# 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 (Template)

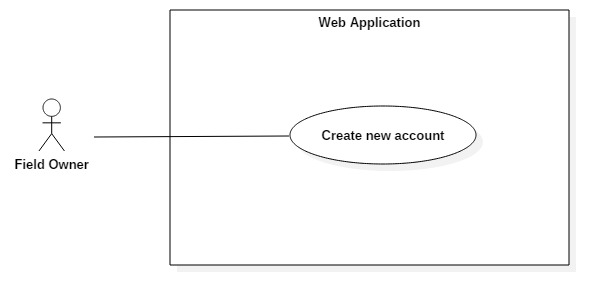
### List of Use Case

#### Field Owner Overview Use Case

<Hinh o day>

*Figure x: <Admin> Overview Use Case*

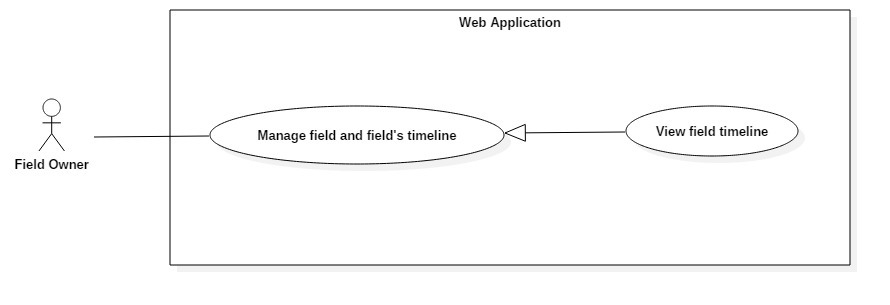
##### <Field Owner> Create account (FO01)



*Figure x+1: <Field Owner> Create account (*FO01*)*

|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE – UC\_FO01** | | | |
| **Use Case No.** | FO01 | **Use Case Version** | 0.1 |
| **Use Case Name** | Create account | | |
| **Author** | HuanPM | | |
| **Date** | 17/9/2017 | **Priority** | Normal |
| **Actor:**   * Field Owner   **Summary:**   * This use case allows field owner to create a new account on the system.   **Goal:**   * Field owner have account for logging in.   **Triggers:**   * Field owner open web application.   **Preconditions:**   * N/A.   **Post Conditions:**   * **Success:** An account is created on system. System displays successful message. * **Fail:** System displays error message.   **Main Success Scenario:**   |  |  |  | | --- | --- | --- | | Step | Actor Action | System Response | | 1 | Actor goes to Create account view. | System displays Create account view and requires information from field owner:   * **Username**: free text input, required, length 8–40 characters. * **Field name**: free text input, required, length 8–40 characters. * **Phone**: free text input, required, length 10-11 digits. * **ATM card no:** digit input, required, length 16 digits. * **CVC:** digit input, required, length 3 to 4 digits. | | 2 | Actor inputs information.  [Alternative 1] |  | | 3 | Actor sends command to create account.  [Alternative 1] | System validates information.  [Exception 1, 2, 3, 4, 5] | | 4 |  | System adds a new account.  System display successful message. |   **Alternative Scenario:**  *[Alternative 1]*   |  |  |  | | --- | --- | --- | | Step. | Actor Action | System Response | | 1 | Actor sends command to cancel. | System cancels creating new account and redirect to home page. |   **Exceptions:**   |  |  |  | | --- | --- | --- | | No. | Cause | System Response | | 1 | Invalid Username format. | System shows error message: “Username cannot be blank and must be 8-40 characters long”. | | 2 | Invalid Field name format. | System shows error message: “Field name cannot be blank and must be 8-40 characters long”. | | 3 | Invalid Phone format. | System shows error message: “Phone cannot be blank or contain alphabetical characters and must be 9-11 digits”. | | 4 | Invalid ATM Card format | System shows error message: “ATM Card cannot be blank and must be 16 digits”. | | 5 | Invalid CVC format. | System shows error message: “CVC name cannot be blank and must be 3-4 digits”. |   **Relationships:**   * N/A   **Business Rules:**   * N/A | | | |
|  | | | |

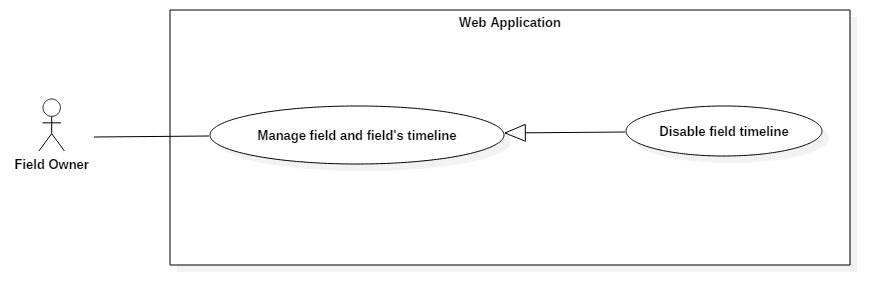
##### <Field Owner> View field timeline (FO02)



*Figure x+1: <Field Owner> View field timeline (FO02)*

|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE – UC\_FO01** | | | |
| **Use Case No.** | FO02 | **Use Case Version** | 0.1 |
| **Use Case Name** | View field timeline | | |
| **Author** | HuanPM | | |
| **Date** | 17/9/2017 | **Priority** | High |
| **Actor:**   * Field Owner   **Summary:**   * This use case allows field owner to view field’s timeline.   **Goal:**   * Field owner can view timeline of a specific field.   **Triggers:**   * Field owner open web application.   **Preconditions:**   * Field owner logged in to the system.   **Post Conditions:**   * **Success:** System displays field timeline. * **Fail:** System displays error message.   **Main Success Scenario:**   |  |  |  | | --- | --- | --- | | Step | Actor Action | System Response | | 1 | Actor goes to Field management view. | System shows field list.  [Exception 1] | | 2 | Actor clicks a field  [Alternative 1] | System displays field details and timeline. |   **Alternative Scenario:**  *[Alternative 1]*   |  |  |  | | --- | --- | --- | | Step. | Actor Action | System Response | | 1 | Actor clicks a disabled field | System displays field details and enable button, |   **Exceptions:**   |  |  |  | | --- | --- | --- | | No. | Cause | System Response | | 1 | There are no fields. | System shows message: “The list is empty”. |   **Relationships:**   * N/A   **Business Rules:**   * N/A | | | |
|  | | | |

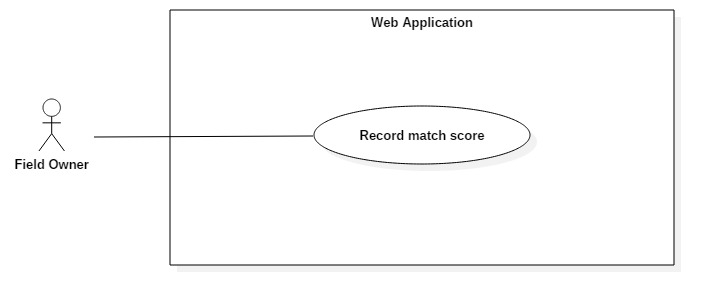
##### <Field Owner>Disable field timeline (FO03)



*Figure x+1: <Field Owner> Disable field timeline (FO03)*

|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE – UC\_FO01** | | | |
| **Use Case No.** | FO03 | **Use Case Version** | 0.1 |
| **Use Case Name** | Disable field timeline | | |
| **Author** | HuanPM | | |
| **Date** | 17/9/2017 | **Priority** | High |
| **Actor:**   * Field Owner   **Summary:**   * This use case allows field owner to disable specific field’s timeline.   **Goal:**   * Field owner can disable a specific timeline of a specific field.   **Triggers:**   * Field owner open web application.   **Preconditions:**   * Field owner logged in to the system.   **Post Conditions:**   * **Success:** System displays successful message. * **Fail:** System displays error message.   **Main Success Scenario:**   |  |  |  | | --- | --- | --- | | Step | Actor Action | System Response | | 1 | Actor goes to Field management view. | System shows field list.  [Exception 1] | | 2 | Actor clicks Disable button on a field. | System displays a modal and requires information from field owners.   * From hours: * From date: * To hours: * To date: | | 3 | Actor inputs required information.  [Alternative 1] | System validates information.  [Exception 2,3,4,5] | | 4 | Actor send commands to submit.  [Alternative 1] | System displays successful message. |   **Alternative Scenario:**  *[Alternative 1]*   |  |  |  | | --- | --- | --- | | Step. | Actor Action | System Response | | 1 | Actor clicks Cancel on modal | System cancels operation and close modal. |   **Exceptions:**   |  |  |  | | --- | --- | --- | | No. | Cause | System Response | | 1 | There are no fields. | System shows message: “The list is empty”. | | 2 |  |  | | 3 |  |  | | 4 |  |  | | 5 |  |  |   **Relationships:**   * N/A   **Business Rules:**   * N/A | | | |
|  | | | |

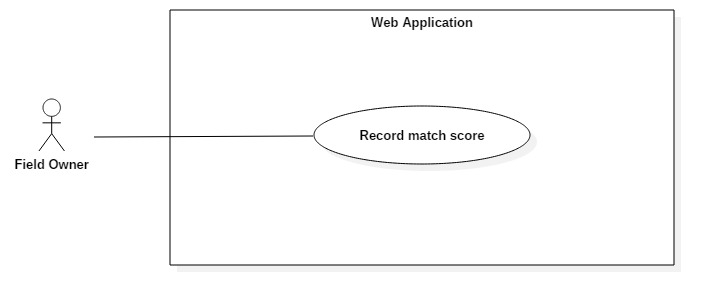
##### <Field Owner>Record match score (FO04)



*Figure x+1: <Field Owner> Record match score (FO04)*

|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE – UC\_FO01** | | | |
| **Use Case No.** | FO04 | **Use Case Version** | 0.1 |
| **Use Case Name** | Record match score | | |
| **Author** | HuanPM | | |
| **Date** | 17/9/2017 | **Priority** | Medium |
| **Actor:**   * Field Owner   **Summary:**   * This use case allows field owner to record match score.   **Goal:**   * Field owner can record match score.   **Triggers:**   * Field owner open web application. * Field owner choose notification about finished match.   **Preconditions:**   * Field owner logged in to the system. * Match has to be finished. * Notification has to be sent to field owner.   **Post Conditions:**   * **Success:** System displays successful message. * **Fail:** System displays error message.   **Main Success Scenario:**   |  |  |  | | --- | --- | --- | | Step | Actor Action | System Response | | 1 | Actor clicks Notifications | System shows notification list.  [Exception 1] | | 2 | Actor clicks a match notification | System displays a modal and requires information from field owners.   * Home Score: * Team Score: * Comments: optional | | 3 | Actor inputs required information.  [Alternative 1] | System validates information.  [Exception 2] | | 4 | Actor send commands to submit.  [Alternative 1] | System displays successful message. |   **Alternative Scenario:**  *[Alternative 1]*   |  |  |  | | --- | --- | --- | | Step. | Actor Action | System Response | | 1 | Actor clicks Cancel on modal | System cancels operation and close modal. |   **Exceptions:**   |  |  |  | | --- | --- | --- | | No. | Cause | System Response | | 1 |  |  | | 2 |  |  |   **Relationships:**   * N/A   **Business Rules:**   * N/A | | | |
|  | | | | |

##### <Field Owner>Rate teams (FO05)



*Figure x+1: <Field Owner> Record match score (FO05)*

|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE – UC\_FO01** | | | |
| **Use Case No.** | FO05 | **Use Case Version** | 0.1 |
| **Use Case Name** | Rate teams | | |
| **Author** | HuanPM | | |
| **Date** | 17/9/2017 | **Priority** | Medium |
| **Actor:**   * Field Owner   **Summary:**   * This use case allows field owner to record match score.   **Goal:**   * Field owner can record match score.   **Triggers:**   * Field owner open web application. * Field owner choose notification about finished match.   **Preconditions:**   * Field owner logged in to the system. * Match has to be finished. * Notification has to be sent to field owner.   **Post Conditions:**   * **Success:** System displays successful message. * **Fail:** System displays error message.   **Main Success Scenario:**   |  |  |  | | --- | --- | --- | | Step | Actor Action | System Response | | 1 | Actor clicks Notifications | System shows notification list.  [Exception 1] | | 2 | Actor clicks a match notification | System displays a modal and requires information from field owners.   * Score: * Comments: optional | | 3 | Actor inputs required information.  [Alternative 1] | System validates information.  [Exception 2] | | 4 | Actor send commands to submit.  [Alternative 1] | System displays successful message. |   **Alternative Scenario:**  *[Alternative 1]*   |  |  |  | | --- | --- | --- | | Step. | Actor Action | System Response | | 1 | Actor clicks Cancel on modal | System cancels operation and close modal. |   **Exceptions:**   |  |  |  | | --- | --- | --- | | No. | Cause | System Response | | 1 |  |  | | 2 |  |  |   **Relationships:**   * N/A   **Business Rules:**   * Field owners gets 15% discount when they do rate teams. | | | |
|  | | | | |

#### 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   **Summary:**   * This use case allows User to edit account profile   **Goal:**   * Account is edited successfully and store in database of the system   **Triggers:**   * User sends command to edit profile   **Preconditions:**   * Actor has accessed the system under user 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 register | 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) * 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] | System shows dictionary view  [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 |   **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 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”. * “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 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”. * “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. | | | |

##### <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:**   * After get search command, the system will get the search value then looking for the right field base on keyword then return the result to user as list field. | | | |

##### <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 chosen field based on time   **Triggers:**   * Actor sends reserve command   **Preconditions:**   * Actor already signed in   **Post Conditions:**   * **Success:** User reserve Field successfully * **Fail:** N/A   **Main Success Scenario:**   |  |  |  | | --- | --- | --- | | Step | Actor Action | System Response | | 1 | Actor chose suitable field from field list | System find time enable to reserve and response to User | | 2 | Actor inputs start time and end time  Actor send Reserve command  [Alternative 1] | System to switch to Paypal payment view | | 3 | User sends command to confirm to continue the payment  [Alternative 2] | System switch to Paypal payment process | | 4 | User completes payment process with Paypal  [Alternative 3]  [Alternative 4]  [Alternative 5] | System switch back and 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 send back command | System back to list of fields window |   **Alternative Scenario 2:**   |  |  |  | | --- | --- | --- | | Step | Actor Action | System Response | | 1 | User sends command to decline | System switch to payment fail view |   **Alternative Scenario 3:**   |  |  |  | | --- | --- | --- | | Step | Actor Action | System Response | | 1 | Paypal responses payment process is fail | System switch to payment fail view |   **Alternative Scenario 4:**   |  |  |  | | --- | --- | --- | | Step | Actor Action | System Response | | 1 | Paypal payment process time out | System switch to payment fail view |   **Alternative Scenario 5:**   |  |  |  | | --- | --- | --- | | Step | Actor Action | System Response | | 1 | User does the payment with not enough money in account | System switch to payment fail view |   **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 Matching request

|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE – UC003** | | | |
| **Use Case No.** | 003 | **Use Case Version** | 1.0 |
| **Use Case Name** | Create matching request | | |
| **Author** |  | | |
| **Date** | 28/10/2017 | **Priority** | Normal |
| **Actor:**   * User   **Summary:**   * This use case allows Actor to create matching request by input time to play   **Goal:**   * Actor can create a Matching request based on time   **Triggers:**   * Actor sends create matching request command   **Preconditions:**   * Actor already signed in   **Post Conditions:**   * **Success:** User reserve Field successfully * **Fail:** N/A   **Main Success Scenario:**   |  |  |  | | --- | --- | --- | | Step | Actor Action | System Response | | 1 | Actor input start time and end time  [Alternative 1] |  | | 2 | Actor send create matching request command | * System take Actor’s current position and time Actor inputted * System will create matching request in database |   **Alternative Scenario 1:**   |  |  |  | | --- | --- | --- | | Step | Actor Action | System Response | | 1 | Actor chose other position by sending choose position command | System load maps and search address field for Actor input a new location  [Exception 1] |   **Exceptions 1:**   |  |  |  | | --- | --- | --- | | Step | Cause | System Response | | 1 | Actor input blank in search address field | System shows warning message “Please input your address”. |   **Relationships:** N/A  **Business Rules:**   * After get create matching request command, system will create matching request in database | | | |

##### <User> Find matching request

|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE – UC003** | | | |
| **Use Case No.** | 003 | **Use Case Version** | 1.0 |
| **Use Case Name** | Find matching request | | |
| **Author** |  | | |
| **Date** | 24/10/2017 | **Priority** | Normal |
| **Actor:**   * User   **Summary:**   * This use case allows Actor to find matching request based on time, location and rank   **Goal:**   * Actor can find suitable matching request base on time, location and rank   **Triggers:**   * Actor sends find matching request command   **Preconditions:**   * Actor already signed in   **Post Conditions:**   * **Success:** list matching requests are shown * **Fail:** N/A   **Main Success Scenario:**   |  |  |  | | --- | --- | --- | | Step | Actor Action | System Response | | 1 | Actor input time to search textbox |  | | 2 | Actor send Search command | System will find in database any record of matching request has time, rank and location suitable for Actor and show records as list of matching requests |   **Alternative Scenario:** N/A  **Exceptions :**  **Exception 1 :**   |  |  |  | | --- | --- | --- | | Step | Cause | System Response | | 1 | Actor did not pick time | System shows warning message “Please pick start time”. |   **Relationships:** N/A  **Business Rules:**   * After get search matching request command, the system will get time, location and rank value then looking for the right matching request then return the result to Actor as a list of matching request. | | | |

##### <User> Accept matching request

|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE – UC003** | | | |
| **Use Case No.** | 003 | **Use Case Version** | 1.0 |
| **Use Case Name** | Accept matching requests | | |
| **Author** |  | | |
| **Date** | 28/10/2017 | **Priority** | Normal |
| **Actor:**   * User   **Summary:**   * This use case allows Actor to accept matching request from other User   **Goal:**   * Actor can accept matching request   **Triggers:**   * Actor sends accept matching request command   **Preconditions:**   * Actor already signed in   **Post Conditions:**   * **Success:** User accept matching request successfully * **Fail:** N/A   **Main Success Scenario:**   |  |  |  | | --- | --- | --- | | Step | Actor Action | System Response | | 1 | Actor input time want to play | System response matching request list based on time and Actor’s rank | | 2 | Actor chose one matching request in matching request list  [Alternative 1] | System to switch to Paypal payment view | | 3 | User sends command to confirm to continue the payment  [Alternative 2] | System switch to Paypal payment process | |  | User completes payment process with Paypal  [Alternative 3]  [Alternative 4]  [Alternative 5] | * System will send notification to User who created matching request * Ask User who created matching request confirm |   **Alternative Scenario 1:**   |  |  |  | | --- | --- | --- | | Step | Actor Action | System Response | | 1 | Actor send back command | System back to previous window and allow Actor input time to play again  Time setting Actor inputted still be saved |   **Alternative Scenario 2:**   |  |  |  | | --- | --- | --- | | Step | Actor Action | System Response | | 1 | User sends command to decline | System switch to payment fail view |   **Alternative Scenario 3:**   |  |  |  | | --- | --- | --- | | Step | Actor Action | System Response | | 1 | Paypal responses payment process is fail | System switch to payment fail view |   **Alternative Scenario 4:**   |  |  |  | | --- | --- | --- | | Step | Actor Action | System Response | | 1 | Paypal payment process time out | System switch to payment fail view |   **Alternative Scenario 5:**   |  |  |  | | --- | --- | --- | | Step | Actor Action | System Response | | 1 | User does the payment with not enough money in account | System switch to payment fail view |   **Exceptions :** N/A  **Relationships:** Extend from find matching request  **Business Rules:**   * After get accept matching request command, system will send notification to User created matching request. | | | |

##### <User> Confirm matching request

|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE – UC003** | | | |
| **Use Case No.** | 003 | **Use Case Version** | 1.0 |
| **Use Case Name** | Confirm matching request | | |
| **Author** |  | | |
| **Date** | 28/10/2017 | **Priority** | Normal |
| **Actor:**   * User   **Summary:**   * This use case allows Actor to confirm matching request   **Goal:**   * Actor can confirm matching request   **Triggers:**   * Actor sends confirm matching request command   **Preconditions:**   * Actor already signed in * Actor created a matching request * Other User accept matching request that Actor created before   **Post Conditions:**   * **Success:** User confirm matching request successfully * **Fail:** N/A   **Main Success Scenario:**   |  |  |  | | --- | --- | --- | | Step | Actor Action | System Response | | 1 | Actor view notification about other User accept Actor’s matching request created | System call payment method via Paypal | | 2 | Actor send confirm command  [Alternative 1] | System to switch to Paypal payment view | | 3 | User sends command to confirm to continue the payment  [Alternative 2] | System switch to Paypal payment process | |  | User completes payment process with Paypal  [Alternative 3]  [Alternative 4]  [Alternative 5] | System will update matching request in database |   **Alternative Scenario 1:**   |  |  |  | | --- | --- | --- | | Step | Actor Action | System Response | | 1 | Actor decline the Accept command | * System delete notification Actor received * Refund for User who accept matching request |   **Alternative Scenario 2:**   |  |  |  | | --- | --- | --- | | Step | Actor Action | System Response | | 1 | User sends command to decline | System switch to payment fail view |   **Alternative Scenario 3:**   |  |  |  | | --- | --- | --- | | Step | Actor Action | System Response | | 1 | Paypal responses payment process is fail | System switch to payment fail view |   **Alternative Scenario 4:**   |  |  |  | | --- | --- | --- | | Step | Actor Action | System Response | | 1 | Paypal payment process time out | System switch to payment fail view |   **Alternative Scenario 5:**   |  |  |  | | --- | --- | --- | | Step | Actor Action | System Response | | 1 | User does the payment with not enough money in account | System switch to payment fail view |   **Exceptions :** N/A  **Relationships:** N/A  **Business Rules:**   * After get confirm matching request command, System will create matching request in database | | | |

##### <User> Ratting Field

|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE – UC012** | | | |
| **Use Case No.** |  | **Use Case Version** | 1.0 |
| **Use Case Name** | Ratting field | | |
| **Author** |  | | |
| **Date** |  | **Priority** | Normal |
| **Actor:**   * User   **Summary:**   * This use case allows Actor rates the Field he/she has just played   **Goal:**   * System save the ratting field to database   **Triggers:**   * User sends create feedback command   **Preconditions:**   * Actor has accessed the system under User role * Ratting the field he/she has just played   **Post Conditions:**   * **Success:** The ratting is created * **Fail:** N/A   **Main Success Scenario:**   |  |  |  | | --- | --- | --- | | Step | Actor Action | System Response | | 1 | User rates quality of the field from 1 to 5 stars  [Alternative 1] | System save the ratting from User, calculate ratting of Field Owner. |   **Alternative Scenario 1:**   |  |  |  | | --- | --- | --- | | Step | Actor Action | System Response | | 1 | User re-ratting quality of the field | System update the ratting from User, calculate ratting for Field Owner. |   **Relationships:**  **Business Rules:** | | | |

##### <User> Ratting Opponent

|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE – UC012** | | | |
| **Use Case No.** |  | **Use Case Version** | 1.0 |
| **Use Case Name** | Ratting opponent | | |
| **Author** |  | | |
| **Date** |  | **Priority** | Normal |
| **Actor:**   * User   **Summary:**   * This use case allows Actor rates the opponent he/she has just played with   **Goal:**   * System save the ratting opponent to database, calculate ratting for Actor’s opponent   **Triggers:**   * User sends create feedback command   **Preconditions:**   * Actor has accessed the system under User role * Ratting Actor’s opponent he/she has just played with   **Post Conditions:**   * **Success:** The ratting is created * **Fail:** N/A   **Main Success Scenario:**   |  |  |  | | --- | --- | --- | | Step | Actor Action | System Response | | 1 | Actor rates opponent from 1 to 3 stars  [Alternative 1] | System save the ratting from User, calculate ratting for opponent. |   **Alternative Scenario 1:**   |  |  |  | | --- | --- | --- | | Step | Actor Action | System Response | | 1 | User re-ratting quality of the field | System update the ratting from User, calculate ratting of Field Owner. |   **Relationships:**  **Business Rules:** | | | |

##### <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 create report User or Field Owner who has bad behavior   **Goal:**   * System save the report to database   **Triggers:**   * User sends report command   **Preconditions:**   * Actor has accessed the system under User 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 blank | System show warning message “Report must be 10 - 500 characters” |   **Relationships:**  **Business Rules:** | | | |

#### Firebase

##### <Firebase Console> Send notification

|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE – UC012** | | | |
| **Use Case No.** |  | **Use Case Version** | 1.0 |
| **Use Case Name** | Send notification | | |
| **Author** |  | | |
| **Date** |  | **Priority** | Normal |
| **Actor:**   * Firebase Console   **Summary:**   * This use case allows system listen to firebase data change and send notification to User or Field Owner   **Goal:**   * Firebase system sends successfully notification   **Triggers:**   * User accept matching request or Field be reserved successfully   **Preconditions:**  **Post Conditions:**   * **Success:** Notification is sent * **Fail:**   **Main Success Scenario:**   |  |  |  | | --- | --- | --- | | Step | Actor Action | System Response | | 1 | Firebase data has changed | System listen Firebase when database on Firebase changed.   * If matching request of User be accepted, notification will be sent to that User * If Field reserved successfully, notification will be sent to that Field Owner   [Exception 1] |   **Exceptions 1:**   |  |  |  | | --- | --- | --- | | Step | Cause | System Response | | 1 | Notification sent fail | System retries send notification after 30 minutes |   **Relationships:**  **Business Rules:** | | | |

#### Guest

##### <Guest> Register

|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE – UC001** | | | |
| **Use Case No.** |  | **Use Case Version** | 1.0 |
| **Use Case Name** | Register | | |
| **Author** | HieuPT | | |
| **Date** | 24/10/2017 | **Priority** | Normal |
| **Actor:**   * Guest   **Summary:**   * This use case allows Guest to register new account   **Goal:**   * Account is registered successfully and store in database of the system   **Triggers:**   * Guest sends command to register   **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 register | 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 register  [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> Login

|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE – UC002** | | | |
| **Use Case No.** | 002 | **Use Case Version** | 1.0 |
| **Use Case Name** | Login | | |
| **Author** |  | | |
| **Date** | 24/10/2017 | **Priority** | Normal |
| **Actor:**   * Guest   **Summary:**   * This use case allows Guest login to the system on website and mobile application   **Goal:**   * Guest login successfully with the proper role   **Triggers:**   * Guest send the login 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 login | System requires identity information form Guest:   * Username: free text input * Password: free text input | | 2 | Guest inputs information |  | | 3 | Guest sends command to login to system  [Alternative 1] | Guests will login 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 login 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 : Login specification