CS568 SYSTEMS DOCUMENT

# VARUN MITTAL

I was provided a partial SRS by CS568 Systems for the RoboCon Convoy and I was working as requirements engineer by the company to expand on top of the existing SRS and evolve into a fully functional and exhaustive SRS with all the high-level functionality down to low-level design in a top-down manner with detailed requirements for all the components of the system and operational modes that the RoboCon System can execute.

I used IBM Dynamic Object Oriented Requirements System (DOORS) which is a requirement management tool. It is a client–server application, with a Windows-only client and servers for Linux, Windows, and Solaris.

Some of the potential faults that I identified with the RoboCon Convoy SRS and how I tried to resolve this were:

1. **IPC (Inter Process Communication) between RCU and ACU**

There was no mention of how the RoboCon Convoy handles the messages exchanged between the individual RCUs and between RCU and ACU which involves cross-platform communication and can raise issues of interoperability as RCU runs on Linux OS whereas the ACU runs on Windows OS. There is also no mention of how the RoboCon Convoy synchronizes the speed of the individual RCUs so when the convoy is executing movement mode so as to prevent collision between any two RCUs.

Also since the RCUs are going to use a multi-threading process for passing the messages between themselves on Linux platform there can be many cases of occurrence of deadlocks due to lack of concurrency control, memory leaks, and insufficient memory address space due to lack of a garbage collector mechanism.

1. **Emergency Brake and abrupt stopping due to failure of tracking system**

When the convoy is executing the movement mode operation, and as the convoy uses the GPS , IR sensor and Video Camera for carrying out the tracking between the RCU Follower and RCU Guide while travelling on the route. If all of the tracking modes - GPS , IR and Camera fail then the RoboCon convoy will apply the emergency brake and abruptly stop in the route and send an error message to the ACU.

1. **Follower RCU and Guide RCU are not properly distinguished**

The leader RCU can also be a guide RCU for the RCU immediately behind it and also it is just mentioned in SRS that every RCU can be configured as a leader or follower but there is no proper technique for identifying the RCU leader or RCU follower which can lead to a lot of ambiguities in the requirements. So we can associate a unique id tag with every RCU for doing this process and every leader RCU will have two ids for storing the information as both guide and leader and every RCU except the leader RCU will have two ids for storing the information as guide RCU and follower RCU.

1. **Shape and type of Obstacle and calculate deviation angle from path**

There is no mention in SRS of how the leader RCU can discern the shape and type of obstacle and how it can invoke the obstacle avoidance algorithm accordingly. We can do this by putting an RFID tag on the obstacle so that the leader RCU can get the obstacle and calculate the angle of deviation from the original route and try to minimize this angle so as to successfully execute the movement mode.

1. **Algorithm Visualizer Tool**

The CS568 Algorithm Developers can also provide an algorithm visualizer tool for comparing the different types of AI and Image Processing algorithms and selecting the best one depending upon the performance metrics provided by the developers.

1. **Proposed Budget and Timeline**

The RoboCon Development Team can also provide a fixed budget and timeline for developing the RoboCon System so that the number of requirements to be implemented are less, requirements don’t have to modified too much and there is very less chance of occurrence of scope creep in the SRS.

1. **Path Planning Algorithm**

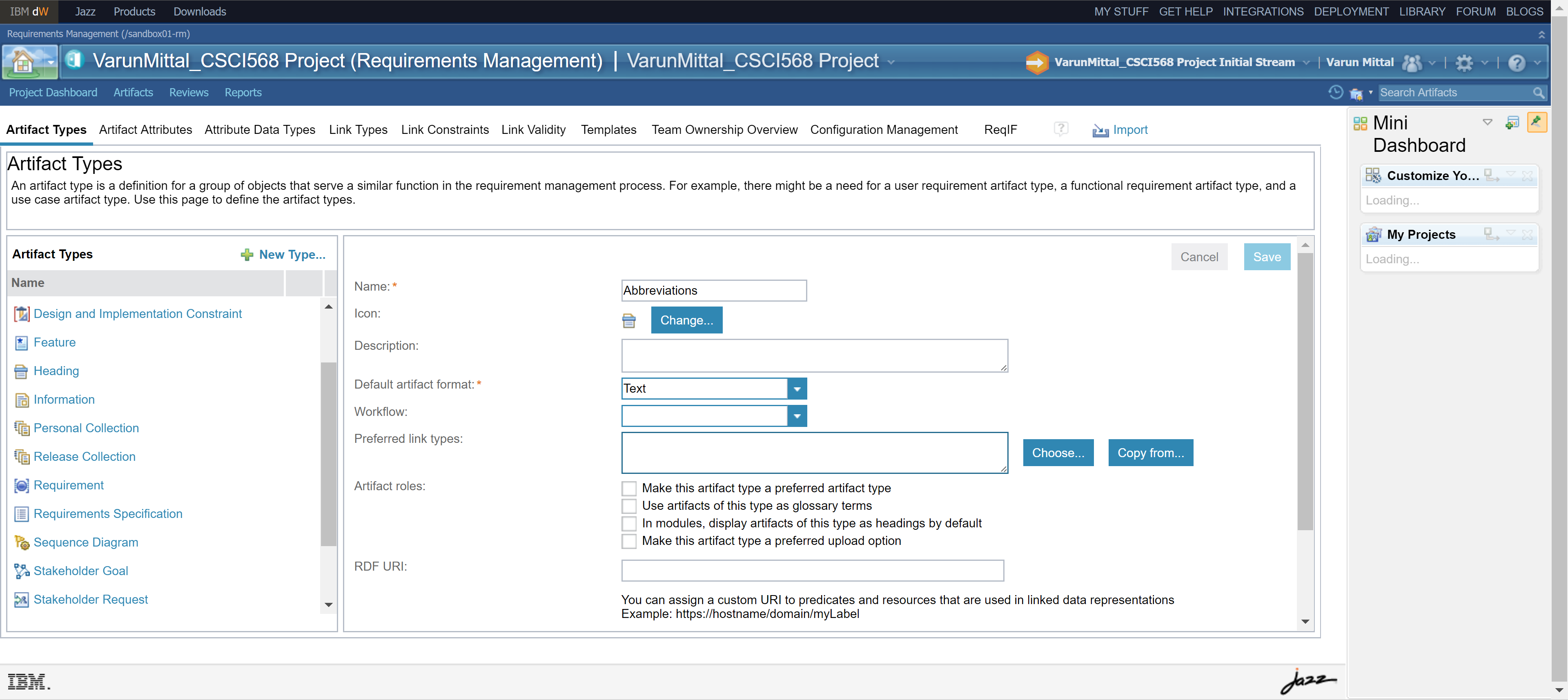
We can also implement a path planning algorithm like Greedy Best First Search, Dijkstra’s , A\*,Breadth First Search ,Depth First Search algorithms for calculating a more effective short route to reach the waypoint in a more short period of time.

1. **No checking mechanism for final waypoint**

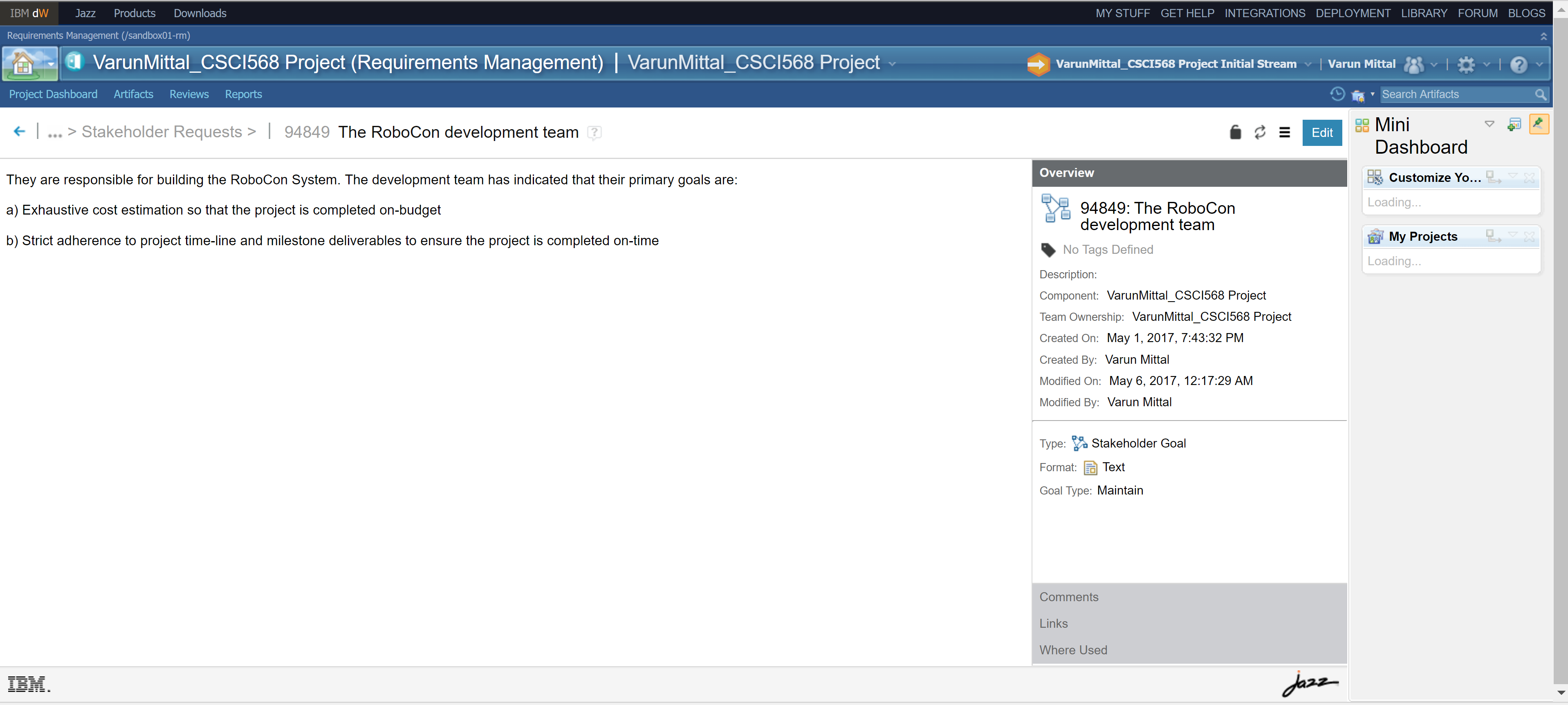
There is no mention of condition for checking if the waypoint reached is the final destination in the route and this can also lead to lot of ambiguities in the requirements as there is no clear specification for when the RoboCon Convoy will finally stop and terminate the movement mode thus saving resources and providing a more successful and comprehensive demonstration for the customers. So, we need to add this checking condition for resolving the ambiguities in the requirements.

I have given the following customized options in IBM DOORS

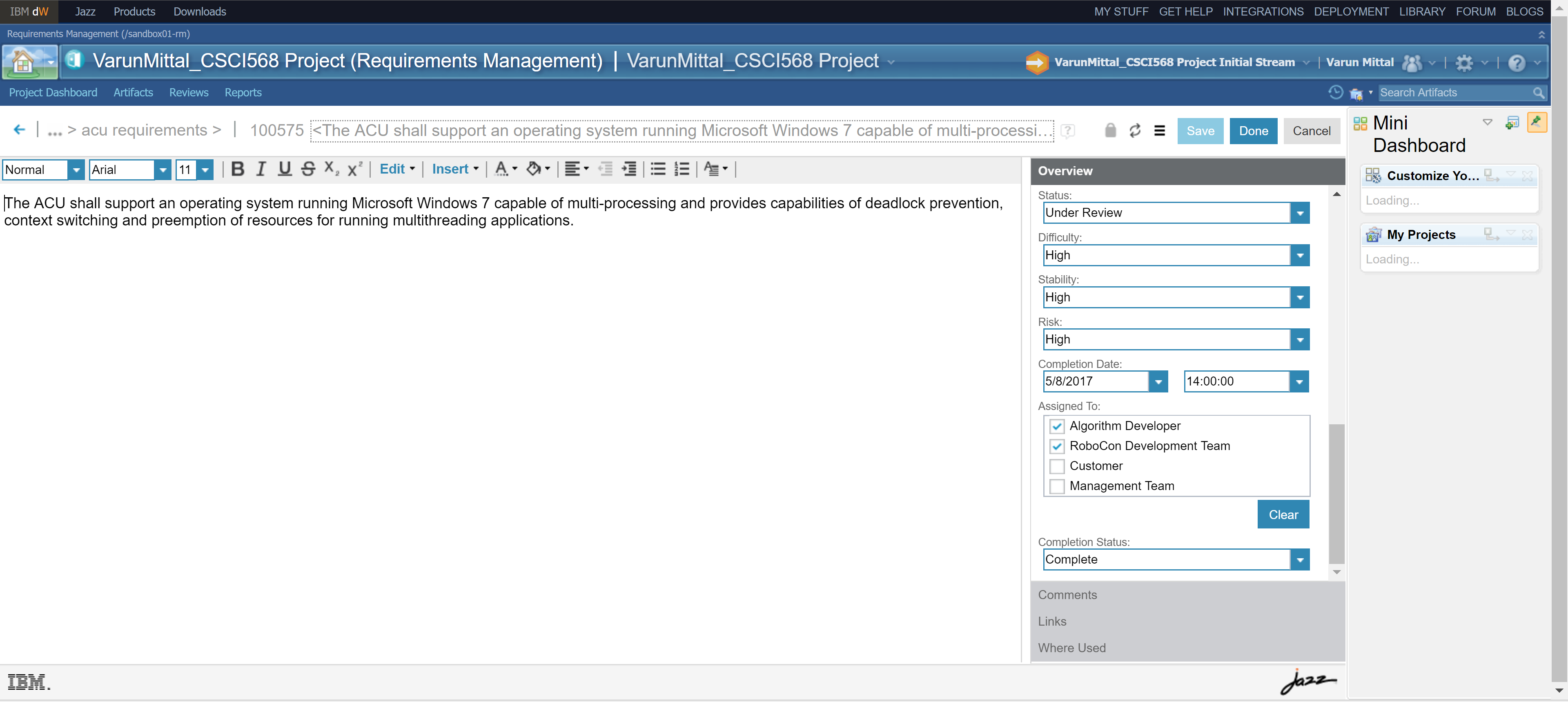
1. **Artifact Types:**
   1. Stakeholder Goal
   2. Class Diagram
   3. Use Case
   4. Activity Diagram
   5. Sequence Diagram
   6. Abbreviations
   7. Design Constraint



1. **Custom Artifact Attribute Types**
   1. **Stakeholder Goal**
      1. Goal has two values Maintain and Achieve

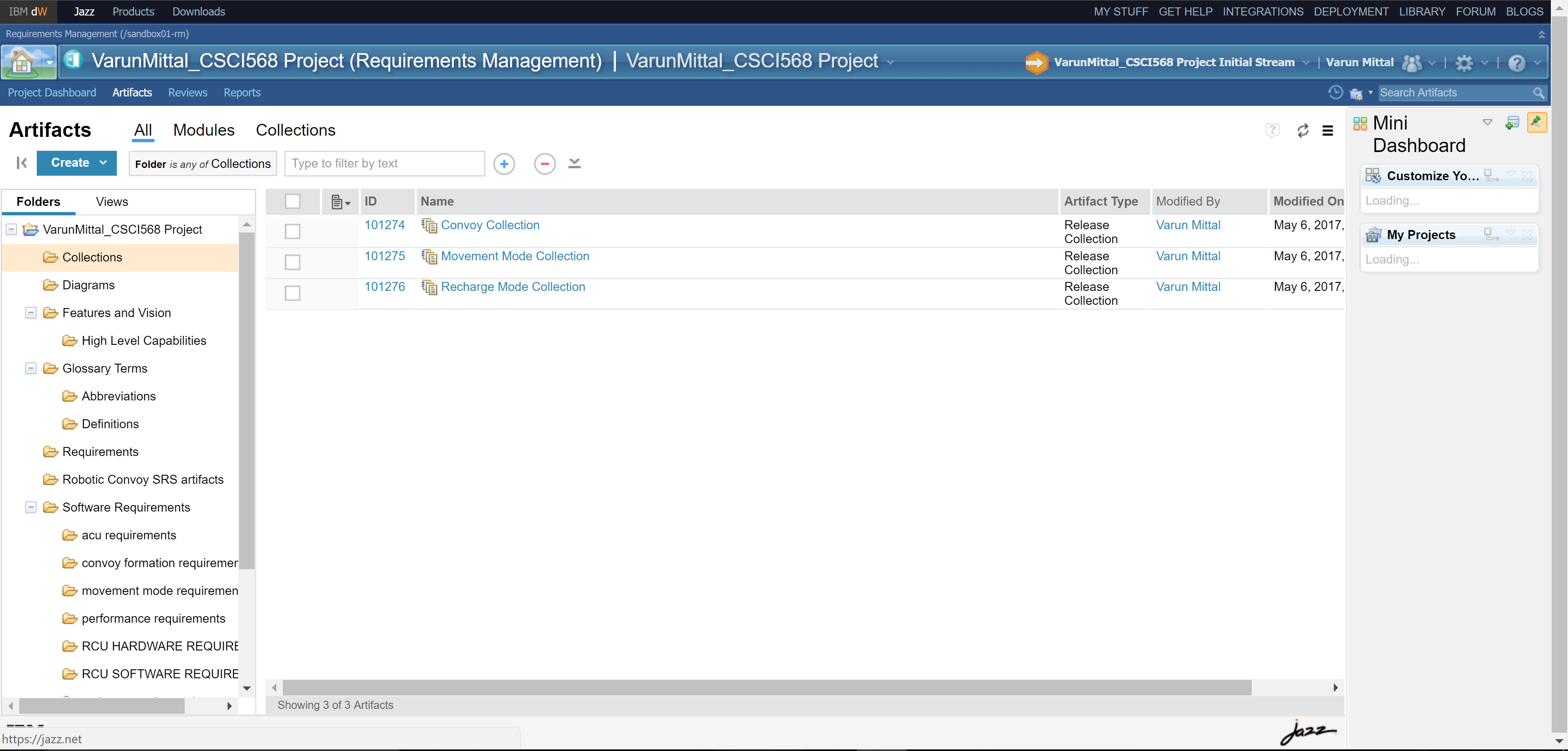


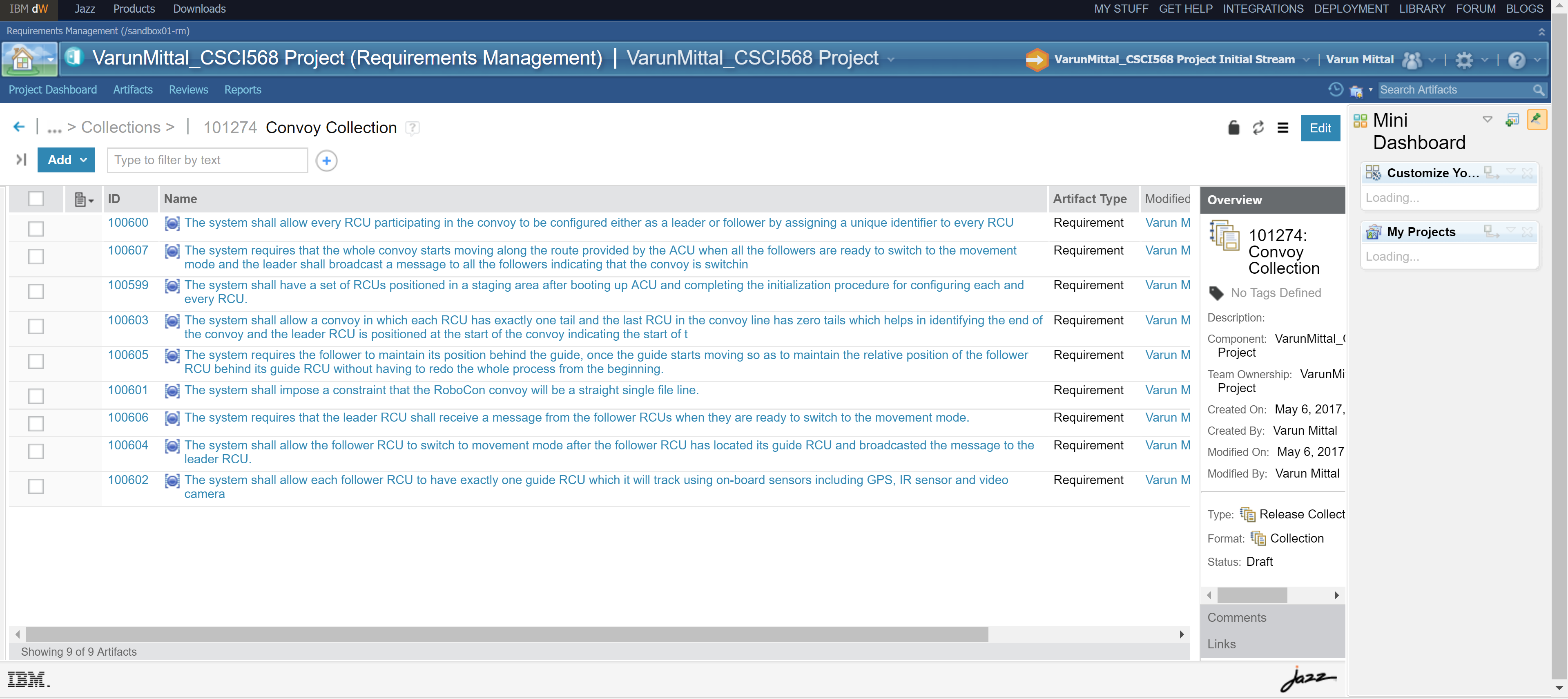
* 1. **Requirement**
     1. Completion Date with value as 8/5/2017 for stating the final deliverance of the RoboCon System
     2. Completion Status with values as Partially Complete, Completed for tracking the current state of the requirement
     3. Assigned To with values as Developer, RoboCon Development Team, Management Team, Customer for assigning responsibilities to individual stakeholders involved in the system
     4. Category with values as Portability, Security for identifying the different types of non-functional requirements



1. **Collections:**
   1. Convoy Formation Collection
   2. Movement Mode Collection
   3. Recharge Mode Collection

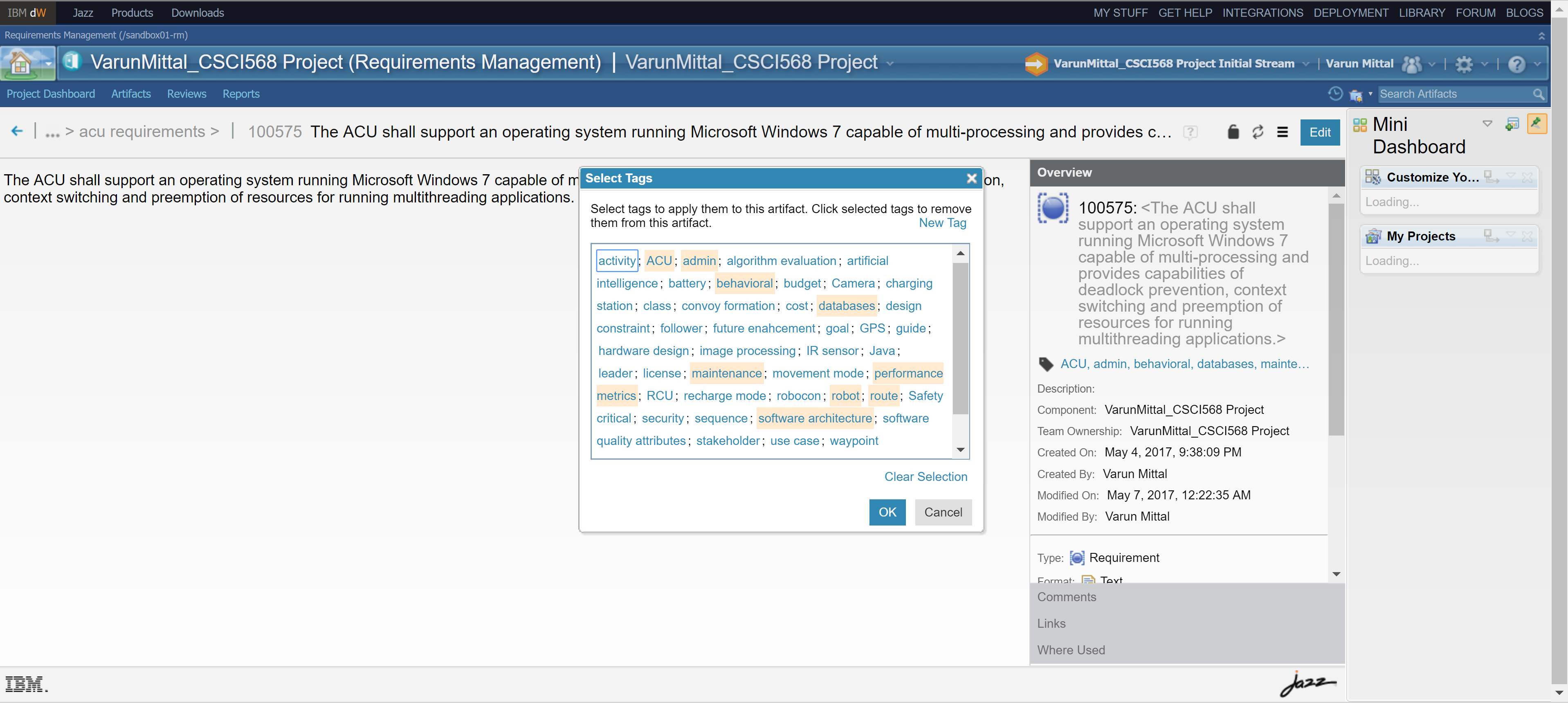
for grouping all the related requirements so that the developer can view them on the fly.



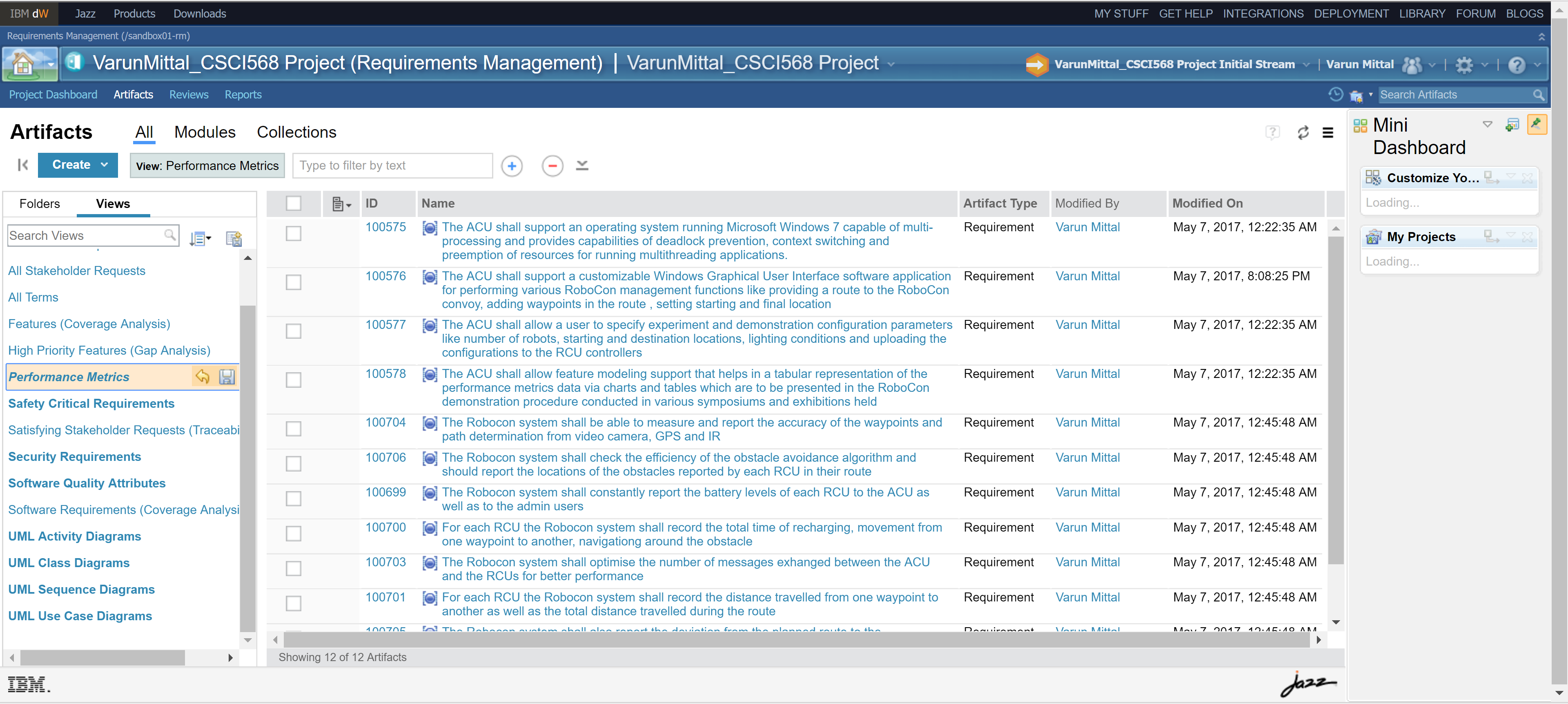


1. **Tags:**

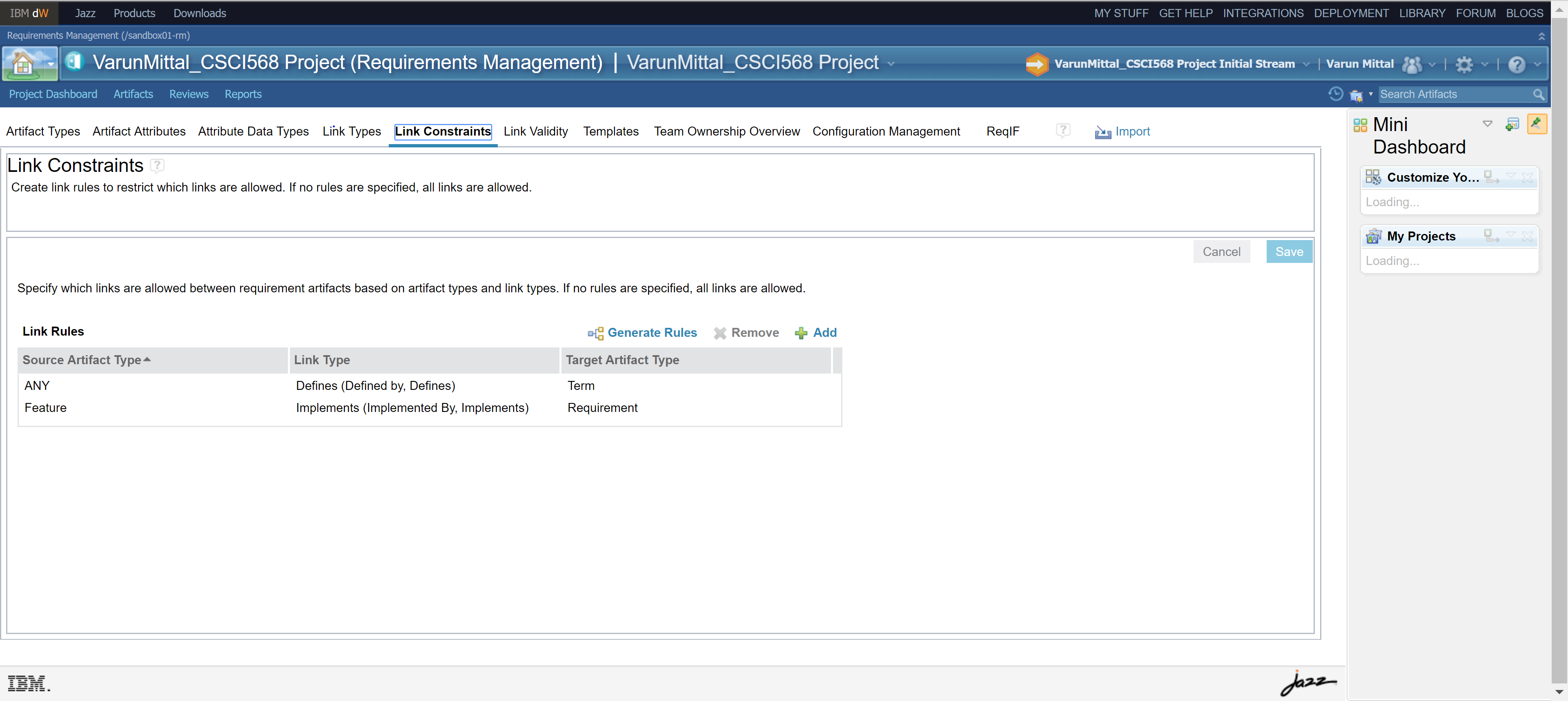
activity; ACU; admin; algorithm evaluation; artificial intelligence; battery; behavioral; budget; Camera; charging station; class; convoy formation; cost; databases; design constraint; follower; future enhancement; goal; GPS; guide; hardware design; image processing; IR sensor; Java; leader; license; maintenance; movement mode; performance metrics; RCU; recharge mode; robocon; robot; route; Safety critical; security; sequence; software architecture; software quality attributes; stakeholder; use case; waypoint



1. **Custom Views:**
   1. Performance Metrics
   2. Safety Critical Requirements
   3. Software Quality Attributes
   4. Security Requirements
   5. UML Class Diagrams
   6. UML Activity Diagrams
   7. UML Sequence Diagrams
   8. UML Use Case Diagrams



1. **Custom Link Types**
   1. Defined by/ Defines for linking the ANY type of Artifact to Glossary Term
   2. Implemented by/Implements for linking the Feature artifact to Requirement artifact



I have followed the SRS IEEE Template and organized my artifacts according to that in IBM DOORS.

I have given the Requirements for each of the components along with requirements for each of the operational modes so as to resolve the ambiguities as much as possible and clearly separated the requirements for operational modes to be implemented in separate stages.