**Project: JEDi Signage**

**Media Deploy to Player**

| **Test Case ID** | DEPL-General-Deploy | **Developed By** | Jared Miller |
| --- | --- | --- | --- |
| **Test Priority** | High | **Development Date** | 2/14/2018 |
| **Module Name** | Management Interface - Frontend | **Test Executed By** | TODO |
| **Test Title** | Media Deploy to Player | **Test Execution Date** | TODO |
| **Testing Tool** | TODO |  |  |
| **Description** | Verify that activating a deploy sequence will send metadata and media to designated player system | | |

| **Preconditions** | Player configured, user logged in, layout not currently used by player available |
| --- | --- |
| **Dependencies** | None |

| **Step** | **Test Steps** | **Test Data** | **Expected Result** | **Actual Result** | **Status (Pass/Fail)** | **Notes** |
| --- | --- | --- | --- | --- | --- | --- |
| 1 | Navigate to main menu |  | Navigation buttons visible |  |  |  |
| 2 | Navigate to player menu |  | Player listing visible |  |  |  |
| 3 | Choose to edit chosen player with "edit" button |  | Editable player info visible, including "add layout" button |  |  |  |
| 4 | Add selected unused layout to player |  |  |  |  |  |
| 4.1 | Select "add layout" |  | List of layouts visible, each with "+" button |  |  |  |
| 4.2 | Select "+" for unused layout |  | Layout added, layout list removed |  |  |  |
| 5 | Choose "deploy" button |  | User informed of deployment start, status displayed |  |  |  |
| 6 | Await the result of the deployment |  | "Successful deployment" message |  |  |  |

| Post Condition | Status (Pass/Fail) |
| --- | --- |
| New metadata and media were deployed via SCP (verify with stub) |  |
| Player was sent command to reload data via SSH (verify with stub) |  |

**Project: JEDi Signage**

**Media Upload and Info Endpoints - JPEG**

| **Test Case ID** | UPLD-jpeg-api | **Developed By** | Jared Miller |
| --- | --- | --- | --- |
| **Test Priority** | High | **Development Date** | 2/14/2018 |
| **Module Name** | Management Interface - API | **Test Executed By** | TODO |
| **Test Title** | Media Upload and Info Endpoints - JPEG | **Test Execution Date** | TODO |
| **Testing Tool** | TODO |  |  |
| **Description** | Verify that posting a jpeg file to the POST /api/media/upload endpoint creates a temporary file and corresponding pending entry in the server datastore, POST /api/media/:id sets info and marks as completed, that GET /api/media/:id returns the correct info for each phase, and that /media/:id returns the raw file. | | |

| **Preconditions** | Test tool authenticated |
| --- | --- |
| **Dependencies** | None |

| **Step** | **Test Steps** | **Test Data** | **Expected Result** | **Actual Result** | **Status (Pass/Fail)** | **Notes** |
| --- | --- | --- | --- | --- | --- | --- |
| 1 | POST file.jpeg to /api/media/upload | File.jpeg | API returns status 200 with message body of  {  "status":"OK",  "id":"[ID of new media item]"  } |  |  |  |
| 2 | GET /api/media/:id | ID from step 1 | API returns status 200 with message body of media item info. Item status is "pending" |  |  |  |
| 3 | POST /api/media/:id | ID from step 1, data to fill fields | API returns status 200 with message body of  {  "status":"OK",  "id":"[same ID as (1)]"  } |  |  |  |
| 4 | GET /api/media/:id | ID from step 1 | API returns status 200 with message body of media item info, as set in (3). Item status is "confirmed" |  |  |  |
| 5 | GET /media/:id | ID from step 1 | Server serves file.jpeg |  |  |  |

| Post Condition | Status (Pass/Fail) |
| --- | --- |
| Row exists in MediaItem table for file.jpeg with appropriate data |  |
| File.jpeg exists in server /media directory |  |

**Project: JEDi Signage**

**Media Upload Endpoint File Type Validation (Security)**

| **Test Case ID** | UPLD-security-filetypes | **Developed By** | Jared Miller |
| --- | --- | --- | --- |
| **Test Priority** | Medium | **Development Date** | 2/14/2018 |
| **Module Name** | Management Interface - API | **Test Executed By** | TODO |
| **Test Title** | Media Upload Endpoint File Type Validation | **Test Execution Date** | TODO |
| **Testing Tool** | TODO |  |  |
| **Description** | The system should not accept files which are not valid image files for upload. Attempts to upload these files should return an error message and not be saved to the server except as rejected items in the database. | | |

| **Preconditions** | Test tool authenticated |
| --- | --- |
| **Dependencies** | None |

| **Step** | **Test Steps** | **Test Data** | **Expected Result** | **Actual Result** | **Status (Pass/Fail)** | **Notes** |
| --- | --- | --- | --- | --- | --- | --- |
| 1 | POST file.jpeg to /api/media/upload | File.jpeg | API returns status 200 with message body of  {  "status":"OK",  "id":"[ID of new media item]"  } |  |  | Verify an actual image file is accepted |
| 2 | POST invalid.jpeg to /api/media/upload | Invalid.jpeg, which is any non-image file with an extension of jpeg | API returns status 415 (unsupported media type), with a message body of  { "status":"Invalid file" } |  |  | Verify that files are checked by contents |
| 3 | POST text.txt to /api/media/upload | Test.txt (any text file) | API returns status 415 (unsupported media type), with a message body of  { "status":"Invalid file" } |  |  | Or, upload any non-relevant file type |
| 4 | POST video.mp4 to /api/media/upload | Video.mp4 (any video file) | API returns status 415 (unsupported media type), with a message body of  { "status":"Invalid file" } |  |  | Though a media file, the system does not currently support video |

| Post Condition | Status (Pass/Fail) |
| --- | --- |
| Row exists in MediaItem table for file.jpeg as pending |  |
| Row exists in MediaItem table for each additional file as "rejected" |  |
| File.jpeg exists on the server, but none of the other submitted files |  |

**Project: JEDi Signage**

**User Login Flow**

| **Test Case ID** | LOGIN-flow | **Developed By** | Jared Miller |
| --- | --- | --- | --- |
| **Test Priority** | Medium | **Development Date** | 2/17/2018 |
| **Module Name** | Management Interface - Frontend | **Test Executed By** | Jared Miller |
| **Test Title** | Verify correct login flow | **Test Execution Date** | 2/17/2018 |
| **Testing Tool** | Nightwatch |  |  |
| **Description** | The system should correctly redirect an unauthenticated user to /login, prompt for credentials, accept credentials, and redirect to the menu. Invalid credentials submissions should be shown the login prompt again(See LOGIN use case for details) | | |

| **Preconditions** | Test tool not authenticated, user account test:password exists |
| --- | --- |
| **Dependencies** | None |

| **Step** | **Test Steps** | **Test Data** | **Expected Result** | **Actual Result** | **Status (Pass/Fail)** | **Notes** |
| --- | --- | --- | --- | --- | --- | --- |
| 1 | Navigate to / |  | User agent redirected to /login, username and password prompt visible, as well as a "login" button |  | Pass |  |
| 2 | Enter an invalid username and password, click login button | bad:bad | User agent remains at /login, prompt is displayed again with message "Invalid login" |  | Pass |  |
| 3 | Enter a valid username and password, click login button | test:password | User agent is given a "session" cookie, directed to / and can see the menu |  | Pass |  |

| **Post Condition** | **Status (Pass/Fail)** |
| --- | --- |
| User agent has session cookie | Pass |
| Database has corresponding Session row pointing to correct User | TODO |

**Media Upload and Info Endpoints - JPEG**

| **Description** | Verify that posting a jpeg file to the POST /api/media/upload endpoint creates a temporary file and corresponding pending entry in the server datastore, POST /api/media/:id sets info and marks as completed, that GET /api/media/:id returns the correct info for each phase, and that /media/:id returns the raw file. | | |
| --- | --- | --- | --- |

| **Step** | **Test Steps** | **Test Data** | **Expected Result** |
| --- | --- | --- | --- |
| 1 | POST file.jpeg to /api/media/upload | File.jpeg | API returns status 200 with message body of  {  "status":"OK",  "id":"[ID of new media item]"  } |
| 2 | GET /api/media/:id | ID from step 1 | API returns status 200 with message body of media item info. Item status is "pending" |
| 3 | POST /api/media/:id | ID from step 1, data to fill fields | API returns status 200 with message body of  {  "status":"OK",  "id":"[same ID as (1)]"  } |
| 4 | GET /api/media/:id | ID from step 1 | API returns status 200 with message body of media item info, as set in (3). Item status is "confirmed" |
| 5 | GET /media/:id | ID from step 1 | Server serves file.jpeg |

**Project: JEDi Signage**

**Schedule Modification**

| **Test Case ID** | Schedule 1 | **Developed By** | Ariunbold Batsaikhan |
| --- | --- | --- | --- |
| **Test Priority** | Medium | **Development Date** | 4/15/2018 |
| **Module Name** | Playlist Interface | **Test Executed By** | Ariunbold Batsaikhan |
| **Test Title** | Schedule modification | **Test Execution Date** | 4/15/2018 |
| **Testing Tool** | TODO |  |  |
| **Description** | Sub System Testing. Schedule interface is under management interface and playlist is most important and repetitive task. It requires to show signanes to the audience and connected to all other components of the system. | | |

| **Preconditions** | Test tool authenticated |
| --- | --- |
| **Dependencies** | None |

| **Step** | **Test Steps** | **Test Data** | **Expected Result** | **Actual Result** | **Status (Pass/Fail)** | **Notes** |
| --- | --- | --- | --- | --- | --- | --- |
| 1 | Select profile | Eric | Entered eric’s account | Logged-in | pass |  |
| 2 | Change Schedule | Only Mon and Wed On |  | Schedule Confirmed | pass |  |
| 3 | Schedule deployed and played |  |  | Signage played on time | pass |  |
| 4 |  |  |  |  |  |  |

| Post Condition | Status (Pass/Fail) |
| --- | --- |
| New schedule must be updated in layout, media and profiles |  |
|  |  |
|  |  |

**Project: JEDi Signage**

**Management Interface DataStore**

| **Test Case ID** | Datastore-1 | **Developed By** | Ariunbold Batsaikhan |
| --- | --- | --- | --- |
| **Test Priority** | Medium | **Development Date** | 4/15/2018 |
| **Module Name** | Management Interface | **Test Executed By** | Ariunbold Batsaikhan |
| **Test Title** | Data Retrieval and Maintenance test | **Test Execution Date** | 4/15/2018 |
| **Testing Tool** | DB Inspect |  |  |
| **Description** | Data quality check in existing data. Visual inspection, duplication and efficiency check by combining subsets of tables. | | |

| **Preconditions** | Test tool authenticated, SQL Injected |
| --- | --- |
| **Dependencies** | None |

| **Step** | **Test Steps** | **Test Data** | **Expected Result** | **Actual Result** | **Status (Pass/Fail)** | **Notes** |
| --- | --- | --- | --- | --- | --- | --- |
| 1 | select \* from all\_tables | DB | Database returns everything from it | Database returned it | Pass |  |
| 2 | SELECT user, sign, priority FROM table1 INNER JOIN table2 ON table1.user = table2.user; | DB | Database return joint elements from  same user name from each tables. | Database returned correct inner join query. | pass |  |
| 3 |  |  |  |  |  |  |
| 4 |  |  |  |  |  |  |

| Post Condition | Status (Pass/Fail) |
| --- | --- |
| Database must be storing datas in the subsystem with normal functionalities. | Pass |
|  |  |
|  |  |

**Project: JEDi Signage**

**Secure Shell Protocol**

| **Test Case ID** | SSH-test1 | **Developed By** | Ariunbold Batsaikhan |
| --- | --- | --- | --- |
| **Test Priority** | Medium | **Development Date** | 4/15/2018 |
| **Module Name** | Player | **Test Executed By** | Ariunbold Batsaikhan |
| **Test Title** | SSH Security Check | **Test Execution Date** | 4/15/2018 |
| **Testing Tool** | Rebex SSH Check |  |  |
| **Description** | This free SSH testing tool checks the configuration of given server accessible over internet. We don't ask you for any login or password, this service only returns information available during SSH handshake - notably supported encryption and MAC algorithms, and an overview of offered server public keys. | | |

| **Preconditions** | Test tool authenticated, System connected to the internet |
| --- | --- |
| **Dependencies** | None |

| **Step** | **Test Steps** | **Test Data** | **Expected Result** | **Actual Result** | **Status (Pass/Fail)** | **Notes** |
| --- | --- | --- | --- | --- | --- | --- |
| 1 | Go to the rebex testing tool website | https://sshcheck.com/ |  |  |  |  |
| 2 | Type in server and port info | Port 20/21 |  | Connection Timed Out | Fail | Network uses some local ports |

| Post Condition | Status (Pass/Fail) |
| --- | --- |
| None |  |

**Project: JEDi Signage**

**Secure Shell Protocol**

| **Test Case ID** | SSH-test1 | **Developed By** | Ariunbold Batsaikhan |
| --- | --- | --- | --- |
| **Test Priority** | Medium | **Development Date** | 4/15/2018 |
| **Module Name** | Player | **Test Executed By** | Ariunbold Batsaikhan |
| **Test Title** | SSH Security Check | **Test Execution Date** | 4/15/2018 |
| **Testing Tool** | Rebex SSH Check |  |  |
| **Description** | This free SSH testing tool checks the configuration of given server accessible over internet. We don't ask you for any login or password, this service only returns information available during SSH handshake - notably supported encryption and MAC algorithms, and an overview of offered server public keys. | | |

| **Preconditions** | Test tool authenticated, System connected to the internet |
| --- | --- |
| **Dependencies** | None |

| **Step** | **Test Steps** | **Test Data** | **Expected Result** | **Actual Result** | **Status (Pass/Fail)** | **Notes** |
| --- | --- | --- | --- | --- | --- | --- |
| 1 | Go to the rebex testing tool website | https://sshcheck.com/ |  |  |  |  |
| 2 | Type in server and port info | Port 20/21 |  | Connection Timed Out | Fail | Network uses some local ports |

| Post Condition | Status (Pass/Fail) |
| --- | --- |
| None |  |