**SOFTWARE REQUIREMENT SPECIFICATION**

**FOR**

**<PROJECT NAME>**

**[17-12-2019]**

**CDAC, MUMBAI**

**Revision History:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Version** | **Date** | **Authored By** | **Reviewed By** | **Reason for Change** |
| 00 | 14/12/2019 | Team No. 6   1. Pooja Jadhav [2052020105]   2.  3.  4.  5.  6.  7.  8.  9.  10. | Mr. Amit Upadhyay | 1st release |

**Table of Contents**

|  |  |  |
| --- | --- | --- |
| **1. Introduction ..................................................................................................................................** | | **3** |
| 1.1 | Purpose .................................................................................................................................... | 3 |
| 1.2 | Scope ............................................................................................................ | 3 |
| 1.3 | Definitions, Acronyms,and Abbreviations .......................................................................... | 3 |
| 1.4 | References ........................................................................................................................... | 4 |
| 1.5 | Overview ................................................................................................................................ | 4 |

|  |  |  |
| --- | --- | --- |
| **2. Use Case Diagram........................................................................................................................** | | **5** |
| 2.1 | Diagram .................................................................................................................. | 5 |
| 2.2 | Description ..................................................................................................................... | 5 |

|  |  |
| --- | --- |
| **3. Class Diagram ………………………………………………………………** | |
| 3.1 | Diagram ........................................................................................................................... | 6 |
| 3.2 | Description…………………………………………………………………………… | 6 |

|  |  |
| --- | --- |
| 1. **Dynamic Model** | |
| 4.1 | Interaction diagram : Sequence Diagram.............................................................................….. | 7 |
| 4.2 | Description ................................................................................................... | 7 |
|  |  |  |
| 1. **Activity Diagram .............................................................................................** | | **8** |
| 5.1 | Diagram ...................................................................................................... | 8 |
| 5.2 | Description ................................................................................................................. | 8 |
|  | |  |
| 1. **Data Design…………………………………………………………………………………………** | | **9** |
| 6.1 Entity Relationship(ER) Diagram | | 9 |
| 6.2 ER to Relation Mapping | | 9 |
|  | |  |
| 1. **Supplementary Documentation** | | **10** |
| 7.1 Tools used to create diagrams | | 10 |

**Software Design Document**

**1 Introduction**

The Software Design Document is a document to provide documentation which will be used to aid in software development by providing the details for how the software should be built. Within the Software Design Document are narrative and graphical documentation of the software design for the project including use case models, sequence diagrams, collaboration models, object behavior models, and other supporting requirement information.

**1.1 Purpose**

The purpose of the Software Design Document is to provide a description of the design of a system fully enough to allow for software development to proceed with an understanding of what is to be built and how it is expected to built. The Software Design Document provides information necessary to provide description of the details for the software and system to be built.

**1.2 Scope**

This Software Design Document is for a base level system which will work as a proof of concept for the use of building a system the provides a base level of functionality to show feasibility for large scale production use. This Software Design is focused on the base level system and critical parts of the system. For this particular Software Design Document, the focus is placed on generation of the documents and modification of the documents. The system will be used in conjunction with other existing systems and will consist largely of a document interaction facade that abstracts document interactions and handling of the document objects.

**1.3 Definitions, Acronyms, and Abbreviations**

● **Data Objects –** Data objects are Java objects with predefined structures capable of holding data in a structure that is quickly and easily accessible by other parts of the software system. They provide also can help provide a convenient abstraction of the data in a database so that it can be retrieved into a format, such as a denormalized format, that makes access and manipulation of the data easier than if the database had to be called directly. http://java.sun.com/products/jdo/

● **Editable Form Layout**- A user interface presentation layout in which the contents of a document are presented to a user in the format of a form predefined editable areas based on the type of document which is being edited. This type of layout allows for changes to be made in a specific manner so that the data used in the form can be reassembled into a structured data format for transfer to other systems and archival.

● **JDBC/ODBC –** These two acronyms stand for Java Database Connectivity and Open Database Connectivity API's which allow for standardized database access and interaction from software products. JDBC: **http://www.learnthat.com/define/view.asp?id=106** .

ODBC: **http://en.wikipedia.org/wiki/ODBC**

● **PDF** – Portable Document Format, http://en.wikipedia.org/wiki/Portable\_Document\_Format

● **Required Field –** A critical field is a field in a data set for a document that is required for successful document generation. For example, missing parties in a case, missing county location of court, or other data elements that are required to create a valid legal document.

**1.4 References**

● XML Legal Documents Utility Software Development Plan

○ Version 1.0, Last Updated on 2007-01-31

**1.5 Overview**

The Software Design Document is divided into 11 sections with various subsections. The

sections of the Software Design Document are:

1 Introduction

2 Use Cases

3 Class Diagram

4 Dynamic Model

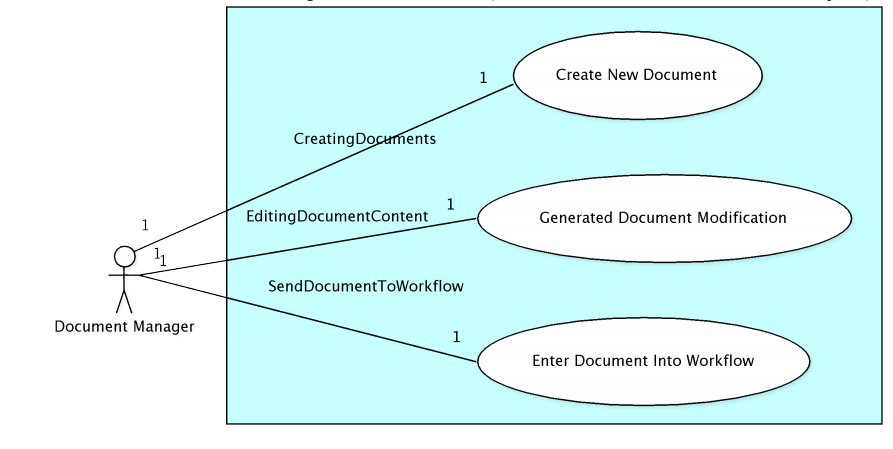
5 Activity Diagram

6 Data Design

7 Supplementary Documentation

1. **Use Case Diagram**
   1. Diagram

Document Manager- Essential Use Cases (“Enter Document into Workflow” for future update)



* 1. Description

The Document Manager is a user who works with legal documents. This is an abstraction of the specific users as they all perform similar actions, but for different reasons. For example, a court clerk and an attorney both sign documents, but an attorney does so to state that they created or agree to the documents and the court clerk does so to state that the document has been received and is now secured with a secure hash to detect modification. The mechanics and the processes used for each are the same to apply their respective digital signatures, but the intent and meaning of each application of a digital signature is different.

1. **Class Diagram**
   1. Diagram
   2. Description
2. **Dynamic Model**
   1. Interaction Diagram: Sequence Diagram
   2. Description
3. Activity Diagram
   1. Diagram
   2. Description
4. **Data Design**

**6.1 Entity Relationship Diagram**

**6.2 ER to Relational Mapping**

1. **Supplementary Information**
   1. Tools used to draw the diagrams

* UML Modeling Tools
* ArgoUML – Version 0.24, http://argouml.tigris.org/
* Entity Relationship Diagramming Tools
* Dia – Version 0.95, http://live.gnome.org/Dia