**Integration and Unit Test Strategy**

To implement Quality Control on the Planetarium application with the use of Jira and automation testing. The goal is to build upon the first couple of sprints to look for defects within the application and provide constructive and detailed feedback on user workability and experience. In this sprint the focus is on Integration and unit testing with the use of the source code.

1. High Priority Value Tests
   1. User Registration Password is in plaintext
   2. User is unable to create planet
   3. User is unable to create moon
   4. User is unable to delete planet
   5. User Login Password in plain text
2. Risk
   1. Tests are based on priority level, severity level, and time constraints.
   2. Highest impact and highest probability of failure
   3. Be aware of time constraints
   4. Be aware of blockers that may arise i.e. QC, Interviews
3. Tools
   1. Selenium
   2. Cucumber
   3. Jira
   4. DBeaver
   5. IDE of choice
4. Data
   1. Data is validated from manual testing conducted in Sprint 1
   2. Automation process has been established in Sprint 2, build off of that
   3. Conduct data inspection to ensure data meets requirements
5. DevSecOps
   1. Source code is accessible
   2. Testing data stored in DBeaver
   3. Using Maven and dependencies that function with Maven
   4. GitHub branch rules implemented
6. Testing Environment
   1. Local environment is being used for testing
7. Testing Efficiencies
   1. Consolidate tests if testing for the same thing
   2. Plan before testing
   3. Make use of Jira board feature to organize tests
   4. Organize process before beginning
   5. Communication
   6. Peer programming
   7. Pair programing
   8. GitHub should be used for organization