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 from Google Chrome

Outputs: A confirmation response from Google Chrome 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.

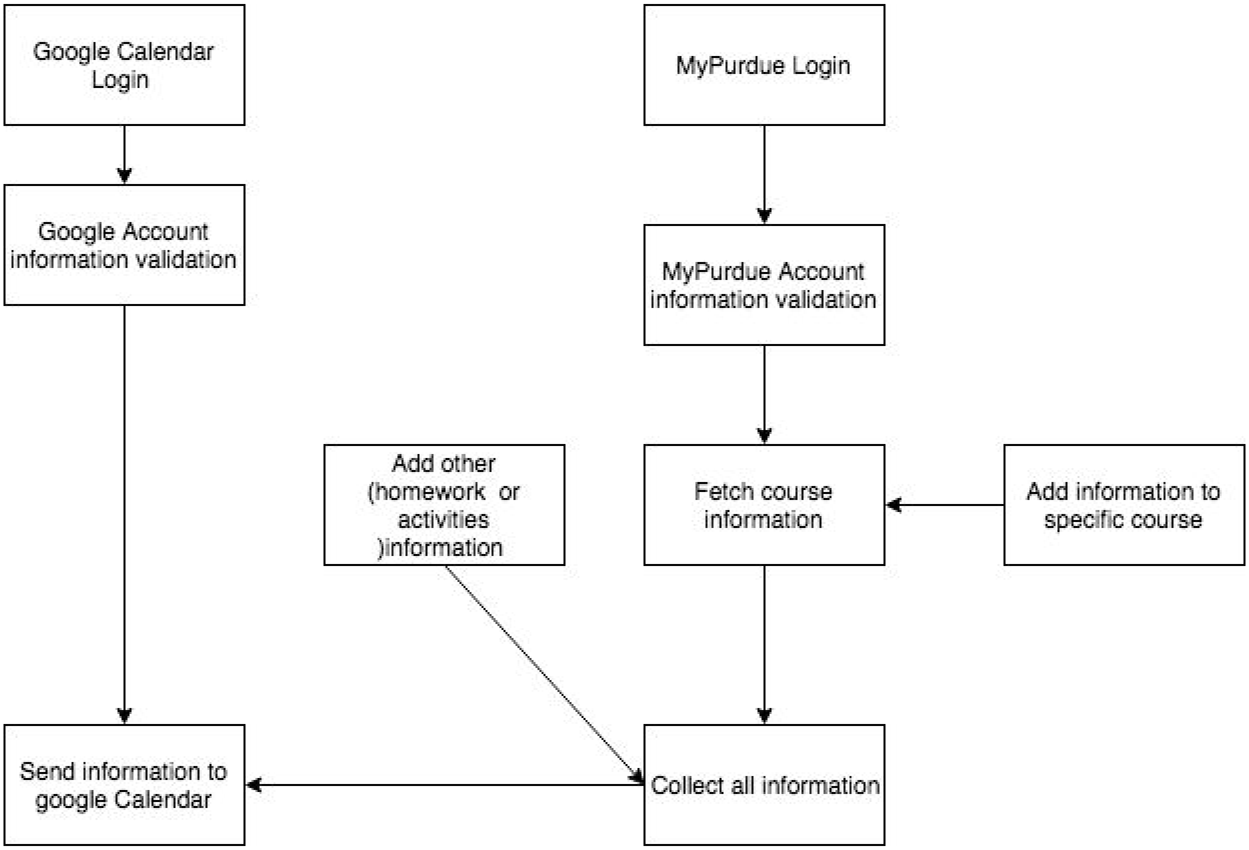
Along with an error response if any errors may occur

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.


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	PurdueSync		
Defect No.	**Description**	**Severity**	**How To Correct**
0	Application always returns 400 on Google Calendar.	1	Make sure to update the Chrome Extension ID on binary
1	Application extension would try and process online	3	Create a validator for online and physical classes.

Module	Google Calendar		
Defect No.	**Description**	**Severity**	**How To Correct**
0	Application should not crash with special characters	3	Make sure to automatically detect and remove any characters that might break the program
1	Application should NOT send requests to google servers if some internal error occurs	1	All internal tools should include a stubs that check for 400 messages before connecting to google api servers
2	Application should be able to handle null fields	1	Replace all null fields with blank spaces if it is a required field

Regression Testing Defect Log

Module	Purdue Sync		
Defect No.	**Description**	**Severity**	**How To Correct**
0	Items would only return time slot and not day of the week.	2	Make sure the time and day of the week are extracted using HTML table evaluation. Ensure return value is valid ISO time format.
1	Application would try to work on non MyPurdue pages.	3	Ensure application only renders functional HTML on relevant HTML.

Module	Google Calendar		
Defect No.	**Description**	**Severity**	**How To Correct**
0	Application should not crash with special characters	3	Stubs should contain test cases to see if any special characters are accepted
1	Application should NOT send requests to google servers if some internal error occurs	1	Stubs should include 400 messages to test if transmission is still sent
2	Application should be able to handle null fields	1	Stubs should check if any null fields have made it through the method