Team Jelly: Deliverable 3

CSCC01

October 23, 2016

Team 25

Members: Dennis Tra, Henry Liu, Angelina Choi, Kelly Mo, Kris Lai

TABLE OF CONTENTS

Overview of progress from deliverable 2 to deliverable 3

Our first official sprint has been both challenging and successful. The team had initially planned to complete 1 story point per developer a day but there were setbacks which impeded our developers from starting their development. As a result, 5 story points required from each developer a week was condensed into a two-day timeline. This changed our burndown chart's actual time which resulted in a cliff like curve.

Our team saw changes in the product backlog after a TA meeting where it was established that the previous user story 1 was not a user story, but rather a non-feature requirement. This also changed our task board and allowed us to completed another user story. We are currently still on track as 25 user stories were still completed during this sprint.

We were able to implement 2 working components into this sprint. The ability to print all records with an update time bigger than an update time, stored locally in a text file, from the Nasa and exoplanet catalogue. We were also able to implement the front end user script which prompts the user for input and calls the corresponding scripts. In terms of coding, we had initially decided on using shell commands for both the front end as well as the back end. After consultation with our TA, we established that it would be more practical to develop with python in order to allow easier maintenance and modifications to our code in the future. This has changed tools towards using a python development tool such as Wing. Now currently both the front and back end are implement with python scripts.

System Design

Our Initial system designed was to complete the project using shell script. After consulting with our TA, we decided that a more practical use was to use python for our backend as it is easier to maintain which provides more flexibility in the future if changes are needed.

Back End Python

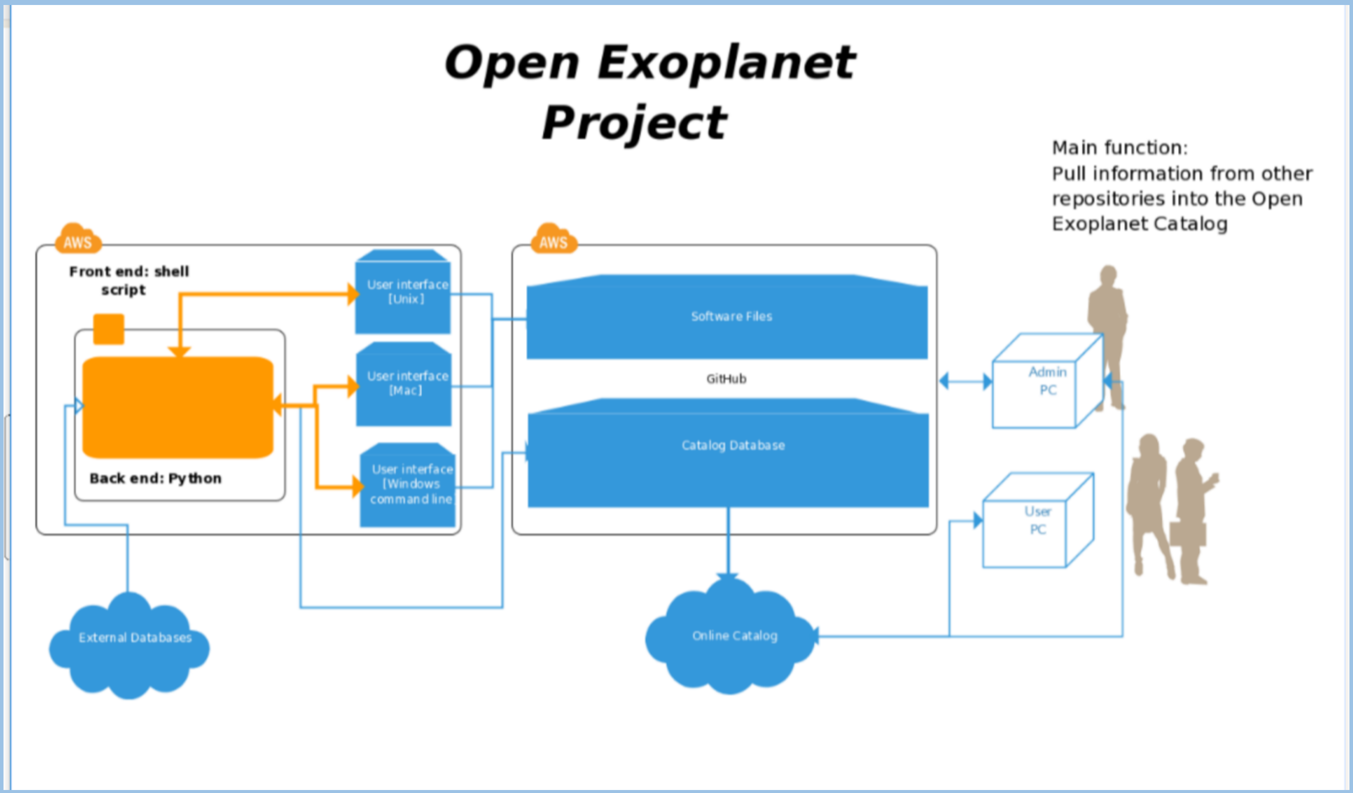
There are currently two different portions for the backend python scripts

GitHub Push: Which automatically pushes updates from our local system to a shared repository.

Database interaction: Currently pulls data from http://exoplanetarchive.ipac.caltech.edu/ and http://exoplanet.eu/. It then prints out the names of records with a last modified date, on both databases, that is greater than or equal to the last commit date stored in a text file on your local machine.

Front End User Interface

The front end interface of our project is a front end python script run through command line. This interface then calls the corresponding back end scripts of Update GitHub or Pull from Database.



Product Backlog – Sprint 1

Our Product Backlog initially contained 4 user stories with only 3 expected to be completed in this sprint. After a meeting with our TA, we established that our first user story was a non-feature requirement. Our revised product backlog contains the following stories.

Note:

-Changed Values represented by (OLD -> NEW)

-1 Story point = 0.5 developer hours = 30 minutes

Priority: P Story points: SP Developer Hours: H	User Story
P: 2 -> 1 **SP**: 8 -> 4 **H**: 2	As Anne (an admin), I want a command that will output a manual of all commands and how to use them
P: 3 -> 2 **SP**: 8 -> 10 **H**: 5	As Hanno (an admin), I want a command to list all, line by line, all planetary systems that have been updated in other catalogues since the last commit
P: 4 -> 3 **SP**: 11 **H**: 5.5	As Hanno (an admin), I want to be able to manually push and commit updates from other catalogues (Exoplanet.eu, exoplanetarchive.ipac.caltech.edu) to OEC so that I can immediately add updated information to my catalogue

From our meeting with our TA, we established that the following user story was a non-feature requirement. That allowed us to remove it from our Product backlog for the current sprint and move all other stories up by one priority.

Priority: P Story points: SP Developer Hours: H	User Story
P: 1 - > 0 **SP**: 2 -> 0 **H**: 1 -> 0	As Anne (an admin), I want a command that will output a manual of all commands and how to use them

Sprint Plan – Sprint 1

Dennis -D

Henry – H

Kris Lai – KL

Kelly Mo – KM

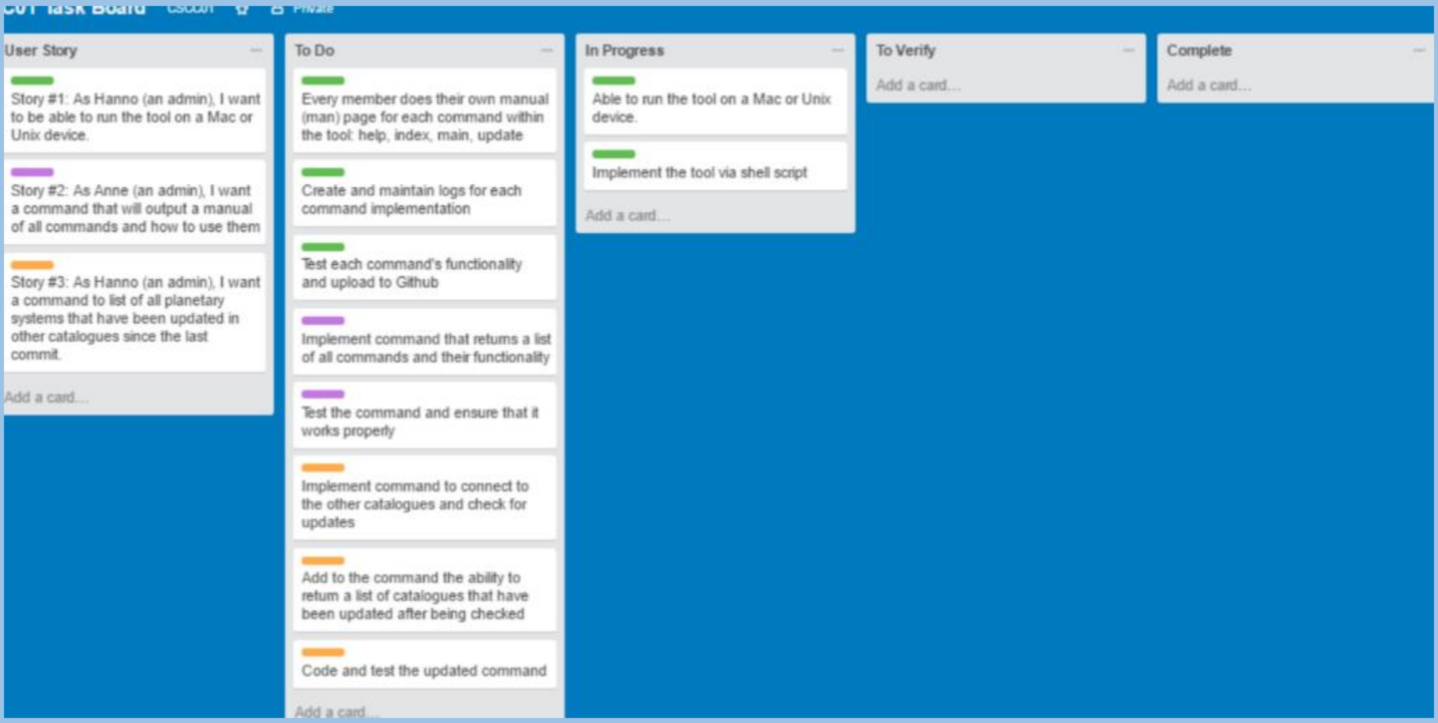
Angelina - A

User Story	Task	D1	D2	D3	D4	D5	D6	D7
Story #1: As Anne (an admin), I want a command that will output a manual of all commands and how to use them	Task 1: Every member does their own manual (man) page for each command within the tool: help, index, main, update						KL 3	
	Task 2: Need a main.py to be the first program to run						KL 1	
Story #2: As Hanno (an admin), I want a command to list of all planetary systems that have been updated in other catalogues since the last commit.	Task 3: Figure out how to check the updated dates on Exoplanet Catalogue (http://exoplanet.eu/catalog). Reading the source code somehow?						D2	D1
	Task 4: Figure out how to check the updated dates on NASA (http://exoplanetarchive.ipac.caltech.edu/docs/data.html). Reading the source code somehow?						H1	H2
	Task 5: Figure out how to keep track of last commit date. eg. text file?							D1
	Task 6: Add to the command the ability to print a list of catalogues that have been updated after being checked							H2
	Task 7: Test the command and ensure that it works properly							KL1

Story #3: As Hanno (an admin), I want to be able to manually push and commit updates from other catalogues (Exoplanet.eu, exoplanetarchive.ipac.caltech.edu) to my git repository so that I can immediately add updated information to my catalogue	Task 8: Figure how to call git commands in Python						A5 KM 1	
	Task 9: Commit repository with message option						KM 2	KM2
	Task 10 : Test pushing a file to a git repository							D1

Task Board

Throughout our sprint, our task board underwent many changes as Story 1 that was initially on our task board was removed as that story was not deemed necessary. We also added in another story as this removed story allowed us to fit in another user story which resulted in a projected story point completion of 25 story points for the current sprint.



User Story

- 🟩 Story #1: As Hanno (an admin), I want to be able to run the tool on a Mac or Unix device.
- 🟪 Story #2: As Anne (an admin), I want a command that will output a manual of all commands and how to use them
- 🟧 Story #3: As Hanno (an admin), I want a command to list of all planetary systems that have been updated in other catalogues since the last commit.

Add a card...

To Do

- 🟩 Every member does their own manual (man) page for each command within the tool: help, index, main, update
- 🟩 Create and maintain logs for each command implementation
- 🟩 Test each command's functionality and upload to Github
- 🟪 Implement command that returns a list of all commands and their functionality
- 🟪 Test the command and ensure that it works properly
- 🟧 Implement command to connect to the other catalogues and check for updates
- 🟧 Add to the command the ability to return a list of catalogues that have been updated after being checked
- 🟧 Code and test the updated command

Add a card...

In Progress

- 🟩 Able to run the tool on a Mac or Unix device.
- 🟩 Implement the tool via shell script

Add a card...

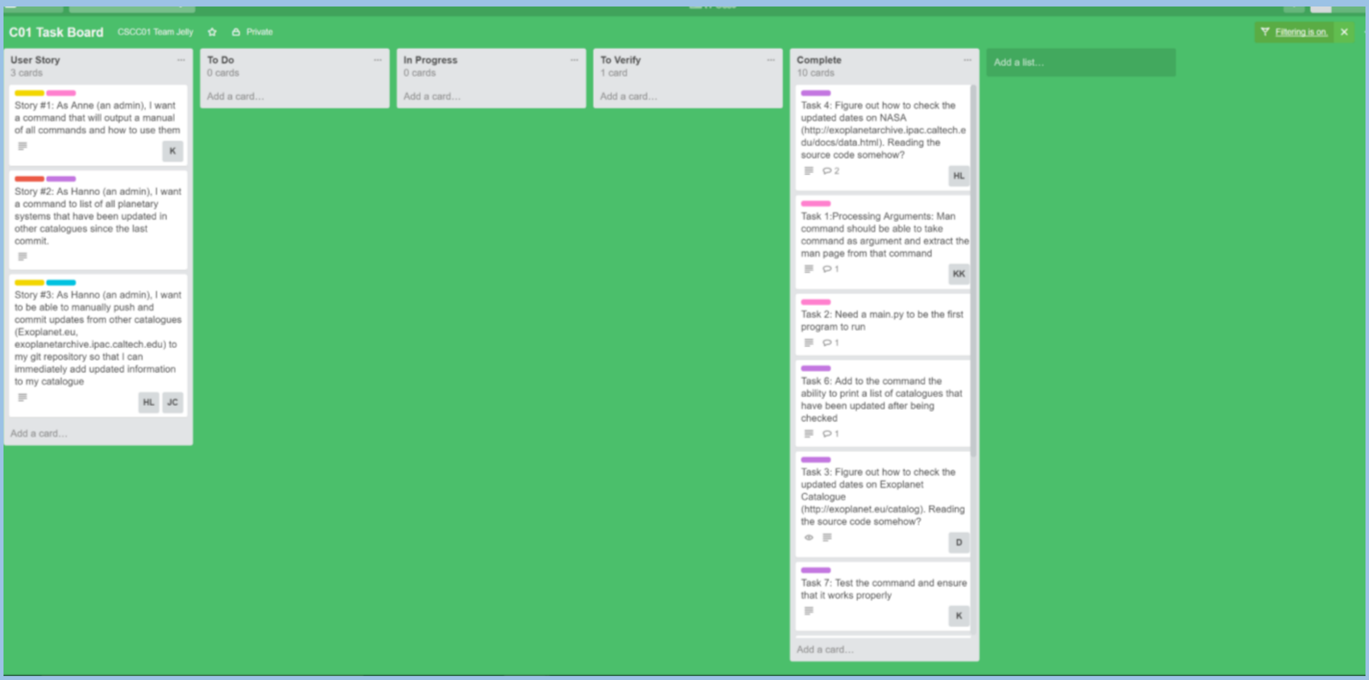
To Verify

Add a card...

Complete

Add a card...

Task board at the end of our first sprint



C01 Task Board CSCC01 Team Jelly Private

Filtering is on.

User Story
3 cards

Story #1: As Anne (an admin), I want a command that will output a manual of all commands and how to use them
K

Story #2: As Hanno (an admin), I want a command to list of all planetary systems that have been updated in other catalogues since the last commit.

Story #3: As Hanno (an admin), I want to be able to manually push and commit updates from other catalogues (Exoplanet.eu, exoplanetarchive.ipac.caltech.edu) to my git repository so that I can immediately add updated information to my catalogue
HL JC

Add a card...

To Do
0 cards

Add a card...

In Progress
0 cards

Add a card...

To Verify
1 card

Add a card...

Complete
10 cards

Task 4: Figure out how to check the updated dates on NASA (http://exoplanetarchive.ipac.caltech.edu/docs/data.html). Reading the source code somehow?
2
HL

Task 1:Processing Arguments: Man command should be able to take command as argument and extract the man page from that command
1
KK

Task 2: Need a main.py to be the first program to run
1

Task 6: Add to the command the ability to print a list of catalogues that have been updated after being checked
1

Task 3: Figure out how to check the updated dates on Exoplanet Catalogue (http://exoplanet.eu/catalog). Reading the source code somehow?
D

Task 7: Test the command and ensure that it works properly
K

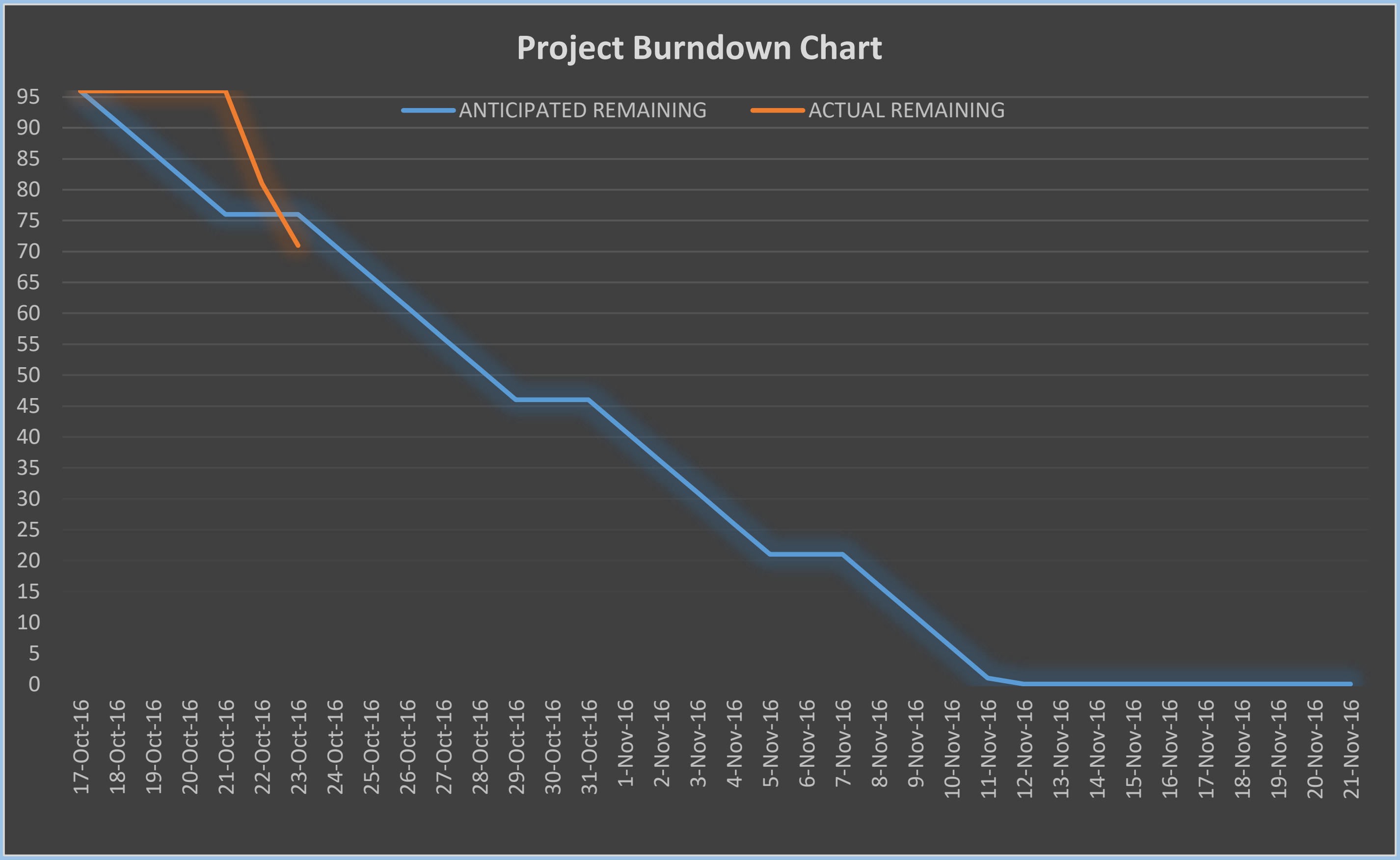
Add a card...

Add a list...

Burndown Chart

Our Estimated project velocity was 25 story points every sprint. We initially planned to complete 1 story point per developer each day over a 5-day work week to complete a total of 5 story points a week per developer. Our actual project velocity was still 25 points as we were able to accomplish our initial goal through the first week of our sprint.

After our first sprint, we have collectively completed 5 story points per developer but over a 2-day period as oppose to over a 5-day period. As a result, our burn down chart has a straight line the first 5-days and a much steeper slopes the last 2-days.


Project Burndown Chart
ANTICIPATED REMAINING
ACTUAL REMAINING
17-Oct-16
18-Oct-16
19-Oct-16
20-Oct-16
21-Oct-16
22-Oct-16
23-Oct-16
24-Oct-16
25-Oct-16
26-Oct-16
27-Oct-16
28-Oct-16
29-Oct-16
30-Oct-16
31-Oct-16
1-Nov-16
2-Nov-16
3-Nov-16
4-Nov-16
5-Nov-16
6-Nov-16
7-Nov-16
8-Nov-16
9-Nov-16
10-Nov-16
11-Nov-16
12-Nov-16
13-Nov-16
14-Nov-16
15-Nov-16
16-Nov-16
17-Nov-16
18-Nov-16
19-Nov-16
20-Nov-16
21-Nov-16