# Software Requirement Specifications

## User Requirement Specification

### Guest Requirement

Guest is a person who loves to play football. Guest can do the following functions:

* Sign up
* Sign in

### User Requirement

User is a person who represents a football team to reserve field. User can do the following functions:

* Manage account:
  + View profile
  + Edit profile
* Search field
* Reserve field
* Manage reservation requests:
  + Create reservation requests
  + View reservation requests
  + Cancel reservation requests
* Accept match
* View list of opponents
* Send feedback
* Send report
* Checkout transaction.

### Field Owner Requirement

Field owner is a person who has football fields and let users reserve his/her fields by using our application. The following functions are available for field owner to use:

* Manage account:
  + View profile
  + Edit profile
* Manage field:
  + Update timeline
  + Create field
  + Update price
  + Disable field
  + Remove field
  + Create promotions
* Send feedback
* Send report

### Admin Requirement

Admin is an employee from our company. Admin is responsible for managing service provider system and accounts. Admin can do following functions:

* Update users’ accounts
* Review reports

### Staff Requirement

Staff is an employee from our company. Staff’s primary job is to manage profit. Staff can do following functions:

* Manage profit:
  + View sales statistics.
  + Set percent commission.

### Payment System Requirement

Payment system is an online payment portal. Payment system’s function is to execute online transactions.

## Software Requirement Specification (WIP)

### External Interface Requirement (Template)

#### User Interface

* The user interface uses Vietnamese as main language for users and English for Staff, Manager and Admin on both Mobile and Web application.

#### Hardware Interface

* Android Smartphone: Nexus 4 or later.

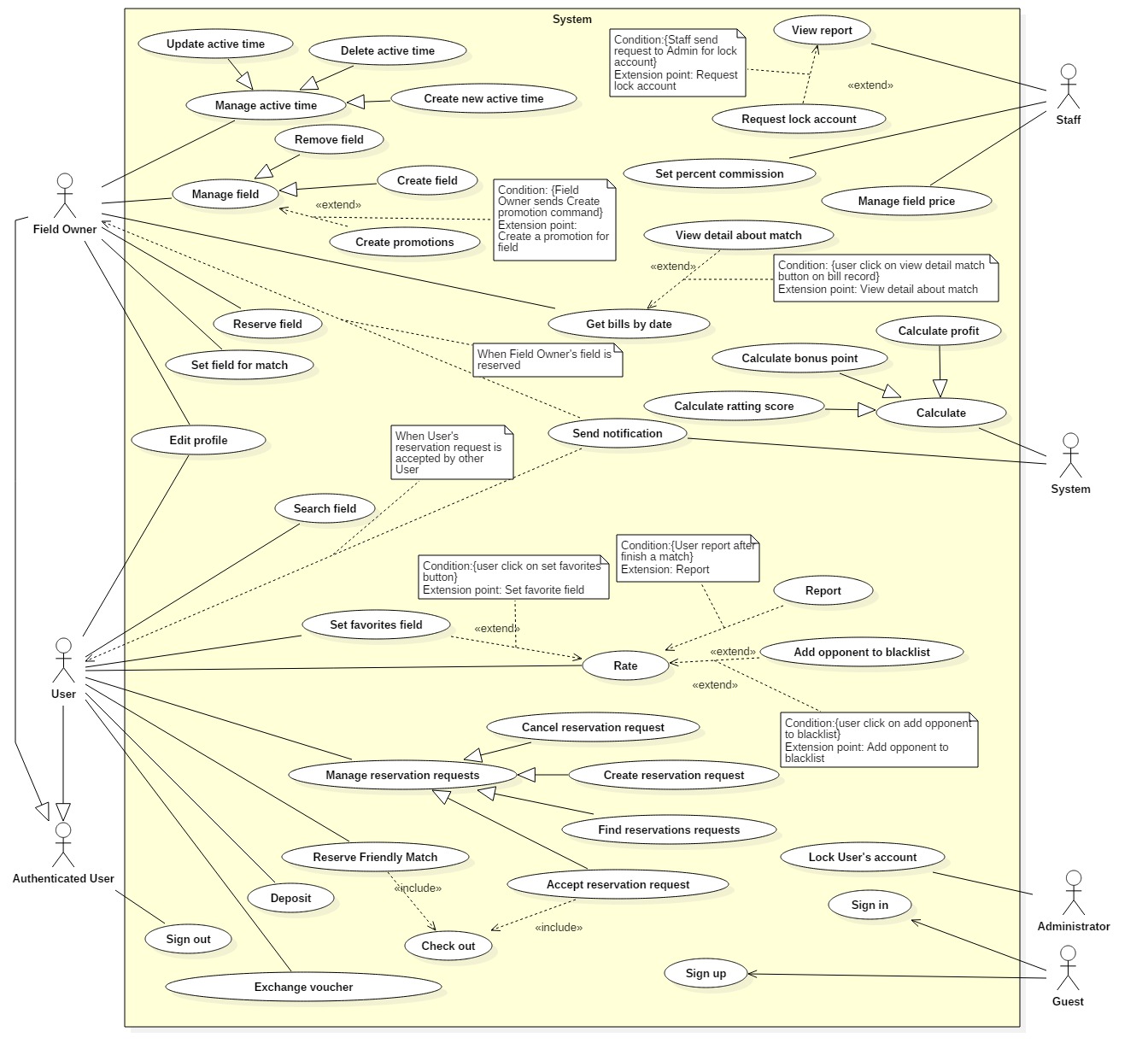
#### Software Interface

* Web application: work with Firefox (v30 or above), Chromes (v25 or above)
* Mobile Application: Android 4 (or later)

#### Communication Protocol

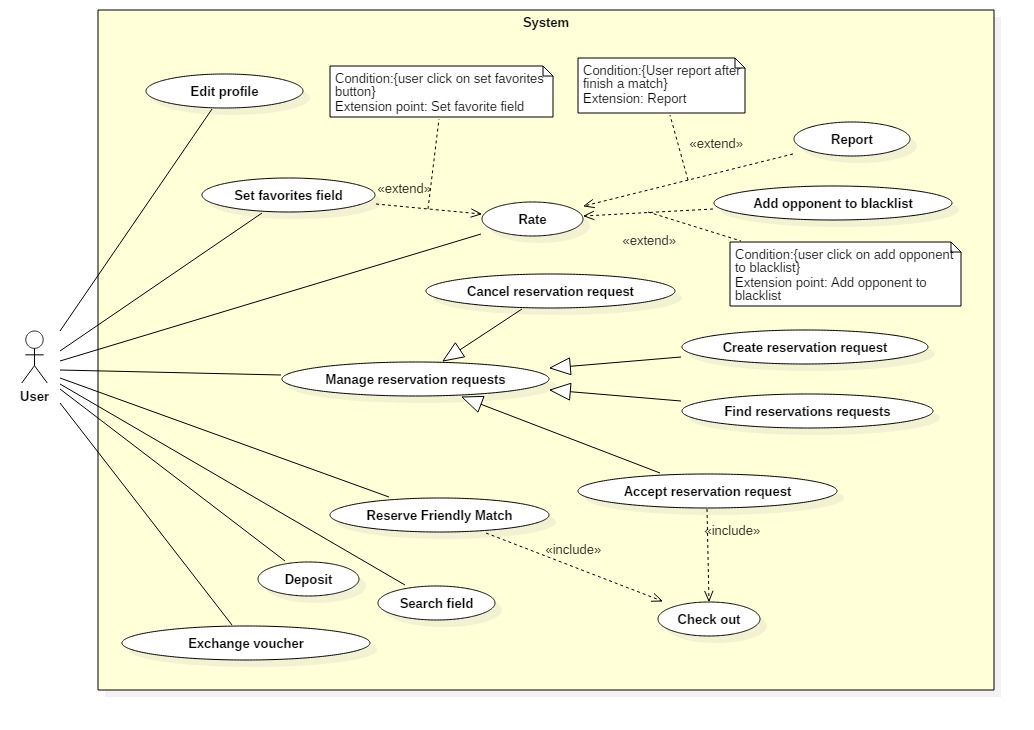
* Use HTTP protocol 1.1 for communication between the web browser and the web server.
* Use HTTP protocol 1.1 for communication between the mobile application and the web service.
* Use HTTP protocol 1.1 for communication between the server and the Microsoft service.

### System Overview Use Case

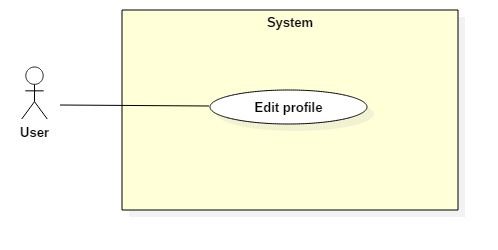


### List of Use Case

#### User Use case

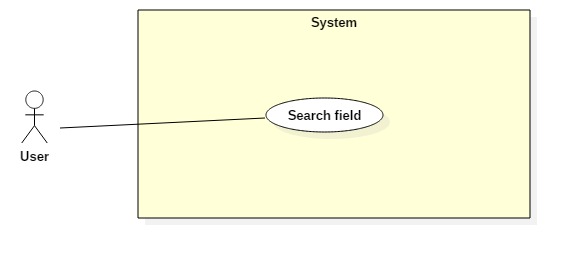


##### <User> Edit profile



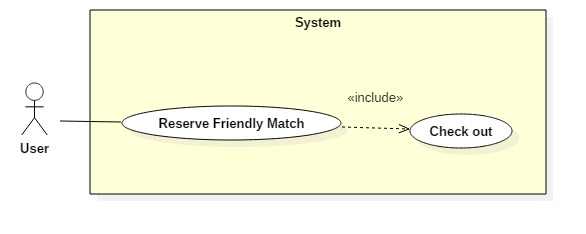
|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE – UC006** | | | |
| **Use Case No.** | **006** | **Use Case Version** | **1.0** |
| **Use Case Name** | Edit profile | | |
| **Author** |  | | |
| **Date** | 24/10/2017 | **Priority** | **Normal** |
| **Actor:**   * User, Field Owner   **Summary:**   * This use case allows Actor to edit account profile.   **Goal:**   * Account is edited successfully and store in database of the system.   **Triggers:**   * Actor sends command to edit profile.   **Preconditions:**   * Actor has accessed the system under user role or field owner role.   **Post Conditions:**   * **Success:** Account will be edit. * **Fail:** System shows messages.   **Main Success Scenario:**   |  |  |  | | --- | --- | --- | | Step | Actor Action | System Response | | 1 | Actor sends command to request edit profile. | System requires information from Actor:   * Email: free text input, required, regex [^[\_A-Za-z0-9-\\+]+(\\.[\_A-Za-z0-9-]+)\*@"+"[A-Za-z0-9-]+(\\.[A-Za-z0-9]+)\*(\\.[A-Za-z]{2,})$] * Full name: free text input, required, length (10-50) * New Password: free text input, required, length (6-12) * Repeat new password: free text input, required, length (6-12) * Address: free text input, required, length (6-500) * Phone: free number input, length(10-12), positive integer, value: [0,9] | | 2 | Actor inputs information |  | | 3 | Actor sends command to save edit.  [Alternative 1]  [Alternative 2] | System shows edit successfully.  [Exception 1]  [Exception 2]  [Exception 3] |   **Alternative Scenario 1:**   |  |  |  | | --- | --- | --- | | Step | Actor Action | System Response | | 1 | Actor sends command to reset. | System reset all field to blank. |   **Alternative Scenario 2:**   |  |  |  | | --- | --- | --- | | Step | Actor Action | System Response | | 1 | Actor sends command to pick location via google maps. | System save location Actor picked to Address field. |   **Exceptions 1:**   |  |  |  | | --- | --- | --- | | Step | Cause | System Response | | 1 | Actor input already exist email. | System shows warning message “Email already exist”. |   **Exceptions 2:**   |  |  |  | | --- | --- | --- | | Step | Cause | System Response | | 1 | Actor does not input required field. | System notices that Actor need to input all these field:   * “Email”: System display warning message: “please fill out this field”. * “Full name”: System display warning message: “please fill out this field”. * “Password”: System display warning message: “please fill out this field”. * “Repeat password”: System display warning message: “please fill out this field”. * “Address”: System display warning message: “please fill out this field”. * “Phone”: System display warning message: “please fill out this field”. |   **Exceptions 3:**   |  |  |  | | --- | --- | --- | | Step | Cause | System Response | | 1 | Actor input wrong some fields with requirement. | System notices that Actor need to re-input all these field:   * “Email”: System display warning message: “Email invalid! ([me@example.com)](mailto:me@example.com))”. * “Email”: System display warning message: “Email must be 10 - 254 characters”. * “Full name”: System display warning message: “Full name must be 10 - 50 characters”. * “Password”: System display warning message: “Password must be 6 - 12 characters”. * “Repeat password”: System display warning message: “Repeat password does not match password”. * “Address”: System display warning message: “Full name must be 6 - 500 characters”. * “Phone”: System display warning message: “Phone must be numbers”. |   **Relationships: N/A**  **Business Rules:**   * After edited, new information of account will be stored in database of the system. * Approximate address depend on GPRS and Google maps. * Only Field Owner has longitude and latitude. | | | |

##### <User> Search Field



|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE – UC003** | | | |
| **Use Case No.** | 003 | **Use Case Version** | 1.0 |
| **Use Case Name** | Search Field | | |
| **Author** |  | | |
| **Date** | 24/10/2017 | **Priority** | Normal |
| **Actor:**   * User   **Summary:**   * This use case allows Actor to search Field by input keyword.   **Goal:**   * Actor can find available Field base on keyword.   **Triggers:**   * Actor sends search Field command.   **Preconditions:**   * Actor already signed in.   **Post Conditions:**   * **Success:** list field are shown. * **Fail:** N/A   **Main Success Scenario:**   |  |  |  | | --- | --- | --- | | Step | Actor Action | System Response | | 1 | Actor input field address into search textbox . |  | | 2 | Actor send Search Field command.  [Alternative 1] | System will find in database any record of field table keyword like input text and show that record as list of fields. |   **Alternative Scenario 1:**   |  |  |  | | --- | --- | --- | | Step | Actor Action | System Response | | 1 | Actor inputs blank in textbox. | System show 10 fields near Actor based on Actor’s location. |   **Exceptions :** N/A  **Relationships:** N/A  **Business Rules:**   * Use likely search for search command | | | |

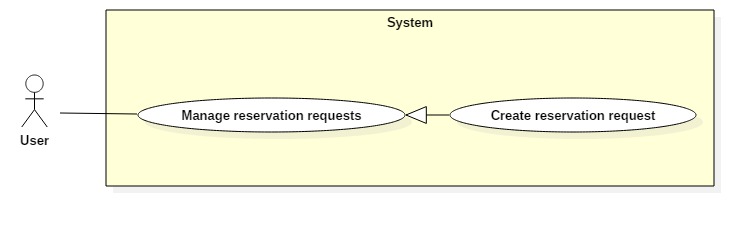
##### <User> Reserve Friendly match



|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE – UC003** | | | |
| **Use Case No.** | 003 | **Use Case Version** | 1.0 |
| **Use Case Name** | Reserve Friendly match | | |
| **Author** |  | | |
| **Date** | 24/10/2017 | **Priority** | Normal |
| **Actor:**   * User   **Summary:**   * This use case allows Actor to reserve field by chose field and time.   **Goal:**   * Actor can reserve a field.   **Triggers:**   * Actor sends reserve command.   **Preconditions:**   * Actor has accessed the system under user role.   **Post Conditions:**   * **Success:** User reserve Field successfully. * **Fail:** N/A   **Main Success Scenario:**   |  |  |  | | --- | --- | --- | | Step | Actor Action | System Response | | 1 | Actor chose expected field from list of field. | System find active time of field and response to User. | | 2 | Actor inputs start time and end time.  Actor send Reserve command. | * System response information about match (field owner, field type, start time, end time, price) and reserve time slot for Actor. * System auto switch back to active time of field view and cancel reservation time slot after 1 minute.   [Exception 1] | | 3 | Actor send checkout command.  [Alternative 1] | System show successful view  System will send notification to Field Owner and create match in database. |   **Alternative Scenario 1:**   |  |  |  | | --- | --- | --- | | Step | Actor Action | System Response | | 1 | Actor cancel reservation request. | System back to list of fields window.  Request did not save to database. |   **Exceptions 1:**   |  |  |  | | --- | --- | --- | | Step | Cause | System Response | | 1 | Actor’s balance is not enough to pay for the match. | System shows warning message “Not enough money. Please deposit now”.  System shows link to Deposit. |   **Exceptions :** N/A  **Relationships:** N/A  **Business Rules:**   * After get reserve command, the system will get the field information, time then send notification to field owner and create match in database. | | | |

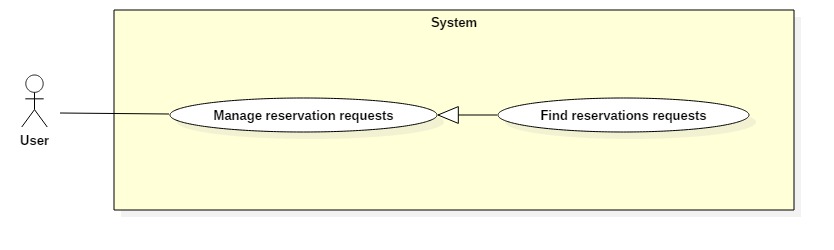
Table : Reserve Friendly match specification

##### <User> Create reservation request



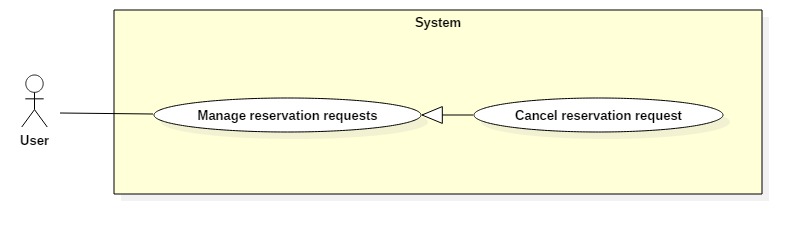
|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE – UC003** | | | |
| **Use Case No.** | 003 | **Use Case Version** | 1.0 |
| **Use Case Name** | Create reservation request | | |
| **Author** |  | | |
| **Date** | 28/10/2017 | **Priority** | Normal |
| **Actor:**   * User   **Summary:**   * This use case allows Actor to create reservation request by input time.   **Goal:**   * Actor can create a reservation request based on time.   **Triggers:**   * Actor sends create reservation request command.   **Preconditions:**   * Actor has accessed the system under user role.   **Post Conditions:**   * **Success:** Actor create reservation request successfully and System save request to database. * **Fail:** N/A   **Main Success Scenario:**   |  |  |  | | --- | --- | --- | | Step | Actor Action | System Response | | 1 | Actor sends command to create reservation request. | System requires information from Actor:   * Start time. * End time. * Number of hours want to play. * Field type. * Current position of Actor. * Search radius. | | 2 | Actor send create reservation request command  [Alternative 1] | * System take Actor’s current position and time Actor inputted. * System will create reservation request in database.   [Exception 1] |   **Alternative Scenario 1:**   |  |  |  | | --- | --- | --- | | Step | Actor Action | System Response | | 1 | Actor type other address in address field. | System support Actor choose position by Google API.  System update new position inputted by Actor. |   **Exceptions 1:**   |  |  |  | | --- | --- | --- | | Step | Cause | System Response | | 1 | Actor’s balance is not enough to pay for the match. | System shows warning message “Not enough money. Please deposit now”.  System shows screen to Deposit. |   **Relationships:** N/A  **Business Rules:**   * System gets Actor current position as default. * System will suggest field within search radius of Actor. * System calculates price for the match, if Actor don’t have enough money, system will switch Actor to deposit screen. * If Actor creates reservation request successfully, System will decrease Actor’s available balance, transfer the money to account payable. * Actor just can create reservation request from now to 7 days after. | | | |

##### <User> Find reservation request



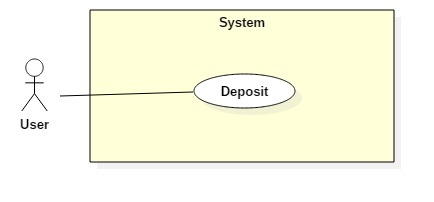
|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE – UC003** | | | |
| **Use Case No.** | 003 | **Use Case Version** | 1.0 |
| **Use Case Name** | Find reservation request | | |
| **Author** |  | | |
| **Date** | 24/10/2017 | **Priority** | Normal |
| **Actor:**   * User   **Summary:**   * This use case allows Actor to find reservation request based on time, location and rank.   **Goal:**   * Actor can find suitable reservation request base on time, location and rank.   **Triggers:**   * Actor sends find reservation request command.   **Preconditions:**   * Actor has accessed the system under user role.   **Post Conditions:**   * **Success:** list matching requests are shown. * **Fail:** N/A   **Main Success Scenario:**   |  |  |  | | --- | --- | --- | | Step | Actor Action | System Response | | 1 | Actor send find reservation request command. | System requires information from Actor:   * Start time. * End time. * Number of hours want to play. * Field type. * Current position of Actor. * Search radius. | | 2 | Actor send find command. | System will find in database any record of reservation request suitable for Actor and show records as list of matching requests. | | 3 | Actor choose 1 matching request.  Actor send accept command.  [Alternative 1]  [Alternative 2] | System removes reservation request. New match is created. System save new match information into database.  [Exception 1]  [Exception 2] |   **Alternative Scenario 1:**   |  |  |  | | --- | --- | --- | | Step | Actor Action | System Response | | 1 | Actor type other address in address field. | System support Actor choose position by Google API.  System update new position inputted by Actor. |   **Exception 1 :**   |  |  |  | | --- | --- | --- | | Step | Cause | System Response | | 1 | Actor did not input required field. | System shows warning message “Please input required field”. |   **Exceptions 2:**   |  |  |  | | --- | --- | --- | | Step | Cause | System Response | | 1 | Actor’s balance is not enough to pay for the match. | System shows warning message “Not enough money. Please deposit now”.  System shows screen to Deposit. |   **Relationships:** N/A  **Business Rules:**   * If Actor accepts reservation request successfully, system update balance. * If there is no suitable result when Actor send find command, system displays message: ”Không có đối thủ phù hợp” and allows Actor back to find reservation request. The inputted information was saved. * Actor can not find out opponent if the opponent added Actor to the black list. | | | |

##### <User> Cancel reservation request



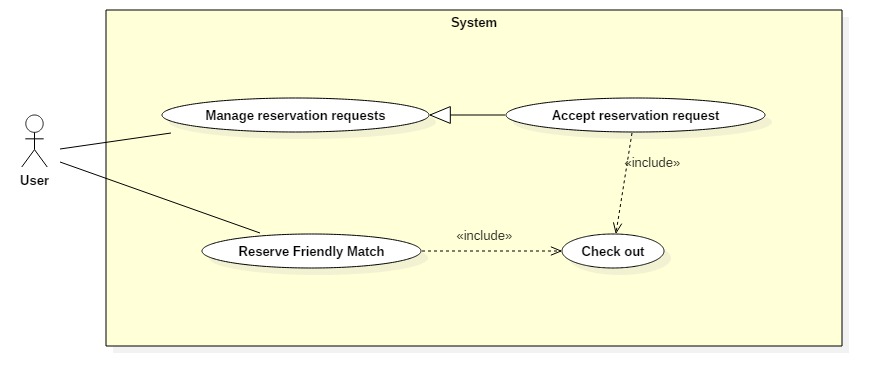
|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE – UC003** | | | |
| **Use Case No.** | 003 | **Use Case Version** | 1.0 |
| **Use Case Name** | Cancel reservation request | | |
| **Author** |  | | |
| **Date** | 28/10/2017 | **Priority** | Normal |
| **Actor:**   * User   **Summary:**   * This use case allows Actor to cancel reservation request.   **Goal:**   * Actor can cancel reservation request Actor created before.   **Triggers:**   * Actor sends cancel reservation request command.   **Preconditions:**   * Actor already signed in. * Actor cancel the reservation request he/she created. * No one accepts the reservation request yet.   **Post Conditions:**   * **Success:** User confirm matching request successfully. * **Fail:** N/A   **Main Success Scenario:**   |  |  |  | | --- | --- | --- | | Step | Actor Action | System Response | | 1 | Actor send cancel reservation request command. | System check if the reservation request is not accepted yet, then:   * System notify successfully cancel reservation request to Actor. * Delete Reservation request in database.   [Exception 1] |   **Exceptions 1:**   |  |  |  | | --- | --- | --- | | Step | Cause | System Response | | 1 | This reservation request was accepted by other User. | System shows warning message “Your reservation request was accepted”. |   **Relationships:** N/A  **Business Rules:** | | | |

##### <User> Deposit



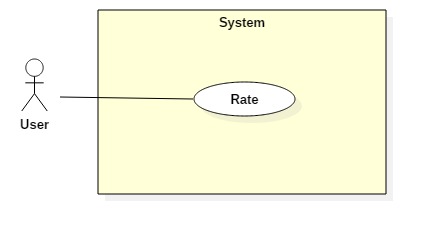
|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE – UC012** | | | |
| **Use Case No.** |  | **Use Case Version** | 1.0 |
| **Use Case Name** | Deposit | | |
| **Author** |  | | |
| **Date** |  | **Priority** | Normal |
| **Actor:**   * User   **Summary:**   * This use case allows Actor deposit.   **Goal:**   * System update Actor’s balance.   **Triggers:**   * User sends deposit command.   **Preconditions:**   * Actor has accessed the system under User role.   **Post Conditions:**   * **Success:** Update Actor’s balance in database. * **Fail:** N/A   **Main Success Scenario:**   |  |  |  | | --- | --- | --- | | Step | Actor Action | System Response | | 1 | User sends command to confirm to continue the payment.  [Alternative 1] | System switch to Paypal payment process. | |  | User completes payment process with Paypal.  [Alternative 2]  [Alternative 3]  [Alternative 4] | * System will send notification to User and show Actor’s balance. * Update Actor’s balance in database. |   **Alternative Scenario 1:**   |  |  |  | | --- | --- | --- | | Step | Actor Action | System Response | | 1 | User sends command to decline. | System switch to payment fail view. |   **Alternative Scenario 2:**   |  |  |  | | --- | --- | --- | | Step | Actor Action | System Response | | 1 | Paypal responses payment process is fail. | System switch to payment fail view. |   **Alternative Scenario 3:**   |  |  |  | | --- | --- | --- | | Step | Actor Action | System Response | | 1 | Paypal payment process time out. | System switch to payment fail view. |   **Alternative Scenario 4:**   |  |  |  | | --- | --- | --- | | Step | Actor Action | System Response | | 1 | User does the payment with not enough money in account. | System switch to payment fail view. |   **Relationships:**  **Business Rules:** | | | |

##### <User> Check out



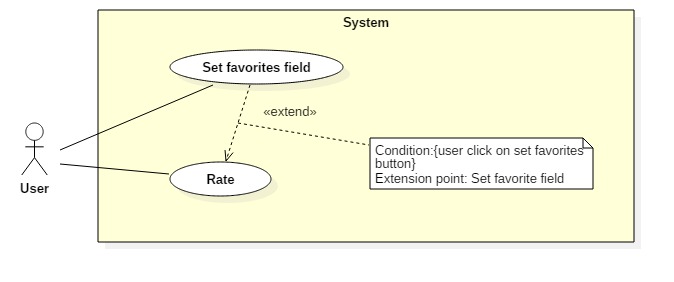
|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE – UC012** | | | |
| **Use Case No.** |  | **Use Case Version** | 1.0 |
| **Use Case Name** | Deposit | | |
| **Author** |  | | |
| **Date** |  | **Priority** | Normal |
| **Actor:**   * User   **Summary:**   * This use case allows Actor check out.   **Goal:**   * System pay the bill.   **Triggers:**   * User sends check out command.   **Preconditions:**   * Actor has accessed the system under User role.   **Post Conditions:**   * **Success:** Update Actor’s balance in database. * **Fail:** N/A   **Main Success Scenario:**   |  |  |  | | --- | --- | --- | | Step | Actor Action | System Response | | 1 | User sends command to check out. | System notify successfully and update balance in database.  [Exception 1] |   **Exceptions 1:**   |  |  |  | | --- | --- | --- | | Step | Cause | System Response | | 1 | Actor’s balance is not enough to pay the bill. | System shows warning message “Not enough money. Please deposit now”.  System shows link to Deposit. |   **Relationships:**  **Business Rules:** | | | |

##### <User> Rate



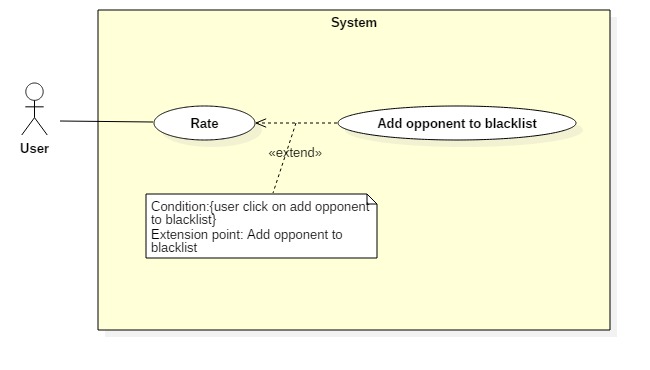
|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE – UC012** | | | |
| **Use Case No.** |  | **Use Case Version** | 1.0 |
| **Use Case Name** | Rate opponent | | |
| **Author** |  | | |
| **Date** |  | **Priority** | Normal |
| **Actor:**   * User   **Summary:**   * This use case allows Actor rates the opponent or field Actor had just played.   **Goal:**   * System calculates the ratting and save to database.   **Triggers:**   * User sends ratting command.   **Preconditions:**   * Actor has accessed the system under User role. * Ratting opponent Actor played with. * Ratting field Actor played.   **Post Conditions:**   * **Success:** The ratting is created, calculate and save to database. * **Fail:** N/A   **Main Success Scenario:**   |  |  |  | | --- | --- | --- | | Step | Actor Action | System Response | | 1 | Actor rates opponent from 1 to 3 stars. | System save the ratting from Actor, calculate ratting and update to database. |   **Relationships:**  **Business Rules:** | | | |

##### <User> Set favorites field



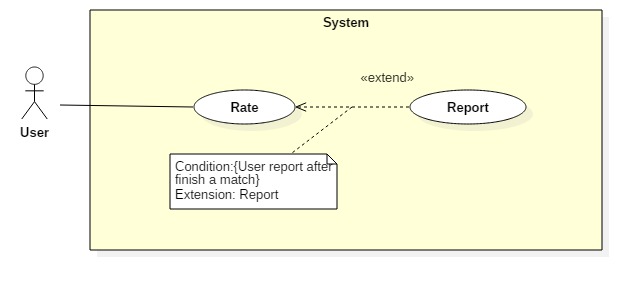
|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE – UC012** | | | |
| **Use Case No.** |  | **Use Case Version** |  |
| **Use Case Name** | Set favorites field | | |
| **Author** |  | | |
| **Date** |  | **Priority** | Normal |
| **Actor:**   * User   **Summary:**   * This use case allows Actor set a field as a favorite field.   **Goal:**   * System update favorite field list.   **Triggers:**   * Actor sends report command.   **Preconditions:**   * Actor has accessed the system under User role or Field Owner role.   **Post Conditions:**   * **Success:** The report is created. * **Fail:** N/A   **Main Success Scenario:**   |  |  |  | | --- | --- | --- | | Step | Actor Action | System Response | | 1 | Actor send add to favorite command. | System save field to Actor’s favorite field list. |   **Relationships:**  **Business Rules:**   * The field in favorite list has higher priority (than others field) when Actor search field. | | | |

##### <User> Set blacklist



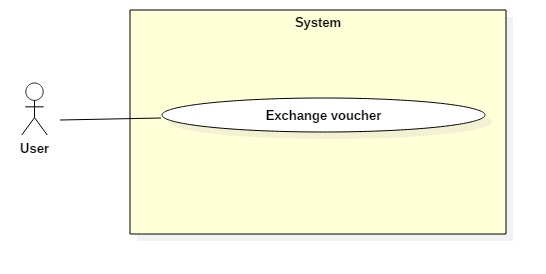
|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE – UC012** | | | |
| **Use Case No.** |  | **Use Case Version** | 1.0 |
| **Use Case Name** | Set blacklist | | |
| **Author** |  | | |
| **Date** |  | **Priority** | Normal |
| **Actor:**   * User   **Summary:**   * This use case allows Actor put a User to blacklist.   **Goal:**   * System update Actor’s black list.   **Triggers:**   * Actor sends report command.   **Preconditions:**   * Actor has accessed the system under User role or Field Owner role.   **Post Conditions:**   * **Success:** The report is created. * **Fail:** N/A   **Main Success Scenario:**   |  |  |  | | --- | --- | --- | | Step | Actor Action | System Response | | 1 | Actor send put User on blacklist command. | System save User to Actor blacklist. |   **Relationships:**  **Business Rules:**   * User that in Actor’s blacklist can not find out Actor when using find matching request function. | | | |

##### <User> Report



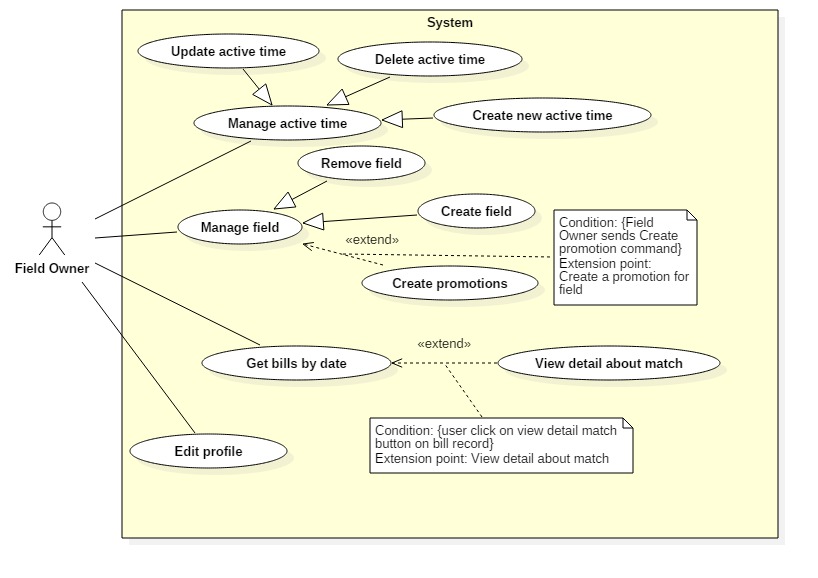
|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE – UC012** | | | |
| **Use Case No.** |  | **Use Case Version** | 1.0 |
| **Use Case Name** | Report | | |
| **Author** |  | | |
| **Date** |  | **Priority** | Normal |
| **Actor:**   * User   **Summary:**   * This use case allows Actor to report other User or Field Owner who has bad behavior.   **Goal:**   * System save the report to database.   **Triggers:**   * Actor sends report command.   **Preconditions:**   * Actor has accessed the system under User role or Field Owner role.   **Post Conditions:**   * **Success:** The report is created. * **Fail:** N/A   **Main Success Scenario:**   |  |  |  | | --- | --- | --- | | Step | Actor Action | System Response | | 1 | Actor inputs the reason why create report.  Actor send report command.  [Alternative 1] | System create the report and save the report to the database.  [Exception 1] |   **Alternative Scenario 1:**   |  |  |  | | --- | --- | --- | | Step | Actor Action | System Response | | 1 | User send cancel command. | * System back to previous page. * Report has not been create. |   **Exceptions 1:**   |  |  |  | | --- | --- | --- | | Step | Cause | System Response | | 1 | User inputs less than 10 characters. | System show warning message “Report must be 10 - 500 characters”. |   **Relationships:**  **Business Rules:** | | | |

##### <User> Exchange voucher

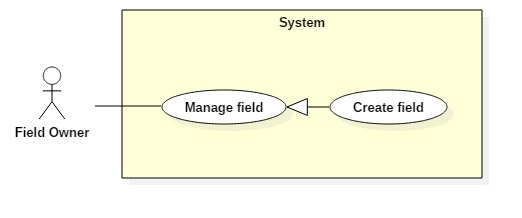


|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE – UC003** | | | |
| **Use Case No.** | 003 | **Use Case Version** | 1.0 |
| **Use Case Name** | Create reservation request | | |
| **Author** |  | | |
| **Date** | 28/10/2017 | **Priority** | Normal |
| **Actor:**   * User   **Summary:**   * This use case allows Actor to exchange bonus point to voucher.   **Goal:**   * Update voucher bonus point after exchange, save voucher in database.   **Triggers:**   * Actor sends exchange voucher command.   **Preconditions:**   * Actor has accessed the system under user role.   **Post Conditions:**   * **Success:** Actor update Actor’s balance and bonus point in database. * **Fail:** N/A   **Main Success Scenario:**   |  |  |  | | --- | --- | --- | | Step | Actor Action | System Response | | 1 | Actor sends command to exchange voucher | System search in database and response as a list of vouchers. | | 2 | Actor send exchange voucher command.  [Alternative 1] | System check if bonus point enough to exchange voucher then:   * Update Actor’s bonus point and balance. * Create voucher code, response to Actor and save to database.   [Exception 1] |   **Alternative Scenario 1:**   |  |  |  | | --- | --- | --- | | Step | Actor Action | System Response | | 1 | Actor send cancel command. | System back to previous window and allow Actor choose other voucher. |   **Exceptions 1:**   |  |  |  | | --- | --- | --- | | Step | Cause | System Response | | 1 | Actor’s bonus point is not enough to exchange the voucher. | System shows warning message “Not enough bonus point. Please choose other voucher”.  System back to previous window and allow Actor choose other voucher. |   **Relationships:** N/A  **Business Rules:** | | | |

#### Field Owner

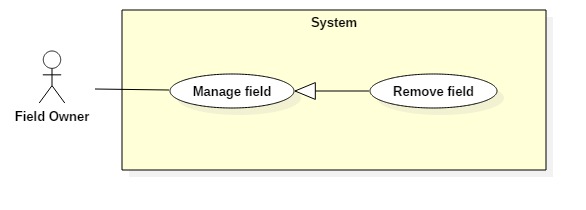


##### <Field Owner> Create field



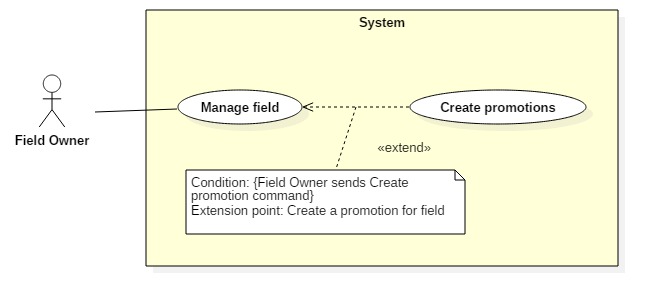
|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE – UC003** | | | |
| **Use Case No.** | 003 | **Use Case Version** | 1.0 |
| **Use Case Name** | Create field | | |
| **Author** |  | | |
| **Date** |  | **Priority** | Normal |
| **Actor:**   * Field Owner   **Summary:**   * This use case allows Actor to create field.   **Goal:**   * Actor can create a new field.   **Triggers:**   * Actor sends create field command.   **Preconditions:**   * Actor has accessed the system under field owner role.   **Post Conditions:**   * **Success:** Actor create field successfully and System save it to database. * **Fail:** N/A   **Main Success Scenario:**   |  |  |  | | --- | --- | --- | | Step | Actor Action | System Response | | 1 | Actor sends command to create field. | System requires information from Actor:   * Field name. * Field Type. | | 2 | Actor send create field command. | System will create a new field and save to database.  [Exception 1] |   **Exceptions 1:**   |  |  |  | | --- | --- | --- | | Step | Cause | System Response | | 1 | Actor input blank in required field. | System shows warning message “Please input required information”. |   **Relationships:** N/A  **Business Rules:** | | | |

##### <Field Owner> Delete field



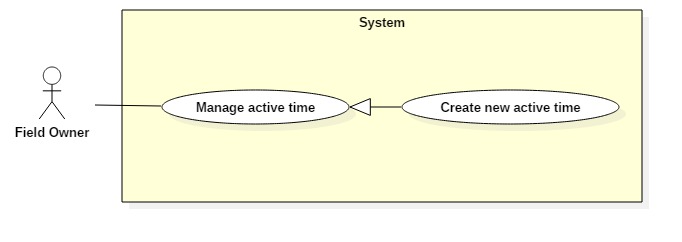
|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE – UC003** | | | |
| **Use Case No.** | 003 | **Use Case Version** | 1.0 |
| **Use Case Name** | Delete field | | |
| **Author** |  | | |
| **Date** |  | **Priority** | Normal |
| **Actor:**   * Field Owner   **Summary:**   * This use case allows Actor to delete field.   **Goal:**   * Actor can delete a field.   **Triggers:**   * Actor sends delete field command.   **Preconditions:**   * Actor has accessed the system under Field Owner role.   **Post Conditions:**   * **Success:** Actor delete field successfully. * **Fail:** N/A   **Main Success Scenario:**   |  |  |  | | --- | --- | --- | | Step | Actor Action | System Response | | 1 | Actor sends command to delete field. | System change the field’s status from active to pending. |   **Exceptions:** N/A  **Relationships:** N/A  **Business Rules:**   * After get change field’s status from active to pending, system will not allow to create new matching request after 3 days. | | | |

##### <Field Owner> Create promotion



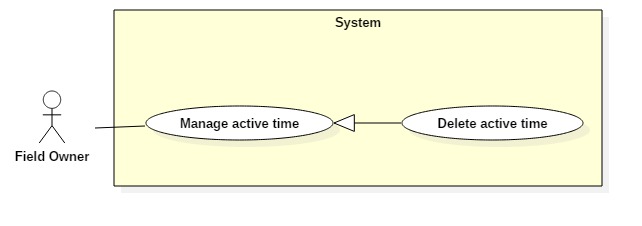
|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE – UC003** | | | |
| **Use Case No.** | 003 | **Use Case Version** | 1.0 |
| **Use Case Name** | Create promotion | | |
| **Author** |  | | |
| **Date** |  | **Priority** | Normal |
| **Actor:**   * Field Owner   **Summary:**   * This use case allows Actor to create promotion.   **Goal:**   * Actor can create a promotion.   **Triggers:**   * Actor sends create matching request command.   **Preconditions:**   * Actor has accessed the system under field owner role.   **Post Conditions:**   * **Success:** Actor create field successfully and System save it to database. * **Fail:** N/A   **Main Success Scenario:**   |  |  |  | | --- | --- | --- | | Step | Actor Action | System Response | | 1 | Actor sends command to create field. | System requires information from Actor:   * Start time * End time * Start date * End date * Promotion content * Percent discount * Field type | | 2 | Actor send create matching request command.  [Alternative 1] | * System will create a new promotion with inputted information and save to database.   [Exception 1] |   **Alternative Scenario 1:**   |  |  |  | | --- | --- | --- | | Step | Cause | System Response | | 1 | Actor send cancel command. | * System back to previous window. * Promotion will not be created. |   **Exceptions 1:**   |  |  |  | | --- | --- | --- | | Step | Cause | System Response | | 1 | Actor input blank in required field. | System shows warning message “Please input required information”. |   **Relationships:** N/A  **Business Rules:**   * Calculate bill with promotion’s price. | | | |

##### <Field Owner> Create active time



|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE – UC003** | | | |
| **Use Case No.** | 003 | **Use Case Version** | 1.0 |
| **Use Case Name** | Create active time | | |
| **Author** |  | | |
| **Date** |  | **Priority** | Normal |
| **Actor:**   * Field Owner   **Summary:**   * This use case allows Actor to create active time.   **Goal:**   * Actor can create active time for the field.   **Triggers:**   * Actor sends create active time command.   **Preconditions:**   * Actor has accessed the system under field owner role.   **Post Conditions:**   * **Success:** Actor create field successfully and System save it to database. * **Fail:** N/A   **Main Success Scenario:**   |  |  |  | | --- | --- | --- | | Step | Actor Action | System Response | | 1 | Actor sends command to create active time | System requires information from Actor:   * Start time * End time * Field Type * Day in week * Price of field per hour | | 2 | Actor send create active time command | * System will create active time and save to database   [Exception 1] |   **Exceptions 1:**   |  |  |  | | --- | --- | --- | | Step | Cause | System Response | | 1 | Actor input blank in required field | System shows warning message “Please input required information”. |   **Relationships:** N/A  **Business Rules:**   * Apply active time to all fields of Actor. | | | |

##### <Field Owner> Delete active time



|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE – UC003** | | | |
| **Use Case No.** | 003 | **Use Case Version** | 1.0 |
| **Use Case Name** | Delete active time | | |
| **Author** |  | | |
| **Date** |  | **Priority** | Normal |
| **Actor:**   * Field Owner   **Summary:**   * This use case allows Actor to delete active time.   **Goal:**   * Actor can delete active time.   **Triggers:**   * Actor sends delete active field command.   **Preconditions:**   * Actor has accessed the system under Field Owner role.   **Post Conditions:**   * **Success:** Actor delete active time successfully. * **Fail:** N/A   **Main Success Scenario:**   |  |  |  | | --- | --- | --- | | Step | Actor Action | System Response | | 1 | Actor sends command to delete field. | System changes the field’s status from active to pending. |   **Exceptions:** N/A  **Relationships:** N/A  **Business Rules:**   * After change active time’s status from active to pending, system will not allow to create new matching request after 3 days. | | | |

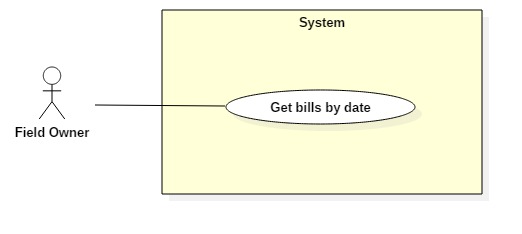
##### <Field Owner> Reserve match

|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE – UC003** | | | |
| **Use Case No.** | 003 | **Use Case Version** | 1.0 |
| **Use Case Name** | Reserve match | | |
| **Author** |  | | |
| **Date** | 24/10/2017 | **Priority** | Normal |
| **Actor:**   * Field Owner   **Summary:**   * This use case allows Actor to reserve match.   **Goal:**   * Actor can reserve a field.   **Triggers:**   * Actor sends reserve command.   **Preconditions:**   * Actor has accessed the system under field owner role.   **Post Conditions:**   * **Success:** Actor reserve Field successfully. * **Fail:** N/A   **Main Success Scenario:**   |  |  |  | | --- | --- | --- | | Step | Actor Action | System Response | | 1 | Actor chose expected field from list of field. | System finds active time of field and response to Actor. | | 2 | Actor inputs start time and end time.  Actor send Reserve command.  [Alternative 1] | System shows successful view.  System creates match in database.  [Exception 1] |   **Alternative Scenario 1:**   |  |  |  | | --- | --- | --- | | Step | Actor Action | System Response | | 1 | Actor cancel reservation request | System backs to list of fields window  Match did not save to database |   **Exceptions 1:**   |  |  |  | | --- | --- | --- | | Step | Cause | System Response | | 1 | There is no active time. | System shows warning message “Đặt sân không thành công”. |   **Exceptions :** N/A  **Relationships:** N/A  **Business Rules:**   * After get reserve command, the system will change status of active time in database. | | | |

##### <Field Owner> Set field for match

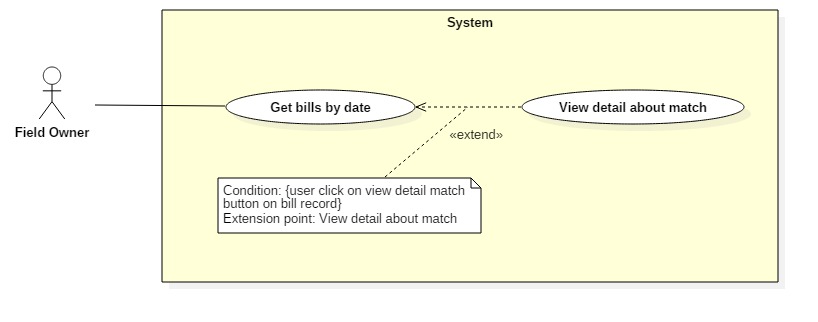
|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE – UC012** | | | |
| **Use Case No.** |  | **Use Case Version** | 1.0 |
| **Use Case Name** | Set blacklist | | |
| **Author** |  | | |
| **Date** |  | **Priority** | Normal |
| **Actor:**   * Field Owner   **Summary:**   * This use case allows Actor set field for match.   **Goal:**   * System update field for match.   **Triggers:**   * Actor sends set command.   **Preconditions:**   * Actor has accessed the system under Field Owner role.   **Post Conditions:**   * **Success:** Set field for match successfully. * **Fail:** N/A   **Main Success Scenario:**   |  |  |  | | --- | --- | --- | | Step | Actor Action | System Response | | 1 | Actor send put User on blacklist command. | System save User to Actor blacklist. |   **Relationships:**  **Business Rules:**   * User that in Actor’s blacklist can not find out Actor when using find matching request command. | | | |

##### <Field Owner> Get bill by date



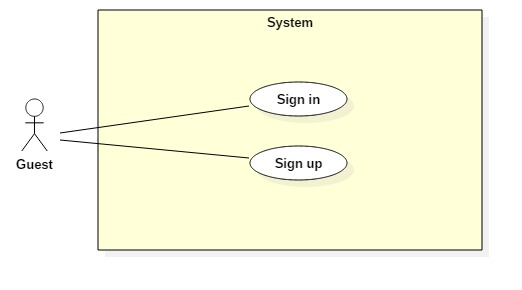
|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE – UC003** | | | |
| **Use Case No.** | 003 | **Use Case Version** | 1.0 |
| **Use Case Name** | Get bill by date | | |
| **Author** |  | | |
| **Date** |  | **Priority** | Normal |
| **Actor:**   * Field Owner   **Summary:**   * This use case allows Actor to get bill by input date.   **Goal:**   * Actor can see what bill be created in the day inputted.   **Triggers:**   * Actor sends get bill by date command.   **Preconditions:**   * Actor has accessed the system under field owner role.   **Post Conditions:**   * **Success:** System shows list of bill based on inputted date successfully. * **Fail:** N/A   **Main Success Scenario:**   |  |  |  | | --- | --- | --- | | Step | Actor Action | System Response | | 1 | Actor sends command to create field | System requires information from Actor:   * Start time * End time * Start date * End date * Promotion content * Percent discount * Field type | | 2 | Actor send create matching request command.  [Alternative 1] | * System will create a new promotion with inputted information and save to database.   [Exception 1] |   **Alternative Scenario 1:**   |  |  |  | | --- | --- | --- | | Step | Cause | System Response | | 1 | Actor send cancel command. | * System back to previous window. * Promotion will not be created. |   **Exceptions 1:**   |  |  |  | | --- | --- | --- | | Step | Cause | System Response | | 1 | Actor input blank in required field. | System shows warning message “Please input required information”. |   **Relationships:** N/A  **Business Rules:**   * Calculate bill with promotion’s price. | | | |

##### <Field Owner> View detail about match

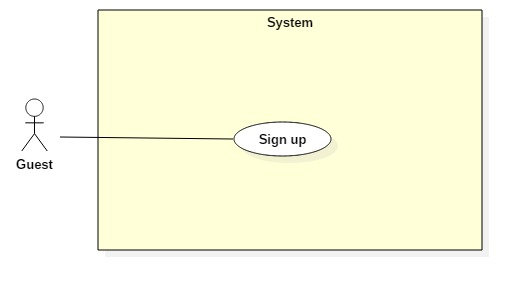


|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE – UC003** | | | |
| **Use Case No.** | 003 | **Use Case Version** | 1.0 |
| **Use Case Name** | View detail about match | | |
| **Author** |  | | |
| **Date** |  | **Priority** | Normal |
| **Actor:**   * Field Owner   **Summary:**   * This use case allows Actor to view detail about match.   **Goal:**   * Actor can see what bill be created in the day inputted.   **Triggers:**   * Actor sends get bill by date command.   **Preconditions:**   * Actor has accessed the system under field owner role.   **Post Conditions:**   * **Success:** System shows detail about match successfully. * **Fail:** N/A   **Main Success Scenario:**   |  |  |  | | --- | --- | --- | | Step | Actor Action | System Response | | 1 | Actor sends command to view detail about match | System responses details about match |   **Relationships:** N/A  **Business Rules:** | | | |

#### Guest



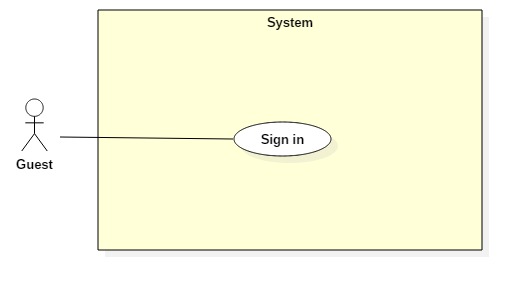
##### <Guest> Sign up



|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE – UC001** | | | |
| **Use Case No.** |  | **Use Case Version** | 1.0 |
| **Use Case Name** | Sign up | | |
| **Author** |  | | |
| **Date** |  | **Priority** | Normal |
| **Actor:**   * Guest   **Summary:**   * This use case allows Guest to sign up new account.   **Goal:**   * Account is sign up successfully and store in database of the system.   **Triggers:**   * Guest sends command to sign up.   **Preconditions:**   * Actor has not accessed in the system.   **Post Conditions:**   * **Success:** New account will be created. * **Fail:** Systemshows messages.   **Main Success Scenario:**   |  |  |  | | --- | --- | --- | | Step | Actor Action | System Response | | 1 | Guest sends command to request sign up. | System requires information from Guest:   * Email: free text input, required, regex [^[\_A-Za-z0-9-\\+]+(\\.[\_A-Za-z0-9-]+)\*@"+"[A-Za-z0-9-]+(\\.[A-Za-z0-9]+)\*(\\.[A-Za-z]{2,})$] * Full name: free text input, required, length (10-50) * Username: free text input, required, length (9-20) * Password: free text input, required, length (6-12) * Repeat password: free text input, required, length (6-12) * Phone: free number input, required, length(10-12) positive integer, value:[0,9] | | 2 | Guest inputs information |  | | 3 | Guest sends command to sign up.  [Alternative 1]  [Alternative 2] | System shows login view  Account registered.  [Exception 1]  [Exception 2]  [Exception 3]  [Exception 4] |   **Alternative Scenario 1:**   |  |  |  | | --- | --- | --- | | Step | Actor Action | System Response | | 1 | Guest sends command to reset. | System reset all field to blank. |   **Alternative Scenario 2:**   |  |  |  | | --- | --- | --- | | Step | Actor Action | System Response | | 1 | Guest send command to back to login view. | System shows login view.  Account isn’t created. |   **Exceptions 1:**   |  |  |  | | --- | --- | --- | | Step | Cause | System Response | | 1 | Guest input already exist username. | System show warning message “User name already exist”. |   **Exceptions 2:**   |  |  |  | | --- | --- | --- | | Step | Cause | System Response | | 1 | Guest input already exist email. | System shows warning message “Email already exist”. |   **Exceptions 3:**   |  |  |  | | --- | --- | --- | | Step | Cause | System Response | | 1 | Guest does not input required field. | System notices that guest need to input all these field:   * “Email”: System display warning message: “please fill out this field”. * “Full name”: System display warning message: “please fill out this field”. * “Username”: System display warning message: “please fill out this field”. * “Password”: System display warning message: “please fill out this field”. * “Repeat password”: System display warning message: “please fill out this field”. * “Phone”: System display warning message: “please fill out this field”. |   **Exceptions 4:**   |  |  |  | | --- | --- | --- | | Step | Cause | System Response | | 1 | Guest input wrong some fields with requirement. | System notices that guest need to re-input all these field:   * “Email”: System display warning message: “Email invalid! ([me@example.com)](mailto:me@example.com))”. * “Email”: System display warning message: “Email must be 10 - 254 characters”. * “Full name”: System display warning message: “Full name must be 10 - 50 characters”. * “Username”: System display warning message: “Username must be 6 - 20 characters”. * “Password”: System display warning message: “Password must be 6 - 12 characters”. * “Repeat password”: System display warning message: “Repeat password does not match password”. * “Phone”: System display warning message: “Phone must be numbers”. |   **Relationships:** N/A  **Business Rules:**   * After registered, information of account will be stored in database of the system with role “Owner” and status is “active” when using web. * After registered, information of account will be stored in database of the system with role “User” and status is “active” when using android app. | | | |

Table : Register specification

##### <Guest> Sign in



|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE – UC002** | | | |
| **Use Case No.** | 002 | **Use Case Version** | 1.0 |
| **Use Case Name** | Sign in | | |
| **Author** |  | | |
| **Date** | 24/10/2017 | **Priority** | Normal |
| **Actor:**   * Guest   **Summary:**   * This use case allows Guest sign in to the system on website and mobile application.   **Goal:**   * Guest sign in successfully with the proper role.   **Triggers:**   * Guest send the sign in command.   **Preconditions:**   * Guest has an account.   **Post Conditions:**   * **Success:** Guest accesses the system successfully. * **Fail:** System shows error message “Invalid username or password”.   **Main Success Scenario:**   |  |  |  | | --- | --- | --- | | Step | Actor Action | System Response | | 1 | Guest sends command to request sign in. | System requires identity information form Guest:   * Username: free text input. * Password: free text input. | | 2 | Guest inputs information |  | | 3 | Guest sends command to sign in to system.  [Alternative 1] | Guests will sign in system with their specific role. |   **Alternative Scenario:**   |  |  |  | | --- | --- | --- | | Step | Actor action | System Response | | 1 | Guest input invalid username or password. | System show error message “Invalid username or password”. |   **Exceptions:** N/A  **Relationships:** N/A  **Business Rules:**  - After sign in to system, guest will be redirected to specific view based on their role on the system: owner or user.   * If role is “User”, the system will display to User view. * If role is “Owner”, the system will display to Owner view. | | | |

Table : Sign in specification

#### Authenticated user



##### <Authenticated user> Sign out



|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE – UC010** | | | |
| **Use Case No.** |  | **Use Case Version** |  |
| **Use Case Name** | Sign out | | |
| **Author** |  | | |
| **Date** |  | **Priority** | Normal |
| **Actor:**   * Authenticated user.   **Summary:**   * This use case allows Authenticated user signs out the system.   **Goal:**   * Authenticated user signs out the system successfully, the session is killed.   **Triggers:**   * Authenticated User send request to sign out.   **Preconditions:**   * Actors has accessed the system.   **Post Conditions:**   * **Success:** Authenticated user logouts successfully. * **Fail:** N/A   **Main Success Scenario:**   |  |  |  | | --- | --- | --- | | Step | Actor Action | System Response | | 1 | Actor sends command to Sign out. | System clears session state if any, takes user out of the system.  System displays sign in view. |   **Exceptions:** N/A  **Relationships: N/A**  **Business Rules:**   * After sign out, role “Authenticated User” will become “Guest”. | | | |

#### Staff