## Main Interface Agent - Tree Lot Coordinator

|  |  |
| --- | --- |
| Use Case Name: | 1. Register a Scout |
| Description: | |
| Register a Boy Scout in Troop 209 to sell trees (will already be a scout) | |
| Preconditions: | |
| 1. Must be a Scout in Troop 209, therefore has a Troop ID | |
| Workflow: | |
| 1. Scout approaches Tree Lot Coordinator(TLC) and requests to register. 2. TLC requests the Scout provide their last name, first name, middle name, date of birth, phone number, email, and troop ID 3. Scout provides their last name, first name, middle name, date of birth, phone number, email, and troop ID to the TLC 4. TLC verifies that Troop ID provided in step 3 does not already exist as a Troop ID on another file in the Scout Folder 5. TLC validates the information given by the Scout in step 3 6. TLC creates a new Scout record with the validated information from step 5 7. TLC sets Status to “active” and sets DateStatusUpdated to the current date in the Scout record 8. TLC files the Scout record into the Scout Folder 9. TLC notifies the Scout that the Scout is successfully registered | |
| Results: | |
| 1. A new Scout has been registered | |
| Alternates: | |
| 1. Scout’s Troop ID is already in the system 2. Email or Phone number is invalid | |
| Entities Involved: | |
| Tree Lot Coordinator, Scout, Scout Folder, Scout record | |

|  |  |
| --- | --- |
| Use Case Name: | 1. Update a Scout |
| Description: | |
| Updates a Scouts record in the Scout Folder | |
| Preconditions: | |
| 1. Scout must exist in the Scout Folder | |
| Workflow: | |
| 1. Scout approaches TLC and requests to update info for Scout 2. The TLC requests the scout to provide one or more of part of the first name, part of the last name, and email 3. Scout provides one or more of first name, last name, and email to the TLC 4. TLC retrieves a collection of Scout records that match the given info and presents them to the scout to choose the correct one 5. The Scout indicates the desired record from the given Scout collection provided by the TLC in step 4 to the TLC 6. TLC retrieves the desired Scout record selected in step 5 from the Scout collection provided in step 4 7. TLC requests the Scout to provide the information to be changed 8. The Scout provides the information to be changed to the TLC 9. TLC validates the new information provided in step 8 for correctness 10. TLC updates the requested information in the Scout record retrieved in step 6 11. TLC changes DateStatusUpdated to the current date in the Scout record retrieved in step 6 12. TLC refiles the updated Scout record into the Scout Folder 13. TLC notifies the Scout that the desired information is updated | |
| Results: | |
| 1. The Scout’s record in the Scout folder has been updated | |
| Alternates: | |
| 1. The Scout does not exist in the Scout Folder 2. The given information returns an empty collection 3. Email or phone number is invalid | |
| Entities Involved: | |
| Tree Lot Coordinator, Scout, Scout Folder, Scout record | |

|  |  |
| --- | --- |
| Use Case Name: | 1. Delete a Scout |
| Description: | |
| Sets a Scout’s status to ‘Inactive’ | |
| Preconditions: | |
| 1. Scout exists in the Scout Folder | |
| Workflow: | |
| 1. Scout approaches TLC and requests to be removed from the System 2. TLC requests the scout provide one or more of first name, last name, and email 3. The Scout provides one or more of first name, last name, email to the TLC 4. TLC retrieves a collection of Scout records that match the given info from step 3 5. TLC provides records to Scout from the collection created in step 4 and asks them to choose the desired Scout record 6. Scout indicates to the TLC the desired Scout record from the given collection provided in step 5 7. TLC retrieves the desired Scout record from the Scout record collection created in step 4 8. TLC provides the Scout record retrieved in step 7 and asks Scout to verify they are sure they want to be unregistered 9. Scout verifies that they wish to be unregistered to the TLC 10. TLC sets Status on Scout record retrieved in step 7 to ‘Inactive’ 11. TLC updates DateStatusUpdated to the current date in the Scout record retrieved from step 7 12. TLC refiles Scout record to the Scout Folder 13. TLC notifies the Scout that the information is updated | |
| Results: | |
| 1. The Scout’s status in the Scout folder has been updated to ‘Inactive’ | |
| Alternates: | |
| 1. The Scout does not exist in the Scout Folder 2. The given info returns an empty collection 3. Scout decides at step 8 they don’t want to unregister | |
| Entities Involved: | |
| Tree Lot Coordinator, Scout, Scout Folder, Scout record | |

|  |  |
| --- | --- |
| Use Case Name: | 1. Add a Tree |
| Description: | |
| Add a Tree record to the Tree folder | |
| Preconditions: | |
| 1. Tree barcode prefix matches an existing Tree type 2. Tree barcode does not already exist in the Tree Folder | |
| Workflow: | |
| 1. Scout approaches TLC and requests to add a Tree 2. TLC requests the Scout to provide the barcode for the new Tree 3. Scout provides TLC with barcode 4. TLC extracts the prefix from the barcode provided by the Scout 5. TLC retrieves the Tree Type record from the Tree Type folder using the barcode prefix extracted in step 4 6. TLC verifies that the barcode given in step 3 does not already exist as an ID for a Tree record in the Tree Folder 7. TLC creates a new Tree record with the barcode provided by the Scout as the primary key, the Tree Type as the ID of the Tree Type record retrieved in step 5, the Status to ‘Available’, and the DateStatusUpdated to the current date 8. TLC asks the Scout if there are any notes to add to the Tree record created in step 7 9. The Scout provides notes for the Tree record created in step 7 10. TLC adds the notes provided by the Scout in step 9 to the Tree record created in step 7 11. TLC files Tree record created in step 7 to the Tree Folder 12. TLC notifies the Scout that the addition of the Tree to the Tree folder has been successful | |
| Results: | |
| 1. A new Tree record is filed into the Tree Folder | |
| Alternates: | |
| 1. Tree barcode prefix does not match an existing Tree type 2. Tree barcode already exists in the Tree Folder | |
| Entities Involved: | |
| Tree Lot Coordinator, Scout, Tree, Tree Folder, Tree record, Tree Type Folder, Tree Type record | |

|  |  |
| --- | --- |
| Use Case Name: | 1. Update a Tree |
| Description: | |
| Update a tree information in the tree folder. This use case includes features to search for a tree | |
| Preconditions: | |
| 1. Tree must exist in tree folder | |
| Workflow: | |
| 1. Scout approaches TLC and request to update a tree 2. TLC requests the Scout to provide the barcode of the tree on which the update is to be carried out 3. The Scout provides the barcode value to the TLC 4. TLC retrieves the Tree record using the barcode information provided by the Scout in step 3 5. TLC requests the Scout to provide the information to be changed on the Tree record 6. The Scout provides the information to be changed on the Tree record to the TLC 7. TLC validates the new information provided in step 6 8. TLC updates the requested information in the Tree record retrieved in step 4 9. TLC changes DateStatusUpdated to the current date in the Tree record retrieved in step 4 10. TLC refiles the updated Tree record into the Tree Folder 11. TLC notifies the Scout that the desired information is updated in the Tree record | |
| Results: | |
| 1. The tree information is updated in the Tree Folder | |
| Alternates: | |
| 1. Barcode value is invalid 2. Information provided to be updated is invalid | |
| Entities Involved: | |
| Tree Lot Coordinator, Scout, Tree folder, Tree record | |

|  |  |
| --- | --- |
| Use Case Name: | 1. Delete a Tree |
| Description: | |
| To remove a tree from the tree folder | |
| Preconditions: | |
| 1. Tree exists in the tree folder | |
| Workflow: | |
| 1. Scout approaches TLC and requests to delete a tree 2. The TLC requests the barcode of the tree that the Scout wish to remove 3. Scout provides the TLC with the information requested in step 2 4. TLC retrieves the Tree record using the barcode information provided by the scout in step 3 5. TLC asks Scout to confirm Tree deletion 6. Scout confirms the information from step 5 7. TLC removes Tree record from Tree folder 8. TLC notifies the Scout of successful deletion | |
| Results: | |
| 1. Tree is successfully deleted from the tree folder | |
| Alternates: | |
| 1. Tree is sold 2. Scout does not confirm deletion | |
| Entities Involved: | |
| Scout, TLC, Tree record, Tree Folder | |

|  |  |
| --- | --- |
| Use Case Name: | 1. Add a Tree Type |
| Description: | |
| This will add a new tree type to the Tree Type folder | |
| Preconditions: | |
| 1. Tree Type does not already exist in the Tree Type Folder | |
| Workflow: | |
| 1. Scout approaches TLC and requests to add a new Tree Type 2. TLC requests the Scout to provide the Barcode prefix, Type Description, and cost of the new Tree Type 3. Scout provides TLC with the Barcode prefix, Type Description, and cost of the new Tree Type 4. TLC creates a new Tree Type record with the information provided by the Scout in step 3 5. TLC verifies that the BarcodePrefix created in step 3 does not already exist as a BarcodePrefix for an already existing Tree record in the Tree Folder 6. TLC files Tree Type record created in step 10 to the Tree Type Folder 7. TLC notifies the Scout that the addition of the Tree Type to the Tree Type folder has been successful | |
| Results: | |
| 1. A new tree type will be successfully added to the Tree Type folder | |
| Alternates: | |
| 1. Tree Type already exists in the Tree Type Folder | |
| Entities Involved: | |
| Scout, TLC, Tree Type record, Tree Type Folder | |

|  |  |
| --- | --- |
| Use Case Name: | 1. Update a Tree Type |
| Description: | |
| To update information of the Tree type in the Tree Type folder | |
| Preconditions: | |
| 1. Tree Type exists in Tree Type folder | |
| Workflow: | |
| 1. Scout approaches TLC and requests to update a Tree Type 2. TLC retrieves the collection of Tree Types 3. TLC provides the collection found in step 2 to the Scout and asks   them to choose the desired Tree Type   1. Scout indicates to the TLC the desired Tree Type from the given   collection provided in step 3   1. The TLC retrieves the desired Tree Type record requested in step   4   1. The TLC provides the Scout with the Notes in the Tree Type record retrieved   in step 5 and asks the Scout to update them   1. The Scout updates the Notes in the Tree Type record provided in   step 6 and returns them to the TLC   1. The TLC verifies the information provided by the Scout in step 7 2. TLC updates the Notes with the information verified in step 8 in   the Tree Type record retrieved in step 5   1. TLC refiles the Tree Type record updated in step 9 to the Tree   Type Folder   1. TLC notifies the Scout that the Tree Type selected in step 10 has been   successfully updated in the Tree Type folder | |
| Results: | |
| 1. Selected Tree Type is updated | |
| Alternates: | |
| 1. Tree Type cannot be found in the collection of Tree Type records 2. Scout attempts to update notes with invalid information | |
| Entities Involved: | |
| Scout, TLC, Tree Type Folder, Tree Type record | |

|  |  |
| --- | --- |
| Use Case Name: | 1. Open a Shift |
| Description: | |
| Creates a new shift record | |
| Preconditions: | |
| 1. TLC doesn’t have a record of an open session ID | |
| Workflow: | |
| 1. Scout approaches TLC and requests Opening a Shift 2. TLC requests that the Scout provide the StartTime, EndTime, and StartingCash amount in the Cash register 3. Scout provides TLC with the information requested in step 2 4. TLC creates a Session record with the current date as the StartDate, StartingCash from the information provided in step 3, TotalCheckTransationAmount as 0, EndingCash as 0, and EndingTime and StartingTime from the information provided in step 3 5. TLC files the Session Record created in step 4 into the Session folder 6. TLC retrieves the auto generated Session ID from the just filed Session record 7. TLC retrieves the collection of all Scout records in the Scout Folder and provides them to the Scout and requests the Scout select which Scout records will work the shift 8. For each Scout record the Scout selects:    1. TLC creates a Shift record with the ID from step 6 as the SessionID, the selected Scout’s ID as ScoutID, and StartTime as the Start Time of the Session record created in step 4    2. TLC requests the CompanionName for this Scout, the EndingTime for this scout and the number of hours the companion will work from the Scout    3. Scout provides all the above requested information to the TLC    4. TLC adds adds all the provided information provided in step 8.c to the Shift record    5. TLC files the Shift record in the Shift folder 9. TLC notifies the Scout that a Session has been created and all of the Scouts from the selected shift had been added to the shift folder | |
| Results: | |
| 1. A Session record is created and filed in the Session Folder 2. Shift records are created related to the Session record and are filed in the Shift Folder | |
| Alternates: | |
| 1. Scout(s) are not found when retrieving the collection of Scout records 2. Not all scouts/companions are present at the   start of shift | |
| Entities Involved: | |
| TLC, Scout, Scout Folder, Scout record, Session Folder, Session record, Shift Folder, Shift record | |

|  |  |
| --- | --- |
| Use Case Name: | 1. Sell a Tree |
| Description: | |
| A tree is sold to a customer | |
| Preconditions: | |
| 1. Tree Exists in Tree Folder 2. Tree’s Status is ‘Available’ 3. Tree’s Tree Type exists in Tree Type Folder | |
| Workflow: | |
| 1. Scout approaches TLC and requests to sell a Tree to a Customer 2. TLC requests the Tree’s barcode from the Scout 3. Scout Provides the Tree’s barcode to the TLC 4. TLC retrieves the Tree from the Tree folder where the barcode matches the barcode provided by the Scout in step 3 5. TLC verifies that the Tree’s Status is ‘Available’ or ‘Damaged’ 6. TLC extracts the barcode prefix from the barcode provided by the Scout in step 3 7. TLC retrieves the Tree Type record from the Tree Type Folder using the barcode prefix extracted in step 6 8. TLC retrieves the Cost for the Tree Type retrieved in step 7 9. TLC asks Scout to acknowledge the cost 10. Scout confirms cost to the TLC 11. TLC asks the Scout if the payment type will be Cash or Check 12. The Scout responds to TLC with requested payment type 13. TLC requests the Scout provides one or more of Customer Name, Customer Phone, and Customer Email 14. The Scout provides one or more of Customer Name, Customer Phone, and Customer Email 15. TLC retrieves the SessionID for the current Session 16. TLC creates a Transaction record with:     1. the ID retrieved in step 15 as SessionID     2. ‘Tree Sale’ as TransactionType     3. the barcode provided in step 3 as Barcode     4. the amount provided in step 10 as the TransactionAmount     5. payment type provided in step 12 as PaymentMethod     6. Customer’s Name, Phone, and Email as provided in step 14     7. The current date for TransactionDate and DateStatusUpdated     8. The current time for TransactionTime 17. TLC files Transaction record created in step 16 in the Transaction Folder 18. TLC updates the Tree Record retrieved in step 4, changing the Status to ‘Sold’ and DateStatusUpdated to the current date 19. TLC refiles the Tree record updated in step 18 20. TLC notifies Scout that the Transaction record has successfully been created and filed | |
| Results: | |
| 1. A Transaction record is created and added to the Transaction folder | |
| Alternates: | |
| 1. The Tree has already been sold or removed 2. The Tree is not found in Tree Folder 3. The Tree Type does not exist | |
| Entities Involved: | |
| TLC, Customer, Session, Transaction Folder, Transaction record, Tree Folder, Tree record, Tree Type Folder, Tree Type record | |

|  |  |
| --- | --- |
| Use Case Name: | 1. Close a Shift |
| Description: | |
| Close an existing shift record | |
| Preconditions: | |
| 1. TLC must have a record of an open session ID | |
| Workflow: | |
| 1. Scout approaches TLC and requests to close a Shift 2. TLC retrieves the collection of all Transaction records associated with the active Session ID from the Transaction folder 3. TLC retrieves the Session record associated with the active Session ID from the Session folder 4. For each Transaction from the collection retrieved in step 2 that has TransactionType value as “cash”, TLC adds TransactionAmount from the Transaction record to the EndingCash in the Session record retrieved in step 3 5. For each Transaction from the collection retrieved in step 2 with TransactionType value as “check”, TLC adds TransactionAmount from the Transaction record to the TotalCheckTransaction in the Session record retrieved in step 3 6. TLC sets the EndingCash to the sum of StartingCash value in the Session record plus EndingCash in the Session record 7. TLC refiles the Session record in the Session Folder 8. TLC notifies the scout that the shift is ended | |
| Results: | |
| 1. Session is closed 2. Cash total is updated 3. Check transactions are totaled | |
| Alternates: | |
| 1. TLC doesn’t have a record of an existing session ID | |
| Entities Involved: | |
| TLC, Scout, Session Folder, Session record, Shift Folder, Shift record | |