

# **CSCC01 Project Deliverable #3**

## **Team Name**

SQL Injectors

## **Team Members**

Nadeem Haidar, Andres Moreno, Junaid Patel,  
Yujian Chen, Minsoo Park

## **Team #2**

**October 27, 2015**

## Sprint / Product Backlog and Task board Snapshots

- Can be found on the GitHub repository in the planning folder

### First Sprint Plan

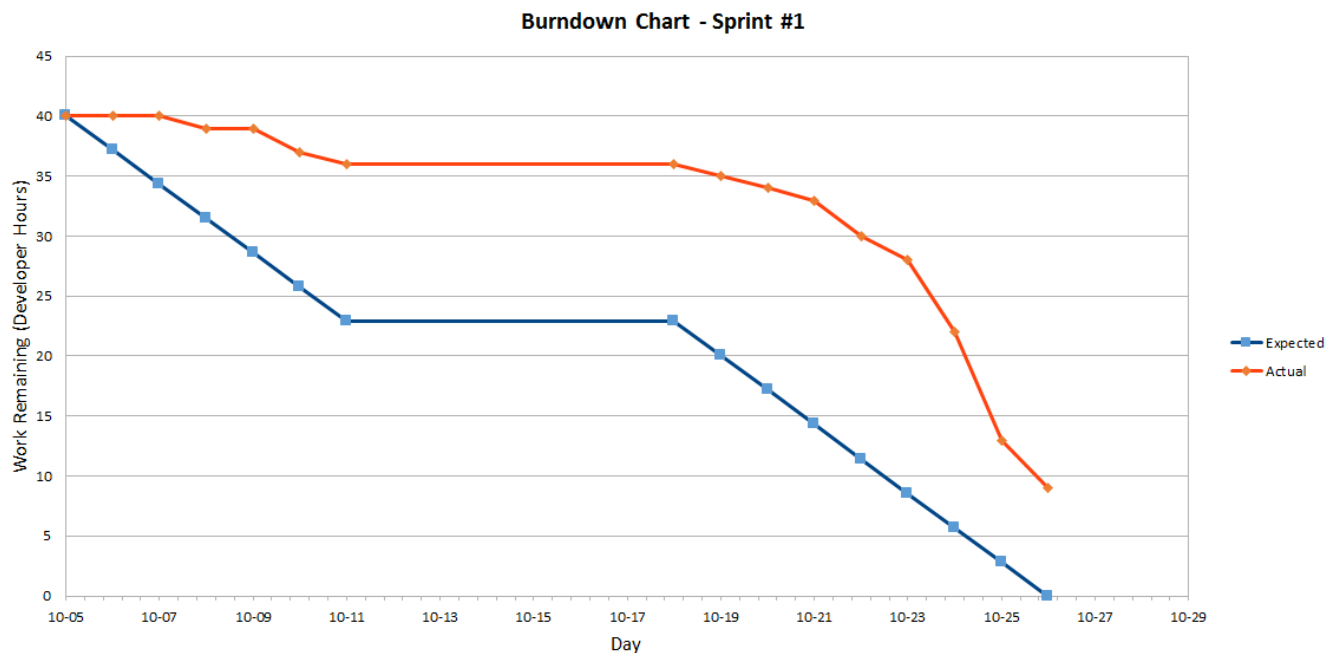
**Initial Plan:** We planned a schedule of around 4 hours each of work per week. The details on which tasks we would work on each day are below. The numbers in the brackets specify what task was being worked on, the list of tasks are visible on Trello.

Date	Nadeem	Junaïd	Andres	Minsoo	Alex
Start	0	0	0	0	0
Oct. 6	0	0	1 (t1)	0	0
Oct. 7	0	0	0	0	0
Oct. 8	1 (t7)	0	0	0	1 (t4)
Oct. 9	2 (t7) + 1 (t8)	0	0	0	1 (t4)
Oct. 10	0	0	1 (t3)	0	0
Oct. 11	0	0	1 (t3)	0	0
Oct. 19	0	1 (t1) + 1 (t2)	1 (t3)	2 (t7)	0
Oct. 20	0	2 (t2)	0	1 (t7) + 1 (t8)	0
Oct. 21	0	0	0	0	1 (t5)
Oct. 22	1 (t8)	0	1 (t3)	2 (t8)	1 (t5)
Oct. 23	3 (t8)	0	1 (t3)	0	2 (t5)
Oct. 24	0	0	1 (t3)	0	0
Oct. 25	0	1 (t2)	1 (t9)	1 (t8)	2 (t6)
Oct. 26	0	1 (t2) + 2 (t3)	0	1 (t8)	0

**Log of Work:** We were not able to follow this plan. This was mostly because we had not taken the time to research how Zotero plugins work and we did not account for the time that would take. Because of this most of the time that was spent was just researching and getting a basic application working. For future sprints we have learned that we need to include every task such as researching and we also cannot be as optimistic as we were on the time estimates. Below is a daily log of when we were actually able to work on it.

Date	Nadeem	Junaid	Andres	Minsoo	Alex
Start	0	0	0	0	0
Oct. 6	0	0	0	0	0
Oct. 7	0	0	1	0	0
Oct. 8	0	0	0	0	0
Oct. 9	1	1	0	0	0
Oct. 10	0	0	1	0	1
Oct. 11	0	0	1	0	0
Oct. 19	0	0	0	1	0
Oct. 20	0	0	0	1	0
Oct. 21	0	1	0	0	1
Oct. 22	2	0	0	1	1
Oct. 23	1	1	0	0	1
Oct. 24	3	1	1	0	0
Oct. 25	1	2	3	2	3
Oct. 26	1	1	1	1	0

## Burndown Chart



None of us were able to work on it during reading week, so we did not allocate time during it.

## Current State of the Project

The current state of the project is a little behind what we were expecting but some progress has been made. We were not able to follow the original plan as well as we thought we could and had to have an urgent meeting to discuss the plan for this sprint and make some changes to our current sprint. This meeting removed some user stories from our current sprint and added several other tasks we needed to work on so that we can make more significant progress.

We had to make several changes to our original Sprint and Product Backlog. Originally we had planned to complete 3 user stories and start working on another in this sprint. However, after working on the development we realized this was too ambitious. One of the main issues was that we did not account for the large amount of time that we all had to spend in researching how to develop Zotero plugins and learning how all the components work. Also, after doing research into how to develop custom item types, we realized this was a very difficult and could not be completed in a reasonable amount of time (due to having to directly modify the database and have the new item type work with the output styles).

Due to these issue we decided to have a Skype meeting on Saturday, October 24th to revise our plan. In this meeting we decided to remove user stories: **A02**, **A07**, and **A01** from our sprint backlog, because they were very hard to implement as explained above. Also we realized we needed to add in a task to research how to develop Zotero plugins. We also added in user story **B02** to our sprint Backlog since it was related to what we were working on for B01 and could reasonably be accomplished. After making this change, this improved and gave us more realistic goals to complete.

The largest problem we encountered was in figuring out how to start developing Zotero plugins and getting everything set up. This was largely because the hello world example had a lot of files that fit together in a certain way and we needed to understand how it all worked and because there is no full documentation of the API. Once we all had a clear understanding of how it worked, we were able to create an initial application and speed up our development. For future sprints we have learned that we need to split up the user stories into smaller more detailed tasks and include everything that is needed to be done including researching and other tasks such as the deliverable.

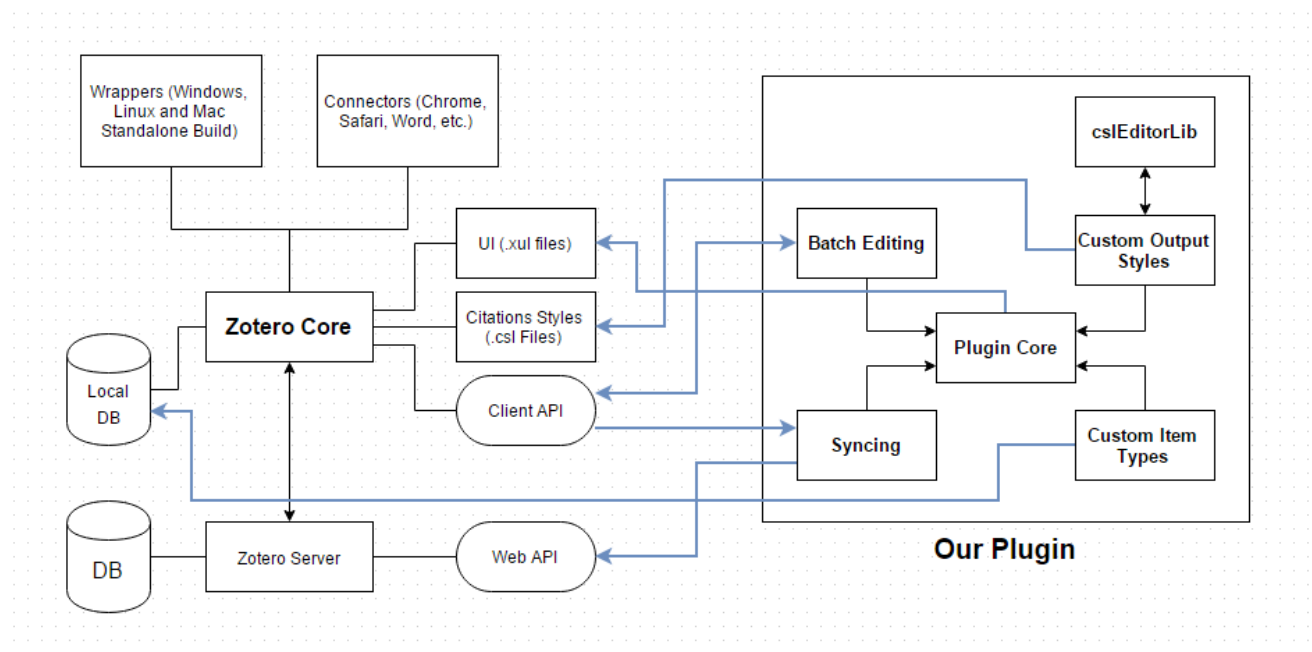
**A01:** As Natalie, a researcher, I want to be able to add custom fields to a specific record type, so that I can keep track of information to the article that I can't otherwise keep track of

**A02:** As Natalie, a researcher, I want to be able to add new record types with user defined fields, so that I can add sources that do not fit into any existing record type

**A07:** As Natalie, a researcher, I want to be able to name the record types I create and for them to appear in the list of record types, so that they will be easier to find

**B02:** As John I want to be able to select several entries and add a tag to all of them, so that I can easily add tags to a group of related entries.

## System Design



## Zotero

The Zotero Core contains all the logic and UI code for Zotero, and is a Firefox plugin. The windows, Linux and Mac versions are standalone wrappers of Zotero, so they use for the most part the same code and GUI.

The client side has access to a local database that stores item types, entries and other configurations.

It also stores a list of .csl files each of which specify a citation style.

The client has a JavaScript client API that allows you to interact with the user's entries, and the local database.

The Zotero core communicates with the server to sync the user's record types and for the group library feature.

The server has a database that stores the user's entries, their info, and group libraries.

The server has a web API that allows you to communicate with it, with it you can make read and write requests to the database and get other information such as the field types a specific item type has.

## Our Plugin

Our plugin will be divided in 4 main components: batch editing, syncing, custom item types, and custom output styles. Each of these components will be used in the main core that will handle initialing the Zotero connection. Our user interface will be defined in files for each component and a main UI that will be inserted into Zotero's UI file.

The batch editing component will handle all the API calls to batch edit tags and will only be working with the client API.

The syncing will handle updating items in group libraries when a user edits their local copy. It will be notified by the client that a change was made and will then update the version on the group by contacting the Web API.

The Custom Output Styles component will take advantage of an open source library [cslEditorLib \(https://github.com/citation-style-language/csl-editor\)](https://github.com/citation-style-language/csl-editor) to create and customize the .csl files and then it will add them to the list of files Zotero uses.

The Custom Item Types component will have to directly connect to the local database to make the changes since it is not able to be done through an API. The impact this will have on group libraries and if it an outputting the entry will have to be further researched.