# 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.

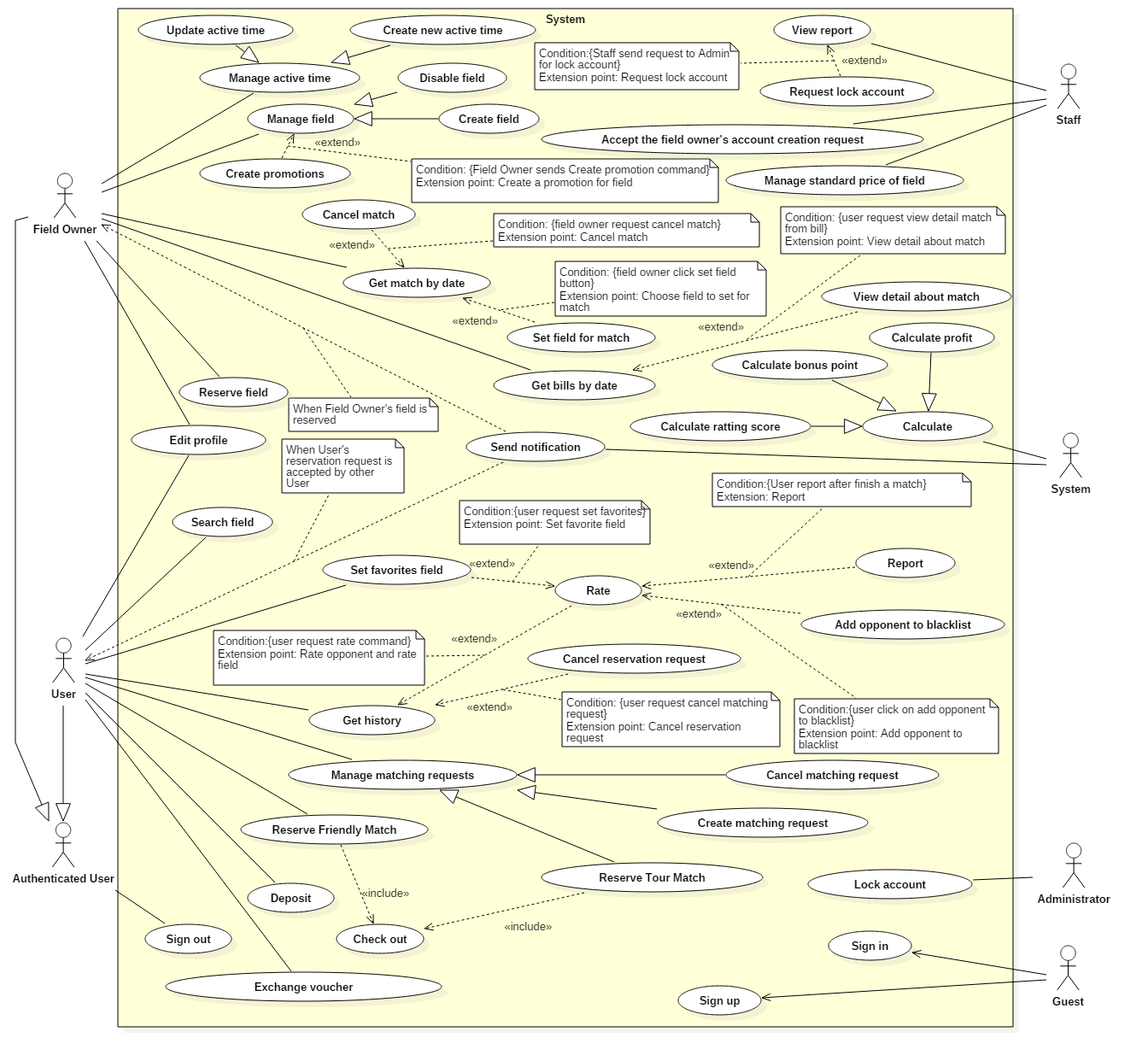
#### Software Interface

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

#### Communication Protocol

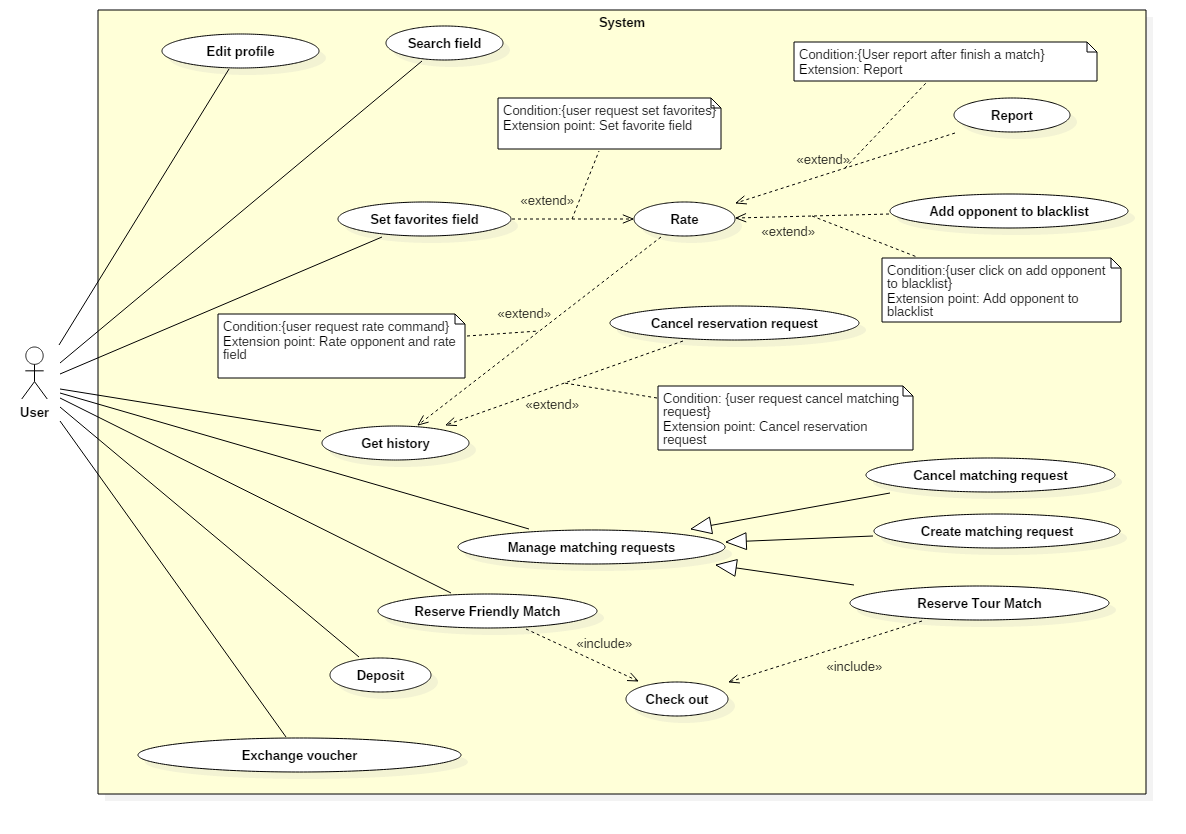
* Use HTTP protocol 1.1 for communication between the web browser and the web server.
* Use HTTP protocol 1.1 for communication between the mobile application and the web service.
* Use HTTP protocol 1.1 for communication between the server and the Microsoft service.

### System Overview Use Case

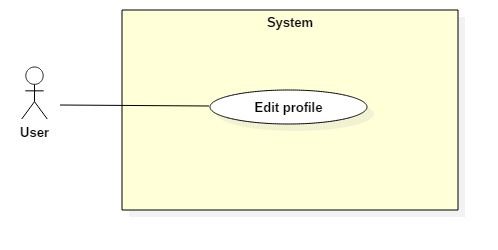


### List of Use Case

#### User Use case

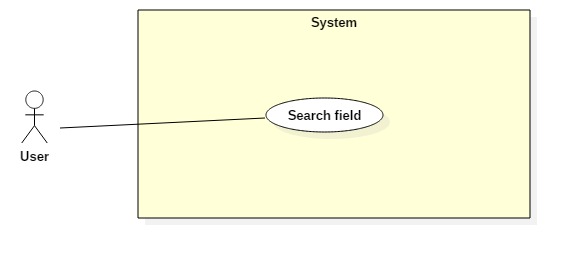


##### <User> Edit profile



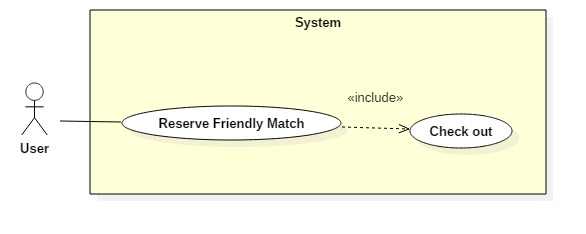
|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE – UC003** | | | |
| **Use Case No.** | **003** | **Use Case Version** | **1.0** |
| **Use Case Name** | Edit profile | | |
| **Author** |  | | |
| **Date** | 24/10/2017 | **Priority** | **Normal** |
| **Actor:**   * User, Field Owner   **Summary:**   * This use case allows Actor to edit account profile.   **Goal:**   * Profile of account are updated.   **Triggers:**   * Actor sends command to edit profile.   **Preconditions:**   * Actor has accessed the system under user role or field owner role.   **Post Conditions:**   * **Success:** Information of account will be updated and display. * **Fail:** System shows error messages.   **Main Success Scenario:**   |  |  |  | | --- | --- | --- | | Step | Actor Action | System Response | | 1 | Actor sends command to request edit profile. | System requires information from Actor.   * 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). System will automatically get the current position. * 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:**   |  |  |  | | --- | --- | --- | | No | Cause | System Response | | 1 | Actor input already exist phone. | System shows warning message “This phone number has been used to create an account”. | | 2 | Actor does not input required field. | System notices that Actor need to input all these 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”. * “Phone”: System display warning message: “please fill out this field”. | | 3 | Actor input wrong some fields with requirement. | System notices that Actor need to re-input all these field:   * “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”. * “Phone”: System display warning message: “Phone must be numbers”. |   **Relationships: N/A**  **Business Rules:**   * After edited, new information will be save to profile of account. * Approximate address depends on GPRS and Google maps. * Only Field Owner has longitude and latitude, it will be get based on address through Google API from input of user. | | | |

##### <User> Search Field



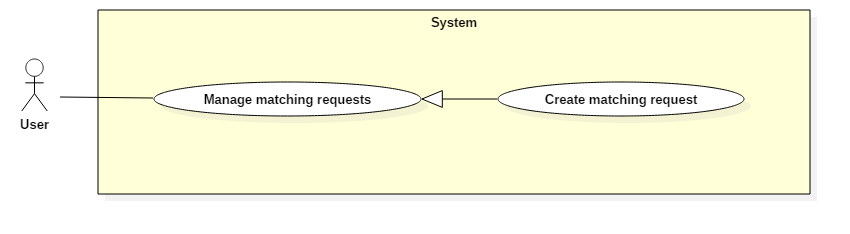
|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE – UC004** | | | |
| **Use Case No.** | 004 | **Use Case Version** | 1.0 |
| **Use Case Name** | Search Field | | |
| **Author** |  | | |
| **Date** | 24/10/2017 | **Priority** | Normal |
| **Actor:**   * User   **Summary:**   * This use case allows User to search Field Owner by input keyword.   **Goal:**   * User can find available Field Owner base on keyword.   **Triggers:**   * Actor sends search Field Owner command.   **Preconditions:**   * Actor has accessed the system under user role or field owner role.   **Post Conditions:**   * **Success:** list field owners have suitable name are display. * **Fail:** N/A   **Main Success Scenario:**   |  |  |  | | --- | --- | --- | | Step | Actor Action | System Response | | 1 | User inputs keyword to search |  | | 2 | User sends Search Field command.  [Alternative 1] | System will search and return any field owners whose name matches the keyword entered. |   **Alternative Scenario 1:**   |  |  |  | | --- | --- | --- | | Step | Actor Action | System Response | | 1 | User inputs blank in textbox. | System show 10 field owners near based on user’s location. |   **Exceptions :**   |  |  |  | | --- | --- | --- | | No | Cause | System Response | | 1 | No field owner matches the keyword inputted. | System shows warning message “Không tìm thấy sân phù hợp”. |   **Relationships:** N/A  **Business Rules:**   * Use likely search for search command | | | |

##### <User> Reserve Friendly match



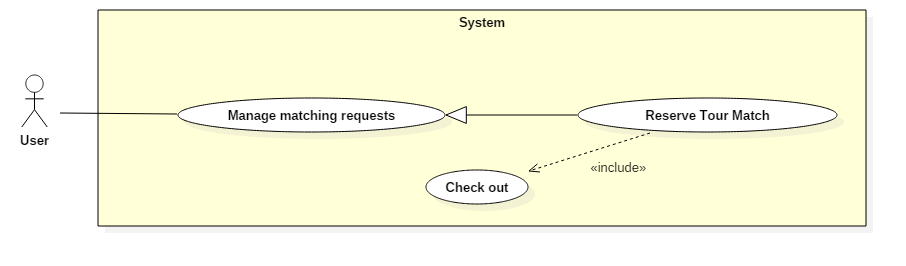
|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE – UC005** | | | |
| **Use Case No.** | 005 | **Use Case Version** | 1.0 |
| **Use Case Name** | Reserve Friendly match | | |
| **Author** |  | | |
| **Date** | 24/10/2017 | **Priority** | Normal |
| **Actor:**   * User   **Summary:**   * This use case allows User to reserve field by chose field and time.   **Goal:**   * User can reserve a field.   **Triggers:**   * User sends reserve command.   **Preconditions:**   * Actor has accessed the system under user role.   **Post Conditions:**   * **Success:** User reserve friendly match successfully. * **Fail:** N/A   **Main Success Scenario:**   |  |  |  | | --- | --- | --- | | Step | Actor Action | System Response | | 1 | User choose expected field from list of fields. | System finds uptime of field and response to User. | | 2 | User chooses field type and date. | System responses free time of field. | | 3 | User inputs start time, end time and sends reservation field command. | System responses information about match follow user’s choice (field owner, field type, start time, end time, price). | | 3 | User sends checkout command.  [Alternative 1] | System show successful view.  System will deduct money from the wallet of the user automatically, send notification to Field Owner and save information about friendly match.  [Exception 1]  [Exception 2] |   **Alternative Scenario 1:**   |  |  |  | | --- | --- | --- | | Step | Actor Action | System Response | | 1 | User cancels reservation command. | System back to information detail of field owner view |   **Exceptions:**   |  |  |  | | --- | --- | --- | | No | Cause | System Response | | 1 | Another reservation request has same time, that has reserved and paid off in advance. The time frame is already reserved by another user. | System shows message “Sân bạn đặt đã hết khung giờ trống”. | | 2 | User’s balance is not enough to pay for the match. | System shows message “Bạn không đủ tiền để thanh toán”.  System require deposit to checkout. |   **Relationships:** Include Checkout  **Business Rules:**   * For the field owner has an optimal time frame, user must reserve the entire time frame. (start time and end time of reservation request must equal start time and end time of optimal time frame). * For regular time frames, the minimum duration that a user must reserve is one hour. User is not limited about maximum duration as long as field owner have enough free time. * User can review the field reservation information in the reservation history. * The field reservation request of user is successful if and only if the user has paid. * After field reservation request to be paid, the system will get the field information, time then send notification to field owner and save information about friendly match. | | | |

##### <User> Create matching request



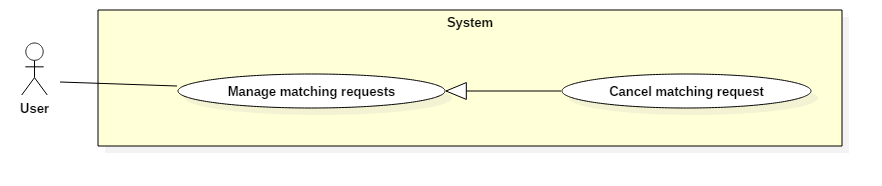
|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE – UC003** | | | |
| **Use Case No.** | 003 | **Use Case Version** | 2.0 |
| **Use Case Name** | Create Matching Request | | |
| **Author** |  | | |
| **Date** | 28/10/2017 | **Priority** | Normal |
| **Actor:**   * User   **Summary:**   * This use case allows User to create a matching request.   **Goal:**   * User can create a matching request.   **Triggers:**   * User sends create matching request command.   **Preconditions:**   * Actor has accessed the system under user role.   **Post Conditions:**   * **Success:** User creates matching request successfully. * **Fail:** N/A   **Main Success Scenario:**   |  |  |  | | --- | --- | --- | | Step | Actor Action | System Response | | 1 | User requires the matching request function. | System requires information from User:   * Start time and end time. * Duration. * Field type. * Distance expected.   System will automatically get the current location using GPS fill to address information. | | 2 | User inputs information and sends creation new matching request command.  [Alternative 1] | System takes information inputted and save information of matching request. System will deduct money from wallet of user automatically.  [Exception 1]  [Exception 2] |   **Alternative Scenario 1:**   |  |  |  | | --- | --- | --- | | Step | Actor Action | System Response | | 1 | User chooses other position by input address to “address” text input. | System call google map API to get address same with input text of user. |   **Exceptions:**   |  |  |  | | --- | --- | --- | | Step | Cause | System Response | | 1 | User’s balance is not enough to pay for the match. | System shows warning message “Không đủ tiền để tạo yêu cầu mới”.  System shows link to Deposit. | | 2 | User input blank in search address field or wrong address. | System shows warning message “Không tìm thấy địa chỉ phù hợp”. |   **Relationships:** N/A  **Business Rules:**   * Start time and end time when user input to create matching request is time frame that user is free. So, it may be larger than or equal to expected duration of user. * Duration of matching request must be longer than one hour. * The expected distance is radius for the system to find the opponent, it is also used as the search radius of the field. * System will take up some money from user’s wallet, this amount is half the maximum of reservation field fee. This amount will be refunded if the actual reservation field fee is lower. * After create new request, system will save information matching request and find suitable opponent. If the suitable opponent is found, the system will notify the user. | | | |

##### <User> Reserve Tour Match



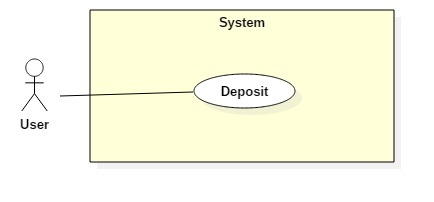
|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE – UC008** | | | |
| **Use Case No.** | 008 | **Use Case Version** | 1.0 |
| **Use Case Name** | Reserve Tour Match | | |
| **Author** |  | | |
| **Date** | 24/10/2017 | **Priority** | Normal |
| **Actor:**   * User   **Summary:**   * This use case allows user reserves tour match.   **Goal:**   * User can reserve tour match.   **Triggers:**   * User sends reservation tour match command.   **Preconditions:**   * Actor has accessed the system under user role.   **Post Conditions:**   * **Success:** User reserves tour match successfully. * **Fail:** N/A   **Main Success Scenario:**   |  |  |  | | --- | --- | --- | | Step | Actor Action | System Response | | 1 | User requires the matching request function.  [Alternative 1] | System requires information from User:   * Start time and end time. * Duration. * Field type. * Distance expected.   System will automatically get the current location using GPS fill to address information. | | 2 | User inputs information and sends find opponent command.  [Alternative 2] | System takes information inputted, find suitable opponents and return list suitable opponents.  Display list opponent.  [Exception 1]  [Exception 2] | | 3 | User chooses an opponent in list and sends choose suitable field command.  [Alternative 3] | System will find and return suitable field list.  Get first field in suitable field list and display  [Exception 3] | | 4 | User send checkout command.  [Alternative 4] | System show successful view.  System will deduct money from wallet of user automatically, send notification to Field Owner, opponent about tour match and save information about tour match.  [Exception 4]  [Exception 5] |   **Alternative Scenario 1:**   |  |  |  | | --- | --- | --- | | Step | Actor Action | System Response | | 1 | User receives notification about matching request has opponent |  |   **Alternative Scenario 2:**   |  |  |  | | --- | --- | --- | | Step | Actor Action | System Response | | 1 | User chose other position by input address to “address” text input. | System call google map API to get address same with input text of user.  [Exception 1] |   **Alternative Scenario 3:**   |  |  |  | | --- | --- | --- | | Step | Actor Action | System Response | | 1 |  | If duration of request is less than end time – start time of matching request.  System will require user input start time and end time equal expected duration. | | 2 | User input start time | System will response end time automatically with end time – start time equal duration of matching request. | | 3 | User sends choose suitable field command. | System will find and return list suitable field.  Get first field in list suitable field and display.  System will continue processing like step 4 of main scenario.  [Exception 3] |   **Alternative Scenario 4:**   |  |  |  | | --- | --- | --- | | Step | Actor Action | System Response | | 1 | User chooses another field in suitable field list. | System will display information of the selected field.  System will continue processing like step 4 of main scenario. |   **Exception 1 :**   |  |  |  | | --- | --- | --- | | No | Cause | System Response | | 1 | User input blank and wrong address. | System shows warning message “Không tìm thấy địa chỉ phù hợp”. | | 2 | System does not find suitable opponents | System shows warning message: “Không tìm thấy đối thủ phù hợp”. | | 3 | System does not find suitable field. | System shows warning message: “Không tìm thấy sân phù hợp.”  System will require extend searching radius. | | 4 | User’s balance is not enough to pay for the match. | System shows message “Bạn không đủ tiền để thanh toán”.  System shows “Nạp tiền” button to Deposit. | | 5 | Another reservation request has same time, that has reserved and paid off in advance. The time frame is already reserved by another user. | System shows message “Sân bạn đặt đã hết khung giờ trống”.  System will get another field in list suitable field and display. |   **Relationships:** Include Checkout  **Business Rules:**   * System will find similar matching with the criteria:   + Duration is equal or longer than duration of user inputted  + End time of information inputted must be longer than start time of matching request in database.  + Location of user have matching request in database must be within the searching radius request.  + Have same field type.  + 2 users must have difference ranking score is not exceed 50 points.  + Opponent is not in the user’s blacklist.   * System will suggest suitable field for match with criteria:   + Search for field in the search radius requested by two users, the returned list contain common field of two lists found.  + If in the returned list, there are common favorites field of two user or one of two users, system will only recommend these fields.  + If in the returned list, there is no favorites field of any user, system will return all returned list   * User pays haft reservation field fee; the other half is paid by opponent. * When user reserves tour match successful, system will deduct money from wallet of opponent and return money that was taken when the opponent created matching request. | | | |

##### <User> Cancel matching request



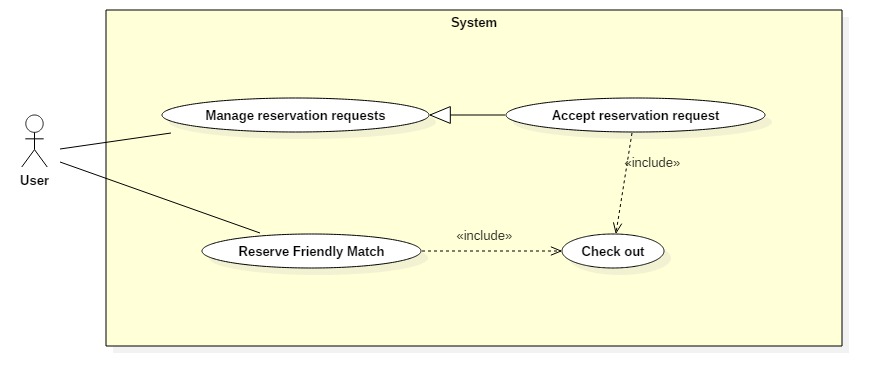
|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE – UC007** | | | |
| **Use Case No.** | 007 | **Use Case Version** | 1.0 |
| **Use Case Name** | Cancel matching request | | |
| **Author** |  | | |
| **Date** | 28/10/2017 | **Priority** | Normal |
| **Actor:**   * User   **Summary:**   * This use case allows User to cancel matching request.   **Goal:**   * User can cancel matching request.   **Triggers:**   * User sends cancel reservation request command.   **Preconditions:**   * Actor has accessed the system under user role. * Matching request of user is not accepted by another user.   **Post Conditions:**   * **Success:** User cancel matching request successfully. * **Fail:** N/A   **Main Success Scenario:**   |  |  |  | | --- | --- | --- | | Step | Actor Action | System Response | | 1 | User sends request to see list of matching requests. | System will response list of matching requests was created by user before.  [Exception 1] | | 2 | User choose a matching request and send cancel matching request command. | System check if the matching request is not accepted yet, then system notify successfully cancel matching request to User and display history match request of user.  System will refund money that was taken when user creates matching request.  [Exception 2] |   **Exceptions 1:**   |  |  |  | | --- | --- | --- | | Step | Cause | System Response | | 1 | User not has any matching request. | System will show message: “Bạn không có yêu cầu tìm kiếm đối thủ nào”. | | 2 | This matching request was accepted by another User. | System shows warning message “Yêu cầu tìm kiếm đối thủ của bạn đã được chấp nhận”. |   **Relationships:** N/A  **Business Rules:**   * After the matching request is canceled, system will save to history. User can view this history. If a user cancels too many requests, the system will alert the user. * If matching request is accepted, the user can still cancel tour match but user must compensate. | | | |

##### <User> Deposit



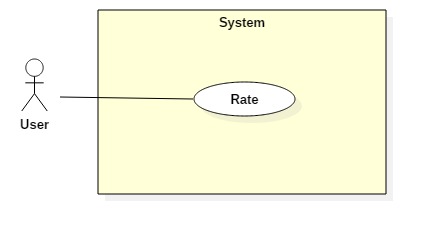
|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE – UC016** | | | |
| **Use Case No.** | 016 | **Use Case Version** | 1.0 |
| **Use Case Name** | Deposit | | |
| **Author** |  | | |
| **Date** |  | **Priority** | Normal |
| **Actor:**   * User   **Summary:**   * This use case allows Actor deposit money to wallet in system.   **Goal:**   * Actor can increase balance of user’s wallet in system.   **Triggers:**   * Actor sends deposit command.   **Preconditions:**   * Actor has accessed the system under User or Field owner role.   **Post Conditions:**   * **Success:** Actor deposits money to wallet in system successful. * **Fail:** N/A   **Main Success Scenario:**   |  |  |  | | --- | --- | --- | | Step | Actor Action | System Response | | 1 | Actor sends command to requires deposit function. | System will show current balance and available balance and requires user input the amount that the actor needs to deposit. | | 2 | Actor inputs the amount and sends deposit command. | System will show many options about form for actor to deposit. | | 3 | Actor chooses a form to deposit | System will call 3rd party payment gateways.  [Exception 1] | | 4 | Actor performs the payment  [Alternative 1] | System will show messages “Bạn đã nạp tiền thành công” and show account’s balance.  [Exception 2] |   **Alternative Scenario 1:**   |  |  |  | | --- | --- | --- | | Step | Actor Action | System Response | | 1 | Actor sends command to decline. | System displays payment gateways for user chooses. |   **Relationships:** N/A  **Business Rules:**   * The currency of the system is Vietnam dong. * If the user’s payment gateways use a difference currency or not supports for Vietnam dong currency, the system will support convert automatically. * The system is not responsible when the users have problems with their payment gateway. | | | |

##### <User> Check out



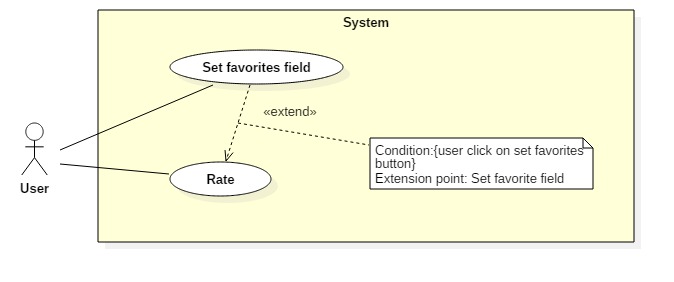
|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE – UC012** | | | |
| **Use Case No.** |  | **Use Case Version** | 1.0 |
| **Use Case Name** | Deposit | | |
| **Author** |  | | |
| **Date** |  | **Priority** | Normal |
| **Actor:**   * User   **Summary:**   * This use case allows Actor check out.   **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



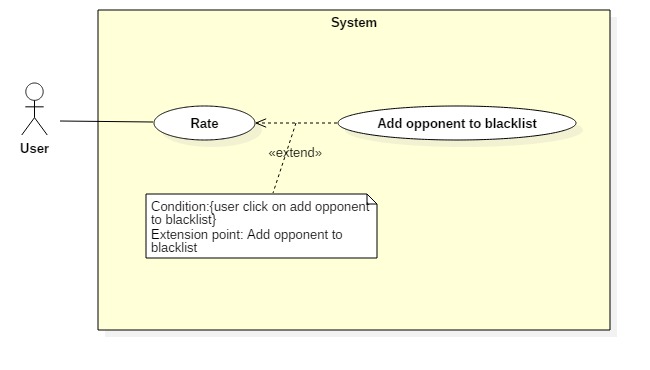
|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE – UC012** | | | |
| **Use Case No.** |  | **Use Case Version** | 1.0 |
| **Use Case Name** | Rate opponent | | |
| **Author** |  | | |
| **Date** |  | **Priority** | Normal |
| **Actor:**   * User   **Summary:**   * This use case allows Actor rates 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



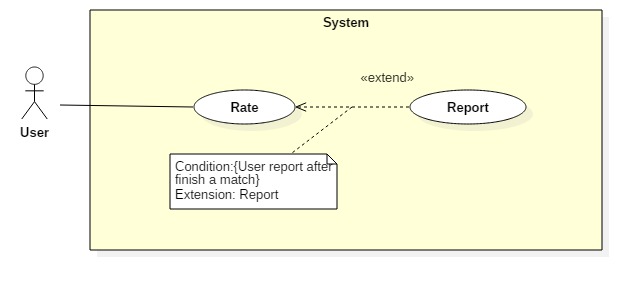
|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE – UC012** | | | |
| **Use Case No.** |  | **Use Case Version** |  |
| **Use Case Name** | Set favorites field | | |
| **Author** |  | | |
| **Date** |  | **Priority** | Normal |
| **Actor:**   * User   **Summary:**   * This use case allows Actor set 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



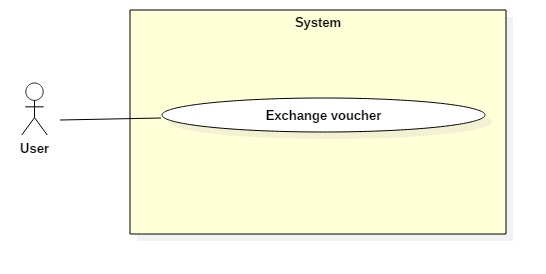
|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE – UC012** | | | |
| **Use Case No.** |  | **Use Case Version** | 1.0 |
| **Use Case Name** | Set blacklist | | |
| **Author** |  | | |
| **Date** |  | **Priority** | Normal |
| **Actor:**   * User   **Summary:**   * This use case allows Actor put a 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



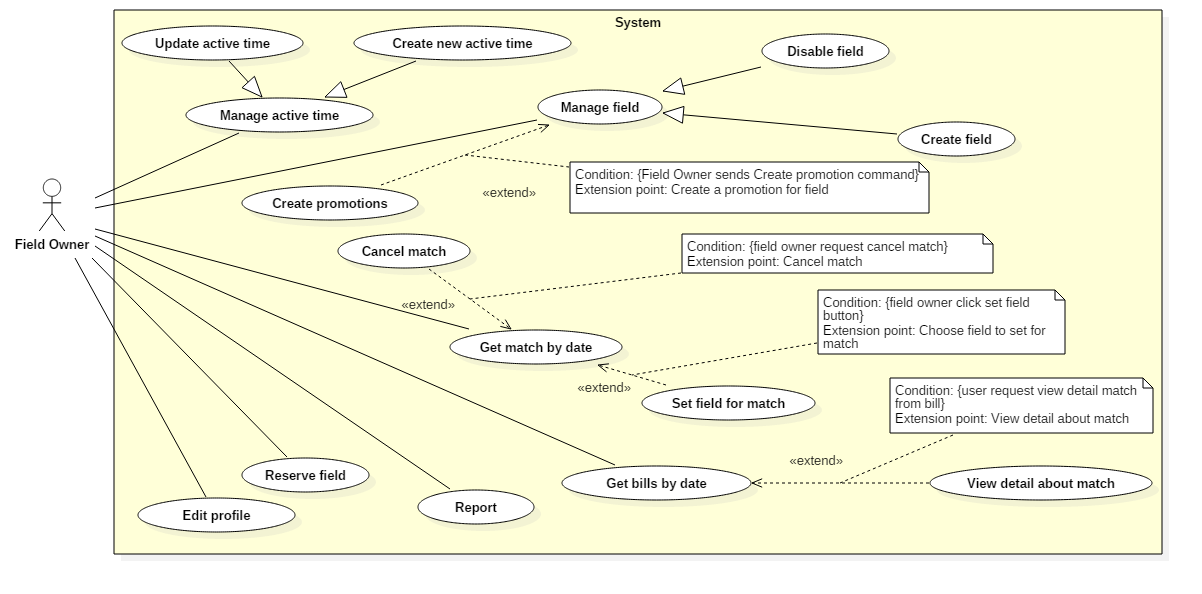
|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE – UC012** | | | |
| **Use Case No.** |  | **Use Case Version** | 1.0 |
| **Use Case Name** | Report | | |
| **Author** |  | | |
| **Date** |  | **Priority** | Normal |
| **Actor:**   * User   **Summary:**   * This use case allows Actor to report other 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

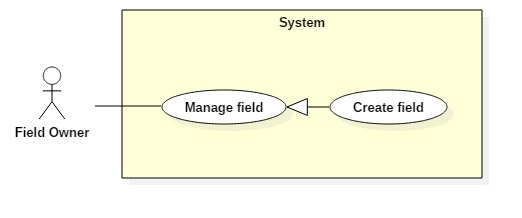


|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE – UC003** | | | |
| **Use Case No.** | 003 | **Use Case Version** | 1.0 |
| **Use Case Name** | Create reservation request | | |
| **Author** |  | | |
| **Date** | 28/10/2017 | **Priority** | Normal |
| **Actor:**   * User   **Summary:**   * This use 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

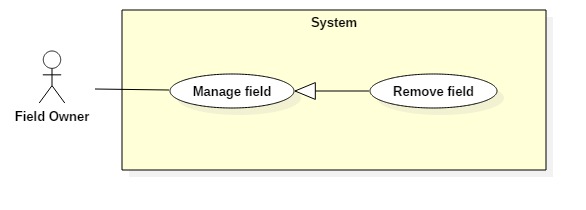


##### <Field Owner> Create field



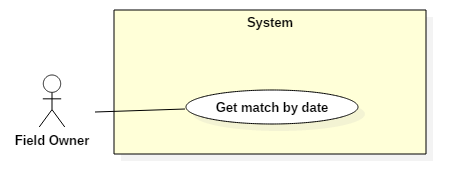
|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE – UC003** | | | |
| **Use Case No.** | 003 | **Use Case Version** | 1.0 |
| **Use Case Name** | Create field | | |
| **Author** |  | | |
| **Date** |  | **Priority** | Normal |
| **Actor:**   * Field Owner   **Summary:**   * This use case allows Field owner to create field.   **Goal:**   * Field Owner can create a new field.   **Triggers:**   * Field owner sends create field command.   **Preconditions:**   * Actor has accessed the system under field owner role.   **Post Conditions:**   * **Success:** Field owner create field successfully. * **Fail:** N/A   **Main Success Scenario:**   |  |  |  | | --- | --- | --- | | Step | Actor Action | System Response | | 1 | Field owner sends a request to create field. | System requires information:   * Field name. * Field Type. | | 2 | Field owner sends create field command. | System will create a new field.  [Exception 1]  [Exception 2] |   **Exceptions:**   |  |  |  | | --- | --- | --- | | No | Cause | System Response | | 1 | Field owner inputs blank in field name. | System shows warning message “Vui lòng nhập tên sân”. | | 2 | Field owner inputs field name already exists. | System shows warning message “Tên sân này đã tồn tại”. |   **Relationships:** N/A  **Business Rules:**   * Field owner must be responsible for creating new field because the field can be reserved as soon as the new field is created finished. | | | |

##### <Field Owner> Delete field



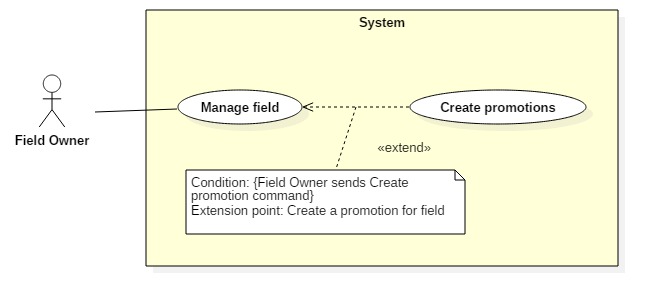
|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE – UC003** | | | |
| **Use Case No.** | 003 | **Use Case Version** | 1.0 |
| **Use Case Name** | Delete field | | |
| **Author** |  | | |
| **Date** |  | **Priority** | Normal |
| **Actor:**   * Field Owner   **Summary:**   * This use case allows Actor 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> Get match by date



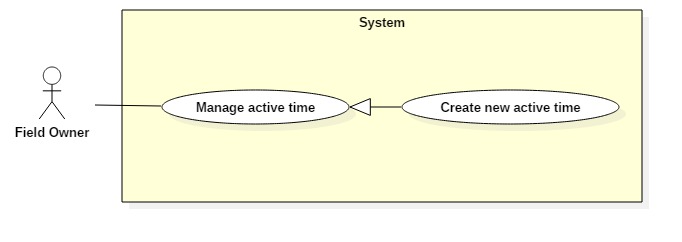
|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE – UC0019** | | | |
| **Use Case No.** | 019 | **Use Case Version** | 1.0 |
| **Use Case Name** | Get match by date | | |
| **Author** | QuyMM | | |
| **Date** |  | **Priority** | Normal |
| **Actor:**   * Field Owner   **Summary:**   * This use case allows Field owner get match by date.   **Goal:**   * Field Owner can get match by date.   **Triggers:**   * Field owner sends get match by date command.   **Preconditions:**   * Actor has accessed the system under field owner role.   **Post Conditions:**   * **Success:** Field owner viewed matches by date. * **Fail:** N/A   **Main Success Scenario:**   |  |  |  | | --- | --- | --- | | Step | Actor Action | System Response | | 1 | Field owner sends a request to view reserved matches. | System return reserved matches today and display.  [Exception 1] | | 2 | Field owner chooses a date in the future.  [Alternative 1] | System will return reserved matches in the day selected and display.  [Exception 1] |   **Alternative Scenario 1:**   |  |  |  | | --- | --- | --- | | Step | Cause | System Response | | 1 | Field owner sends a request to get matches in the next day | System will return reserved matches in the next day and display. |   **Exceptions:**   |  |  |  | | --- | --- | --- | | No | Cause | System Response | | 1 | There are no matches today | System shows warning message “Không có trận đấu nào được đặt hôm nay”. |   **Relationships:** N/A  **Business Rules:**   * Field owner can only view matches in the current and in the next 7 days because the user only reserved field within 7 days. * The system will use difference color when display matches to help field owner distinguish matches are reserved by field owner and matches are reserved by user. * The field owner can view matches of the day on the field if the matches was set field. | | | |

##### <Field Owner> Create promotion



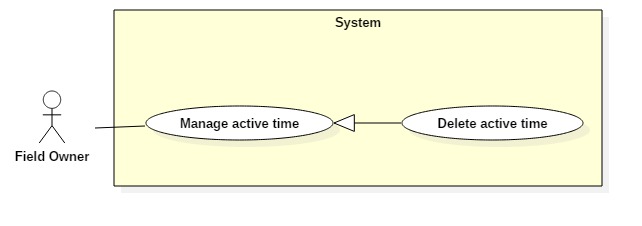
|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE – UC003** | | | |
| **Use Case No.** | 003 | **Use Case Version** | 1.0 |
| **Use Case Name** | Create promotion | | |
| **Author** |  | | |
| **Date** |  | **Priority** | Normal |
| **Actor:**   * Field Owner   **Summary:**   * This use case allows 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. | | | |

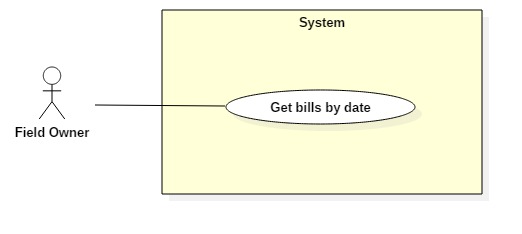
##### <Field Owner> Reserve match

|  |  |  |  |
| --- | --- | --- | --- |
| **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

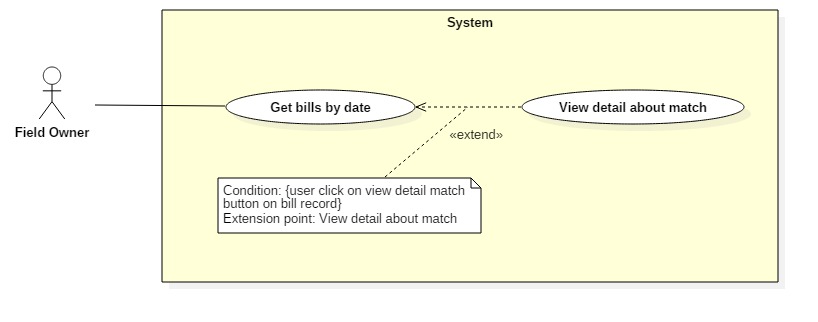
|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE – UC012** | | | |
| **Use Case No.** |  | **Use Case Version** | 1.0 |
| **Use Case Name** | Set blacklist | | |
| **Author** |  | | |
| **Date** |  | **Priority** | Normal |
| **Actor:**   * Field Owner   **Summary:**   * This use case allows Actor 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



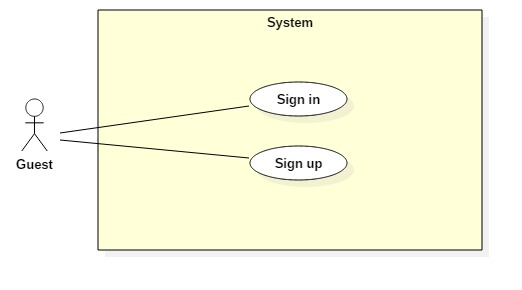
|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE – UC003** | | | |
| **Use Case No.** | 003 | **Use Case Version** | 1.0 |
| **Use Case Name** | Get bill by date | | |
| **Author** |  | | |
| **Date** |  | **Priority** | Normal |
| **Actor:**   * Field Owner   **Summary:**   * This use case allows 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

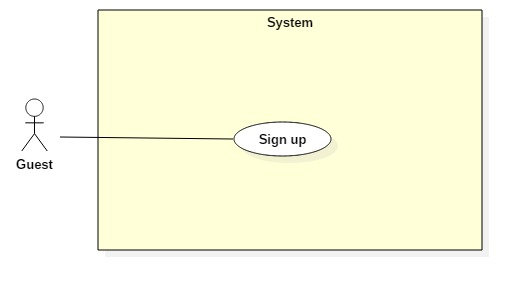


|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE – UC003** | | | |
| **Use Case No.** | 003 | **Use Case Version** | 1.0 |
| **Use Case Name** | View detail about match | | |
| **Author** |  | | |
| **Date** |  | **Priority** | Normal |
| **Actor:**   * Field Owner   **Summary:**   * This use case 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



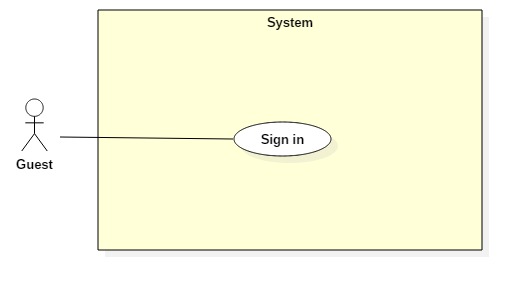
##### <Guest> Sign up



|  |  |  |  |
| --- | --- | --- | --- |
| **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