# ChronoTimer 1009 Requirements Specification

Document Version 1.1

3/22/2015

## Project Team:

Adam Permann – Developer

Austin Sonderman – Developer

Cooper Patton – Developer

Aaron Kroupa – Developer

Jeremy Frederick Le Veque – Developer

## Document Author(s):

Adam Permann – Developer

## Customer Representative(s):

?

# I. Introduction

This requirements document for the ChronoTimer 1009 serves as an overall reference for the software being constructed, which specifies the functional and non-functional requirements that the software system will provide.

The requirements document specifies the services that the system will provide and the constraints under which the system must operate.

# II. Functional Requirements

All of the ChronoTimer 1009’s functional Requirements are detailed in this section. Priority and Stability are rated on a scale of 1 – 10 where 1 is the highest and 10 is the lowest measure.

1. The ability to turn the system on.
   1. The system shall allow users to turn the system on to an initial state.

Origin: Product Management and initial client requirements.

Priority: 1 Stability: 1

1. The ability to turn the system off.
   1. The system shall allow users the basic function to turn the system off.

Origin: Product Management and initial client requirements.

Priority: 1 Stability: 1

1. The ability to reset the system.
   1. The system shall allow users the ability to reset the system.

Origin: Product Management and initial client requirements.

Priority: 2 Stability: 1

1. The ability to exit the system.
   1. The system shall allow users the ability to fully exit the system.

Origin: Product Management initial client requirements.

Priority: 2 Stability: 1

1. The ability to create an event of type Individual.
   1. The system shall allow users to create an Individual event, which allows individual starts and finishes for one participant.

Origin: Product Management and initial client requirements.

Priority: 3 Stability: 1

1. The ability to create an event of type Group.
   1. The system shall allow users to create a Group event that allows for the timing of multiple participants with one start, multiple times, and individual finishes.

Origin: Product Management and initial client requirements.

Priority: 3 Stability: 3

1. The ability to create an event of type Parallel.
   1. The system shall allow the ability to create an event, which handles individual starts for multiple participants and finishes.

Origin: Product Management and initial client requirements.

Priority: 4 Stability: 9

1. The ability to create an event of type Parallel Group.
   1. The system shall allow the ability to create an event that handles one group start and individual finishes

Origin: Product Management and initial client requirements.

Priority: 4 Stability: 9

1. The ability to enter participants for the events.
   1. The system shall allow the ability for participants to be added to the system for any event.

Origin: Product Management and initial client requirements.

Priority: 2 Stability: 1

1. The ability for a participant to DNF (Did Not Finish) an event.
   1. The system shall allow for participants to have a DNF entered for an event in which they did not finish.

Origin: Product Management and initial client requirements.

Priority: 3 Stability: 1

1. The ability for a participant to have multiple event records stored in the system.
   1. The system shall allow for participants to have many records stored in the timing system for later retrieval.

Origin: Product Management and initial client requirements.

Priority: 4 Stability: 1

1. The ability for event data to be stored in an event log.
   1. The system shall allow for an entire event’s data be stored in an event log in the system for later retrieval and use.

Origin: Product Management and initial client requirements.

Priority: 4 Stability: 1

1. The ability to “cancel” a start for a participant or participants.
   1. The system shall allow a cancel command for a participant who has an invalid start.

Origin: Product Management and initial client requirements.

Priority: 3 Stability: 1

1. The ability for multiple channels to be used for timing purposes for the ChronoTimer.
   1. The system shall allow for multiple channels which trigger a start, finish or DNF

Origin: Product Management and initial client requirements.

Priority: 3 Stability: 5

1. The ability to connect sensors to the channels, which allow triggering of timing events.
   1. The system shall allow for one sensor to be connected to a channel, which will allow for the automatic triggering of events instead of a manual push of the channel button on the hardware.

Origin: Product Management and initial client requirements.

Priority: 5 Stability: 5

1. The ability to export event and participant data to an external device (ie. USB stick).
   1. The system shall allow for users to export recorded event and participant data to be retrieved via an external device hooked up to the ChronoTimer hardware.

Origin: Product Management and initial client requirements.

Priority: 6 Stability: 3

1. The ability to print event data from an event log.
   1. The system shall allow printing of event data from the printer on the hardware from a saved log of events.

Origin: Initial client requirements

Priority: 7 Stability: 1

1. The abi

# III. Nonfunctional Requirements

1. The ChronoTimer software will be written in Java using Eclipse IDE in conjunction with Google’s app engine.

Origin: Product Management

Priority: 1 Stability: 1

1. Upon Initialization with the “ON” command the timer software will be in a default state where all of the channels are disabled the default event is Individual and the run number is 1.

Origin: Client Requirements

Priority: 3 Stability: 1

1. Upon initialization the timer’s system time will be set to the current internal clock, which, is linked to a global clock.

Origin: Client Requirements

Priority: 1 Stability: 1

1. Channels must be configurable to allow for these types of Sensors to be connected to them for event triggering: Eye, Gate, Pad and a manual button on the console.

Origin: Client Requirements

Priority: 4 Stability: 6

1. Participant Id numbers must be in the range of 0 – 99999 and be unique for each participant.

Origin: Product Management

Priority: 6 Stability: 1

1. Event record data and participant record data will stay in the system until the system is reset.

Origin: Client Requirements

Priority: 2 Stability: 1

1. Timing will be done in hundredths of seconds in the range of 0.00 – 9999.99 seconds.

Origin: Client Requirements

Priority: 1 Stability: 1

# IV. Constraints

1. Not having the hardware available during software creation.

# V. Requirements Dependency Traceability Matrix

Provide a cross-reference matrix showing related requirements as shown in the example

below. The matrix is used to identify dependencies between requirements to identify

when one requirement must be completed before another can be implemented.

Is dependent on requirement

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | REQ 1 | REQ 2 | REQ 3 | REQ 4 |
| REQ 1 | X |  |  |  |
| REQ 2 |  | X |  |  |
| REQ 3 |  |  | X |  |
| REQ 4 |  |  |  | X |

# VI. Development and Target Platforms

The expected development platform for the ChronoTimer 1009 software is intended to emulate the actual hardware because that hardware is in production during the same timeframe of this development effort. In order to emulate the hardware for during Sprint 1 we are using console-based input and test input files. For Sprint 2 we are upgrading the software to have basic Java GUI that resembles the layout of the actual hardware. Finally during Sprints 3 and 4 we are utilizing a Google’s web service platform and constructing the GUI on a web page.

The target platform for the ChronoTimer 1009 after development completes is the hardware that is being developed now which the software will be ported to. This hardware consists of a simple processor, embedded in a field box, with a display, printer and various keyboard controls.

# VII. Project Glossary

In sections 2 and 3, functional and non-functional requirements respectively, Priority and Stability are both measured on a scale of 1 – 10 with 1 being the highest and 10 being the lowest.

# VIII. Document Revision History

This section includes a list of significant changes that have been made to this document after the 1.0 version has been submitted for assessment.

|  |  |
| --- | --- |
| Version | 1.1 |
| Name(s) | Adam Permann |
| Date | 3/22/15 |
| Change Description | Added details about Sprint 2 and 3. |