

## **P5 CS414 Team Project**

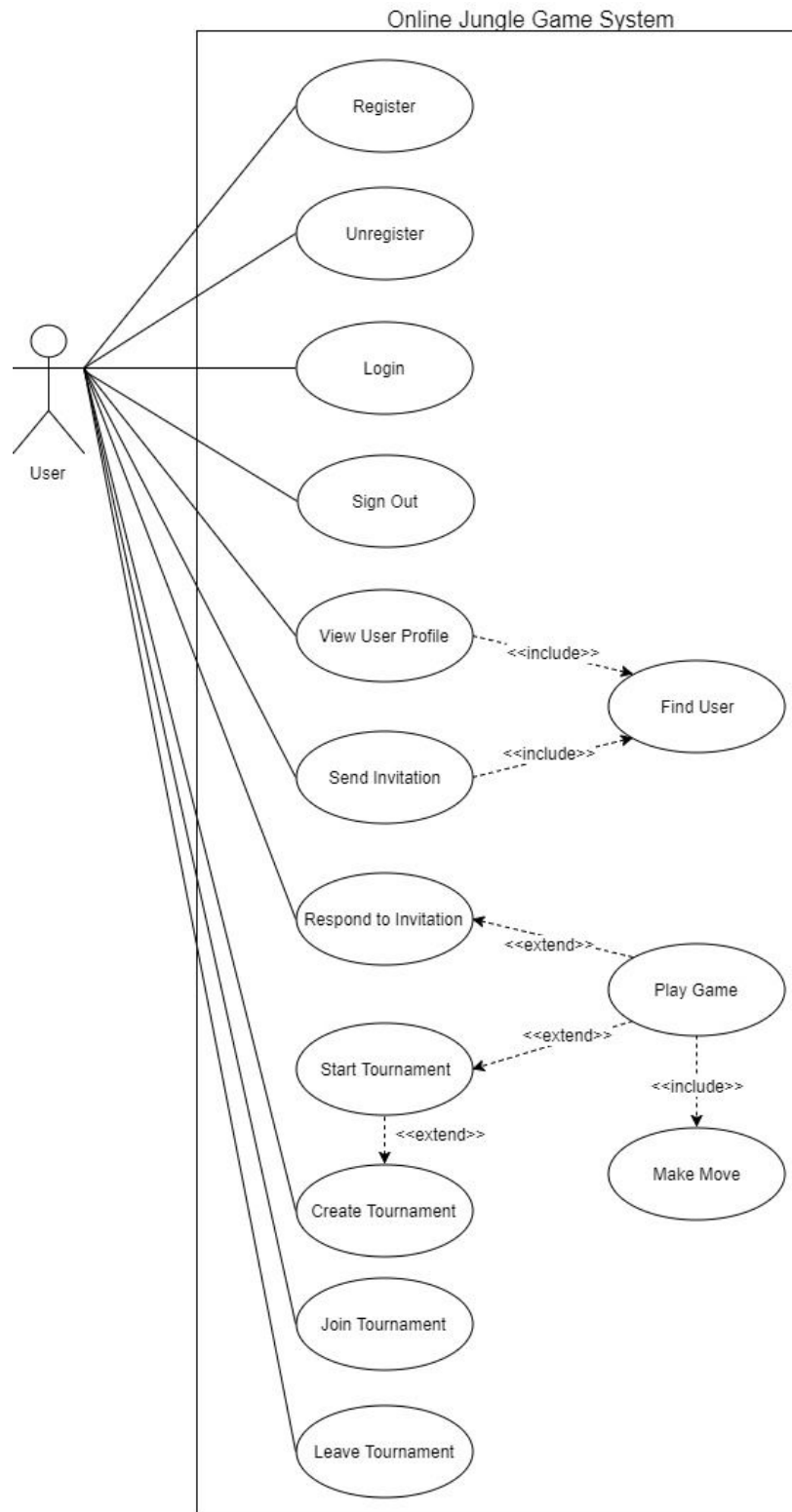
Team: And Yet It Compiles

Members: Laura South, Isacc Mauro, Duck Keun Yang, Brian Larson, Yan Wang

### **Core Requirements**

1. Any person can register to the system. The registration requires an email (which is unique), a password, and a nickname (which is also unique).
2. A registered user can create a new game. The registered user becomes a player of the created game.
3. A registered user can invite another registered user (or set of registered users) to join a created game.
4. A registered user can accept or reject an invitation to join a game. If the user accepts the invitation, she becomes a player of the game.
5. A registered user can be part of different games at the same time.
6. A registered user only has access to the games she is a player of.
7. A player can quit a game at any time.
8. A registered user can unregister from the system.
9. The system must record the history of games played by a user. The record of a game includes the opponent, start date and time, end date and time, and the end result of the game (i.e., win, loss, tie, draw, abandoned, etc.)
10. A registered user has a profile, which consists of her nickname and history of played games. User profiles are only visible to other registered users.
11. A game cannot start until the minimum number of players required for the game have joined.
12. Once a game starts, new players cannot join.
13. The systems must determine which player starts the game according to the rules of the X game. If there are no specific rules, the user who has created the game is the one making the first move.
14. The system determines whose turn it is according to the game rules.
15. A player can only make moves in her active games.
16. A player can only make moves if it is her turn to play.
17. Players can only make allowed moves. Allowed moves are given by the game rules.
18. The system saves the state of active games. Players can play asynchronously but following the turn rules.
19. The system must determine when a game is over. The system must also determine who is the winner and the loser of each game, when there is a tie, or when there is a draw according the game rules.

## Use Case Diagram



## Use Case Descriptions

ID:	1
Use Case Name:	Register
Overview:	Enters user information into the system and allows them to access other features in the system..
Scope:	Online Jungle Game Software
Level:	User Goal
Actors:	Actor: User
Stakeholders and Interests:	User: Wants to register his account, and play the jungle game System Administrator: Wants to store the new user's information in the Database System without an error. Wants to ensure that there is no duplicate users in the database.
Preconditions:	None
Success Guarantee:	<ol style="list-style-type: none"> <li>1. User's information has been saved into the Database System.</li> <li>2. User is registered in the system</li> </ol>
Main Success Scenario:	<ol style="list-style-type: none"> <li>1. System presents forms that are needed for registering a new user into the database.</li> <li>2. User fills the forms, and submits it.</li> <li>3. System saves User's information to the database.</li> <li>4. User's information include Email and password.</li> <li>5. System validates the user input and stores it</li> <li>6. User is redirected to the login screen.</li> </ol>
Extensions:	1a. If the email or the nickname are not unique in the database, or the password is invalid: <ol style="list-style-type: none"> <li>1. System notifies user and do not save User's information to the database.</li> </ol>
Special Requirements:	None
Technology and Data Variations List:	1a. User information entered by keyboard. 1b. The password must be at least 8 characters. 1c. The password must include at least one special character. 1d. The username must contain letters and numbers
Frequency of	Could happen once when User executes the Online Jungle Game

Occurrence:	Software.
Miscellaneous:	

ID:	2
Use Case Name:	Unregister
Overview:	Removes an existing user from the system.
Scope:	Online Jungle Game Software
Level:	User Goal
Actors:	Actor: User
Stakeholders and Interests:	User: Wants to remove his account. System Administrator: Wants to ensure that the removed users personal information does not exist in the database
Preconditions:	<ol style="list-style-type: none"> <li>1. User is registered in the system.</li> <li>2. User is logged on to the system.</li> </ol>
Success Guarantee:	<ol style="list-style-type: none"> <li>1. User's personal information is removed from the database</li> <li>2. The user can no longer log in with that account</li> </ol>
Main Success Scenario:	<ol style="list-style-type: none"> <li>1. The user selects the unregister operation</li> <li>2. The user is prompted if they are sure they want to unregister</li> <li>3. User selects Yes.</li> <li>4. The database removes the user's email and password from the database</li> <li>5. User is logged out from the system.</li> </ol>
Extensions:	1a. If User selects No: <ol style="list-style-type: none"> <li>1. User's information remains in the database</li> </ol>
Special Requirements:	<ol style="list-style-type: none"> <li>1. User's personal information should be removed (email, password), but their nickname will remain</li> </ol>
Technology and Data Variations List:	1a. User's decision is made by keyboard.
Frequency of Occurrence:	Could happen multiple times after User is registered and exists in the database
Miscellaneous:	None

ID:	3
Use Case Name:	Login
Scope:	Online Jungle Game Software
Level:	User Goal
Actors:	Actor: User
Stakeholders and Interests:	User: Wants to log on to the system to use system functionality, such as playing a game. System Administrator: Wants to make sure the user is authenticated properly
Preconditions:	1. User is registered in the system.
Success Guarantee:	1. User is logged on to the system.
Main Success Scenario:	<ol style="list-style-type: none"> <li>1. System represents forms for the username and the password.</li> <li>2. User enters the username and the password, and submits.</li> <li>3. System attempts to authenticate the user based on the information provided</li> <li>4. User is logged on to the system, and can access the other features in the system.</li> </ol>
Extensions:	<p>3a. If the username does not exist:</p> <ol style="list-style-type: none"> <li>1. System notifies User that the username submitted does not exist.</li> <li>2. Return to the step 1 in Main Success Scenario.</li> </ol> <p>3b. If the username and the password does not match:</p> <ol style="list-style-type: none"> <li>1. System notifies User to make sure the password is correct.</li> <li>2. Return to the step 1 in Main Success Scenario.</li> </ol>
Special Requirements:	1. If a user inputs their password incorrectly 5 times in a row, they are blocked from attempting again for 24 hours
Technology and Data Variations List:	2a. The username and the password entered by keyboard.
Frequency of Occurrence:	Could be nearly continuous.
Miscellaneous:	Open Issues:

ID:	4
Use Case Name:	Find User
Scope:	Online Jungle Game Software
Level:	User Goal
Actors:	Actor: User
Stakeholders and Interests:	User: Wants to find another user to see their profile or invite them to a game System Administrator: Wants to ensure that the user being searched for is a valid user
Preconditions:	1. User is logged on to the system.
Success Guarantee:	1. User is notified by the system that the submitted username is valid and registered in the system.
Main Success Scenario:	1. System represents a form asking the username to the User 2. User enters the username and submits it. 3. The system finds the other user's information, and notifies User that the username is valid or not.
Extensions:	3a. If the username is valid: 1. User can send an invitation, or view user profile. 3b. If the username is not valid: 1. User can submit a different username or cancel Find User
Special Requirements:	1. Answering User's request should be done in 2 seconds.
Technology and Data Variations List:	2a. The username is entered by keyboard.
Frequency of Occurrence:	Could be nearly continuous.
Miscellaneous:	None

ID:	5
Use Case Name:	View User Profile

Overview:	When a user wants to view a user profile, either their own or another user's.
Scope:	Online Jungle Game Software
Level:	User Goal
Actors:	Actor: User
Stakeholders and Interests:	User: Wants to view the user profile System administrator: Wants to display the correct user profile with all relevant information
Preconditions:	<ol style="list-style-type: none"> <li>1. User is logged on to the system.</li> <li>2. The profile that is being checked needs to be for a valid, registered user</li> </ol>
Success Guarantee:	<ol style="list-style-type: none"> <li>1. System displays the selected profile to the user</li> </ol>
Main Success Scenario:	<ol style="list-style-type: none"> <li>1. include(Find User)</li> <li>2. User selects profile menu.</li> <li>3. The system displays the specified user profile stored in the database The user's profile is consists of nickname, and a history of played games. The record of a game includes the opponent, start date and time, end date and time, and the result of the game (iwin, loss, tie, draw, abandoned).</li> </ol>
Extensions:	None
Special Requirements:	<ol style="list-style-type: none"> <li>1. Answering User's request should be done in 2 seconds.</li> </ol>
Technology and Data Variations List:	None
Frequency of Occurrence:	Could be nearly continuous.
Miscellaneous:	<p>Open Issues:</p> <ol style="list-style-type: none"> <li>1. How many game records will be shown in the user profile?</li> <li>2. How long should the system hold the game record after the game has finished?</li> </ol>

ID:	6
Use Case Name:	Send Invitation

Overview:	Sends invitation to another user.
Scope:	Online Jungle Game Software
Level:	User Goal
Actors:	Primary Actor: User Secondary Actor: None
Stakeholders and Interests:	User: Wants to send an invitation to another user so they can play a game together System administrator: Wants to make sure the invitation is to a valid user
Preconditions:	<ol style="list-style-type: none"> <li>1. User is logged on to the system</li> <li>2. The invitation recipient is a valid, registered user</li> </ol>
Success Guarantee:	<ol style="list-style-type: none"> <li>1. An invitation is successfully sent to the recipient</li> </ol>
Main Success Scenario:	<ol style="list-style-type: none"> <li>1. include(Find User)</li> <li>2. User selects invitation option.</li> <li>3. System checks the other user's current status.</li> <li>4. System sends an invitation to the other user.</li> </ol>
Extensions:	2a. The other user is not online: <ol style="list-style-type: none"> <li>1. System notifies User that the other user is offline.</li> </ol>
Special Requirements:	None
Technology and Data Variations List:	None
Frequency of Occurrence:	Could be nearly continuous.
Miscellaneous:	None

ID:	7
Use Case Name:	Respond to Invitation
Overview:	Receives invitation for user to accept or decline.
Scope:	Online Jungle Game Software



Level:	User Goal
Actors:	Primary Actor: User Secondary Actor: None
Stakeholders and Interests:	User: Wants to respond to the invitation, and have a game created if they accept System administrator: Wants
Preconditions:	<ol style="list-style-type: none"> <li>1. User is logged on to the system.</li> <li>2. User has received an invitation from another user.</li> </ol>
Success Guarantee:	<ol style="list-style-type: none"> <li>1. A game is created, and the two users are put into the game</li> </ol>
Main Success Scenario:	<ol style="list-style-type: none"> <li>1. System notifies that User has received invitation, and asks to select either 'Accept' or 'Decline'.</li> <li>2. User chooses whether to accept or decline</li> </ol>
Extensions:	<ol style="list-style-type: none"> <li>2a. User declines the invitation <ol style="list-style-type: none"> <li>1. System notifies the sender that User has declined the invitation.</li> <li>2. User is returned to the main screen.</li> </ol> </li> <li>2b. User accepts the invitation <ol style="list-style-type: none"> <li>1. The system creates a game session</li> <li>2. The system adds the sender and recipient to the game session, and notifies the users that they have been added to the game.</li> <li>3. Include (Play game)</li> </ol> </li> </ol>
Special Requirements:	None
Technology and Data Variations List:	None
Frequency of Occurrence:	One for every invitation sent
Miscellaneous:	None

ID:	8
Use Case Name:	Play Game
Overview:	Main gameplay loop, contains sequences related to playing an active game.

Scope:	Online Jungle Game Software
Level:	User Goal
Actors:	Primary: User(player) Secondary: Database System
Stakeholders and Interests:	User: Wants to interact with the system to play jungle game. Opponent: Same as User. System Administrator: Wants to ensure that the game state is valid throughout the game, according to the game rules
Preconditions:	<ol style="list-style-type: none"> <li>1. User is registered in the system.</li> <li>2. User is logged on to the system.</li> <li>3. User has accepted an invitation, or a user's invitation has been accepted</li> <li>4. The initial game state has been setup by the system.</li> </ol>
Success Guarantee:	<ol style="list-style-type: none"> <li>1. The game has ended.</li> <li>2. User won/lost/left.</li> <li>3. System logs game record to the database.</li> </ol>
Main Success Scenario:	<ol style="list-style-type: none"> <li>1. The system displays the game board to both players</li> <li>2. The system chooses the user who sent the invitation as the player with the first move</li> <li>3. User/Opponent includes(Make move)</li> <li>4. Repeat step 3 until the win condition is met by one of the players.</li> <li>5. When the win condition is met, the game is ended. User/Opponent receive a notification of the game result (win/loss)</li> <li>6. System stores the game record to the database.</li> <li>7. The user is returned to the main screen.</li> </ol>
Extensions:	<p>all. At any time, if User decides to quit game:</p> <ol style="list-style-type: none"> <li>1. The game is ended.</li> <li>2. System stores the game record as an abandon</li> </ol> <p>all. At any time, if User selects different game tab:</p> <ol style="list-style-type: none"> <li>1. The game is suspended, and Opponent waits for User to make move.</li> <li>2. User is redirected to the another game session with different Opponent.</li> </ol> <p>all. At any time, if User's system fails:</p> <ol style="list-style-type: none"> <li>1. The game is ended, and Opponent is returned to the main screen.</li> <li>2. System updates User's game history with an abandon.</li> <li>3. System updates Opponent's game history with a win.</li> </ol> <p>all. At any time, if System fails:</p>

	<ol style="list-style-type: none"> <li>1. The game is ended, and User/Opponent are notified that System failed.</li> <li>2. System does not update the record of the game.</li> <li>3. User/Opponent are returned to the main screen.</li> </ol> <p>3a. If it's User's turn:</p> <ol style="list-style-type: none"> <li>1. System notifies the user that it is their turn</li> <li>2. include(Make Move).</li> </ol> <p>3b. If it's Opponent's turn:</p> <ol style="list-style-type: none"> <li>1. User waits for Opponent to include(Make Move).</li> </ol>
Special Requirements:	None
Technology and Data Variations List:	None
Frequency of Occurrence:	Could be nearly continuous.
Miscellaneous:	<p>Open Issues:</p> <ol style="list-style-type: none"> <li>1. How long System should keep the state of suspended game?</li> </ol>

ID:	9
Use Case Name:	Make Move
Overview:	During an active game, a player makes a valid move in the game, according to the game rules.
Scope:	Online Jungle Game Software
Level:	User Goal
Actors:	<p>Primary Actor: User</p> <p>Secondary Actor: None</p>
Stakeholders and Interests:	<p>User: Wants to make a move in the jungle game.</p> <p>Opponent: Same as User.</p> <p>System Administrator: Wants to ensure that the game state remains valid throughout the game, based on the game rules</p>
Preconditions:	<ol style="list-style-type: none"> <li>1. The User is currently in an active game.</li> <li>2. It is the User's turn.</li> <li>3. The game is in a valid state.</li> </ol>
Success Guarantee:	<ol style="list-style-type: none"> <li>1. The game is in a valid state according to the Jungle game rules.</li> </ol>

	<ol style="list-style-type: none"> <li>2. System updates the current state of the game for the user and the opponent</li> <li>3. It is now the opponent's turn.</li> </ol>
Main Success Scenario:	<ol style="list-style-type: none"> <li>1. User chooses a piece to move.</li> <li>2. User chooses a game tile to move the piece to.</li> <li>3. The game state is updated for both User and Opponent, and Opponent is notified that it is now the opponent's turn.</li> </ol>
Extensions:	<p>2a. If the move is invalid according to the jungle game rules:</p> <ol style="list-style-type: none"> <li>1. User is notified and asked to make a valid move.</li> <li>2. System does not update the game state</li> </ol>
Special Requirements:	None
Technology and Data Variations List:	<ol style="list-style-type: none"> <li>1. A piece is selected by mouse.</li> <li>2. A tile is selected by mouse.</li> </ol>
Frequency of Occurrence:	Could be nearly continuous.
Miscellaneous:	None.

ID:	10
Use Case Name:	Sign out
Overview:	The user signs out of the XGame system
Scope:	Online Jungle Game Software
Level:	User Goal
Actors:	<p>Primary Actor: User</p> <p>Secondary Actor: None</p>
Stakeholders and Interests:	<p>User: Wants to sign out from the system</p> <p>System Administrator: Wants to ensure that the database shows that the user offline</p>
Preconditions:	<ol style="list-style-type: none"> <li>1. The user is currently registered</li> <li>2. The user is currently signed in</li> </ol>
Success Guarantee:	<ol style="list-style-type: none"> <li>1. The user is signed out</li> <li>2. The system returns to the login screen</li> </ol>

Main Success Scenario:	<ol style="list-style-type: none"> <li>1. User selects the sign out function</li> <li>2. The system prompts the user if they are sure they want to log out (yes/no)</li> <li>3. The user selects yes and is logged out of the system</li> <li>4. The system updates the database to indicate that the user is now offline</li> </ol>
Extensions:	2a. If the user selects no: <ol style="list-style-type: none"> <li>1. The user is not logged out</li> <li>2. The system does not update the database</li> </ol>
Special Requirements:	None
Technology and Data Variations List:	
Frequency of Occurrence:	Could be nearly continuous.
Miscellaneous:	None.

ID:	11
Use Case Name:	Create Tournament
Overview:	The user creates a jungle tournament
Scope:	Online Jungle Game Software
Level:	User Goal
Actors:	Primary Actor: User Secondary Actor: None
Stakeholders and Interests:	User: Wants to creates a tournament System Administrator: Wants to ensure that the tournament is created
Preconditions:	<ol style="list-style-type: none"> <li>1. The user is currently registered</li> <li>2. The user is currently signed in</li> </ol>
Success Guarantee:	<ol style="list-style-type: none"> <li>1. The tournament is created</li> <li>2. The system returns to the main screen</li> </ol>
Main Success Scenario:	<ol style="list-style-type: none"> <li>1. User selects the create tournament option</li> <li>2. User specifies tournament name and the maximum number of players</li> </ol>

Extensions:	2a. If the tournament name is already exist: 1. The system notifies the user that the tournament is already exist 2. The tournament is not created
Special Requirements:	None
Technology and Data Variations List:	
Frequency of Occurrence:	Could be nearly continuous.
Miscellaneous:	None.

ID:	12
Use Case Name:	Join Tournament
Overview:	The user joins the tournament
Scope:	Online Jungle Game Software
Level:	User Goal
Actors:	Primary Actor: User Secondary Actor: None
Stakeholders and Interests:	User: Wants to join the tournament System Administrator: Wants to ensure that the user is registered to the tournament
Preconditions:	1. The user is currently registered 2. The user is currently signed in
Success Guarantee:	1. The user is registered to the tournament
Main Success Scenario:	1. The user selects join tournament option 2. The user specifies the tournament name
Extensions:	2a. If the tournament name does not exist: 1. The system notifies the user that the tournament does not exist 2. The user is not registered to the tournament 2b. If the tournament is full: 1. The system notifies the user that the tournament does not exist 2. The user is not registered to the tournament

Special Requirements:	None
Technology and Data Variations List:	
Frequency of Occurrence:	Could be nearly continuous.
Miscellaneous:	None.

ID:	13
Use Case Name:	Leave Tournament
Overview:	The user leaves the tournament
Scope:	Online Jungle Game Software
Level:	User Goal
Actors:	Primary Actor: User Secondary Actor: None
Stakeholders and Interests:	User: Wants to leave the tournament System Administrator: Wants to ensure that the user is deregistered from the tournament
Preconditions:	<ol style="list-style-type: none"> <li>1. The user is currently registered</li> <li>2. The user is currently signed in</li> <li>3. The user is registered to the tournament</li> </ol>
Success Guarantee:	<ol style="list-style-type: none"> <li>1. The user is deregistered from the tournament</li> </ol>
Main Success Scenario:	<ol style="list-style-type: none"> <li>1. The user selects leave tournament option</li> <li>2. The user specifies the tournament name</li> </ol>
Extensions:	<p>2a. If the tournament name does not exist:</p> <ol style="list-style-type: none"> <li>1. The system notifies the user that the tournament does not exist</li> </ol> <p>2b. If the tournament has already started:</p> <ol style="list-style-type: none"> <li>1. The system notifies the user that the tournament has begun</li> <li>2. The user is not deregistered from the tournament</li> </ol>
Special Requirements:	None
Technology and	

Data Variations List:	
Frequency of Occurrence:	Could be nearly continuous.
Miscellaneous:	None.

ID:	14
Use Case Name:	Start Tournament
Overview:	The user signs out of the XGame system
Scope:	Online Jungle Game Software
Level:	User Goal
Actors:	Primary Actor: User Secondary Actor: None
Stakeholders and Interests:	User: Wants to start the tournament System Administrator: Wants to ensure that the tournament is started and all the games are initiated
Preconditions:	<ol style="list-style-type: none"> <li>1. The user is currently registered</li> <li>2. The user is currently signed in</li> <li>3. The tournament has more than 2 players</li> <li>4. The tournament is not started</li> <li>5. The user is the owner of the tournament</li> </ol>
Success Guarantee:	<ol style="list-style-type: none"> <li>1. The tournament is started</li> <li>2. The game is started for all the players in the tournament</li> </ol>
Main Success Scenario:	<ol style="list-style-type: none"> <li>1. The user selects start tournament option</li> <li>2. The user specifies the tournament name</li> </ol>
Extensions:	<p>2a. If the tournament name does not exist:</p> <ol style="list-style-type: none"> <li>1. The system notifies the user that the tournament does not exist</li> </ol> <p>2b. If the tournament has already started:</p> <ol style="list-style-type: none"> <li>1. The system notifies the user that the tournament has begun</li> <li>2. The tournament is not started</li> </ol>
Special Requirements:	None
Technology and Data Variations List:	



Frequency of Occurrence:	Could be nearly continuous.
Miscellaneous:	None.