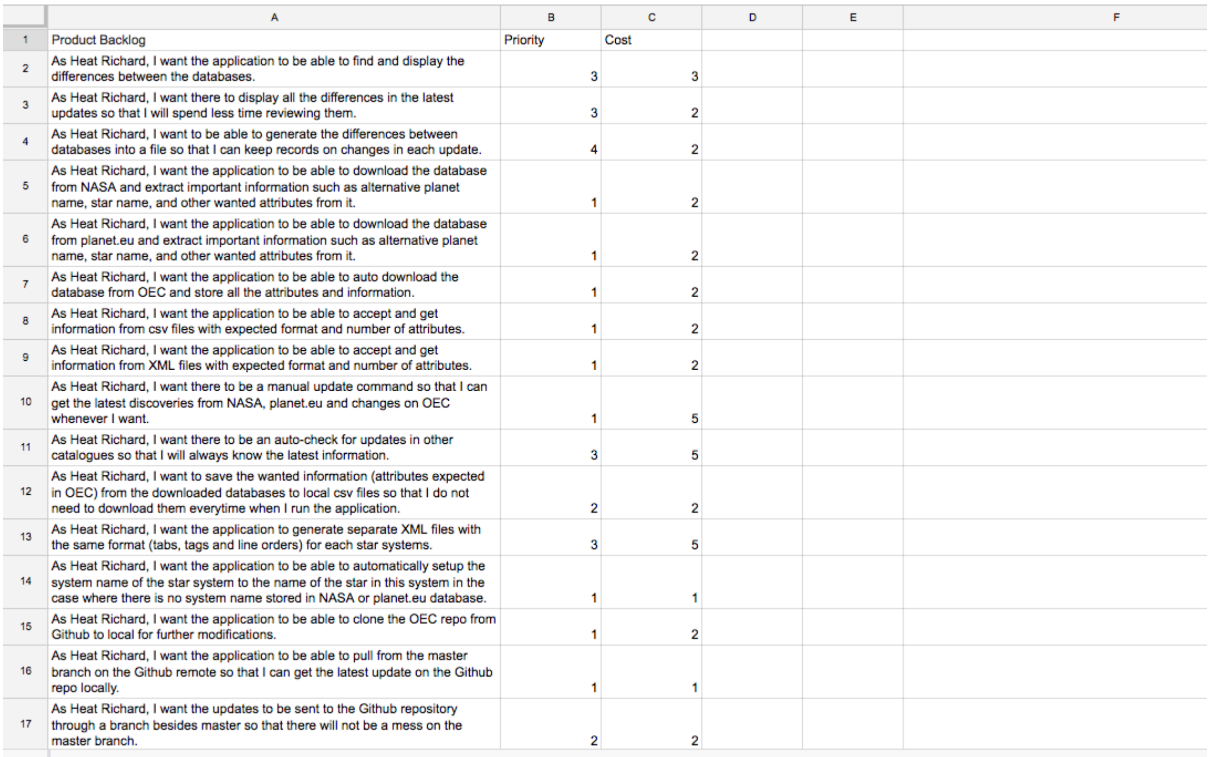
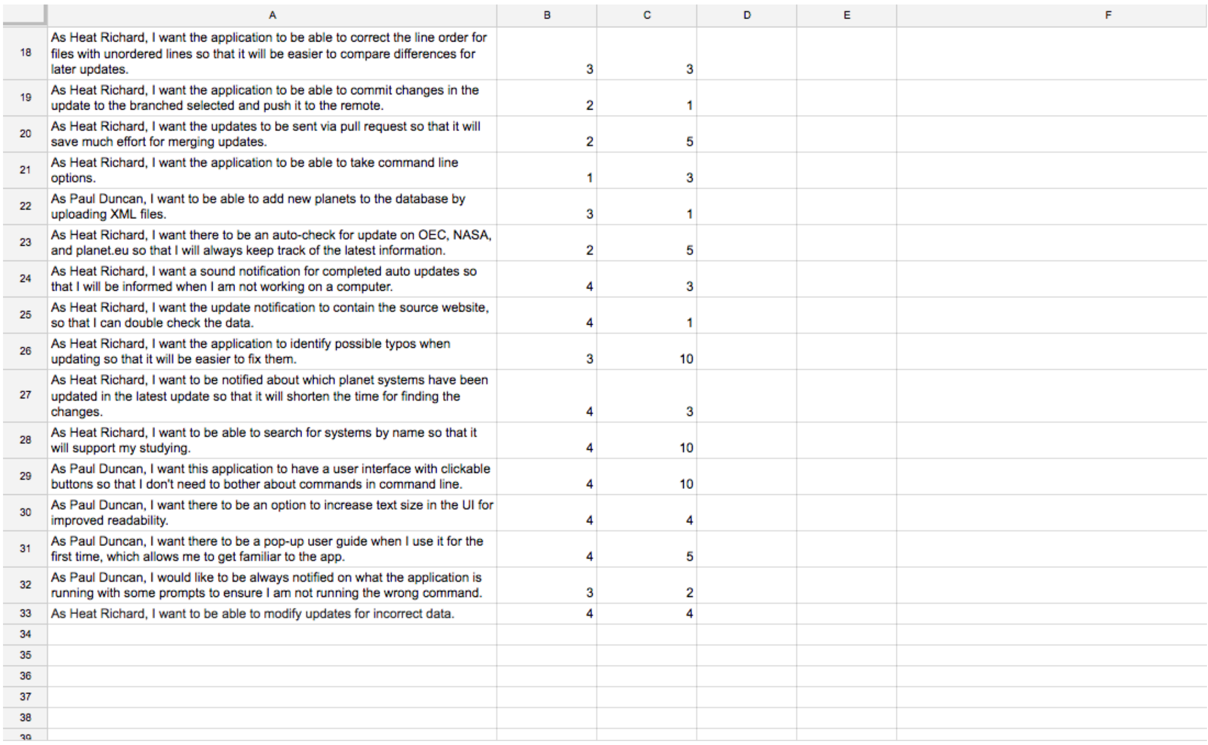
Team TMNT (Team #22) Deliverable 5

Menu

Product Backlog



	A	B	C	D	E	F
1	Product Backlog	Priority	Cost			
2	As Heat Richard, I want the application to be able to find and display the differences between the databases.	3	3			
3	As Heat Richard, I want there to display all the differences in the latest updates so that I will spend less time reviewing them.	3	2			
4	As Heat Richard, I want to be able to generate the differences between databases into a file so that I can keep records on changes in each update.	4	2			
5	As Heat Richard, I want the application to be able to download the database from NASA and extract important information such as alternative planet name, star name, and other wanted attributes from it.	1	2			
6	As Heat Richard, I want the application to be able to download the database from planet.eu and extract important information such as alternative planet name, star name, and other wanted attributes from it.	1	2			
7	As Heat Richard, I want the application to be able to auto download the database from OEC and store all the attributes and information.	1	2			
8	As Heat Richard, I want the application to be able to accept and get information from csv files with expected format and number of attributes.	1	2			
9	As Heat Richard, I want the application to be able to accept and get information from XML files with expected format and number of attributes.	1	2			
10	As Heat Richard, I want there to be a manual update command so that I can get the latest discoveries from NASA, planet.eu and changes on OEC whenever I want.	1	5			
11	As Heat Richard, I want there to be an auto-check for updates in other catalogues so that I will always know the latest information.	3	5			
12	As Heat Richard, I want to save the wanted information (attributes expected in OEC) from the downloaded databases to local csv files so that I do not need to download them everytime when I run the application.	2	2			
13	As Heat Richard, I want the application to generate separate XML files with the same format (tabs, tags and line orders) for each star systems.	3	5			
14	As Heat Richard, I want the application to be able to automatically setup the system name of the star system to the name of the star in this system in the case where there is no system name stored in NASA or planet.eu database.	1	1			
15	As Heat Richard, I want the application to be able to clone the OEC repo from Github to local for further modifications.	1	2			
16	As Heat Richard, I want the application to be able to pull from the master branch on the Github remote so that I can get the latest update on the Github repo locally.	1	1			
17	As Heat Richard, I want the updates to be sent to the Github repository through a branch besides master so that there will not be a mess on the master branch.	2	2			

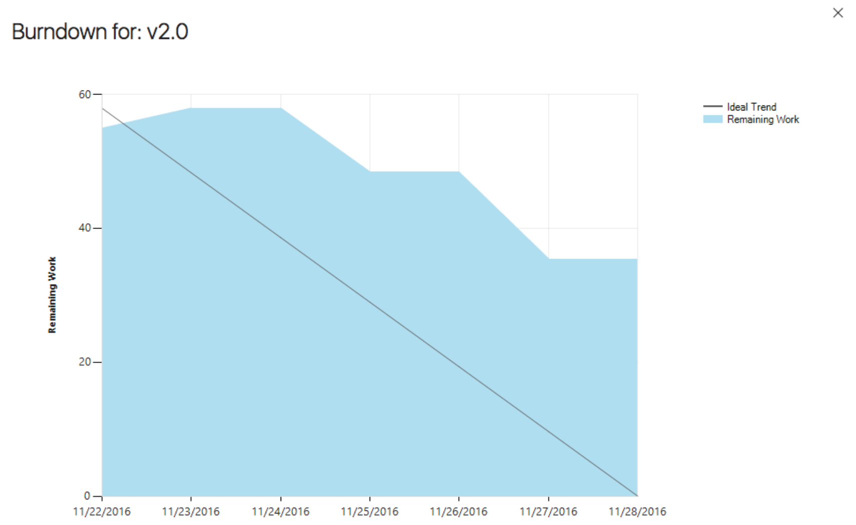


	A	B	C	D	E	F
18	As Heat Richard, I want the application to be able to correct the line order for files with unordered lines so that it will be easier to compare differences for later updates.	3	3			
19	As Heat Richard, I want the application to be able to commit changes in the update to the branched selected and push it to the remote.	2	1			
20	As Heat Richard, I want the updates to be sent via pull request so that it will save much effort for merging updates.	2	5			
21	As Heat Richard, I want the application to be able to take command line options.	1	3			
22	As Paul Duncan, I want to be able to add new planets to the database by uploading XML files.	3	1			
23	As Heat Richard, I want there to be an auto-check for update on OEC, NASA, and planet.eu so that I will always keep track of the latest information.	2	5			
24	As Heat Richard, I want a sound notification for completed auto updates so that I will be informed when I am not working on a computer.	4	3			
25	As Heat Richard, I want the update notification to contain the source website, so that I can double check the data.	4	1			
26	As Heat Richard, I want the application to identify possible typos when updating so that it will be easier to fix them.	3	10			
27	As Heat Richard, I want to be notified about which planet systems have been updated in the latest update so that it will shorten the time for finding the changes.	4	3			
28	As Heat Richard, I want to be able to search for systems by name so that it will support my studying.	4	10			
29	As Paul Duncan, I want this application to have a user interface with clickable buttons so that I don't need to bother about commands in command line.	4	10			
30	As Paul Duncan, I want there to be an option to increase text size in the UI for improved readability.	4	4			
31	As Paul Duncan, I want there to be a pop-up user guide when I use it for the first time, which allows me to get familiar to the app.	4	5			
32	As Paul Duncan, I would like to be always notified on what the application is running with some prompts to ensure I am not running the wrong command.	3	2			
33	As Heat Richard, I want to be able to modify updates for incorrect data.	4	4			
34						
35						
36						
37						
38						
39						

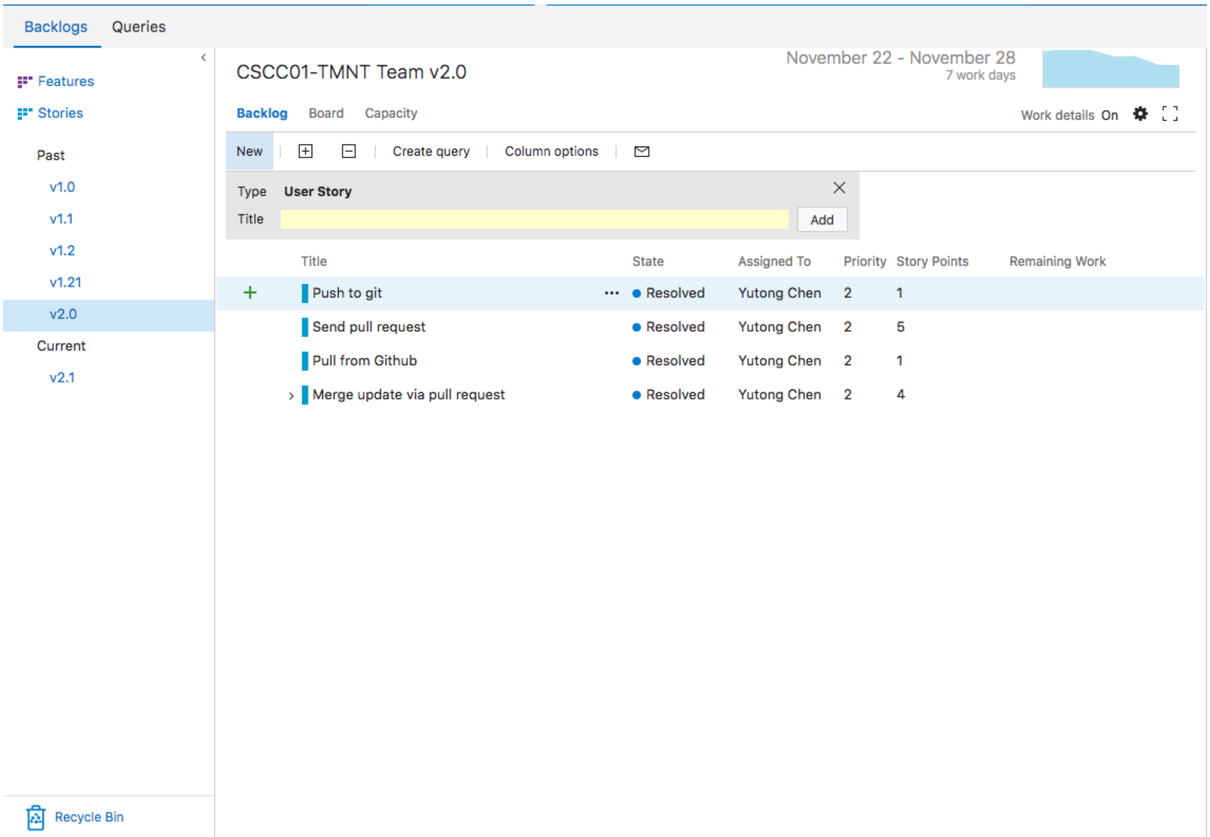
Release Plan

Release	Freeze	Publishing	Content
v1.0	Oct. 23	Oct. 24	Basic working copy for this project, create our own object designs to store information, download OEC database from systems.xml, clone and pull a Github repo from remote to local, and have a command line interface.
v1.1	Oct. 30	Oct. 31	Finish manual updates from planet.eu and NASA, can create git branches for the cloned repo and can do commit locally to save the change, take different command line input, display difference through git diff.
v1.2.0	Nov. 13	Nov. 14	Implement merge data from different format, generate xml, implement push from local to remote with Github, finish creating pull request from Github remote, basic uninterruptable auto update and email notification.
v1.2.1	Nov. 20	Nov. 21	Optimize merge algorithm and continue implementing generate xml, implement display difference/ display updates in command line, make the auto update to be interruptible.
v2.0	Nov. 27	Nov. 28	Creating basic UI for the application, implement line order rebasing, finish push local commit to remote for Github.
v2.1	Dec. 3	Dec. 4	Add minor features and wrap up.

Burndown Chart



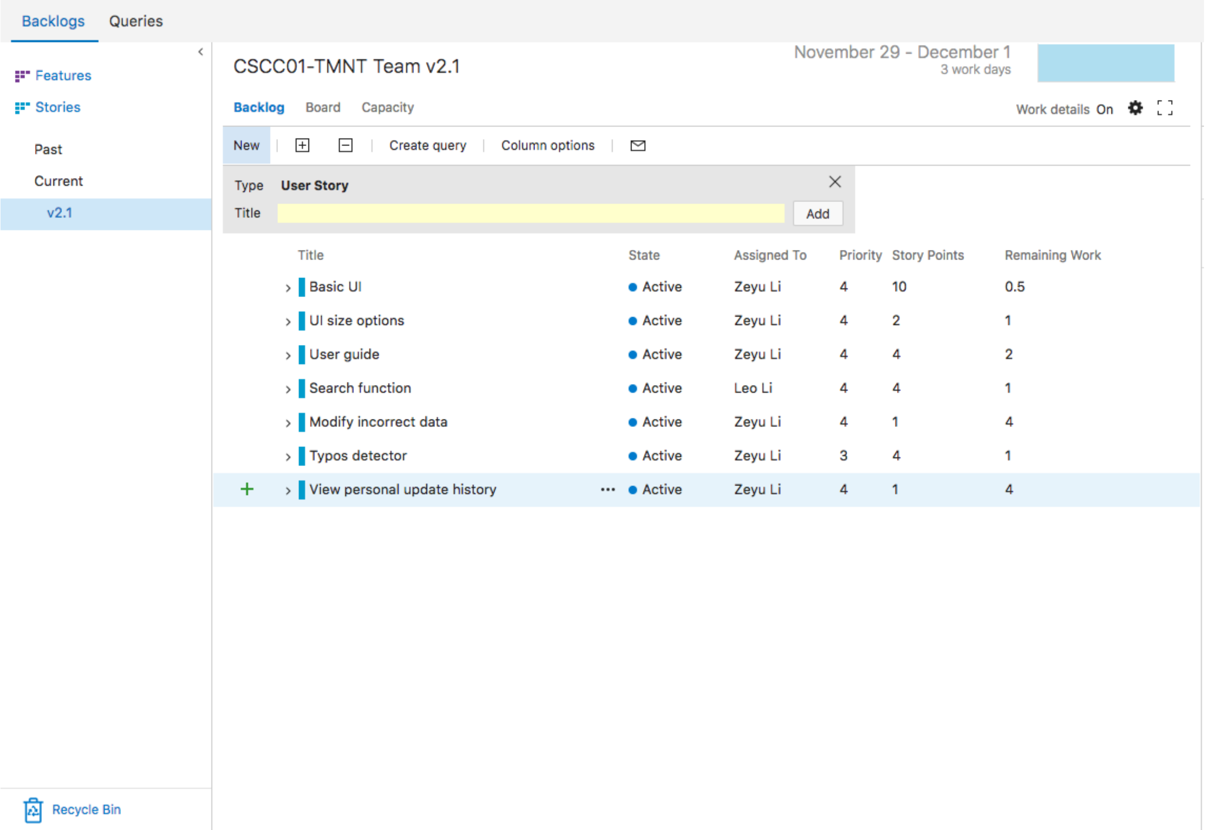
Sprint Plan



CSCC01-TMNT Team v2.0

November 22 - November 28
7 work days

Title	State	Assigned To	Priority	Story Points	Remaining Work
Push to git	● Resolved	Yutong Chen	2	1	
Send pull request	● Resolved	Yutong Chen	2	5	
Pull from Github	● Resolved	Yutong Chen	2	1	
Merge update via pull request	● Resolved	Yutong Chen	2	4	

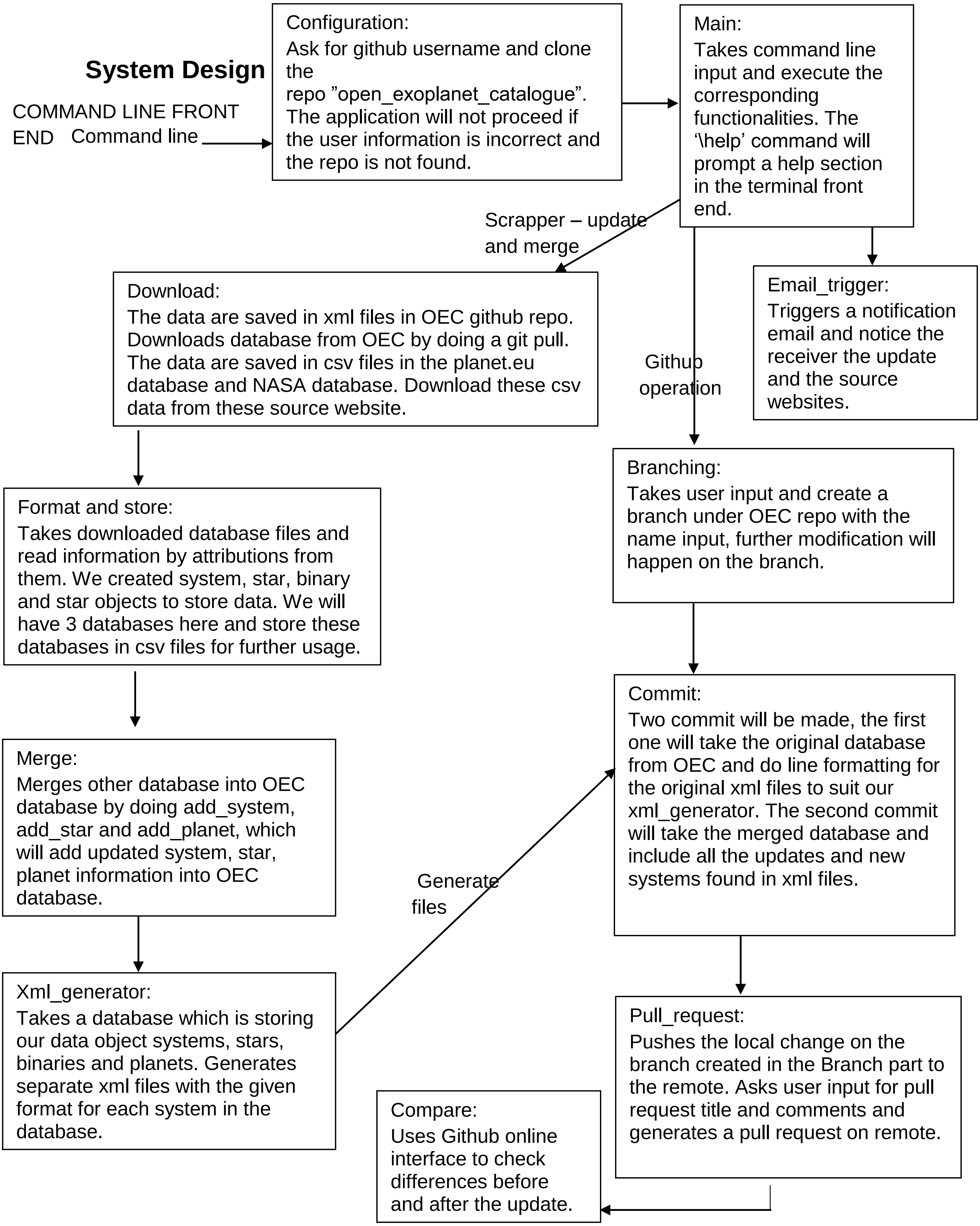


CSCC01-TMNT Team v2.1

November 29 - December 1
3 work days

Title	State	Assigned To	Priority	Story Points	Remaining Work
Basic UI	● Active	Zeyu Li	4	10	0.5
UI size options	● Active	Zeyu Li	4	2	1
User guide	● Active	Zeyu Li	4	4	2
Search function	● Active	Leo Li	4	4	1
Modify incorrect data	● Active	Zeyu Li	4	1	4
Typos detector	● Active	Zeyu Li	3	4	1
View personal update history	● Active	Zeyu Li	4	1	4

System Design



COMMAND LINE FRONT END

Command line

Configuration:

Ask for github username and clone the repo ”open_exoplanet_catalogue”. The application will not proceed if the user information is incorrect and the repo is not found.

Main:

Takes command line input and execute the corresponding functionalities. The ‘\help’ command will prompt a help section in the terminal front end.

Scrapper – update and merge

Github operation

Email_trigger:

Triggers a notification email and notice the receiver the update and the source websites.

Download:

The data are saved in xml files in OEC github repo. Downloads database from OEC by doing a git pull. The data are saved in csv files in the planet.eu database and NASA database. Download these csv data from these source website.

Format and store:

Takes downloaded database files and read information by attributions from them. We created system, star, binary and star objects to store data. We will have 3 databases here and store these databases in csv files for further usage.

Merge:

Merges other database into OEC database by doing add_system, add_star and add_planet, which will add updated system, star, planet information into OEC database.

Xml_generator:

Takes a database which is storing our data object systems, stars, binaries and planets. Generates separate xml files with the given format for each system in the database.

Generate files

Branching:

Takes user input and create a branch under OEC repo with the name input, further modification will happen on the branch.

Commit:

Two commit will be made, the first one will take the original database from OEC and do line formatting for the original xml files to suit our xml_generator. The second commit will take the merged database and include all the updates and new systems found in xml files.

Pull_request:

Pushes the local change on the branch created in the Branch part to the remote. Asks user input for pull request title and comments and generates a pull request on remote.

Compare:

Uses Github online interface to check differences before and after the update.

Changes Made in Agile Planning

Moved all the epics and the stories that we will never be able to implement in this term to the final release. Separated huge user stories into small pieces and distributed them into different sprints. We also changed the previous two-week sprint into two one-week sprint.

Code Inspection

Each team member’s code inspection summary is in the files/d5-inspection repo on Github

Inspection Meeting

YouTube video address: https://www.youtube.com/watch?v=S343ytWoSNM

Project Saga

Team TMNT was formed when a group of computer science students came together to take on the challenge of automating updates to the OEC. Keeping the project requirements in mind, the group set to work, brainstorming, creating personas, user stories, and eventually, a mostly fleshed-out system design.

Soon, the team was coding over (mostly) week-long sprints. The key functionalities were focused on first – taking input through the command line, reading CSV and XML input. Then, components that cloned the OEC and merged the three databases' data would be added. After that, other stories like pushing to Git would be addressed.

But as the coding began, the bugs and issues inevitably followed. Why do these handy modules support Python 2 but not 3? How do we deal with conflicts between two databases? Why does our scrape from the NASA database suddenly not work??

The TMNT persevered. Their Python 3 code was converted to Python 2. Professor Rein was consulted. It turned out that the NASA site was down.

Fighting against the burndown chart, the gang came out of the project's struggles as better developers. They improved their ability to work on one project with multiple people at once, demonstrating better version control over the course of the project; they gained experience writing code that deals with large amounts of data that may or may not conflict; they saw firsthand the role granularity plays in keeping code manageable and keeping testing simple and effective. They also learned that pickle isn't a fan favourite, and that Python 2 is apparently more popular than its successor.

Months of effort have culminated in what is perhaps one of the most ambitious projects undertaken by the members of TMNT. In the future, each developer is sure to look back to this project and remember the importance of having a solid system design (especially before too much coding happens!).

Project Velocity

Estimated project velocity v2.0: 11
Actual project velocity v2.0: 10
Estimated project velocity v2.1:26
Actual project velocity v2.1: 0

The work in release 2.1 are mostly what we left over as epics. We did some research on these user stories and figured out that they will take much time to implement and we will never be able to do them in this particular term. As it is the end of the school term, all the assignment due dates are quite close, and we were not able to finish all the burndown for v2.0.

Validation Activities

We met Professor Rein once and one of the TA once in person for the validation activities (we did this before the previous deliverable but we did not include it in the deliverable). We also contacted the professor through email for some validations. We asked about which database on NASA we were going to download. He also clarified that if the data from NASA or planet.eu does not have a system name, the system name should be the same as the name of a star in the system. The second we meet with the TA and she provided us a list of expected attributes for OEC. Through emails, we asked about how are were going to use the xml files for information, and how to set the last update time for the database. He confirmed that we need to use all the xml files under open_exoplanet_catalogue/systems repo, and we should change the last update time to current time if there are any updates.

Difference between Deliverables

According to TA's feedback, we have changed several things for this deliverable. We remade a product back with google sheet which will display all the user stories instead of the summary of our user stories. We added a description of the validation activities in this deliverable which was what we missed. We also moved the system design diagram into the document. A title page with group name, log and the deliverable menu was added.