
Software Design Specifications

for

EMPTrack-[Employee Management System]

Prepared by:

Jaswanth Sandra	SE22UARI060	se22uari060@mahindrauniversity.edu.in
Sudeep Reddy	SE22UARI168	se22uari168@mahindrauniversity.edu.in
K Hariprasad	SE22UARI062	se22uari062@mahindrauniversity.edu.in
M Charan	SE22UARI033	se22uari033@mahindrauniversity.edu.in
M Yashwanth	SE22UARI187	se22uari187@mahindrauniversity.edu.in
K Goutham	SE22UARI069	se22uari069@mahindrauniversity.edu.in

Document Information

Title:			
Project Manager:		Document Version No:	
		Document Version Date:	
Prepared By:		Preparation Date:	

Version History

Ver. No.	Ver. Date	Revised By	Description	Filename

Table of Contents

1 INTRODUCTION	4
1.1 PURPOSE	4
1.2 SCOPE	4
1.3 DEFINITIONS, ACRONYMS, AND ABBREVIATIONS	4
1.4 REFERENCES	4
2 USE CASE VIEW	4
2.1 USE CASE	4
3 DESIGN OVERVIEW	4
3.1 DESIGN GOALS AND CONSTRAINTS	5
3.2 DESIGN ASSUMPTIONS	5
3.3 SIGNIFICANT DESIGN PACKAGES	5
3.4 DEPENDENT EXTERNAL INTERFACES	5
3.5 IMPLEMENTED APPLICATION EXTERNAL INTERFACES	5
4 LOGICAL VIEW	5
4.1 DESIGN MODEL	6
4.2 USE CASE REALIZATION	6
5 DATA VIEW	6
5.1 DOMAIN MODEL	6
5.2 DATA MODEL (PERSISTENT DATA VIEW).....	6
5.2.1 <i>Data Dictionary</i>	6
6 EXCEPTION HANDLING	6
7 CONFIGURABLE PARAMETERS	6
8 QUALITY OF SERVICE	7
8.1 AVAILABILITY.....	7
8.2 SECURITY AND AUTHORIZATION	7
8.3 LOAD AND PERFORMANCE IMPLICATIONS	7
8.4 MONITORING AND CONTROL	7

1 Introduction

[The introduction of the Software Design Specifications should provide an overview of the entire Software Design. It should include the purpose, scope, definitions, acronyms, abbreviations, references, and overview of the Software Design Specifications.]

1.1 Purpose

*[This section defines the purpose of the **Software Design Specifications**, in the overall project documentation, and briefly describes the structure of the document. The specific audiences for the document should be identified, with an indication of how they are expected to use the document.]*

1.2 Scope

*[A brief description of what the **Software Design Specifications** applies to; what is affected or influenced by this document.]*

1.3 Definitions, Acronyms, and Abbreviations

*[This subsection should provide the definitions of all terms, acronyms, and abbreviations required to properly interpret the **Software Design Specifications**. This information may be provided by reference to the project Glossary.]*

1.4 References

*[This subsection should provide a complete list of all documents referenced elsewhere in the **Software Design Specifications**. Each document should be identified via an URL link to the source document library if available. If not, specify the sources from which the references can be obtained. This information may be provided by reference to an appendix or to another document.]*

2 Use Case View

[Based on the requirements document, this section lists use cases or scenarios from the use-case model if they represent some significant, central functionality of the final system, or if they have a large design coverage - they exercise many design elements, or if they stress or illustrate a specific, delicate point of the design. Provide a use case diagram for use cases that pertains to the software design]

2.1 Use Case

[If the functional requirements document does not contain sufficient information for a given use case, for each use case pertaining to the design, include a subsection with its name, its brief description, and usage steps or procedures for the particular use case.]

3 Design Overview

[This section provides an overview of the entire software design. It references and complies with the high-level design interface contracts, requirements and high-level module decomposition approach.]

3.1 Design Goals and Constraints

[This section describes the software requirement constraints and objectives that have some significant impact on the design. It also captures the special constraints that may apply: design and implementation strategy, development tools, team structure, schedule, legacy code, and so on.]

3.2 Design Assumptions

[This section describes assumptions that have significant impacts on the software design.]

3.3 Significant Design Packages

[This subsection describes the overall decomposition of the design model in terms of its package dependencies, hierarchy and layers]

3.4 Dependent External Interfaces

[The high level design identified external interfaces that are depended upon by the application. This section indicates where these interfaces are used at the internal module level.]

The table below lists the public interfaces this design requires from other modules or applications.

External Application and Interface Name	Module Using the Interface	Functionality/Description
<Application owning the interface and the interface name>	<Expected Module which uses the interface>	<Describe how the interface will be used to implement required business transactions>

3.5 Implemented Application External Interfaces (and SOA web services)

[The high level design identified interfaces that are provided or owned by the application. This section indicates which internal modules are responsible for implementing these interfaces.]

The table below lists the implementation of public interfaces this design makes available for other applications.

Interface Name	Module Implementing the Interface	Functionality/Description
<Interface name>	<Expected Module which implements the interface>	<Describe how the interface is implemented>

4 Logical View

[This section describes the detailed design. The description should consist of a series of layers. The top most layers should show how the application modules interact to complete the key use cases from Section 2. The more detailed layers should expand each module's separate interaction responsibility and show how the separate classes within the

module collaborate to implement the required behaviour. Depending on project needs, this decomposition may continue until each method in the layer can be described in a page or less of pseudo-code or text.]

4.1 Design Model

[This section describes the software model from class design perspective. It decomposes the software into modules and then the significant classes that make up each module. The model should depict significant classes and class utilities that make up the system through one or more class diagrams. For each significant class, describe its responsibilities, as well as important relationships, operations, and attributes.]

4.2 Use Case Realization

[For each use case defined in section 2, provide detailed descriptions of how that use case scenario is realized within the design. At the highest level, show the interactions between modules using sequence or activity diagrams that implement each key use case (i.e. the business transaction implementation). For each interaction in the highest level, expand to a lower level sequence or activity diagram that shows how each class within the module collaborates to implement the required behaviour.]

5 Data View

[This section describes the persistent data storage perspective of the system. This section is may not apply if there is little or no persistent data, or the translation between the Design Model and the Domain Model is trivial]

5.1 Domain Model

[Provide the entity model that represents the persistent data and relationships between them. The entities in this model are typically domain objects or data transfer objects.]

5.2 Data Model (persistent data view)

5.2.1 Data Dictionary

6 Exception Handling

[This section should describe exceptions that are defined within the application, the circumstances in which they can be thrown and handled, how the exceptions are logged, and the expected follow-up action needed.]

7 Configurable Parameters

[This section may be more applicable to J2EE-based systems than Ruby on Rails applications. It should describe parameters used by the application that are configurable. For simple configuration, the name and definition/usage of each parameter should be provided. For complex configuration, the XML schema associated with the configuration should be provided. For each configurable parameter, indication should be made as to which parameters are dynamic (can be changed without restarting the application).]

This table describes the simple configurable parameters (name / value pairs).

Configuration Parameter Name	Definition and Usage	Dynamic?
<Parameter name>	<Definition and Usage of Parameter>	<Yes/No>

--	--	--

8 Quality of Service

[This section describes aspects of the design related to application availability, security, performance, and monitoring and control in production.]

8.1 Availability

[This section should reference the availability business requirement for the application. It should highlight any aspects of the design specifically intended to support availability at the required level. It should describe any aspects of the design that can impact availability (mass update, data loading, housekeeping or periodic maintenance that requires downtime).]

8.2 Security and Authorization

[This section should reference the business requirements related to feature and data access security and authorization, and describe the authorization features and qualifier design intended to implement the requirements. This section should also describe any application specific design related to authorization outside of the Authorization framework, and how setup and management of the user access is exposed to those responsible for user administration.]

8.3 Load and Performance Implications

[In the system requirements document for the application, load projections and performance requirements are provided. This section should describe the implications of load and performance requirements on the detailed design components (expected business transaction execution rate, message processing rate, database table growth projections, etc.). Sufficient detail should be provided to support the creation of load and performance test plans.]

8.4 Monitoring and Control

[This section should describe controllable processes implemented by the application (message handlers, daemons). This section should also describe measurable values that the application will publish for monitoring from these processes.]