SE 3XA3: Module Internal Specification
MAC Schedule Importer

Team 12, 0C
Cassandra Nicolak, nicolace
Michelle Leung, leungm16
Winnie Liang, liangw15

December 6, 2018

Contents

List of Tables

List of Figures

Table 1: **Revision History**

Date	**Version**	**Notes**
2018-11-07	1.0	Rough Draft
2018-12-04	1.1	Updated renames and added new methods.

1 Introduction

The purpose of this module internal specification document is to provide a complete description of the specifications to design the MAC Schedule Importer. The project is a redesign ofthe open-source Chrome extension, UMD Google Calendar Schedule Importer, which imports the class schedule for students at the University of Maryland into Google Calendar. The reimplementation will be modified to allow students from McMaster University to import their schedules from Mosaic through a Desktop application.

2 Module Hierarchy

This section provides an overview of the module design. Modules are summarized in a hierarchy decomposed by secrets in Table 2. The modules listed below, which are leaves in the hierarchy tree, are the modules that will actually be implemented.

Level 1	**Level 2**
Hardware-Hiding Module	
Behaviour-Hiding Module	converter connector guiClient
Software Decision Module	parseMosaic

Table 2: Module Hierarchy

3 MIS of ParseMosaic

3.1 Uses

scrapy, subprocess

3.2 Interface Syntax

3.2.1 Exported Access Programs

Name	In	Out	Exceptions
parse	response: TextResponse	~~dataList~~ data_list: list of string tuples	-
~~runMe~~ run_me	passed_url: url (str)	GUI	-

∗Note: Exceptions are still in development.

3.3 Interface Semantics

3.3.1 State Variables

~~dataList~~ data_list: list of string tuples

3.3.2 Environmental Variables

process : CrawlerSpiderProcess() from the Scrapy Library

3.3.3 State Invariant

$0 \leq |$~~dataList~~ data_list$|$

3.3.4 Assumptions

parse() is called before ~~runMe()~~ run_me().

3.3.5 Access Program Semantics

parse(response)

- transition: ~~dataList~~ data_list:= modify ~~dataList~~ data_list so that it uses the Scrapy library to parse data from *response* and stores a list of string tuples with the tuple containing (course name, component, schedule, location, dates) of each course.
- exception : None

~~runMe()~~ run_me()

- output := ~~dataList~~ data_list
- exception: None

4 MIS of Converter

4.1 Uses

parseMosaic

4.2 Interface Syntax

4.2.1 Exported Access Programs

Name	**In**	**Out**	**Exceptions**
extract_date	input: str	start: str, end: str	-
to_military	input: str	mil: str	-
extract_weekdays	input: str	weekdays: str	-
rfc_output	date_str, time_str: str	start_date_time, end_date_time: str	-
convert	input: list of string tuples	output: list of dictionaries	-

*Note: Exceptions are still in development.

4.3 Interface Semantics

4.3.1 State Variables

None

4.3.2 Environmental Variables

None

4.3.3 State Invariant

None

4.3.4 Assumptions

None, unless stated in the access program.

4.3.5 Access Program Semantics

extract_date(input)

- output: start := $(\exists i | i \in input \wedge input[i] =\text{'-'} : start = input[0..i-1])$
- output: end := $(\exists i | i \in input \wedge input[i] =\text{'-'} : end = input[i+1..|input|])$
- exception: None

to_military(input)

- output: mil := returns a string of the military time given a 12-hour time input string.
- exception: None

extract_weeekdays(input)

- output: weekdays := given a string containing weekdays, ex. ”MoTWeThFr”, it returns a string with capitalized string with commas between the weekdays, ex.”MO,TU,WE,TH,FR”
- exception: None

rfc_output(date_str, time_str)

- output: a start and end dateTime in RFC 2232 format, and a rrule in Rfc 5545 format
- exception: None

convert(input)

- output: a list of dictionaries containing calendar event parameters
- exception: None

5 MIS of Connector

5.0.1 Uses

sys, os, googleapiclient, socket, oauth2client, httplib2

5.1 Interface Syntax

5.1.1 Exported Access Programs

Name	In	Out	Exceptions
login	-	GUI	ServerNotFoundError
logout	-	GUI	-
check_perms	-	bool	-
create_cal	name: str	Google Calendars	AccessTokenRefreshError, ServerNotFound Error, gaierror, ConnectionResetError
insert_events	-	True, None	AccessTokenRefreshError, ServerNotFound Error, gaierror, ConnectionResetError
get_num_events	-	|event_list|: int	AccessTokenRefreshError, ServerNotFound Error, gaierror, ConnectionResetError
check_insertion	-	bool	-
remove_new_cal	-	bool	AccessTokenRefreshError, ServerNotFound Error, gaierror, ConnectionResetError
push_to_schedule	-	bool	-

5.2 Interface Semantics

5.2.1 State Variables

service: Object
cal_id: string
bodies: dictionary

5.2.2 Environmental Variables

None

5.2.3 Assumptions

None, unless noted in the access programs

5.2.4 Access Program Semantics

login

- transition: service := Object
- exception: None

logout

- transition: service := None
- exception: None

check_perms

- output: Bool, true if permissions were granted, False otherwise.
- exception: None

create_cal

- transition: cal_id := new calendar id
- exception: client.AccessTokenRefreshError, when access token fails to refresh.
- exception: gaierror, internet fails
- exception: ConnectionResetError, internet fails
- exception: ServerNotFoundError, internet fails

insert_events

- output: inserts event into google calendar
- exception: client.AccessTokenRefreshError, when access token fails to refresh.
- exception: gaierror, internet fails
- exception: ConnectionResetError, internet fails
- exception: ServerNotFoundError, internet fails

get_num_events

- output: number of events in calendar: cal_id
- exception: client.AccessTokenRefreshError, when access token fails to refresh.
- exception: gaierror, internet fails
- exception: ConnectionResetError, internet fails
- exception: ServerNotFoundError, internet fails

check_insertion

- output: True if number of events in Google calendars matches the number of event parameters sent to Google. False otherwise

remove_new_cal

- output: removal of calendar: cal_id
- exception: client.AccessTokenRefreshError, when access token fails to refresh.
- exception: gaierror, internet fails
- exception: ConnectionResetError, internet fails
- exception: ServerNotFoundError, internet fails

push_to_schedule

- output: Pushes events to Google calendars. Returns True if successful. False otherwise.

6 MIS of Setup

6.1 Uses

sys, os, cx_freeze

6.2 Interface Syntax

6.2.1 Exported Access Programs

Not applicable.

6.3 Interface Semantics

6.3.1 State Variables

build_exe_options

6.3.2 Environmental Variables

os.environ['TCL_LIBRARY']
os.environ['TK_LIBRARY']

6.3.3 State Invariant

None

6.3.4 Assumptions

None, unless stated in the access program.

6.3.5 Access Program Semantics

This module is a configuration file for cx_freeze, a software that converts Python programs into executable applications. The module consists of import statements that imports all relevant libraries and a setup function of cx_freeze. The function contains an input file parameter with the input Python program guiClient.py and build options that include the packages that is required to build the program.

7 MIS of guiClient

7.0.1 Uses

PySimpleGUI, parseMosaic, connector, converter, urllib, webbrowser

7.1 Interface Syntax

7.1.1 Exported Access Programs

Name	In	Out	Exceptions
~~convertURL~~ convert_url	str	str	-
~~parseMosaic~~ parse_mosaic	str	~~fetchedList: list of tuple of str~~ fetched_list: list of tuple of str	-
set_list	parsed_list: list of tuple of str	fetched_list: list of tuple of str	-
~~printSched~~ print_sched	~~fetchedList: list of tuple of str~~ fetched_list: list of tuple of str	str	-
fetch	str	str	-
conn	-	Object	-
login	-	bool	-
logout	-	bool	AttributeError
~~pushSchedule~~ push_schedule	-	bool	-
fetch_button	-	-	-
fetch_popup	-	bool	-
login_button	-	-	-
import_button	-	-	AttributeError

7.2 Interface Semantics

7.2.1 State Variables

~~fetchFLG~~ _fetch_flg: bool
~~fetchedList~~ _fetched_list: list of string tuples
~~googleConn~~ _google_conn: None

7.2.2 Environmental Variables

layout: list of lists of type GUI
window: GUI
menu_def: list of lists of str

7.2.3 State Invariant

|~~fetchedList~~ _fetched_list| ≥ 0

7.2.4 Assumptions

None, unless stated by the access programs.

7.2.5 Access Program Semantics

~~convertURL(userInput)~~ convert_url(user_input)

- output: ~~userURL~~ user_url:= given a url from a path name (str), it returns the absolute file path (str).
- exception: None

~~parseMosaic(url)~~ parse_mosaic(url)

- output: url:= Parses the html document from *url* and displays the *url* on the GUI.
- exception: None

set_list()

- output: url:= Sets *_fetched_list* to *parsed_list*.
- exception: None

~~printSched(fetchList)~~ print_sched(fetch_list)

- output: out:= Converts ~~fetchList~~ fetch_list into the following format of str: "Course, Type, When, Location, Start/End Dates"

fetch(url)

- output: out:= Parses url using~~parseMosaic()~~ parse_mosaic() function and returns the schedule using ~~printSched()~~ print_sched().
- exception: None

conn()

- transition: Converts the output of ~~parseMosaic~~ parse_mosaic to Google API inputs.
- exception: None

login()

- output: out:= Creates a link to google's api service when the user logs into their Google account. Returns true if there is a service, otherwise false.

- exception: None

logout()

- ~~transition: deletes access key to user's account.~~
- output: out:= Deletes access key to user's account. Returns true if this is successful, otherwise false.
- exception: ~~None~~ AttributeError

~~pushSchedule()~~ push_schedule()

- output: out:= Uploads event items to a Google Calendar. Returns true if the import is successful and false otherwise.

fetch_button()

- output: url:= Updates the *tbxSchedule* textbox with a str.
- exception: None

fetch_popup()

- output: url:= Displays a pop-up window to the user. Returns true if user selects 'Yes', otherwise false.
- exception: None

login_button()

- transition: url:= Opens a new connection with *conn()*.
- output: url:= Updates the *tbxLogin* textbox with a str.
- exception: None

import_button()

- transition: url:= Updates the *tbxImport* textbox with a str.
- exception: AttributeError