

Figure 1: Overview Use case

**2.3.1<Guest> Overview Use case**



Figure 2: <Guest> Overview Use case

**2.3.1.1 <Guest>** **Login Use Case Diagram**



Figure 3: <Guest> Overview Use case

|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE – LRA001** | | | |
| **Use Case No.** | LRA001 | **Use Case Version** | 2.0 |
| **Use Case Name** | Login | | |
| **Author** | Dinh Huu Toan | | |
| **Date** | 18/05/2015 | **Priority** | Normal |
| **Actor:**   * Guest.   **Summary:**   * This use case allows guest to login to the system.   **Goal:**   * Allow authentication and authorization of the system.   **Triggers:**   * Guest want to login into system. * To login: Guests go to the login page, enter username and password, and click “Login” button to login.   **Preconditions:** N/A.  **Post Conditions:**   * Success: Guest is authorized, redirect to corresponding page. * Fail: System will show error on current page.   Main Success Scenario:   |  |  |  | | --- | --- | --- | | **Step** | **Actor Action** | **System Response** | | 1 | Guest goes to login page. | System show a login form, includes:   * Username: textbox, min length: 6, max length: 30, required. * Password: textbox, password box, min length: 6, max length: 20, required. * Login button. | | 2 | Guess enter username and password into textboxes.  Press “Login” button. | Guest is logged into system, authorized with corresponding role. Redirect to corresponding page.  [Exception 1,2] |   **Alternative Scenario:** N/A  Exceptions:   |  |  |  | | --- | --- | --- | | **No** | **Actor Action** | **System Response** | | 1 | No input in “Username” or “Password” textboxes | Show error message: “Please input username and password!” | | 2 | Username/Password not in range [5,30] | Show error message: “Invalid username or password! Please try again!” | | 3 | Input invalid “Username” and “Password” | Show error message: “Invalid username or password! Please try again!” |   **Relationships:** N/A  **Business Rules:**   * Only staff and admin can have accounts. * Only active accounts can log in. | | | |

Table1: < Guest > Login

**2.3.1.2 <Guest>** **Serach Laptop Use Case Diagram**



Figure 4: **< Guest >** **Search Laptop Use Case Specification**

|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE – LRA002** | | | |
| **Use Case No.** | LRA002 | **Use Case Version** | 2.0 |
| **Use Case Name** | Search Laptop | | |
| **Author** | Dinh Huu Toan | | |
| **Date** | 18/05/2015 | **Priority** | High |
| **Actor:**   * Guests   **Summary:**   * This use case allows guest to search and view laptops brief information.   **Goal:**   * A list of appropriate laptops will be shown to user.   **Triggers:**   * Guest wants to search for laptops. * Input search text to search textbox then click “Search”.   **Preconditions:**   * N/A.   **Post Conditions:**   * **Success:** A list of appropriate laptops is shown along with laptops’ information: name, review statistic, price, and images. * **Fail:** Show an error message.   **Main Success Scenario:**   |  |  |  | | --- | --- | --- | | Step | Actor Action | System Response | | 1 | Guest inputs the laptop’s name into textbox or they can leave it empty. |  | | 2 | Guest clicks “Search” button | * System will show a list of appropriate laptops as a grid which contains 3 laptops in one row and order by product name. [Alternative 1] * A laptop will show in a square frame with following information: * Laptop’s image * Name * Review statistic (Number of Positive/Negative/Neutral reviews) * Link to detail page |   **Alternative Scenario:**   |  |  |  | | --- | --- | --- | | No | Actor Action | System Response | | 1 | System cannot find searched laptops | * System will show a dialog with message: “This laptop is not available. Would you like to receive information about this laptop when it’s available?”, a textbox for inputting email, and “Submit”/”No, Thanks!” buttons |   **Exceptions:** N/A  **Relationships:** N/A  **Business Rules:**   * Guest can leave the search textbox empty, then all laptops should be listed. * Results will be shown in alphabetical order for easy search | | | |

**Table2: <** **Guest** **> Search Laptop**

**2.3.1.3 <Guest>** **View laptop detail Use Case Diagram**



Figure 5: **< Guest >** **View laptop detail Use Case Specification**

|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE – LRA003** | | | |
| **Use Case No.** | LRA003 | **Use Case Version** | 2.0 |
| **Use Case Name** | View laptop detail | | |
| **Author** | Dinh Huu Toan | | |
| **Date** | 18/05/2015 | **Priority** | High |
| **Actor:**   * Guests   **Summary:**   * This use case allow guests to read laptop full details.   **Goal:**   * Show all details of choosing laptop.   **Triggers:**   * Guest wants to read laptop details includes: name, reviews, specs, price (if available). * Click “Details” link on each result found.   **Preconditions:**   * N/A.   **Post Conditions:**   * **Success:** Navigate to new page shows laptops’ information: name, reviews statistic, price, and images. * **Fail:** Show an error message.   **Main Success Scenario:**   |  |  |  | | --- | --- | --- | | Step | Actor Action | System Response | | 1 | Guest clicks “Detail” link on one result | * Navigate to “Detail” page with following information:   + Name of laptop   + Image slide show   + Statistic: number of Positive/Negative/Neutral reviews   + Specs   + Reviews   + Suggested laptops |   **Alternative Scenario:** N/A  **Exceptions:** N/A  **Relationships:** Search Laptop  **Business Rules:**   * Details of laptop should be organized well so that guest can easily read them. | | | |

**Table 3: <** **Guest** **> View laptop detail**

**2.3.2 <Staff> “Manage parser” Overview use case**

****

**2.3.1.4 <Staff>** **Activate/Deactivate Use Case Diagram**



Figure 6: **<Staff>** **Activate/Deactivate Use Case Specification**

|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE –** LRA004 | | | |
| **Use Case No.** | LRA004 | **Use Case Version** | 2.0 |
| **Use Case Name** | Activate/Deactivate Parser | | |
| **Author** | Dinh Huu Toan | | |
| **Date** | 18/05/15 | **Priority** | High |
| **Actor:**   * Staff.   **Summary:**   * This use case allows staff to activate/deactivate a parser.   **Goal:**   * Selected parser will be activated or deactivate.   **Triggers:**   * Staff wants to activate or deactivate a parser in the system. * Staff clicks “Manage parser” menu, then clicks “Activate” button or “Deactivate” button.   **Preconditions:**   * User has logged in with staff role.   **Post Conditions:**   * **Success**: Selected parser is activated or Deactivated. Show success message. * **Fail:** Show error message.   **Main Success Scenario:**   |  |  |  | | --- | --- | --- | | Step | Actor Action | System Response | | 1 | Staff clicks “Manage parser” menu. | Show “Manage parser” page which includes:   * A table with 4 columns:   + No   + Parsed links   + Status   + Action | | 2 | Staff clicks “Activate” button.  [Alternative 1] | That parser is activated. The button text is changed into “Deactivate”. Show success message.  [Exception 1] |   **Alternative Scenario:**   |  |  |  | | --- | --- | --- | | Step | Actor Action | System Response | | 1 | Staff clicks “Deactivate” button. | That parser is deactivated. The button text is changed into “activate”. Show success message.  [Exception 2] |   **Exceptions:**   |  |  |  | | --- | --- | --- | | Step | Actor Action | System Response | | 1 | Error while activating parser. | Show error message “activating parser fail” | | 2 | Error while deactivating parser. | Show error message “deactivating parser fail” |   **Relationships:** N/A  **Business Rules:** N/A | | | |

**Table 4: <Staff>** **Activate/Deactivate**

2.3.1<Staff> “Train machine” Overview Use case



Figure 1: <Staff> “Train machine” Overview Use case

2.3.1**<Staff>** **Merge laptop Use Case Diagram**



Figure 6: **<Staff>** **Merge laptop Use Case Specification**

|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE –** LRA006 | | | |
| **Use Case No.** | LRA006 | **Use Case Version** | 2.0 |
| **Use Case Name** | Merge Laptop | | |
| **Author** | Dinh Huu Toan | | |
| **Date** | 18/05/15 | **Priority** | High |
| **Actor:**   * Staff   **Summary:**   * This use case allows staff to merge duplicated laptop names.   **Goal:**   * Training machine to merge product names which are duplicated.   **Triggers:**   * Staff wants to train machine to merge product names which are duplicated. * To train machine, staff must do the following steps:   + Click “Manage duplicated items” menu.   + Select products and click “Merge” button   **Preconditions:**   * User must log in the system with staff role.   **Post Conditions:**   * **Success**: Laptops will be saved to the database. Show success message. * **Fail:** Nothing is saved to the database. Show error message.   **Main Success Scenario:**   |  |  |  | | --- | --- | --- | | Step | Actor Action | System Response | | 1 | From the staff’s page, staff clicks “Manage duplicated items” link. | System will get log file and show “Manage duplicated items” page with tables has 3 columns:   * Checkbox. * Laptop names. * Radio button   [Alternative 1] | | 2 | Staff selects products on checkbox and select main product on radio button to merge, then click “Merge” button. | * System will save checked products with products’ ids equal to product’s id of main product. * System will show a message: “Merge successfully”.   [Alternative 2] [Exceptions 1] |   **Alternative Scenario:**   |  |  |  | | --- | --- | --- | | No | Actor Action | System Response | | 1 | Log file is empty. | System will show a message: “No data found”. | | 2 | Staff selects products on checkbox and select main product on radio button to merge, then click “Merge” button. | * If product does not exist in the database, then add new dictionary. * System will show a message: “Merge successfully”.   [Exceptions 1] |   **Exceptions:**   |  |  |  | | --- | --- | --- | | No | Actor Action | System Response | | 1 | Fail to insert or update to database. | Display error message: “An error occurred. Please try again later!” |   **Relationships:**   * Auto parse data * Force parse data   **Business Rules:**   * A laptop can have many names. Merging should base on the most common name | | | |

**Table 4: <Staff>** **Merge laptop**

2.3.1**<Staff>** **Merge laptop Use Case Diagram**

****

Figure 6: **<Staff>** **Split laptop Use Case Specification**

|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE – LRA007** | | | |
| **Use Case No.** | LRA007 | **Use Case Version** | 2.0 |
| **Use Case Name** | Split products | | |
| **Author** | Dinh Huu Toan | | |
| **Date** | 18/05/15 | **Priority** | High |
| **Actor:**   * Staff   **Summary:**   * This use case allows staff to split duplicated laptop names.   **Goal:**   * Training machine to split product names which are duplicated.   **Triggers:**   * Staff wants to train machine to split product names which are duplicated. * To train machine, staff must do the following steps:   + Click “Manage duplicated items” menu.   + Select products and click “Split” button   **Preconditions:**   * User must log in the system with staff role.   **Post Conditions:**   * **Success**: Laptops will be saved to the database. Show success message. * **Fail:** Nothing is saved to the database. Show error message.   **Main Success Scenario:**   |  |  |  | | --- | --- | --- | | Step | Actor Action | System Response | | 1 | From the staff’s page, staff clicks “Manage duplicated items” link. | System will get log file and show “Manage duplicated items” page with tables has 3 columns:   * Checkbox. * Laptop Names. * Radio button   [Alternative 1] | | 2 | Staff selects products on checkbox to split, then click “Split” button. | * System will add new product to database. * System will show a message: “Split successfully”.   [Alternative 2] [Exceptions 1] |   **Alternative Scenario:**   |  |  |  | | --- | --- | --- | | Step | Actor Action | System Response | | 1 | Log file is empty. | System will show a message: “No data found” | | 2 | Staff selects products on checkbox to split, then click “Split” button. | If product is already in the database, update its information. [Exceptions 1] |   **Exceptions:**   |  |  |  | | --- | --- | --- | | Step | Actor Action | System Response | | 1 | Fail to insert or update to database. | Display error message: “An error occurred. Please try again later!” |   **Relationships:**   * Auto parse data * Force parse data   **Business Rules:** Some different laptops have similar name. | | | |

**Table 4: <Staff>** **Split laptop**

2.3.1<Member> Overview Use case



Figure 1: <Member> Overview Use case

**2.3.1.1 <Member>** **Edit profile Use Case Diagram**



Figure 2: **<Member> Edit profile Use case Specification**

|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE – LRA001** | | | |
| **Use Case No.** | LRA001 | **Use Case Version** | 1.0 |
| **Use Case Name** | Edit profile | | |
| **Author** | HonNV | | |
| **Date** | 17/05/2015 | **Priority** | Normal |
| **Actor:**   * Member   **Summary:**   * This use case allows members to edit their profile in the system.   **Goal:**   * Profile will be updated to the system.   **Triggers:**   * Members wants to edit their profile. * From the sidebar:   + “Manage profile” tag -> “Edit profile” subtag.   **Preconditions:**   * User must login the system with Member or Admin role.   **Post Conditions:**   * **Success:** Profile will be updated to the database. * **Fail:** Show error message on the current page.   **Main Success Scenario:**   |  |  |  | | --- | --- | --- | | Step | Actor Action | System Response | | 1 | Click “Manage profile” tag. | Navigate to “Manage profile” tag which contains:   * “Edit profile” tag. * “Change password” tag. | | 3 | Member clicks “Edit profile” tag. | Navigate to “Edit profile” page which contains:   * “Username”: label. * Username: textbox, min length: 5, max length: 100, disabled. * “Email”:label. * Email:texboxt, regular expression: /^[\_a-z0-9-]+(\.[\_a-z0-9-]+)\*@@[a-z0-9-]+(\.[a-z0-9-]+)\*(\.[a-z]{2,4})$/. * “Save change”: button. | | 4 | Member fills data to the form.  Member clicks “Save change” button. | Update profile to the database.  [Exception 1] |   **Alternative Scenario:**   |  |  |  | | --- | --- | --- | | No | Actor Action | System Response | |  | N/A | N/A |   **Exceptions:**   |  |  |  | | --- | --- | --- | | No | Actor Action | System Response | | 1 | Invalid email format. | “Email” textbox border color is red. |   **Relationships:** N/A  **Business Rules:**   * Member’s profile is updated. | | | |

Table1: <Member> Edit profile

**2.3.1.2 <Member>** **Change password Use Case Diagram**



Figure 3: <Authorized user> Change Password

Use Case Specification

|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE – LRA002** | | | |
| **Use Case No.** | LRA002 | **Use Case Version** | 1.0 |
| **Use Case Name** | Change password | | |
| **Author** | HonNV | | |
| **Date** | 17/05/2015 | **Priority** | Normal |
| **Actor:**   * Member.   **Summary:**   * This use case allows Member to change password.   **Goal:**   * Member changes password successfully.   **Triggers:**   * Member logins to website. * Member clicks “Change password” tag.   **Preconditions:** Member has register into the system.  **Post Conditions:**   * **Success:** New password has been updated into database. * **Fail:** Show error message.   **Main Success Scenario:**   |  |  |  | | --- | --- | --- | | Step | Actor Action | System Response | | 1 | Member clicks “Change password” tag. | Show “Change password ” page with includes:   * Old passwrod: textbox, password box, min length: 6, max length: 255, required. * New password: textbox, password box, min length: 6, max length: 255, required. * Confirm new password: textbox, password box, min length: 6, max length: 255, required. * “Save change”: button. | |  | Member clicks “Save change” button [Alternative 1] | Show successful message and return to home page. [Exception 1,2, 3,4,5,6] |   **Alternative Scenario:**   |  |  |  | | --- | --- | --- | | No | Actor Action | System Response | | 1 | Length of password is not in range. | Show error message: “Password length must be between 6-255characters!” | | 2 | “Old password” is wrong. | Show error message:” Wrong password.” | | 3 | “Old password” is a blank. | Show error message:” Please enter a password” | | 4 | “New password” is a blank. | Show error message:” Please enter your new password” | | 5 | “Confirm new password” is a blank. | Show error message:” Please enter your confirm new password” | | 6 | “New password” and “Confirm new password” is not similar. | Show error message: “New password and Confirm new password do not match”. |   **Exceptions:**   |  |  |  | | --- | --- | --- | | No | Actor Action | System Response | | 1 |  |  | |  |  |  |   **Relationships:** Manage profile.  **Business Rules:**   * New password has been updated into database. * Use MD5 encryption. | | | |

Table2: <Member> Change Password

**2.3.1.3 <Member>** **Retrieve password Use Case Diagram**



Figure 4: **<Member> Retrieve password Use case Specification**

|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE – LRA003** | | | |
| **Use Case No.** | LRA003 | **Use Case Version** | 1.0 |
| **Use Case Name** | Retrieve password | | |
| **Author** | HonNV | | |
| **Date** | 17/05/2015 | **Priority** | Normal |
| **Actor:**   * Member   **Summary:**   * This use case allows member to retrieve password in the system.   **Goal:**   * User’s password will be retrieve.   **Triggers:**   * Members wants to retrieve password. * Member clicks “Sent” button.   **Preconditions:**   * N/A   **Post Conditions:**   * **Success:** Password will be retrieved to member. -> New password will be created and sent to member. * **Fail:** Show error message on the current page.   **Main Success Scenario:**   |  |  |  | | --- | --- | --- | | Step | Actor Action | System Response | | 1 | Click “Forgot password” link. | Navigate to “Forgot password” page which contains:   * Email:regular expression: /^[\_a-z0-9-]+(\.[\_a-z0-9-]+)\*@@[a-z0-9-]+(\.[a-z0-9-]+)\*(\.[a-z]{2,4})$/ required. * “Sent”: button. | | 2 | Member fills data to the email.  Member clicks “Sent” button. | Insert new password into database. New Password is sent to email.  [Exception 1, 2] |   **Alternative Scenario:**   |  |  |  | | --- | --- | --- | | No | ctor Action | System Response | |  | N/A | N/A |   **Exceptions:**   |  |  |  | | --- | --- | --- | | No | Actor Action | System Response | | 1 | Invalid email format. | “Email” textbox border color is red. | | 2 | Email does not exist in the system. | “Email” textbox border color is red. |   **Relationships:** N/A  **Business Rules:**   * The new password is saved to the system. | | | |
|  | | | |

Table 3: < Member > Retrieve password

**2.3.1.4 <Member>** **Logout Use Case Diagram**



Figure 36: < Member > Log out

Use Case Specification

|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE – LRA004** | | | |
| **Use Case No.** | LRA004 | **Use Case Version** | 1.0 |
| **Use Case Name** | Logout | | |
| **Author** | HonNV | | |
| **Date** | 17/05/2015 | **Priority** | Normal |
| **Actor:**   * Member.   **Summary:**   * This use case allows Member to log out of the system.   **Goal:**   * Member logs out of the system. Member’s current session is destroyed.   **Triggers:**   * Member wants to log out. * Member clicks “Logout” link.   **Preconditions:**   * Member has logged in to the system.   **Post Conditions:**   * **Success**: Member’s current session is destroyed. Redirect to homepage. * **Fail:** Show error message.   **Main Success Scenario:**   |  |  |  | | --- | --- | --- | | Step | Actor Action | System Response | | 1 | Member clicks “Logout” link. | * Destroy Member’s session. Redirect to homepage. [Exception 1] |   **Alternative Scenario:**   |  |  |  | | --- | --- | --- | | No | Actor Action | System Response | |  | N/A | N/A |   **Exceptions:**   |  |  |  | | --- | --- | --- | | No | Actor Action | System Response | | 1 | Fail to destroy Member’s session | Redirect to error page. |   **Relationships:** N/A  **Business Rules:**   * Member clicks “Logout” link, * Member’s role will be changed to guest. | | | |

Table4: < Member > Log out

##### <System> Auto Parse Data

Use Case Diagram



Figure 9: <System> Auto Parse Data

Use Case Specification

|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE – LRA005** | | | |
| **Use Case No.** | LRA005 | **Use Case Version** | 1.0 |
| **Use Case Name** | Auto Parse Data | | |
| **Author** | HonNV | | |
| **Date** | 17/05/2015 | **Priority** | High |
| **Actor:**   * System.   **Summary:**   * System can parse data automatically from many websites at specified time.   **Goal:**   * Get data from many websites.   **Triggers:**   * The time hits configured time.   **Preconditions:**   * Parse time has been configured.   **Post Conditions:**   * **Success:** New data is inserted to database. * **Fail:** Nothing is changed in the database.   **Main Success Scenario:**   |  |  |  | | --- | --- | --- | | Step | Actor Action | System Response | | 1 | Server checks the current time. If it hits configured time, parse process starts. | * Send request to the parsed link. * Fetch data from the response based on the inputted XPaths. * Validate data [Exception 1]. * If data is valid, insert to database [Alternative 1]. |   **Alternative Scenario:**   |  |  |  | | --- | --- | --- | | Step | Actor Action | System Response | | 1 | Server checks the current time. If it hits configured time, parse process starts. | * If fetched product is already in the database, update its information. |   **Exceptions:**   |  |  |  | | --- | --- | --- | | No | Actor Action | System Response | | 1 | Data is invalid. | * Don’t insert to database. |   **Relationships:** N/A  **Business Rules:**   * If product’s name is empty, consider it invalid. | | | |

Table 5: <System> Auto Parse Data



**Figure 1: <Admin>** Overview Use case

**2.3.1.1 < Admin >** Create Account **Use Case Diagram**

**** Figure 2: **< Admin >** Create Account **Use Case Specification**

|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE – LRA001** | | | |
| **Use Case No.** | LRA001 | **Use Case Version** | 1.0 |
| **Use Case Name** | Create Account | | |
| **Author** | Nguyen Manh Khuong | | |
| **Date** | 17/05/2015 | **Priority** | Normal |
| **Actor:**   * Admin   **Summary:**   * This use case allows admin to create new account in the system.   **Goal:**   * New account will be added to the system.   **Triggers:**   * Admin wants to create new account. * From the sidebar:   + Choose “Manage Account”.   + Choose “Create account” button.   **Preconditions:**   * User must login the system with admin role.   **Post Conditions:**   * **Success:** New account will be added to the database. * **Fail:** Show error message on the current page.   **Main Success Scenario:**   |  |  |  | | --- | --- | --- | | Step | Actor Action | System Response | | 1 | Click menu item “Manage Account”. | Navigate to “List of Accounts” page which contains:   * “Create account” button. * A table with 5 columns:   + Username.   + Email.   + Role.   + Status.   + Edit. * “Username”: label. * “Email”: label. * “Role”: label. * “Enable”: button or “Disable”: button * “Edit”: button. | | 2 | Admin clicks “Create account” button. | Navigate to “Create new account” popup which contains:   * Username: textbox, min length: 5, max length: 100, required. * Password: textbox, min length: 5, max length: 100, required. * Confirm password: textbox, min length: 5, max length: 100, required. * Email address: regular expression: /^[\_a-z0-9-]+(\.[\_a-z0-9-]+)\*@@[a-z0-9-]+(\.[a-z0-9-]+)\*(\.[a-z]{2,4})$/ required. * Role: drop down list, “Staff" default. * “Create account”: button. * Status: drop-down list, “Enable” default. * “Clear”: button. * “Create”: button | | 3 | Admin fills data to the form.  Admin clicks “Create account” button.  [Alternative 1] | Insert new account to the database and close popup  [Exception 1, 2, 3,4,5] |   **Alternative Scenario:**   |  |  |  | | --- | --- | --- | | No | Actor Action | System Response | | 1 | Admin clicks “Clear” button”. | Clear all input data. |   **Exceptions:**   |  |  |  | | --- | --- | --- | | No | Actor Action | System Response | | 1 | “Username” is not in range [5-100]. | “Username” textbox border color is red. | | 2 | “Password” is not in range  [5-100]. | “Password” textbox border color is red. | | 3 | “Username” has already existed. | Show message: “Username has already existed!” and “Username” textbox border color is red. | | 4 | “Re-password” and “Password” do not match. | “Password” textbox border color is red.  “Re-password” textbox border colour is red. | | 5 | Invalid email format. | “Email address” textbox border colour is red. |   **Relationships:** N/A  **Business Rules:**   * Each created account must be in active state when insert to database. | | | |

Table1: < Admin > Create Account

**2.3.1.2 < Admin >** Edit Account **Use Case Diagram**

 Figure 3: **< Admin >** Update Account **Use Case Specification**

|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE – LRA002** | | | |
| **Use Case No.** | LRA002 | **Use Case Version** | 1.0 |
| **Use Case Name** | Update Account | | |
| **Author** | Nguyen Manh Khuong | | |
| **Date** | 17/05/2015 | **Priority** | Normal |
| **Actor:**   * Admin, Staff   **Summary:**   * this use case allows admin to update all account's information.   **Goal:**   * New information will be updated to the database.   **Triggers:**   * Admin want to change some information.   **Preconditions:**   * User must log into the system.   **Post Conditions:**   * **Success:** New information will be updated to the database. * **Fail:** Show error message.   **Main Success Scenario:**   |  |  |  | | --- | --- | --- | | Step | Actor Action | System Response | | 1 | Click menu item “Manage Account”. | Navigate to “List of Accounts” page which contains:   * “Create account” button. * A table with 5 columns:   + Username.   + Email.   + Role.   + Status.   + Edit. * “Username”: label. * “Email”: label. * “Role”: label. * “Enable”: button or “Disable”: button * “Edit”: button. | | 2 | Click “Edit” button. | Navigate to “Edit account” popup, which contains:   * Username: textbox, disabled. * Email address: regular expression: /^[\_a-z0-9-]+(\.[\_a-z0-9-]+)\*@@[a-z0-9-]+(\.[a-z0-9-]+)\*(\.[a-z]{2,4})$/ required. * Role: drop down list, “Staff" default. * Status: drop-down list. Default choice base on account current status. * “Save changes”: button. | | 3 | Admin updates “Email”, “Role”, “Status”.  Click “Save changes” button.  [Alternative 1] | New information will be updated to the database and  Close the popup.  [Exception 1] |   **Alternative Scenario:**   |  |  |  | | --- | --- | --- | | No | Actor Action | System Response | | 1 | Admin clicks “Cancel” button”. | Back “admin” page. |   **Exceptions:**   |  |  |  | | --- | --- | --- | | No | Actor Action | System Response | | 1 | Invalid email format. | Show error message: “Email address must be valid format!” |   **Relationships:** N/A  **Business Rules:**   * Cannot edit “Username”. | | | |

Table1: < Admin > Update Account

****Figure 3: **< Admin >** **Activate/Deactivate Account Use Case Specification**

|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE – LRA003** | | | |
| **Use Case No.** | LRA003 | **Use Case Version** | 1.0 |
| **Use Case Name** | Activate/Deactivate Account | | |
| **Author** | Nguyen Manh Khuong | | |
| **Date** | 17/05/2015 | **Priority** | Normal |
| **Actor:**   * Admin   **Summary:**   * This user case allows admin to activate or deactivate user account.   **Goal:**   * Disabled accounts will be enabled again or opposite.   **Triggers:**   * Admin wants to activate an account so that it can be used to log in the system again or opposite.   **Preconditions:**   * User must log in the system with admin role.   **Post Conditions:**   * **Success:** The selected account will be enabled. * **Fail:** No account is enabled.   **Main Success Scenario:**   |  |  |  | | --- | --- | --- | | Step | Actor Action | System Response | | 1 | Click menu item “Manage Account”. | Navigate to “List of accounts” page which contains:   * “Create account” button. * A table with 5 columns:   + Username.   + Email.   + Role.   + Status.   + Edit. * “Username”: label. * “Email”: label. * “Role”: label. * “Deactivate”: button * “Edit”: button. | | 2 | Click “Disable” button.  [Alternative 1] | The account will be activating and button name change to “Enable” or opposite.  [Exception 1] |   **Alternative Scenario:** N/A   |  |  |  | | --- | --- | --- | | No | Actor Action | System Response | | 1 | Click “Disable” button. | The account will be deactivating and button name change to “Disable”.  [Exception 2] |   **Exceptions:**   |  |  |  | | --- | --- | --- | | No | Actor Action | System Response | | 1 | Cannot enable the account due to database connection. | Show error message: “Cannot enable the account! Please try again later.” | | 2 | Cannot disable the account due to database connection. | Show error message: “Cannot disable the account! Please try again later.” |   **Relationships:** Manage Account  **Business Rules:**   * Admins cannot enable/disable account of themselves or other admins. | | | |

Table2: <Admin> Activate/Deactivate Account



Figure 4: **< Staff >** **Configure System Use Case Specification**

|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE – LRA005** | | | |
| **Use Case No.** | **LRA005** | **Use Case Version** | 1.0 |
| **Use Case Name** | Configure System | | |
| **Author** | Nguyen Manh Khuong | | |
| **Date** | 17/05/2015 | **Priority** | High |
| **Actor:**   * Staff.   **Summary:**   * This use case allows staff to configure the system.   **Goal:**   * System parameters will be configured by admin.   **Triggers:**   * Staff wants to change system parameter, such as “Parser running time automatically everyday”. * From the sidebar:   + Choose “Configure System” tag.   **Preconditions:**   * User must log in the system with Staff role.   **Post Conditions:**   * **Success:** New parameter will be saved to configuration file. * **Fail:** Nothing is saved to configuration file.   **Main Success Scenario:**   |  |  |  | | --- | --- | --- | | Step | Actor Action | System Response | | 1 | Click at “Configure System” tag. | Show “Configure System” page which contains:   * “Parser running time automatically everyday”: textbox, min time: 00:00, max time: 23.58, type: time, required. * “Update”: button. | | 2 | Staff changes system parameters. |  | | 3 | Click on “Save” button.  [Exception 1] | * Save data to configuration file. * Show success message. |   **Alternative Scenario:** N/A  **Exceptions:**   |  |  |  | | --- | --- | --- | | No | Actor Action | System Response | | 1 | Connection to database error. | Show error message: “Cannot update new system configuration! Please try again later.” |   **Relationships:** N/A  **Business Rules:**   * “Parser running time automatically everyday” mean time the system auto parser product by link staff approved. | | | |