Software

Development

&

Documentation Guide

James C. Vernon, St. Petersburg College – Clearwater Campus

For SPC Programming Students

Use this guide as a reference to develop software programs, systems, and libraries. This guide is intended to be abstract enough so that a few key points can get you to a solution.

## Use this guide by breaking up your problem into three key phases:

* Analysis: First what are you trying to accomplish? Are you printing Hello World to the terminal or are you creating an extensive library that requires man classes and class libraries? Analysis leads to defining the problem on paper.
* Documentation: I also called this whiteboarding here. Integral to going from analysis to documentation is scribbling on a pad, or whiteboarding, or brainstorming. The point is that in documentation you start to further refine for diagraming.
* Diagraming: In this step you will use a mix of traditional diagrams in software engineering. Example diagram types are UML, EERD, Data Model, View Model, State Model, Network Diagram.

### Two example programs are given

1. **Hello World Program – First Program most coders write just prints Hello World to standard out**
2. **Employee Management System – Extensive software application that handles large amounts of data regarding employees, jobs, pay, and benefits**

# Overview

## Description

The following are written descriptions of what the program or software actually does. Hello World is enough to describe whereas Employee Management System will require much more. So much that for an example it’s too much to write out.

Hello world is a program which is compiled and executed in a native environment and simply prints the string Hello World to the standard output of the machine in which it is executed on.

The Employee Management System of NO-NAME Company is a robust, fully integrated, highly available, distributed system that persists data through well established RDBMS and is fully integrated using terminal applications, web & mobile applications, native desktop applications, embedded applications. Employee Management System serves as the core record management system for Human Resource applications concerning… a bunch more jargon that has nice selling points.

# Analysis

Analysis is the brainstorming session. Analysis is This is the main problem said simply. Here is where you scope your domain. Notice that it might be already stated. Here is one simple one, and one a tad more complex.

|  |  |  |
| --- | --- | --- |
| Break down: | | **Use this space for your own notes** |
| Problem Statement(s) | * The Hello World Program prints Hello World to Standard Out. * The Employee Management System manages employees at the NO-NAME Company in mutli domains |  |
| Domain Analysis/User Scope | * The domain for the Hello World Program is environment in which it is executed. Standard Out. * The Domain(s) for the Employee Management System at the NO-NAME company  1. **Personnel Info**    1. Name    2. Address 2. **Role & Task Management**     1. Job Description    2. Job Role    3. Scheduling 3. **Pay & Tax Information**    1. Tax Service    2. Ret Plans 4. **Medical Insurance**    1. Providers    2. Plans 5. Etc.… | Where you are trying to constrain information.  Specify domains |
| Must Do | * Hello World must print to Standard Out * Employee Management System **must**    + Manage All Employee Information   + Comply with State and Federal Laws   + Publish an Archive Backup periodically   + Etc.… |  |
| Must Not Do | * Hello World is not constrained but should not do anything other than print a line of text to standard out * Employee Management System **must not:**   + Duplicate Individual personnel records   + Have more than one medical Insurance at a time   + Duplicate any payment transactions   + Allow access to privileged resources from unauthorized users   + Etc… |  |

# (Whiteboarding)Documentation

|  |  |  |  |
| --- | --- | --- | --- |
| Break down: | | | Use this space for your own notes |
| **Entities** | |  |  |
|  | **Interfaces** | * Hello World – None * Employee management System…   + ITransaction, IPerson, IJob, ISchedule, IManager, …   + Interfaces are defined as entities are identified and created during analysis |  |
| **Abstract Classes** | * Hello World – None * Employee Management System * Person, Job, Calendar, TaxTransaction, etc… * Abstract classes provide a class reference to handle concrete classes |  |
| **Concrete Classes** | * HelloWorld – Program.cs * Employee Management System * People   + Manager   + Lead   + Tech   + Sales * Payment Manager   + PaymentTransaction   + MedicalBenefitRequest   + MedicalBenefitRespone * Etc…. |  |
| **Data Storage** | | * Hello World – None * Employee Management System – Extensive, highly available, reliant, disaster recovery enabled sharded multi node database(s) Postgres instances |  |
| **External Dependencies** | | * Hello world – None * Employee Management Software   + Object Relational Mapping Software   + Medical Data Ingest   + Financial Data Ingest   + Etc..... |  |
| **Program State** | | * Hello World – None * Employee Management System – Too many to list requires a user interaction storyboard! |  |
| **Testing** | | Hello World – None works or doesn’t  Employee Management System – Testing should be embedded in each library |  |

# Diagraming

The Hello World Program Does not require diagraming. The Employee Management system will require an extensive amount of diagraming though. Only a few excerpts need to be shared here though.

A screenshot of a diagram

AI-generated content may be incorrect.