[TOPCODER]
SOFTWARE

Expense Entry 1.1 Requirements Specification

1. Scope

1.1 Overview

The Expense Entry custom component is part of the Time Tracker application. It provides an abstraction of an expense entry that an employee enters into the system on a regular basis. This component handles the persistence and other business logic required by the application.

Version 1.1 will enhance the existing version 1.0 design and maintain the same underlying API. New classes and methods will be added to meet the additional requirements.

Note: the name of the package has changed.

1.2 Logic Requirements

1.2.1 Batch Operations

The new design will provide batch versions of the CRUD (Create/Read/Update/Delete) operations of the persistence layer. This means they will accept an array rather than single instances. By caller's choice, the batch operations can be made atomic (all-or-nothing).

1.2.2 Reject Reason

Expense entries of a contractor may be rejected by a project manager. The manager will select one or more reasons from a drop down list of enumerations. The reason(s) will be given as ID's and linked to the expense entry. The status of the expense entry will be changed to REJECTED.

The reject reasons will be maintained externally and stored in the following table:

```
CREATE TABLE reject_reason (
       reject_reason_id          integer NOT NULL,
       description          varchar(256) NOT NULL,
       creation_date          datetime year to second NOT NULL,
       creation_user          varchar(64) NOT NULL,
       modification_date      datetime year to second NOT NULL,
       modification_user      varchar(64) NOT NULL,
       PRIMARY KEY (reject_reason_id)
);
```

The link between expense entries and reject reasons will be stored:

```
CREATE TABLE exp_reject_reason (
       ExpenseEntriesID          integer NOT NULL,
       reject_reason_id          integer NOT NULL,
       creation_date          datetime year to second NOT NULL,
       creation_user          varchar(64) NOT NULL,
       modification_date      datetime year to second NOT NULL,
       modification_user      varchar(64) NOT NULL,
       PRIMARY KEY (ExpenseEntriesID, reject_reason_id)
);
```

1.2.3 Expense Entry Search

This functionality adds the ability to search for all expense entries based on some criteria. The search criteria can be a combination of any of the following search filters.

1.2.3.1 Search Based on Description

This search will return all expense entries with a description that contains a given string. The specified sub-string can appear anywhere in the expense entry description.

1.2.3.2 Search Based on Expense Status

This search will return all expense entries with the specified expense status ID.

1.2.3.3 Search Based on Expense Types

This search will return all expense entries with the specified expense types ID.

1.2.3.4 Search Based on Users

- Return all expense entries created by the specified user name
- Return all expense entries modified by the specified user name

1.2.3.5 Search Based on Billable Flag

- Return all expense entries with Billable flag set to true
- Return all expense entries with Billable flag set to false

1.2.3.6 Search Based on Reject Reason

This search will return all expense entries with the specified reject reason ID.

1.2.3.7 Search within an Amount Range

The amount range is given as a pair of Min Amount and Max Amount. The range can be open-ended. This means it has the following modes of operations:

- Return all expense entries with amount greater than or equal to Min Amount
- Return all expense entries with amount less than or equal to Max Amount
- Return all expense entries with amount between the Min Amount and Max Amount, inclusive

1.2.3.8 Search within a Date Range

The date range is given as a pair of Begin Date and End Date, for either creation date or modification date. The range can be open-ended. This means it has the following modes of operations:

- Return all expense entries created on or after the Begin Date
- Return all expense entries modified on or after the Begin Date
- Return all expense entries created on or before the End Date
- Return all expense entries modified on or before the End Date
- Return all expense entries created between the Begin Date and End Date, inclusive
- Return all expense entries modified between the Begin Date and End Date, inclusive

1.3 Required Algorithms

None.

1.4 Example of the Software Usage

The Time Tracker application will use this component to perform operations related to expense entries.

1.5 Future Component Direction

Other database systems maybe plugged in for some client environments.

2. Interface Requirements

2.1.1 Graphical User Interface Requirement

None.

2.1.2 External Interfaces

None.

2.1.3 Environment Requirements

- Development language: Java 1.4
- Compile target: Java 1.3, Java 1.4

2.1.4 Package Structure

com.cronos.timetracker.entry.expense

3. Software Requirements

3.1 Administration Requirements

3.1.1 What elements of the application need to be configurable?

None.

3.2 Technical Constraints

3.2.1 Are there particular frameworks or standards that are required?

None.

3.2.2 TopCoder Software Component Dependencies:

- Configuration Manager
- DB Connection Factory

**Please review the TopCoder Software component catalog for existing components that can be used in the design.

3.2.3 Third Party Component, Library, or Product Dependencies:

Informix Database.

3.2.4 QA Environment:

- Solaris 7
- RedHat Linux 7.1
- Windows 2000
- Windows Server 2003
- Informix

3.3 Design Constraints

The component design and development solutions must adhere to the guidelines as outlined in the TopCoder Software Component Guidelines. Modifications to these guidelines for this component should be detailed below.

3.4 Required Documentation

3.4.1 Design Documentation

- Use-Case Diagram
- Class Diagram
- Sequence Diagram
- Component Specification

3.4.2 Help / User Documentation

- Design documents must clearly define intended component usage in the ‘Documentation’ tab of Poseidon.