Defect Programmer Assignments

High Level Design & Low Level Design

The purpose of this document is to provide with a template for documenting both HLD & LLD.

**Document Control :**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Project Revision History** | | | | | | | | |
|  |  |  | |  |  |  |  |  |
| **Date** | **Version** | **Author** | **Brief Description of Changes** | | | | **Approver Signature** | |
|  |  |  |  | | | |  | |
|  |  |  |  | | | |  | |
|  |  |  |  | | | |  | |

[1. Introduction 3](#_Toc368912248)

[1.1. Intended Audience 3](#_Toc368912249)

[1.2. Acronyms/Abbreviations 3](#_Toc368912250)

[1.3. Project Purpose 3](#_Toc368912251)

[1.4. Key Project Objectives 3](#_Toc368912252)

[1.5. Project Scope and Limitation 3](#_Toc368912253)

[1.5.1. In Scope 3](#_Toc368912254)

[1.5.2. Out of scope 3](#_Toc368912255)

[1.6. Functional Overview 3](#_Toc368912256)

[1.7. Assumptions, Dependencies & Constraints 3](#_Toc368912257)

[1.8. Risks 3](#_Toc368912258)

[2. Design Overview 3](#_Toc368912259)

[2.1. Design Objectives 3](#_Toc368912260)

[2.1.1. Recommended Architecture 3](#_Toc368912261)

[2.2. Architectural Strategies 3](#_Toc368912262)

[2.2.1. Design Alternative 3](#_Toc368912263)

[2.2.2. Reuse of Existing Common Services/Utilities 3](#_Toc368912264)

[2.2.3. Creation of New Common Services/Utilities 3](#_Toc368912265)

[2.2.4. User Interface Paradigms 3](#_Toc368912266)

[2.2.5. System Interface Paradigms 3](#_Toc368912267)

[2.2.6. Error Detection / Exceptional Handling 3](#_Toc368912268)

[2.2.7. Memory Management 3](#_Toc368912269)

[2.2.8. Performance 3](#_Toc368912270)

[2.2.9. Security 3](#_Toc368912271)

[2.2.10. Concurrency and Synchronization 3](#_Toc368912272)

[2.2.11. Housekeeping and Maintenance 3](#_Toc368912273)

[3. System Architecture 3](#_Toc368912274)

[3.1. System Architecture Diagram. (Not Necessary) 3](#_Toc368912275)

[3.2. System Use-Cases 3](#_Toc368912276)

[3.3. Subsystem Architecture 3](#_Toc368912277)

[3.4. System Interfaces 3](#_Toc368912278)

[3.4.1. Internal Interfaces 3](#_Toc368912279)

[3.4.2. External Interfaces 3](#_Toc368912280)

[4. Detailed System Design 3](#_Toc368912281)

[4.1. Key Entities 3](#_Toc368912282)

[4.2. Detailed-Level Database Design 3](#_Toc368912283)

[4.2.1. Data Mapping Information 3](#_Toc368912284)

[4.2.2. Data Conversion 3](#_Toc368912285)

[4.3. Archival and retention requirements 3](#_Toc368912286)

[4.4. Disaster and Failure Recovery 3](#_Toc368912287)

[4.5. Business Process workflow 3](#_Toc368912288)

[4.6. Business Process Modeling and Management (as applicable) 3](#_Toc368912289)

[4.7. Business Logic 3](#_Toc368912290)

[4.8. Variables 3](#_Toc368912291)

[4.9. Activity / Class Diagrams (as applicable) 3](#_Toc368912292)

[4.10. Data Migration 3](#_Toc368912293)

[4.10.1. Architectural Representation 3](#_Toc368912294)

[4.10.2. Architectural Goals and Constraints 3](#_Toc368912295)

[4.10.3. Logical View 3](#_Toc368912296)

[4.10.4. Architecturally Significant Design Packages 3](#_Toc368912297)

[4.10.5. Data model 3](#_Toc368912298)

[4.10.6. Deployment View 3](#_Toc368912299)

[5. Environment Description 3](#_Toc368912300)

[5.1. Time Zone Support 3](#_Toc368912301)

[5.2. Language Support 3](#_Toc368912302)

[5.3. User Desktop Requirements 3](#_Toc368912303)

[5.4. Server-Side Requirements 3](#_Toc368912304)

[5.4.1. Deployment Considerations 3](#_Toc368912305)

[5.4.2. Application Server Disk Space 3](#_Toc368912306)

[5.4.3. Database Server Disk Space 3](#_Toc368912307)

[5.4.4. Integration Requirements 3](#_Toc368912308)

[5.4.5. Jobs 3](#_Toc368912309)

[5.4.6. Network 3](#_Toc368912310)

[5.4.7. Others 3](#_Toc368912311)

[5.5. Configuration 3](#_Toc368912312)

[5.5.1. Operating System 3](#_Toc368912313)

[5.5.2. Database 3](#_Toc368912314)

[5.5.3. Network 3](#_Toc368912315)

[5.5.4. Desktop 3](#_Toc368912316)

[6. References 3](#_Toc368912317)

[7. Appendix 3](#_Toc368912318)

# 

# Introduction

The introduction of the software requirement specifications provides an overview

of the entire Software. The aim of this document is to gather, analyze and give an

in-dept insight into the Defect Programmer Assignment by defining the problem

statement in detail.

## Intended Audience

This document is intended to be read by the client

|  |  |
| --- | --- |
| CPP Unit |  |
| Valgrind |  |

## Acronyms/Abbreviations

|  |  |
| --- | --- |
| UT | Unit Test |
| IT | Integrated Test |
|  |  |
|  |  |

## Project Purpose

Clients of software development company report defects in software they

Purchased. These defects are assigned to programmers depending on the

functional area they are handling.

## Project Scope and Limitation

### In Scope

It can take multiple files with defects as input and check the input for

validity and assigns the employee to the valid defects.

Invalid defects are copied to a separate file and unassigned defects are

copied to a separate file.

### Out of scope

It doesn’t reassign the employees after they have been freed as their work.

It doesn't correct the invalid defects in the input file.

It doesn’t check for invalid employees

It doesn’t check for duplicates of input

It doesn’t check for incorrect categories for each parameter in the input file.

## Functional Overview

## Assumptions, Dependencies & Constraints

Operating systems: It is made for windows 7 or above

End-user characteristics: It is assumed that they are not invalid employees,

they are not duplicates of input, there are no incorrect categories for each

parameter in the input file.

# Design Overview

## Design Objectives

### Recommended Architecture

## Architectural Strategies

### Design Alternative

### Reuse of Existing Common Services/Utilities

### Creation of New Common Services/Utility

### User Interface Paradigms

### System Interface Paradigms

### Error Detection / Exceptional Handling

## System Use-Cases

A system use casedisplays the relationships between consumers and

providers of application services.

## Subsystem Architecture

The Model Subsystem has the responsibility of handling the data in the

system. The Subsystems can contain other Subsystems or they can be split

into separate Subsystems

## System Interfaces

The logical characteristics of each interface between the software product and the hardware components of the system ' 'the logical characteristics of each interface between the software product and its users '

### Internal Interfaces

## Disaster and Failure Recovery

# Environment Description

The system environment is primarily the set of variables that define or control

certain aspects of process execution. Windows uses environment variables to

store valuable information about system processes, resource usage, file path,

and more.

## Time Zone Support

The time functions access and reformat the current system date and time. You

do not need to specify any special flag to the compiler to use the time

functions. Include the header file for these functions in the program.

## Language Support

English is only supported language.

## User Desktop Requirements

**Operating systems**: Windows 10/11

**CPU**: 1vCPU with 1 GHz or faster processor

**RAM**: 1024 MB

**Hard drive**: 100 MB or more

**putty**: Release 0.77

## Server-Side Requirements

server-side code has full access to the server operating system and the

developer can choose what programming language (and specific version) they

wish to use. Developers typically write their code using web frameworks.

### Deployment Considerations

There are new deployment options in Windows 10 that help you simplify the

deployment process and automate migration of existing settings and

applications.

### Application Server Disk Space

Windows Server does not allow ATA/PATA/IDE/EIDE for boot, page, or data

drives. The following are the estimated minimum disk space requirements

for the system partition. Minimum: 32 GB

### Database Server Disk Space

Database server (Windows) :- Windows Server 2012 R2,2016,2019

Database server( ) :-

### Integration Requirements

System integration involves both the integration of components and the

integration of functions. High-level functions can be obtained by integrating

components into a comprehensive working system instead of individual

disconnected subsystems

### Jobs

### Network

Adequate bandwidth is required to provide a responsive experience.

Bandwidth requirements vary based on the volume at a client site.

Bandwidth is not the only determining factor for application

responsiveness.

### Others

## Configuration

Only configurations that directly apply for use with Configuration Manager

are included here.

### Operating System

operating system provides its own version of system-management support

in order to promote ease of use and to improve security and integrity.

### Database

Database configuration is to set up the connection details required to

connect to your database from Visual Paradigm. Database configuration is

required because: Select Tools > DB > Database Configuration… from the

toolbar. At the top left corner of the Database Configuration window, set

the (Programming) Language of your project.

### Network

Network configuration is the process of assigning network settings,

policies, flows, and controls. In a virtual network, it’s easier to make

network configuration changes because physical network devices

appliances are replaced by software, removing the need for extensive

manual configuration.

### Desktop

Configuration represents the way in which a system is set up, related to

the assortment of components that make up the system, it can refer to

hardware, software, or a combination of both.

**Change Log**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **QMS Template Version Control (Maintained by QA)** | | | | | |
|  |  |  |  |  |  |
| **Date** | **Version** | **Author** | | **Description** | |
| 28-May-2015 | 1.0 | QA Team | | Initial Version | |
|  |  |  | |  | |
|  |  |  | |  | |
|  |  |  | |  | |