SE 3XA3: Software Requirements Specification MAC Schedule Importer

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

December 5, 2018

Contents

List of Tables

List of Figures

Table 1: **Revision History**

Date	**Version**	**Notes**
2018-10-05 1	1.0	Rev 0
2018-12-05 1	2.0	Rev 1

This document describes the functional and nonfunctional requirements for MAC Schedule Importer. As well, the document includes various constraints and issues this project may encounter. This document is written for the purpose of informing the users of the overview of the project as well as the requirements and criterias the project will meet. The template that the team decided to use for the Software Requirements Specification (SRS) is a subset of the Volere template (1).

1 Project Drivers

1.1 The Purpose of the Project

The purpose of this project is to reimplement the 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 to their Google Calendar from the Mosaic website through a Desktop application.

1.2 The Client, the Customer, and Other Stakeholders

The stakeholders of this project are currently the development team (0C) as well as the students and educational staff of the McMaster University course SFWRENG 3XA3. Other stakeholders of the application include McMaster University students and McMaster's University Technology Services. Additionally, potential stakeholders include third parties such as Google, open-software developers and other universities that may be interested in our product for their usage.

The client for this project will be McMaster students and staff, since the motivation for the tool to be made was from the students' frustration towards the irregular shut downs that the Mosaic website has. Moreover, a potential client for this application in the future is Mosaic themselves as they may be interested in hiring the software development team to produce the application as an additional feature for their website.

On the other hand, the customers of the application are the staff and students of McMaster University as well as anyone that uses Mosaic. Additionally, other universities are potential customers since they may be interested in using our application for their university. Google may also be interested in our program and can be a future customer for this application.

1.3 Users of the Product

The main users of this application will be McMaster students who would like to have their academic schedules easily accessible through their Google Calendars. Other users include McMaster University staff, software developers, and interested organizations such as personnel from Mosaic and Google. The user group may vary from having very basic to advance knowledge of computers, software applications and mobile devices. For example, students with a computer and software development background will find the application easier to use in comparison to other students that do not have such background knowledge. However, users with no prior software knowledge will be able to use the product. The minimum requirement is that users are expected to be able to use and have access to a computer. Moreover, the user is assumed to be connected to the Internet to have access to Mosaic and Google Calendars from their preferred browser.

2 Project Constraints

2.1 Mandated Constraints

The primary constraint for this application is that it must have similar functionality and design to the original source code. The application must be a reimplementation of the UMD Schedule Importer. Additionally, the user's operating system must also be taken into consideration. Another constraint is that Mosaic shuts down for weekly maintenance every Thursday from 18:00 - 22:00. Also, Mosaic may have outages from certain periods of time which will impact development and testing. Lastly, all deliverables for this application must be completed prior to December 5th, 2018. As well, the user must have a computer (laptop, personal computer, etc.) which must be connected to the Internet.

2.2 Naming Conventions and Definition

- Mosaic: McMaster University's online administrative information system.
- MacID: A McMaster University student's login account.
- UMD Schedule Importer: University of Maryland's Google Calendar Schedule Importer that was developed by a previous student.
- Functional Requirements: Requirements that describe the behaviour of the application.
- UI: User Interface.
- Non-functional Requirements: Requirements that describe the qualities of the application.
- User: A person who will be using the final version of the application.
- Calendar: Refers to a user's Google Calendar.
- Schedule: A user's academic timetable accessed through Mosaic.
- SFWRENG 3XA3: Software Engineering 3XA3 - Software Engineering Practice and Experience: Software Project Management. A course at McMaster University.
- Stakeholder: A person, company, or an organization that has interest and concerns pertaining to the product.
- Chrome Extension: A software program tailored to Chrome that customizes the behaviour and preference of the user.
- Google: A technology company that provides Internet-related services and products.
- Constraint: A limitation or restriction that the application must abide to.
- Assumption: Something that is certified and accepted as true without proofs.

2.3 Relevant Facts and Assumptions

The application will be intended to be used by McMaster University students who are currently enrolled in courses. Users must also have a valid MacID and password in order to log into Mosaic. Users must have a valid Google Mail account as well. It will be assumed that users have, at a minimum, general knowledge to operate computers in order to run the desktop application. The user is also assumed to be connected to the Internet with no connection and speed problems. The user is assumed to have access to a computer in order to use the product.

3 Functional Requirements

3.1 The Scope of the Work

Deliverables:

- Requirements Documentation
- Functioning Software Deadlines
- Problem Statement Revision 0 (September 21)
- Development Plan Revision 0 (September 28)
- Requirements Document Revision 0 (October 5)
- Proof of Concept Demonstration (October 9)
- Test Plan Revision 0 (October 26)
- Test Report Revision 0 (October 26)
- Design Documentation Revision 0 (November 9)
- Final Demonstration Revision 1 (November 27)
- Problem Statement Revision 1 (December 1)
- Development Plan Revision 1 (December 1)
- Requirements Document Revision 1 (December 2)

- Test Plan Revision 1 (December 3)
- Test Report Revision 1 (December 3)
- User Documentation (December 4)
- Final Documentation (December 5)

3.2 The Scope of the Product

The application will first require the student to download a html file of their schedule which they can retrieve through the application's link to Mosaic. Next, the program will request the user to upload the file to the application where the application will display schedule information in text format. The user will then confirm that the schedule is correct and the application will direct the student to log in to their Google account. Once the program acquires the user's permission to access their Google account and their Google Calendar, the application will import the schedule into Google Calendars.

3.3 Functional and Data Requirements

- Requirement number : FR01
  The interface shall notify the user if it is unable to access Mosaic.
  Rationale: To inform the user of why the application cannot proceed.

- Requirement number : FR02
  The application must notify the users of the exact information they will import into their Google Calendar prior to proceeding.
  Rationale: To ensure the information being inserted into the calendar is correct.

- Requirement number : FR03
  The application must account for multiple uses from a user.
  Rationale: To allow for changes in a users schedule.

- Requirement number : FR04
  The application must have a simple GUI.
  Rationale: Target users may not know how to use a consoles command line. A GUI is also more visually appealing.

- Requirement number : FR05
  The user interface must have a 'Help' option that explains to how to use the application to the user.
  Rationale: To allow for easy and quick use of the application.

- Requirement number : FR06
  The user interface must show the user the changes to their existing Google Calendar before changing it.
  Rationale: To allow users to check the changes.

- Requirement number : FR07
  The application must ask for confirmation to confirm that the listed changes are correct.
  Rationale: To allow users to confirm their changes.

- Requirement number : FR08
  This requirement is incorporated into FR06.

- Requirement number : FR09
  The application must request permission to access the user's personal information in their Mosaic and Google account.
  Rationale: To ensure users are informed of processing private information.

- Requirement number : FR10
  The application must have an option that allows a user to exit the application at any given time.
  Rationale: To allow users to exit the application.

- Requirement number : FR11
  The application must confirm with the user that they can only fetch their Mosaic calendar once.
  Rationale: To ensure that the user is ready to retrieve their Mosaic schedule.

- Requirement number : FR12
  The application must indicate that the application has successfully imported the schedule into Google Calendar.
  Rationale: To inform users on the progress status.

- Requirement number : FR13
  The application must indicate that the application has successfully accessed the user's Google account.
  Rationale: To inform users on the progress status.

4 Non-functional Requirements

4.1 Look and Feel Requirements

- Requirement number : NF01
  The application shall comply with accessibility standards and not prevent users affected by colour blindness from using the application.
  Rationale: This will ensure that colour on the UI will not impact the use of the application.
  Fit Criterion: All information will be visible and not be dependant on colour.
  Priority: High
  History: Created October 4, 2018

- Requirement number : NF02
  The application will have a simple UI design with minimal buttons.
  Rationale: This will allow users to quickly perform an import into their calendar.
  Fit Criterion: The users should be able to perform the import in less than five steps.
  Priority: Medium
  History: Created October 4, 2018

4.2 Usability and Humanity Requirements

- Requirement number : NF03
  The application shall be intuitive and easy to use for any users who have at least completed high school or equivalent.
  Rationale: Users must be able to the application.
  Fit Criterion: The intended user is a McMaster University student which would have at least general knowledge on how to use computer applications.

Priority: High
History: Created October 4, 2018

- Requirement number : NF04
The application shall be easy to install.
Rationale: In order to fulfill requirement NF03, users must have the application installed in order for it to be used.
Fit Criterion: The intended user is a McMaster University student which would have at least general knowledge on how to install computer software.
Priority: Medium
History: Created October 4, 2018

- Requirement number : NF05
All text displayed on the interface will be in the English language.
Rationale: The application is intended for users who attend McMaster University where the primary language is English.
Fit Criterion: All information should be at a basic English reading level.
Priority: Medium
History: Created October 5, 2018

- Requirement number : NF06
This non-functional requirement is incorporated into NF03.

- Requirement number : NF07
The application shall guide users step by step during use.
Priority: Medium
History: Created October 5, 2018

4.3 Performance Requirements

- Requirement number : NF08
The application shall be able to be operated by a single user on a desktop computer or laptop.
Rationale: In order to fulfill requirement NF03, users must be able to operate the application.
Fit Criterion: Users are able to run and use the application on their

desktop computer or laptop.
Priority: High
History: Created October 5, 2018

- Requirement number : NF09
The program is expected to respond to a user's input in a reasonable amount of time, 0.5 seconds. This time limit may be revised after testing.
Rationale: User retention and satisfaction.
Fit Criterion: Users are able to run and use the application on their personal desktop computer or laptop.
Priority: Medium
History: Created October 5, 2018

- Requirement number : NF10
The application should be made available on a reliable site. Currently Github.
Rationale: The application is expected to be in high demand prior and during enrollment dates. As it is a standalone product, we do not have to consider server crashes. However, we do have to consider the host.
Priority: High
History: Created October 5, 2018

4.4 Operational and Environmental Requirements

- Not applicable. The application is not intended to be run in unusual external environments.

4.5 Maintainability and Support Requirements

- Requirement number : NF11
*Note: This non-functional requirement is removed for the current implementation of the product due to time constraints. However, future versions of the application will include this requirement.
The programming language used will be currently supported on multiple operating systems.
Rationale: The application can be easily modified and maintained if needed.

Fit Criterion: The programming language is supported on Windows and Linux.
Priority: Low
History: Created October 5, 2018

- Requirement number : NF12
The source code will be visible to the public and free to use.
Rationale: The application can be maintained by current developers or interested users.
Fit Criterion: The source code can be accessed by the public.
Priority: Low
History: Created October 5, 2018

- Requirement number : NF13
A user's email address will be listed.
Rationale: For support, feedback, and inquiries.
Fit Criterion: Current developers can be contacted by the public.
Priority: Low
History: Created October 5, 2018

4.6 Security Requirements

- Requirement number : NF14
The application shall not store or upload user information.
Rationale: This is required to protect a user's personal information.
Fit Criterion: The application will not have the ability to store user data.
Priority: Low
History: Created October 5, 2018

4.7 Cultural and Political Requirements

- Requirement number : NF15
The application shall not display any images, symbols, or text that are culturally or politically offensive and controversial to any country, race, religious and ethnic group.
Rationale: To ensure all users will enjoy the experience of using the application.

Fit Criterion: Texts, images and symbols will be evaluated before including in the application.
Priority: Low
History: Created October 5, 2018

4.8 Legal Requirements

- Requirement number : NF16
  The application shall abide to all relevant laws, regulations, and standards pertaining to privacy and intellectual property.
  Rationale: To ensure the legality of the application.
  Fit Criterion: The application shall be verified by relevant standardkeepers to certify that the application adheres to relevant standards.
  Priority: Low
  History: Created October 5, 2018

4.9 Health and Safety Requirements

This section is not in the original Volere template, but health and safety are issues that should be considered for every engineering project.

- Requirement number : NF17
  The application shall not impair the health and safety of users.
  Fit Criterion: Users will be able to run the application with no harm to their health and safety.
  Priority: Low
  History: Created October 5, 2018

- Requirement number : NF18
  The application shall not impair and harm the environment and cause extreme pollution that may contaminate and jeopardize the use's safety.
  Fit Criterion: Users will be able to run the application with no harm to the environment.
  Priority: Low
  History: Created December 3, 2018

5 Project Issues

5.1 Open Issues

Porting Capabilities

- The original source code that the application is based on is written in a web-based programming language and is designed for Google Chrome applications. The reimplemented application will be written in a different programming language and not as a Google Chrome extension.

5.2 Off-the-Shelf Solutions

Mosaic has an available schedule exporter. However, it is limited since it cannot be used with certain browsers and requires a waiting period of one day before schedule changes can be included in the export. Hence, we would like to create a more reliable and efficient application.

Additionally, the open-source UMD Importer is another off-the-shelf solution. However, the application is tailored towards University of Maryland rather than McMaster University. Moreover, the UMD Importer supports only Chrome browser. Thus, we would like to create a product that enables the user to use a variety of browsers.

5.3 New Problems

Not applicable. The application will not affect the existing UMD Importer or Mosaic in any way.

5.4 Tasks

The project's tasks are dictated by the deliverable outline provided by the SFWRENG 3XA3 course. The final implementation and all documentation must be completed by December 5th, 2018. See Deliverables under section 3.1 Scope of Work for a detailed list of what is required.

5.5 Migration to the New Product

Not applicable. The application is not intended for University of Maryland students. The application shall incorporate functions of the UMD Google Calendar Schedule Importer to McMaster University using Mosaic and similar platforms.

5.6 Risks

Mosaic's schedule layout may change in the future. Hence, the application must be updated to account for the changes as the application will become non-functional.

5.7 Costs

- Finanical:

  Not Applicable. The application shall be free and open for all McMaster University students.

  Additionally, the product shall not require and rely on any external resources that requires to be bought.

- Time:

  - 12 lab sessions x 2 hours each
    = 24 mandatory hours spent in lab.
  - 9 In-person meetings x 1 hours each
    = 9 hours.
  - These figures are approximations, real times may vary:
    - 10 hours of online meetings
    - 15 hours for development
    - 5 hours for testing
    - 25 hours for documentation

  Total estimated hours needed: 88 hours.

5.8 User Documentation and Training

No user training will be required. The software shall be straightforward and intuitive to use. Also, the application will guide users when necessary through the help option. A user guide (documentation) will list the features of the program and reiterate instructions.

5.9 Waiting Room

Integration with Facebook events, Mac Rec Schedule, and custom personal events.

5.10 Ideas for Solutions

The original source code uses Google's API. There's an API Client Library for the Python programming language. It is important that the final implementation of the application is an executable. This conversion must be taken into consideration to make installation easy for the end-user.

References

[1] James and S. Robertson, “Volere Requirements Specification Template Edition 13 - August 2007,” 2007.

6 Appendix

Not applicable. There are no added resources that are included in this documentation.

6.1 Symbolic Parameters

Not applicable. The product does not have symbolic parameters.