COS 333 Project Overview Proposal
16 March 2012

Project Title: PAL (Princeton Academic Linker)
Project Manager: Wenley Tong

Group Members:
- Masha Okounkova (mokounko@princeton.edu)
- Prerna Ramachandra (pramacha@princeton.edu)
- Wenley Tong (wenleyt@princeton.edu)

## Overview

Princeton students presently have a number of different websites they may need to visit in order to keep track of everything going on in their academic lives, not to mention anything else they may do. Blackboard, ICE, SCG, and (in some departments) Piazza are sites that host useful and sometimes necessary information regarding courses. Needing multiple windows or tabs just to do everything class-related can be annoying, and the interface for Blackboard has many unused features that simply clutter the window. PAL will be a Chrome Extension for students users of Blackboard and other class-related sites for Princeton undergraduates.

PAL will reorganize most content of Blackboard into a single window, allowing for easy access to desired . This will include access to documents posted in Blackboard, announcements posted by professors, discussion boards, and possibly grades.

Having a common interface to all of these websites which is simple and provides convenient access to the information stored on them will help students better manage their resources and time to store retrieve this information.

## Functionality

An interface to improve access to and interaction with Blackboard.

Student User:
- Masha wants to just check in on her classes. Upon logging in to Blackboard, instead of the standard Blackboard page, Masha will be presented with a newly rendered main page that will show all changes / updates since she last logged in (e.g. new announcements, assignment postings). Masha reads some of them and sees that the assignment and reading list for the week in COS 595 has been posted. She clicks on the COS 595 tab on the left will be a side bar and goes to the course's page. Masha sees the new assignment in the Assignments widget and clicks the link to the document; it starts downloading right away.

- Prerna wants to check the discussion board for ENG 617 to see if anyone has answered her question regarding post-Revolutionary French allegories. She logs on to Blackboard and sees

on the main page that there is a new post, thanks to PAL. After clicking on the ENG 617 tab, she goes to the Discussion Board widget and submits her own post.

The main body will contain two columns of widgets. A widget is a representation of a feature of Blackboard (e.g. Assignments, Announcements, Course Materials, Discussion Board). Widgets are resizable and internally scrollable. Preferences for widget layout will stored in both

Faculty User:
- Will continue to use Blackboard as they have in the past. This is definitely a branch that could be explored to expand the system, but is not within the intended scope of the project. Future improvements to the system? Currently let's just focus on student interfaces, since we are most familiar with the issues they face.

## Design

The PAL extension will require two phases of operation: Mining and Rendering.

- Mining phase. Use a combination of regex's, string searches, and XMLHttpRequests, get all relevant user content (Assignments, Course Materials, Syllabus, Course Description, Announcements, etc) and store into a well-organized Course object (with helper objects like Instructor and Documents).
- Rendering phase. Use HTML, CSS and more Javascript to present a nicely formatted, centralized portal to all of the user's BB information. The fewer changes of pages, the better, as javascript will re-run fully every time the user moves to a new URL. Ideally, we can control the rendering of the page entirely and keep the user in the same real location.

We decided to use a private GitHub repository for version control. Where not otherwise specified, we will use Python with Django to build the website components.

No administrative system is necessary through the browser because almost all the information originates in Blackboard.

During development, the extension will the housed on our local machines and tested on our own local browsers.

## Milestones

- (DONE -- Mar 16th) All team members have working environments to write, test, and run code on their local machines.
- Get whitelisted by CAS.
- Acquire a domain. (Preferably pal.tigerapps.org).
- (DONE -- Mar 19th) PAL redirects the user to CAS where she logs in. Upon returnign to PAL, she is presented with a page with their ticket number and/or her netID. This represents a fully CAS-interacting system.

[ Timeline reoriented -- Switch to Chrome Extension ]

- Print all content (names and links of documents, links to other places, folder links to other

pages) to the console.
- Store all information in OOP-class data structures before printing to console
- Present user with basic formatted GUI and full content excluding Tools.
(- Begin UI testing.)
- Store information in browser extension cache to avoid repeat mining of same content and slow performance.
- Present user with dynamically changing GUI and content. Probably implement with more javascript.
- Add Tools.

## Risk + Open Issues

- Sending information back to Blackboard. Some classes will have assignments submitted via Blackboard. Implementing this feature in PAL will require careful manipulation of the original Blackboard pages on top of the passive mining required for everything else.
- Copy-righted features. Related to the previous issue, Blackboard has the capability of only allowing some computers access to video reserves. We do not know how it does this, though hopefully this won't be a problem since PAL piggybacks on authenticated browsers.
- Incorporating the Tools.