**Functional Description – Sort Modification**

Document Version: 1.0 (Draft)

**Document Control**

**Version History**

|  |  |  |  |
| --- | --- | --- | --- |
| **Date** | **Version #** | **Author** | **Comments** |
| 2/26/2012 | 0.1 | [ssaxena@appthority.com](mailto:ssaxena@appthority.com) | Created the template |
| 03/06/2012 | 0.1 | [ssaxena@appthority.com](mailto:ssaxena@appthority.com) | Initial Draft with TBD functionalities |
|  |  |  |  |
|  |  |  |  |

**Approvals**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **#** | **Approver Name** | **Project Role** | **Date** | **Signature/ Electronic Approval** |
|  |  |  |  |  |
|  |  |  |  |  |

**Table of Contents**

1. Introduction 4

2. Sort Modification Scope 5

2.1 “As is” High Level Overview 5

3. Functional Modification: Sort 7

3.1 My Apps - Modification 7

3.1.1 My App – List View 7

3.2 Explore Modification (To be Decided – Likely Future Modification) 8

3.2.1 Explore – List View 8

3.3 Group Modification (To be Decided – Likely Future Modification) 9

3.4 My Policy Modification (To be Decided – Likely Future Modification) 10

3.4.1 Define Policy 10

3.4.2 Remediate 10

3.4.3 Global App Whitelist 11

3.4.4 Global App Blacklist 11

3.4.5 EULA 11

4. Non-Functional Modifications 12

4.1 Database Modifications 12

4.2 Portal Modifications 12

# Introduction

The purpose of this document is to provide description of changes for implementing Sort functionality across the portal.

The document covers high-level functional and non-functional modifications that are required for changes to Sort functionality – underlying implementation changes should be covered in design document.

References: Refer following documents for high level “As-is” system and mappings:

* Functional Description.docx
* Functionality-API-Q-Model-Map.xlsx

# Sort Modification Scope

## “As is” High Level Overview

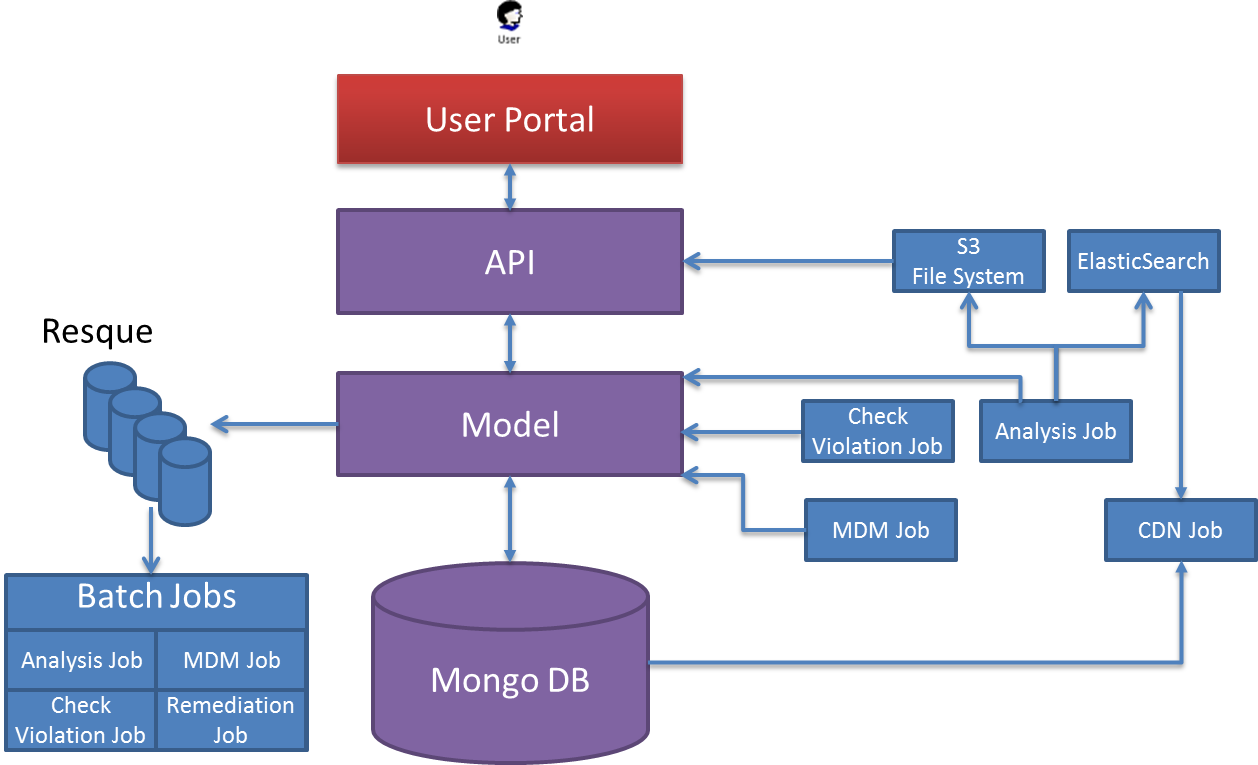


Figure 1: High-level components overview of the Appthority Portal Ecosystem  
Figure 1, high-level components overview of the Appthority Portal Ecosystem depicts different components namely applications, middleware and databases that either interact with each other directly or have dependency on a shared resource. The components are as following:

* User Portal: Available to authorized Enterprise users, portals offers functionality to user to identify risk associated with their mobile App. User has access

1. Global Apps allowed by an Enterprise
2. Private App that user may download and install from marketplace outside the organization.

* Portal API: Portal APIs are deployed to support User Portal. All the user requests from the Portal are routed through Portal API. Portal API uses data model layer to service the user request. Portal API also populates Resque.
* Model: Model is the abstraction of underlying database and provides uniform mechanism to access the database.
* Resque: Resque are populated by Portal API’s and hold the data for Analytics jobs
* Analytics Jobs: Analytics Job is responsible for processing the request from user to analyze the application. It pulls the data from Queue to process the job. After processing the App, Analytics Job updates the database, search and S3 file system.
* CDN Jobs: CDN job pulls data from Database and updates the content delivery system for improving the response for global users (non-US users)
* MDM Job: Description MDM job is responsible for connecting to the users MDM (example: MobileIron, AirWatch) and update the database with users MDM configuration like user and groups, devices etc.

# Functional Modification: Sort

User portal offers sort functionality to the users on the following portal pages:

* My Apps
* Explore
* Groups
* Policy

## My Apps - Modification

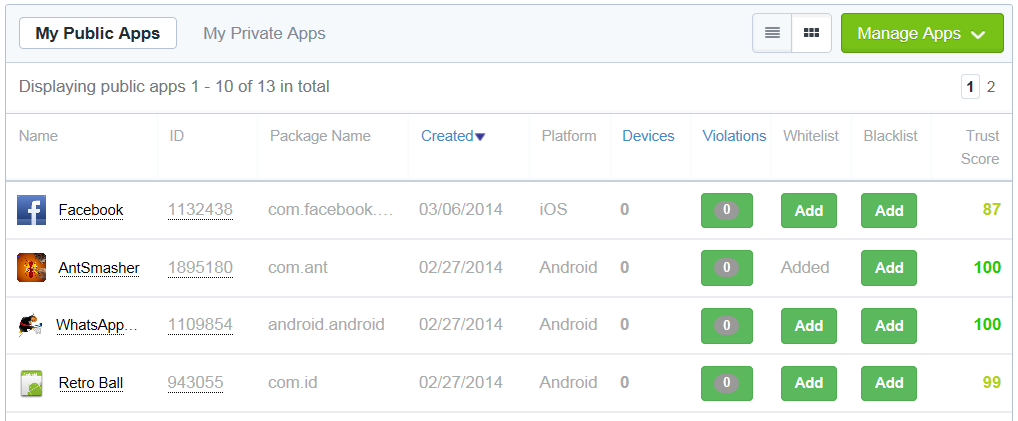


Figure: Current view of My App in list view

Currently My Apps allows user to sort on following fields in list view:

* Created
* Devices
* Violations

### My App – List View

My Public Apps and My Private Apps tabs provide list view. The modification should allow user to sort on following fields in list view, in addition to existing sort:

1. Name: Sort by Name in ascending (A through Z) and descending Alphabetical (Z through A) order.
   1. Primary field: Name
   2. Secondary field: <TBD>
2. Package Name: Sort by Package Name in ascending (A through Z) and descending Alphabetical (Z through A) order.
   1. Primary field: Package Name
   2. Secondary field: <TBD>
3. Platform: Sort by Platform Name in ascending (A through Z) and descending Alphabetical (Z through A) order.
   1. Primary field: Platform Name
   2. Secondary field: <TBD>
4. Whitelist: Sort by Whitelist (Added/Not Added)
   1. Primary field: Whitelist
   2. Secondary field: <TBD>
5. Blacklist: Sort by Blacklist (Added/Not Added)
   1. Primary field: Blacklist
   2. Secondary field: <TBD>
6. Trust Score: Sort by Trust Score ascending (0 through 100) and descending (100 through 0) numeric order
   1. Primary field: Trust Score
   2. Secondary field: <TBD>

## Explore Modification (To be Decided – Likely Future Modification)

Explore allows user to view the Popular 50, Top 50, Bottom 50 Apps for both Android and iOS in list view and grid view.

### Explore – List View

The modification should allow user to sort list view on following fields:

1. Name: Sort by Name in ascending (0-9, A through Z) and descending alphabetical (Z through A, 9-0) order.
   1. Primary field: Name
   2. Secondary field: <TBD>
2. Created: Sort on calendar date in ascending and descending order
   1. Primary field: Created
   2. Secondary Field: <TBD>
3. Trust Score: Sort on trust score on ascending (0-100) and descending order (100-0)
   1. Primary field: Trust Score
   2. Secondary Field: <TBD>

## Group Modification (To be Decided – Likely Future Modification)

Group allows user to view devices allocated to the group of end user. This modification will allow user to sort the group list by following fields:

1. First Name: Sort by First Name in ascending (A through Z) and descending Alphabetical (Z through A) order.
   1. Primary field: First Name
   2. Secondary field: <TBD>
2. Last Name: Sort by Last Name in ascending (A through Z) and descending Alphabetical (Z through A) order.
   1. Primary field: Last Name
   2. Secondary field: <TBD>
3. Phone: Sort by phone# in ascending (0 through 9) and descending (9 through 0) order.
   1. Primary field: Phone
   2. Secondary field: <TBD>

## My Policy Modification (To be Decided – Likely Future Modification)

### Define Policy

Define Policy allows user to view the Behavior configuration in a policy for a group and update it. This modification will allow user to Sort on following fields:

1. Behavior: Sort by Behavior in ascending (A through Z) and descending (Z through A) order.
   1. Primary field: Behavior
   2. Secondary field: <TBD>
2. Enabled: Sort by Enabled
   1. Primary field: Enabled
   2. Secondary field: <TBD>
3. Enforcement Action: Sort by Enforcement Action in ascending (A through Z) and descending (Z through A) order.
   1. Primary field: Enforcement Action
   2. Secondary field: <TBD>

### Remediate

Remediate allows user to initiate remedial action for device not compliant with the policy in a group. This modification will allow user to Sort on following fields:

1. Behavior: Sort by Behavior in ascending (A through Z) and descending (Z through A) order.
   1. Primary field: Behavior
   2. Secondary field: <TBD>
2. Device Count: Sort by Device Count in ascending and descending order
   1. Primary field: Device Count
   2. Secondary field: <TBD>
3. Enforcement Action: Sort by Enforcement Action in ascending (A through Z) and descending (Z through A) order.
   1. Primary field: Enforcement Action
   2. Secondary field: <TBD>

### Global App Whitelist

Global App whitelist allows user to list all the apps in white list and remove them from whitelist. This modification will allow user to Sort the Apps on Package name in ascending (A through Z) and descending (Z through A) order.

### Global App Blacklist

Global App blacklist allows user to list all the apps in black list and remove them from blacklist. This modification will allow user to Sort the Apps on Package name in ascending (A through Z) and descending (Z through A) order.

### EULA

EULA lists the Apps for which EULA is approved/rejected or Pending.

For Apps with approved/reject EULA, user will be able to sort on following fields:

1. Name: Sort by Name in ascending (A through Z) and descending Alphabetical (Z through A) order.
   1. Primary field: Name
   2. Secondary field: <TBD>
2. Package Name: Sort by Package Name in ascending (A through Z) and descending Alphabetical (Z through A) order.
   1. Primary field: Package Name
   2. Secondary field: <TBD>
3. Platform: Sort by Platform Name in ascending (A through Z) and descending Alphabetical (Z through A) order.
   1. Primary field: Platform Name
   2. Secondary field: <TBD>
4. Created: Sort on calendar date in ascending and descending order
   1. Primary field: Created
   2. Secondary Field: <TBD>
5. Trust Score: Sort by Trust Score ascending (0 through 100) and descending (100 through 0) numeric order
   1. Primary field: Trust Score
   2. Secondary field: <TBD>

# Non-Functional Modifications

To implement Sort modifications following non-functional modifications will be required:

1. Database: Current database design does not efficiently support changes to functionality. Hence, the design needs to be changed to support enhancement to existing functionality as well as foreseeable functionality that may be added in future.
2. Portal Modifications to interact with new schema/Database

## Database Modifications

Current database is in Mongo DB, a columnar DB. Due to existing issues, this database needs to be migrated to relational database management system. The relational database selected for this modification PostgreSQL. Following changes are required:

1. Create new schema for PostgreSQL
   * Support existing Portal functionality and enhancements as described in Chapter 3 through new schema
   * Support batch job’s existing functionality through new schema
2. Migrate data from Mongo DB to PostgreSQL
   * Bulk data Import
   * Import incremental changes till all the background jobs migrate to PostgreSQL
3. Automate creation of database (schema, test data load) through scripts
4. Admin Task: Support for clustering
5. Admin Task: Support for backup
6. Support following future functionality:
   * Functionalities identified in Chapter 3 as “To be Decided – Likely Future Modification”
   * Capability to add multiple login users for an organization
   * Support data segregation between organizations
   * ~~Search the user’s Apps (Global and Private)~~

## Portal Modifications

Apart from building enhancements as described in chapter 3, the Portal Application (Client/Controller/Model) should be modified to support:

1. Interaction with new schema/database: Portal uses APIs to pull data for display. The APIs uses Models that are based on current Mongo DB architecture. The Model’s needs to be refactored to use data in PostgreSQL.
2. Services Architecture: Current API architecture needs to be refactored to confirm to uniform Services Architecture.

## Job Modifications

All the jobs associated with processing of queues and CDN needs to migrate to new database. The jobs are as follows:

* MDM Job
* App Analysis Job
* Check Violation Job
* Remediation Job
* CDN Job