**Admin Maintenance - Fully Dressed Use Case Model #3**

**Revision History**

|  |  |  |  |
| --- | --- | --- | --- |
| **Version** | **Date** | **Description** | **Author** |
| Inception Draft | Sep 20, 2018 | First Draft. To be refined during elaboration. | Eric Guzman,  Mike Peralta, Alessandro Quezada |
| Elaboration review | Oct 24, 2018 | Changed use case number 12 added system shutdown and system booting for running maintenance like reviewing flagged accounts. Added a use case diagram as well. | Mike, Eric, Alessandro |
| Elaboration 2 | Dec 11, 2018 | Fixed layout fully dressed uses cases are in separate docs. Added description for SSD. | Eric Guzman, Alessandro Quezada |

|  |  |  |  |
| --- | --- | --- | --- |
| Use Cases | | | |
| **Use Case #** | **Actor** | **Goal** | **Team Member** |
| 3 | Admin | Admin Maintenance | Mike |

# Use Case Model

# 

# Use Case 3 - Admin Maintenance - Fully Dressed

|  |  |
| --- | --- |
| **Scope** | Asteroids |
| **Level** | User Goal |
| **Primary Actor** | Admin |
| **Stakeholders and Interests** | Owners want a cost-effective/time-effective method to remove Bad Players from the system, and also apply updates and patches. This improves the quality of the game/social ecosystem, increasing overall Player satisfaction, increasing user loyalty and thus revenue.  Admins want to do their job effectively and time-efficiently, and want a company that thrives.  All Players want an updated+safe system to use.  Good Players want as few Bad Players as possible, because Bad Players make the game and social environment less enjoyable.  Bad Players want to be able to cheat or break the rules of the system and get away with it. Haha Bad Players: This Use Case isn’t for you. |
| **Preconditions** | At least one Player in the system has been flagged for abuse, breaking rules, gameplay cheating by our heuristic analysis anti-cheating algorithms (HAACS), etc |
| **Success Guarantee** | Admin was able to manually review flagged players, and successfully apply judgment (false positive, ignore, forgive, defer, ban, etc) to all players reviewed.  Players have had the judgment permanently recorded to their account (some Players have been temporarily/permanently banned from the system, others reset to good standing, etc).  System had patches successfully applied. System was restarted, if required. |
| **Main Success Scenario** | Admin wants to apply updates, and manually review some players that have been flagged for potential rule-breaking and/or cheating within the system. Admin authenticates with the system. Admin asks the system for the next incident/record that needs to be manually reviewed. Admin is presented with Player’s information, activity history, and other data relevant to the incident/record. Admin determines Player has indeed broken rules and/or cheated. Admin asks System to ban Player. System bans Player. Incident is considered resolved by the System. Admin continues to review zero or more flagged players as desired or until no more unresolved incidents remain. Admin is finished reviewing. System aggregates all judgments (Admin actions) for further analysis and improvements (ie: to improve HAACS). Admin then asks the system to apply updates. System applies updates. System notifies Admin that a restart is required. Admin performs a system shutdown. Admin then starts system again. |
| **Extensions** | 1. \*\*\* If the Player has been flagged incorrectly; False positive \*\*\*   1. Admin requests the system mark the incident as a false positive. 2. System records the incident as a False Positive; No negative consequences will be applied to the Player’s account 3. Player is notified of the False Positive with an Apology, if the Player was initially notified for the flag.   2. \*\*\* If the Admin cannot reasonably make a determination \*\*\*   1. Admin requests to the system, to ignore this incident; System records that this Admin has requested to ignore this incident. 2. Admin is no longer presented with this flag incident, but other Admins can still see it (unless they have ignored it, too)   3. \*\*\* If all active Admins ignore the same specific incident \*\*\*   1. System detects that all recently active Admins have ignored some specific incident, thus it is reasonable to believe the incident will go unresolved 2. System will attach an extra message to the flag incident that all Admins had ignored it, with a reminder that the Admins should try their best to resolve every incident 3. System clears the “Ignore” status for this incident, for all Admins; Admins will now see this in their review area again   4. \*\*\* If an Admin forgives an incident \*\*\*   1. Admin requests to the system, that this incident be marked as a forgiveness 2. System records forgiveness, but maintains the incident’s association with the Player’s account, so it can be considered in future flag incidents 3. Player is notified of the situation   5. \*\*\* If an Admin defers an incident \*\*\*   1. Admin requests System defer incident (perhaps because a judgement cannot be made until additional information becomes available) 2. System records incident deferred, for that amount of time 3. Incident considered temporarily resolved by System, and will not show up for review to any Admin until the defer time has elapsed   5. \*\*\* If no updates were found \*\*\*   1. Admin will simply not need to shutdown+start the system after. |
| **Special Requirements** | Once any Admin has begun viewing an incident, no other Admins should be able to see the same incident until the first Admin is finished (ie: Made a judgment, ignored, etc) |
| **Technology and Data Variations List** |  |
| **Frequency of Occurrence** | Will occur more as our user base increases  In the beginning: perhaps very rarely  Once the user base is large: Very frequently; Perhaps as a function of population size and other factors |
| **Miscellaneous** | Appeal Process?  Detect sources of False Positives and False Negatives? (ie: Bad Players, bad algorithms, inappropriate policies / protocols, etc) |
| **System Sequence Diagram**    The Admin maintenance starts with receiving the latest HACCS report and making a decision based on that report. After, the Admin can apply updates to the server. | |

# 