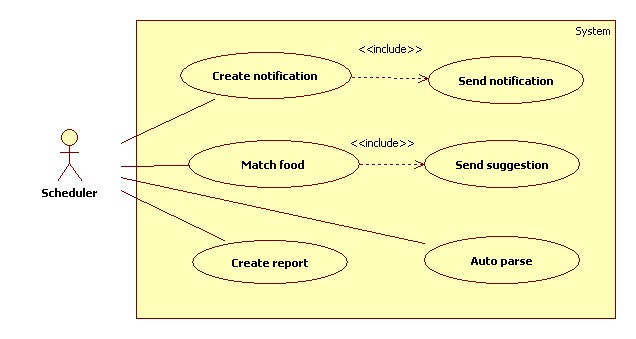
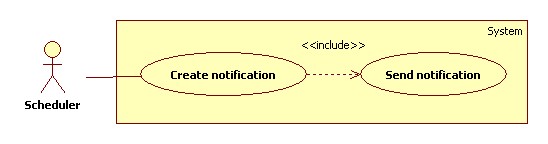
**<Scheduler> Overview Use case**

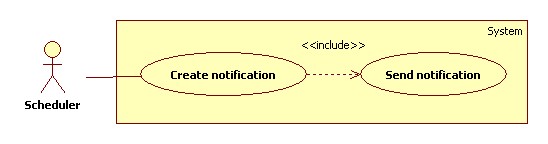


**Figure: <Scheduler> Overview Use case**

****

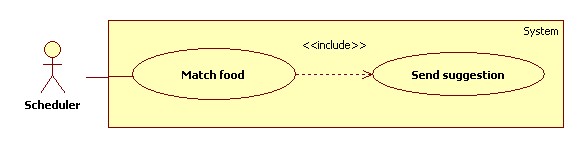
**Figure: <Scheduler> Create notification use case diagram**

|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE - <UC Number>** | | | |
| **Use Case No.** |  | **Use Case Version** | 1.0 |
| **Use Case Name** | Create notification | | |
| **Author** | Phan Hoàng Giáp | | |
| **Date** | 19/01/2016 | **Priority** | High |
| **Actor:**   * Scheduler   **Summary:**   * This use case allows scheduler to create notifications for all users about all expiring foods in their refrigerator.   **Goal:**   * Scheduler can create notifications.   **Triggers:**   * System time passes 0 o’clock.   **Preconditions:** N/A  **Post Conditions:**   * Success: Log file is generated. * Fail: Log file is generated.   **Main Success Scenario:**   |  |  |  | | --- | --- | --- | | **Step** | **Actor Action** | **System Response** | | **1** | Server checks the current time. If it passes 0 o’clock and scheduler status is “Ready”, create notification process starts. | System gets all foods information of all users from storage:   * Food name * Food expiring day * Number of notification day   If (expiring day minus current time is less than or equal to number of notification day)   * Create notifications * Insert notifications to storage * Generate log file.   Scheduler is changed to “Notification created”. |   **Alternative Scenario:** N/A  **Exception:** N/A  **Relationships:** Send notification  **Business Rules:**   * Every day at 0 o’ clock, scheduler status is changed to “Ready”. * E very day at 0 o’ clock, system will check all food expiring day of all users. * Conditions for creating notification are:   + Expiry date - current date ≤ Number of Notification date.   + Number of Notification date is set by user for each food. If user did not set, the default value will be 3 days. * When new notification is created, its status is “Pending”. * After creating notification process, scheduler status is changed to “Notification created”.   **Log file structure:** NOTIFICATION PROCESS LOG FILE  File name: notification.log  Tạo file lúc: {Created date}, {Create time}   |  |  |  | | --- | --- | --- | | STT | User account | Số notification | |  |  |  | |  |  |  |   Tổng thời gian process: {Total elapsed time}  Tổng notification được tạo: {Total notification} | | | |

****

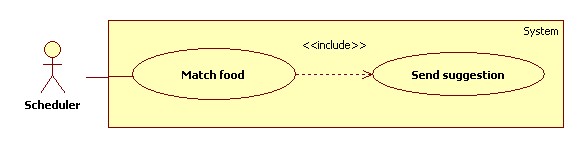
**Figure: <Scheduler> Send notification use case diagram**

|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE - <UC Number>** | | | |
| **Use Case No.** |  | **Use Case Version** | 1.0 |
| **Use Case Name** | Send notification | | |
| **Author** | Phan Hoàng Giáp | | |
| **Date** | 19/01/2016 | **Priority** | High |
| **Actor:**   * Scheduler   **Summary:**   * This use case allows scheduler to send notifications for all users about all expiring foods in their refrigerator at specified time.   **Goal:**   * Scheduler can send notifications to user.   **Triggers:**   * System time passes 0 o’clock.   **Preconditions:**   * Notification time has been configured.   **Post Conditions:**   * Success: Notifications are sent to user. Log file is generated. * Fail: Nothing is sent to user. Log file is generated.   **Main Success Scenario:**   |  |  |  | | --- | --- | --- | | **Step** | **Actor Action** | **System Response** | | **1** | Server checks the current time. If it hits notification time, sending notification process starts. | If notifications status is “Pending”, system sends notifications to account of users. [Exception 1] |   **Alternative Scenario:** N/A  **Exception:**   |  |  |  | | --- | --- | --- | | **No** | **Cause** | **System Response** | | **1** | Send notification fail. | System will try to send notification again after 30 minutes. |   **Relationships:** Create notification  **Business Rules:**   * At notification time, system will send notifications to account of all users. Users can set their own notification time. * When notification is created, the default status will be “Pending” until it is sent. After sending, the status will be changed to “Finish”. * The scheduler will retry sending 3 times.   **Log file structure:** NOTIFICATION PROCESS LOG FILE  File name: notification.log  Tạo file lúc: {Created date}, {Create time}   |  |  |  | | --- | --- | --- | | STT | User account | Số notification | |  |  |  | |  |  |  |   Tổng thời gian process: {Total elapsed time} | | | |

****

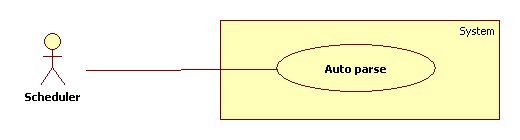
**Figure: <Scheduler> Match food use case diagram**

|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE - <UC Number>** | | | |
| **Use Case No.** |  | **Use Case Version** | 1.0 |
| **Use Case Name** | Match food | | |
| **Author** | Phan Hoàng Giáp | | |
| **Date** | 19/01/2016 | **Priority** | High |
| **Actor:**   * Scheduler   **Summary:**   * This use case allows scheduler to suggest dishes based on food in user’s refrigerator.   **Goal:**   * Scheduler can match food in refrigerator.   **Triggers:**   * System time passes 0 o’clock.   **Preconditions:** N/A  **Post Conditions:**   * Success: Log file is generated. * Fail: Log file is generated.   **Main Success Scenario:**   |  |  |  | | --- | --- | --- | | **Step** | **Actor Action** | **System Response** | | **1** | Server checks the current time. If it passes 0 o’clock and scheduler status is “Notification created”, matching process starts. | System gets all foods name of all users from storage.  System matching foods with recipes in database.  If foods match with recipes:   * Create suggestion with matched recipes. * Insert suggestion to storage. * Generate log file.   Scheduler status is changed to “Matching finished”. |   **Alternative Scenario:** N/A  **Exception:** N/A  **Relationships:** Send suggestion  **Business Rules:**   * Every day at 0 o’ clock, system will check all foods in all users account. * In the matching algorithm, the results must be order by points. Points are given by these rules:   + 1 point for each food which not expiring.   + 2 points for each expiring food. * After matching food process, scheduler status is changed to “Matching finished”.   **Log file structure**: SUGGESTION PROCESS LOG FILE  File name: suggestion.log  Tạo file lúc: {Created date}, {Create time}   |  |  |  |  | | --- | --- | --- | --- | | STT | User account | Thời gian matching | Tổng số dish suggestion | |  |  |  |  | |  |  |  |  |   Tổng thời gian: {Total elapsed time} | | | |

****

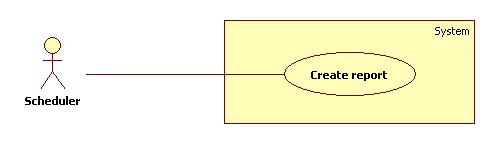
**Figure: <Scheduler> Send suggestion use case diagram**

|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE - <UC Number>** | | | |
| **Use Case No.** |  | **Use Case Version** | 1.0 |
| **Use Case Name** | Send suggestion | | |
| **Author** | Phan Hoàng Giáp | | |
| **Date** | 19/01/2016 | **Priority** | High |
| **Actor:**   * Scheduler   **Summary:**   * This use case allows scheduler to suggest dishes based on food in user’s refrigerator at specified time.   **Goal:**   * Scheduler can match food in refrigerator and suggest dishes to all users.   **Triggers:**   * System time passes 0 o’clock.   **Preconditions:**   * Dish suggestion time has been configured.   **Post Conditions:**   * Success: Dishes information is sent to user. Log file is generated. * Fail: Nothing is sent to user. Log file is generated.   **Main Success Scenario:**   |  |  |  | | --- | --- | --- | | **Step** | **Actor Action** | **System Response** | | **1** | Server checks the current time. If it hits configured time, sending suggestion process starts. | System sends suggestion include dishes information to account of all users. [Exception 1] |   **Alternative Scenario:** N/A  **Exception**   |  |  |  | | --- | --- | --- | | **No** | **Cause** | **System Response** | | **1** | Send suggestion fail. | System will try to send suggestion again after 30 minutes. |   **Relationships:** Match ingredient  **Business Rules:**   * At sending suggestion time, system will send notifications to account of all users. Users can set their own sending time. * The scheduler will retry sending 3 times.   **Log file structure**: SUGGESTION PROCESS LOG FILE  File name: suggestion.log  Tạo file lúc: {Created date}, {Create time}   |  |  |  |  | | --- | --- | --- | --- | | STT | User account | Thời gian matching | Tổng số dish suggestion | |  |  |  |  | |  |  |  |  |   Tổng thời gian: {Total elapsed time} | | | |

****

**Figure: <Scheduler> Get recipes by HTML parsing use case diagram**

|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE - <UC Number>** | | | |
| **Use Case No.** |  | **Use Case Version** | 1.0 |
| **Use Case Name** | Auto parse | | |
| **Author** | Phan Hoàng Giáp | | |
| **Date** | 19/01/2016 | **Priority** | High |
| **Actor:**   * Scheduler.   **Summary:**   * System can parse resource automatically from many websites at specified time.   **Goal:**   * Get resources from culinary websites.   **Triggers:**   * System time passes 0 o’clock on Monday.   **Preconditions:**   * Configuration files has been created by staff.   **Post Conditions:**   * Success: New data is inserted to storage. Log file is generated. * Fail: Nothing is changed in the storage. Log file is generated.   **Main Success Scenario:**   |  |  |  | | --- | --- | --- | | **Step** | **Actor Action** | **System Response** | | **1** | Server checks the current time. If it passes 0 o’clock of Monday, parse process starts. | Send request to the chosen websites.  Fetch data from the response based on the inputted XPaths in configuration files.  Validate data [Exception 1].  If data is valid, insert to storage [Alternative 2].  Generate log file. |   **Alternative Scenario:**   |  |  |  | | --- | --- | --- | | **No** | **Actor Action** | **System Response** | | **1** | Server checks the current time. If it passes 0 o’clock of Monday, parse process starts. | If fetched link resource is already in the storage, update its information.  Generate log file. |   **Exception**   |  |  |  | | --- | --- | --- | | **No** | **Cause** | **System Response** | | **1** | Data is invalid | Generate log file. |   **Relationships:** N/A  **Business Rules:**   * Every day at 0 o’ clock, system will send request to parsed websites. * System fetch data and insert to storage. * Recipes data must be validated before saved to database. * If link resource exists in storage, update its information.   **Log file structure:** AUTO PARSE PROCESS LOG FILE  Filename: parserecipe.log  Tạo file lúc: {Created date}, {Create time}   |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | | STT | Website parse | Thời gian parse | Số thực đơn nhận được | Insert thành công | Insert thất bại | |  |  |  |  |  |  | |  |  |  |  |  |  |   Tổng thời gian parse:{Total elapsed time}  Tổng sản phẩm parse được:{Total parsed recipes} | | | |

****

**Figure: <Scheduler> Create report use case diagram**

|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE - <UC Number>** | | | |
| **Use Case No.** |  | **Use Case Version** | 1.0 |
| **Use Case Name** | Create report | | |
| **Author** | Phan Hoàng Giáp | | |
| **Date** | 19/01/2016 | **Priority** | Normal |
| **Actor:**   * Scheduler   **Summary:**   * System can get user behavior as their searching foods.   **Goal:**   * System can get user searching keywords.   **Triggers:**   * System time passes 0 o’clock.   **Preconditions:**   * N/A   **Post Conditions:**   * Success: New data is saved into storage. Log file is generated. * Fail: Nothing is changed in the storage. Log file is generated.   **Main Success Scenario:**   |  |  |  | | --- | --- | --- | | **Step** | **Actor Action** | **System Response** | | **1** | Server checks the current time. If it is after 0 o’clock and scheduler status is “Matching finished”, create report process starts. | Send request to account of all users.  Fetch search keywords.  Validate data. [Exception 1]  If data is valid, insert to storage.  Generate log file  Scheduler status is changed to “Finished” |   **Alternative Scenario:** N/A  **Exception**   |  |  |  | | --- | --- | --- | | **No** | **Actor Action** | **System Response** | | **1** | Data is invalid | Generate log file |   **Relationships:** N/A  **Business Rules:**   * Every day at 0 o’clock system request search keywords from all users account. * Search keywords will be stored in database of the system as a user attribute. * Log file structure:   CREATE REPORT LOG FILE  Tạo file lúc: {Created date}, {Create time}   |  |  |  | | --- | --- | --- | | STT | User account | Search keyword | |  |  |  | |  |  |  |   Tổng thời gian: {Total elapsed time} | | | |