# TCU Computer Science Dept.

## ReadySet Go

## Use-Case 01: Register an Account

**Version 1.1**

# Revision History

| Date | Version | Description | Author |
|---|---|---|---|
| 10/02/20 | 1.0 | Initial Draft | Christian Arciniega |
| 4/4/21 | 1.1 | Added research request paths | Ryan Moncrief |
| | | | |
| | | | |

# UC01 Register an Account

**Brief Description:**

Each user on the ReadySet Go website will need to register an account in order to access the website

**Primary Actor**: Player/Researcher

**Level**: User goal

**Stakeholders and Interests**: Players and Researchers

**Preconditions**: None

**Postconditions**: User will need to confirm email in order to make initial login.

**Trigger**: User selects “Register” on the login page

**Main Success Scenario**:

1. User enters their desired Username
2. System verifies that Username entered by user meets Username specifications.
3. System requests registered Usernames from database
4. Database returns registered Usernames
5. System verifies that Username entered by user doesn’t already exist in Database
6. User enters their full name
7. User enters their Email address
8. System verifies Email address entered by user is valid
9. User enters desired Password
10. System verifies password entered by user meets password specifications
11. User reenters desired Password
12. System verifies password reentered by user matches the password entered in the previous field
13. User verifies reCAPTCHA
14. User clicks “Register”
15. System requests registered Email addresses from database
16. Database returns registered email addresses
17. System verifies that Email address entered by user doesn’t exist in database
18. System stores information entered by the user into database
19. System sends confirmation email to email address entered by the user
20. System notifies user that registration was successful and that they need to confirm the email to proceed
21. User verifies email via confirmation link sent to registered email address
22. System allows login to verified user

**Extensions**:

2a. Username entered by user is invalid

2a1. System notifies user that username is invalid

2a2. User enters another Username

5a. Username entered by user already exists in database

5a1. System notifies user that username already exists

5a2. User enters another Username

8a. Email address entered by user in invalid

8a1. System notifies user that Email address in invalid

8a2. User enters another Email address

10a. password entered by user is invalid

10a1. System notifies user that password is invalid

10a2. User enters new password

12a. password reentered by user doesn't match password entered in previous field

12a1. System notifies user that passwords don't match

12a2. User reenters password that matches password entered in previous field

13a. ReCAPTCHA isn't verified

13a1. System notifies user that reCAPTCHA wasn't verified

13a2. User verifies reCAPTCHA

13b. User checks box to apply for researcher role

13b1. System notifies admin upon account creation of requesting user

13b2. Admin accepts researcher request

13b2a. Admin denies researcher request

13b2b. System sets requesting user's role to player

13b2c. System notifies user of decision via email

13b3. System sets requesting user's role to researcher

13b4. System notifies user of decision via email.

17a. Email address entered by User already exists in database

17a1. System notifies user that the Email address entered is already registered to the ReadySet Go website

17a2. User enters a different Email address

**Priority**: High

**Secondary Actors**: Admin User

**Special Requirements:**

**All user information must be stored securely.**

**Open Issues**:

# ReadySet Go
# Use-Case 02: Login

**Version 1.1**

# Revision History

| Date | Version | Description | Author |
|---|---|---|---|
| 10/02/20 | 1.0 | Initial Draft | Christian Arciniega |
| 4/10/21 | 1.1 | Added Forgot Password and Password Reset | Ryan Moncrief |
| | | | |
| | | | |

# UC02 Login

**Brief Description:**

A user logs in to the ReadySet Go website

**Primary Actor**: Player/Researcher

**Level**: Subfunction Level

**Stakeholders and Interests**: Players and Researchers

**Preconditions**: User must have an account already registered and confirmed to the ReadySet Go website

**Postconditions**: User can access all ReadySet Go features in their specific role(player or researcher.)

**Trigger**: User selects “Login/Register” on the ReadySet Go website

**Main Success Scenario**:

1. User enters Email address associated with their account
2. User enters password associated with their account
3. User selects “Login”
4. System requests accounts from database
5. Database returns registered accounts
6. System verifies if email address is registered and if password entered by user matches that account
7. System redirects user to the landing page

**Extensions**:

1a. Email address or password entered by user is incorrect

1a1. System notifies user that there is an error in Email address or password

1a2. User reenters Email and Password

2a. User forgets password associated with their account

2a1. User selects “Forgot Password?”

2a2. System prompts user to enter the email address associated with their account.

2a3. User enters valid email address.

2a4. System sends confirmation email to the entered email address.

2a5. User clicks link sent to their email address.

2a6. User enters a new password.

2a7. User confirms new password.

2a8. User selects “Submit”

2a9. System updates database with new password.

**Priority**: High

**Secondary Actors**: None

**Special Requirements:** Verification of account is secure and doesn’t expose sensitive information.

**Open Issues**:

# ReadySet Go
# Use-Case 3: Player Views User Statistics

**Version 1.0**

# Revision History

| Date | Version | Description | Author |
|---|---|---|---|
| 10/1/2020 | 1.0 | Initial Draft | Ryan Moncrief |
| 12/6/2020 | 1.1 | Revision based on instructor feedback | Ryan Moncrief |
| | | | |
| | | | |

# Use-Case 3: Player Views User Statistics

**Brief Description:**

Each user will have an account on the website. There will be various statistics attached to their account such as the number of games played, win rate, etc. The user will be able to view these statistics from the website.

**Primary Actor**: Player

**Level**: User goals

**Stakeholders and Interests**: Players, Researchers

**Preconditions**: The user has an account on the website.

The user is logged in

**Postconditions**: System retrieves and displays user statistics

**Trigger**: The user has clicked on the link that will take them to their profile page

**Main Success Scenario**:

1. Player indicates they wish to see their user statistics
2. System retrieves the user statistics stored in the database
3. System displays the user statistics on the profile page

**Extensions**:

None

**Priority**: Medium

**Secondary Actors**: Database System

**Special Requirements:**

**Open Issues**:

# ReadySet Go
# Use-Case 4: Player Views Saved Game File

**Version 1.2**

# Revision History

| Date | Version | Description | Author |
|---|---|---|---|
| 10/1/2020 | 1.0 | Initial Draft | Ryan Moncrief |
| 12/8/2020 | 1.1 | Revised Success Scenario | Ryan Moncrief |
| 4/4/2021 | 1.2 | Added recently saved games extensions | Ryan Moncrief |
| | | | |

# Use-Case 4: Player Views Saved Game File

**Brief Description:**

After a player has finished playing a game with an AI, the user can choose to save the game file to their account to view later. In addition, researchers can view the games that elapsed during AI training. The user will select which game they wish to replay, and then be taken to a page where they may watch the game be replayed in its entirety.

**Primary Actor**: Player

**Level**: User Goal

**Stakeholders and Interests**: Player, Researcher

**Preconditions**: User has an account on the website

The user is logged into their account

**Postconditions**: The system retrieves and plays back a game file

**Trigger**: Player wishes to watch a game file

**Main Success Scenario**:

1. Player indicates they want to view their saved Go games
2. System retrieves list of saved game files from database
3. System retrieves the 10 most recently played games
4. Player selects the game file they would like to view
5. System retrieves specific game file from database
6. System opens window for game file playback, and plays the game file
7. Player closes the playback window

**Extensions**:

3a. Player selects to permanently save a recently played game

3a. System copies game file into user's permanent saved games in database

4a. The player has no saved games

4a1. There are no games to view, exit use case.

**Priority**: Medium

**Secondary Actors**: Database system

**Special Requirements:**

**Open Issues**:

# ReadySet Go
# Use-Case 5: Player Plays Go Game Vs. AI Agent

**Version 1.0**

# Revision History

| Date | Version | Description | Author |
|---|---|---|---|
| 10/1/2020 | 1.0 | Initial Draft | Ryan Moncrief |
| 4/4/2021 | 1.1 | Added automatic save of recent game | Ryan Moncrief |
| | | | |
| | | | |

# Use-Case 5: Player Plays Go Game Vs. AI Agent

**Brief Description:**

This is one of the primary functions of the website. Users will be able to select an AI agent to play against, then enter into an interface to play a game of Go against the selected AI agent. There will also be a designated "helper agent" AI that can suggest a move to the player if the player indicates they want a hint. After the game has concluded, the player will have the option to save the game file to their account. The player will then be able to exit the interface and return to the menu screen.

**Primary Actor**: Player

**Level**: User goal

**Stakeholders and Interests**: Player, Researcher

**Preconditions**: Player has an account on the website

Player is logged into their account

**Postconditions**: Player has finished a game against an AI

Player statistics are updated

Game file is saved to player account

**Trigger**: Player wishes to play a Go game vs. AI

**Main Success Scenario**:

1. Player indicates they wish to play a Go game versus an AI agent.
2. Player selects the parameters of the game versus an AI agent.
3. Player indicates they wish to start the game.
4. System moves to gameplay interface.
5. Player makes a move in the Go game.
6. Opponent AI agent analyzes the player's move.
7. Opponent AI agent makes a move in the Go game.
    a. Return to step 6 until game concludes.
8. System displays results of game.
9. Player indicates they wish to exit the game.
10. System updates player statistics on profile page.
11. System temporarily saves game file into player's account
12. System moves to game parameter setup page.

**Extensions**:

6a. Player indicates that they want a hint.

6a1. Helper AI agent suggests move to player

6a2. Continue to step 6 of primary scenario

6b. Player indicates they want to exit the game prematurely.

6b1. System closes game interface

6b2. System moves back to game parameter setup page

6b3. Exit use case.

9a. Player indicates they wish to permanently save the game file for the game they just played.

9a1. System saves game file to player's profile

9a2. System notifies player of the result of the save attempt

9a3. Continue to step 9 of primary scenario

**Priority**: High

**Secondary Actors**: Database system, Various Go AI Agents

**Special Requirements:**

**Open Issues**:

# ReadySet Go
# Use-Case 6: Publish Trained Agent to be Playable

**Version <1.0>**

# Revision History

| Date | Version | Description | Author |
|---|---|---|---|
| 10/02/2020 | 1.0 | Initial Draft | Kien Nguyen |
| | | | |
| | | | |
| | | | |

# Use-Case 6: Publish Trained Agent to be Playable

**Brief Description:**

After the Researcher is done training an agent on the server, they will have an option to publish that agent so that other users in the server could play against

**Primary Actor**: User with role of Researcher

**Level**: User Goal

**Stakeholders and Interests**: Player, Researcher

**Preconditions**:

1. User has an account on the website under the role of Researcher.
2. User has logged in as a Researcher.
3. An agent residing on the server that is not visible to the Player has been trained by the user.

**Postconditions**:

1. The specified agent for publication is now visible for the Player and the Researcher to play against.

**Trigger**:

1. The Researcher indicates that they want to publish an agent trained by them so that other users could play against it.

**Main Success Scenario**:

1. The Researcher indicates that they want to publish an agent trained by them so that other users could play against it.
2. The system checks if the specified agent is on the server.
3. The system asks the Researcher to confirm their action.
4. The Researcher confirms their action.
5. The specified agent is now visible for other users to play against.
6. The agent state changes from hidden to public on the database.

**Extensions**:

4a. The Researcher declines the confirmation.

1. The Researcher declines the publication of the specified agent.
2. The agent state remains the same on the database.

**Priority**: Medium

**Secondary Actors**: The database

**Special Requirements:**

**Open Issues**:

# ReadySet Go
# Use-Case 7: Edit and Delete User Profile

**Version <1.0>**

# Revision History

| Date | Version | Description | Author |
|---|---|---|---|
| 2/10/2020 | 1.0 | Initial Draft | Derek Isensee |
| 12/8/2020 | 1.1 | Revised Title and Success Scenario | Ryan Moncrief |
| | | | |
| | | | |

# Use-Case 7: Edit and Delete User Profile

**Brief Description:**

A user can edit various identifying factors on their profile, such as their username and password. Also, a user can delete their profile from the website database entirely.

**Primary Actor**: Player/Researcher

**Level**: User goal

**Stakeholders and Interests**: Players and researchers

**Preconditions**: User has a pre-existing account

**Postconditions**: Profile is edited or deleted

**Trigger**: User wishes to edit or delete their account

**Main Success Scenario**:

1. User wishes to edit their profile
2. User edits their profile information
3. System confirms changes are valid
4. System updates the database with new profile information and displays to user
5. User wishes to delete their profile
6. System confirms with user that they want to delete the profile
7. User confirms again that they wish to delete their profile
8. System deletes all user data from the database

**Extensions**:

2a. User decides to discard changes

2b System does not update the database, and the use case is exited

3a. Changes are not valid

3b System notifies user that changes are not valid

3c Return to step 2 of success scenario

**Priority**: Low

**Secondary Actors**: Database system

**Special Requirements:**

Edits must be easy to do and be viewable immediately after being confirmed.

Deleted accounts must have their respective users' information deleted appropriately.

**Open Issues**:

# ReadySet Go
# Use-Case 8: View Training Metrics

**Version <1.1>**

# Revision History

| Date | Version | Description | Author |
|---|---|---|---|
| 10/03/2020 | 1.0 | Initial Draft | Kien Nguyen |
| 4/4/2021 | 1.1 | Added deletion and live training metrics | Ryan Moncrief |
| | | | |
| | | | |

# Use-Case 8: View Training Metrics

**Brief Description:**

A researcher can view the progress of a newly trained AI agent in real time. The system will display all parameters used in training, as well as the current progress of training, and an ELO graph.

**Primary Actor**: User with role of Researcher

**Level**: User Goal

**Stakeholders and Interests**: Researcher

**Preconditions**:

1. User has an account on the website under the role of Researcher.
2. User has logged in as a Researcher.
3. An agent residing on the server that is previously trained by the Researcher.

**Postconditions**:

1. Necessary metrics or results are visible to the Researcher upon the specified agent.

**Trigger**:

1. The Researcher indicates they want to view the metrics or results of a specified agent that is previously trained by them.

**Main Success Scenario**:

1. The Researcher indicates that they want to view the metrics or results of a specified agent that is previously trained by them.
2. The system verifies the specified agent is on the server.
3. System retrieves agent parameters and current training metrics
4. System displays parameters, live training metrics and ELO graph

**Extensions**:

4a. The Researcher chooses to delete the agent

4a1. System verifies that the user wishes to delete agent

4a2. User confirms the deletion

4a3. System deletes AI agent data

**Priority**: Medium

**Secondary Actors**: The database, the server file system.

**Special Requirements:**

**Open Issues**:

# ReadySetGo
# Use-Case 9: User deletes a saved game

**Version 1.0**

# Revision History

| Date | Version | Description | Author |
|---|---|---|---|
| 10/5/2020 | 1.0 | Initial version | Ryan Clements |
| | | | |
| | | | |
| | | | |

# Use-Case 9: User deletes a saved game

**Brief Description:**

The user has access to a list of recent games. There is also a list of saved games the user can modify. Deleting a game removes the game from the list and memory.

**Primary Actor**: Player

**Level**: User Goal

**Stakeholders and Interests**: User

**Preconditions**:

1. The user has a saved game

**Postconditions**:

1. The game is removed from the user's saved game list and backend memory

**Trigger**:

1. The user indicates they want the saved game to be deleted.

**Main Success Scenario**:

1. User views of list of their saved games.
2. User indicates that they want one of the saved games to be deleted.
3. The system removes the saved game from the list and deletes it from memory.

**Extensions**:

None

**Priority**: Low

**Secondary Actors**: Database, File system

**Special Requirements:**

The game must be deleted.

**Open Issues**:

# ReadySetGo
# Use-Case 10: Researcher trains an AI Agent

**Version 1.1**

# Revision History

| Date | Version | Description | Author |
|---|---|---|---|
| 10/5/2020 | 1.0 | Initial version | Ryan Clements |
| 4/4/2021 | 1.1 | Modified training description to train new agents | Ryan Moncrief |
| | | | |
| | | | |

# Use-Case 10: Researcher trains an AI Agent

**Brief Description:**

A user with the role of researcher can train new AI agents. Through a GUI any researcher can set the parameters and begin a training session with any AI agent.

**Primary Actor**: Researcher

**Level**: User Goal

**Stakeholders and Interests**: User

**Preconditions**:

1. The user has the role researcher

**Postconditions**:

1. A new AI agent is created

**Trigger**:

1. The confirms they want to train an AI with the set parameters

**Main Success Scenario**:

1. Researcher enters parameters for AI training session
2. Researcher verifies and confirms training session
3. System verifies parameters
5. System conducts training
6. System saves AI agent data to researcher profile

**Extensions: N/A**

**Priority**: High

**Secondary Actors**: Database, File system, AI agents

**Special Requirements:**

Initiating a training session must be easy and straightforward.

**Open Issues**:

# ReadySetGo
# Use-Case 11: Admin Deletes User Data

**Version 1.0**

# Revision History

| Date | Version | Description | Author |
|---|---|---|---|
| 4/4/2021 | 1.0 | Initial version | Ryan Moncrief |
| | | | |
| | | | |
| | | | |

# Use-Case 11: Admin Deletes User Data

**Brief Description:**

Admin users will be able to view and delete any users on the site

**Primary Actor**: Admin

**Level**: Admin Goal

**Stakeholders and Interests**: User, Admin

**Preconditions**:

2. The user has the admin role

**Postconditions**:

2. A user on the website is deleted

**Trigger**:

2. The admin wishes to view and delete users

**Main Success Scenario**:

1. Admin wishes to view all users on the site
2. System displays all users
3. Admin indicates that they wish to delete a user
4. System verifies the deletion
5. Admin confirms the deletion
6. System deletes all user data

**Extensions:**

5a. Admin denies the confirmation

5a1. User data is not deleted. Exit use case.

**Priority**: High

**Secondary Actors**: Database, File system,

**Special Requirements:**

**Open Issues**:

# ReadySetGo
# Use-Case 11: Admin Accepts Researcher Requests

**Version 1.0**

# Revision History

| Date | Version | Description | Author |
|---|---|---|---|
| 4/4/2021 | 1.0 | Initial version | Ryan Moncrief |
| | | | |
| | | | |
| | | | |

# Use-Case 11: Admin Accepts Researcher Requests

**Brief Description:**

When a user creates an account on the website, they have the option to request to become a researcher, which will allow them to train AI agents, as well as play against them. Admin users must review and accept or deny these requests before any role change is made.

**Primary Actor**: Admin

**Level**: Admin Goal

**Stakeholders and Interests**: User, Researcher

**Preconditions**:

1. The user has the admin role
2. At least one user has requested to become a researcher

**Postconditions**:

1. A user on the website is promoted to the researcher role

**Trigger**:

The admin wishes to review researcher requests

**Main Success Scenario**:

1. System displays all researcher requests from new users.
2. Admin approves user request to become a researcher
3. System changes the new user's role to researcher
4. System notifies the new user of the decision via email

**Extensions:**

2a. Admin denies the request to become a researcher

2a1. System sets the new user's role to player

2a2. System notifies the new user of the decision via email.

2a3. Exit use case.

**Priority**: High

**Secondary Actors**: Database, File system,

**Special Requirements:**

**Open Issues**: