System Design & Technical Architecture

****

|  |  |
| --- | --- |
| Information Systems Branch | Economy Sector |

Version <version number> - <draft or final>

August 22, 2017

Table of Contents

[Revision History 4](#_Toc491166667)

[1. Overview 5](#_Toc491166668)

[1.1. Purpose 5](#_Toc491166669)

[1.2. Business Context 5](#_Toc491166670)

[1.3. In Scope 5](#_Toc491166671)

[1.4. Out of Scope 5](#_Toc491166672)

[2. Abbreviation List 6](#_Toc491166673)

[3. Roles and Responsibilities 7](#_Toc491166674)

[4. Project References 8](#_Toc491166675)

[5. Design Considerations 9](#_Toc491166676)

[5.1. Assumptions 9](#_Toc491166677)

[5.2. Constraints 9](#_Toc491166678)

[5.3. Risks 9](#_Toc491166679)

[5.4. Contingency 9](#_Toc491166680)

[5.5. Strategies 9](#_Toc491166681)

[6. Technical Architecture 11](#_Toc491166682)

[6.1. System Overview 11](#_Toc491166683)

[6.2. Logical View 11](#_Toc491166684)

[6.3. Component View 11](#_Toc491166685)

[6.4. Component Descriptions 11](#_Toc491166686)

[6.5. Technical Requirements 12](#_Toc491166687)

[6.5.1. Software 12](#_Toc491166688)

[6.5.2. Hardware 12](#_Toc491166689)

[6.5.3. Capacity 12](#_Toc491166690)

[6.5.4. Dependencies / External Services 12](#_Toc491166691)

[6.6. DevOps 12](#_Toc491166692)

[6.7. Solution Structure 12](#_Toc491166693)

[6.8. Data Access 12](#_Toc491166694)

[6.8.1. Data Management 13](#_Toc491166695)

[6.9. Security 13](#_Toc491166696)

[6.9.1. Authentication 13](#_Toc491166697)

[6.9.2. Authorization 13](#_Toc491166698)

[6.10. Auditing 13](#_Toc491166699)

[6.11. Caching 13](#_Toc491166700)

[6.12. Logging 14](#_Toc491166701)

[6.13. Testing 14](#_Toc491166702)

[7. Privacy 15](#_Toc491166703)

[8. Database Design 16](#_Toc491166704)

[8.1. System Files 16](#_Toc491166705)

[9. System Integrity Controls 17](#_Toc491166706)

# Revision History

|  |  |  |  |
| --- | --- | --- | --- |
| Date | Version | Author | Description |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

# Overview

## Purpose

This section should provide a high-level description of the purpose of the System Design & Technical Architecture Document. This may include a description of how the document relates to organizational goals and/or objectives and how the new system will meet those goals and objectives.

This document provides a high-level description of the technical solution for the <Insert Project Name> project. This solution was conceived with the goal of addressing all requirements described in the <Insert Project Name> Business Requirements Document. This document describes the major components that will make up the solution, the dependencies between them, and how they will work together.

This document is intended to:

* Capture all design considerations
* Capture architecture level technical design decisions.
* Convey the technical architecture to the Ministry Technical Architect for review and approval.
* Support the onboarding of new developers into the project more efficiently and effectively.
* Enable a smoother transition into operations (DevOps).

This is a living document. It is expected that the technical architects (both Ministry and Vendor) will update this document as the project progresses until the project is complete and the system is promoted into production.

## Business Context

Provide a brief summary of the business problem the application will solve.

## In Scope

Provide a list/description of all in-scope areas of the application this document covers.

## Out of Scope

Provide a list/description of all out of scope areas of the application this document does not cover.

# Abbreviation List

Provide a list of abbreviations and descriptions for abbreviations that are unique to this application and are used in this document.

|  |  |
| --- | --- |
| Abbreviation | Description |
|  |  |
|  |  |

# Roles and Responsibilities

System design can cross many different groups within an organization to ensure requirements are gathered and met for all stakeholders. As such, a roles and responsibilities section may be necessary to provide the team with clarification on who performs various roles. This section also serves as a list of points of contact for the team and stakeholders should issues and concerns arise which need to be addressed.

# Project References

This section should describe what references exist which guide the system design. These references may be internal or external. Examples of references include white papers. System analyses, organizational standards, industry standards, meeting minutes/summaries, and findings. This section should provide a list of such references but the descriptions should be general and not include much detail since the documents on the list can be referred to individually if more information is needed.

# Design Considerations

## Assumptions

Provide a list/description of all major assumptions made prior to or during business requirements gathering and documentation. These may concern such issues as: related software or hardware, operating systems, end-user characteristics, and possible and/or probable changes in functionality.

## Constraints

This section should describe the constraints associated with the system design. Constraints are the result of various conditions beyond the scope of the project that affect and limit the system design. These may be due to hardware, software, business processes, organizational/industry standards, or other conditions which affect the system design. This section should provide a description of what the constraints are and how they affect or limit the system design.

## Risks

Identify any risks or roadblocks which may impact cycle/project delivery and proposed mitigation strategies

## Contingency

Identify backup mechanisms in place in case of extended downtime.

(what processes are affected and how can business carry on)

## Strategies

* Describe any design decisions or strategies that affect the overall organization of the system.
* Describe the reasoning for each decision/strategy, and how priorities were balanced or traded-off.
* Describe significant alternatives that were considered and the reasons for rejecting them

Design decisions may concern (but are not limited to) things like:

* Use of a particular tool or library
* Future plans for extending or enhancing the system
* Error detection and recovery
* Persistence
* Concurrency
* Communication mechanisms
* Resource management

# Technical Architecture

## System Overview

Provide a high-level diagram/description of the application architecture. Describe how the higher-level components collaborate with each other in order to achieve the required results.

Example diagram:



## Logical View

Insert any related logical views or provide a reference to where they are stored. Include data flow diagrams related to the logical process.

Outline the functional breakdown, describing how a component was further divided into subcomponents, and the relationships and interactions between them.

If there are parts of the system that already existed, then only describe the relationships and interactions between the old parts and the new parts. Pre-existing parts that are modified or enhanced need to be described only to the extent that is necessary to provide a sufficient understanding of the nature of the changes that are being made.

## Component View

Provide a high-level diagram/description of the application architecture.

## Component Descriptions

Describe the components that make up the diagram

## Technical Requirements

### Software

List all the dependent software used by the application. For example:

* Database (version)
* Development tools
* 3rd party tools
* DevOps tools
* Data conversion tools

### Hardware

If applicable, describe any minimum hardware requirements for the application

### Capacity

Describe current and projected CPU, memory and disk space usage for the application

### Dependencies / External Services

If applicable, describe any dependencies on and interoperability requirements for external services

## DevOps

Describe process, tools, scripts used to support Ministry DevOps

For example:

* NodeJS – Packages used
* PowerShell scripts

## Solution Structure

Describe the organization and structure of the Visual Studio solution including the various projects/layers used

* Domain
* Services
* Persistence
* Reports

## Data Access

Describe the data access layer design including:

* ORM
* server and database logins
* security considerations (e.g. multiple connection strings)
* encryptio

### Data Management

Describe the following as applicable:

* data conversion
* migration
* archival implementation/strategy

## Security

Describe the measures included in the system design to ensure the system is secure and that the integrity of the system and data are maintained. This is an important consideration in the design of the system as failure to secure and control the system and its data can result in significant loss of time, money, and other resources.

Include all security considerations and decision points. For example:

* Authentication
* Authorization
* Encryption

### Authentication

Outline the authentication method used for the application. When SiteMinder is in use provide details on how SiteMinder is configured for the application (e.g. IDIR and/or BCeID authentication).

### Authorization

Outline the authorization mechanism used within the application and how Access Control Lists (ACL) are implemented.

### Auditing

Outline how application auditing is handled to withstand any legal obligations (accountability and traceability).

## Caching

Outline how caching and cache invalidation is handled at different levels (C#, JavaScript, CSS).

## Logging

Outline how application logging is implemented, where logs are stored, and any application/business specific considerations made.

## Testing

Outline the types of tests and their implementation considerations

* Unit Tests
* Automated UI Tests
* Profiling
* Load Tests

# Privacy

Describe information that will be stored in the system, and if any information is personally identifiable information (PII).

Identify data that needs to be redacted/cleansed/scrambled in order to send backups to external vendors for further system enhancements

# Database Design

Describe the design of all database management system (DBMS) files and non-DBMS files associated with the system. Provide a comprehensive data dictionary showing data element name, type, length, source, validation rules, maintenance (create, read, update, delete (CRUD) capability), data stores, outputs, aliases, and description. The Data Design information can be included as an appendix or recorded in a separate Database Design Document (DDD), as appropriate, which would be referenced here.

## System Files

Provide the detailed description of all non-DBMS files and include a narrative description of the usage of each file that identifies if the file is used for input, output, or both, and if the file is a temporary file. Also provide an indication of which modules read and write the file and include file structures (refer to the data dictionary). As appropriate, the file structure information should include the following:

* Estimate of the file size or volume of data within the file, including overhead resulting from file access methods
* Definition of the update frequency of the file (If the file is part of an online transaction-based system, provide the estimated number of transactions per unit of time, and the statistical mean, mode, and distribution of those transactions.)
* Backup and recovery specifications

# System Integrity Controls

Provide design specifications for the following minimum levels of control and any additional controls as appropriate or necessary:

* Internal security to restrict access of critical data items to only those access types required by users/operators
* Audit procedures to meet control, reporting, and retention period requirements for operational and management reports
* Verification processes for additions, deletions, or updates of critical data
* Ability to identify all audit information by user identification, network terminal identification, date, time, and data accessed or changed.