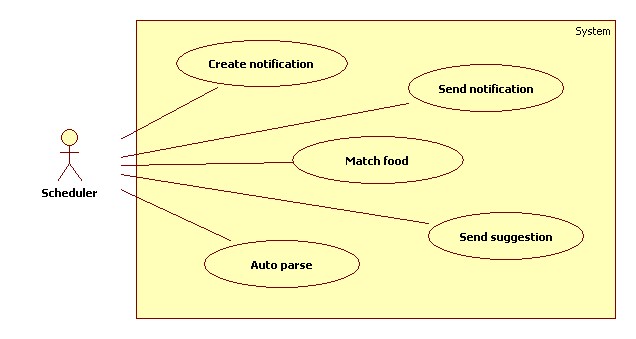
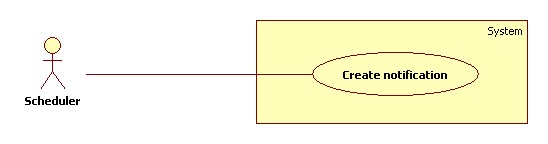
**<Scheduler> Overview Use case**

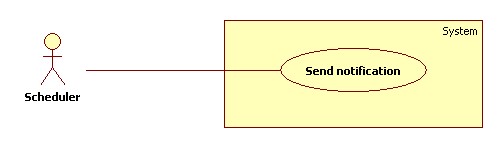


**Figure: <Scheduler> Overview Use case**

****

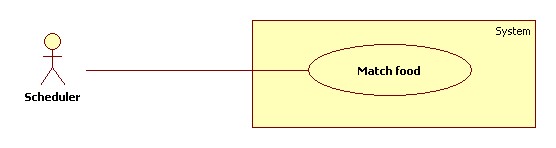
**Figure: <Scheduler> Create notification use case diagram**

|  |  |  |  |
| --- | --- | --- | --- |
| **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 at specified time.   **Goal:**   * Scheduler can create notifications.   **Triggers:**   * The time hit configured time.   **Preconditions:**   * The creating notification time has been configured.   **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 hits configured time and scheduler status is “Ready”, creating notification process starts. | System gets all foods information of all users from storage:   * Food name * Food expiring day * Number of notification day   [Alternative 1] | | **2** | Scheduler status is changed from “Ready” to “Notification created”. |  |   **Alternative Scenario:**   |  |  |  | | --- | --- | --- | | **No** | **Actor Action** | **System Response** | | **1** | If there is remaining time. | * Create notifications * Insert notifications to storage * Generate log file. |   **Exception:** N/A  **Relationships:** N/A  **Business Rules:**   * Every day at 0 o’ clock, scheduler status is changed from “Matching finished” to “Ready”. * After creating notification process, scheduler status is changed to from “Ready” to “Notification created”. * Creating notification time can be configured by staff or it is set as default at 0 o’clock. * System will check all food expiring day of all users. * Conditions for creating notification are:   + Remain time: Expire day - Current day ≤ Number of notification day.   + Number of Notification day is set by user for each food. If user does not set up yet, it is set as system default. * When a new notification is created, notification’s status is “Pending”.   **Log file structure:** CREATE NOTIFICATION PROCESS LOG FILE  File name: createnotification.log  Created date: , Create time:   |  |  |  | | --- | --- | --- | | STT | User account | Number of notification created | |  |  |  | |  |  |  |   Total elapsed time:  Total notifications created: | | | |

****

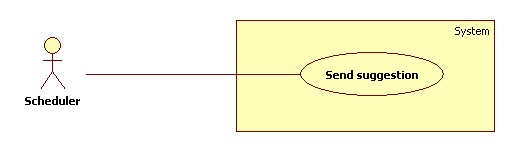
**Figure: <Scheduler> Send notification use case diagram**

|  |  |  |  |
| --- | --- | --- | --- |
| **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:**   * The time hits configured time.   **Preconditions:**   * Notifications was created. * Notification time has been configured.   **Post Conditions:**   * Success: Notifications are sent to user. 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 hits notification time, sending notification process starts. | System sends notification messages to account of users.  [Alternative 1] |   **Alternative Scenario:**   |  |  |  | | --- | --- | --- | | **No** | **Actor Action** | **System Response** | | **1** | System sends notification fail. | System will try to resend notification. |   **Exception:** N/A  **Relationships:** N/A  **Business Rules:**   * Scheduler had created notifications. At notification time, system sends notifications to account of all users. Users can set their own notification time. * Sent messages are notifications in “Pending” status. After sending, notification’s status is changed from “Pending” to “Finish”. * The scheduler will retry sending maximum three times.   **Log file structure:** SEND NOTIFICATION PROCESS LOG FILE  File name: sendnotification.log  Created date: , Create time:   |  |  |  | | --- | --- | --- | | STT | User account | Number of notification sent | |  |  |  | |  |  |  |   Total elapsed time:  Total sent notifications: | | | |

****

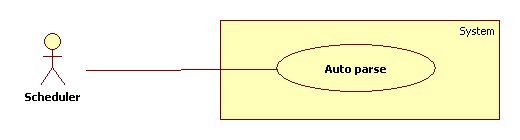
**Figure: <Scheduler> Match food use case diagram**

|  |  |  |  |
| --- | --- | --- | --- |
| **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 match food in user’s refrigerator to suggest dishes at specified time.   **Goal:**   * Scheduler can match food in refrigerator.   **Triggers:**   * The time hit configured time.   **Preconditions:**   * Matching food time has been configured.   **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 hit configured time 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.  [Alternative 1] | | **2** | Scheduler status is changed from “Notification created” to “Matching finished”. |  |   **Alternative Scenario:**   |  |  |  | | --- | --- | --- | | **No** | **Actor Action** | **System Response** | | **1** | If foods match with recipes: | * Create suggestion with matched recipes. * Insert suggestion to storage. * Generate log file. |   **Exception:** N/A  **Relationships:** N/A  **Business Rules:**   * Matching food time can be configured by staff or it is set as default at 0 o’clock. * After matching food process, scheduler status is changed from “Notification created” to “Matching finished”. * 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. * When a new dish suggestion is created, its status is “Pending”.   **Log file structure**: MATCHING FOOD PROCESS LOG FILE  File name: matchfood.log  Created date: , Create time:   |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | | STT | User account | Elapsed time | Total dishes received | Matching successful | Matching failed | |  |  |  |  |  |  | |  |  |  |  |  |  |   Total elapsed time:  Total dish suggestion: | | | |

****

**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 send suggest dishes to all users at specified time.   **Goal:**   * Scheduler can send suggest dishes to all users.   **Triggers:**   * The time hit configured time.   **Preconditions:**   * Dishes were suggested. * Sending suggestion time has been configured.   **Post Conditions:**   * Success: Dishes information is sent to user. 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 hits configured time, sending suggestion process starts. | System sends suggestion messages to account of all users.  [Alternative 1] |   **Alternative Scenario:**   |  |  |  | | --- | --- | --- | | **No** | **Actor Action** | **System Response** | | **1** | System sends notification fail. | System will try to resend notification. |   **Exception:** N/A  **Relationships:** N/A  **Business Rules:**   * Scheduler had matched food. At sending suggestion time, system sends suggestions to account of all users. Users can set their own sending time. * Sent messages are suggestions in “Pending” status. After sending, dishes suggestion’s status is changed from “Pending” to “Finish”. * The scheduler will retry sending maximum three times.   **Log file structure**: SEND SUGGESTION PROCESS LOG FILE  File name: sendsuggestion.log  Created date}, Create time:   |  |  |  |  | | --- | --- | --- | --- | | STT | User account | Elapsed time | Number of sent suggestion | |  |  |  |  | |  |  |  |  |   Total elapsed time: | | | |

****

**Figure: <Scheduler> Auto parse 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:**   * This use case allows scheduler to parse resource automatically from culinary websites at specified time.   **Goal:**   * Scheduler can get resources from culinary websites.   **Triggers:**   * The time hit configured time.   **Preconditions:**   * Configuration files have been created. * Parse time has been configured.   **Post Conditions:**   * Success: New data is inserted to storage. 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 hits configured time, parse process starts. | Send request to the chosen websites.  Fetch data from the response based on configuration files.  Data is saved to temporary storage.  Validate data [Exception 1].  If fetched link resource is not exist yet, insert to storage [Alternative 1].  Generate log file. |   **Alternative Scenario:**   |  |  |  | | --- | --- | --- | | **No** | **Actor Action** | **System Response** | | **1** | If fetched link resource is already in the storage. | Update its information. |   **Exception**   |  |  |  | | --- | --- | --- | | **No** | **Cause** | **System Response** | | **1** | Data is invalid | Generate log file. |   **Relationships:** N/A  **Business Rules:**   * Parse time can be configured by staff or it is set as default at 0 o’clock on Monday. * At configured time, system will send request to parsed websites. * System fetch data and insert to storage. * Recipes data must be validated before saved to database. * Valid data: parsed resource is correct with the configuration files * If fetched resource exists in storage, update its information.   **Log file structure:** AUTO PARSE PROCESS LOG FILE  Filename: parserecipe.log  Created date: , Create time:   |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | | STT | Website parse | Elapsed time | Total recipes received | Insert successful | Insert failed | |  |  |  |  |  |  | |  |  |  |  |  |  |   Total elapsed time:  Total parsed recipes: | | | |