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.

1.2 Logic Requirements

1.2.1 Expense Entries

An expense entry represents the date and amount of money an employee has spent for a particular project and client. It is normally used for payroll and billing purposes.

The following attributes will be maintained:

- Description – a brief description of the expense entry
- Date – the date for the expense entry
- Amount – the amount of money the employee spent (cannot be negative)
- Expense Type – the type of expense
- Status – the status of the expense entry
- Billable – a flag to indicate whether the entry is billable to client

The database schema to store time entries is as follows:

```
CREATE TABLE ExpenseEntries (
    ExpenseEntriesID      integer NOT NULL,
    ExpenseTypesID        integer NOT NULL,
    ExpenseStatusesID     integer NOT NULL,
    Description           varchar(64) NOT NULL,
    Date                  date NOT NULL,
    Amount                money NOT NULL,
    Billable              smallint NOT NULL,
    CreationDate          date NOT NULL,
    CreationUser          varchar(64) NOT NULL,
    ModificationDate      date NOT NULL,
    ModificationUser      varchar(64) NOT NULL,
    PRIMARY KEY (ExpenseEntriesID)
);
```

1.2.2 Expense Types

Each expense entry has an associated type of expense made by the employee.

The following expense types have been identified:

- Air Transportation
- Rail Transportation
- Car Rental
- Auto Mileage
- Lodging

- Meals
- Entertainment
- Phone

Note: New task types may be added or modified as per application requirements.

The database schema to store expense types is as follows:

```
CREATE TABLE ExpenseTypes (
    ExpenseTypesID      integer NOT NULL,
    Description         varchar(64) NOT NULL,
    CreationDate        date NOT NULL,
    CreationUser        varchar(64) NOT NULL,
    ModificationDate    date NOT NULL,
    ModificationUser    varchar(64) NOT NULL,
    PRIMARY KEY (ExpenseTypesID)
);
```

1.2.3 Expense Entry Statuses

Each expense entry has a status assigned to it. The status will be modified by an administrator or the supervisor of the employee.

The following statuses have been identified:

- Pending Approval – the expense entry is pending approval by an administrator or supervisor
- Approved – the expense entry has been approved by an administrator or supervisor
- Not Approved – the expense entry is not approved by an administrator or supervisor

Note: New statuses may be added or modified as per application requirements.

The database schema to store expense entry statuses is as follows:

```
CREATE TABLE ExpenseStatuses (
    ExpenseStatusesID   integer NOT NULL,
    Description         varchar(20) NOT NULL,
    CreationDate        date NOT NULL,
    CreationUser        varchar(64) NOT NULL,
    ModificationDate    date NOT NULL,
    ModificationUser    varchar(64) NOT NULL,
    PRIMARY KEY (ExpenseStatusesID)
);
```

1.2.4 Pluggable Persistence

All entities defined above will be backed by a database. The design will provide the necessary API to store and retrieve data from the database.

For the initial version, the Informix database system will be used as persistence storage for this component and the Time Tracker application. Other database systems should be pluggable into the framework.

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.topcoder.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.