1. Classification of Components

Module Description

Google Authentication

Module in charge of setting up login credential for interface with Google Calendar API.

Inputs: User's Google Account login credentials.

Outputs: A confirmation response from Google OAuth server with authentication token.

Purdue Authentication

Module in charge of setting up login credential for interface with Purdue Calendar API.

Inputs: User's Purdue Career Account login credentials.

Outputs: A confirmation response from Purdue SAML server with authentication token.

Google Calendar

Module in charge of interfacing with Google Calendar API and adding/creating events on the user's Google Calendar.

Inputs: JSON of user's classes and events to be added onto user's Google Calendar.

Outputs: A confirmation response from Google OAuth server with authentication token. Along with an error response if any errors may occur

Purdue Calendar

Module in charge of interfacing with Purdue Calendar to fetch user's classes and exams.

Inputs: User's Purdue Career Account authentication token.

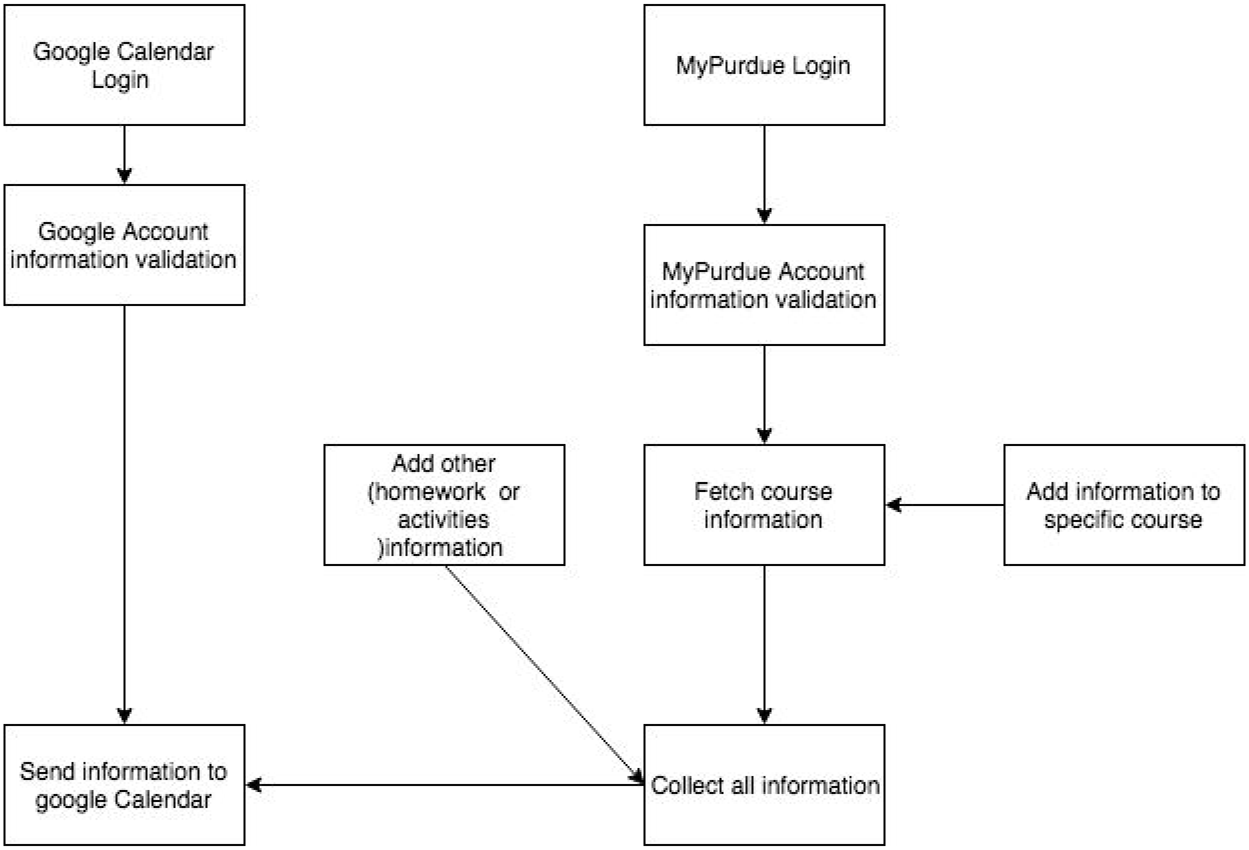
Outputs: A confirmation response from Purdue server with the user's classes and exams in a JSON.

Site UI

Module in charge of all the webpages and UI for login and calendar manipulation.

Inputs: User's account credentials and event information.

Outputs: Displays the calendar to be exported to Google Calendar.


Google Calendar Login
Google Account information validation
MyPurdue Login
MyPurdue Account information validation
Add other (homework or activities)information
Fetch course information
Add information to specific course
Send information to google Calendar
Collect all information

Incremental Testing Technique

We choose to follow bottom-up incremental testing. The first reason is building drivers for the modules in our software is comparatively easier than building stubs. We will have less cost to test than top-down incremental testing. The second reason is high risk modules are lower level. Most of our function depends on the connection and communication between different servers. The correctness of the lower level data processing is more significant than high level. It will be better to find the bugs in the lower level first and move to testing the top level to minimize our risk. The third reason is our development plan is front end develop after backend. Bottom-up incremental test the backend part first and the frontend part which perfectly fit our development plan.

2. Incremental and Regression Testing (75 points)

Automation

Not yet implemented.

Incremental Testing Defect Log

Module	Google Authentication		
Defect No.	**Description**	**Severity**	**How To Correct**
0	Google OAuth API needs whitelisted domain and redirect URI.	1	Whitelist localhost on Google Dashboard and run python web server

Module	Purdue Authentication		
Defect No.	**Description**	**Severity**	**How To Correct**
1	Authentication request was denied by Purdue’s CAS authenticator.	1	The application was submitted to be registered with ITAP, to get clearance to use the CAS authentication server.
2	Temporary authentication was not redirecting back to Boiler Schedule homepage.	2	The CASAuthenticator object was provided with a “bounce_redirect” URL.
3	The CASTGC cookie and authentication ticket were unavailable after temporary authentication.	2	The ticket was stored, and used to update the user’s “Logged In” status for their Purdue account.

Module	Google Calendar		
Defect No.	**Description**	**Severity**	**How To Correct**
0	Application should not crash if event contains special characters	2	Create an special character remover.
1	Application should be able to handle null events	1	Make sure that all inputs have pre-existing default values
2	Application should be able to store results temporarily in case of technical difficulties	1	Make a “session storage” that allows for data to be stored temporarily

Regression Testing Defect Log

Module	Google Authentication		
Defect No.	**Description**	**Severity**	**How To Correct**
0	Handling of redirect URI response error idpiframe_initialization_failed	2	Check domain URIs in Google Console and reset browser cache

Module	Purdue Authentication		
Defect No.	**Description**	**Severity**	**How To Correct**
1	Fixing redirection after temporary authentication.	2	Created a validator for the CASAuthenticator object before it’s use to make sure that the

			object has necessary redirection URLs.)
2	Storing CASTGC cookie and authentication ticket after temporary authentication.	2	Created a validator in the auth module to export the ticket after authentication. (Behaviour dependent on ITAP approval of application).

Module	Google Calendar		
Defect No.	**Description**	**Severity**	**How To Correct**
0	Special character check	2	Make sure that there are grammatical checks that remove the characters
1	Null object check	1	Assign all DOM’s values when initialized
2	Store fields in case of crash	1	Use session storage to allow for temporary memory