|  |  |
| --- | --- |
|  |  |

Technical Specification Document



Project Name: SQL Server Emailing System

Issue Title: SDM 31462 – Hard Coded Emails

Date: September, 2022

# Prepared By

|  |  |
| --- | --- |
| **Document Owner(s)** | **Project/Organization Role** |
| Justin Pope | MIS |

# Modification Version Control

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Issue. Version** | **Date** | **Author** | **Tag ID** | **Change Description** |
| V1.0 | 09/01/2022 | Justin Pope | SDM 31462 | Hard Coded Emails |

# Table of Contents

[Prepared By 2](#_Toc96335885)

[Modification Version Control 2](#_Toc96335886)

[Table of Contents 3](#_Toc96335887)

[1 Design Overview 5](#_Toc96335888)

[1.1 Business Case/Scope 5](#_Toc96335889)

[1.2 Audience 5](#_Toc96335890)

[1.3 Reference 5](#_Toc96335891)

[1.4 Terms and Definitions 6](#_Toc96335892)

[1.5 Technical Design Diagram 6](#_Toc96335893)

[1.5.1 Application Architecture Diagram 6](#_Toc96335894)

[1.5.2 Integration Architecture Diagram 8](#_Toc96335895)

[1.6 Assumptions/Dependencies 11](#_Toc96335896)

[1.7 Outstanding Questions /Open issues 11](#_Toc96335897)

[2 Data Design 11](#_Toc96335898)

[2.1 Database Management System Files 11](#_Toc96335899)

[2.2 Non-Database Management System Files 11](#_Toc96335900)

[3 Detailed Design 12](#_Toc96335901)

[3.1 Hardware Detailed Design 12](#_Toc96335902)

[3.2 Software Detailed Design 12](#_Toc96335903)

[3.2.1 Module [X] 13](#_Toc96335904)

[3.2.2 Module [X] 13](#_Toc96335905)

[3.3 Communications Detailed Design 13](#_Toc96335906)

[4 External Interface Design 13](#_Toc96335907)

[4.1 Interface Architecture 13](#_Toc96335908)

[4.2 Interface detailed design 13](#_Toc96335909)

[5 Human-Machine Interface 14](#_Toc96335910)

[5.1 Interface Design Rules 14](#_Toc96335911)

[5.1.1 Inputs 14](#_Toc96335912)

[5.1.2 Outputs 14](#_Toc96335913)

[5.2 Navigation Hierarchy 15](#_Toc96335914)

[5.2.1 Screen [x.1] 15](#_Toc96335915)

[5.2.3 Screen [x.2] 15](#_Toc96335916)

[6 Appendices 15](#_Toc96335917)

[Appendix A: Analysis Documentation 15](#_Toc96335918)

[Appendix B: Issues 15](#_Toc96335919)

[Appendix C: Other supporting documentation 15](#_Toc96335920)

[Appendix D: Benchmark Design and Development Estimate 15](#_Toc96335921)

# Design Overview

SDM 31462 – Hard Coded Emails was created when researching a problem with emails that were being sent that were hard coded to send to an employee that no longer was employed at Gabreilla White. This issue raises the need for a system to track what emails are in the system and being sent.

## 1.1 Business Case/Scope

This system is primarily to be used to track emails and have a single reference point for email related data. Secondarily, it can be used to send emails that are stored on the system.

It will be standardized for all SQL Server based emails to use and follow through this system. This is not intended to replace the current notification system that is established for services but eventually can be implemented with this system.

## 1.2 Audience

This is a technical document targeted for technical readers that will maintain the designed application. It is assumed the reader has some technical background related to the technologies mentioned in the above description

## 1.3 Reference

List all references that are included in the document. Also include applicable policies and procedures associated with the document and its contents.

| Ref# | Document Name | URL |
| --- | --- | --- |
| 1 | System Diagram | [sample-integration-architecture.drawio - diagrams.net](https://app.diagrams.net/) |

## 1.4 Technical Design Diagram

The following diagram is how the email system is to be used.

### 1.5.1 Application Architecture Diagram

A picture containing table

Description automatically generated

Figure - System Architecture

The email system consists of a header table, message table, and a procedure.

### 1.5.2 Integration Architecture Diagram

* How are my services currently organized and exposed internally/externally?
* How does the partner want to integrate with my system, e.g., internal networks, protocols, etc.?
* How do I secure, track, and manage the integration of my exposed services?

Use useful components in the diagram:

* Grouped components into layers and bounded contexts – an indication of internal/external API gateways and services
* Annotations with additional information – references links to API catalogues where detailed service data attributes can be obtained to assess gaps
* Application details and context – services are named appropriately to allow quick assessment of requirements vs. actual
* Example below:

Diagram

Description automatically generated

## 1.6 Assumptions/Dependencies

1. List assumptions and concerns
2. ……..
3. ……..

## 1.7 Outstanding Questions /Open issues

1. List questions and open issues
2. ……..
3. ……..

# Data Design

This section outlines the design of the database management system (DBMS) and non-DBMS files associated with the system. Provide a data dictionary showing data element name, type, length, source, validations rules, maintenance, data stores, outputs, aliases, and description. Attach as an appendix if needed

## 2.1 Database Management System Files

Describe how the database will be designed including the follow information as appropriate:

* Logical model; provide normalized table layouts, entity relationship diagrams and other logical design information
* DBMS schemas, sub-schemas, records, sets, tables, storage page size, etc.
* Access methods (such as indexed, via set, sequential, random access, sorted pointer array, etc.)
* Estimate the database file size or volume of data within the file
* Definition of the updates frequency of the database tables, views, and files.
* Estimates on the number of transactions the database may have to process

## Non-Database Management System Files

Describe all non-DBMS files including narratives on the usage of each file. Identify if the file is used for input, out or both; identify temporary files; which modules read and write the file, etc.

* Identify records structures, record keys, indexes, and reference data elements within the records
* Define record length and blocking factors
* Define file access method, such as index-sequential, virtual sequential, random access, etc.
* Estimate the file size or volume of data within the file

# Detailed Design

This section describes the proposed design in detail. Provide the necessary information for the development team to integrate the hardware components, write the software code, so that the hardware and software components will provide a functional product. Every design item should map back to the Functional Requirements Document. Captured in the Requirements Traceability Matrix

## 3.1 Hardware Detailed Design

In this section, provide enough information for the developers to build and/or procure the system’s hardware. Include the following information (as applicable)

* Connector specifications
* Details of hardware items, such as servers, printers etc.
* Memory and storage space requirements
* Processor requirements

## 3.2 Software Detailed Design

Describe the overall system software and organization. List and describe the software modules (including functions, subroutines, or classes), programming languages and development tools. Describe all software required to support the system and specify the physical location of all software systems. Identify database platforms, compilers, utilities, operating systems, communications software, etc.

Provide diagrams that illustrate the segmentation levels down to the lowest level. Include names and reference numbers for all features on the diagrams. Include a narrative that expands on and enhances the understanding of the functional breakdown.

In this section provide enough detailed information for the developers to write the source code for all modules in the system and/or integrating software programs

For each module, provide the following information:

* Narrative introduction to each module, its function(s), the conditions under which it is used (called or scheduled), process, logic, interfaces to other modules, interfaces to external systems, security requirements, etc.
* Graphical representation of the module processing, logic, flow of control, and algorithms, using charts, diagrams, flowcharts as appropriate
* Data elements, record structure, and file structures associated with module input and output.
* Report layout
* Specify any call routines or bridging programs to integrate the application.

### 3.2.1 Module [X]

Processing

Provide a processing narrative for each module. Explain the process by which each module interacts with other parts of the system, including other modules. Describe the data elements and data structures which provides input to each module, how the module transforms the data, and the data elements which are output

### 3.2.2 Module [X]

Processing

Provide a processing narrative for each module. Explain the process by which each module interacts with other parts of the system, including other modules. Describe the data elements and data structures which provides input to each module, how the module transforms the data, and the data elements which are output

## 3.3 Communications Detailed Design

Describe communications within the system, such as Local Area Networks (LANs), buses, etc. Include the communications architecture(s) being implemented. Provide a diagram depicting the communications path between the system and subsystem modules

# External Interface Design

## 4.1 Interface Architecture

## 4.2 Interface detailed design

# Human-Machine Interface

Describe the Human-machine Interface (i.e., user interface) relative to the user.

## 5.1 Interface Design Rules

Identify conventions and standards for designing the user interface

### 5.1.1 Inputs

Identify the input media used by the user for providing information to the system, such as data entry screens, optical character readers, bar scanners, etc.

Identify the messages associated with operator inputs, including the following:

* Form(s) if the input data is keyed or scanned for data entry
* Access restrictions
* Security considerations

### 5.1.2 Outputs

Describe the system output design relative to the user. System outputs include reports, data display screens, query results, etc.

Identify the following, if appropriate:

* Access restrictions or security considerations
* Description of the purpose of the output
* Report requirements, including frequency for periodic reports
* Screen contents (provide a graphic representation of each layout. Define all data elements associated with the layout)

## 5.2 Navigation Hierarchy

Provide a diagram of the navigation hierarchy that shows how a user moves through the user interface

### 5.2.1 Screen [x.1]

Provide the layout of all input data screens or graphical user interfaces. Provide a graphic representation of each interface, for example, a low-resolution screenshot. Define all data elements associated with each screen or GUI or reference the data dictionary.

### 5.2.3 Screen [x.2]

Provide the layout of all input data screens or graphical user interfaces. Provide a graphic representation of each interface, for example, a low-resolution screenshot. Define all data elements associated with each screen or GUI or reference the data dictionary.

# Appendices

## Appendix A: Analysis Documentation

## Appendix B: Issues

## Appendix C: Other supporting documentation

## Appendix D: Benchmark Design and Development Estimate

Benchmark Design and Development estimate. Please list the task name, hours needed to complete and a benchmark estimate date of completion. **Remember to add task and hours for testing, documentation and go live support**

| Task | Hours | Benchmark est. Date |
| --- | --- | --- |
| <description > | <hours> | <MM/DD/YYYY> |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |