# Report No. 3 Software Requirement Specification

## User Requirement Specification

### Unauthorized User Requirement

Unauthorized is user does not login to this system. Unauthorized only has two functions.

* Sign up.
* Login.

### Authorized User Requirement

Authorized User is user who has logged into system. Authorized User only has two functions.

* Edit profile.
* Logout.

### Borrower Requirement

Borrower can use some following functions:

* + 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
* Manage books (nhap, tim kiem)

### Admin Requirement

Admin is the user who responsible for managing accounts for the whole system, they can use some following functions:

* Manage accounts

### Emulator Requirement

Emulator is the device which can interact with mobile phone, 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:

* + Suggest promotions
  + Auto extend subscription
  + Crawl bus routes
  + Auto generate server token

## System Requirement Specification



### External Interface Requirement

#### User Interface

* The user interface uses English language in both web admin and mobile application.
* The user interface displays best on 1366x768 resolutions on desktop, and 1080x1920 on mobile.

#### Hardware Interface (RFID, beacon, emulator)

#### Software Interface(API, thu vien support)

#### 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.

### 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** | January 16, 2017 | **Priority** | High |
| **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 role.   **Triggers:**   * Unauthorized User sends Login command.   **Preconditions:**   * Valid username and password are input.   **Post Conditions:**   * **Success:** Unauthorized User is logged into the system as Authorized User. * **Fail:** N/A   **Main Success Scenario:**   |  |  |  | | --- | --- | --- | | Step | Actor Action | System Response | | 2 | Unauthorized User inputs username and password, then sends Login command.  [Exception 1] | Unauthorized User will be logged into System with their specified role.  The system redirects to the role’s view.  [Alternative 1] |   **Alternative Scenario:**  [Alternative 1]   |  |  |  | | --- | --- | --- | | Step | Cause | System Response | | 1 | Unauthorized user leaves the username and password field empty. | System displays an error message. |   **Exception:**  [Exception 1]   |  |  |  | | --- | --- | --- | | Step | Actor Action | System Response | | 1 | Unauthorized User inputs wrong username and password. | System displays an error message. |   **Relationships: N/A**  **Business Rules:**   * System authenticates user by checking the username and password. * Unauthorized User enters password in a hidden text field and the password must be encrypted before sending to server. * After logged in to the system, Unauthorized User will be redirected to a specific view: * If the role is “Admin”, the system will display Admin view. * If the role is “Librarian”, the system will display Librarian view. * If the role is “Borrower”, the system will display Borrower view. | | | |

##### Authorized User

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

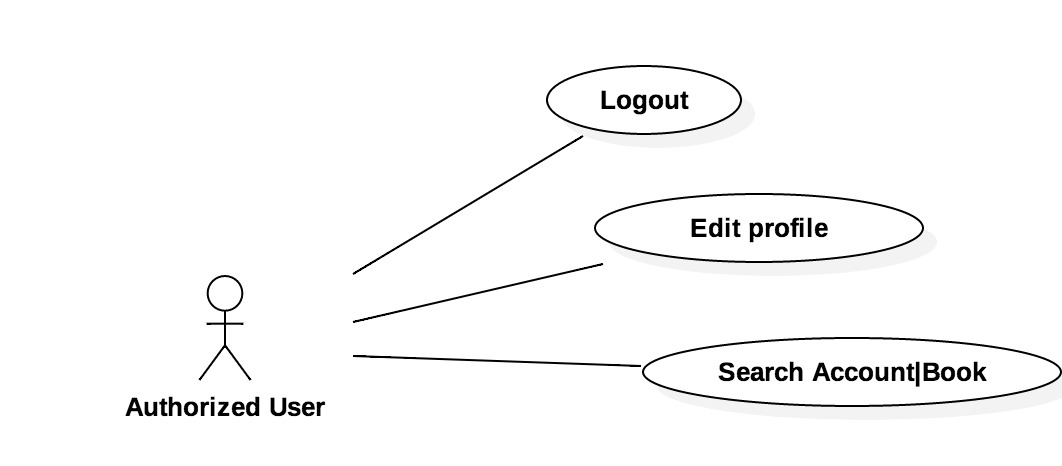


Figure 6: <Authorized User> Overview Use Case

###### <Authorized User> Logout

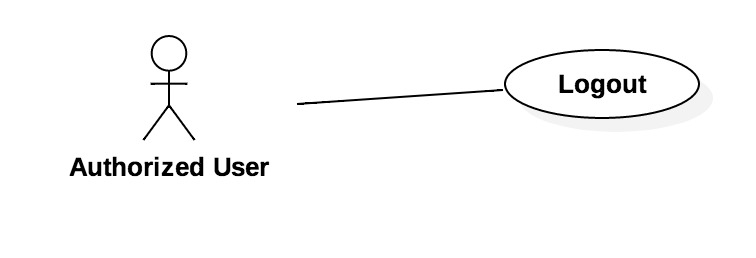


Figure 7: <Authorized User> Logout

|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE – JWL02** | | | |
| **Use Case No.** | JWL02 | **Use Case Version** | 1.0 |
| **Use Case Name** | Logout | | |
| **Author** | Nguyen Tuan Anh | | |
| **Date** | January 16, 2017 | **Priority** | Normal |
| **Actor:**   * Authorized User   **Summary:**   * This use case allows Authorized User to logout.   **Goal:**   * Authorized User can logout of the system.   **Triggers:**   * Authorized User sends Logout command.   **Preconditions:**   * Actor logged in system with role Authorized User.   **Post Conditions:**   * **Success:** Authorized User is logged out. * **Fail:** N/A   **Main Success Scenario:**   |  |  |  | | --- | --- | --- | | Step | Actor Action | System Response | | 1 | Authorized User sends Logout command. | System clears the current session of the user then redirects to login screen. |   **Alternative Scenario:** N/A  **Exceptions:** N/A  **Relationships:** N/A  **Business Rules:**   * After logged out of the system, user no longer has permission to access any authorized functions of the system. | | | |

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

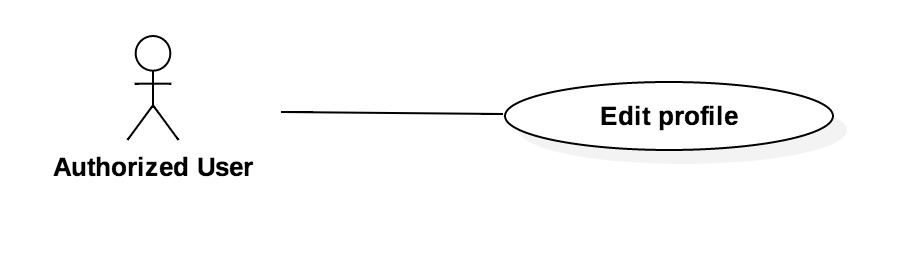


Figure 8: <Authorized User> Edit Profile

|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE – JWL03** | | | |
| **Use Case No.** | JWL03 | **Use Case Version** | 1.0 |
| **Use Case Name** | Edit Profile | | |
| **Author** | Nguyen Tuan Anh | | |
| **Date** | January 16, 2017 | **Priority** | Low |
| **Actor:**   * Authorized User   **Summary:**   * This use case allows Authorized User to edit his/her personal information.   **Goal:**   * Authorized User’s personal information is updated as he/she sends Update command.   **Triggers:**   * Authorized User sends Edit Profile command.   **Preconditions:**   * The Actor must be logged in the System.   **Post Conditions:**   * **Success:** Authorized User’s personal information is updated in the database. * **Fail:** Error message is displayed.   **Main Success Scenario:**   |  |  |  | | --- | --- | --- | | Step | Actor Action | System Response | | 1 | Authorized User sends Edit Profile command. | System lists out the information of the account:   * “Username”: label, not editable * “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”: label Student, Teacher, Officer, …; not editable * “Subscription End Date”: date label, not editable * Update command. | | 2 | Authorized User updates information. |  | | 3 | Authorized User sends Update 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: N/A**  **Exception:**   |  |  |  | | --- | --- | --- | | No | Cause | System Response | | 1 | Authorized User inputs Full Name field with wrong format. | System displays error message: “Full name length is 6-100 characters.” | | 2 | Authorized User inputs Password field with wrong format. | System displays error message: “Password length is 6-100 characters.” | | 3 | Authorized User inputs wrong Confirm Password field. | System displays error message: “Confirm Password must match Password.” | | 4 | Authorized User inputs Email field with wrong format. | System displays error message: “Wrong Email format. Please input something like abc@somemail.com” | | 5 | Authorized User inputs Address field with wrong format. | System displays error message: “Address length is 6-100 characters.” | | 6 | Authorized User inputs Phone Number field with wrong format. | System displays error message: “Phone Number should be all numbers at 8-12 length.” | | 7 | Authorized User 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:**   * User updates account profile in case of changing their personal information. * Password must be encrypted before saving to the system. * User cannot update Username, Account Type, and Subscription End Date. | | | |

###### <Authorized User> Search Account/Book

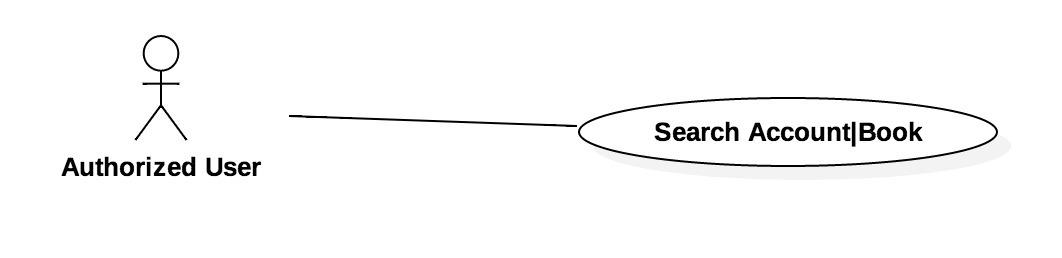


Figure 9: <Authorized User> Search Account/Book

|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE – JWL04** | | | |
| **Use Case No.** | JWL04 | **Use Case Version** | 1.0 |
| **Use Case Name** | Search | | |
| **Author** | Nguyen Tuan Anh | | |
| **Date** | February 11, 2017 | **Priority** | Low |
| **Actor:**   * Authorized User   **Summary:**   * This use case allows Authorized User to search Account/Book.   **Goal:**   * Authorized User’s can get list of Account/Book that match his/her input.   **Triggers:**   * Authorized User sends Search command.   **Preconditions:**   * The Actor must be logged in the System. * The Actor can only search Account with role Admin or Librarian. * The Actor can search Book with all authorized roles.   **Post Conditions:**   * **Success:** List of Account/Book is showed based on the Actor’s search criteria. * **Fail:** N/A.   **Main Success Scenario:**   |  |  |  | | --- | --- | --- | | Step | Actor Action | System Response | | 1 | Authorized User finds the search input above the Account table or the Book table (depends on the Actor’s current view). |  | | 2 | Authorized User changes text in search input and sends Search command. | System loads the search results with their information:  Account:   * UserId * Fullname * Email * Address * Date of Birth * Phone Number * Place of Work * Role * Is in library * Is activated   Book:   * Title * Publisher * Published year * Number of Pages * Number of Copies   [Alternative 1] |   **Alternative Scenario:**  [Alternative 1]   |  |  |  | | --- | --- | --- | | Step | Cause | System Response | | 1 | There is no Book/Account matches search criteria. | System shows message: “Cannot find searched Book/Account.” |   **Exception: N/A**  **Relationships: N/A**  **Business Rules:**   * Each Account has a unique userId. * Account is searched by userId, and Book is searched by title. * Only Admin and Librarian can search for Account. * Book can be search by all Authorized User. | | | |

#### Web Application

##### Admin

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

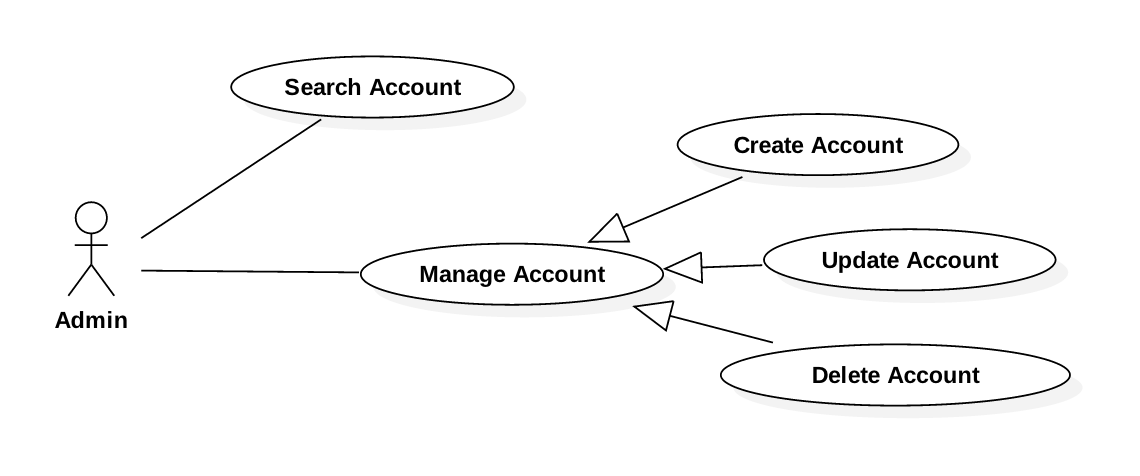


Figure 9: <Admin> Overview Use Case

###### <Admin> Search Account

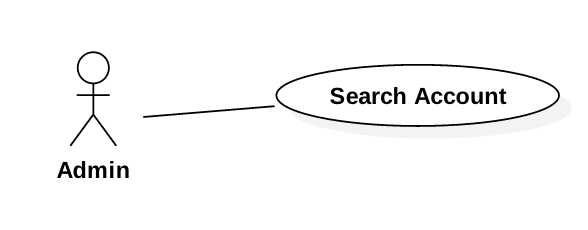


Figure 10: <Admin> Search Account

|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE – JWL04** | | | |
| **Use Case No.** | JWL04 | **Use Case Version** | 1.0 |
| **Use Case Name** | Search Account | | |
| **Author** | Nguyen Tuan Anh | | |
| **Date** | January 16, 2017 | **Priority** | Normal |
| **Actor:**   * Admin   **Summary:**   * This use case allows Admin to search for an account.   **Goal:**   * Admin can get a list accounts that match input username or full name.   **Triggers:**   * Admin changes the content of the search field.   **Preconditions:**   * Actor logged in system with role Admin.   **Post Conditions:**   * **Success:** List of matching accounts based on search term is showed. * **Fail:** N/A   **Main Success Scenario:**   |  |  |  | | --- | --- | --- | | Step | Actor Action | System Response | | 1 | Admin inputs wanted username or full name. | System loads the result each time the input search field changes. Information for each account:   * Username * Full name * Date of Birth * Phone No. * Status (active/deactivated)   [Alternative 1] |   **Alternative Scenario:**   |  |  |  | | --- | --- | --- | | Step | Cause | System Response | | 1 | There is no account that matches the search term. | System shows message: “No account found.” |   **Exceptions:** N/A  **Relationships:** N/A  **Business Rules:**   * Each account has a unique username, such as student ID, or identity ID. | | | |

###### <Admin> Create Account

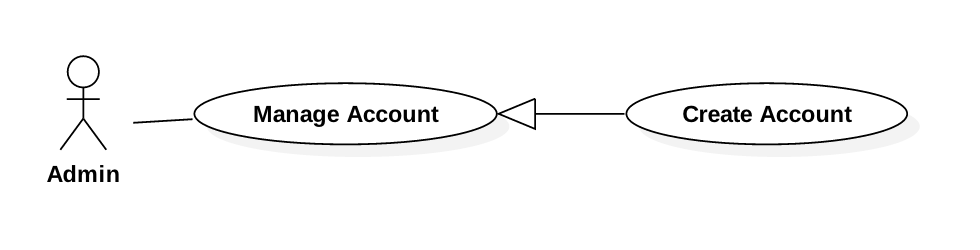


Figure 11: <Admin> Create Account

|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE – JWL05** | | | |
| **Use Case No.** | JWL05 | **Use Case Version** | 1.0 |
| **Use Case Name** | Create Account | | |
| **Author** | Nguyen Tuan Anh | | |
| **Date** | January 16, 2017 | **Priority** | Normal |
| **Actor:**   * Admin   **Summary:**   * This use case allows Admin to create an account.   **Goal:**   * Admin can create a new account for another user (Borrower/Librarian).   **Triggers:**   * Admin sends Create Account command.   **Preconditions:**   * Actor logged in system with role Admin.   **Post Conditions:**   * **Success:** A new account with input information is added to the System database. * **Fail:** N/A   **Main Success Scenario:**   |  |  |  | | --- | --- | --- | | Step | Actor Action | System Response | | 1 | Admin sends Create Account 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 | Authorized User input the information. |  | | 3 | Authorized User sends Submit command. | * System validates the input then create a new account. * System displays a successful message: “New Account for [username] is added.”   [Exception 1, 2, 3, 4, 5, 6, 7] |   **Alternative Scenario:** N/A  **Exceptions:**   |  |  |  | | --- | --- | --- | | No | Cause | System Response | | 1 | Admin inputs Username field with wrong format. | System displays error message: “Full name length is 6-50 characters.” | | 2 | Admin inputs Full Name field with wrong format. | System displays error message: “Full name length is 6-100 characters.” | | 3 | Admin inputs Password field with wrong format. | System displays error message: “Password length is 6-50 characters.” | | 4 | Admin inputs wrong Confirm Password field. | System displays error message: “Confirm Password must match Password.” | | 5 | Admin inputs Email field with wrong format. | System displays error message: “Wrong Email format. Please input something like abc@somemail.com” | | 6 | Admin inputs Address field with wrong format. | System displays error message: “Address length is 6-100 characters.” | | 7 | Admin inputs Phone Number field with wrong format. | System displays error message: “Phone Number should be all numbers at 8-12 length.” | | 8 | Admin 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 account has a unique username, such as student ID, or identity ID. * Password must be encrypted before saving to the system. | | | |

###### <Admin> Update Account

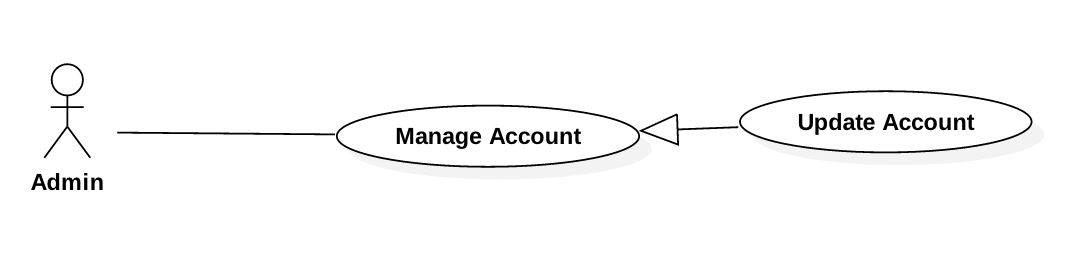


Figure 12: <Admin>Update Account

|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE – JWL06** | | | |
| **Use Case No.** | JWL06 | **Use Case Version** | 1.0 |
| **Use Case Name** | Update Account | | |
| **Author** | Nguyen Tuan Anh | | |
| **Date** | January 16, 2017 | **Priority** | Low |
| **Actor:**   * Admin   **Summary:**   * This use case allows Admin to update another account information.   **Goal:**   * Admin can update another Actor’s personal information as he/she sends Submit command.   **Triggers:**   * Admin sends Update command.   **Preconditions:**   * Actor logged in system with role Admin.   **Post Conditions:**   * **Success:** The account’s personal information is updated in the database. * **Fail:** Error message is displayed.   **Main Success Scenario:**   |  |  |  | | --- | --- | --- | | Step | Actor Action | System Response | | 1 | Admin 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 * Update command. | | 2 | Admin updates information. |  | | 3 | Admin 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: N/A**  **Exception:**   |  |  |  | | --- | --- | --- | | No | Cause | System Response | | 1 | Admin inputs Username field with wrong format. | System displays error message: “Full name length is 6-50 characters.” | | 2 | Admin inputs Full Name field with wrong format. | System displays error message: “Full name length is 6-100 characters.” | | 3 | Admin inputs Password field with wrong format. | System displays error message: “Password length is 6-50 characters.” | | 4 | Admin inputs wrong Confirm Password field. | System displays error message: “Confirm Password must match Password.” | | 5 | Admin inputs Email field with wrong format. | System displays error message: “Wrong Email format. Please input something like abc@somemail.com” | | 6 | Admin inputs Address field with wrong format. | System displays error message: “Address length is 6-100 characters.” | | 7 | Admin inputs Phone Number field with wrong format. | System displays error message: “Phone Number should be all numbers at 8-12 length.” | | 8 | Admin 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:**   * Password must be encrypted before saving to the system. * Each account has a unique username, such as student ID, or identity ID. | | | |

###### <Admin> Delete Account

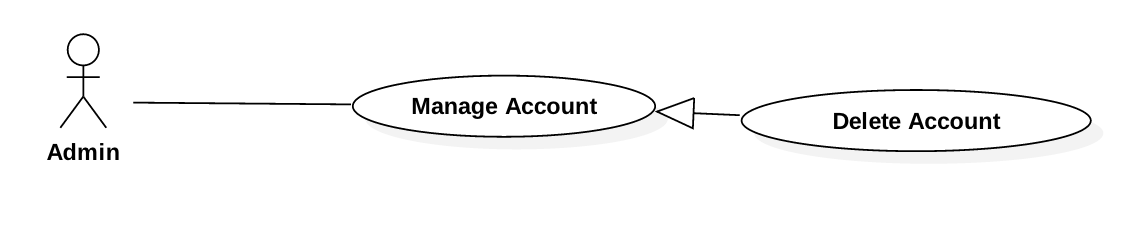


Figure 13: <Admin> Delete Account

|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE – JWL07** | | | |
| **Use Case No.** | JWL07 | **Use Case Version** | 1.0 |
| **Use Case Name** | Delete Account | | |
| **Author** | Nguyen Tuan Anh | | |
| **Date** | January 16, 2017 | **Priority** | Low |
| **Actor:**   * Admin   **Summary:**   * This use case allows Admin to delete another account.   **Goal:**   * Admin can delete any other account.   **Triggers:**   * Admin sends Update command.   **Preconditions:**   * Actor logged in system with role Admin.   **Post Conditions:**   * **Success:**   + The account’s delete date is updated in the database.  + The account can no longer be viewed or do any functions of the System.   * **Fail:** Error message is displayed.   **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:   * “Username”: label * “Full name”: label * “Date of Birth”: label * “Phone number”: label * “Account Type”: label * System asks for confirmation. | | 2 | Admin sends Confirm command.  [Alternative 1] | * System update the account’s delete date in the database. * System displays a successful message: “the Account [username] is Successfully deleted” |   **Alternative Scenario:**   |  |  |  | | --- | --- | --- | | No | Cause | System Response | | 1 | Admin sends Cancel command. | System closes the Delete Account pop up. |   **Exception: N/A**  **Relationships: N/A**  **Business Rules:**   * Deleted account’s record is still kept in the database, only the delete date is updated. * Deleted account cannot be viewed or do anything function of the System. | | | |

##### Librarian

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

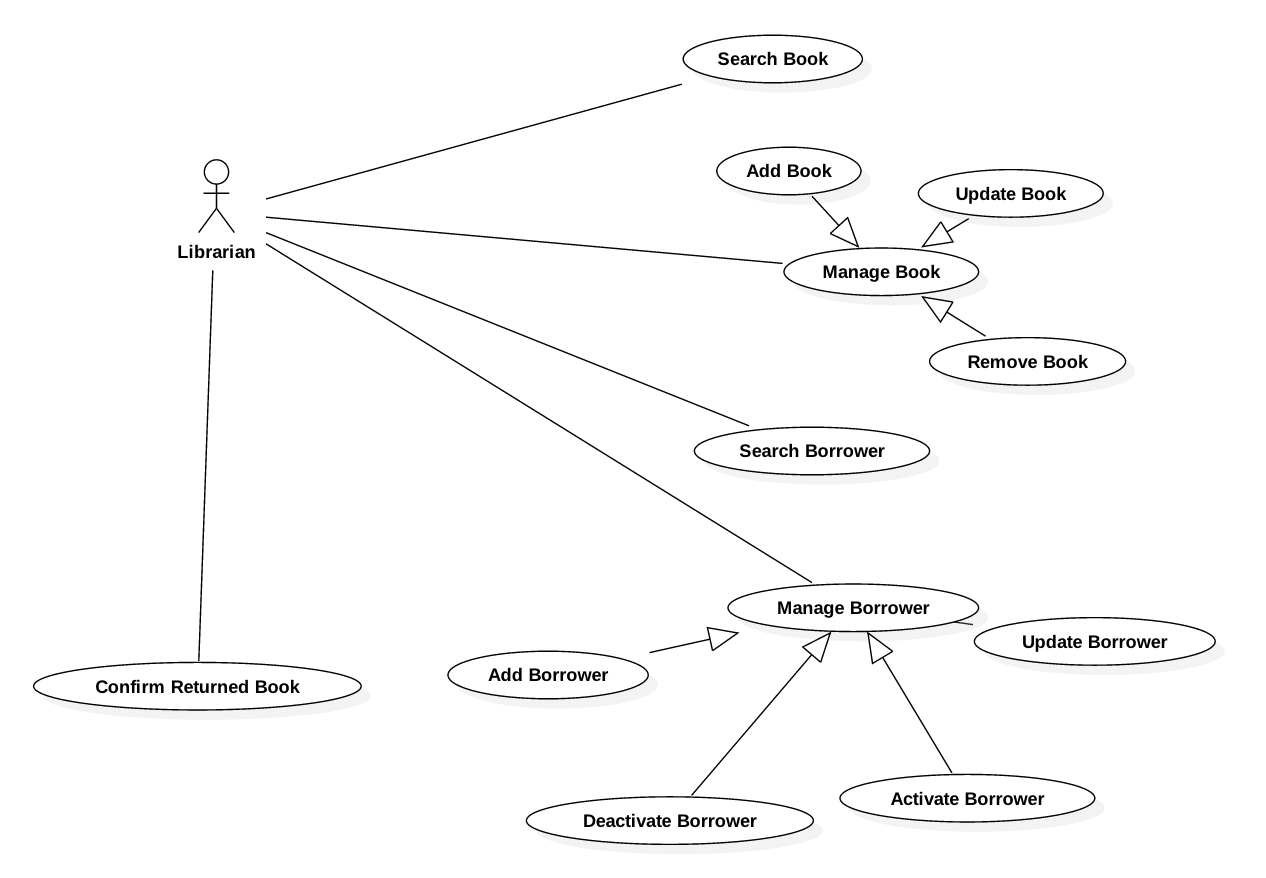


Figure 14: <Librarian> Overview Use Case

###### <Librarian> Search Book



Figure 15: <Librarian> Search Book

###### <Librarian> Add Book

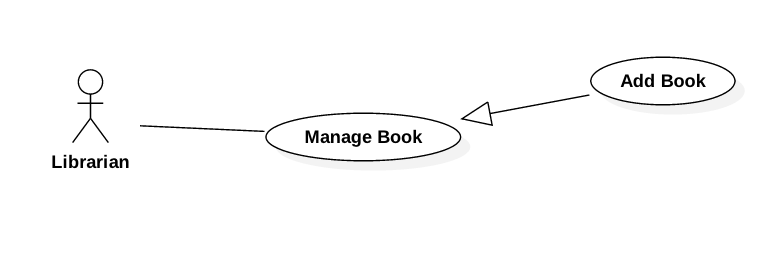


Figure 16: <Librarian> Add Book

###### <Librarian> Update Book

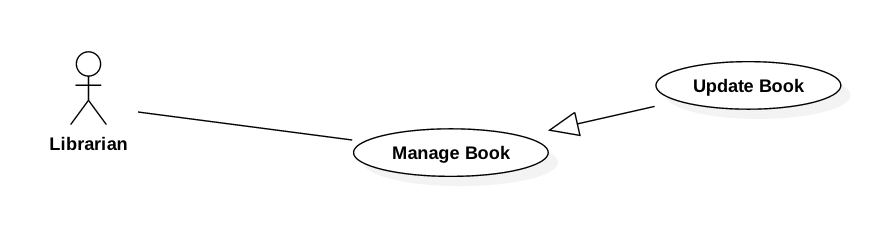


Figure 17: <Librarian> Update Book

###### <Librarian> Remove Book

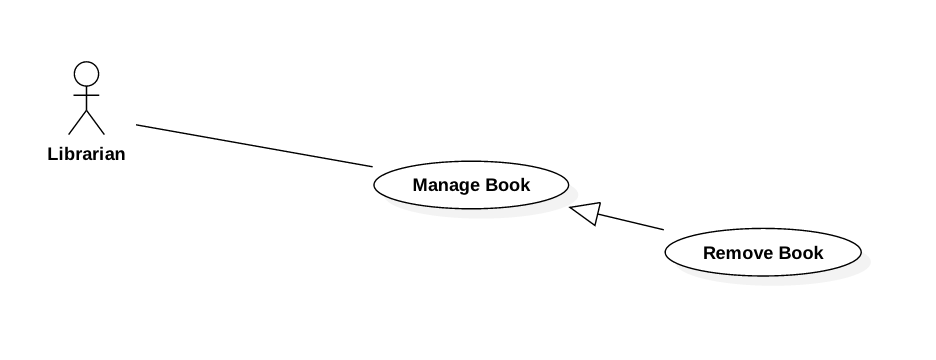


Figure 18: <Librarian> Remove Book

###### <Librarian> Add Borrower



Figure 19: <Librarian> Add Borrower

###### <Librarian> Update Borrower

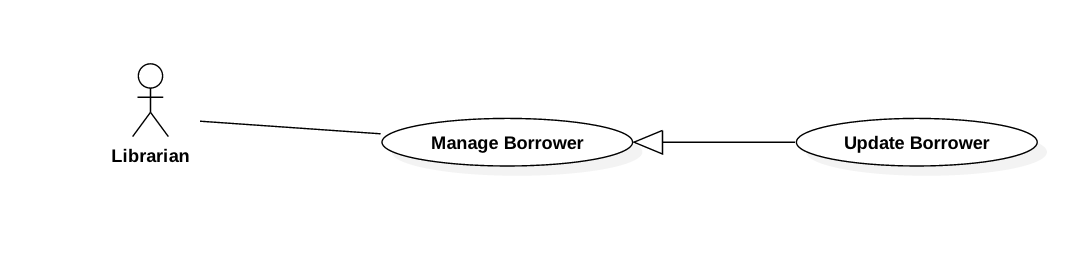


Figure 20: <Librarian> Update Borrower

###### <Librarian> Activate/Deactivate Borrower

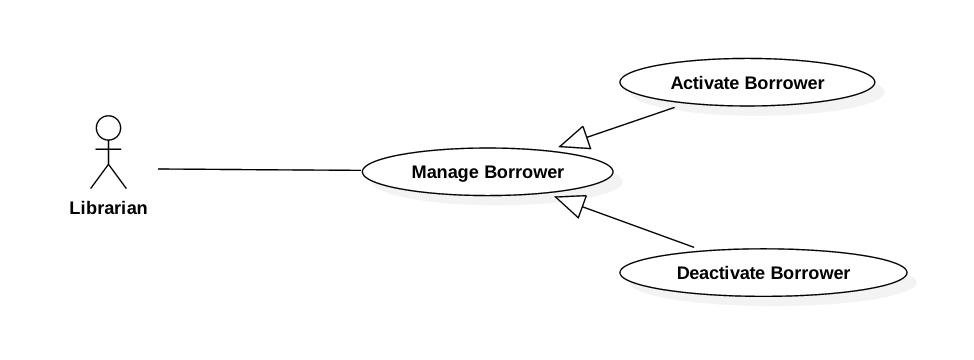


Figure 21: <Librarian> Activate/Deactivate Borrower

###### <Librarian> Search Borrower



Figure 22: <Librarian> Search Borrower

###### <Librarian> Confirm Returned Book

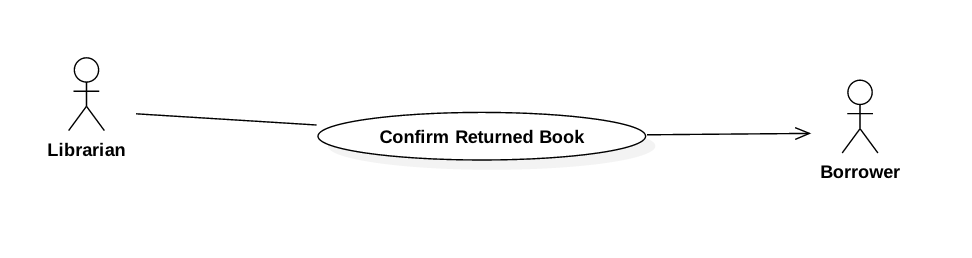


Figure 23: <Librarian> Confirm Returned Book

#### Mobile Application

##### Borrower

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

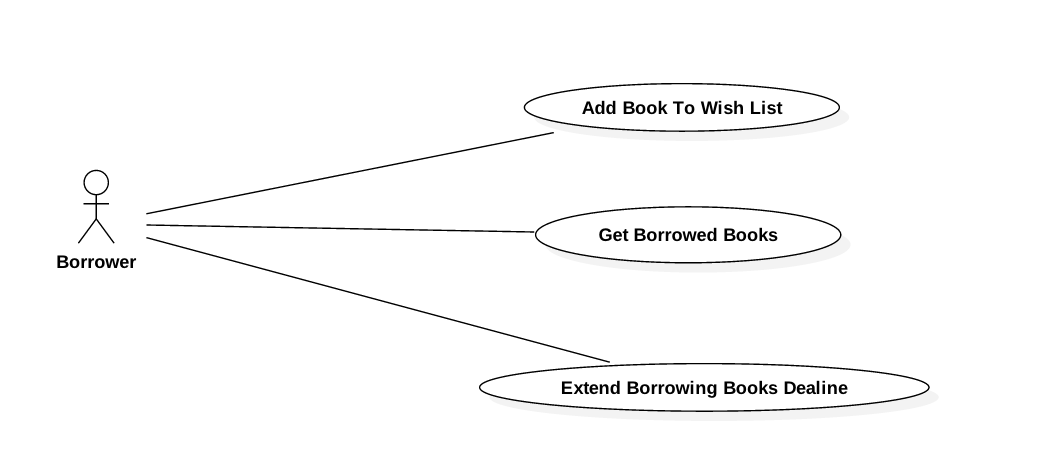


Figure 24: <Borrower> Overview Use Case

###### <Borrower> Search Book



Figure 25: <Borrower> Search Book

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



Figure 26: <Borrower> Add Book To Wish List

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



Figure 27: <Borrower> Get Borrowed Books

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



Figure 28: <Borrower> Extend Borrowed Book Deadline

##### Automatic Handler

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

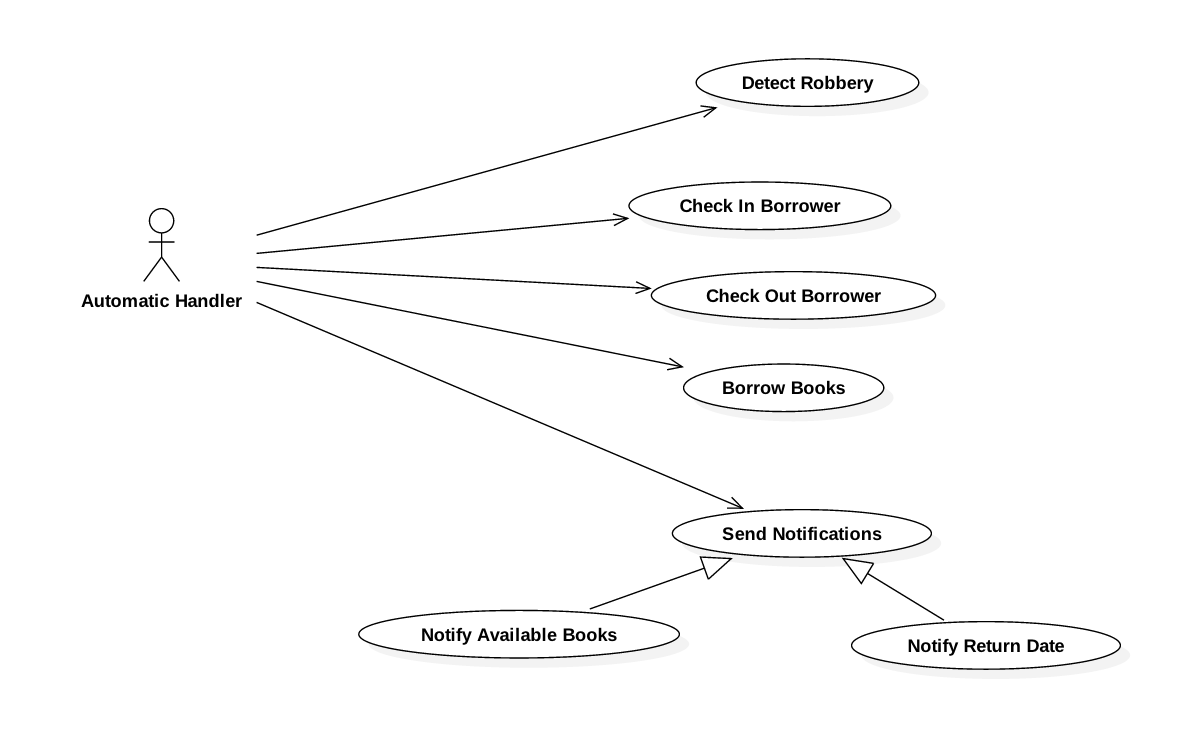


Figure 29: <Automatic Handler> Overview Use Case

###### <Automatic Handler> Detect Robbery

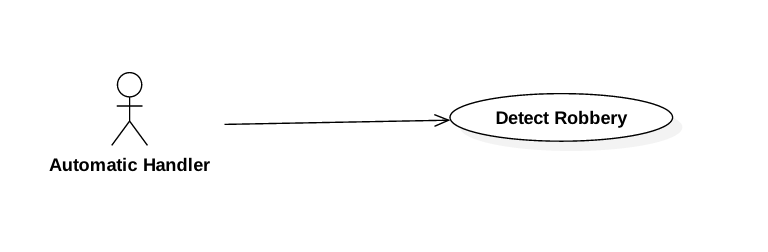


Figure 30: <Automatic Handler> Detect Robbery

###### <Automatic Handler> Check In Borrower

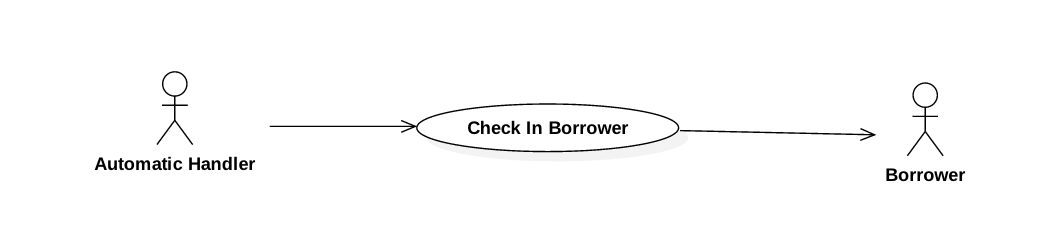


Figure 31: <Automatic Handler> Check In Borrower

###### <Automatic Handler> Check Out Borrower

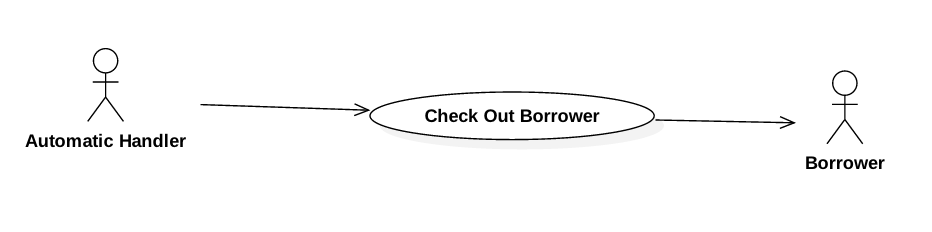


Figure 32: <Automatic Handler> Check Out Borrower

###### <Automatic Handler> Borrow Books

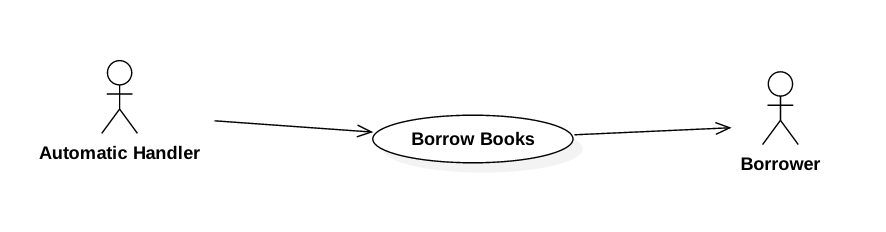


Figure 33: <Automatic Handler> Borrow Books

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

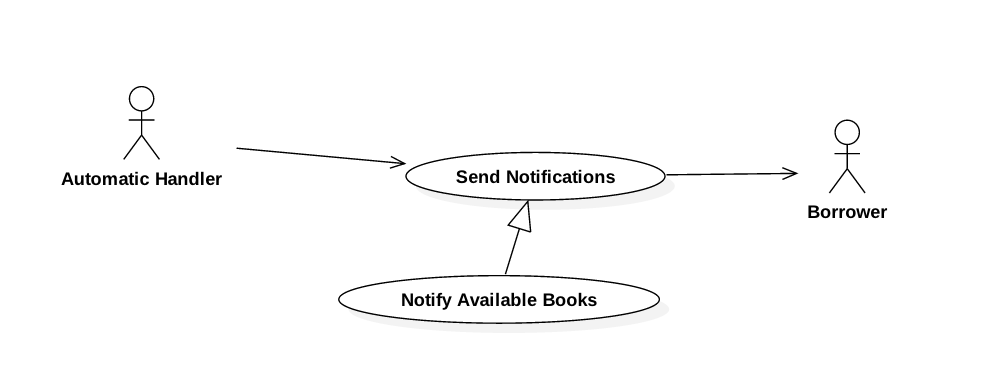


Figure 34: <Automatic Handler> Notify Available Books

###### <Automatic Handler> Notify Return Date

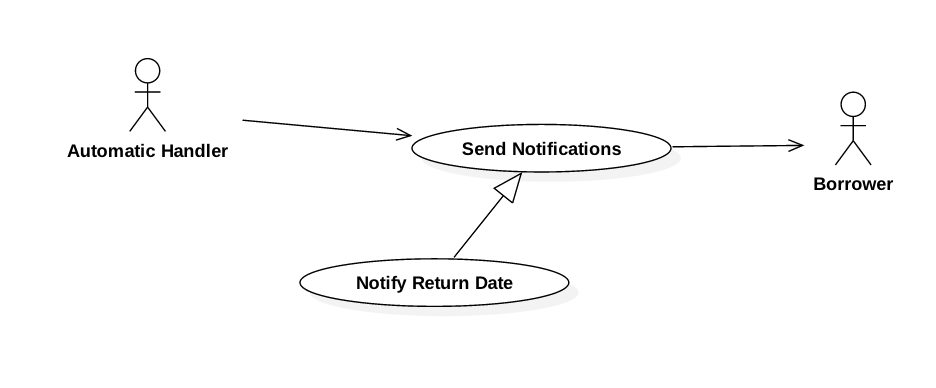


Figure 35: <Automatic Handler> Notify Return Date

## 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

1. 

Figure 45: Conceptual Diagram

**Data Dictionary:**

|  |  |
| --- | --- |
| Entity Name | Description |
| User | Contains all properties about user. |
| Passenger | Contains all properties about passenger info. |
| Admin | Contains all properties about admin info. |
| Staff | Contains all properties about staff info. |
| Manager | Contains all properties about manager info. |
| Card | Contains all properties about NFC card. |
| Ticket Type | Contains all properties about ticket type for based on bus route, each ticket type has diffirent price. |
| Bus Route | Contains all properties about bus route in Ho Chi Minh City. |
| Ticket | Contains all properties about ticket to specify which card buy ticket belongs to which ticket type on which bus route. |
| Credit Plan | Contains all info about credit plan which will be chosen while adding credit to card. |
| Payment Transaction | Contains all info about transaction when adding credit to card via cash, credit plan or scratch card. |
| Promotion | Contains all info about promotion. |
| Scratch Card | Contains all info about scratch card. |
| Offer Subscription | Contains all info about offer subscription. |
| User Subscription | Contains all info about user subscription to specify who subscribe which offer subscription. |

Table 35: Conceptual Diagram Data Dictionary