SE 3XA3: Module Guide
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

1 Introduction

1.1 General Overview and Introduction

This document provides a complete description of the module guide for the MAC Schedule Importer, a desktop application that imports the user's Mosaic schedule into their Google calendar.

The intended audience of this document includes the following:

- Software Development/Design teams, including Team 0C: This document provides a method for designers and developers to check for the feasibility, correctness and consistency of their program.
- New team members and third party software development teams: This document can be a reference guide which enables developers that are unfamiliar with the project to easily comprehend the general system design and structure.
- Maintainers: The document can be used to locate specific modules for maintenance and the described module decomposition in the module guide allows a better understanding of the system when changes are made.

The module guide document is organized as follows:

- Section 2 lists the anticipated and unlikely changes of the software requirements.
- Section 3 summarizes the module decomposition that was constructed according to the likely changes.
- Section 4 specifies the connections between the software requirements and the modules.
- Section 5 gives a detailed description of the modules.
- Section 6 includes two traceability matrices. One checks the completeness of the design against the requirements provided in the SRS. The other shows the relation between anticipated changes and the modules.
- Section 7 describes the use relation between modules.

Table 1: **Revision History**

Date	**Version**	**Notes**
11/10/2018	1.0	Rev 0
12/04/2018	2.0	Rev 1

1.2 Motivation and Design Choices

For large software system designs, the decomposition into smaller components, such as modules and sub-modules, is a necessity for organization and structure. Moreover, modular decomposition enables future modifications to various system components without altering majority of the system. Hence, Team 0C followed the 'design for change' pattern for the MAC Schedule Importer project. For example:

- Components of the system that are anticipated to change are encapsulated into 'secrets'.
- Each module contains one secret.
- Other members of the system that requires data from a module can only acquire the information through public access methods that are defined in that module.

2 Anticipated and Unlikely Changes

2.1 Anticipated Changes

Design for change was incorporated into the software design for MAC Schedule Importer. The anticipated changes, which will ideally change only the secrets hidden in modules without affecting the whole project, are listed below.

AC1: The format of Mosaic schedule.

AC2: The operating systems which the software interfaces.

AC3: The hardware on which the software is running.

AC4: The syntax and functions of Scrapy.

AC5: The Google Calendar layout.

2.2 Unlikely Changes

The following design decisions are not intended to change as modifications to these decisions lead to multiple subsequent changes to the whole system.

UC1: Input/Output devices (Input: File, Output: Screen).

UC2: There will always be a source of input data external to the software.

UC3: The parsing algorithm.

UC4: The main purpose of the program - To import a Mosaic schedule into Google Calendar.

3 Module Hierarchy

The module hierarchy is an overview of the general structure and design for the module. In Table 2, the modules are shown in a hierarchy decomposition according to their respective secrets. The models below are the 'leaves' of the hierarchy tree and will be implemented.

M1: *Hardware-Hiding Module

M2: parseMosaic Module

M3: convertor Module

M4: connect Module

M5: guiClient Module

M6: setup Module

*Note: The Hardware-Hiding Module is not implemented for this software-based project. Inclusion of this module is for formality purposes.

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

Table 2: Module Hierarchy

4 Connection Between Requirements and Design

The design of the MAC Schedule Importer is designed to satisfy the requirements developed in the SRS document. During this stage, the system is decomposed into modules. The connection between requirements and modules is listed in Table 3. It is recommended to read Section 3 and Section 5 before reading Section 6.

5 Module Decomposition

The modules of MAC Schedule Importer are decomposed according to the principle of "information hiding" proposed by David Parnas. Each module is comprised of *Secrets*, *Service*, and *Implemented by*. The *Secrets* in a module is a noun. It is a statement of the hidden design decision for the module. The *services* specifies what the module will accomplish without specifying what methods the module will use to accomplish it. The *Implemented By* title is a tentative idea for the implementing software of each module.

5.1 Hardware Hiding Modules (M1)

Secrets: The algorithm and data structure that are used to implement the virtual hardware.

Services: Serves as a virtual hardware used by the rest of the system. This module provides the interface between the hardware and the software so that the system can use it to display outputs or to accept inputs.

Implemented By: OS

5.2 Behaviour-Hiding Module

5.2.1 Input Format Module (M3)

Secrets: The format and structure of the input data.

Services: It converts the retrieved input data into the data structure used by the output module.

Implemented By: convertor

5.2.2 Output Module (M4)

Secrets: The format and structure of the output data.

Services: It converts the data into the output format and displays the results.

Implemented By: connecter

5.2.3 User Interface Module (M5)

Secrets: The format and structure of user interface.

Services: It interacts with the user to retrieve the user's Google account and Mosaic schedule file.

Implemented By: guiClient

5.3 Software Decision Module

5.3.1 Retrieve Input Module (M2)

Secrets: The method to retrieve the input data.

Services: It obtains the input data and passes the information to the input format module to convert the data into a usable format.

Implemented By: parseMosaic

5.3.2 Convert Python to Executable Input Module (M6)

Secrets: The method to covert the python program into a executable application.

Services: It obtains the input python program and returns the corresponding executable application.

Implemented By: setup

6 Traceability Matrix

This section shows two traceability matrices: between the modules and the requirements and between the modules and the anticipated changes.

Req.	**Modules**
FR01	M2, M5
FR02	M5
FR03	M4, M5
FR04	M5
FR05	M5
FR06	M5
FR07	M5
FR08	N/A
FR09	M4, M5
FR10	M5
FR11	M6
FR12	M6
FR13	M6
NF01	M5
NF02	M5
NF03	M5
NF04	M5
NF05	M5
NF06	M5
NF07	M5
NF08	M5
NF09	M2, M3, M4, M5
NF10	
NF11	M2, M3, M4, M5
NF12	
NF13	M5
NF14	M2, M3, M5
NF15	M5
NF16	M2, M3, M4, M5
NF17	M2, M3, M4, M5

Table 3: Trace Between Requirements and Modules

AC	Modules
AC1	M2
AC2	M5
AC3	M5
AC4	M2
AC5	M4

Table 4: Trace Between Anticipated Changes and Modules

7 Use Hierarchy Between Modules

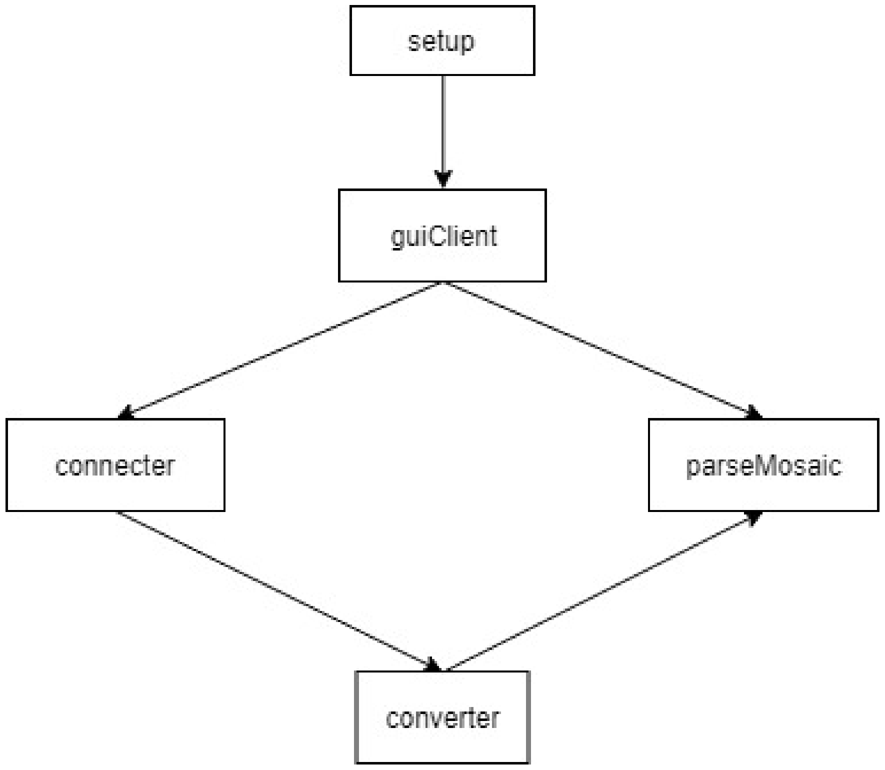

Figure 1: Use hierarchy among modules

8 Schedule

Please refer to the link to find the current Gantt Chart: Gantt03