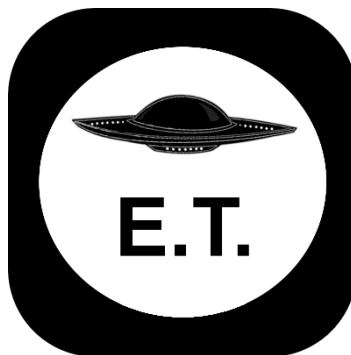


# Open Exoplanet Catalogue

Deliverable 5: Overview

By: Team 5



# OEC Table of Contents

|  |          |
|--|----------|
| <b>Additional Persona.....</b>   | <b>4</b> |
| 1.1 Assistant Ross Geller .....  | 4        |
| <b>Product Backlog.....</b>  | <b>4</b> |
| 2.1 Download Data from Other Catalogues .....                                | 4        |
| 2.2 Only Fetch Datafields in OEC.....  | 4        |
| 2.3 Convert Data from Catalogues into XML.....                               | 5        |
| 2.4 Merge Updates and Changes via Pull on Git .....                          | 5        |
| 2.5 Pull Request Notifications.....  | 5        |
| 2.6 Daily Update Check .....   | 5        |
| 2.7 Convert Data into Standard Units Used by OEC .....                       | 5        |
| 2.8 No Duplicate Updates .....   | 5        |
| 2.9 Manually Initiate Update Check.....                                      | 6        |
| 2.10 OEC Commit Messages .....   | 6        |
| 2.11 Data Value Change Threshold .....                                       | 6        |
| 2.12 First Synchronization .....   | 6        |
| 2.13 Handle Human Error while Updating Data.....                             | 7        |
| 2.14 Git Tutorial .....  | 7        |
| <b>Release Plan.....</b>   | <b>7</b> |
| 3.1 Sprint 1 .....   | 7        |
| 3.2 Sprint 2 .....   | 7        |
| 3.3 Sprint 3 .....   | 8        |
| 3.4 Sprint 4 .....   | 8        |
| 3.5 Sprint 5 .....   | 8        |
| 3.6 Sprint 6 .....   | 8        |
| 3.7 Sprint 7 .....   | 8        |
| <b>Evidence of System Validation .....</b>                                   | <b>9</b> |
| 4.1 Descriptiona and Evidence of System Validation Activities .....          | 9        |
| <b>Overview of Entire Project &amp; Deliverable 5.....</b>                   | <b>9</b> |
| 5.1 Brief Overview of Entire Project Saga.....                               | 9        |
| 5.2 How did the Estimated Project Velocity Change with each Deliverable..... | 9        |

|     |  |    |
|-----|--|----|
| 5.3 | How does work done for Deliverable 5 differ..... | 10 |
|-----|--|----|

# Additional Persona

## 1.1 Assistant Ross Geller

- 24 years, male.
- Graduate student at University of Toronto studying Astrophysics.
- Works part time (2 days per week) for Professor Cooper.
- Currently enrolled in 3 courses, and also involved in the Astronomy and Space Exploration Society (UTASX) at U of T.
- Uses Facebook, twitter on a daily basis.
- Good computer skills - capable of using computer software such as Microsoft Office and Google Drop box.
- Has not been exposed to highly technical software such as GitHub.
- Carries personal windows laptop, sometimes uses his friend's MacBook, but have never heard of Linux or Unix operating systems.
- Prefers intuitive software.
- Hardworking when involved in Astrophysics.
- Becomes frustrated when caught in technical difficulties such as a printer connection issue (i.e. no patience for technology)
- Capable of following instructions; Prefers tutorials for software/tools compared to learning by trial and error.

# Product Backlog

**Priority Scale (Low 1 - 5 High)**

**Cost in Story Points (1 Story Point = 1 Developer Hour)**

## 2.1 Download Data from Other Catalogues

**User Story ID: 4.1**

**Priority: 5**

**Cost: 3**

As Prof. Copper, I want to be able to download csv data from other catalogues from a given URL; so that it can then be used to generate updates.

## 2.2 Only Fetch Data fields in OEC

**User Story ID: 4.2**

**Priority: 5**

**Cost: 2**

As Prof. Cooper, I want to fetch data from other catalogues only if its data field exists in the OEC (i.e. there should not be an update notification if another catalogue updated its value for "Chance of Living Organisms" in Planet X, and the OEC does not have a corresponding XML data value to "Chance of Living Organisms."

## **2.3 Convert Data from Catalogues into XML**

**User Story ID: 4.3**

**Priority: 5**

**Cost: 6**

As Prof. Cooper, I want to automatically generate XML system pages (one system per XML page) for the OEC for data taken from the NASA Exoplanet Archive and Exoplanet.eu.

## **2.4 Merge Updates and Changes via Pull on Git**

**User Story ID: 2.9**

**Priority: 5**

**Cost: 15**

As Prof. Cooper, I want to be able to merge the updates/changes into the OEC via a pull request on GitHub. The program should generate a separate pull request for each XML system page generated.

## **2.5 Pull Request Notifications**

**User Story ID: 2.3**

**Priority: 5**

**Cost: 3**

As Prof. Cooper I would like to be notified via pull requests if an update has been generated (new XML system page created) from monitored catalogues so that I can choose whether or not to update the existing catalogue.

## **2.6 Daily Update Check**

**User Story ID: 2.4**

**Priority: 5**

**Cost: 3**

As Prof. Cooper I would like the system to be able to check for updates daily.

## **2.7 Convert Data into Standard Units Used by OEC**

**User Story ID: 2.7**

**Priority: 5**

**Cost: 5**

As Prof. Cooper, I want updates for the OEC to automatically convert data from the Nasa Exoplanet Archive and Exoplanet.eu into the standard units of measurement used by the OEC.

## **2.8 No Duplicate Updates**

**User Story ID: 2.6**

**Priority: 4**

**Cost: 8**

As Prof. Cooper, I only want to be notified ONCE when discrepancies exist between other catalogues and the OEC (and the other catalogue was updated more recently than the OEC) - unless the other catalogue's data containing the discrepancy is updated again later.

## **2.9 Manually Initiate Update Check**

**User Story ID: 2.5**

**Priority: 3**

**Cost: 2**

As Prof. Cooper I would like to be able to manually initiate an update check via terminal command.

## **2.10 OEC Commit Messages**

**User Story ID: 2.8**

**Priority: 3**

**Cost: 3**

As Prof. Cooper, I want updates for the OEC to contain commit messages that specify the reference URL for each update (i.e. the URL for that planet in either the Nasa Exoplanet Archive or Exoplanet.eu), and contain a hash tag identifying that the update was automatically generated by the program.

## **2.11 Data Value Change Threshold**

**User Story ID: 2.10**

**Priority: 3**

**Cost: 4**

As Prof. Cooper, I want the program to not generate updates if the only differences in data values are below certain predefined thresholds (to be provided by client/specialist TA and do not include changes below the currently displayed number of decimals).

## **2.12 First Synchronization**

**User Story ID: 2.11**

**Priority: 2**

**Cost: 5**

As Prof. Cooper, I want the first run of the program to check all planets in the other catalogues and create pull requests for all available updates for planets that have been updated more recently in the other catalogues than in the OEC. The first run should set a benchmark for future runs (i.e. monitor the dates that the planets were last updated in the other catalogues, so that no future updates are generated if the date of last update has not changed.)

### 2.13 Handle Human Error while Updating Data

**User Story ID: 2.13**

**Priority: 1**

**Cost: 15**

As Prof. Cooper, I want the program to try and identify data in other catalogues containing possible typos/human error (hypothetical example being "Kepler-16 b" is "Kepler 16 b") and attempt to match it with the corresponding XML page in the OEC. In the given example this would mean creating a pull request for "Kepler-16 b" (OEC name) instead of "Kepler 16 b" (Nasa name).

### 2.14 Git Tutorial

**User Story ID: 2.14**

**Priority: 1**

**Cost: 1**

As Rose Geller (Assistant of the Professor), I want a step-by-step instruction for basic operations of GitHub such as to clone, pull, merge, etc. so that I can use the program if needed.

## Release Plan

**Sprint Duration: 7 Days (including weekends).**

### 3.1 Sprint 1 (Deliverable 3)

**Oct 17 - Oct 23**

**Project Velocity: 11 story points (1 story point == 1 developer hour)**

The initial release 0.1 will contain basic program functionality (core functions upon which all future functions will be based). Release 0.1 will be able to download csv files from a given URL, and given a mapping of csv columns to XML tags will be able to generate a well-formatted XML page in the style used by the OEC (1 system per page) for each exoplanet in the csv file.

**User Stories Implemented:**

- 4.1 Download Data from Other Catalogues
- 4.2 Only Fetch Data fields In OEC
- 4.3 Convert Data from Catalogues into XML

### 3.2 Sprint 2

**Oct 24 - Oct 30**

**Project Velocity: 18 story points (1 story point == 1 developer hour)**

Release 0.2

**User Stories Implemented:**

- 2.9 Merge Updates and Changes via Pull on Git
- 2.3 Pull Request Notifications

- 3.3 Sprint 3**  
**Oct 31 - Nov 6**  
**Project Velocity: 14.5 story points (1 story point == 1 developer hour)**  
Release 0.3  
**User Stories Implemented:**  
2.7 Convert Data into Standard Units Used by OEC  
2.4 Daily Update Check
- 3.4 Sprint 4 (Deliverable 4)**  
**Nov 7 - Nov 13**  
**Project Velocity: 14.5 story points (1 story point == 1 developer hour)**  
Release 0.4  
**User Stories Implemented:**  
2.12 Handle Alias while Updating Data  
2.6 No Duplicate Updates
- 3.5 Sprint 5**  
**Nov 14 - Nov 20**  
**Project Velocity: 9 story points (1 story point == 1 developer hour)**  
Release 0.5  
**User Stories Implemented:**  
2.10 Data Value Change Threshold  
2.5 Manually Initiate Update Check  
2.8 OEC Commit Messages
- 3.6 Sprint 6**  
**Nov 21 - Nov 27**  
**Project Velocity: 20 story points (1 story point == 1 developer hour)**  
Release 1.0  
**User Stories Implemented:**  
2.11 First Synchronization  
2.13 Handle Human Error while Updating Data
- 3.7 Sprint 7 (Deliverable 5)**  
**Nov 28 – Dec 1(4 Days)**  
**Project Velocity: 1 story point (1 story point == 1 developer hour)**  
Release 1.1  
**User Stories Implemented:**  
2.14 Tutorial



Our Release Plan has slightly changed as we have deleted the user stories *Delete Catalogue* and *Specify Catalogues to Monitor* from Sprint 6.

## Evidence of System Validation

### 4.1 Description and Evidence of System Validation Activities

Our team have ensured that we have built the “right product” by sending emails to our client (Professor Hanno) and also attending his office hours to present our software demo. Each feature that was implemented closely matches the requirements provided by Professor Hanno. For example, we have implemented update notifications via Git pull requests and also kept the installation process of our software to be as simple as possible (as specified by Prof. Hanno). As a result, our team have verified that the software we are currently developing meets the needs and expectations of our client.

## Overview of Entire Project & Deliverable 5

### 5.1 Brief Overview of Entire Project Saga

Our team used the first two deliverables to get to know each other and also to come up with personas and user stories for our program. In deliverable 3, we started the implementation of our software with basic features such as downloading the csv files and creating mappings from the monitored catalogues. Deliverable 4 was where our team completed most of the core features of our program such as the GitHub operations. Our team was stuck on the development of this section of the program for a period of time. During this time, we switched from using Hub to the GitHub API. Eventually, we were able to tackle the technical problem and continue with our development process. By this time, our group already had the core features implemented for the OEC Synchronization software. In Deliverable 5, our team continued development process by implementing features that had a lower priority such as the feature to handle human error while updating data.

### 5.2 How did the estimated project change with each deliverable?

The estimated project fluctuated from approximately 10 to 20 story points depending on the workload for the group members during each sprint. When there were many assignments due for other courses during the week, we would estimate a comparatively lower project velocity. On the other hand, when group members had more time to work on the project during the week, we would estimate a relatively higher project velocity.

**5.3 How does the work done for deliverable 5 compare to the work done on all previous deliverables, both in terms of progress and end result?**

In deliverable 5, we begin to wrap up the development of our program by implementing features that are of lower priority compared to those features implemented in the earlier deliverables. There were six user stories implemented, three of which are of priority 3 and one of priority 2 and two of priority one. Progress and end result are similar to the earlier deliverables where our group was able to implement all user stories planned.