ENSC 305W/440W Grading Rubric for Functional Specification

Criteria	Details	Marks
Introduction/Background	Introduces basic purpose of the project.	**/05%**
Content	Document explains the functionality of the proposed product without excessive design content (i.e., outlines the "what" rather than the "how").	**/10%**
Technical Correctness	Ideas presented represent valid functional specifications that must be considered for a marketed product. Specifications are presented using tables, graphs, and figures where possible (rather than over-reliance upon text).	**/15%**
Process Details	Complete analysis of problem. Justification for chosen functionalities. Sources of ideas referenced. Specification distinguishes between functions for present project version and later stages of project (i.e., proof-of-concept, prototype, and production versions). Comprehensively details current constraints.	**/20%**
Engineering Standards	Outlines specific engineering standards that apply to the device or system and lists them in the references.	**/10%**
Sustainability/Safety	Issues related to sustainability issues and safety of the device are carefully analyzed. This analysis must cover the "cradle-to-cradle" cycle for the current version of the device and should outline major considerations for a device at the production stage.	**/10%**
Conclusion/References	Summarizes functionality. Includes references for information from other sources.	**/05%**
Presentation/Organization	Document looks like a professional specification. Ideas follow logically.	**/05%**
Format Issues	Includes letter of transmittal, title page, executive summary, table of contents, list of figures and tables, glossary, and references. Pages are numbered, figures and tables are introduced, headings are numbered, etc. References and citations are properly formatted.	**/10%**
Correctness/Style	Correct spelling, grammar, and punctuation. Style is clear concise, and coherent. Uses passive voice judiciously.	**/10%**
CEAB Outcomes: Below Standards, Marginal, Meets, Exceeds	8.2 Responsibilities of an Engineer: 8.5 Integration of Standards: 9.2 Sustainability:	

October 17, 2014

Dr. Andrew Rawicz
Faculty of Applied Science, School of Engineering Science
Simon Fraser University
Burnaby, BC, V5A 1S6

Re: ENSC 440W/ENSC 305 Capstone Project: Functional Specifications for an automated animal feeder to improve laboratory experimentation

Dear Professor Rawicz,

Please allow this report to serve as the Functional Specifications for our ENSC 440 and ENSC 305 capstone project. This report will illustrate in detail the behavior of our automated feeder which automates the feeding of animals in a research laboratory setting. Our design consists of a sliding motorized door which separates the food from the animal, and a Google Calendar user interface allowing the researcher to control when and for how long the food is available.

The purpose of this functional specification is to present an overview of the functionality of the proposed product without extreme design content. Our team will reference this document throughout the completion of our product.

We hereby confirm that this report was written entirely by the members of Optimaus; we have received no external advice or assistance in writing this report. We also confirm this report has not been previously submitted for academic credit at this or any other institution.

Optimaus consists of four determined and talented senior engineering students. Rob Lepine, Kevin Killy, Kenny Woo and Kyle Griffith. If you have any questions or concerns about our functional specifications, please feel free to contact Rob Lepine at 778-689-8720 or by email at rlepine@sfu.ca.

Sincerely,

Robert Lepine
Optimaus - CEO

Optimaus

The Functional Specifications for AutoFeed

An Automated Animal Feeder for Laboratory Testing

Prepared for:

Andrew Rawicz - ENSC 440
Steve Whitmore - ENSC 305
Respected staff of
School of Engineering Science
Simon Fraser University

Project Members

Robert Lepine
Kyle Griffith
Kenny Woo
Kevin Killy

Contact:

CEO, Robert Lepine
rlepine@sfu.ca

EXECUTIVE SUMMARY

Optimaus wishes to solve a problem experienced by researchers that involves the routine feeding of laboratory animals. Currently, researchers need to feed rodents on a set schedule. They also need to remove all food that they fed the animals after a set amount of time. Optimaus hopes to make a device which can automatically dispense and retract food at programmable times. We are working closely with the Circadian Rhythms lab at Simon Fraser University to gather requirements and cages for prototyping a solution to automate the feeding of the rats in the lab. Optimaus consists of senior engineering students dedicated to design and implement a reliable product for the client. Our product could relieve graduate students from spending hundreds of hours a semester doing mundane tasks. Therefore, AutoFeed could also save research labs costs associated to graduate student wages. We have carefully planned to design, develop and test AutoFeed toward a working product.

The development of this product will be done in three stages: two development stages to obtain a working prototype and one final stage to deliver a final product installed in the Animal Research Centre (ARC) lab.

Development Stage I:

- Build the main mechanical elements and have them mounted to a cage.
- Have the ability to move the motor to open and close the door.

Development Stage II:

- A hopper will be mounted onto the side of the cage to allow for bulk storage of food.
- A motor should be controlled by the Google Calendar API so the door opens and restricts access to the food in the hopper on a schedule.
- A feedback system using push buttons will tell the user whether the door is open or closed.

Development Stage III:

- The prototype will be copied 8 to 16 times to be used in the Lab by the technicians.
- All wiring to the lab computer and the cages will be completed.
- Lab technicians will be able to use a Google Calendar to schedule feeding.

At the end of the second development stage a working prototype of the project will be complete. This should be finished no later than December 3rd for the Demo Date. Development Stage III will be completed if time and funding permits.

In the following document, detailed functional specifications are outlined for the system and the subsystems of the AutoFeed project. The intended audience for this document is the designers, testers and developers of AutoFeed. Allow this document to serve as a guideline; it will be updated as necessary.

TABLE OF CONTENTS

LIST OF FIGURES

GLOSSARY OF TERMS

API
Application programming interface. Provides a means of communicating with a particular application using programming functions provided in various programming languages.

Arduino
An open-source microcontroller used to control physical objects, in our case, motors.

Controller
Refers to the microcontroller being used, in this case, the Arduino.

Dispensing Apparatus
Mechanical apparatus that allows and restricts acess to food during desired feeding times.

Google Calendar
Google's online calendar application. This will be used to schedule desired feeding times.

PWM output
Pulse width modulation digital outputs. These outputs will be used to drive the motors required to operate the dispensing apparatus.

Raspberry Pi
A single-board computer. This will be used to interface with the Google Calendar API at a high level, before signalling the Arduino of desired actions in the physical world.

User Interface
This refers to a graphical view that will allow users to create and edit the feeding schedule. We will be using Google Calendar.

1 INTRODUCTION

Canada is one of the most automated countries in the world. In all industries, Canadians are consistently being relieved of doing routine tasks as automation takes over. This is a major component to what makes Canada a first-world country and separates us from the rest of the world. At Optimaus, we wish to further progress the automation movement in the laboratory research industry. In research environments, people who are highly educated are sometimes forced to do mundane tasks when a simple solution of automation could save many man-hours. Optimaus wishes to automate the feeding of rodents in an SFU lab. At irregular intervals, students are currently required to enter the labs in the middle of the night to administer food for research purposes. Our project, AutoFeed, will automatically feed the rats in order to relieve the students from coming to the lab at awkward times of the day or night. Our product will do something known as temporal restriction: a research technique that involves feeding the rats for a given amount of time then removing food entirely. Therefore, we have made the retraction of food a priority to achieve satisfied clients. The requirements for AutoFeed as proposed by Optimaus are described in these functional specifications.

1.1 Scope

This document will outline the functional requirements for Optimaus' product AutoFeed. This document describes the functionality of the system including the mechanical elements, the microcontroller, the user's Google Calendar interface and how these systems work together. All detailed functional requirements will be discussed. This document will be referred to as a guide while designing, developing and testing AutoFeed in order to meet the client's requirements and provide a reliable product.

1.2 Intended Audience

This document is intended for the designers at Optimaus. It should be used as a guide to ensure the project is made with all functional requirements in mind. It will also be used for Optimaus's marketing department when attracting more clients.

1.3 Classification

The functional requirement specification is shown as follows:

[Rn-p], where

- 'R' is an abbreviation for requirement, 'n' is the functional requirement number and 'p' stands for one of the following three development phases:
    - I. Proof-of-Concept stage
    - II. Ongoing developing (both proof-of-concept and final production) stage
    - III. Final production stage

2 GENERAL REQUIREMENTS

The general system requirements and system overview for AutoFeed are outlined in this section. The specific requirements for the different stages are presented as well.

2.1 System Overview

The AutoFeed can be represented by the following high level system block diagram shown below.

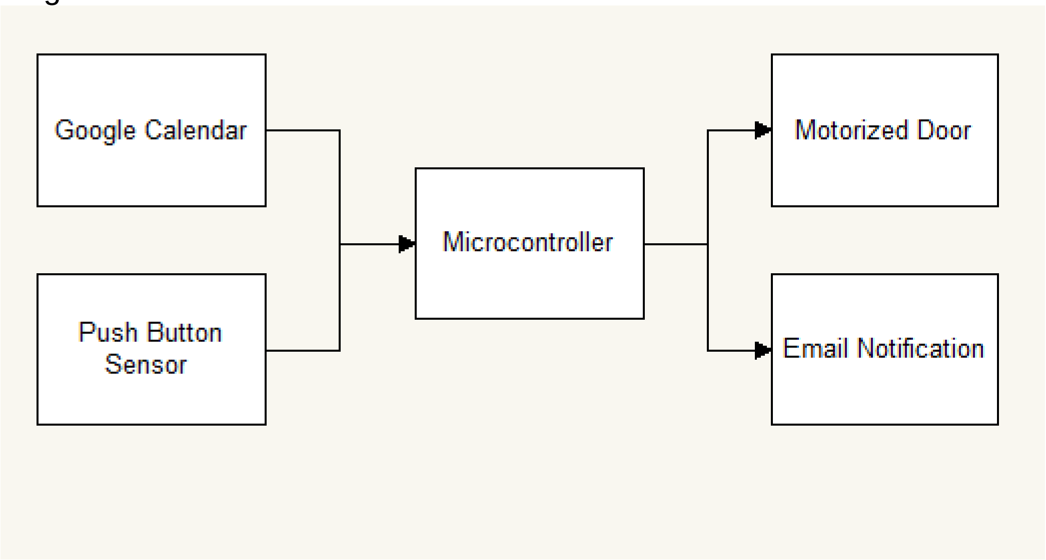

Figure 1: Block Diagram of the AutoFeed system

AutoFeed consists of two major features. The first major feature is the motorized door that is coupled to events on a Google Calendar. The Google Calendar interface allows the user to set feeding schedules for the animal remotely. The second feature is a set of two feedback push buttons mounted on the mechanical frame. These pushbuttons will be placed at both the top and bottom of the path of the motorized door such that one gets pressed with the door is either fully open or fully closed. The pushbuttons serve as a feedback system and will shut off the motor once the door has finished opening or closing. In the case that the door does not fully open or close, the pushbuttons will also allow us to send an email notification to the user telling that the door is blocked or not functioning correctly.

There will be three Development Stages for AutoFeed:

2.1.1 Development Stage 1

The main goal of AutoFeed is to allow for temporal feeding automatically. For our first proof of concept the goal is to have a working physical system. This means mounting a sliding door controlled by a motor on the outside of a cage. This phase will be dedicated to building the mechanical elements mounting them to the cage, and having the microcontroller communicate with the motor so that the door can open and close. The door should be mounted in such a way that a food hopper can be mounted onto the outside of the cage. The motor should close without harming the animals. The door should close going up to limit the chance of an appendage getting clipped. The door should move smoothly and slowly. At the end of this stage the physical requirements, excluding the push button feedback feature, for AutoFeed should be working and complete.

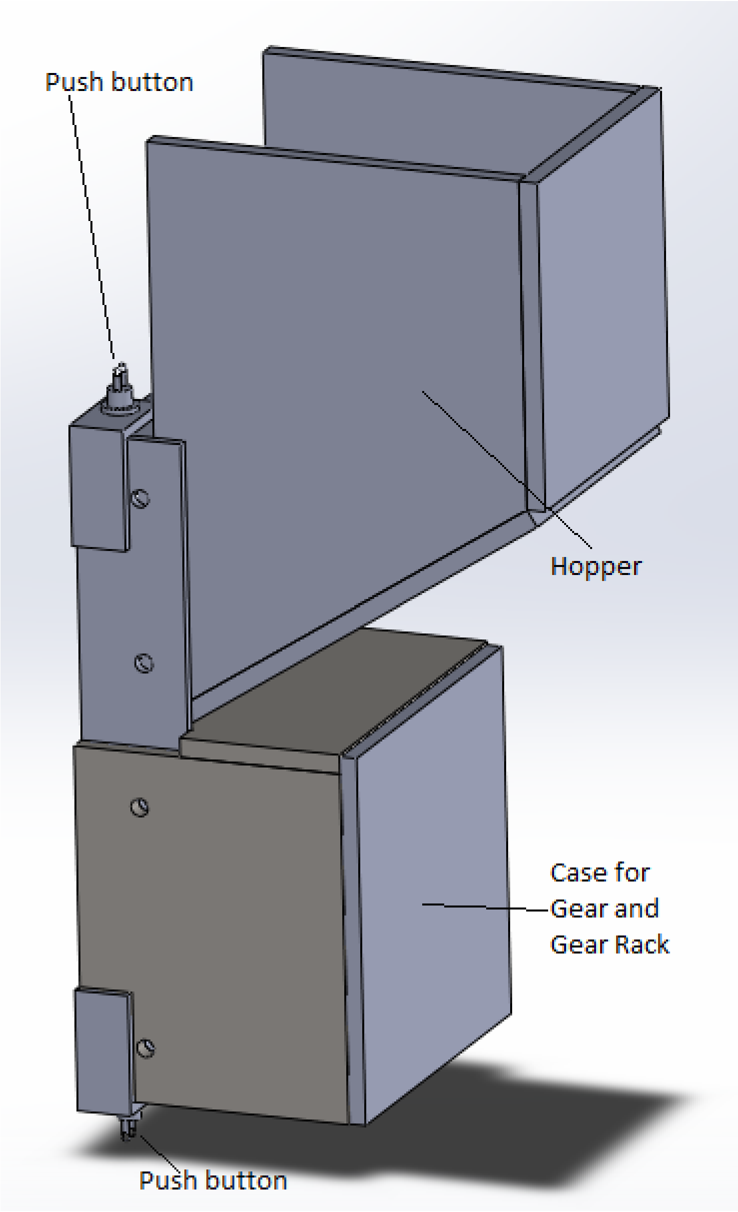

Figure 2: AutoFeed after Development Stage 1

2.1.2 Development Stage 2

After completing the mechanical functionality of AutoFeed we shall begin Development Stage II by making the device communicate through the Google Calendar API. At the end of this stage the user will be able to program times for

feeding by creating events on a Google Calendar. This will allow researchers to not be in the lab for feeding and make the process automatic.
Through communicating to the microcontroller from the Google Calendar a signal will be sent to the motor every time an event starts or finishes. This signal will power the motor to either open or close the door. The microcontroller will then cause the motor to turn and change the input. We should be able to see the motor being controlled by the MCU which is activated by a scheduled event from the Google Calendar API. Upon finishing this stage we will have a finished product and will be able to present. We should be able to show the door open and close automatically. This output will be due to the events scheduled by the events from the Google Calendar. If time and funding permits we will continue on to Development Stage III.

2.1.3 Development Stage 3

Once the first finished product is made we will test it for reliability. Once we are certain that it works properly and the design is optimized we will begin producing multiple feeders. In this stage we will have to make more copies of the mechanical elements and use the same microcontroller to control up to sixteen cages. We will need to wire all sixteen motors in the ARC lab and educate the client on how to use the cages. Furthermore, we will need to extend our code to accommodate several cages, and have a standard for how the user creates events in their Google Calendar such that they know which event will control which cage. In order to complete this stage we will need to secure funding from the client and have to be able to promise that our system is reliable and will work for an extended period of time. To ensure the products are the same as the final prototype, we will use CorelDraw designs to cut acrylic with Engineering Science's laser cutting printer.

The user interface shall be based off of Google Calendar to minimize learning time for the client. A Gmail account will be made to control the cages. This Gmail account will be used on the ARC computer stations and remotely for the researchers to schedule the feeding times of the animals.

2.2 General Requirements

R1-I All kinetic mechanisms may not harm the animals in any way.
R2-II Food must be able to be presented to the animals, and then fully removed at any time interval.
R3-I Must not allow the animals to take the food and save it for later.
R4-II Must give appropriate feedback when the mechanism isn't closed or opened.
R5-III Must fit the size specifications of all the four rooms in the ARC in SFU (both for length of cables and size of shelves).
R6-III Dispensing mechanism must be easy to clean.
R7-II The feeding mechanism cannot have exposed acrylic or plastic edges to allow the animals to chew and possibly deform the apparatus.

R8-III Dispensing mechanism must be scalable to a minimum of eight separate cages.
R9-III All elements of AutoFeed must not interfere with electronics already in place at ARC.
R10-III The retail price of a fifteen cage AutoFeed system should be around $1000.
R11-III The MCU must be able to control at least eight cages.

2.3 Electrical Requirements

R12-I The Raspberry Pi shall require its own power supply.
R13-I The Arduino microcontroller shall require its own power supply.
R14-III One motor should be run at a time to save power.
R15-III Power supplied by the microcontroller should be able to support one running motor and 15 idle motors.

2.4 Physical Requirements

R16-I The food hopper should be able to hold 300g of food.
R17-I The sliding door should open and close with negligible friction.
R18-I The gear should sit on the treads appropriately to facilitate smooth movement.
R19-I The mechanical elements should be no longer than 22cm to fit on standard cage.
R20-I The gears should be housed separately and inaccessible to the animal from within the cage.
R21-II The push buttons should be placed in such a way that they are pressed fully when the door is opened or closed.
R22-I The frame should be made out of metal instead of acrylic to prevent chewing from animals.
R23-I Must include bars in the design to prevent rats from stealing large pieces to save for later.
R24-I Must not have any plastic edges exposed to animals because they may chew through the plastic.

2.5 Reliability and Durability

R25-I AutoFeed shall be durable for daily usage, minimizing any signs of wear.
R26-II The door must open and close when scheduled.
R27-II The Raspberry Pi must be able to detect a crash or fault in both its own system and the Arduino, and restart the malfunctioning device accordingly.

2.6 Usability Requirements

R28-III The creation of events for sixteen separate cages should be self-explanatory with limited training needed.

R29-III Feedback from the push button must be sent to the user in the case of a malfunction.
R30-III The user should be able to schedule a weeks' worth of feeding without having to check the cages

2.7 Standards Requirements

R31-I All electronics, cables and connections should be compliant with CSA Canadian Electrical Code in regards to safety

2.8 Security Requirements

R32-III The user shall have their information password protected.
R33-III The user shall be able to access the calendar remotely from the Raspberry Pi without compromising personal credentials.
R34-III There shall be no storing of user credentials anywhere in our system

3 MICROCONTROLLER REQUIREMENTS

The controlling unit consists of an Arduino microcontroller and a Raspberry-Pi. The main purpose of this unit is to communicate between the user's Google Calendar feeding schedule and the motors of the sliding doors on each feeder. The Raspberry Pi shall be connected to the World Wide Web via an Ethernet cable.

R35-II The microcontroller shall receive signals from the Raspberry-Pi via a USB cable type A to Type B.
R36-II The microcontroller and the Raspberry Pi shall be powered by separate power adapters connected to a wall outlet.
R37-II Jumper cables shall be used to connect the microcontroller to the external units.
R38-II The Raspberry Pi shall be accessible via LAN through password protected SSH as well as by conventionally connecting a monitor.
R39-II The Arduino board should have at least 8 PWM outputs to power the electric motors that open and close the food doors.

4 MECHANICAL REQUIREMENTS

The main bulk of our project relies in the functional requirements of the motorized sliding door. This door must function properly and regularly in order for the project to be a success.

R40-I The door must prevent feeding during closed periods and allow feeding during open periods.
R41-I The housing must be designed such that food does not get into the mechanical gears or gear train.

R42-I The door must be able to open without reaching past the bottom of the cage as it would be blocked by the ledge on which the cage rests.
R43-I The door must be able to open and close while the gear tracks keep in contact with the gears as to maintain the position of the door.

5 GOOGLE CALENDAR REQUIREMENTS

We are integrating our product with 3rd party (Google) software. Therefore, we must be aware of current all protocols or transitions happening in the Google APIs. The following specifications will allow us to ensure we can successfully integrate AutoFeed with a Google Calendar.

R44-II Must use version 3 of Google API with OAuth 2.0 support to prevent code from becoming obsolete.
R45-II Must not hard code any user credentials and fully adopt OAuth 2.0 as the means for user authentication.
R46-II Must handle and renew authentication tickets from Google to successfully receive Google Calendar data.
R47-II Must develop a simple but robust standard for how to couple different calendar events to different cages such that each cage can be controlled individually

6 USER DOCUMENTATION

To guarantee that the user has no difficulties adopting AutoFeed into their practices, we need to supply a comprehensive user`s manual. This section documents the functional specifications of a non-technical manual for AutoFeed.

R48-III The user manual shall have a logo, email address, and phone number that users can contact.
R49-III The user manual should be intuitive and easy to use and targeted to an academic audience without technical knowledge.
R50-III The user manual should have steps on how to maintain the AutoFeed with step-by-step explanation and appropriate illustrative pictures.
R51-III The user manual shall be written in English.
R52-III The user manual shall have directions on proper cleaning techniques for various pieces.
R53-III The user manual should explain the mounting procedure on new cages.

7 SYSTEM TEST PLAN

The AutoFeed product will be tested on live rats and mice before it is available on the market. For disclosure, the premise behind AutoFeed is that it is designed to allow and disallow access to food for lab animals, namely rats and mice. Therefore testing procedures will be done on mice and rats. This does not prevent the use of AutoFeed to animals other than mice or rats, however, we cannot guarantee that the AutoFeed product will achieve user satisfaction if it is used on these other animals.

Test Items

I. Test that the dispensing apparatus does not injure the animals in any way
II. Test that when the slider door is open, the animals can reach the food, and not be blocked by the width of the bars
III. Test that the hopper is of the correct size to hold enough food for one fully grown rat (biggest consumer) for 1 week (roughly 300g). And that the food slides to within reach of the animals inside of the storage hopper.
IV. Test the reliability of the push buttons
V. Test the reliability of the rack/pinion and motor set
VI. Test to make sure that the leftover food particles don't get stuck in the mechanical system
VII. Multiple cages can be controlled by the Google Calendar to both open and close slider doors
VIII. Test that in the event of feedback push button not being pressed at the expected time, an email will be sent to a group of specified recipients
IX. Test possible power surges from exterior power source

Testing Procedure

Testing on individual features will be done throughout the prototyping process. However, once integrated, all of the features working together will still need to be put through strenuous test situations to assure the functionality of AutoFeed. The quality assurance of the AutoFeed product is particularly important because it will be accessible by live animals, of which are expensive and fragile because they are part of experimental lab groups. Therefore ensuring consistent and safe food deployment and removal is crucial to the systems overall function. After each of the individual features of AutoFeed are tested, the system will then be put through the following set up tests, that it is required to pass, before being put into production.

1. **Single Cage Without living Specimen:** The dispensing system will be filled with the food pellets that will be given to the rats during feeding time. The system will be controlled by the Google Calendar, where it will be programmed to close the slider door, and open the slider door, multiple times within the testing period. To test the email notification system, the slider door will be manually blocked to stop it from pressing the feedback switch. This test will satisfy the following test conditions: **IV), V), VI), VIII).**

2. **Single Cage With Living Specimen:** The cage will then be populated by a living rat of the same species that will be in the laboratory. The testing apparatus will be set up otherwise the same as in the first test. The rat will be

left in the cage for at least 2 days. This test will satisfy the following conditions: **I), II), II), VI).**

3. **Multiple Cage Without Living Specimen:** Multiple cages will be set up with the AutoFeed dispensing apparatus and connected to the controller. Each cage will be programmed with a different feeding schedule and will be verified if they are accurate to the required times. This test will satisfy the following conditions: **iv), v), vii).**

4. **Connect the Fuse to a Power Supply:** Test the maximum current that can be put through the fuse. This test will satisfy condition **ix).**

8 CONCLUSION

The functional specifications listed above describe all behavioural requirements for an adequate AutoFeed product. Moving into the design phase of our project, Optimaus is committed to including all of these functional specifications into AutoFeed. At Optimaus, our priority is to develop a solution that animal feeding laboratories want to use. We believe that, with the implementation of these core features, AutoFeed will develop into a valuable addition to the animal research laboratories.

9 REFERENCES

Raspberry Pi Foundation. (2014). *Getting Started with Raspberry Pi* [Online]. Available: http://www.raspberrypi.org/learning/teachers-classroom-guide/getting-started-guide.md

Arduino. (2014). *Getting Started with Arduino* [Online]. Available: http://arduino.cc/en/Guide/HomePage

M. Barr. (2007). *Introduction to Pulse Width Modulation (PWM)* [Online]. Available: http://www.barrgroup.com/Embedded-Systems/How-To/PWM-Pulse-Width-Modulation

Google Developers. (2014). *Get Started with the Google Calendar API* [Online]. Available: https://developers.google.com/google-apps/calendar/get_started