|  |  |
| --- | --- |
| Use Case Number | 9.1 |
| Use Case Name | OfflineActionQueue |
| Participating Actors | User |
| Goal | System saves actions |
| Trigger | User commits action while offline |
| Precondition | System has no connection  User has committed action via other Use Case |
| Postcondition | On commit, user is informed action was queued |
| Basic Flow | 1. User commits action 2. System detects lack of connection 3. System sends alert box of impossibility and requests permission to queue (auto-execute) 4. User grants permission 5. System adds action to queue |
| Exceptions | 4 If user doesn’t grant permission  4.1.1 User rejects notification  4.1.2 Action is not queued  4.1.3 Action is saved for User’s later consideration |
| Qualities | Require minimal but notable input |
| Constraint | Queue should be able to hold any number of actions |
| Includes | Creating an inventory item use case  Proposing trade UC |
| Extends |  |
| Related Artifacts | US09.01.01, US09.02.01 |
| Notes | Saved(4.1.3) actions are similar to counter proposal files |
| Open Issues | US01.01.01, US03.01.01, will be answered by use cases referenced by this UC |
|  |  |

Glossary

Action refers to sequences of events detailed by other use cases, which result in changes. (creating an inventory, proposing trades)

Execute refers to actions be done

Commit refers to end of other use cases where the action, would execute if the app had a connection.

|  |  |
| --- | --- |
| Use Case Number | 9.2 |
| Use Case Name | OfflineActionQueueExcecute |
| Participating Actors |  |
| Goal | Execute queued actions |
| Trigger | Connection is established |
| Precondition | System lacked Connection  User has granted queued actions with permission to auto-execute  System has connection |
| Postcondition | OAQ(Offline Action Queue) is empty  User has waiting notification that offers to show details about actions that were executed |
| Basic Flow | 1. System checks that actions are still possible 2. System executes actions 3. System sends User Notification that actions were executed |
| Exceptions | 1. Actions unable to be executed  1.1 System moves Action to saved/limbo file for User review  1.2 System sends User Notification that actions need to be reviewed |
| Qualities | Use Case requires no awareness of User |
| Constraint |  |
| Includes |  |
| Extends | BrowseFriendsInventories |
| Related Artifacts | US09.01.01, US09.02.01 |
| Notes |  |
| Open Issues |  |
|  |  |

|  |  |
| --- | --- |
| Use Case Number | 9.3 |
|  |  |
| Use Case Name | CacheFriendsInventories |
| Participating Actors |  |
| Goal | Cache friends inventories for offline perusal |
| Trigger | Begin Viewing a Friends Inventory |
| Precondition | Friends Inventory is downloaded  Working connection |
| Postcondition | Cached inventories are up to date |
| Basic Flow | 1. System checks if friends inventories have changed (1 var) 2. System throws out old cache and saves new one |
| Exceptions |  |
| Qualities | Should require minimal processing |
| Constraint | Should be able to be deactivated in user preferences |
| Includes |  |
| Extends |  |
| Related Artifacts | US09.03.01 |
| Notes |  |
| Open Issues | If we go with the trade multiplicity of 0..n borrower items for 1 owner item theres no reason to view or hold friends inventories seperately |
|  |  |