



Precision Development

DELIVERABLE #5

University of Toronto Scarborough

**Authors: Brandon Chow, Daniel Persaud, Jason Zheng,
Kenny Lam, Wilfred Wong**

Date : Nov 24, 2015

Table of Contents

1 CURRENT STATE OF PROJECT.....	3
2 PRODUCT BACKLOG.....	4
3 SPRINT #3.....	7
3.1 SPRINT BACKLOG.....	7
3.2 SPRINT TASK BOARD/PLAN.....	7
3.3 SPRINT BURNDOWN CHART.....	7
4 SYSTEM DESIGN DIAGRAM.....	8
5 TESTING STRATEGY.....	9
5.1 TEST APPROACH.....	9
5.1.1 Unit Testing.....	9
5.1.2 Regression Testing.....	9
5.2 LIMITATIONS.....	10
5.3 EXAMPLE OF TEST SUITE/VALIDATION STRATEGY (COUPLING/LINKING DOCUMENTS).....	10
5.3.1 Validation Strategy.....	11

1 Current State of Project

As of November 24th, 2015 our team has implemented the following major features: batch tag editing, adding custom fields, coupling documents, citation editor (CSL) have been completed, however, testing is still in progress. The Batch Editing feature supports adding, deleting, modifying and merging of tags for the user's entire library, collection, or a list of items that the user chooses to add to the batch editing queue. Adding custom fields is done by item type, the user provides an item type (i.e. book), a field type (text, number, date), and field name, the custom field is then added to all items of the specified type. The coupling documents feature will link a 'master' document to one or several 'slave' documents in group libraries; any changes made to the main fields in master document will propagate to the slave documents. Note that for security and design reasons notes and custom fields are not able to be coupled. The citation editor currently allows the user to create custom citation formats with an easy to use GUI and it also provides a live preview as it is being created.

In addition all of the above features are integrated seamlessly into the Zotero Firefox application as a plugin as well as the Zotero standalone client. We designed the GUI to mimic Zotero's native GUI so that the users will already have a good feel of how the tools work. Installation of our plugin is as easy as installing an add-on in Firefox (from our provided xpi file). We also have a really helpful user-documentation at ZotPie.readthedocs.org (still in progress) in addition to a developer documentation for our plugin (also on ZotPie.readthedocs.org). Most importantly, as per client specifications, all of this can be done offline! We have also not modified any of Zotero's source code, which means that Zotero can safely be updated without breaking our plugin. A standalone python batch editor and backup tool for Zotero is also available for users and developers because we switched from python to JavaScript, however, they require internet connectivity to be used.

Story ID's 6, 8, 9, 10, 11, and 12 were dropped due to lack of time.

****Note: There were no changes done to the user story content/personas.**

2 Product Backlog

Product Backlog				
User Story ID	User Story	Value	Risk	Estimated Cost
1	As Anne a professor, I want to be able to batch select tags so that I can perform deletes/renames/and merges on all of them at once.	3	Med High	20 30 Dev Hours
2	As Anne a professor, I want the option to couple documents so changes made to one instance of a record will be reflected in the copies.	4	Med High	20 30 Dev Hours
3	As Britney a student, I want to be able to add more fields to an existing record type so that I can classify the record better.	4	Med	40 Dev Hours
4	As Anne a professor, I want an easy-to-use UI for creating custom bibliography styles so that I don't have to fiddle with XML files.	4	High	20 30 Dev Hours

5	As Britney a student, I want the ability to work offline so I can work while commuting.	1	High	100 Dev Hours
6	As Anne a professor, I want to have a log of the changes I've made so that I can keep track of my progress.	2	Med	30 Dev Hours
7	As Britney a student, I want to be able to export custom bibliography template styles so that I can share it with others.	2	Low	40 Dev Hours
8	As Britney a student, I wish to import my records from other bibliographic tools that I have previously used.	1	High	30 Dev Hours
9	As Anne a professor, I want a notification feature so that I can come back to records that are incomplete.	3	Med	30 Dev Hours
10	As Anne a professor, I'd liked to be able to attach a photo to my records so that I can visually identify a record by the book cover.	1	Low	20 Dev Hours

11	As Britney a student, I would like a feature to automatically create and update a local backup of my records so that I don't need to worry about losing my work.	3	Low	10 Dev Hours
12	As Anne a professor, I'd like to have email integration so that I can share my work with my colleagues without opening another email client.	1	Med	30 Dev Hours

3 Sprint #3

3.1 Sprint Backlog

Our sprint backlog is located here:

- User Stories/Tasks that were swapped out are colored in red
- User Stories/Tasks that were swapped in are colored in green

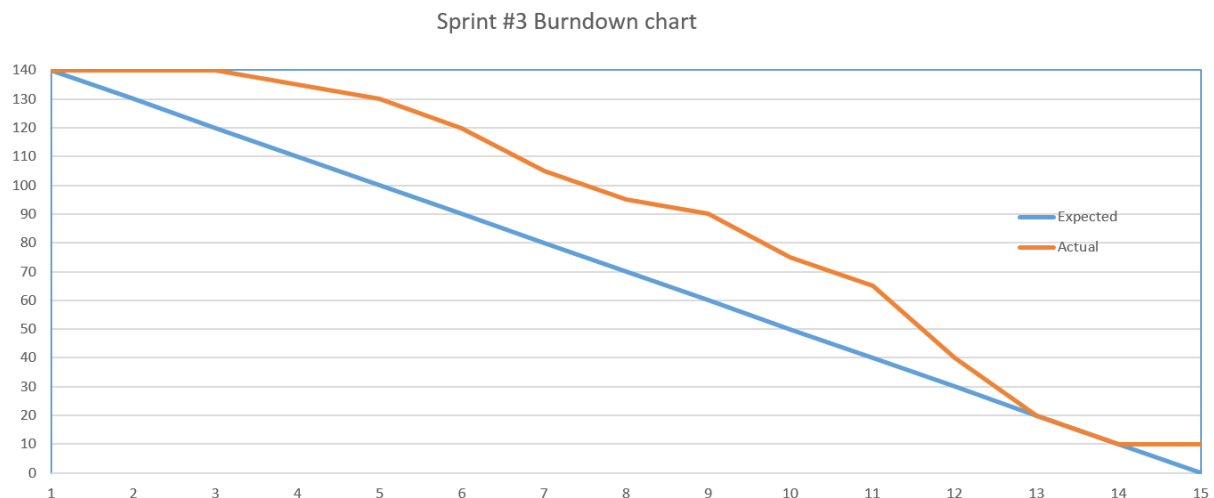
<https://docs.google.com/spreadsheets/d/1x4ZO0lsaXkctc8lxdIhvAAT5QEFSecOz8eK0PieLRWok/edit?pli=1#gid=1883449992&vpid=A1>

3.2 Sprint Task Board/Plan

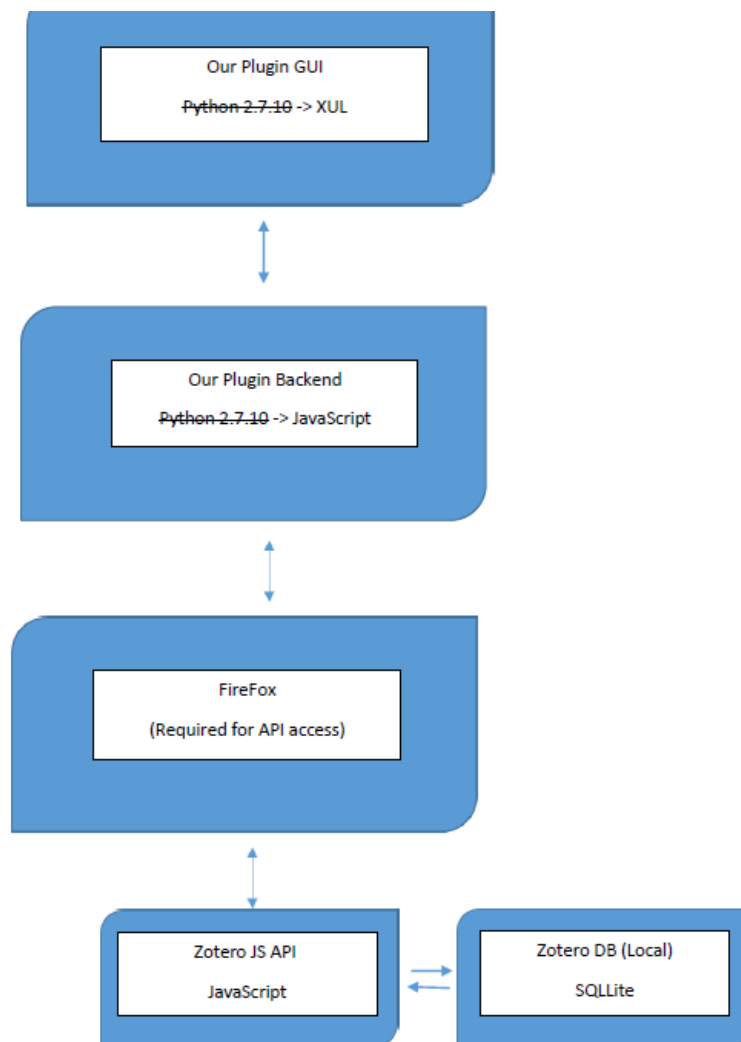
Our task board is located here (snapshots are located under “INSTRUCTIONS”):

<https://trello.com/b/KeNPWB3W/sprint-3-task-board-precision-development>

3.3 Sprint Burndown Chart



4 System Design Diagram



System Design Explanation:

- Our Plugin GUI now uses XUL files instead of Python 2.7.10, just as before the user will interact with this part of our plugin to perform tasks.
- Our Plugin Backend has changed from Python 2.7.10 to JavaScript to better interact with the JavaScript API provided by Zotero. It takes and processes input from the GUI and through firefox interacts with the JavaScript API.
- FireFox, needed to expose Zotero Internals (i.e. the JavaScript API) to our backend for use.

-
- Our Plugin now uses the JavaScript API instead of the PyZotero API, this has the advantage of making changes locally. The Zotero Online DB will be updated with any changes made offline, or allow the user resolve any conflicts between the offline DB and the Online DB in a similar way to git or other version control systems (this is a built in Zotero function).
 - The Local Zotero database (DB) is a SQLite database that Zotero Standalone/Firefox/Chrome uses to store all user data (through the JS API).

5 Testing Strategy

Zotpie is an addon developed by Precision Development that provides tools to Zotero and extends on previously existing tools to Zotero.

Testing will cover:

- Functionality of batch tag editor and future iterations to ensure main functionality remains bug free.
- Regression on custom field editor and document linker and if these components remain working upon future updates of Zotero.
- Functionality of CSL editor and the correctness of the exported data.

5.1 Test Approach

Type of testing	Reasoning	Tools
Unit Testing	Test individual modules to ensure working state of units of the program.	JUnit
Regression	To ensure future iterations or updates of Zotero does not break functionality.	Selenium, Jenkins-CI
Acceptance Testing	To ensure that the product meets the required specifications outlined by the customer.	Selenium, Manual

5.1.1 Unit Testing

Developers are required define a set of unit tests prior to developing working code to follow Test Driven Development procedures. Unit testing can be done through JUnit, a JavaScript unit testing framework.

5.1.2 Regression Testing

Regression testing will be done in an automated form using Selenium and Jenkins-CI. This allows developers to catch bugs early on. After each major unit of work, a test suite should be run against the up to date codebase to ensure stability. The regression tests should also be run on each new Zotero update to ensure that the latest build of Zotero does not conflict with our current version of Zotpie.

5.2 Limitations

Due to that lack of time, resources and the expertise, covering all planned testing will not be feasible. Given the time planned, we cannot develop test suites to cover all code implement test tools, such as Selenium and Jenkins-CI, and provide proper maintenance for these tools. Also, expertise in JavaScript, web programming and addon development, prior to the project, was very limited. Research on various testing frameworks and how to implement and develop test suites for such frameworks was a new experience for most of the developers.

5.3 Example of Test Suite/Validation Strategy (Coupling/Linking documents)

Description	Check if unlinking a master and a slave stop change propagation
Setup	<ol style="list-style-type: none">1. Create a record in a collection in user library2. Create 2 records in a collection in a group library
Instructions	<ol style="list-style-type: none">1. Open link records utility2. Click on Unlink Records tab3. Chose record in user library from the collection record was made in setup4. Right list should have 2 records to choose from the group library that was created in setup5. Select one, then press unlink6. Go to Zotero main window, select the record in user library7. Change the Title to "my book"8. Check both of the records in group library
Expected Result	- Record that is unlinked should not have a title, record that is still linked should have title my book

Description	Checks if linking a master and a slave propagates changes from the master, not the slave
Setup	<ul style="list-style-type: none"> • Create a record in a collection in user library • Create 2 records in a collection in a group library
Instructions	<ol style="list-style-type: none"> 1. Open link records utility 2. Click on Link Records tab 3. Chose record in user library from the collection record was made in setup 4. Chose record in group library from the collection record was made in setup 5. Press Link 6. Go to the Zotero main window, select the record in user library 7. Change the Creator name to "John Doe" (first name John, last name Doe) 8. Change the Title to "My book" 9. Check the group record that was linked 10. Go through steps 3-9 for the other record created, with a different Creator name and title (Mary Sue, my binder) 11. Go to any of the 2 group items, change Creator name to "Foo Bar"
Expected Result	<ul style="list-style-type: none"> - After first iteration of steps 3-9, the first group item should have creator John Doe and title book. - After second iteration, both group items should have the creator Mary Sue and title my binder - The Name of the Master should still be Mary Sue, Foo Bar should stay the same

Description	Check if deleting an item will get rid of the links
Setup	<ol style="list-style-type: none"> 1. Create a record in a collection in user library 2. Create 2 records in a collection in a group library 3. Link the record in user to the 2 records in group
Instructions	<ol style="list-style-type: none"> 1. Go to Zotero main window, select the record in user library 2. Delete it completely (as in from the trash can) 3. Go to link records utility, go to the Unlink Records tab 4. There should be no record to select on the left list 5. Go to link documents tab, check the collection the record in user library belong to under "Collection of Original:", the record deleted should not exist
Expected Result	<ul style="list-style-type: none"> - Original record should be gone from selection, linking the 2 group records should not raise "these items are linked" flags.

5.3.1 Validation Strategy

Couple documents should be able to propagate changes from any one user record to how ever many different group records necessary. This is necessary from our requirements of the user story. However, for security of the user's info, any changes to groups should not affect the master copy. Therefore, the first acceptance tests makes sure that a) changes can be made to multiple slaves b) changes in slaves are not reflected in the master. The behavior of unlinking documents is very straight forward, undoes all linking so that none of the propagation happens anymore, this is evident in the 2nd acceptance test. The final acceptance tests is in the case the user deletes a master record that is linked to several group items. The change should be that all of the links should dissipate and not show on the linked documents utility unlink records tab. By these 3 acceptance tests, we cover the base functionality of the required coupling of user library records to group records.