wrong Email format. Please input something like abc@somemail.com” | | 6 | Librarian inputs Address field with wrong format. | System displays error message: “Address length is 6-100 characters.” | | 7 | Librarian inputs Phone Number field with wrong format. | System displays error message: “Phone Number should be all numbers at 8-12 length.” | | 8 | Librarian inputs Place of Work field with wrong format. | System displays error message: “Place of Work length is 6-100 characters.” |   **Relationships:** [<Librarian> Search Borrower](#_<Librarian>_Search_Borrower)  **Business Rules:**   * Password must be encrypted before saving to the system. * Each account has a unique username, such as student ID, or identity ID. | | | |

###### <Librarian> Update List Borrowed Books

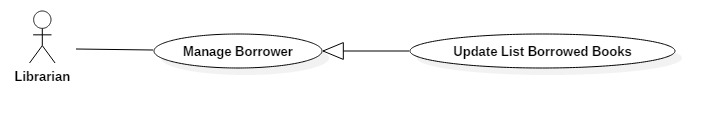


Figure 20: <Librarian> Update Borrower

|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE – JWL12** | | | |
| **Use Case No.** | JWL16 | **Use Case Version** | 1.0 |
| **Use Case Name** | Update Borrowed Books List | | |
| **Author** | Dang Nhat Thien | | |
| **Date** | February 13, 2017 | **Priority** | Low |
| **Actor:**   * Librarian   **Summary:**   * This use case allows Librarian to update list borrowed books of a borrower.   **Goal:**   * Librarian can update list borrowed books of a borrower as he/she sends Submit command.   **Triggers:**   * Librarian Click “Remove” button beside a specific book. * Click “Add” button below the list. * Librarian sends Update command.   **Preconditions:**   * Actor logged in system with role Librarian. * Librarian find out specific Borrower via [<Librarian> Search Borrower](#_<Librarian>_Search_Borrower)   **Post Conditions:**   * **Success:** The borrowed books of a user is updated in the database. * **Fail:** N/A   **Main Success Scenario:**   |  |  |  | | --- | --- | --- | | Step | Actor Action | System Response | | 1 | Librarian sends Update Borrowed Books command on a specific Borrower. | System shows list of borrowed books of this Borrower.  Update command. | | 2 | Librarian updates list of borrowed books by:  A. Click “Remove” button beside a specific book to remove this book from list.    B.1. Click “Add” button below the list to Add book to list.  B.2. Librarian enter RFID and click on “Add” button on the pop-up. | A. System hide that book from the list.  B.1 A pop-up is displayed with Text Input for librarian can input RFID of specific book.  B.2 The system find the book by RFID and display on the list.  [Alternative 1]: Can’t find out the book by RFID. | | 3 | Admin sends Submit command. | System displays a successful message: “Borrowed books of Borrower [username] is update successfully” |   **Alternative Scenario:**   |  |  |  | | --- | --- | --- | | No | Cause | System Response | | 1 | Can’t find out the book by RFID. | Show message: “Book has [RFID] not found!” |   **Exception:** N/A  **Relationships:** [<Librarian> Search Borrower](#_<Librarian>_Search_Borrower)  **Business Rules:** N/A | | | |

#### Mobile Application

##### Borrower

###### <Borrower> Overview Use Case

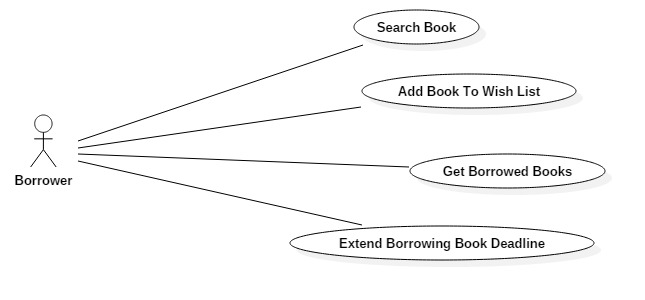


Figure 24: <Borrower> Overview Use Case

###### <Borrower> Search Book



Figure 25: <Borrower> Search Book

|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE – JWL13** | | | |
| **Use Case No.** | JWL17 | **Use Case Version** | 1.0 |
| **Use Case Name** | Search Book | | |
| **Author** | Dang Nhat Thien | | |
| **Date** | February 12, 2017 | **Priority** | Normal |
| **Actor:**   * Borrower   **Summary:**   * This use case allows Borrower to search book.   **Goal:**   * Borrower can get a list book that match input book’s name or book’s author.   **Triggers:** N/A  **Preconditions:**   * Actor logged in system with role Borrower.   **Post Conditions:**   * **Success:** List of matching books based on search term is showed. * **Fail:** A Dialog will be showed and display message: “Search fail. Please check your internet connection”.   **Main Success Scenario:**   |  |  |  | | --- | --- | --- | | Step | Actor Action | System Response | | 1 | Borrower inputs wanted book’s id, book’s name or book’s author.  Then click “Search” button. | System loads the result each time the Librarian click on “Search” button. Information for each account:   * Book Title * Publisher * Description * Publish year * Number of pages * Number of copies * Book type * Position * ISBN   [Alternative 1]: There is no book that matches the search term.  [Exception 1]: Internet connection is lost. |   **Alternative Scenario:**   |  |  |  | | --- | --- | --- | | Step | Cause | System Response | | 1 | There is no book that matches the search term. | System shows message: “No book found.” |   **Exceptions:**   |  |  |  | | --- | --- | --- | | No | Cause | System Response | | 1 | Internet connection is lost. | System show error message via Dialog. |   **Relationships:** N/A  **Business Rules:**   * Each book has a unique id. | | | |

###### <Borrower> Add Book To Wish List



Figure 26: <Borrower> Add Book To Wish List

|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE – JWL14** | | | |
| **Use Case No.** | JWL18 | **Use Case Version** | 1.0 |
| **Use Case Name** | Add Book To Wish List | | |
| **Author** | Dang Nhat Thien | | |
| **Date** | February 12, 2017 | **Priority** | Normal |
| **Actor:**   * Borrower   **Summary:**   * This use case allows Borrower Add Book To Wish List.   **Goal:**   * The book is saved as wish book of specific Borrower.   **Triggers:** After find out a book, Borrower click on “Add to wish list” button in Book detail.  **Preconditions:**   * Actor logged in system with role Borrower. * Borrower find out the book. [<Borrower> Search Book](#_<Borrower>_Search_Book)   **Post Conditions:**   * **Success:** A dialog will be showed and display message: “Book [title] has been add to your wish list!”. * **Fail:** A dialog will be showed and display message: “Add fail. Please check your internet connection”.   **Main Success Scenario:**   |  |  |  | | --- | --- | --- | | Step | Actor Action | System Response | | 1 | After find out a book, Borrower click on “Add to wish list” button in Book detail. | * System save to database and display success message.   [Exception 1] Lost Internet Connection |   **Alternative Scenario:** N/A  **Exceptions:**   |  |  |  | | --- | --- | --- | | No | Cause | System Response | | 1 | Lost Internet Connection | System show error message via Dialog. |   **Relationships:** [<Borrower> Search Book](#_<Borrower>_Search_Book)  **Business Rules:** N/A | | | |

###### <Borrower> Get Borrowed Books



Figure 27: <Borrower> Get Borrowed Books

|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE – JWL15** | | | |
| **Use Case No.** | JWL19 | **Use Case Version** | 1.0 |
| **Use Case Name** | Get Borrowed Books | | |
| **Author** | Dang Nhat Thien | | |
| **Date** | February 12, 2017 | **Priority** | Normal |
| **Actor:**   * Borrower.   **Summary:**   * This use case allows Borrower Get Borrowed Books.   **Goal:**   * Borrower can get a list borrowed books.   **Triggers:** Borrower click on “Show borrowed books” item on Left Slide Menu.  **Preconditions:**   * Actor logged in system with role Borrower.   **Post Conditions:**   * **Success:** List borrowed books is displayed. * **Fail:** A dialog will be showed and display message: “Loading fail. Please check your internet connection”.   **Main Success Scenario:**   |  |  |  | | --- | --- | --- | | Step | Actor Action | System Response | | 1 | Borrower click on “Show borrowed books” item on Left Slide Menu. | System show list borrowed books.  [Exception 1] Lost Internet Connection. |   **Alternative Scenario:** N/A  **Exceptions:**   |  |  |  | | --- | --- | --- | | No | Cause | System Response | | 1 | Lost Internet Connection | System show error message via Dialog. |   **Relationships:** N/A  **Business Rules:** N/A | | | |

###### <Borrower> Extend Borrowed Book Deadline



Figure 28: <Borrower> Extend Borrowed Book Deadline

|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE – JWL16** | | | |
| **Use Case No.** | JWL20 | **Use Case Version** | 1.0 |
| **Use Case Name** | Extend Borrowed Book Deadline | | |
| **Author** | Dang Nhat Thien | | |
| **Date** | February 12, 2017 | **Priority** | Normal |
| **Actor:**   * Borrower   **Summary:**   * This use case allows Borrower Extend Borrowed Book Deadline   **Goal:**   * Borrower can Extend Borrowed Book Deadline.   **Triggers:** N/A  **Preconditions:**   * Actor logged in system with role Borrower.   **Post Conditions:**   * **Success:** A dialog will be showed and display message: “Book [title] has been extends to [deadline]!” * **Fail:** A dialog will be showed and display message: “Loading fail. Please check your internet connection!”.   **Main Success Scenario:**   |  |  |  | | --- | --- | --- | | Step | Actor Action | System Response | | 1 | After get list borrowed books, Borrower click on “Extends” button in Book detail. | System return new deadline and show success message.  [Exception 1] Lost Internet Connection. |   **Alternative Scenario:** N/A  **Exceptions:**   |  |  |  | | --- | --- | --- | | No | Cause | System Response | | 1 | Lost Internet Connection | System display error message. |   **Relationships:** [<Borrower> Get Borrowed Books](#_<Borrower>_Get_Borrowed)  **Business Rules:** N/A | | | |

##### Automatic Handler

###### <Automatic Handler> Overview Use Case

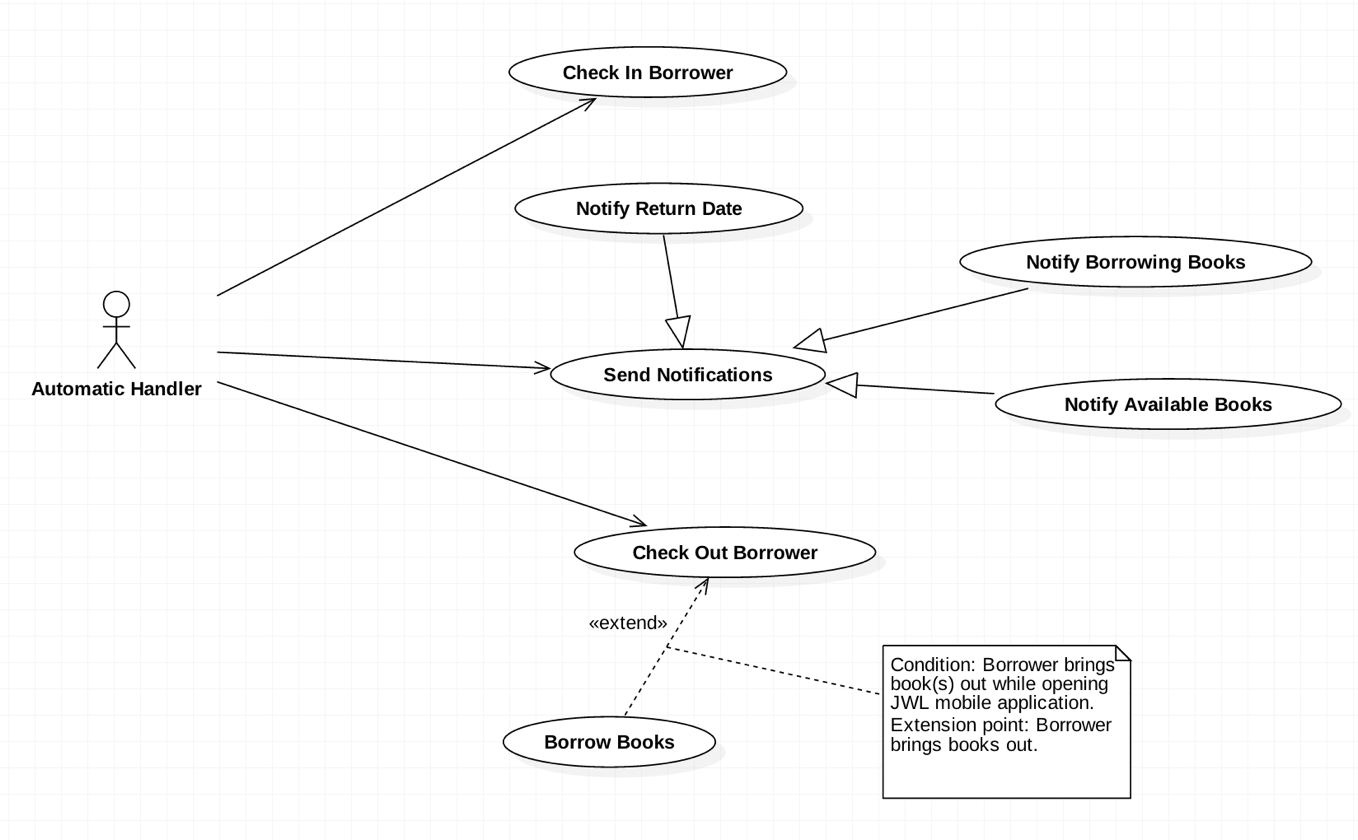


Figure 29: <Automatic Handler> Overview Use Case

###### <Automatic Handler> Check in Borrower

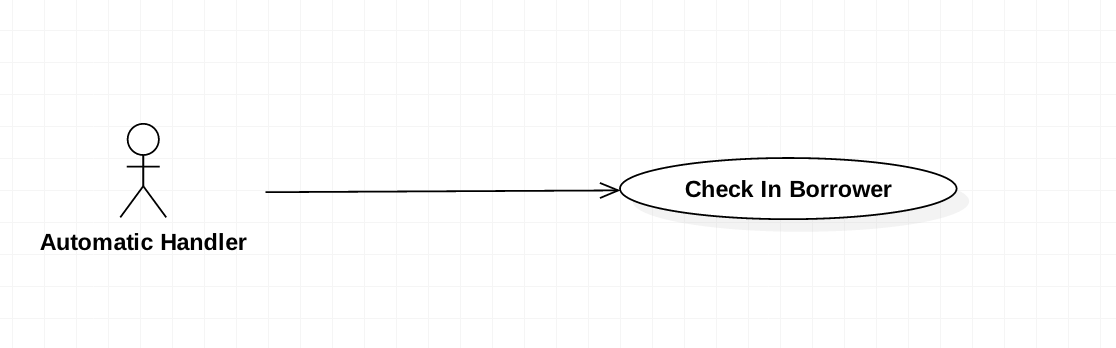


Figure 30: <Automatic Handler> Check in Borrower

|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE – JWL02** | | | |
| **Use Case No.** | JWL21 | **Use Case Version** | 1.0 |
| **Use Case Name** | Check in Borrower | | |
| **Author** | Vo Hong Ha | | |
| **Date** | February 13, 2017 | **Priority** | High |
| **Actor:**   * Automatic Handler   **Summary:**   * This use case allows Automatic Handler to check information of borrowers and confirm their attendance.   **Goal:**   * Authorized Users can enter the library.   **Triggers:**   * Automatic Handler checks the information of borrowers.   **Preconditions:**   * N/A.   **Post Conditions:**   * **Success:** The bell will ring. * **Fail:** N/A   **Main Success Scenario:**   |  |  |  | | --- | --- | --- | | Step | Actor Action | System Response | | 1 | Automatic Handler check the information. | System response the message of result. |   **Alternative Scenario:** N/A  **Exceptions:** N/A  **Relationships:** N/A  **Business Rules:** N/A | | | |

###### <Automatic Handler> Notify return date

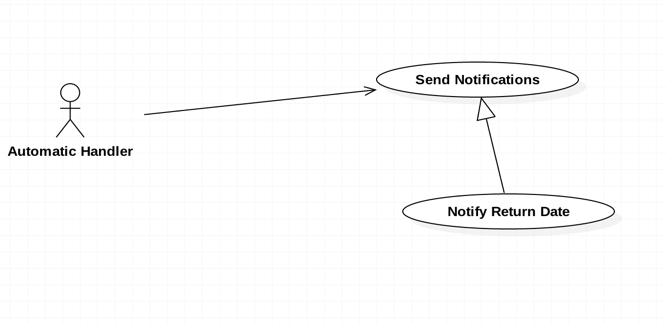


Figure 31: <Automatic Handler> Notify return date

|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE – JWL02** | | | |
| **Use Case No.** | JWL22 | **Use Case Version** | 1.0 |
| **Use Case Name** | Notify return date | | |
| **Author** | Vo Hong Ha | | |
| **Date** | February 13, 2017 | **Priority** | Normal |
| **Actor:**   * Automatic Handler   **Summary:**   * This use case allows Automatic Handler to send a notification to borrower about the return date of books   **Goal:**   * Borrowers are noticed 5 days prior to the return date.   **Triggers:**   * Automatic Handler notices the return date.   **Preconditions:**   * 5 days remaining.   **Post Conditions:**   * **Success:** Notification will be sent. * **Fail:** N/A   **Main Success Scenario:**   |  |  |  | | --- | --- | --- | | Step | Actor Action | System Response | | 1 | Automatic Handler send notification on noticed day. | System push a notification on the application of mobile phone:  “5 days remaining for [bookTitle]” |   **Alternative Scenario:** N/A  **Exceptions:** N/A  **Relationships:** N/A  **Business Rules:** The second notification will be sent on the return date in the case that the book was not returned until that date. | | | |

###### <Automatic Handler> Notify Borrowing Books

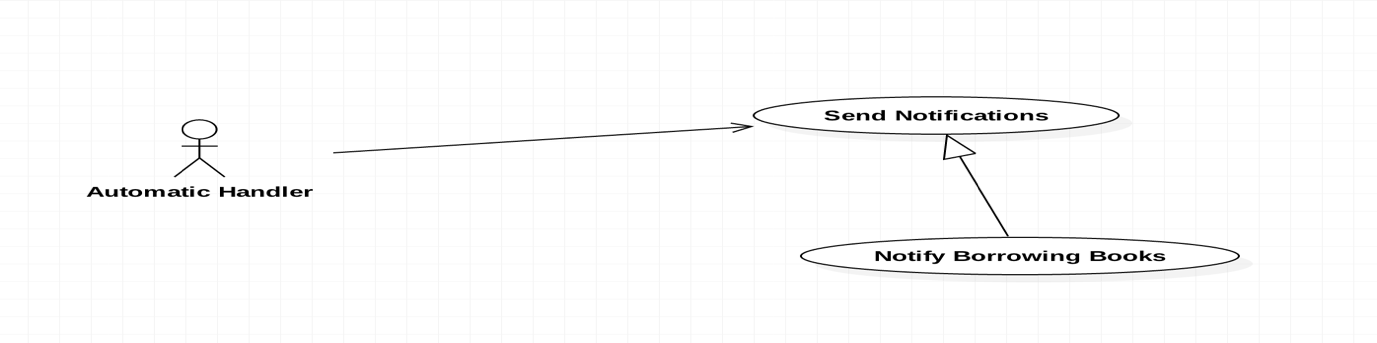


Figure 32: <Automatic Handler> Notify Borrowing Books

|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE – JWL02** | | | |
| **Use Case No.** | JWL23 | **Use Case Version** | 1.0 |
| **Use Case Name** | Notify Borrowing Books | | |
| **Author** | Vo Hong Ha | | |
| **Date** | February 13, 2017 | **Priority** | Normal |
| **Actor:**   * Automatic Handler   **Summary:**   * This use case allows Automatic Handler to send a notification to borrowers about the borrowed book after they borrow   **Goal:**   * Borrowers are noticed those books which they have borrowed   **Triggers:**   * Automatic Handler notices the list of borrowed book   **Preconditions:**   * Borrower check out successfully.   **Post Conditions:**   * **Success:** Notification will be sent with a list of borrowed book * **Fail:** N/A   **Main Success Scenario:**   |  |  |  | | --- | --- | --- | | Step | Actor Action | System Response | | 1 | Automatic Handler send notification right after Borrower check out. | System push a notification on the application of mobile phone:  “You have check out successfully. The list of book is as follow.” |   **Alternative Scenario:** In the case the borrowers don’t check out with books, the notification will not be sent.  **Exceptions:** N/A  **Relationships:** N/A  **Business Rules:** N/A | | | |

###### <Automatic Handler> Notify Available Books

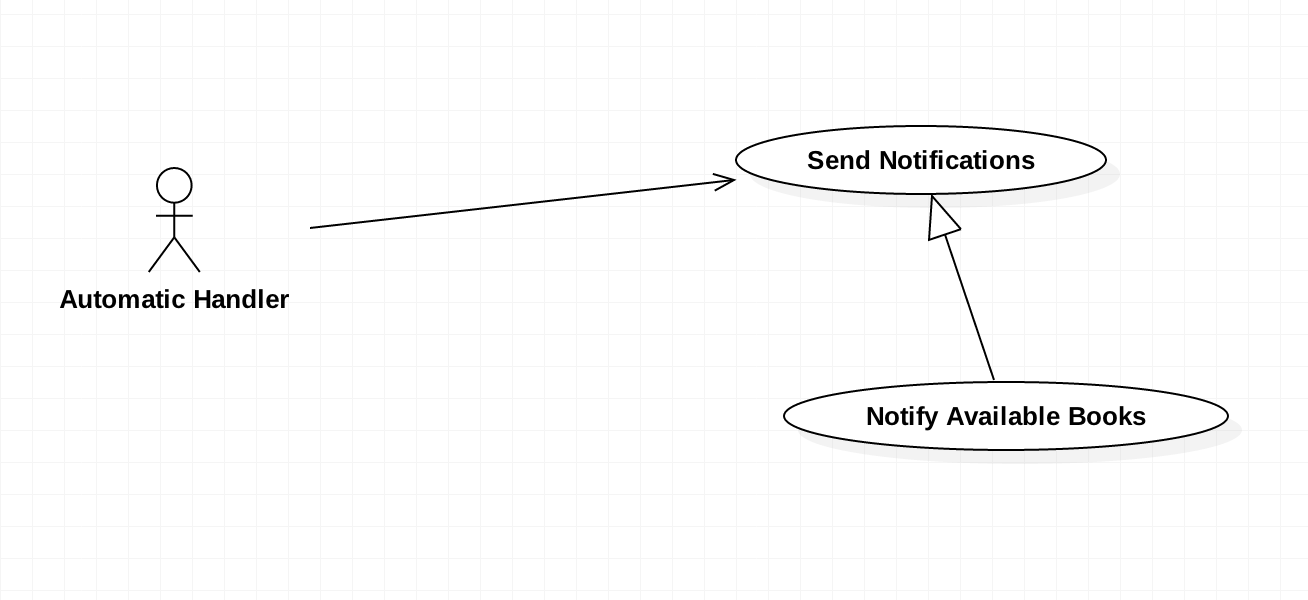


Figure 33: <Automatic Handler> Notify Available Books

|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE – JWL02** | | | |
| **Use Case No.** | JWL24 | **Use Case Version** | 1.0 |
| **Use Case Name** | Notify Available Books | | |
| **Author** | Vo Hong Ha | | |
| **Date** | February 13, 2017 | **Priority** | Normal |
| **Actor:**   * Automatic Handler   **Summary:**   * This use case allows Automatic Handler to send a notification to borrowers about the available book according to borrower’s wish list.   **Goal:**   * Borrowers are noticed those books which are available in the library.   **Triggers:**   * Automatic Handler notices the available books.   **Preconditions:**   * Borrowers have add that books to their wish list and those books are available on the library.   **Post Conditions:**   * **Success:** Notification about the available books will be sent. * **Fail:** N/A   **Main Success Scenario:**   |  |  |  | | --- | --- | --- | | Step | Actor Action | System Response | | 1 | Automatic Handler send notification about available books. | System push a notification on the application of mobile phone:  “Some books of your wish list have available” |   **Alternative Scenario:** N/A  **Exceptions:** N/A  **Relationships:** N/A  **Business Rules:** N/A | | | |

###### <Automatic Handler> Check out Borrower

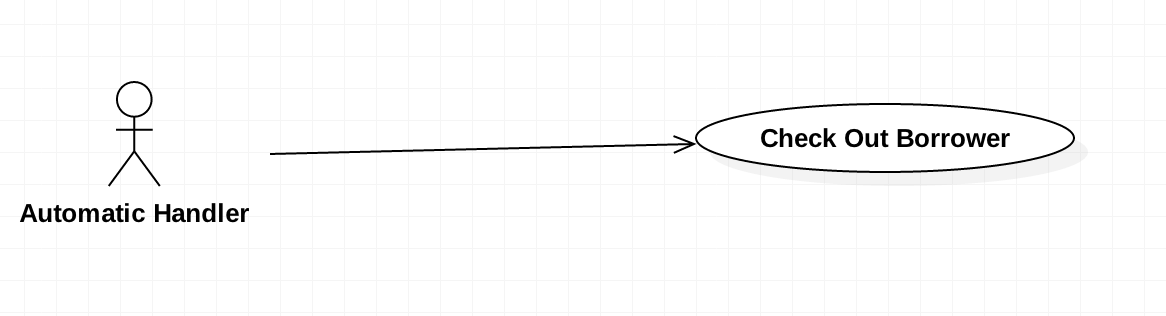


Figure 34: <Automatic Handler> Check out Borrower

|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE – JWL02** | | | |
| **Use Case No.** | JWL25 | **Use Case Version** | 1.0 |
| **Use Case Name** | Check out Borrower | | |
| **Author** | Vo Hong Ha | | |
| **Date** | February 13, 2017 | **Priority** | High |
| **Actor:**   * Automatic Handler   **Summary:**   * This use case allows Automatic Handler to check out the borrower.   **Goal:**   * Check the borrowers who get out of the library.   **Triggers:**   * N/A.   **Preconditions:**   * N/A.   **Post Conditions:**   * **Success:** N/A. * **Fail:** N/A   **Main Success Scenario:**   |  |  |  | | --- | --- | --- | | Step | Actor Action | System Response | | 1 | Automatic Handler check the borrower with RFID reader. | N/A |   **Alternative Scenario:** N/A  **Exceptions:** N/A  **Relationships:** N/A  **Business Rules:** N/A | | | |

###### <Automatic Handler> Check out Borrow Books

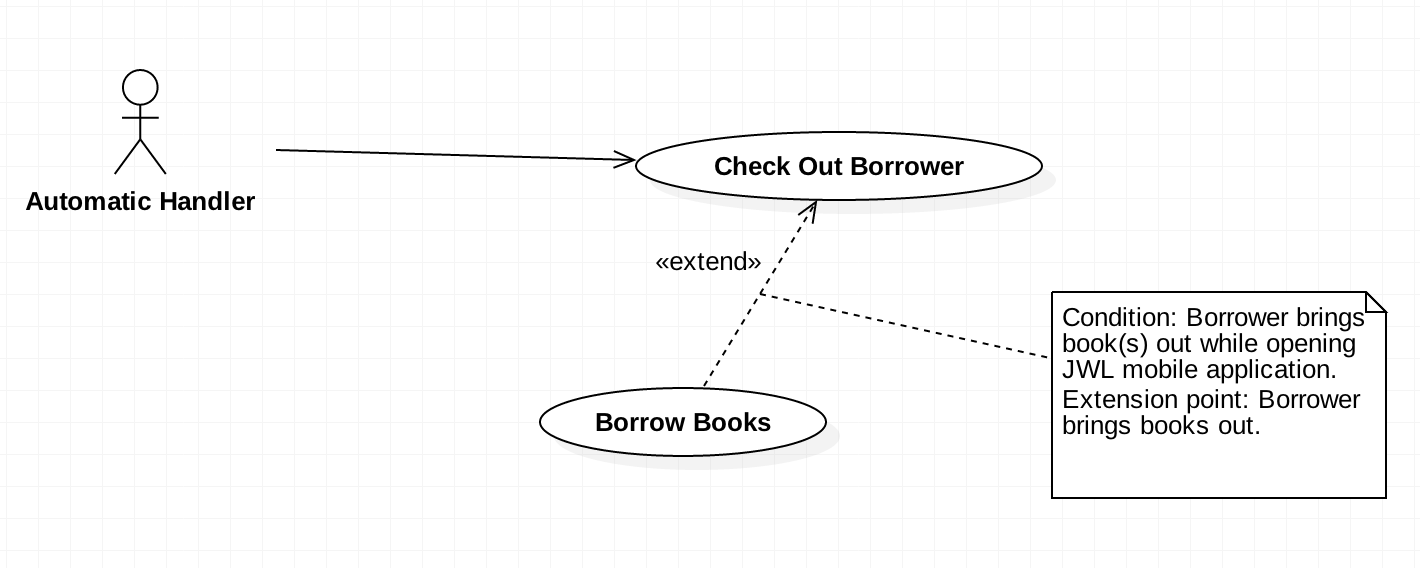


Figure 35: <Automatic Handler> Check out Borrow Books

|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE – JWL02** | | | |
| **Use Case No.** | JWL26 | **Use Case Version** | 1.0 |
| **Use Case Name** | Check out Borrow Books | | |
| **Author** | Vo Hong Ha | | |
| **Date** | February 13, 2017 | **Priority** | High |
| **Actor:**   * Automatic Handler   **Summary:**   * This use case allows Automatic Handler to check out the borrower’s books.   **Goal:**   * Check out the borrower whether they get out with books.   **Triggers:**   * RFID Reader scan the books of the Borrower.   **Preconditions:**   * The books have RFID tag.   **Post Conditions:**   * **Success:** N/A. * **Fail:** N/A   **Main Success Scenario:**   |  |  |  | | --- | --- | --- | | Step | Actor Action | System Response | | 1 | Automatic Handler scan the books with RFID reader. | Add the books to the borrowed list. |   **Alternative Scenario:** N/A  **Exceptions:** N/A  **Relationships:** N/A  **Business Rules:** N/A | | | |

## Software System Attribute



### Usability

* The Android application for emulator requires 10 minutes training for staff.
* The Android application for passenger will take 10-30 minutes to get used to the mobile system completely

### Reliability

* The number of sending notification failure is 1 time per 1000 notifications.
* Timer tasks run at configured time with 100% execution rate.

### Availability

* System is available 24 hours per day and 7 days per week.
* System should take at most 5 hours per month for backup or repairing.

### Security

* Privacy: Each role of user has a specific permission to interact with system.
* System requires SMS verification for all users.
* System always checks authorization and authenticated before doing anything.
* Only admin can grant permission to other roles.

### Maintenanability

* The system is divided into separated modules.
* The code is easy to maintain and upgrade.

### Portability

* The web application is running on Windows Server 2008 or above.
* Mobile application for passenger and emulator runs on Android API greater than 4.1.
* Providing easy installation.

### Performance

* Web appication handles the task within 10 seconds.
* System response time of SMS message depend on telecomunications infrastructurs and server.
* Emulator reads data from NFC card in less than 5 second within the distance between 0 centimeter and 1 centimeter in the condition with no obstacle.

## Conceptual Diagram



Figure 45: Conceptual Diagram

**Data Dictionary:**

|  |  |
| --- | --- |
| Entity Name | Description |

Table 35: Conceptual Diagram Data Dictionary