**Generate My Schedule**

**(G.M.S.)**

CS 3337 Software Engineering

Functional Requirements and Software Design Document (FRD/SDD)

Prepared by:

Group #14

Rawad Moussa

Alex Lamb

Tony J. Hong

Kevin Kazaryan

May 9, 2020

CALIFORNIA STATE UNIVERSITY,

LOS ANGELES



Los Angeles, California

**Generate My Schedule**

(G.M.S.)

Prepared By:

Rawad Moussa  
Alex Lamb

Tony J. Hong  
Kevin Kazaryan

Approved By:

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_May 9, 2020\_\_\_\_\_

Jose Macias Date

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_May 9, 2020\_\_\_\_\_\_

Jose Macias Date

**Document Change Log**

|  |  |  |
| --- | --- | --- |
| Update | Date Released | Changes |
| Draft #1 | 3/1/2020 | Delivery of the FRD/SDD, 1st draft |
| Update | 4/15/2020 | FRD/SDD delivery |
| Final Update | 5/8/2009 | Final FRD/SDD Delivery |

List of TBD Items

|  |  |  |  |
| --- | --- | --- | --- |
| Page | Item | Description | Status |
| Section 2 | Completion and corrections | To be completed during the 2nd delivery of the document | Completed in April 2020 |
| Section 4 | Modules Code | To be completed before final delivery | To be delivered in May 2020 |

**Table Of Contents**

Section Page

1.0 Introduction……………………………………………………………………………………… 7

1.1 Purpose …………………………………………………………………………………………… 7

1.2 Scope …………………………………………………………………………………………….. 7  
1.2.1 Document Organization ……………………………………………………………… 7

1.2.2 Relationship to Other Documents ……………………………………………… 8

1.3 GMS Architecture …………………………………………………………………………… 8

1.3.1 Context Diagram (Level 0 Data Flow Diagram) ………………………… 8

1.3.2 Description and Major Functions ..……………………………………………… 8

1.3.3 Software Considerations ……………………………………………………………… 9

1.4 Documentation of the Development Process ……………………………….. 9

1.5 References ………………………………………………………………………………………. 10

1.5.1 Controlling Documents ………………………………………………………………… 10

1.5.2 Applicable Documents …………………………………………………………………. 10

1.5.3 Standards …………………………………………………………………………………….. 11

2.0 GMS Detailed Function Description ………………………………………………… 11

2.1 Detailed GMS Functional Description ……………………………………………… 11

2.1.1 Level 1 Data Flow Diagram (DFD) ……………………………………………… 12

2.1.2 Detailed Functional Description of GMS Major Sub-Units…………… 13

3.0 GMS Requirements ……………..……………..……………..……………..………….. 14

3.1 GMS Functional Requirements ……………..……………..……………..………… 16

3.2 GMS Non-Functional Requirements ……………………………………………….. 16

A. **Acronyms** ……………………………………………………………………………………….. 17

B. **Data Dictionary**…………………………………………………………………………….. 17

**1.0 INTRODUCTION**

**1.1 Purpose**

The purpose of this document is to:

1. Define a full set of requirements for each of this project’s compositional modules. These sections correspond to a Software Requirements Document, SRD. The requirements are to be used in the implementation (coding) of the software written as part of this project.
2. Define the design for this project. These sections correspond to a Software Design Document (SDD). The design is to be used as a basis for the DFD level 0 and level 1 (DFD 0 and DFD 1).
3. Document the implementation (coding) of this project selected major modules. These sections correspond to a Software Implementation Document (SID). (Time constraints may allow only a partial level of completion).
4. Serve as the basis for the charts to be used during this project oral presentation.

**1.2 Scope**

This documentation is developed as part of the cs3337 project. The Implementation (Section 4) corresponds to a selection of these requirements.

The scope of this document includes the following:

* Project functional and non-functional requirements. These requirements are organized by the key functional units shown on the Level 1 DFD given in section 2.0.
* A trace matrix, relating functional requirements to functional units as described in the DFD 1. Higher level DFDs will not be included/required for this project.
* General descriptions of the hardware that may be necessary for implementation of the project.

**1.2.1 Document Organization**

The organization of this document provides a natural 'flow' or allocation of requirements to each unit included in the DFD level 1 shown in section 2.

**1.2.2 Relationship to Other Documents**

This is a self-contained document. Relationship to other documents in the literature, if any, is given in sub-section 1.5.

**1.3 GMS Architecture**

**1.3.1 Context Diagram (DFD Level 0)**

GMS’s architecture is summarized in the Context Diagram (DFD Level 0) given below. A more detailed Functional Description is given in Section 2 of this document.

Figure 1-1: Level 0 DFD

Optimized

Schedule

**Generate My Schedule  
(GMS)**

1. Main Control
2. Info Management
3. Schedule API
4. Schedule Optimization Visualizer
5. Schedule Distribution

User Input

DFD Level 0

Distinct Employee Schedules

Optimization Visualization

**1.3.2 Description and Major Functions of GMS**

GMS will generate an optimized schedule for employees to minimize labor costs. The program will analyze information about the team of employees and the available working shifts in order to generate an efficient working schedule. GMS will also have the ability to let users send the generated schedules to the user’s employees.

**1.3.3 Software Considerations**

GMS requires specific computer software such as:

* Software that handles accessing and manipulating the inputs in a file.
* Outside software that handles the communication between the user and employees.

**1.4 Documentation of the Development Process**

GMS’s detailed functional description is documented in section 2.0. Section 2 is a brief software description document. The overall detailed functional description is based on higher level DFDs (above level 1). All major functional units are described in detail in this part of the document.

Requirements for GMS are captured in Section 3.0 of this document. This section includes both functional and non-functional software requirements, along with more detailed information about each unit.

**1.5 References**

“Dash Documentation & User Guide.” Plotly, dash.plotly.com/.

Mitchell, Stuart, et al. PuLP: A Linear Programming Toolkit for Python. Department of Engineering Science, The University of Auckland, Auckland, New Zealand, 5 Sept. 2011, www.optimization-online.org/DB\_FILE/2011/09/3178.pdf.

“Optimization and Root Finding (Scipy.optimize)¶.” Optimization and Root Finding (Scipy.optimize) - SciPy v1.4.1 Reference Guide, docs.scipy.org/doc/scipy/reference/optimize.html.

Sarkar, Tirthajyoti. “Linear Programming and Discrete Optimization with Python Using PuLP.” *Medium*, Towards Data Science, 25 Apr. 2019, towardsdatascience.com/linear-programming-and-discrete-optimiza tion-with-python-using-pulp-449f3c5f6e99.

“Welcome to Flask¶.” Welcome to Flask - Flask Documentation (1.1.x), flask.palletsprojects.com/en/1.1.x/.

**1.5.1 Controlling Documents**

“Macias, José, and Richard Cross. “OFFICIAL Project Documentation SRD-SDD Template - Must Use This One.” CS3337 Software Engineering, Https://Csns.calstatela.edu/, csns.calstatela.edu/site/s20/cs3337-1/item/7725647.

**1.5.2 Applicable Documents**

No additional applicable document has been used in the production of this document.

**1.5.3 Standards**

One standard has been used in the creation of this document in 1.5.1 Controlling Documents was used as reference as provided by both Doctors. However, some Standards described in textbooks have been examined as a reference. In particular, the IEEE standard has been briefly discussed in class.

**2.0 DETAILED FUNCTIONAL DESCRIPTION OF GMS**

**2.1 Detailed GMS Functional Description.**

The major tool used to design GMS is the Data Flow Diagram, DFD. The rationale for the selection of DFDs as the preferred design tool is its simplicity and versatility. In the future additional tools may be used if a stronger correlation from Design to Requirement to Implementation and Testing is required.

**2.1.1 Level 1 DFD**

GMS’s major functional subunits are shown in the DFD Level 1 shown below:

Figure 2.1: Level 1 DFD

User

Employees

**User Access**

**Optimized Schedules**

**Operational Data**

**User Input**

**Optimized**

**Schedules**

**Operational Data**

**Optimized**

**Schedules**

**Optimized**

**Schedules**

**Info Request**

**Operational Data**

**Email Schedules**

Database

DFD Level 1

**Optimization Visualization**

**Optimization Visualization**

Employees

**Optimized**

**Schedules**

**2.1.2 Detailed Functional Description of GMS’s Major Units.**

The description of GMS’s major functional units shown in Figure 2.1 follows.

Main control - Module 2.1

The Main Control Module (MCM) acts like the information processing system in the program. All information is sent to this unit before being transferred to other modules. When the initial data is input, it is then transferred through this module, before being sent to each other module for further manipulation and optimization.

Info Management - Module 2.2

The Info Management Module (IMM) lets the user access files about the employee(s), their availability, and the labor demand from the database. The IMM also takes a set of parameters from the user. These parameters specify the schedule start/end and the acceptable shift lengths. All this data is then sent back to the MCM.  
  
Schedule API - Module 2.3

The Schedule API Module (SAM) takes the employee data, employee availability, labor demand, and specified set of parameters from the MCM. This data is then formatted to interface with a library of custom optimization functions. The data is then processed; the schedules are optimized, and the data is sent back to the MCM.

Schedule Visualizer - Module 2.4

The Schedule Visualizer Module (SVM) takes the optimized schedule data from the MCM and renders a graph displaying the labor need vs. the scheduled labor. The SVM also displays several metrics pertaining to the schedule to give a user a visual representation of how effective the schedule is at meeting demand while considering cost.

Schedule Distribution Module - Module 2.5

The Schedule Distribution Module (SDM) also takes the optimized schedule data from the MCM. The SDM then prompts the user to input his email username and password. After inputting the email info, The SDM then emails each employee their distinct schedule.

**3.0 GMS REQUIREMENTS**

**3.1 GMS Functional Requirements**

This Section lists GMS’s functional requirements. This section includes the complete set of functional requirements, along with explanations for cases in which the statement of the requirement was deemed insufficient or requires additional clarification. All requirements relate to the design modules described in Section 2. An effort has been made to standardize the correlation between the design modules and the requirements to make access and organization more consistent. For example, requirement number “n” affecting module 2.1 will be labeled 3.1.n

|  |  |
| --- | --- |
| **Module 2.1: Main Control Module (MCM)** | |
| Requirement No. | Requirement Description |
| 3.1.1 | The MCM shall be accessible to the user in a web application. |
| 3.1.2 | The MCM shall interface all modules together. |
| 3.1.3 | The MCM shall request data from the database once. |
| 3.1.4 | The MCM shall store all necessary employee data in application scope. |
| 3.1.5 | The MCM shall store user defined parameters in application scope. |
| 3.1.6 | The MCM shall allow the user to store optimized schedules in the database after they are returned from the SAM. |
| 3.1.7 | The MCM shall receive the optimization visualization from the SVM. |
| 3.1.8 | The MCM shall display the optimization visualization for the user. |

|  |  |
| --- | --- |
| **Module 2.2: Info Management Module (IMM)** | |
| Requirement No. | Requirement Description |
| 3.2.1 | The IMM shall allow users to access their employees’ personal data from the database. |
| 3.2.2 | The IMM shall allow users to access their employees’ personal availability from the database. |
| 3.2.3 | The IMM shall allow users to access their labor demand from the database. |
| 3.2.4 | The IMM shall take a parameter from the user defining the range of acceptable shift lengths. |
| 3.2.5 | The IMM shall take a parameter from the user defining the operation hours of the schedule. |
| 3.2.6 | The IMM shall send this information back to the MCM to be optimized by the API. |

|  |  |
| --- | --- |
| **Module 2.3: Schedule API Module (SAM)** | |
| Requirement No. | Requirement Description |
| 3.3.1 | The SAM shall assign at most one shift to each employee daily |
| 3.3.2 | The SAM shall receive all data from the MCM after being accessed by IMM. |
| 3.3.3 | The SAM shall receive parameters from the IMM. |
| 3.3.4 | The SAM shall generate the best usage of shifts to cover the labor requirements. |
| 3.3.5 | The SAM shall generate the optimal assignment of employees to the optimal shifts. |
| 3.3.6 | The SAM will determine what is optimal by what assignment is the most cost efficient. |
| 3.3.7 | The SAM will return the optimized schedules to the MCM. |

|  |  |
| --- | --- |
| **Module 2.4: Schedule Visualizer Module (SVM)** | |
| Requirement No. | Requirement Description |
| 3.4.1 | The SVM shall receive the optimized schedule data from the MCM. |
| 3.4.2 | The SVM shall graph the labor need compared to the scheduled labor. |
| 3.4.3 | The SVM shall calculate the cost of the optimized schedule. |
| 3.4.4 | The SVM shall return this info to the MCM to be displayed. |
| 3.4.5 | The SVM shall not store this graph beyond application scope. |

|  |  |
| --- | --- |
| **Module 2.5: Schedule Distribution Module (SDM)** | |
| Requirement No. | Requirement Description |
| 3.5.1 | The SDM shall take the user’s email. |
| 3.5.2 | The SDM shall take the user’s password. |
| 3.5.3 | The SDM shall read the optimized schedule data from the MCM. |
| 3.5.4 | The SDM shall group all shift by the assigned employee id. |
| 3.5.5 | The SDM shall send each employee the corresponding schedule. |
| 3.5.6 | The SDM shall send the emails from the user’s entered email. |

**3.2 GMS Non-Functional Requirements**

This Section collects all the GMS Non-Functional Requirements. All non-functional requirements are numbered “NF – n” where “n” indicates the nth requirement.

|  |  |
| --- | --- |
| NF-1 | GMS must be easily operable by non-technical users. |
| NF-2 | GMS must be quick enough to avoid frustration by users. |
| NF-3 | GMS must output in a format that is highly interfaceable with other business tools. |
| NF-4 | GMS must be scalable to handle different business use-cases. It should be able to handle varying business applications. |
| NF-5 | GMS must be highly accessible by users to facilitate varying business applications. |

A. ACRONYMS

**MCM:** Main Control Module

**IMM:** Info Management Module

**SAM:** Schedule API Module

**SOV:** Schedule Optimization Visualizer

**SDM:** Schedule Distribution Module

B. DATA DICTIONARY

**Highly Accessible:** accessible from the web on common systems in business environments: Windows, MacOS, Linux.

**Highly interfaceable:** A format that is recognizable unanimously. This shall allow the product to be custom fit to business applications. Csv suits this purpose as it is easily manipulatable using Excel or GoogleSheets.

**Non-Technical:** a person who is unskilled in math or business logic. No programming or computer experience beyond an average high school level.

**Quick enough**:The application should be satisfyingly faster than calculating the shifts manually. The user should be granted a sense of satisfaction in how effortlessly his employees receive their schedules.

**Varying business applications:** Should be able to fit most labor demands of differing employee size including but not limited to commercial, private and government sectors.