Test specification

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Test Ref | Req being tested | Test Content | Input | Output | Pass Criteria |
|  | FR1 | Check that a user can log in with a valid account | Valid username and password | The log in should be successful | The user can log in |
|  | FR1 | Check that a user cannot log in using an invalid account | Invalid username and password | The log in should be unsuccessful | The user is unable to log in |
|  | FR1 | Check that a user can register an account via a browser | Acceptable username and password | The account should be registered successfully | The users account has been created |
|  | FR1 | Check that a user can only register with an appropriate username and password | A password that is either too weak or contains invalid characters e.g.” ' “ | The account should not be created and an error message should appear | The account is not created |
|  | FR1 | Check that both username and passwords fields must be filled | One or both Fields are left blank | The login should be unsuccessful and the user should be prompted to fill all fields | The login is unsuccessful and the user is prompted |
|  | FR2 + FR9 | Check that a friend request can be sent and received | A friend request will be sent to another user | The request should appear on the other user's account | The request appears correctly |
|  | FR2 + FR9 | Check that a friend request can be accepted | Select the accept option on the received friend request | The users should appear on each others friends list and a confirmation message sent | The users appear on each others friends list and receive a confirmation message |
|  | FR2 + FR9 | Check that a friend request can be declined | Select the decline option | A message should appear saying the request has been declined | The users do not appear on each others friends lists and a declination message is sent |
|  | FR3 | Check that monster list is displayed correctly | The monster page will be displayed | All monsters owned by that user should be displayed | The monsters owned by that user should be displayed |
|  | FR3 | Check that monster attributes are displayed correctly | The monster page will be displayed | The attributes of each monster on the list should be displayed correctly | The monster attributes are displayed correctly |
|  | FR3 | Check that a new user is allocated a monster and an amount of virtual currency | Create a new user account | The user should be allocated a monster and a sum of virtual currency | The user is allocated a monster and the correct amount of virtual currency |
|  | FR3 | Check that the monster allocation is random | Create multiple user accounts | The basic monster allocation should not be the same for each user | The monster allocation is random |
|  | FR4 | Check that the system allocates a prize for a challenge appropriate to monster | Create a new monster and ensure that the prize is low relative to more powerful monsters | The prize for winning a challenge should scale up relative to monster level | The prize must scale relative to monster level |
|  | FR4 | Check the system puts the two correct monsters together for a battle based on user input | The user selects one of their own monsters and one of their opponents monsters to battle | The two selected monsters are put together to battle | The two correct monsters are selected to battle |
|  | FR4 | Check that the user can accept a battle | Once challenged to a battle the user selects accept | The battle should go ahead | The battle goes ahead |
|  | FR4 | Check that the user can decline a battle | Once challenged to a battle the user selects decline | The battle should be cancelled | The battle is cancelled and does not go ahead |
|  | FR5 | Check that users from different servers can add each other as friends | A user will attempt to add a user from a different server to their friends list | The request should appear correctly on the second users requests page | The request displays correctly |
|  | FR5 | Check that users from different servers can accept requests | A user will attempt to add a user from a different server to their friends list | After accept is selected the users will appear on each others friends lists | The users appear on each others friends lists |
|  | FR5 | Check that users from different servers can decline requests | A user will attempt to add a user from a different server to their friends list | The request should be declined | The request is declined and the users do not appear on each others friends lists |
|  | FR5 | Check that users from different servers can buy monsters | One user will attempt to buy a monster from the other server | The transaction should succeed and one user should gain the monster and one gain the value in virtual currency | The transaction succeeds and the monster and the currency are exchanged |
|  | FR5 | Check that users from different servers can arrange a monster breeding | One user will send a breeding request to a user on a different server | The breeding should go ahead | The breeding goes ahead |
|  | FR5 | Check that users from different servers can arrange a fight | One user will send a fight request to a user on a different server | The fight should happen correctly (winner chosen and prizes allocated correctly) | The winner and prize money are allocated correctly |
|  | FR6 | Check that the user can buy a monster | User selects the buy option on a monster for sale | The user should gain the monster and lost the currency value of the monster | The user gains the monster and loses the value of the monster |
|  | FR6 | Check that the user can sell a monster | The user puts a monster up for sale | The monster is correctly displayed as for sale and once the sale takes place the monster is gone and the user has gained the value of the monster | The monster is put up for sale and sold successfully |
|  | FR6 | Check that the user can remove a friend | On the friends list select the remove friend option from a friend | The friend in question should be removed from the friends list and the user removed from theirs | The friend is removed successfully |
|  | FR6 | Check that the user can offer a monster for breeding | The user will select the offer for breeding option on a monster | The monster will display as up for breeding correctly | The monster displays as up for breeding |
|  | FR6 | Check that the user can purchase breeding with a monster | The user will select the breed option on a monster available to breed | The breeding should go ahead and the cost deducted from the users bank | The breed goes ahead properly and the user is appropriately charged |
|  | FR6 | Check that the user can unregister | The user will select the unregister option | The users account should be unregistered and no longer be valid | The users account is no longer valid |
|  | FR7 | Check that on startup the options to login and register are displayed | The user will start up the program in their browser | The options to log in or register should be displayed | The options to log in and register are displayed |
|  | FR7 | Check that once logged in the user can log out | Once logged in the user will select the log out option | The user should be logged out of the system | The user is logged out of the system |
|  | FR8 | Check that once logged in the user has a list of their monsters statuses | The user will log in and go to their monsters page | The user should see a list of their monsters (with health, diseases etc) | The user is presented with a complete list of their monsters plus their statuses |
|  | FR8 | Check that once logged in the user has a list of their friends | The user will log in and go to their friends page | The user should see a list of their friends | The user sees a correct list of their friends |
|  | FR8 | Check that once logged in the user has a list of monsters available for breeding | The user will log in and go to their friends page | The user should see a list of monsters that are available for breeding | The user sees a correct list of monsters available for breeding |
|  | FR8 | Check that once logged in the user has a list of monsters for sale | The user will log in and go to their friends page | The user should see a list monsters that are available to be bought | The user sees a correct list of monsters available for sale |
|  | FR8 | Check that once logged in the user can see a list of all their requests (breeding, challenges etc) with prize money if applicable | The user will log in and go to their friends page | The user should see a list of their requests plus prizes | The user sees a full list of their requests plus the prizes available |
|  | FR10 | Check that after a fight the users competitors monsters list is updated | Once logged in the user will take part in a battle | The monster list of all competators should be updated | The competitors monster list is updated (losers removed from the list) |
|  | FR10 | Check that after a battle has been won the prize money is transferred to the users account | Once logged in the user will take part in a battle and win | The prize should then be in the users account and their monsters updated to account for injuries | The prize money is in the users account and their monsters updated for injuries |
|  | FR11 | Check that the user can see the wealth of all friends | Once logged in the user navigates to the friends page | All friends wealth should be displayed in order of wealth | Friends are displayed in correct order of wealth |