Milestone 1

Team Sriram
(Susi Cisneros, Eric Henderson, Taylor Purviance and Richard Thai)

23 September 2011

Contents

1 Executive Summary **3**

2 Introduction **3**

3 Client Background **3**

4 Current System **3**

5 User/Stakeholder Description **4**
5.1 User Profile . . . 4
5.2 Stakeholder Profiles . . . 4
5.2.1 Stakeholder: Tim Ekl . . . 4
5.2.2 Stakeholder: Team Sriram . . . 4
5.2.3 Stakeholder: Sriram Mohan . . . 4
5.3 User Environment . . . 4
5.4 User Needs . . . 4
5.4.1 Primary . . . 4
5.4.2 Secondary . . . 5
5.4.3 Optional . . . 5
5.5 Alternatives and Competition . . . 5

6 Product Overview **5**
6.1 Product Perspective . . . 5
6.2 Elevator Statement . . . 5
6.3 Summary of Capabilities . . . 5
6.4 Assumptions and Dependencies . . . 6
6.5 Rough Cost Estimate . . . 6

7 Non-functional Requirements **6**
7.1 General . . . 6
7.2 Testing . . . 6

8 Features **7**
8.1 Feature Listing . . . 7
8.2 Feature-to-Need Correspondence . . . 7

9 Constraints **8**

10 Index and Glossary **8**

11 References **8**

1 Executive Summary

This milestone, the first of a series, documents the context for a software project proposed by the client: Tim Ekl. The primary stakeholders are Team Sriram (the software developers and documentors), Sriram Mohan (the course instructor), and Tim Ekl. The software solution will eventually be open-sourced which means that external developers and end-users will be involved as stakeholders, however these parties can be ignored for the time being. The current issue raised by the client is the inability to easily and consistently locate personal belongings and hardware. This milestone will cover the background of the client, their current system, involved parties, an overview of the product and costs, features of the product, and constraints to the solution; the solution is intended as a simplified asset-tracker scaled for personal usage. This document will provide utility for the next milestone which will focus on the project background, use cases, feature-to-use-case mapping, and Data Flow Diagrams.

2 Introduction

Our client, Tim Ekl, would like a system to keep track of personal belongings. Tim's profession and interests with computers have caused him to accumulate a large collection of computer hardware. Currently, Tim maintains a crude organization system which involves labeling containers in order to categorize the different areas to place objects–whether it be for storage or extraction. However, the current system is not as effective as the client would like since there have been multiple instances in which items were not found when they were needed and replaced unecessarily, resulting in lost capital. In addition, the current system's generic labeling is not a sure-fire method for finding items; they are only meant to guide Tim and help him make educated guesses concerning the general location of his items. This results in lost time looking for items which may not even exist.

The client wishes to replace the current system with one that should be comparable to corporate asset-tracking systems with the capability to tag assets with a wide variety of attributes, though more streamlined since it is intended for use on a personal scale. Since this project is an initiative from a college program, there will be no monetary capital invested towards development. However, it is expected that each developer involved in the project will invest aproximately five hours weekly towards the project; this results in roughly 340 man hours in terms of a time budget for the project. In addition, all developers will respect the client's preference to construct the product utilizing Ruby (programming language) and Sinatra (web framework). There exist more requirements and constraints which will be elaborated on later in this document. Ultimately the product will be a web application backed by a database.

3 Client Background

Tim Ekl is a Rose-Hulman graduate student who possesses a significant amount of computer hardware. He plans on using this system to be able to quickly and easily locate the equipment he wants to use. Tim is an experienced developer and plans on maintaining the system after it is finished.

4 Current System

The client does not have a software solution in place. Currently, Tim has a primitive categorization system in place which involves labeling boxes and then trying to deduce the location of a desired component. The current system poses a few issues such as not always allowing him to find his items, i.e. there have been instances where an item was found after capital was spent to replace it.

5 User/Stakeholder Description

5.1 User Profile

The client will be the main user of the system. While the client is very familiar with sophisticated technologies and software, the final product will be open-source and will be exposed to many more users who will have a broad spectrum of technical proficiency–so the final product will be made with these prospective end users in mind. The client is an accumulator of hardware and disc media, having approximately 1,000 such items which he wants catalogued. Ultimately, identification of the client's items will be based off of a unique identifier, a category, and as many attributes as necessary to describe it.

5.2 Stakeholder Profiles

5.2.1 Stakeholder: Tim Ekl

Role: Client
Success: The final product reasonably satisfies the needs and requirements as defined.
Failure: The final product is not easily usable and cannot be extended without major reworking.

5.2.2 Stakeholder: Team Sriram

Role: Requirements, Specifications, and Development Team
Success: The final product has its requirements defined and prioritized with a satisfactory amount of the high priority features implemented. The client is able to easily extend and work with the source code provided.
Failure: The final product delivered fails to successfully implement the critical features requested by the client.

5.2.3 Stakeholder: Sriram Mohan

Role: Class Instructor
Success: Team Sriram understood the concepts and processes utilized during the course of the project. The client is satisfied with the final product.
Failure: Team Sriram was unable to deduce any concepts or processes discussed during the course. The client is not satisfied with the final product.

5.3 User Environment

- The client uses Chrome whenever possible and prefers that development support Chrome and Firefox browsers.
- The final product ought to operate on a Linux server with standard programming languages, programming frameworks, and Apache. Additional packages can be installed if necessary.

5.4 User Needs

5.4.1 Primary

- (N0) Search for parts based off their attributes.
- (N1) Compatible with a UNIX architecture.
- (N2) Identify items via bar codes.
- (N3) Keep track of the data associated with an asset.

- (N4) Organize search results.

5.4.2 Secondary

- (N5) Insert objects in the system at any point; do not freeze the database.
- (N6) Modify objects; including adding notes to the objects.

5.4.3 Optional

- (N7) Accessible from any computer.
- (N8) View most-recently acquired asset(s).
- (N9) View a summary of inventory data.

5.5 Alternatives and Competition

Tim has researched alternative solutions but has not been able to find one that fits his needs. Alternative systems have been intended for corporate use, are overly complicated, and designed for large scale use. Other issues associated with these products include incompatibility with non-Windows environments, excessive cost, and locally restricted access (not accessible through the Internet). The client has also considered making a system for himself, however he has never had the time to implement it.

6 Product Overview

6.1 Product Perspective

The software will be an independent system to be hosted on the client's servers. The product will replace the current solution by electronically keeping track of asset locations. In addition, the software will also maintain data associated with each asset.

6.2 Elevator Statement

Everyone accumulates items that they currently have no use for, but do not want to dispose of. Given enough time, a person will accumulate more items than they will be able to consciously keep track of. We propose a web application which will allow for simple and streamlined asset tracking. The software will maintain an organizied and systematic inventory which will also keep track of each item's location.

6.3 Summary of Capabilities

Customer Benefit	**Supporting Features**
Expediently track items based off of specific criteria	F1, F4, F5, F6, F8
View statistics of asset inventory	F4, F7
Keep all data in one place	F0, F2
Access data from anywhere with Internet access	F0
Compatible with the client's current user environment	F9
Change data associated with items	F3

6.4 Assumptions and Dependencies

- Availability of network/web hosting.
- Availability of Ruby and Sinatra framework to run the software.
- Operates on the Unix operating system.

6.5 Rough Cost Estimate

Since the solution will be the product of a course project, no monetary capital will be given nor required. Concerning a time budget, 340 man hours will be invested into the final product (20 hours per week given four team members).

7 Non-functional Requirements

7.1 General

- Responsive on the local network.
- Local/client storage on web browsers using SQLite and Sinatra.
- Allow for easy transfer of inventory data; involves a simple copying, tar'ing, moving, and untar'ing of the directory.
- Avoid crashing the application or at least handle the crash gracefully; avoid hard server restarts or corruptions.

7.2 Testing

- Cucumber is the preferred testing framework. RSpec is recommended if Cucumber is not used.
- Manually coded unit tests should be used at minimum if no testing framework is utilized.

8 Features

8.1 Feature Listing

ID	Feature	Priority	Effort	Risk	Stability	Target Release	Assigned To
F0	Web-accesible with an online API	Critical	High	High	Low	1.0	Unassigned
F1	Search for matching attribute (such as a bar code, text description, or unified search); have simple and advanced searches	Critical	High	High	Low	1.0	Unassigned
F2	Add assets to the inventory	Critical	Low	High	Low	1.0	Unassigned
F3	Modify assets in the inventory	Critical	Low	High	Low	1.0	Unassigned
F4	Each category of an asset has a related set of optional attributes	Critical	Low	High	Low	1.0	Unassigned
F5	Use a UPC-A barcode as the unique identifier for each asset	Critical	Low	Medium	Low	1.0	Unassigned
F6	Provide an updated list of recently-added assets	Useful	Medium	Low	Medium	1.5	Unassigned
F7	Generate reports of asset inventory	Useful	High	Low	Medium	2.0	Unassigned
F8	Sort search results based off of barcode, title, and modified/created timestamp	Useful	High	Low	Low	2.0	Unassigned
F9	Deployable on a Linux-based operating system	Critical	Low	High	Low	1.0	Team Sriram

8.2 Feature-to-Need Correspondence

	N0	N1	N2	N3	N4	N5	N6	N7	N8	N9
F0								X		
F1	X		X							
F2				X		X				
F3				X			X			
F4				X		X	X			
F5	X		X	X						
F6									X	
F7										X
F8	X		X		X					
F9		X								

9 Constraints

ID	Source	Constraint	Rationale	Cost
C0	Development	The project must be open source.	The client wants to be able to extend the project's functionality as well as allow third parties to benefit from the software source code.	Low
C1	Time	The project must be completed within five months.	This is the duration of the Junior Project sequence for Team Sriram.	High
C2	Budget	The project must be completed without funding from the client.	This is an academic undertaking and is not for monetary gain.	Low
C3	Software Architecture	The project must utilize REST, not SOAP.	The client strongly prefers not to use SOAP based off of past experience.	Low
C4	Technology	Develop in Ruby utilizing Sinatra for web framework.	The client prefers the development language and framework to be familiar to him for further extension.	Medium

10 Index and Glossary

Assigned to: Developer ultimately responsible for the implementation of a feature (6).

Constraint: A limitation or bound for the development of the project (7).

Cost: How big of a negative impact a constraint will have on the development process (7).

Effort: Expectation of the resources and time consumed for a feature (6).

Feature: A system capability that fulfills a user need (6).

Priority: Description of how essential a feature is to the project (6).

Risk: Probability that a feature will instigate delays in the project (6).

Source: The area that a constraint is tied to (7).

Stability: Likelihood that the understanding of a feature will change (6).

Target Release: Expected release iteration of a testable feature (6).

11 References

(1) Leffingwell, Dean, and Don Widrig. *Managing Software Requirements: a Use Case Approach.* Addison-Wesley, Boston, 2nd Edition, 2003.