Software Engineering Group Project
Tasker Team Tasking System
Requirements Specification

Author: N.W.Hardy
Config. Ref.: SE-QA-RS
Date: 2015-10-02
Version: 1.2
Status: Release

Department of Computer Science,
Aberystwyth University,
Aberystwyth,
Ceredigion, SY23 3DB,
U.K.

CONTENTS

1 INTRODUCTION

1.1 Purpose of this Document

This document describes the requirements for the Software Engineering Group Project 2015-16 for students studying in Aberystwyth University on modules CS22120 and CC22120 Software Engineering Life Cycle. It should be read in the context of the Group Project, taking into account the details of the group project assignment and the group project Quality Assurance (QA) Plan [1]. This requirements specification describes the functions of a software system to manage allocation and monitoring of tasks within a team. It also describes the requirements for the process of constructing the system.

1.2 Objectives

The objectives of this document are:

- to describe the background to the group project application for 2015-16 (*Tasker*);
- to provide details of the criteria that the group project product must meet;
- to describe the types of interaction with the system which must be supported.

2 General Description

2.1 Product perspective

Tasker is to be a network based computer system for the allocation of tasks to members of a team and the monitoring of their progress. A database will hold details of team members and of tasks. A web site will provide access to the database by managers for the creation, allocation and monitoring of tasks, while a desktop application for team members will receive task details and allow reporting of progress and completion.

2.2 Product Features

The product will provide the following features:

- Provision for a database server (*TaskerSRV*) which will receive and store team member and task data and make it accessible;

- A web site (*TaskerMAN*) which allows:
  - addition and maintenance of team member data;
  - creation and allocation of tasks;
  - viewing of task details.
- A desktop application (*TaskerCLI*) to operate both on-line and off-line and permit:
  - download and local storage of tasks for the user;
  - viewing and updating of task details;
  - regular synchronisation of locally stored tasks and the versions on *TaskerSRV*.

3 User Characteristics

The users of the system will have no special requirements, being general users of computer systems with standard user interfaces and an understanding of the tasks allocated to them.

4 SPECIFIC REQUIREMENTS

4.1 Functional Requirements

4.1.1 TaskerSRV

FR1 Manage team member data
For each registered team member, the following data must be held:

- full name
- e-mail address

Creation, update and deletion must be supported. Authentication and authorisation for this service is an optional feature.

FR2 Manage task data
For each task, the following data must be held:

- a title;
- the team member to whom it is allocated;

- the start date;
- the expected completion date;
- a list of tasks elements, with at least one member;
- a status, which may have the value `allocated`, `abandoned` or `completed`.

A task element is free text used to describe a step in the task, and associated free text for comments added by the member carrying out the task to report on progress with the step.

Creation and update of tasks must be supported.

Authentication and authorisation for this service is an optional feature.

4.1.2 TaskerMAN

FR3 Addition and maintenance of team member data
The site should provide for creation, update and deletion of team member data.

FR4 Addition and allocation of task data
The site should provide for the creation of tasks, provision of all data and allocation to a registered team member.

FR5 Reallocation of a task
The site should provide for reallocation of an existing task to a different team member.

FR6 Abandonment of a task
It should be possible to mark a task as `abandoned`.

FR7 View tasks
It should be possible to view the list of tasks, with titles and allocated team member. This list should be sorted by expected completion date. It should be possible to filter this list by task status and/or allocated team member.

Authentication and authorisation of users for all *TaskerMAN* facilities is an optional feature of the product.

4.1.3 TaskerCLI

FR8 User identification
On start-up the user must identify themselves to the application using their e-mail address as recorded in *TaskerSRV*. This address may optionally be recorded locally to speed future startup.

FR8a Local storage of tasks
Copies of tasks for the identified user must be stored locally. This may use files in the home directory of the current user.

FR9 Task synchronisation
Synchronisation of the list of tasks stored locally and the tasks with status `allocated` for the identified user held on *TaskerSRV* must be provided. Additions and deletions (through abandonment or completion) on *TaskerSRV* take priority. Local updates take priority.

FR10 Local editing
The user must be able to select a task from the current list and view its details. It must be possible to edit task step comments. It must be possible to change the task status to `completed`.

FR11 Synchronisaion timing
Synchronisation (FR9) must take place at start-up time and before and after local editing (FR10) if network access is available. It should also occur at least every 5 minutes if no editing has provoked it.

If network access is not available the application should proceed, keeping edits stored locally and available for synchronisation when access is available.

4.2 Interface Requirements

IR1 The User Interface The user interfaces should be intuitive to regular computer users.

IR2 Network Interfaces *TaskerSRV* may, but is not required to use HTTP as the protocol for providing services to *TaskerCLI*.

4.3 Performance Requirements

PR1 Response of program to user input

Any user input should appropriately reflected on the screen within 1 second. This includes provision for lack of network access or network latency, where appropriate messages should be provided to ensure that users are aware of progress if delays result.

PR2 Target computers for system

TaskerCLI should run on any machine supporting Java. *TaskerSRV* may be deployed on any suitable database server. *TaskerMAN* should be deployable on an Apache web server.

4.4 Design Constraints

DC1 programming languages

It is corporate policy to use Java for client program development such as *TaskerCLI*. *TaskerMAN* may be developed in any server-side language you are comfortable with and which is supported by Aberystwyth Information Services or the Computer Science Department.

DC2 Production of Test Data

The functionality of the software will be shown by handling at least 20 team members with a total of 30 allocated tasks.

4.5 Other Requirements

The project will be developed in line with the group project QA plan, detailed in [1]

REFERENCES

[1] *Quality Assurance Plan* C.J. Price, B.P.Tiddeman, N.W. Hardy SE.QA.01 2015-10-02 1.10

DOCUMENT HISTORY

Version	CCF No.	Date	Changes made to Document	Changed by
0.1	N/A	2015-09-28	Initial creation	NWH
1.0	N/A	2015-10-01	Draft for review	NWH
1.1	N/A	2015-10-02	Typos corrected.	NWH
1.2	N/A	2015-11-05	Two FR8 replaced by FR8 and FR8a	NWH