TeamLead Application

Software Requirements Specification

James Williamson

EEC 626 – Software Engineering Project

Spring 2017

**Revision History**

|  |  |  |  |
| --- | --- | --- | --- |
| **Version** | **Author** | **Date** | **Description** |
| 0.1 | James Williamson | 02/14/2017 | Original revision. |
| 0.2 | James Williamson | 02/21/2017 | Updated requirements for ContextSwitch feature. |
| 0.3 | James Williamson | 03/22/2017 | Updated requirements for iteration 4. |

**Table of Contents**

[1) Purpose 4](#_Toc474939931)

[2) Background 4](#_Toc474939932)

[3) Document Conventions 4](#_Toc474939933)

[4) Functional Requirements: ContextSwitch UI 5](#_Toc474939934)

# Purpose

The purpose of this document is to specify the required functionality of the TeamLead application that will be developed for the EEC 626 Software Engineering Project at CSU.

# Background

Leading a software development team brings with it several unique challenges. A software team lead must be adept at development, coaching/mentoring, architecture, release engineering, project planning, and more. This requires the ability to switch tasks and contexts easily, and proficiency in the analysis and management of very large amounts of information. Time management and organizational skills are paramount.

As such, the TeamLead application is proposed in order to aid software team leads by empowering them to leverage of one of today’s most important and prevalent tools – the smartphone. There are myriad tools currently available for information management, communication, and note-taking, but most are either too generic or heavyweight to be consistently useful for the unique demands of this discipline. Worse, the learning curve may be too steep; this is an important consideration in a field where countless tools, frameworks, languages, and platforms must be adopted and utilized. TeamLead will provide a streamlined, user-friendly, and intuitive way for team leads to manage their time, capture and organize important information, and share content with developers.

# Document Conventions

The requirements listed in this document use different verbs to indicate necessity:

* “Shall” – The requirement must be implemented for proper operation of the system.
* “Should” – The requirement may be implemented, subject to time and complexity constraints.

Nested requirements may be listed under base requirements, in some cases. These should be treated as separate individual requirements; this practice is intended to help convey context. These are distinguished by indentation and sub-numbering throughout the document.

x.y) Base requirement

x.y.z) Nested requirement

# Functional Requirements: ContextSwitch UI

This section enumerates functional requirements for the “ContextSwitch” feature, which allows a user to track their time spent during a workday.

## The application shall allow the user to track time spent on various tasks throughout their workday via simple “task” button presses.

### When the user clicks a task button, the application shall log a context switch and begin recording time spent on the specified task.

## The application shall allow the user to add custom task buttons.

### Each task shall be identified by a name string, which is displayed on the top of the button.

### Each task shall maintain a timer depicting the total time elapsed for the task, which is displayed on the bottom of the button.

### The application should support the capability for the user to choose a custom color for each task button.

## The application shall allow the user to delete specified task buttons.

## After a button is pressed, the application shall darken the button color, to indicate that it is the active task.

## After a button is pressed, the application shall repaint the ContextSwitch activity UI once every second to show timer feedback to the user.

## The application shall allow the user to specify a “threshold” time for a task, which is the maximum desired amount of time to spend daily on that activity.

### When the configured threshold time for a task is exceeded, the application shall support the capability to generate a visual alarm.

### When the configured threshold time for a task is exceeded, the application should support the capability to generate an audible alarm.

## Once the workday is concluded, the application shall generate a graph to illustrate to the user how their time was allocated throughout the day.

### The application shall support the capability to generate the workday graph in pie chart format.

### The application should support the capability to generate the workday graph in time-slice format.

### The application should allow the user to index up to ten previous recorded workdays for analysis.

### The application should display the total number of context switches to the user.

## The application shall support the ability to be displayed on the lock screen in focus, facilitating ease of data entry and usability without necessitating the need to unlock a user’s phone.