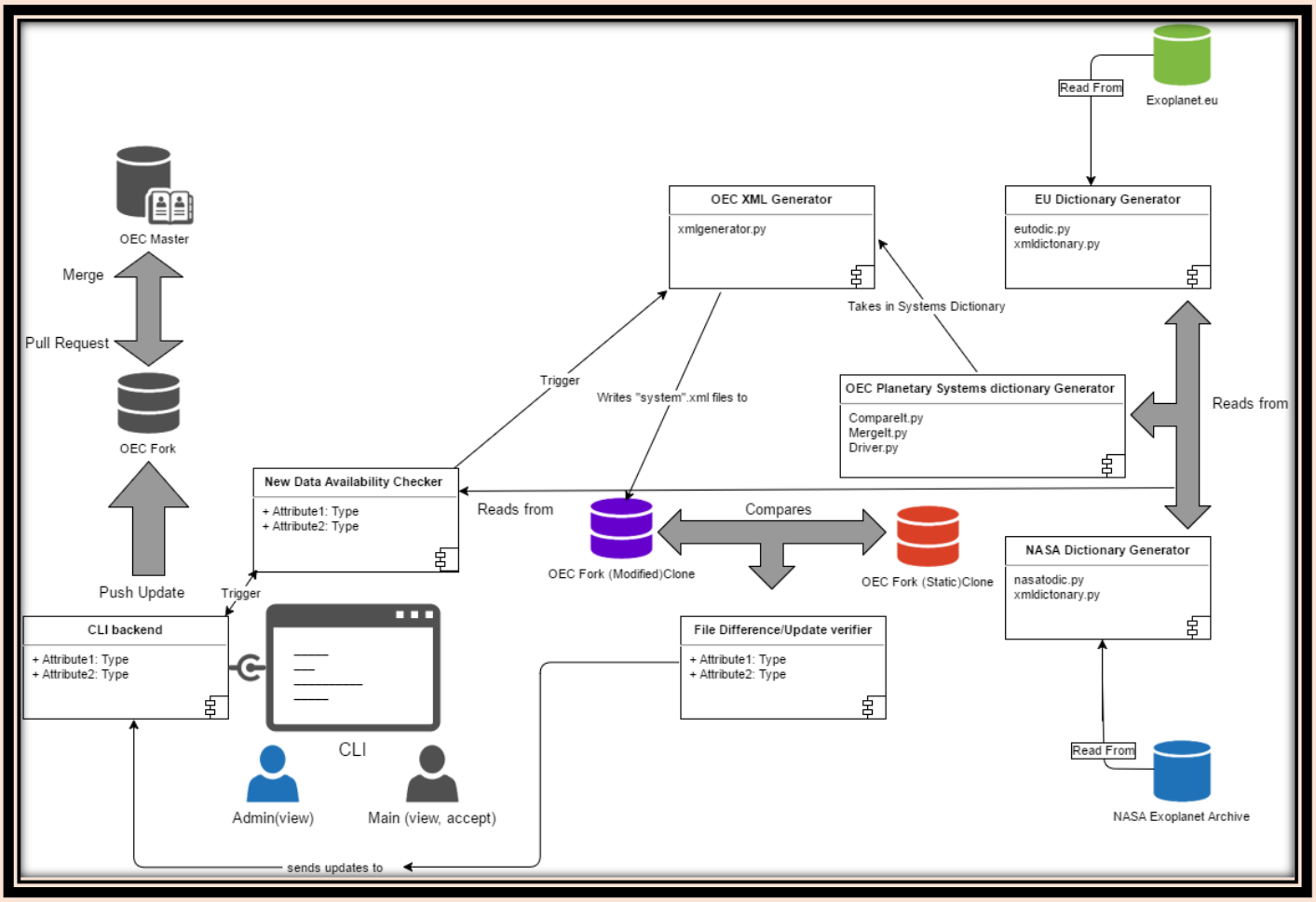
Team 15 – Andromeda

Deliverable 5: Final Report

[Sprint 3]

Table of Contents:

System Design



Completed UML Diagram of the System

The following is a description of the design of our system in the overview of its components and the script files that make the components work and interact with other to fulfill the task:

1. The Master repository of the Open Exoplanet Catalog (OEC)
2. A Fork of the OEC
3. A clone of the fork of the OEC called OEC Fork (Modified) Clone
4. A clone of the fork of the OEC called OEC Fork (Static) Clone
5. The EU Dictionary Generator module reads from the Exoplanet.eu all the data on exoplanets and stores them in a dictionary with keys being the column headings of the csv data file being read from

6. The NASA Dictionary Generator reads from the NASA Exoplanet Archive all the data on exoplanets and stores them in a dictionary with keys being the column headings of the csv data file being read from
7. The OEC Planetary Systems Dictionary Generator uses the base dictionaries from the EU Dictionary Generator and the NASA Dictionary Generator and forms a single dictionary with contents ordered by planetary systems.
    a. Where each planetary system dictionary contains individual planet dictionaries in it
    b. Where each planet dictionary contains data on each planet with keys in the planet dictionary ordered with the data fields headings of the planets such as planet name, planet radius, star name … etc. as keys,
    c. Where keys are strings formatted to be similar to OEC XML tags Ex: “<planet >< radius >”, where the first part contains the parent tags such as <planet>, <star>, <system> and the second part is the heading of the data fields.
    d. This will help with generating XML files with much ease in the next component of the system
8. Then the OEC XML generator uses the OEC Planetary Systems Dictionary to form OEC compatible XML files using the keys that are formatted to be similar to OEC XML tags, where this component writes the files to the OEC Fork Clone, since this is the only clone that is edited its designated modified and the other clone static
9. Then the File Difference/Update verifier compares the OEC Fork (Modified) Clone with the OEC Fork (Static) Clone and decides if new data file shave been generated in comparison to the original files present in the static clone, if then new files have been detected, the CLI Backend receives the necessary data to display the updates from the File Difference/Update verifier
10. The Command Line Interface(CLI), allows a user to accept/reject or edit the updates and transfer the necessary updates to the master repository of the Open Exoplanet Catalog’s located in Git hub via a push of the files from the OEC Fork (modified) Clone to the OEC fork, and then a pull request and merge to the OEC master all managed by the CLI
11. A user Admin with privileges to view the updates only
12. A user Main with privileges to view the updates, edit them and/or accept the updates

Changes Made

1. The addition of clones of the OEC Fork one to remain static called the OEC Fork (Static) Clone for comparison purposes and one to write the generated XML files to called the OEC Fork (Modified) Clone
2. The addition of A user admin for the purpose of just viewing updates
3. The Change from generating XMLs then merging them and then sending them over to the Fork, to the process in the final product where the base dictionaries are merged and then written to a single group of update XMLs, this change helped us remove the need for excess computation to figure differences and similarities between XMLs and then merge them. Now we just use the results of previous components the base dictionaries and generate rather simply system dictionaries and then write them to XMLs in the OEC Fork (Modified) Clone directory called systems

The removal of the update checking component since, inherently our system always generates only system XMLs for updated planets from the NASA and EU catalogs

Product Backlog

Cost: DH (Developer Hours)
Priority: 1 – lowest, 5 – highest

- **User Story # 1:** As Professor Smith, the client, I want the application to run on Unix environment by using software and packages that are UNIX compatible
  - Cost: 0 DH & Priority: 5

- **User Story #2:** As Professor Smith, the client, I want the application to fetch data from the NASA and EU catalogues and I want it to be organized by each star system.
  - Cost: 5 DH & Priority: 5

- **User Story #3:** As Professor Smith, the client, I would like the data that can representable in the OEC to be retrieved from the two catalogues.
  - Cost: 25 DH & Priority: 5

- **User Story #5**: As Professor Smith, the client, I want the application to produce data files in XML format that are usable as OEC XML files.
  - Cost: 25 DH & Priority: 4

- ***User Story #6:** As Professor Smith, the client, I want the application to send me emails if there is any new data available in the EU and NASA catalogues.
  - Cost: 10 DH & Priority: 3

- **User Story #7:** As Professor Smith, the client, I want the application to work on a fork version of the OEC master repository on GitHub.
  - Cost: 8 DH & Priority: 4

- **User Story #7.5:** As Professor Smith, the client, I want the application to check if the new XML file created already exists in the forked OEC repository and report an edit or new addition respectively.
  - Cost: 20 DH & Priority: 4

- **User Story # 7.6:** As Professor Smith, the client, I want the application to let me manually choose which new files should go into the repository.
  - Cost: 15 DH & Priority: 4

- **User Story #8:** As Professor Smith, the client, I want the application to receive daily updates, if any, from the NASA and EU catalogues.
  - Cost: 7 DH & Priority: 3

- **User Story #9:** As Professor Smith, the client, I would like to be notified of all updates available when I log into the application
    - Cost: 13 DH & Priority: 3

- **User Story #11:** As Alice, the graduate student, I want the application to let me view the permissions for updates and the log.
    - Cost: 7 DH & Priority: 2

- **User Story #12:** As Professor Smith, the client, I would like the application to allow human intervention by allowing myself to manually edit errors by modifying the updates directly brought through by the data from the other catalogues
    - Cost: 18 DH & Priority: 3

Release Plan

- We haven't made any changes to our release plan.
- Our last release will be on December 1st 2016

Sprint Plan

- This sprint will end **on December 1st 2016.**

Sprint Backlog for Sprint #3:

[#6] As Professor Smith, the client, I want the application to send me emails if there is any new data available in the EU and NASA catalogues. [Cost: 10 DH & Priority: 4]

Tasks:

- To be done by: Daniel
- Check for updates from the catalogues
- Create file to send email to recipient if updates are available

[#9] As Professor Smith, the client, I would like to be notified of all updates available when I log into the application. [Cost: 18 DH & Priority: 4]

Tasks:

- To be done by: Suhailah & Kelvin
- Run a script that acts as the start up screen of the system
- Use the log-in process to determine whether user is admin or client
- Check for other errors [wrong user, wrong password]

[#11] As Alice, the graduate student, I want the application to let me view the permissions for updates and the log. [Cost: 2 DH & Priority: 2]

Tasks:

- To be done by: Suhailah
- Run script that lets admin see the changes made in the OEC repository

[#12] As Professor Smith, the client, I would like the application to allow human intervention by allowing myself to manually edit errors by modifying the updates directly brought through by the data from the other catalogues [Cost: 5 DH & Priority: 3]

Tasks:

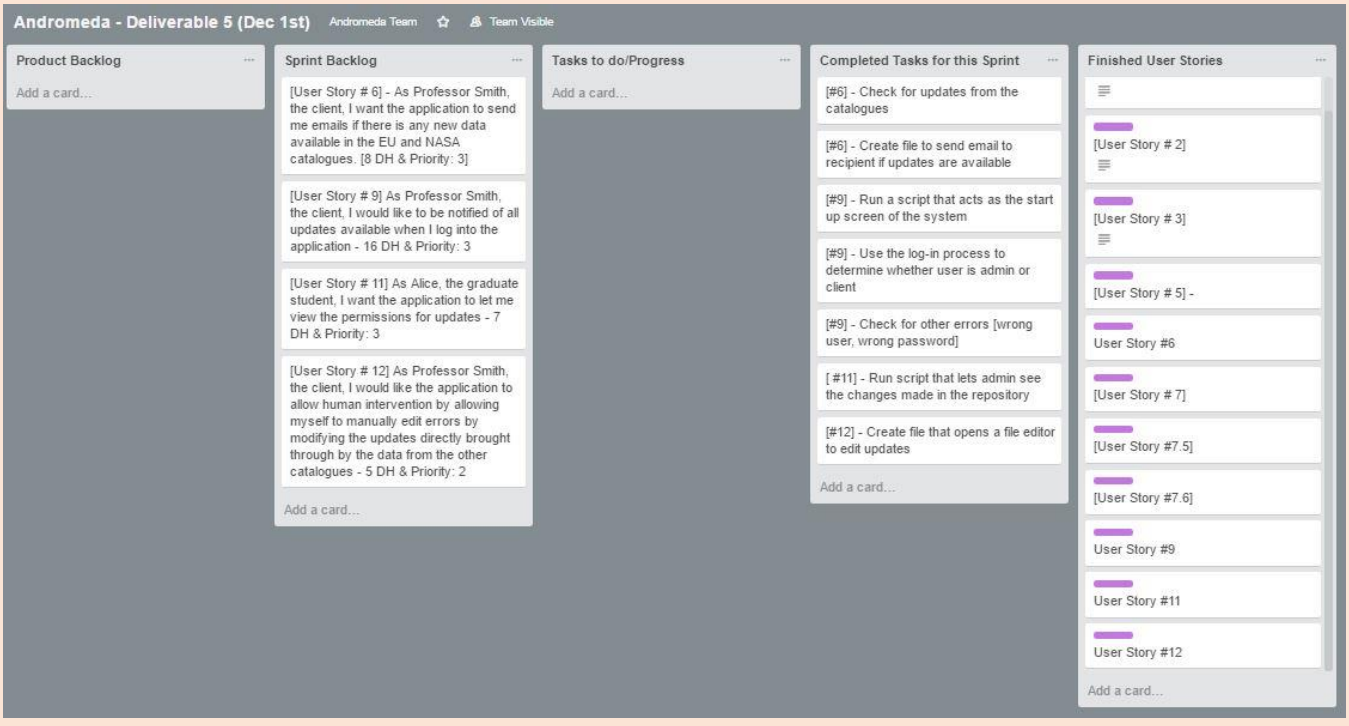
- To be done by: Daniel.
- Create file that opens a file editor to edit updates

Iteration Plan:

Sprint 3 - Deliverable 5			Daniel Matu, Kelvin Ong, Hajoon Choi, Ralph Samson, Suhailah Rahman
Story #	**Priority Level**	**Cost**	**User Story**
	1 - lowest		
	5 - highest		
6	3	10	As Professor Smith, the client, I want the application to send me emails if there is any new data available in the EU and NASA catalogues.
9	4	18	As Professor Smith, the client, I would like to be notified of all updates available when I log into the application
11	3	2	As Alice, the graduate student, I want the application to let me view the permissions for updates
12	2	5	As Professor Smith, the client, I would like the application to allow human intervention by allowing myself to manually edit errors by modifying the updates directly brought through by the data from the other catalogues

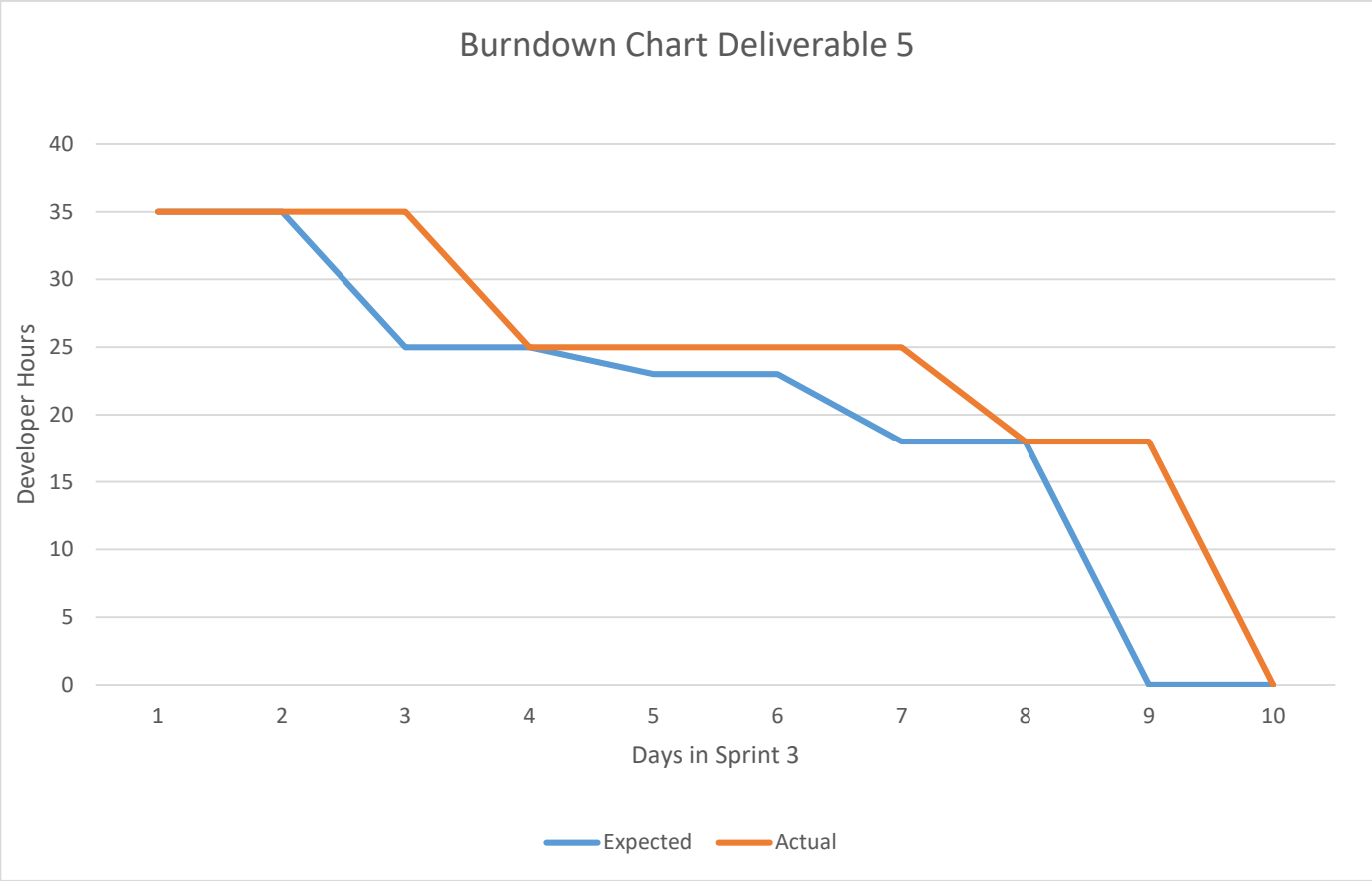
Tasks	**Day 1**	**Day 2**	**Day 3**	**Day 4**	**Day 5**	**Day 6**	**Day 7**	**Day 8**	**Day 9**	**Day 10**
[#6] - Check for updates from the catalogues	DM:2	DM:2	DM:2	0	0	0	0	0	0	0
[#6] - Create file to send email to recipient if updates are available	0	0	DM:2	DM:2	0	0	0	0	0	0
[#9] - Run a script that acts as the start up screen of the system	KO: 1	KO: 2	SR: 2	0	SR: 1	KO: 1	SR: 1	0	SR:1	0
[#9] - Use the log-in process to determine whether user is admin or client	0	0	0	SR: 3	0	SR:1	SR:1	SR: 1	0	0
[#9] - Check for other errors [wrong user, wrong password]		0	0	0	K0:1	0	0	SR:1	0	SR:1
[#11] - Run script that lets admin see the changes made in the repository	0	0	0	0	0	0	SR: 1	SR:1	0	0
[#12] - Create file that opens a file editor to edit updates	0	0	0	0	DM:1	DM: 2	DM: 1	DM:1	0	0
TOTAL HOURS PER DAY	3	4	6	5	3	4	4	4	1	1

Task Board:

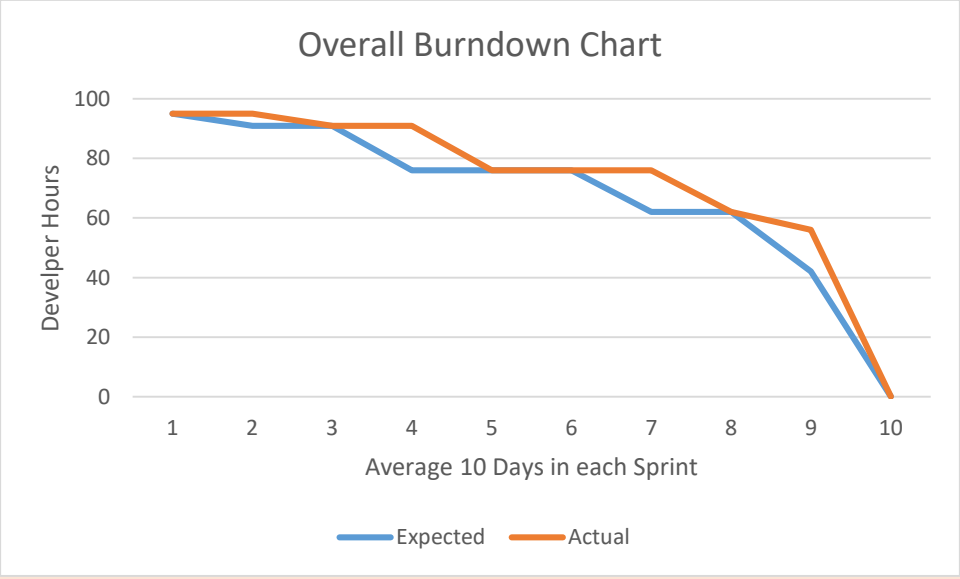


Burn-down Chart:

Sprint 3



Overall



For the over-all chart, we decided to keep the usual 10 says in our sprint to show the average amount of work done in each sprint. This shows that in every sprint, we get the tasks done at the end but not on time during the sprint due to a number of reasons such as fixing bugs for a user story or it is taking more time than usual.

System Validation Activities

For the system validation, we had spoken with Kim Pham, one of the client representatives. We showed her a demo of our initial front-end which involved the log-in screen with the admin/main user homepage. She gave her approval on the front-end and she had also gave us some advice in how to improve our program to make sure we have everything that the client wants.

For evidence, please refer as she will be contacting the professor to let her know which teams have consulted for system validation.

Brief Overview – State of the entire project

- **Transition from Deliverable 4 to Deliverable 5:**
  - We continued on from our previous sprint and added more to the front-end such as the log-in screen, and options for both users: main and admin.
  - Using the concepts and practices of Agile development, the work progressed fairly smoothly, since we had completed most of the back-end in the previous sprints.
  - In this sprint, we had to add the final touches and fix some small bugs to make sure that code doesn't crash.
  - The code inspection from each member helped as each respective code was fixed.
- **Overview of the entire project**
  - In the first deliverable, we brainstormed ideas for this project.
  - Our research phase occurred throughout the second phase where we had to think of the types of users that would help with the implementations. We thought of different user stories and personas
  - We started developing our product in the third deliverable, where we mainly worked on fetching the raw data from the EU and NASA catalogs. We also cleaned the raw data to correspond with the specific OEC format.
  - We started work on our front end, and decided that the command line would be the best way for this program. We worked on to show the suggested differences to the main user and let him accept or reject the changes made from the different catalogs. We also worked on producing XML files that worked with the OEC format.
  - In the last deliverable, we finalized our front-end and added the log-in system for the different type of users. For the back-end, we merged the XMLs.
- **Estimated Project Velocity: 40**
- **Actual Project Velocity: 40**

- **Did we follow with our plan? Or did we have to re-plan? Why and When? How did the work for deliverable 5 compare to the work on all previous deliverables?**
    - In terms of progress and end-result, we have a final product which was finished on time. Through out the sprints, progress was made in a good pace.
    - We did the user stories as expected in each sprint but most of the work were done with the back-end, which the team were constantly improving and making sure it didn't have a problem while running the program
    - We had to re-plan at the beginning of each sprint to make sure we can finish implementing off the planned features at the time of each release. We did end up taking off user stories because they were either an epic or it did not go with our plans at the time.
    - As many developers, we had to work with bugs and obstacles but we made sure we had a final product at the end.
    - The work for deliverable 5 was expected as it was the last sprint, so this time we had to make sure we don't leave any incomplete code or user stories. We used our experiences from the past deliverables to not cram any work last minute.