Team T1

Josh Eads, John Loranger,

Ryan Reams, Chris Stultz

April 14, 2015

CIS 234A

Assignment 2: Standards

## This document specifies the standards for the T1 group regarding Documentation and coding. It is broken down into the following sections:

## Requirements Documentation

## Architectural/Design Documentation

## Technical Documentation

## Testing Standards

## Production Standards

## Appendix A Java Standard Example

## Appendix B Database Standard Example

## Requirements Documentation. This is documentation capturing application requirements.

## Gerkin Language stories with list of questions/answers.

## GUI Pencil diagrams.

## Test cases in Word table format.

## Instructor supplied program documents.

## Instructor supplied Assignments.

## Architectural/Design Documentation. This is documentation about the code to be written. It shall be diagram heavy and text light except for appendix and table based information.

## Object modeling: Word with embedded Visio UML Class and object interaction diagrams.

## Application Overview Architecture: Word with embedded Visio diagrams showing overall and individual diagrams of:

## Presentation Layer (PL) User Views of data such as GUI screens.

## Services Layer (SL). Business functionality API for PL and external.

## Business Layer (BL). Business objects supporting SL.

## Data Layer (DL). Data access objects to support BL.

## Database Design: Word with embedded ERD diagrams showing

## ERD with Entity Sets, Relatinos, Candidate Keys, attributes.

## Relational Schema

## Tables with attibutes/properties including keys & fkeys.

## Views on tables.

## Table Constraints showing:

## Primary/foreign keys.

## Referential Cardinality.

## Column data field range/edit.

## Stored Procedures. Initial design shall have list of candidate stored procedures with specifics growing during development and documented in Technical Documentation.

## Operational.

## Development environment physical/logical database diagram.

## Testing environment diagram.

## Production environment diagram.

## Technical Documentation. This describes our standardized practices for: development, testing, implementation, and ongoing maintenance. It is supplemented with design documents from Requirements Documentation.

## Development Code Standards.

## Git Comments. Use a –m comment for each commit.

## Java Source File and JUnit Comments.

## See Appendix A Java Coding and Documentation Standards.

## SQL Source File and SQL Test Script Documentation.

## See Appendix B SQL Server Coding and Documentation Standards.

## Development API Documentation for each layer of architecture.

## Overview and basic functionality of PL,SL,BL and DL layers.

## Overview of database api and organization.

## Development Environment

## GIT revision control. Git will contain all project documentation and code. It shall be accessed via Eclipse for coding activities.

## Java SE version 8 for PL, SL, BL, DL.

## Eclipse on Windows or Mac for Java development.

## Microsoft SQL server for Database.

## JUnit/Hamcrest for automated Java testing.

## Manual GUI testing on Mac and Windows.

## Scripts for SQL Database testing.

## Maven for build management.

## Development Database standards

## Architecture. See/use/upgrade database architecture documents from Requirements Documentation.

## Package/Stored Procedure API/Documentation. A design goal is to maximize database access via stored procedures rather than raw SQL embedded in Java. Initially in development we will use raw SQL within the Java Data Layer (DL) , but later, as we refactor, we will move raw SQL out of Java Data Layer and into database hosted stored procedures. The Java Data Layer will call these. This set of Stored Procedures shall form the database API and be documented as:

## Description of Stored Procedure API as whole.

## General overview of each stored procedures

## Procedure name.

## Functional description.

## Parameters.

## Returned data.

## Effects on database and stored procedure runtime environment.

## Exception processing.

## Exception names/error codes thrown by API.

## Database console error messages .

## Procedure package/physical file organization.

## Documentation of individual SQL creation/maintenance scripts. See Appendix.

## Testing Standards

## Manual testing of database and GUI are stored as a list of Word files in GIT.

## Java Layers via Eclipse/JUnit/Maven automation.

## Database testing is via scripts and manual user actuation. May try to get Maven to automate.

## Test Environment shall be using development Database server with segregated test area.

## Production Standards

## Production Environment shall be using development Database server with segregated Production area.

## Installation documentation including platform support and file locations

## Runtime information for operators including:

## Console message meaning / action

## Availability Characteristics. Hours of operation etc.

## Security needs/conventions.

## Performance limits. Max Txns/hour, Max users etc.

## CPU / Memory / Network load characteristics.

## Java Layer

## Instructions indicating which version/URI of Java to install.

## Java environment configuration. File locations, verfications, environment variables.

## Extra libraries. Libraries not included in standard Java Runtime.

## Database Layer

## Production network diagram to be created w/production & operations group

## Production server hosting information.

## Documentation for operations group

## Table showing info/warn/operational db console messages and operational action tree.

## Security section showing Security layers for different levels of OSI model.

## Securing of database ip, port with operations group.

## Access policies for admins.

## Access policies for users.

## Security breach operational plan.

## Backup and recovery plans including.

## Disaster scenarios.

## Daily backup.

## Appendix A Java Coding and Documentation Standards. Java’s javadoc application will be used to generate all class level documentation for publishing to a group shared location.

## Coding and documentation Standards. Coding standards shall follow the guidelines set down in Google Java Style standards shown at this URL: https://google-styleguide.googlecode.com/svn/trunk/javaguide.html#s4.3-one-statement-per-line

## Java Coding Variable name general recommendations:

## Follow Google Java Style standards shown at this URL: https://google-styleguide.googlecode.com/svn/trunk/javaguide.html#s4.3-one-statement-per-line

## A few specific recommendations

## Use long descriptive names indicating the business purpose of the variable.

## Example: List<Accounts> listOfAccounts

## When passing in parameters for constructors make the argument name the same as the property name so you can use “this” as the sole differentiator:

## Example: void setLastName( String lastName) { this.lastName = lastName;}

## Index variables for loops should be I,j,k with I on the outermost loop, j the middle, and k the inside.

## Function Names. Function names should be long and descriptive.

## Exception names. Exceptions should have long descriptive names paralleling the exception tree from which it comes. Whenever possible existing Exception names from the Java library should be used.

## In general keep function short ~ 1-10 lines with singular purpose and highly descriptive names. They should describe, in almost a spoken tone, of what the function does and in some sense serves as both documentation and logical problem breakdown. This aids in reading the call stack when a function throws an exception.

## Example:

## boolean isCustomerMad() { return this.getCustomerFrustration() > MAX\_FRUSTRATION; }

## Java Class Example:

Figure Java Class Example Page 1/2

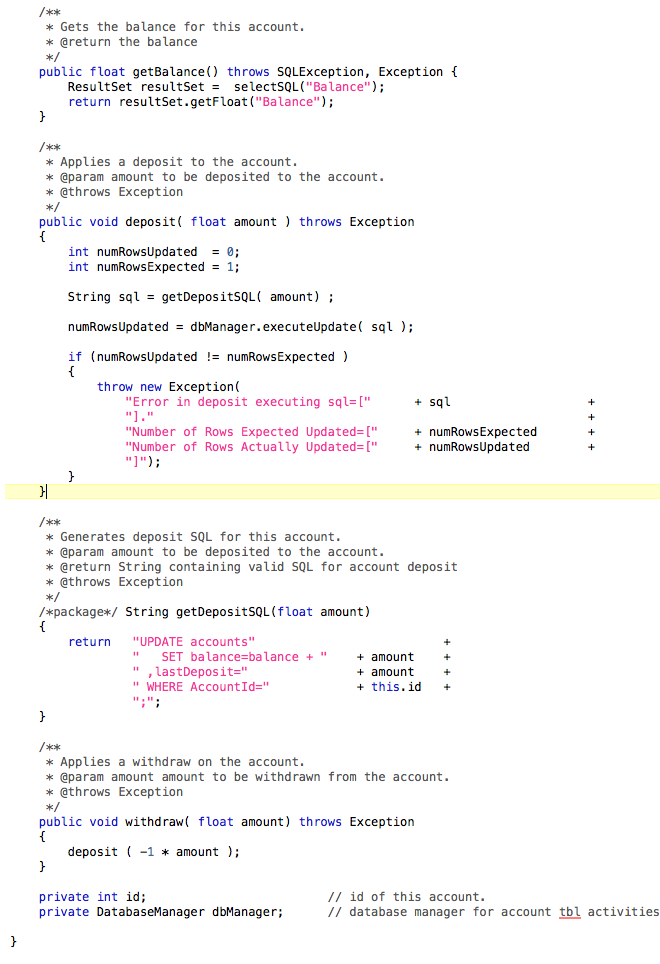


Figure Java Class Example Page 2/2

## Appendix B SQL Server Coding and Documentation Standards.

## Each source code file shall begin with a header of the example form below.

## Singular names for tables

## Singular names for columns

## Schema name for tables prefix i.e. SchemaName.TableName

## Lowercase with underscores separating except table names.

## All SQL keywords uppercase.

## Example package (using Oracle) (specification and implementation)

## Store Proc Specification Example J13HD:Users:john_loranger:Desktop:DatabaseSpecification.png

## Stored Procedure Implementation ExampleJ13HD:Users:john_loranger:Desktop:Screen Shot 2015-04-14 at 8.00.35 PM.png

Figure 3. Header for Stored Proc Implementation

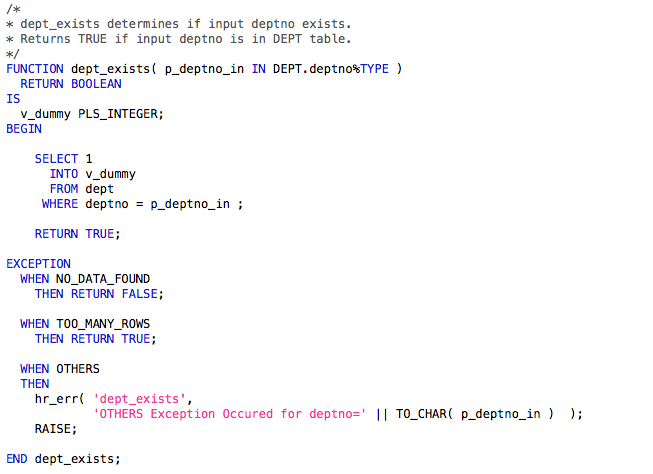


Figure Stored Proc Boolean function example

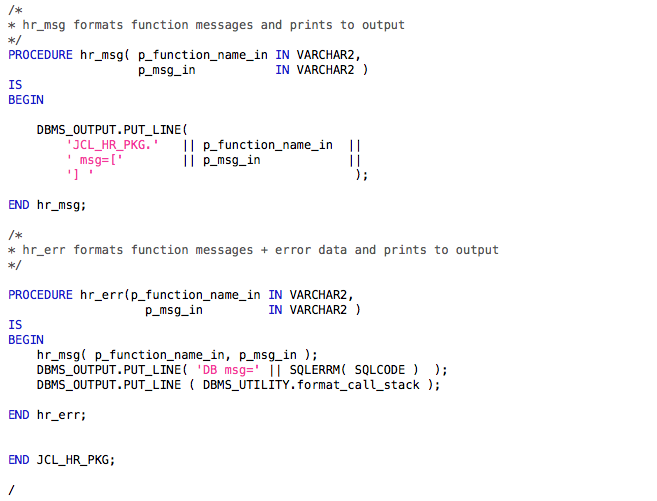


Figure Stored Proc Error Message formatting

## 

## 