**PRD – Inspection/Measurements App**

**Purpose (Why?)**

* Workorders, one of the most commonly used terms in field services, are a norm in every industry when it comes to execution of tasks. But workorders are executed under corrective maintenance when breakdown/defect is identified.
* In an energy industry for example, implications of corrective maintenance can be huge, and hence firms opt for preventive maintenance.
* Regular and scheduled inspections are carried out to monitor health of assets and hence prevent any breakdowns. By following a detailed list of steps, inspections can bring down the maintenance cost drastically.
* Inspection as a use-case is very common across industries and the product can be used as baseline to serve the purpose across cross-functional industries.

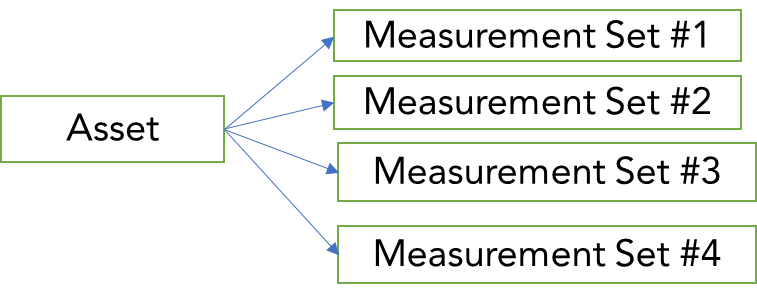
**Who’s It for?**

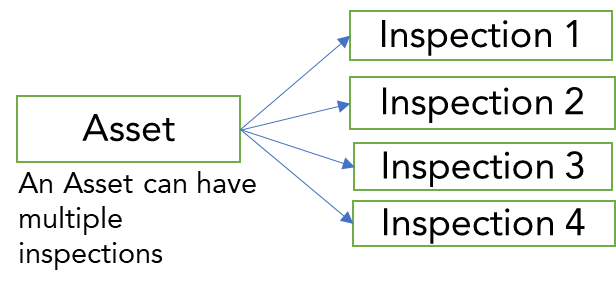
* **Field Technicians**: Those who have technical expertise and are the company’s representatives on the field. They have detailed set of inspection instructions and steps that they follow while executing an inspection.
* **Planning & Tracking Representative**: Those who directly interact with customers to understand the inspection requirements and schedules. They plan, schedule, and track the inspections to better serve customers and facilitate preventive, and corrective maintenances.

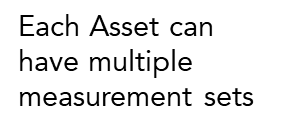
**What is it?**

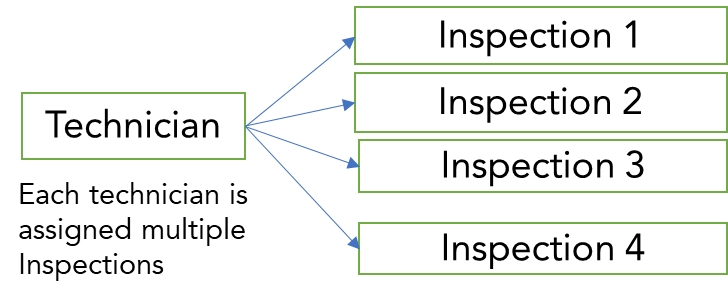
Inspection App is a combination of microApps which are subsequently built on stack of components to facilitate the Inspection workflow. Inspection App is built on 4 primary entities:

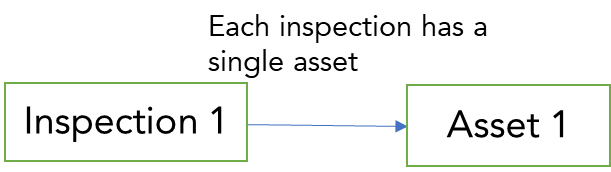
1. **Technician (User of the App)**
2. **Asset (The entity to be inspected)**
3. **Inspection (The task to be executed)**
4. **Measurements (Data points to be captured)**

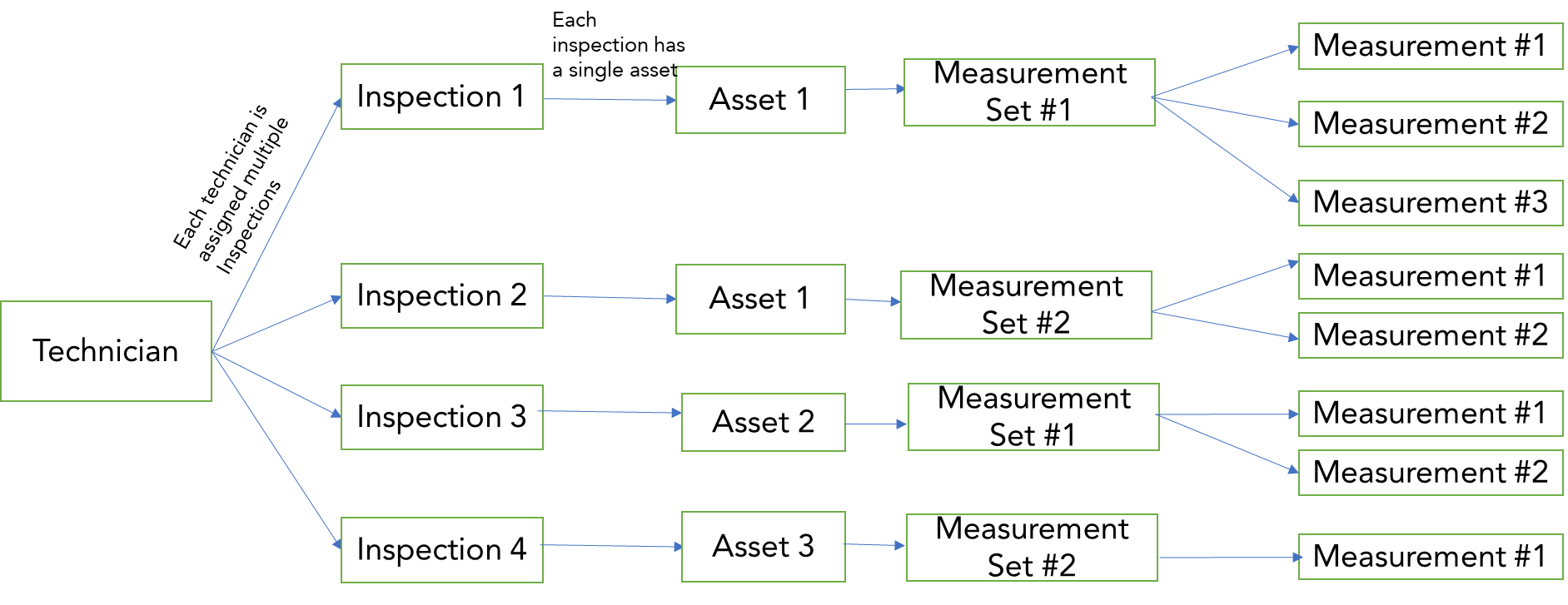
The interaction of these 4 entities serve the Inspection workflow:





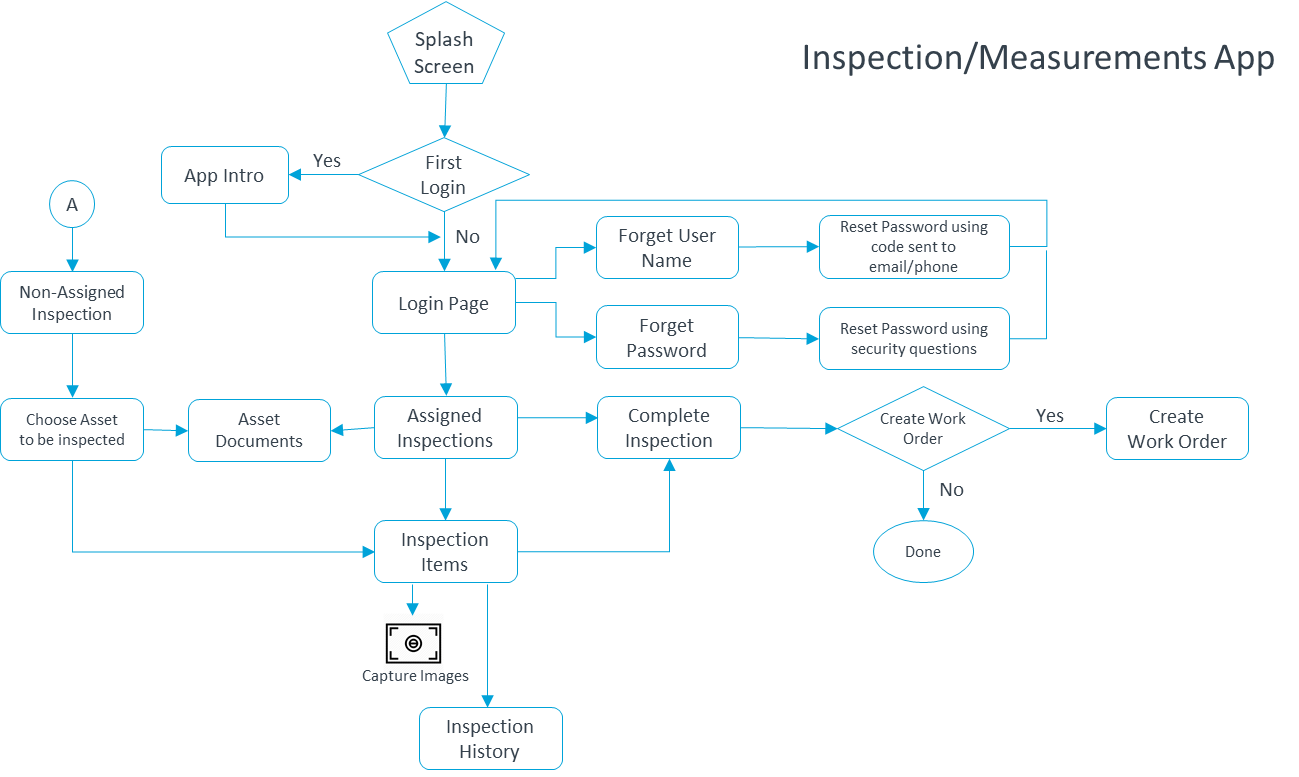


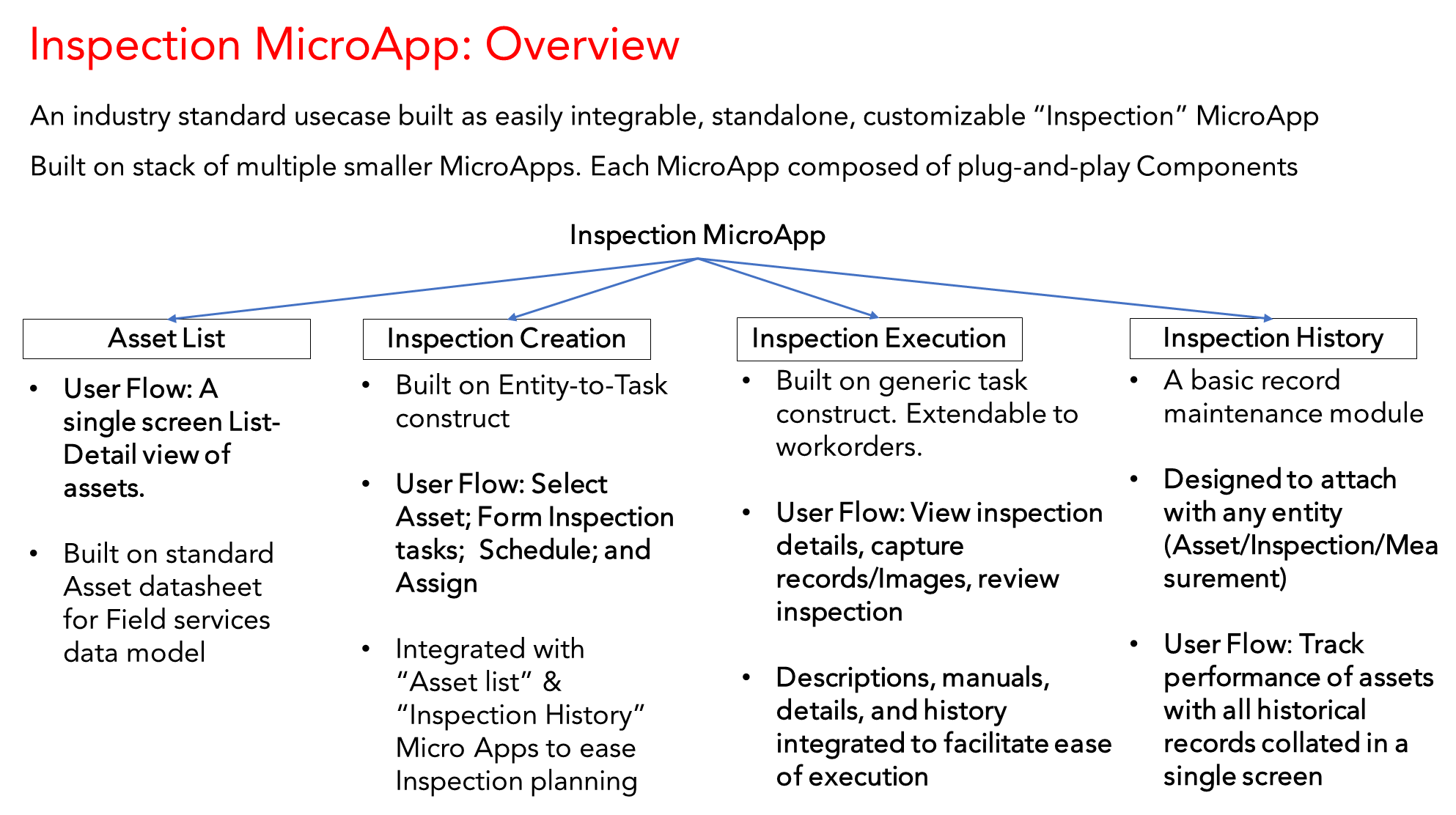




Entity Interaction Diagram

**Inspection App Flow:**

****

****

**Inspection App: Key Features**

**Offline Enabled –** Create/Schedule/Assign/Execute while offline

**List-view of work schedule –** Search/Filter capable for ease-of-use

**Details form(Per Entity to tiniest details) -** Access to unified Information repository anywhere, anytime

**Historical records (History Form) -** Record maintenance without the overhead of clumsy manual paperwork

**PDF Viewer -** View reference manuals anytime during the inspection

**Dynamic Measurement records-** Capture all kinds of records: Numeral, Yes/No, List, or a simple text

**Inspection Images -** Capture pictures per measurement to track/validate the inspection

**Review form –** Review screen for inspection validation at the site

**Signature Capture -** Complete the inspection workflow by last-mile signature authentication

**Create/Schedule/Assign –** Admin can use the same App to create and schedule new inspection requests

**User Stories**

**Epic 01: Create MicroApp for Execution of an inspection**

**User Story 01: As a technician, I should be able to see all the inspections assigned to me in a form of a tasklist, so that I can easily browse through all my tasks.**

**Acceptance Criteria:**

1. Create a component that takes tasks assigned to a user as input and displays in form of a tasklist.
2. Fields for a task should be configurable to use any task database as input. (For this product we have inspection fields, it can also be any other field service task).
3. User should be able to see the ‘task status’ inside the tasklist view.
4. User should be able to see list sorted based on scheduled timestamp (Increasing order of schedule)
5. Once completed, task should not be displayed as an entry in tasklist.

**Definition of Done**

1. Developed as a standalone component with “Asset\_Database” model as the basis.
2. Developed and tested on Android and iOS mobile platforms.
3. Approved by UI designer.
4. Passes testing and security compliance as per the company standards.
5. Integrated as part of “Inspection Execution” microapp.

**User Story 02: As a technician, I should be able to perform basic filter and search on my tasklist, so that I can reach my desired result quickly**

**Acceptance Criteria**

1. Selection exposed as part of component properties.
2. All major elements of tasklist (Those displayed inside rows of main tasklist) provided as options to select from. (Asset name, task location, Task status ….)
3. Properties extended as part of existing worksheet list component.
4. Multiple filters can be applied on the tasklist.
5. All filters can be cleared at once to get original tasklist. Filters can also be cleared stepwise. Result should reflect current filters present as the selection
6. Basic search on tasklist row elements (Asset name, task location, Task status ….)
7. Search on filtered results should be functional (Eg. List filtered based on location, then list searched for Asset Name)

**Definition of Done:**

1. Passes testing and security compliance as per the company standards.
2. Integrated as part of “Inspection Execution” microapp.
3. Approved by UI designer.

**User Story 03: As a technician, I should be able to perform execution of inspection by clicking on one of the rows of my inspection list, so that to ease my execution workflow.**

**Acceptance Criteria:**

1. Unique inspection ID is created for every new inspection of an asset. This should follow standard mentioned in Asset\_Database : “Inspection Details” sheet.
2. Execute Inspection page should have an inspection reference manual document.
3. Page should display Asset Code on which inspection is started and the Unique Inspection ID.
4. Measurement Set selected for inspection displayed as per the UX design.
5. Inspection considered executed only when all measurements are recorded and the ‘measurement set’ inspection is complete.

**Definition of Done:**

1. Passes testing and security compliance as per the company standards.
2. Integrated as part of “Inspection Execution” microapp.
3. Approved by UI designer.

**User Story 04**: **As a technician, I should be able to record measurements parameters of all types (String, Numeral, Boolean, List…), so that I can inspect all kinds of assets**

**Acceptance Criteria:**

1. Created as a measurement component with record type configurable as a component property.
2. List measurement values configurable via component property.
3. “Asset\_Database” model taken as the base back-end database model to fetch the record type, Min/Max, and validation criteria,
4. Measurement description visible as a pop-up, when clicked on info icon attached to each measurement component.
5. Validations done and prompted for each record. (Pop up if numeral measurement violates range of allowed values).

**Definition of Done:**

1. Passes testing and security compliance as per the company standards.
2. Integrated as part of “Inspection Execution” microapp.
3. Approved by UI designer.

**User story 05: As a technician, I should be able to see history per measurement record, so that I can analyze the asset performance per measurement basis**

**Acceptance Criteria**

1. Unique measurement ID created as per Asset Code/Inspection\_ID/ Measurement\_ID combination. (As defined in Asset\_Database)
2. Measurement history icon integrated as part of Measurement Component.
3. Measurement history integrated as part of “Inspection History” usecase

**Definition of Done:**

1. Passes testing and security compliance as per the company standards.
2. Integrated as part of “Inspection Execution” microapp.
3. Approved by UI designer.

**User Story 06: As a technician, I should be able to capture images for every measurement record, so that I can analyze the issue per measurement basis.**

**Acceptance Criteria:**

1. Image capture component integrated inside context of every record capture. Images captured if clicked on camera icon placed inside measurement container.
2. Explore different options of storing and retrieving images. (Binary, Box Adapter, …)
3. Images are stored using unique code for every inspection item (Measurement record) per inspection.
4. Images retrievable in the Inspection History usecase for every measurement.
5. Image capture developed as a separate component.

**Definition of Done:**

1. Passes testing and security compliance as per the company standards.
2. Integrated as part of “Inspection Execution” microapp.
3. Approved by UI designer.

**User Story 07: As a technician, I should be able to capture Signature so that I can complete inspection workflow**

1. Signature capture developed as a separate component.
2. Signature capture integrated as part of Inspection completion workflow.
3. Signature captured per inspection and thus each signature image linked to a unique inspection ID (Refer Inspection\_Header sheet inside “Asset Database”)
4. Inspection saved once Signature Capture is completed.
5. Signature is not retrieved inside Inspection History

**Definition of Done:**

1. Passes testing and security compliance as per the company standards.
2. Integrated as part of “Inspection Execution” microapp.
3. Approved by UI designer.

**Epic 02: Create MicroApp for Inspection Assignment:**

**User Story 01: As a Technician, I should be able to create an inspection on an asset inside Asset database**

**Acceptance Criteria:**

1. Technician can see list of all assets available in the database.
2. Technician can create inspection by selecting any asset in the list.
3. Inspection is auto-assigned to technician, technician doesn’t need to specifically fill the “Assigned To” field of inspection.
4. Technician can select Measurement Set (Mapped to Asset) on which inspection needs to be executed.
5. Technician can enter the scheduled time of inspection.
6. Once created, Inspection should reflect inside the technician’s inspection list.

**Definition of Done:**

1. Passes testing and security compliance as per the company standards.
2. Integrated as part of “Inspection Assignment” MicroApp.
3. Approved by UI designer.

**User Story 02: As an admin, I should be able to create inspection on any asset in the database and assign it to a technician.**

1. Admin can see list of all assets available in the database.
2. Admin can create inspection by selecting any asset in the list.
3. Admin fills the “Assigned To” field of inspection with the technician name to whom he wants to assign the inspection
4. Admin can select Measurement Set (Mapped to Asset) on which inspection needs to be executed.
5. Admin can enter the scheduled time of inspection.
6. Once created, Inspection should reflect inside the technician’s inspection list.

**Definition of Done:**

1. Passes testing and security compliance as per the company standards.
2. Integrated as part of “Inspection Assignment” MicroApp.
3. Approved by UI designer.

**Epic 03: Create MicroApp for Asset Database**

**User Story 01: As a technician or an Admin, I should be able to see all assets available inside my company’s asset database.**

**Acceptance Criteria:**

1. Asset table displayed as a list of assets to the user.
2. Once clicked, row should display details of the asset as described in the Asset Database.
3. Should be modifiable to include an such database (Inventory, Raw Materials …..)
4. Asset database can be accessed through an icon. Once clicked, Database icon should display all the assets inside database.
5. Asset details should contain reference manual document unique to each asset.
6. Asset database details should adhere to “Asset\_Details” table inside “Asset Database” model
7. Asset details filed configurable and adaptable as per the underlying database model

**Definition of Done:**

1. Passes testing and security compliance as per the company standards.
2. Integrated as part of “Asset Database” MicroApp.
3. Approved by UI designer.

**User Story 02: As a technician or Admin, I should be able to see the reference manual document for each asset, so that I can use it as a guide for inspection assignment or execution accordingly.**

**Acceptance Criteria:**

1. Documents referenced through a URL
2. Each asset has one reference document, whose link is provided inside Asset Details as well as Inspection Execution and Inspection details page.
3. Document storage and assignment created as a separate component with “Asset\_Documents” sheet as the database model for storage.
4. There can be multiple reference documents for a single Asset.

**Definition of Done:**

1. Passes testing and security compliance as per the company standards.
2. Integrated as part of “Asset Database” MicroApp.
3. Approved by UI designer.

**User Story 03: As a technician, I should be able to perform basic filter and search on my tasklist, so that I can reach my desired result quickly**

**Acceptance Criteria**

1. Selection exposed as part of component properties.
2. All major elements of asset list (Those displayed inside rows of asset list) provided as options to select from. (Asset name, asset location, Asset Type….)
3. Properties extended as part of existing worksheet list component.
4. Multiple filters can be applied on the Asset list.
5. All filters can be cleared at once to get original Asset list. Filters can also be cleared stepwise. Result should reflect current filters present as the selection
6. Basic search on Asset list row elements (Asset name, task location, Task status ….)
7. Search on filtered results should be functional (Eg. List filtered based on location, then list searched for Asset Name)

**Definition of Done:**

1. Passes testing and security compliance as per the company standards.
2. Integrated as part of “Inspection Execution” microapp.
3. Approved by UI designer.

**Epic 04: Create a MicroApp for Task History**

**User Story 01: As a technician or an admin, I should be able to see a list of history of inspections executed for an asset, so that I can assess the trend of asset’s performance.**

**Acceptance Criteria:**

1. History list displayed as a list of tasks executed in the past.
2. List displays inspection details elements inside each row.
3. History list Should be accessible through a “history icon” on the Assets page.
4. Integrated as part of “Task History” MicroApp.
5. Assess the feasibility of modelling as a standalone component.

**Definition of Done:**

1. Passes testing and security compliance as per the company standards.
2. Integrated as part of “Task History” MicroApp.
3. Approved by UI designer.

**User Story 02: As a technician or an admin, I should be able to see details of inspection executed in the past, so that I can conclude on trend in asset’s performance.**

**Acceptance Criteria:**

1. Details of an inspection task built based on “Inspection\_Detail” table of “Asset Database” model.
2. Fetch the inspection details by clicking on any one of the rows in Task history list.
3. Each inspection detail page assigned a unique Inspection\_ID.
4. Inspection details page displays the measurement results (Selected during the inspection creation).
5. Images captured during the inspection displayed as an image gallery inside the “Inspection Details” page.

**Definition of Done:**

1. Passes testing and security compliance as per the company standards.
2. Integrated as part of “Task History” MicroApp.
3. Approved by UI designer.

**Epic 05: Create MicroApp for login component.**

**User Story 01: As a technician or Admin, I need to know about the purpose of Application when I log in for first time, So that I can understand the usage better.**

**Acceptance Criteria:**

1. First login(registration) displayed as a separate button on the main screen.
2. Once user clicks on the First Login, App Intro screens are displayed to the user.
3. After user views all the intro screens, user is redirected to the login screen.
4. Intro screens must contain information on how a technician and admin can use the application.

**Definition of Done:**

1. Passes testing and security compliance as per the company standards.
2. Integrated as part of “Task History” MicroApp.
3. Approved by UI designer.

**User Story 02: As a technician or an Admin, I should be able to login successfully so that I can access and execute my tasks.**

**Acceptance Criteria:**

1. Use existing login component in the marketplace.
2. Should be able to handle both “Forgot Username” and “Forgot Password” usecases for the login.
3. Technician/Admin privileges defined inside the employee database used for the app deployment.
4. Once successful, technician should be redirected to Technician workflow and Admin to Admin workflow (First screen after login will be different for admin and technician)

**Definition of Done:**

1. Passes testing and security compliance as per the company standards.
2. Integrated as part of “Task History” MicroApp.
3. Approved by UI designer.

**Wireframes**

****