

Software Requirements Document

for

Community Project Tracking

CS 472

Draft 0.1

Prepared by:
Tim Grediagin
Devin Jones
Brent Mello
Blake Eggemeyer

February 29, 2012

Contents

1	Introduction	2
1.1	Purpose	2
1.2	Scope	2
1.3	Definitions	2
1.4	References	2
1.5	Revision tracking:	3
2	Overall description	3
2.1	Product functions	3
2.2	User Characteristics	3
2.3	Constraints	3
2.4	Assumptions and Dependencies	3
2.5	Apportioning of requirements	3
3	Specific requirements	3
3.1	Functional requirements	4
3.1.1	Store Data	4
3.1.2	User Management	4
3.1.3	Reports	4
3.1.4	Security	4
3.1.5	Backups	5
3.2	Non-functional requirements	5
3.2.1	Interface	5
3.2.2	Platform	5
3.3	Performance requirements	5
3.4	Software system attributes	5

1 Introduction

1.1 Purpose

This Software Requirements Document for Community Project Tracking is for use by the customer and development team. This document is a formal listing of the functional and non-functional requirements of the Community Project Tracking system.

1.2 Scope

The Community Project Tracking software will assist users in the storing and summary of various activity data.

1.3 Definitions

1. **GUI:** Acronym for Graphical User Interface. Used to refer to the look and feel the user experiences.
2. **Immediately:** Immediately refers to actions that will begin as soon as the user has given the input for the action to occur.
3. **Should:** Requirements with this marker are desired, but not crucial, and will be a part of the final deliverable contingent on time and progress.
4. **TBD:** Acronym for To Be Determined. This is used in this document to signify that the information necessary for a part of this document is "To Be Determined".
5. **User:** The person, or persons, who operate or interact directly with the product.
6. **Will:** Requirements with this marker are guaranteed to be in the final delivered product.
7. **CPT:** Internal name of the application and java package.
8. **User:** A person who uses the software directly. All users will have individual login accounts.
9. **Worker:** The person(s) responsible for the work done on a timesheet. A User has a corresponding Worker, but there may be Workers without User accounts, e.g., "boyscouts."
10. **Project:** Seen at top of timesheet, e.g., "Clean Team." Internally, I see no difference between "project" and "program", but we should find out if differentiating them externally is critical. I just went with "project" because "program" has a particular meaning to programmers.
11. **Program:**
12. **Activity:** Left column of timesheet, e.g., "Ice Chipping."

13. **Location:** Top row of timesheet, e.g., "CORE 1st - 3rd." The word "location" isn't actually on the sheet.
14. **Tool/Equipment:** Corresponds to "Equipment Used" on the original timesheet. This is the implement used to complete a Task
15. **Task:** The data entered into a timesheet is stored in Tasks, a record comprising a Project, Worker, Activity, Location, Tool, date, hours worked and a comment.
16. **Comment:** Remarks by a User to be stored with their Task record.
17. **Timesheet:** The name for the web page on which the various data are entered.

1.4 References

Written with the IEEE Recommended Practice for Software Requirements Specifications as a reference and guide. The Tsunami SWR and RPC Donor SWR were referenced to find appropriate wording for some sections.

1.5 Revision tracking:

0.1	Feb 10	Document constructed.
-----	--------	-----------------------

2 Overall description

2.1 Product functions

1. TBD

2.2 User Characteristics

2.3 Constraints

2.4 Assumptions and Dependencies

1. **Language:** The interface for the user is in English.
2. **Platform:**

2.5 Apportioning of requirements

3 Specific requirements

The requirements are listed and ordered in a priority list so that their order can be changed at a later date without the section numbers needing to be changed and to allow listing of the priorities in one location.

Priority List

1	3.1.1	Store data regarding operations
2	3.1.3	Generate summary reports of the data that has been gathered
3	3.1.2	Manage user accounts
4	3.1.4	Prevent unauthorized viewing or modifying of data.
5	3.2.1	Highly usable interface
6	3.1.5	Prevent loss or corruption of data
7	3.2.2	Runs on existing platform

3.1 Functional requirements

3.1.1 Store Data

3.1.1.1 Data regarding a staff members work day will be collected and stored.

3.1.1.2 Administrators will be able to create and edit these work records on behalf of any worker.

3.1.1.3 Users will be able to specify a comment along with each work record.

3.1.1.4 Work records (task) will also have a worker and project specified automatically based on the user.

3.1.2 User Management

3.1.2.1 The application will have varying levels of access for users, with administrators having the most control.

3.1.2.2 Administrators will be able to maintain a list of application users.

3.1.2.3 Administrators will be able to maintain a list of locations.

3.1.2.4 Administrators will be able to maintain a list of tools.

3.1.2.5 Administrators will be able to maintain a list of activities.

3.1.2.6 Administrators will be able to maintain a list of workers.

3.1.2.7 Administrators will be able to maintain a list of projects/programs.

3.1.2.8 Users will be able to create a work record of the work done that day by specifying the amount of hours spent doing each activity at each location with each tool.

3.1.3 Reports

3.1.3.1 The software will use the daily input data to generate reports as described by the client.

3.1.3.2 Administrators will be able generate reports by specifying a time period and project to see the total hours spent for each activity at each location with each tool.

3.1.4 Security

The impact of storing personally identifiable time sheet data must be considered.

3.1.5 Backups

3.1.5.1 The ability to make copies of the data for safekeeping must be considered.

3.2 Non-functional requirements

3.2.1 Interface

3.2.1.1 Determine what User wants from the interface.

3.2.1.2 The application will be able to display the local weather for a given day.

3.2.2 Platform

Several existing platforms are available. One of which is on site at the clients offices running a Windows 2003 server. The others are a few off site servers that are used for a number of different projects.

3.3 Performance requirements

3.4 Software system attributes