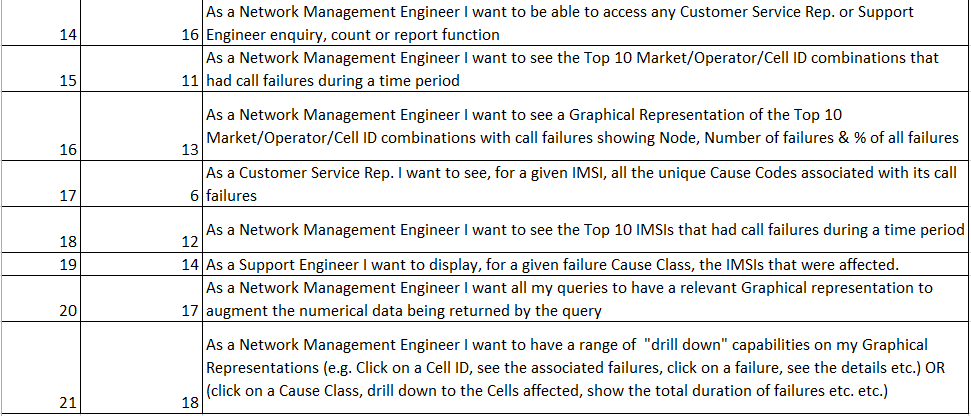
Group Project Report

# Introduction

In the following report we will outline per sprint what we planned, what we achieved and our interpretation of our progress.

# Product Backlog

# 



# 3. Sprint0

## 3.1 Organisation

Andreia Trigueiro Scrum Master

Amulya Cheluvaraja Team Member

Opeyemi Ayeni Team Member

Sean Behan Team Member

Sorcha Bruton Team Member

## 3.2 Sprint Goal

## The system administrator will be able to import data and have any data with errors, highlighted and excluded from the imports. The system administrator will be able to create log in credentials for three types of users. The customer service representative will be able to see details relating to IMSI failures.

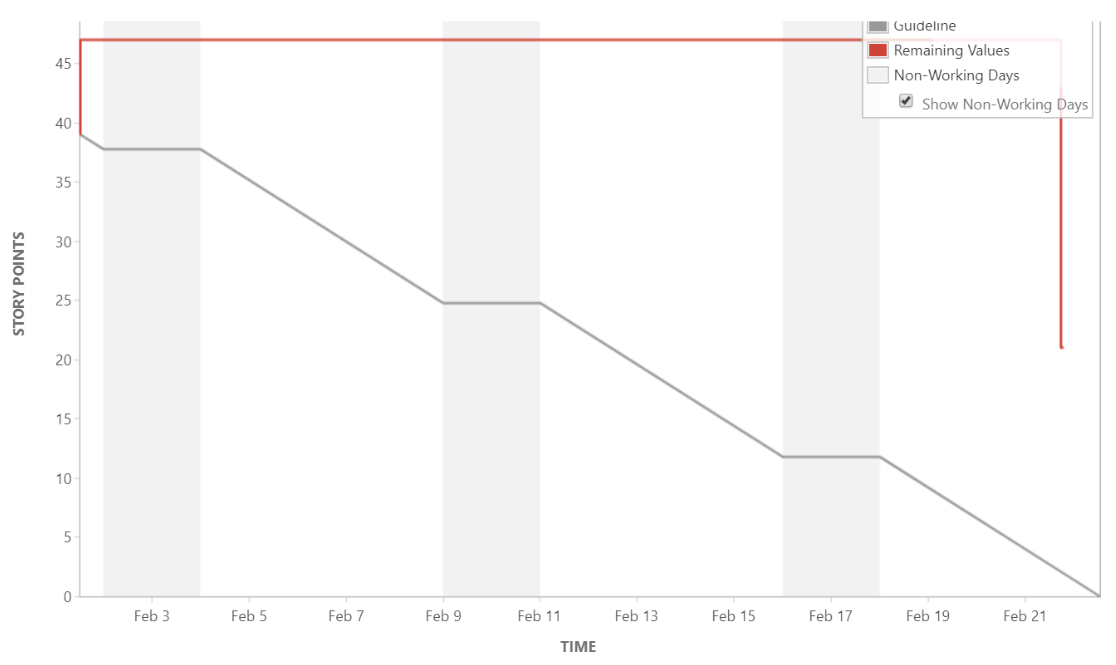
## 3.2 Sprint Plan

## [US#1] As the System Administrator I want to be able to receive and import Datasets.

## [US#3] As the System Administrator I want to have incoming records checked for consistency (i.e. valid date / time values, valid MNC / MCC combinations, valid Event IDs, Cause Codes etc.) and have erroneous records highlighted and excluded.

## [US#2] As the System Administrator I want to be able to assign an Id and password to each User type (i.e. Customer Service Rep., Support Engineer, and Network Management Engineer).

## [US#4] As Customer Service Rep. I want to display, for a given affected IMSI, the Event ID and Cause Code for any / all failures affecting that IMSI.



## 3.3 Test Strategy

## • Used Junit and arquillian for testing.

## • 80% of the code coverage includes unit and integration testing.

## • Could not implement TDD as the technologies used were quite new

## .3.4 Sprint Progress

## 

## 3.5 User Story Elaboration

|  |  |
| --- | --- |
| User Story: 1 | |
| Given | I am logged in as a system administrator and there are files to be imported |
| When | I click on the import button on the home page |
| Then | the database will be populated with this data and a success message will be displayed |

|  |  |
| --- | --- |
| User Story: 1 | |
| Given | I am logged in as a system administrator and there are no files to be imported |
| When | I click on the import button on the home page |
| Then | a message will show that there is no data to import |

|  |  |
| --- | --- |
| User Story: 3 | |
| Given | I am logged in as a system administrator and there are files to be imported which contain records with errors |
| When | I click on the import button on the home page |
| Then | the database will not be populated with the erroneous records and these records will be highlighted and excluded along with a message to reflect this. |

|  |  |
| --- | --- |
| User Story: 3 | |
| Given | I am logged in as a system administrator and there are files to be imported which contain records with no errors |
| When | I click on the import button on the home page |
| Then | there will be nothing to show in the location where erroneous records are highlighted and stored. |

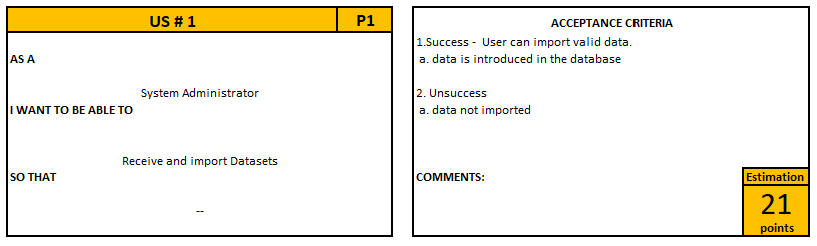
|  |  |
| --- | --- |
| User Story: 2 | |
| Given | I am logged in as a system administrator |
| When | I enter a valid user name, password and user type and click on create user |
| Then | I will have created a valid user and the database will reflect this addition. |

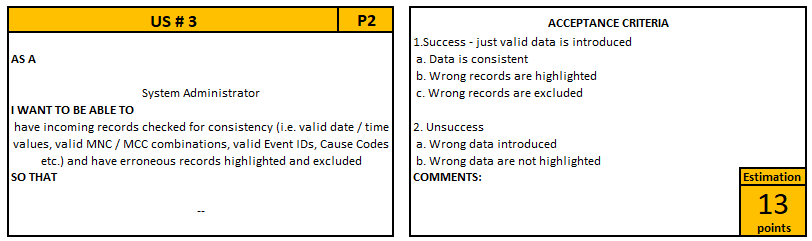
|  |  |
| --- | --- |
| User Story: 2 | |
| Given | I am logged in as a system administrator |
| When | I enter a user name already in use, a valid password and user type and click on create user |
| Then | the user will not be created as user names have to be unique, a message will show that the user name is not unique. |

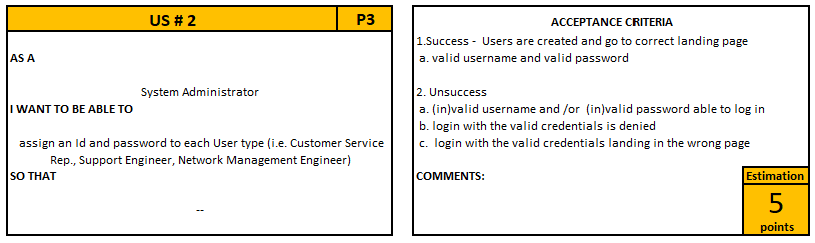
|  |  |
| --- | --- |
| User Story: 2 | |
| Given | I am logged in as a system administrator |
| When | I enter no user name, password and no user type and click on create user |
| Then | no user is created, and an error message is shown to state that all fields must be filled in. |

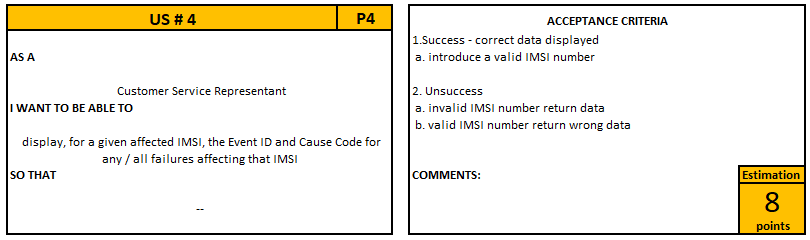
|  |  |
| --- | --- |
| User Story: 4 | |
| Given | I am logged in as a customer service rep and the database is populated with valid data |
| When | I enter an invalid imsi number and click on show records |
| Then | no records will be shown and an error message to show that it is not a valid imsi number will be shown |

|  |  |
| --- | --- |
| User Story: 4 | |
| Given | I am logged in as a customer service rep and the database is populated with valid data |
| When | I enter no imsi number and click on show records |
| Then | no records will be shown and an error message to show that a valid imsi number must be entered. |







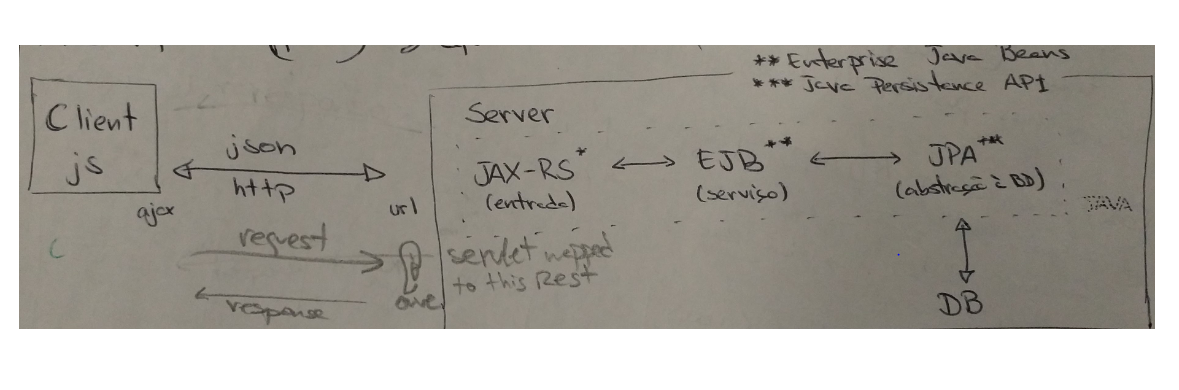


**Definition of Done:**

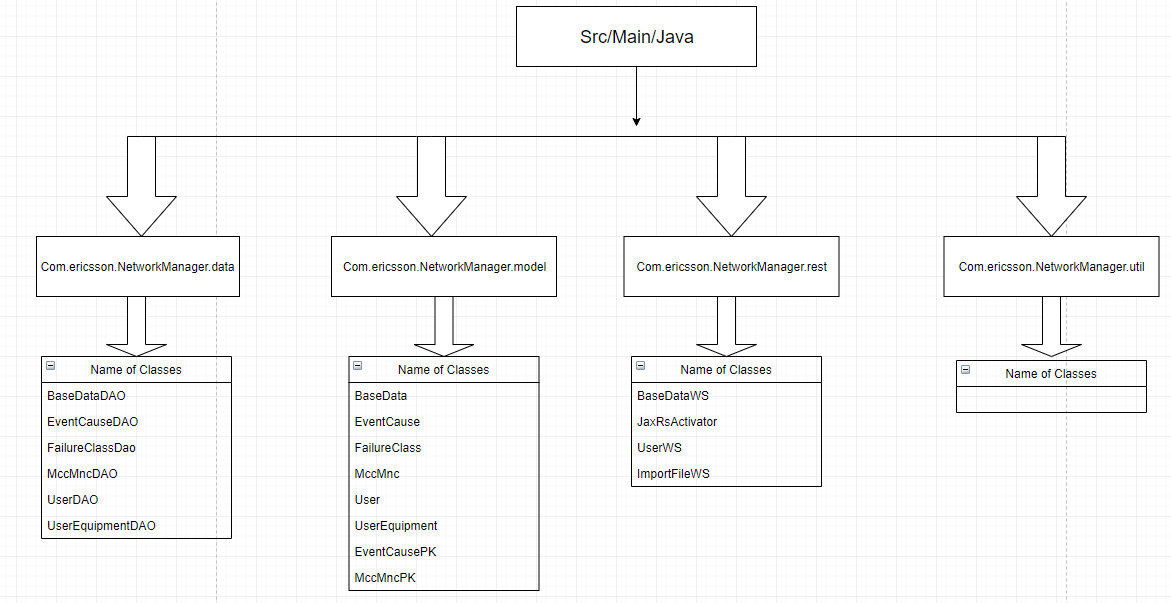
A user story will be classed as done when the code coverage is at least 80%, all relevant documentation\* is updated, code is peer reviewed and completed with 0 known bugs.

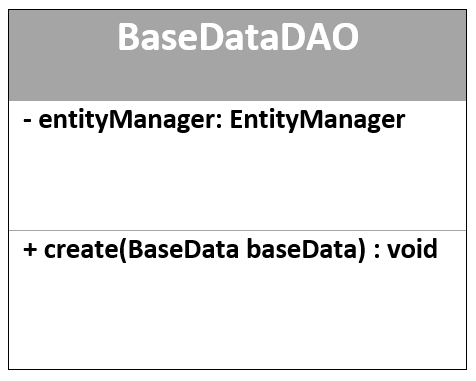
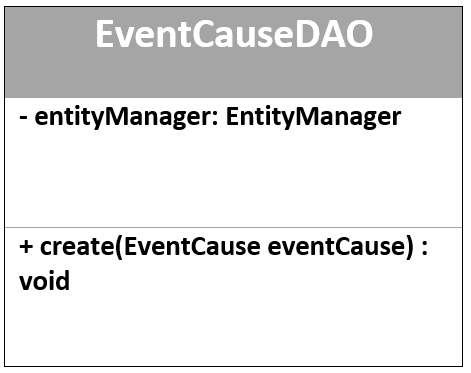
\*burn-down chart, user guide, bug report, manual tests, automatic tests, user acceptance testing, cross browser testing, PMD.

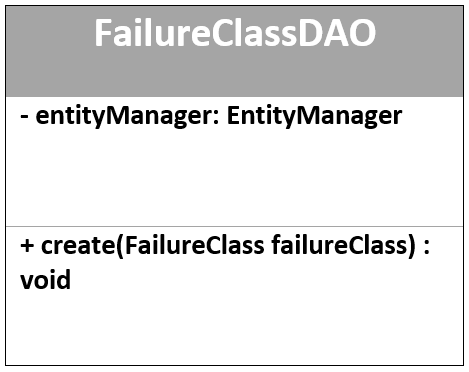
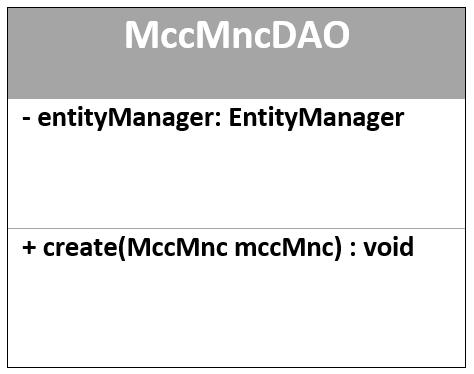
## 3.6 System Architecture

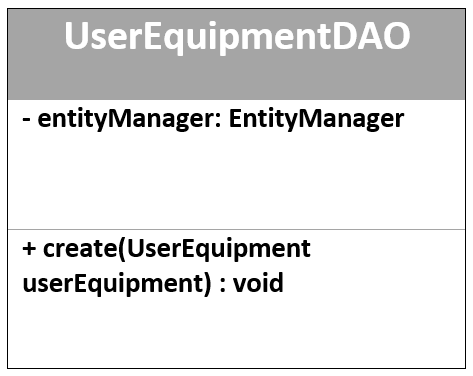
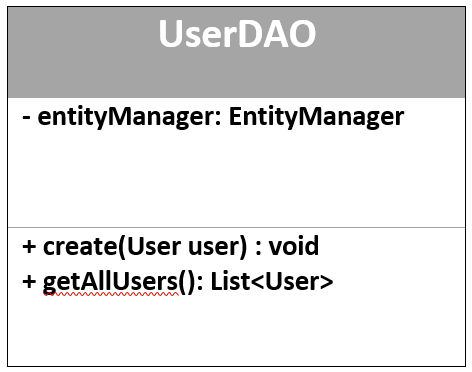


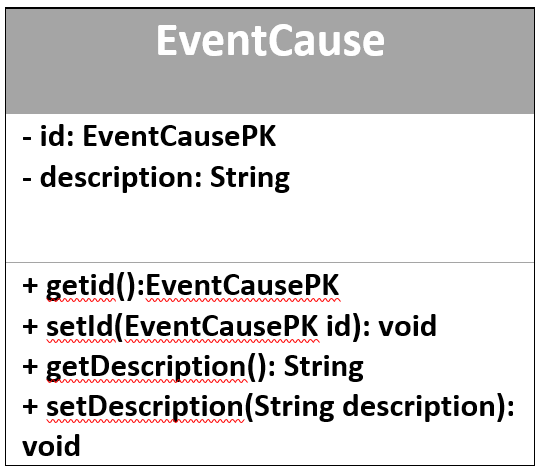
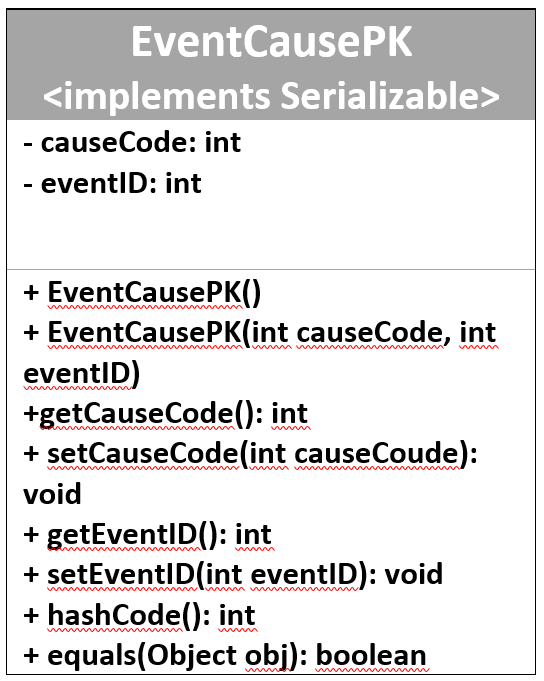
**UML package and Class diagrams:**

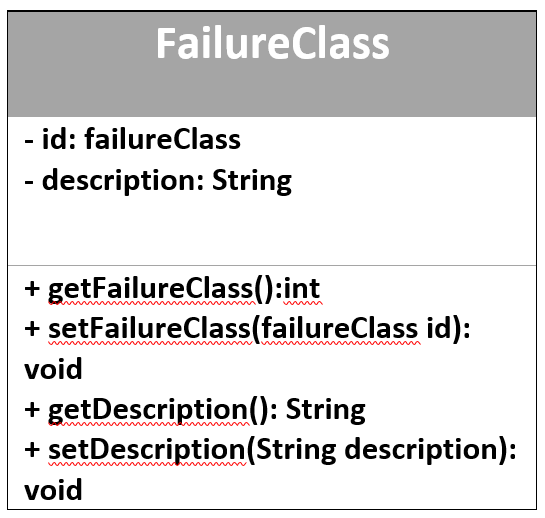
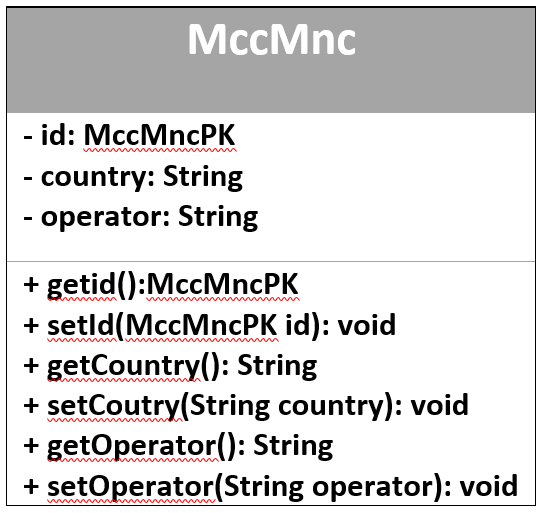


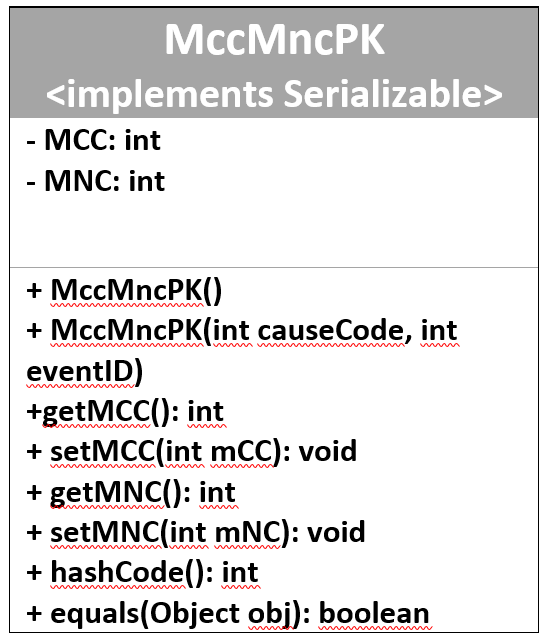
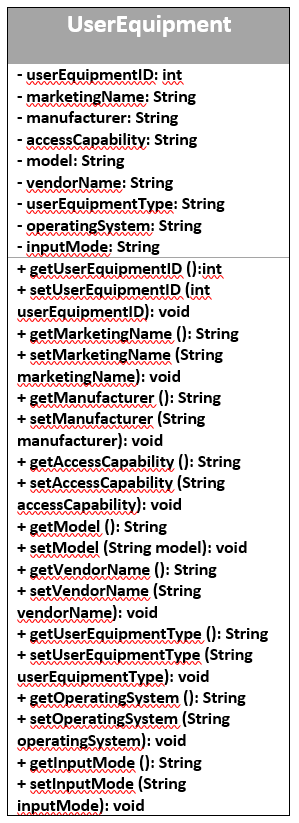
 

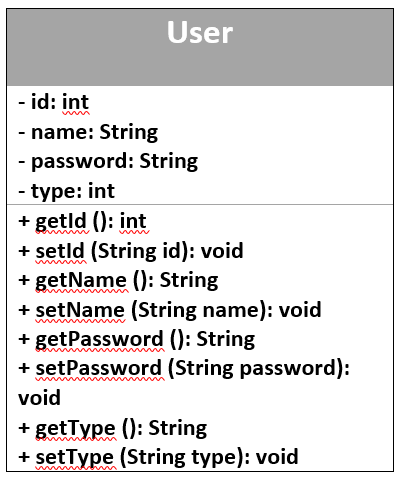
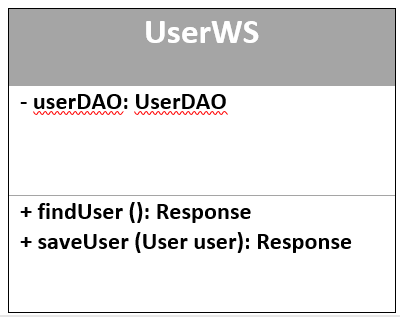
 

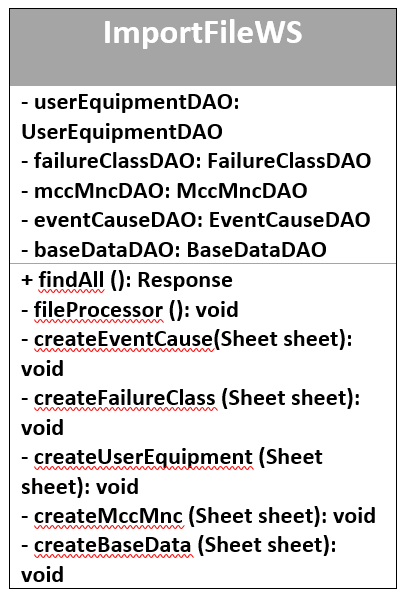
 

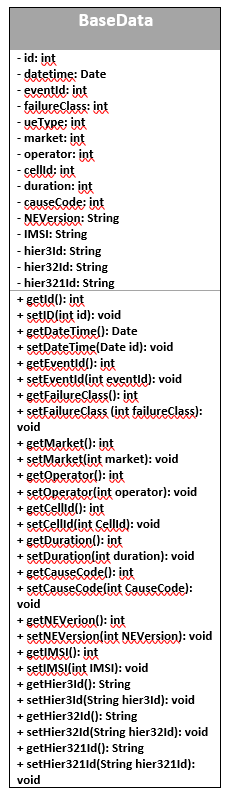
 

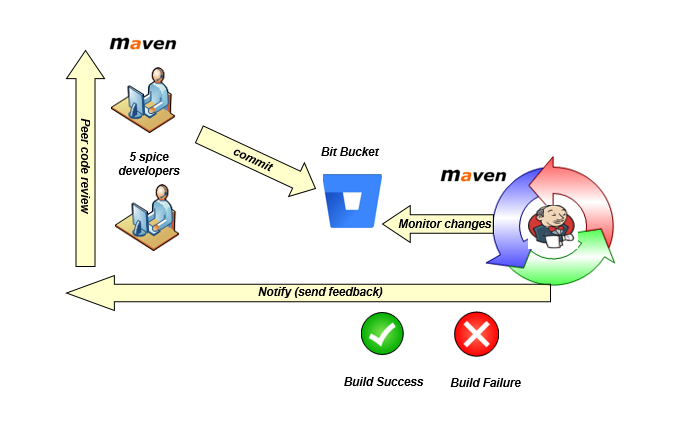
 

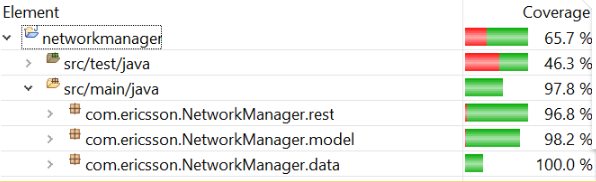


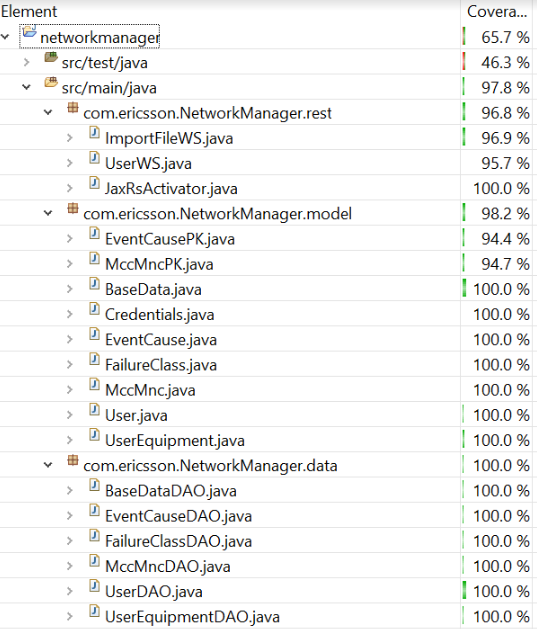
## 3.7 Development Environment

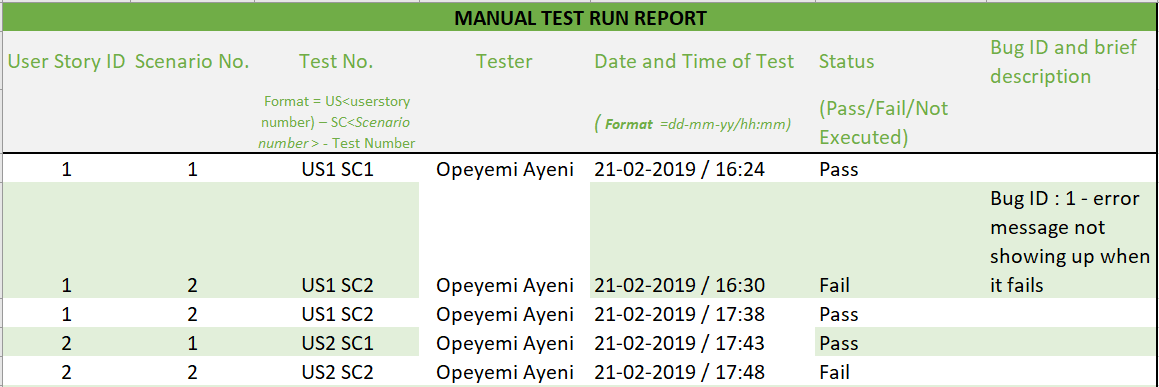


The Development Environment that we used for this sprint was to create our project on a Maven Project which is a project management tool which encompasses a project object model, a set of standards a project cycle and dependency management systems. This maven project was also connected to Bitbucket which is a web-based software project repository hosting service. Where we were able to upload and store working software project folders to the service. Bitbucket allows us to do version control, build control and collaboration. Finally, our Bitbucket was connected to Jenkins which is a build scheduling tool, used for continuous integration which allows us to schedule builds and tests for our code as they are uploaded to Bitbucket, reporting back weather each stage of the tests have passed or failed.

## 3.7 Test Results







## 3.8 Team Decision Log

|  |  |
| --- | --- |
| DATE | DESCRIPTION |
| 31.Jan.2019 | Although the MCC-MNC table is not normalized (repetitions) the team decided for now leave like it we receive |
| 31.Jan.2019 | The user table will be created in the same database than data\_log data |
| 13.Feb.2019 | The team decided to do the code review in person instead of using bitbucket |
| 14.Feb.2019 | The team decided that small decisions such as naming classes etc do not need to be logged in the team decision log. |
| 21.Feb.2019 | The team decided that to meet our definition of done, we will concentrate on achieving our agreed level of code coverage by using junit testing. Arquillian will be used in the next sprint. The WS classes will be tested manually until arquillian is implemented. |

## 3.9 Sprint Retrospective

|  |  |  |
| --- | --- | --- |
| SPRINT #0 RETROSPECTIVE | | |
| **WHAT WENT GOOD?** | **WHAT WASN'T SO GOOD?** | **WHAT WE CAN IMPROVE?** |
| Ericsson Guidelines and checklist preperation work very helpful and good to have. Will keep for the next sprints. | Project set up could have been improved: Arquillian, Jenkins, Maven | Check Jenkins more frequently, needs more attention consistently |
| The good communication and coordination between team members | Tuesdays and Thursdays, no set time for scrum meetings due to classes. | Have also a physical board with user stories and points as a physical reminder for us. |
| The communication between our team and the other teams. | The code review needs to be done more frequently and consistently. | Use PMD as we are going and not just in the testing phase. |
| Friendly enviornment to work in, everyne within the team is comfortable and happy to help each other. | The merging of work with the branches on bitbucket, issues with conflicting code and not merging correctly. | Scrum meeting times defined consistently and will start even if people are no on site at the time. |
|  |  | If we cannot come to college on Friday, do a Scrum meeting on Thursday evening |
|  |  | Do pull and push to bit bucket, with another team member. Any conflicts, talk to the team member who commited the conflicting code and solve the merge conflicts |
|  |  | Merge to the master every evening after code review approval and pull from the master branch each morning. |

## 3.10 Evaluation and Conclusion

For this sprint we unfortunately did not meet our Sprint Goal for one user story. In our sprint goal we specified that “The system administrator will be able to import data and have any data with errors, highlighted and excluded from the imports.” While we achieved the import aspect, we fell short on the validation aspect, meaning we could not say that we achieved our goal for this user story.

This also means that we did not deliver on all the user stories that we committed to, namely the user story “As the System Administrator I want to have incoming records checked for consistency (i.e. valid date / time values, valid MNC / MCC combinations, valid Event IDs, Cause Codes etc.) and have erroneous records highlighted and excluded” was not delivered.

This user story was almost complete but we were having some issues where there were errors in attempting to validate if data was of the correct data type and then redirecting the record to the error records table in our database. It was performing correctly in that the erroneous record was not being introduced to the database but it was not performing the next part correctly in that it was not highlighting and excluding it as stated in the story. Our idea to highlight and exclude, is to put all the erroneous records into a table in the database and create a view of this table on the web page- there by completing the user story.

We don’t feel that we are carrying any technical debt into the Sprint 1. We have returned user story #3 to the product backlog. We feel we have quite a bit of work complete on this so it should be achievable in the next sprint.

# 4. Sprint1

## 4.1 Organisation

Andreia Trigueiro Team Member

Amulya Cheluvaraja Team Member

Opeyemi Ayeni Team Member

Sean Behan Scrum Master

Sorcha Bruton Team Member

## 4.2 Sprint Goal

The system administrator will be able to get any data with errors, highlighted and excluded from the imports. The customer service representative will be able to see details relating to IMSI failures. The support engineer will be able to view a list of all IMSIs with call failures during a given time period

## 4.2 Sprint Plan

## [US#3] As the System Administrator I want to have incoming records checked for consistency (i.e. valid date / time values, valid MNC / MCC combinations, valid Event IDs, Cause Codes etc.) and have erroneous records highlighted and excluded.

## [US#4] As Customer Service Rep. I want to display, for a given affected IMSI, the Event ID and Cause Code for any / all failures affecting that IMSI.

## [US#7] As a Support Engineer I want to see a list of all IMSIs with call failures during a given time period.

## [US#22] As a Team Developer we want to implement all the changes requested by the customer in sprint 0, so that they will be satisfied with the product.

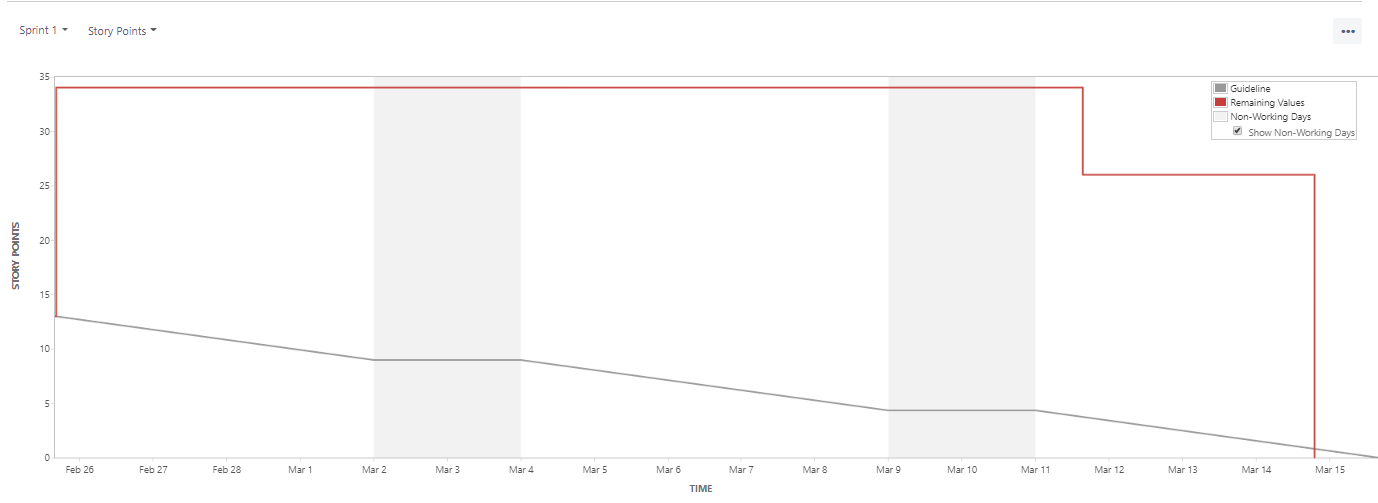


Figure 1- Burn down Chart

## 4.3 Test Strategy

* Junit and arquillian for testing.
* 70% of the code coverage includes unit and integration testing.
* Manual testing
* Regression testing- rerun all tests throughout the day to ensure all are passing.

## .4.4 Sprint Progress

## 

## 4.5 User Story Elaboration

|  |  |
| --- | --- |
| User Story: 3 – scenario 1 | |
| Given | I am logged in as a system administrator and there are files to be imported which contain records with errors |
| When | I click on the import button on the home page |
| Then | the database will not be populated with the erroneous records and these records will be highlighted and excluded along with a message to reflect this. |

|  |  |
| --- | --- |
| User Story: 3 – scenario 2 | |
| Given | I am logged in as a system administrator and there are files to be imported which contain records with no errors |
| When | I click on the import button on the home page |
| Then | there will be nothing to show in the location where erroneous records are highlighted and stored. |

|  |  |
| --- | --- |
| User Story: 4 – scenario 1 | |
| Given | I am logged in as a customer service rep and the database is populated with valid data |
| When | I click on the IMSI tab and enter a valid IMSI number |
| Then | The event id, cause code and description for failures affecting that IMSI will be displayed. |

|  |  |
| --- | --- |
| User Story: 4 – scenario 2 | |
| Given | I am logged in as a customer service rep and the database is populated with valid data |
| When | I click on the IMSI tab |
| Then | No records will be shown and an error message to show that there are no matching records will be shown. |

|  |  |
| --- | --- |
| User Story: 7 – scenario 1 | |
| Given | I am logged in as a support engineer and the database is populated with valid data |
| When | I click on the call failure tab and select a start date/time and an end date/time and click the search button |
| Then | A table will be appear with the date time, IMSI, failure class and failure description that occurred between the two times. |

|  |  |
| --- | --- |
| User Story: 7 – scenario 2 | |
| Given | I am logged in as a support engineer and the database is populated with valid data |
| When | I click on the call failure tab an end date/time but no start date/time and click the search button |
| Then | The table will not be populated and no records will be shown. |

|  |  |
| --- | --- |
| User Story: 22 – scenario 1 | |
| Given | I am logged in as a System Administrator and I am creating a user with valid data |
| When | I want to choose a user type |
| Then | I have a dropdown list that contains the different user types |

|  |  |
| --- | --- |
| User Story: 22 – scenario 2 | |
| Given | I am logged in as a System Administrator and I am creating a user with valid data |
| When | I click the submit button |
| Then | The data is introduced in the database, the form is cleared for the next time the form is opened, and a success message is shown. |

|  |  |
| --- | --- |
| User Story: 22 – scenario 3 | |
| Given | I am logged in as a System Administrator and I am creating a user |
| When | I click submit with a password that does not contain an uppercase Letter an lowercase letter and at least 1 number |
| Then | I will be shown an error message and user will not be added |

|  |  |
| --- | --- |
| User Story: 22 – scenario 4 | |
| Given | I am logged in as a System Administrator and I am creating a user |
| When | I click submit with an empty username field |
| Then | I will be shown an error message and user will not be added |

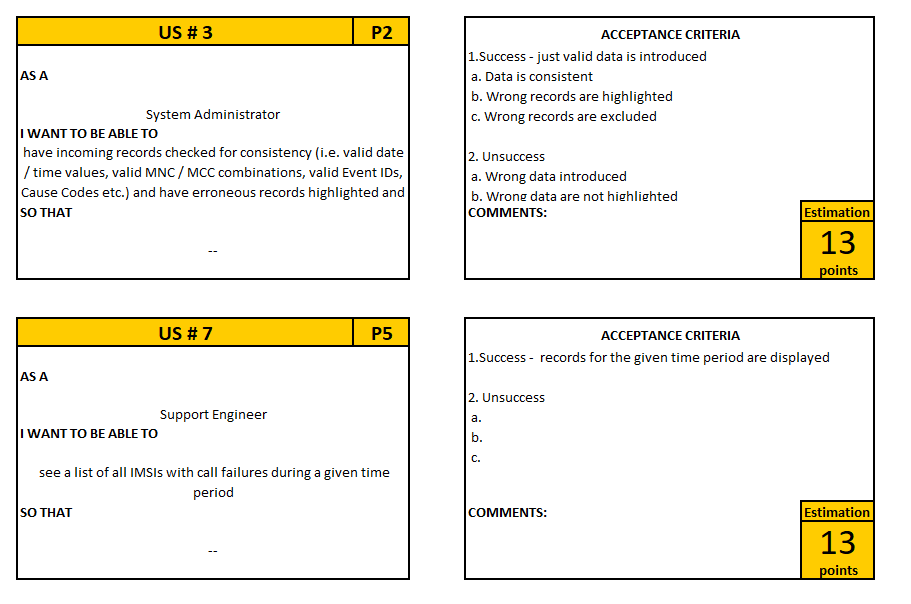
|  |  |
| --- | --- |
| User Story: 22 – scenario 5 | |
| Given | I am logged in as a System Administrator and I am creating a user |
| When | I click submit with an empty confirm password field |
| Then | I will be shown an error message and user will not be added |

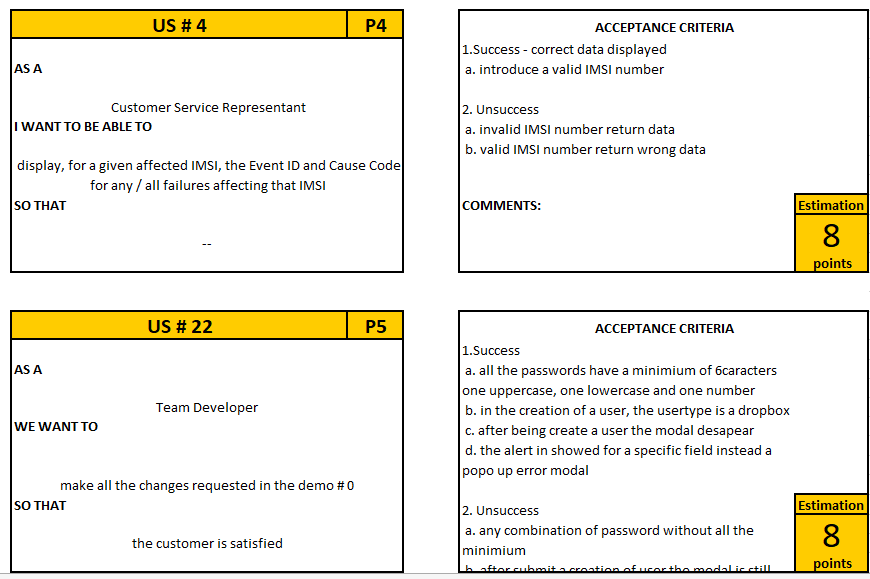
|  |  |
| --- | --- |
| User Story: 22 – scenario 6 | |
| Given | I am logged in as a System Administrator and I am creating a user |
| When | I click submit with password and confirm password fields not matching |
| Then | I will be shown an error message and user will not be added |

|  |  |
| --- | --- |
| User Story: 22 – scenario 7 | |
| Given | I am logged in as a System Administrator and I am creating a user |
| When | I click submit with all the fields filled in correctly |
| Then | The user will be created and the form will clear its contents. |

|  |  |
| --- | --- |
| User Story: 22 – scenario 8 | |
| Given | I am located at the login page |
| When | I enter a valid user name and password |
| Then | I will be successfully logged in and land on the correct landing page for my user type |

|  |  |
| --- | --- |
| User Story: 22 – scenario 9 | |
| Given | I am located at the login page |
| When | I enter an invalid user name or invalid password |
| Then | I will be shown an error message and will not be logged in. |

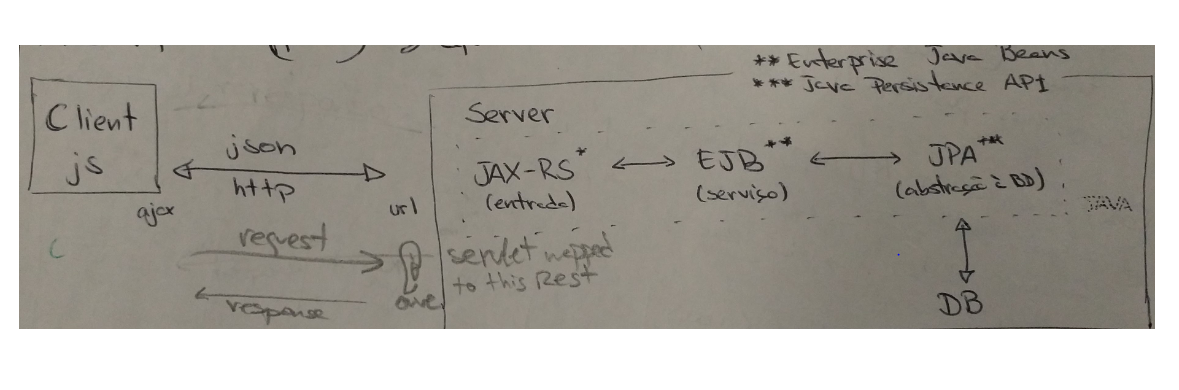


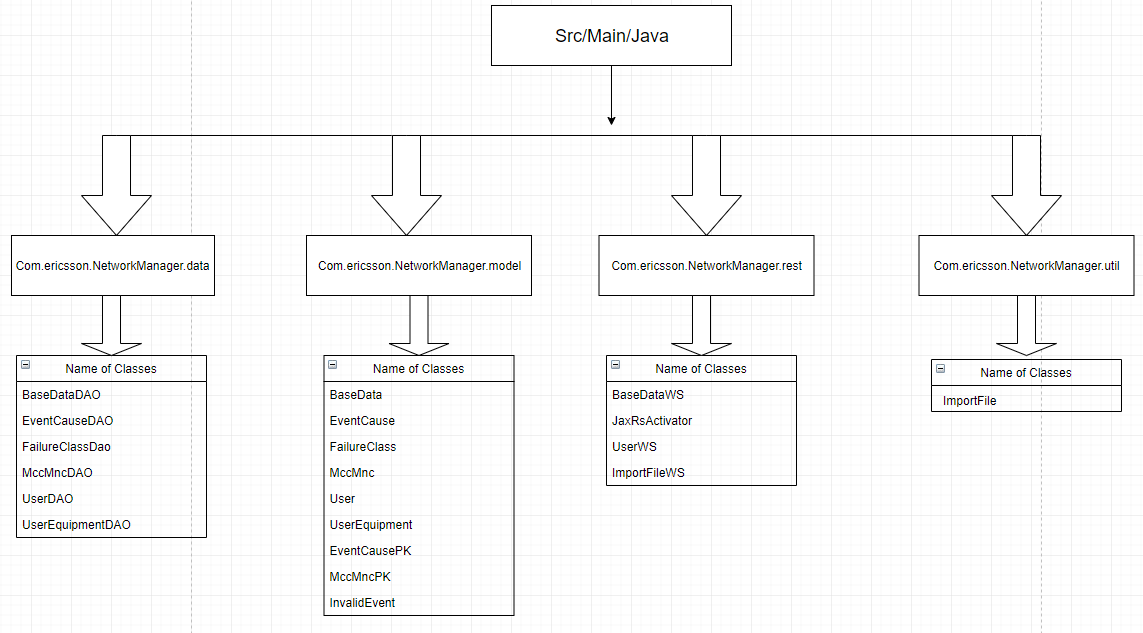


**Definition of Done:**

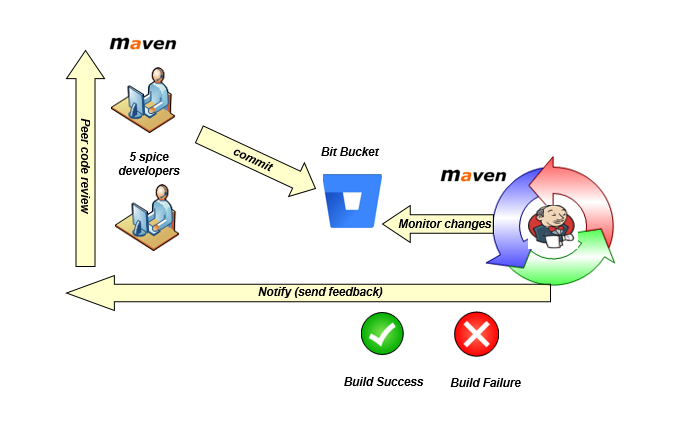
A user story will be classed as done when the code coverage is at least 70%, all relevant documentation\* is updated, code is peer reviewed and completed with 0 known bugs. \*burn-down chart, user guide, bug report, manual tests, automatic tests, user acceptance testing, cross browser testing, PMD.

## 4.6 System Architecture



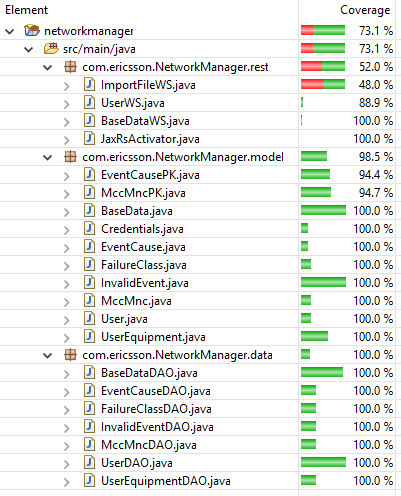
**UML package and Class diagrams:**

## 4.7 Development Environment



The Development Environment that we used for this sprint was to create our project on a Maven Project which is a project management tool which encompasses a project object model, a set of standards a project cycle and dependency management systems. This maven project was also connected to Bitbucket which is a web-based software project repository hosting service. Where we were able to upload and store working software project folders to the service. Bitbucket allows us to do version control, build control and collaboration. Finally, our Bitbucket was connected to Jenkins which is a build scheduling tool, used for continuous integration which allows us to schedule builds and tests for our code as they are uploaded to Bitbucket, reporting back weather each stage of the tests have passed or failed.

## 4.7 Test Results



|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **MANUAL TEST RUN REPORT** | | | | | | |
| **User Story ID** | **Scenario No.** | **Test No.** | **Tester** | **Date and Time of Test** | **Status** | **Bug ID and brief description** |
| 22 | 1 | US22 SC1 | Andreia Trigueiro | 11-03-2019 / 16h18 | PASS | -- |
| 22 | 2 | US22 SC2 | Andreia Trigueiro | 11-03-2019 / 16h22 | PASS | -- |
| 22 | 9 | US22 SC9 | Sorcha Bruton | 11/03/2019 12h45 | FAIL | US22 SC9 Login granted with unregistered user details |
| 22 | 9 | US22 SC9.1 | Sorcha Bruton | 11/03/2019 13h35 | PASS | -- |
| 3 | 1 | US 3 SC1 | Sean Behan | 12/03/2019 15h30 | PASS | -- |
| 22 | 8 | US 22 SC 8 | Sorcha Bruton | 11/03/2019 12h45 | PASS | -- |
| 3 | 2 | US 3 SC 2 | Sean Behan | 12/03/2019 15h30 | FAIL | US3 SC2 error message appearing and then disappearing |
| 3 | 2 | US 3 SC2.1 | Sean Behan | 12/03/2019 15h45 | PASS | -- |
| 7 | 1 | US 7 SC1 | Andreia Trigueiro | 12/03/2019 12h40 | PASS | -- |
| 7 | 2 | US 7 SC2 | Andreia Trigueiro | 12/03/2019 12h45 | PASS | -- |
| 22 | 3 | US22 SC3 | Opeyemi Ayeni | 13/03/2019 / 13h11 | PASS | -- |
| 22 | 4 | US22 SC4 | Opeyemi Ayeni | 13/03/2019 / 13h14 | PASS | -- |
| 22 | 5 | US22 SC5 | Opeyemi Ayeni | 13/03/2019 / 13h16 | FAIL | US 22 SC5 User added without confirming password |
| 22 | 5 | US22 SC5 | Opeyemi Ayeni | 13/03/2019 / 13h14 | FAIL | US 22 SC5.1 User password confirmed but user not added to the database |
| 22 | 5 | US22 SC5.1 | Opeyemi Ayeni | 13/03/2019 / 13h16 | PASS | -- |
| 22 | 6 | US22 SC6 | Opeyemi Ayeni | 13/03/2019 / 13h19 | PASS | -- |
| 22 | 7 | US22 SC7 | Opeyemi Ayeni | 13/03/2019 / 13h122 | PASS | -- |
| 4 | 1 | US 4 SC1 | Amulya Cheluvaraja | 13/03/2019 15h05 | FAIL | US 4 SC1 Enter a valid imsi and description column is empty |
| 4 | 1 | US 4 SC1.1 | Amulya Cheluvaraja | 13/03/2019 15h10 | PASS | -- |
| 4 | 2 | US 4 SC2 | Amulya Cheluvaraja | 13/03/2019 15h05 | FAIL | US 4 SC2 Enter an invalid imsi and no error message showing |
| 4 | 2 | US 4 SC2.1 | Amulya Cheluvaraja | 13/03/2019 15h10 | PASS | -- |
|  |  |  |  |  |  |  |

## 4.8 Team Decision Log

|  |  |
| --- | --- |
| **DATE** | **DECISION** |
| 12 March | The team decided to change the Definition of Done by bringing down the code coverage to 70% as it was difficult to mock the apache poi while testing the validation classes. |

## 4.9 Sprint Retrospective

|  |  |  |
| --- | --- | --- |
| **SPRINT #1 RETROSPECTIVE** | | |
| **WHAT WENT GOOD?** | **WHAT WASN'T SO GOOD?** | **WHAT WE CAN IMPROVE?** |
| Less stress amongst the team during this sprint due to having gotten the project setup out of the way | Some issues encountered with Arquillian testing | Improve timekeeping for sprint preparation |
| Instructions were clearer this time around than in the previou sprint | Stories could have been completed earlier in the sprint | improve on our documentation |
| Jira was improved and was easier to work with | Could have planned around the exams better. | Improve the planning for our tests (get them done earlier) |
| Improvement in team work and Scrum meetings were better organized |  |  |

## 4.10 Evaluation and Conclusion

We are delighted to say we achieved our Sprint goal for this sprint. We managed to complete user story #3 which we failed to deliver on in Sprint 0. We also delivered on the other user stories we committed to meaning no stories were returned to the product backlog. In terms of technical debt, we had set our definition of done to include code coverage of 80%. We realised that this was not achievable and made a team decision to change our definition of done to 70% code coverage. The reason we were not able to achieve it is that we are using Apache Poi in our import process. In order to test this, we would need to use Power Mock which is a testing framework we are not familiar with. We were unable to test it with easy mock or mockito as it is static. To compensate we performed addition manual tests which we are aware we will need to automate in Sprint 2, so as not to carry more technical debt. This is the only aspect that we feel we are carrying some technical debt and this was also highlighted in the feedback we received from our demo.

# 5. Sprint2

## 5.1 Organisation

Andreia Trigueiro Team Member

Amulya Cheluvaraja Team Member

Opeyemi Ayeni Team Member

Sean Behan Team Member

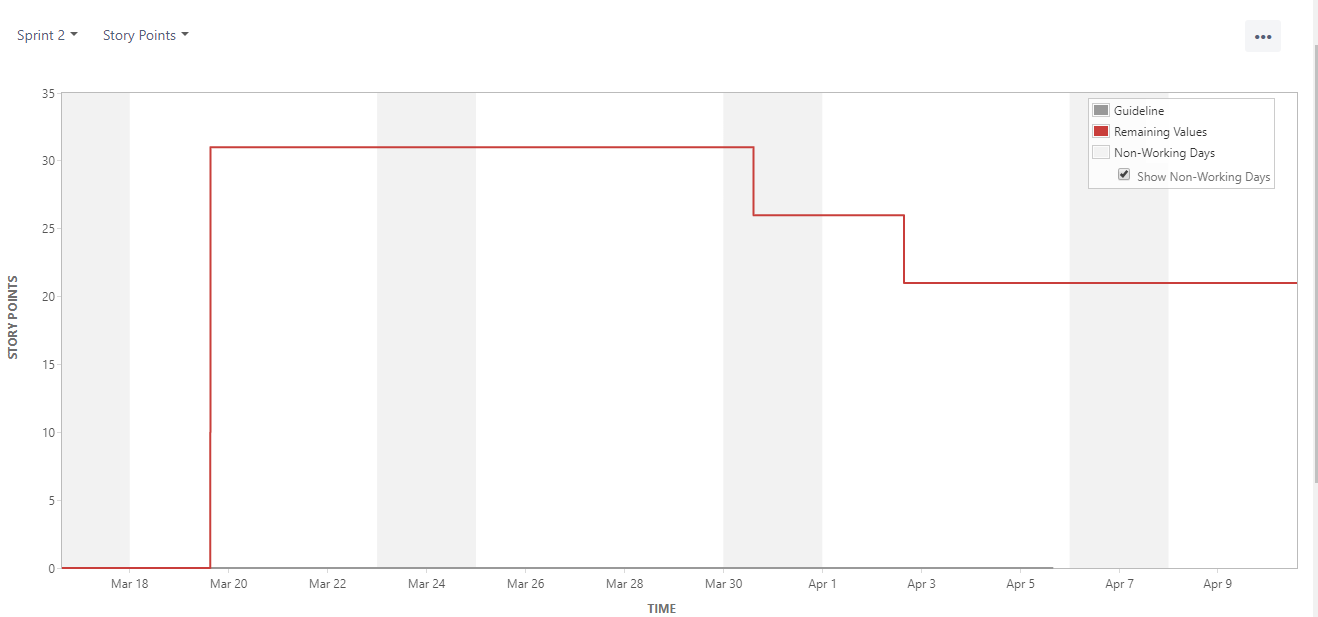
Sorcha Bruton Scrum Master

## 5.2 Sprint Goal

The support engineer will be able to search by phone model for the total number of call failures in a given time period. The network manager will be able to see the total number of call failures and the total duration, for a given IMSI, during a given time period. The network manager will be able to see the number of occurrences of, unique event ID and cause code combinations for a given phone model.

## 5.2 Sprint Plan

* The support engineer will be able to search by phone model for the total number of call failures in a given time period.
* The network manager will be able to see the total number of call failures and the total duration, for a given IMSI, during a given time period.
* The network manager will be able to see the number of occurrences of, unique event ID and cause code combinations for a given phone model.
* Implement all feedback given during sprint 1.



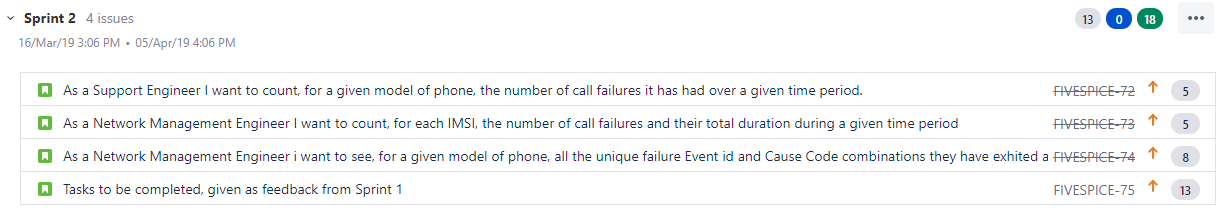
## 5.3 Test Strategy

* Junit and arquillian for testing.
* 70% of the code coverage includes unit and integration testing.
* Manual testing
* Regression testing- rerun all tests throughout the day to ensure all are passing.

## 5.4 Sprint Progress

## 

Sprint 2 Backlog



## 5.5 User Story Elaboration

**Definition of done:**

A user story will be classed as done when the code coverage is at least 70%, all relevant documentation\* is updated, code is peer reviewed and completed with 0 known bugs.

|  |  |
| --- | --- |
| User Story: 8  – scenario 1 | |
| Given | I am logged in as a support engineer and located at the search by phone model table |
| When | I enter a valid dates and a valid phone model |
| Then | the table will be populated with the details of the number of call failures for this phone model within the time range entered. |

|  |  |
| --- | --- |
| User Story: 8  – scenario 2 | |
| Given | I am logged in as a support engineer and located at the search by phone model table |
| When | I enter a valid dates and an invalid phone model |
| Then | no records will be shown. |

|  |  |
| --- | --- |
| User Story: 8  – scenario 3 | |
| Given | I am logged in as a support engineer and located at the search by phone model table |
| When | I enter a valid dates and a valid phone model that has no call failures in the time period entered |
| Then | no records will be shown. |

|  |  |
| --- | --- |
| User Story: 8  – scenario 4 | |
| Given | I am logged in as a support engineer and located at the search by phone model table |
| When | I enter a valid dates and no phone model |
| Then | no records will be shown. |

|  |  |
| --- | --- |
| User Story: 9  – scenario 1 | |
| Given | I am logged in as a network management engineer and located at Get the call failures for IMSI table |
| When | I enter a valid start and end dates |
| Then | the table will be populated with the details of number of call failures and the total duration of each imsi for the entered dates. |

|  |  |
| --- | --- |
| User Story: 9  – scenario 2 | |
| Given | I am logged in as a network management engineer and located at Get the call failures for IMSI table |
| When | I enter invalid start date and end date |
| Then | no records will be shown. |

|  |  |
| --- | --- |
| User Story: 9  – scenario 3 | |
| Given | I am logged in as a network management engineer and located at Get the call failures for IMSI table |
| When | I enter valid start date and end date that has no call failures in the time entered |
| Then | no records will be shown |

|  |  |
| --- | --- |
| User Story: 10  – scenario 1 | |
| Given | I am logged in as a Network Management Engineer and the database is populated with valid data |
| When | I enter a valid UE Type and click on the search button |
| Then | A table will appear with the entered UE Type, manufacturer, model, event id, cause code, description and failure count for the given UE Type |

|  |  |
| --- | --- |
| User Story: 10  – scenario 2 | |
| Given | I am logged in as a Network Management Engineer and the database is populated with valid data |
| When | I enter an invalid UE Type and click on the search button |
| Then | The table will not be populated. |

|  |  |
| --- | --- |
| User Story: 10  – scenario 3 | |
| Given | I am logged in as a Network Management Engineer and the database is populated with valid data |
| When | I click on the search button without putting in any value for the UE Type |
| Then | I will be shown an error message saying "UE Type Cannot be empty" |

|  |  |
| --- | --- |
| User Story: 23  – scenario 1 | |
| Given | I am logged in |
| When | I move in the page |
| Then | the top bar will adapt the title to the current location. |

|  |  |
| --- | --- |
| User Story: 23  – scenario 2 | |
| Given | I am logged in |
| When | I do any search in any table |
| Then | all the results returned are distinct and all the tables have title. |

|  |  |
| --- | --- |
| User Story: 23  – scenario 3 | |
| Given | I am logged in |
| When | new data file is added in specific folder |
| Then | the valid data is added to the database |

## 5.6 System Architecture

**20/03/2019:**

**UserTestWS.java:** public voidtestSaveUserWSFails() added

**25/03/2019:**

**BaseDataDAO.java:**public List <BaseData>getPhoneCountBetweenTimes(String startTime, String endTime, String ueType) added

**BaseDataWS.java:**public ResponsefindPhoneCountByUeType (@(PathParam(“startDateTime”) String startTime, (@(PathParam(“endDateTime”) String endTime, (@(PathParam(“ueType”) String ueType) added

**28/03/2019:**

**BaseDataDAOTest.java:**public voidgetPhoneCountBetweenTimes() added

**BaseDataWSTest.java:**public voidtestFindPhoneCountByUeType () added

**01/04/2019:**

**BaseDataDAO.java:**public List <BaseData>getDistinctUEType() added

**BaseDataWS.java:**public ResponsefindDistinctUE\_Type() added

**BaseDataWSTest.java:**public voidtestFindEventIdAndCauseCodeCombination () added

**02/04/2019:**

**EventCauseDAOTest.java:**public voidtestEventCauseDAO() added

**FailureClassDAOTest.java:**public voidtestFailureClassDAO() added

**MccMncDAOTest.java:**public voidtestMccMncDAO() added

**UserEquipmentDAOTest.java:**public voidtestUserEquipmentDAO() added

**03/04/2019:**

**EventCauseTest.java:**added

**MccMncTest.java:**added

**UserEquipmentTest.java:** added

**04/04/2019:**

**BaseDataDAOTest.java:**public voidsetUp() added

**UtilsDAOTest.java:**public voiddeleteFailureClassTable() added

**UtilsDAOTest.java:**public voiddeleteBaseDataTable() added

**UtilsDAOTest.java:**public voiddeleteUserEquipmentTable() added

**UtilsDAOTest.java:**public voiddeleteMccMncTable() added

**UtilsDAOTest.java:**public voiddeleteEventCauseTable() added

**UtilsDAOTest.java:**public voiddeleteInvalidEventTable() added

**UtilsDAOTest.java:**public voiddeleteAllDataTables() added

**EventCauseTest.java:** public voidtestHasCode() added

**EventCauseTest.java:** public voidtestEquals() added

**FailureClassTest.java:** public voidtestHasCode() added

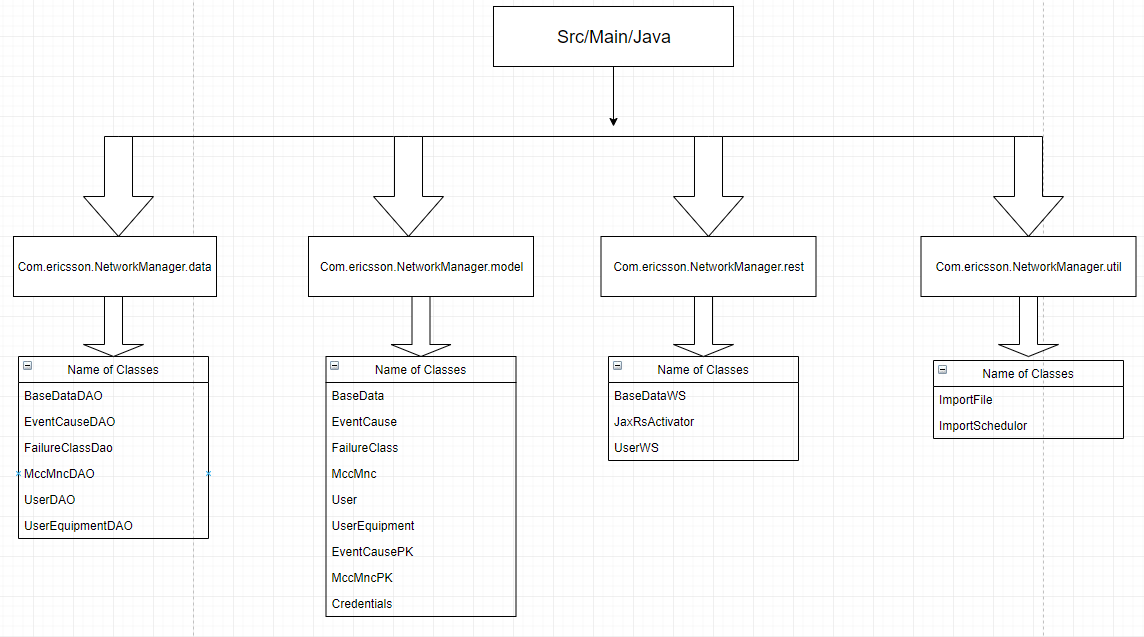
**FailureClassTest.java:** public voidtestEquals() added

**MccMncTest.java:** public voidtestHasCode() added

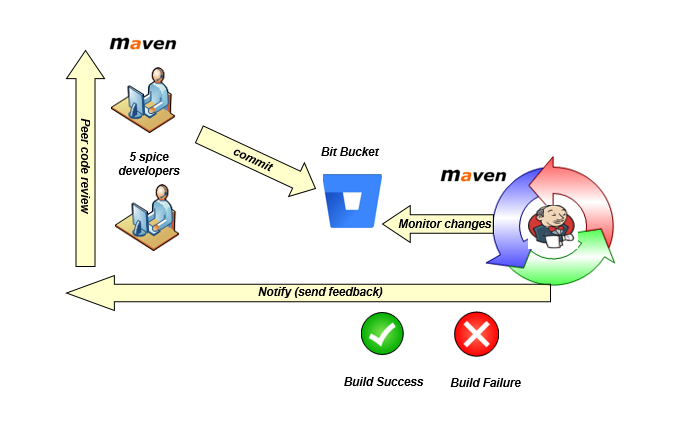
**MccMncTest.java:** public voidtestEquals() added

**UserEquipmentTest.java:** public voidtestHasCode() added

**UserEquipmentTest.java:** public voidtestEquals() added



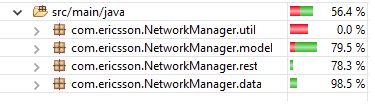
## 5.7 Development Environment



The Development Environment that we used for this sprint was to create our project on a Maven Project which is a project management tool which encompasses a project object model, a set of standards a project cycle and dependency management systems. This maven project was also connected to Bitbucket which is a web-based software project repository hosting service. Where we were able to upload and store working software project folders to the service. Bitbucket allows us to do version control, build control and collaboration. Finally, our Bitbucket was connected to Jenkins which is a build scheduling tool, used for continuous integration which allows us to schedule builds and tests for our code as they are uploaded to Bitbucket, reporting back weather each stage of the tests have passed or failed.

## 5.7 Test Results

**Code Coverage**



**Summary of Tests**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **MANUAL TEST RUN REPORT** | | | | | | | |
| **User Story ID** | **Scenario No.** | **Test No.** | **Tester** | **Date and Time of Test** | **Status** | **Bug ID and brief description** | **Jenkins Version** |
| 8 | 1 | US8 SC1 | Seán Behan | 04/04/2019 13h23 | PASS | -- | d5c5068d3a7f60522b40594360c5e5af0f570dd0 |
| 8 | 2 | US8 SC2 | Seán Behan | 04/04/2019 13h24 | PASS | -- | d5c5068d3a7f60522b40594360c5e5af0f570dd0 |
| 8 | 3 | US8 SC3 | Seán Behan | 04/04/2019 13h26 | PASS | -- | d5c5068d3a7f60522b40594360c5e5af0f570dd0 |
| 9 | 1 | US9 SC1 | Sorcha Bruton | 04/04/201913H32 | PASS | -- | f02dc2d7bd33213e09ed3dbcd267eec109621c10 |
| 9 | 2 | US9 SC2 | Sorcha Bruton | 04/04/2019 13H32 | PASS | -- | f02dc2d7bd33213e09ed3dbcd267eec109621c10 |
| 9 | 3 | US9 SC3 | Sorcha Bruton | 04/04/2019 13H32 | PASS | -- | f02dc2d7bd33213e09ed3dbcd267eec109621c10 |
| 10 | 1 | US10 SC1 | Amulya Cheluvaraja | 04/04/2019 14H23 | PASS |  | 09fc35cbbcd8679d2eea40b285f04feae72ba659 |
| 10 | 2 | US10 SC2 | Amulya Cheluvaraja | 04/04/2019 14H23 | PASS |  | 09fc35cbbcd8679d2eea40b285f04feae72ba659 |
| 10 | 3 | US10 SC3 | Amulya Cheluvaraja | 04/04/2019 14H23 | PASS |  | 09fc35cbbcd8679d2eea40b285f04feae72ba659 |

## 5.8 Team Decision Log

No major decisions were made during this sprint so the decision log is empty.

## 5.9 Sprint Retrospective

## 5.10 Evaluation and Conclusion

We achieved our Sprint Goal in relation to 3 out of 4 user stories committed to in this sprint. In Sprint 1 we received feedback in relation to our import process. The request was to change from a manual import to an automatic import and to improve our code coverage in relation to the import class.

While we completed the task of changing to an automatic import, we were once again unable to improve our code coverage in relation to the import class. In changing to an automatic import we changed the class structure so the class import no longer existed. This functionality was moved to the Import File class and once again as this was Apache Poi, we were unable to mock it and test it. As this class is mainly concerned with file handling, we as a team agreed that spending the time learning how to mock Apache Poi was not the best use of our time. Instead we ensured the validation logic was covered by automatic integration tests. We felt that this alleviated the technical debt from Sprint 1 and left us in a better position technically. We were however, unable to make the user story as done as we had not improved our code coverage relation to the import. We were happy with the feedback we received in relation to this matter as the product owner was happy to see that the validation logic was tested and said he wasn’t overly concerned with testing the file handling aspect. We are happy with our level of testing and do not feel we are carrying any technical debt into the next Sprint.

# 6. Overall Evaluation and Conclusion

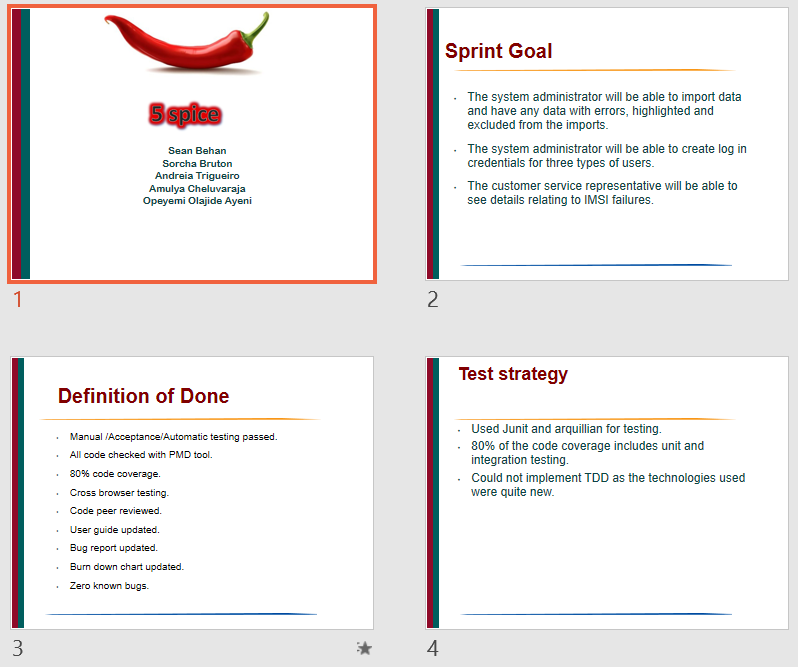
As a team we feel we have delivered a good quality product. We took on board feedback from each sprint and ensured that this was implemented and delivered to a good quality for the next Sprint. In terms of ways of working, we adhered to the scrum and agile way of working. We all benefited from this way of working. We found it to be a very good way to learn from each other. It was also very good for us in terms of making sure we were achieving our goals, working on tasks in tandem and closing them off in a methodical manner.

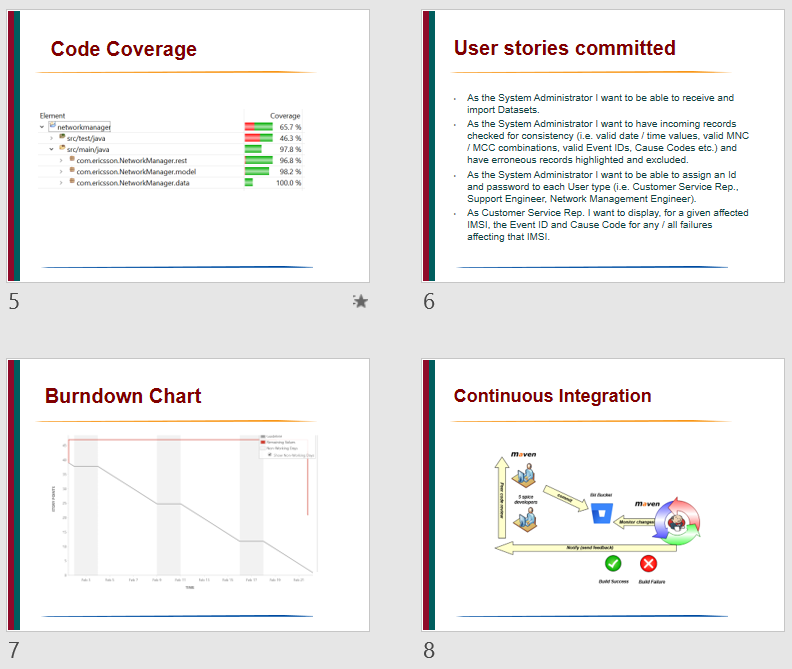
We feel we learned the technologies quite well as we were going along and we don’t feel we could have implemented TDD. If we were to redo the project with the knowledge that we have now, TDD is something that we would consider implementing. The reason we would consider implementing this is because achieving the desired code coverage was something that we consistently had to work very hard on towards the end of each sprint. This caused us unnecessary pressure and stress.

On balance we learned a lot and we as a team are very happy with our output.

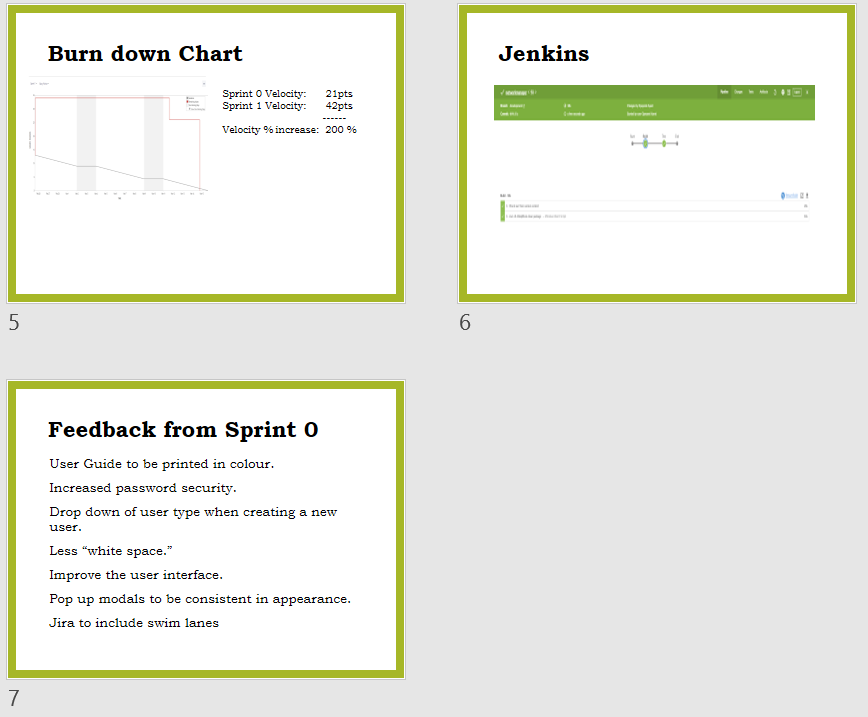
# 7. Appendix1

Slides Demo #0

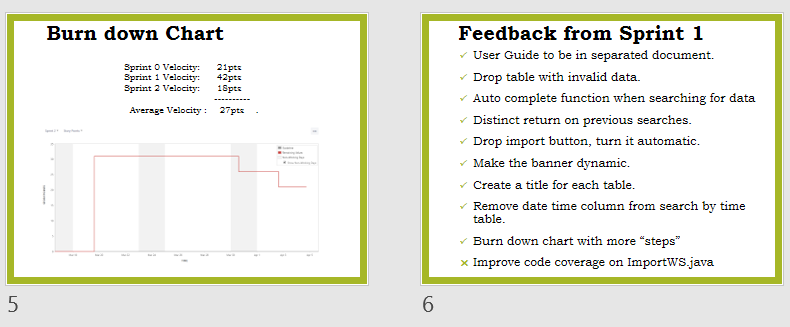
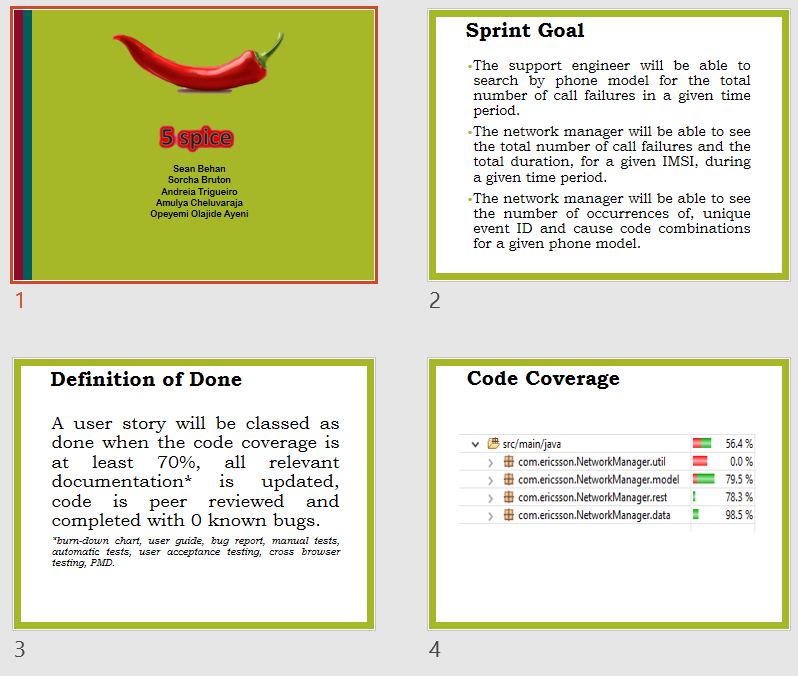




Slides Demo #1



Slides Demo #2



<User Guide>