Software Engineering Group Projects

English Tutor

Requirements Specification

Author: C. J. Price

Config Ref: SE.QA.RS

Date: 30th April 2012

Version: 1.1

Status: Released

Department of Computer Science

Aberystwyth University

Aberystwyth

Ceredigion

SY23 3DB

Copyright © Aberystwyth University 2012

CONTENTS

1. INTRODUCTION 3

1.1 Purpose of this Document 3

1.2 Scope 3

1.3 Objectives 3

2. GENERAL DESCRIPTION 4

2.1 Product Perspective 4

2.2 Product Functions 4

2.3 User Characteristics 4

3. SPECIFIC REQUIREMENTS 5

3.1 Functional Requirements 5

3.2 External Interface Requirements 5

3.3 Performance Requirements 5

3.4 Design Constraints 6

3.5 Other Requirements 6

# INTRODUCTION

## Purpose of this Document

This document describes the requirements for the Software Engineering Group Project 2012 for students studying in Shenyang on the International Software Engineering degree. 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].

## Scope

This requirements specification describes the functions of a desktop computer based system to help students learn English, and the attributes that are expected from the finished product. It also describes the requirements for the process of constructing the system.

## Objectives

The objectives of this document are:

* To describe the background to the group project application for 2012 (English Tutor)
* 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

# GENERAL DESCRIPTION

## Product Perspective

English Tutor is a computer-based software system to assist people in learning English through a series of lessons with audio enhancement, and interactive tests to help the student absorb the lessons.

## Product Functions

The product will provide the following features:

* Presentation of a series of lessons, allowing the student to choose which lesson they wish to follow
* A record of which lessons a student has already covered and when
* Audio examples of how each phrase is pronounced
* Tests to assess how well the student knows the present lesson
* Tests to assess how well the student remembers all lessons learned so far
* Recommendations of where revision is necessary to improve performance

## User Characteristics

The software will be used by second year Computer Science students. These users are familiar with standard computer interfaces. They are, by default, quite lazy, and so the software should provide the indicated features with the fewest possible mouse movements and keystrokes.

# SPECIFIC REQUIREMENTS

## Functional Requirements

*FR1 Establishing identity of user*

When the software starts, it will give a brief summary of what it does, and ask whether the user is a new user or an existing user. For a new user it will establish a unique identity (e.g. by asking for a name). For an existing user, it will find the student's previous record.

*FR2 Starting to use the software (New User)*

When the user has given their details, the software will start with lesson one.

*FR3 Starting to use the software (Existing User)*

An existing user will be given a summary of all of the lessons they have successfully completed, and offered the following choices:

* learn next lesson not successfully completed
* take test for next lesson not successfully completed
* take test to revise all lessons learned so far
* Relearn previously done lesson

*FR4 Learning a lesson*

When a lesson is chosen to be learned, all of the Chinese phrases for the lesson are displayed. When the user clicks on one of them, the equivalent English phrase is displayed on screen and a recording of the phrase is played. The fact that the phrase has been looked at should be indicated on-screen, but the user should be able to look at and listen to the same phrase more than once. The user should ALSO be able to indicate that they have completed the lesson and are ready to take a test.

*FR5 Test for a lesson*

When the user requests a test for a specific lesson, then they should be given a random ordering of ten questions on the material in the test. The questions should be a mixture of:

* questions where the user is shown the Chinese and asked to type the English
* questions where the user is shown the English and asked to type the Chinese
* questions where the English is spoken and the user is asked to type the Chinese

On completion of the 10 questions, the test is scored and recorded. If the student gets more than 8 out of 10, they are allowed to proceed to the next lesson. If a student gets any wrong, they should be shown the correct answer.

*FR6 Test for all lessons learned so far*

The type of questions for this test are the same as for FR5, but the material used should be phrases taken at random from all lessons learned so far. On completion of the 10 questions, the test is scored and recorded. The user should be informed of any that they got wrong, and told which lessons contain the incorrect phrase.

*FR7 Lesson management*

It should be possible to add / delete lessons and phrases in lessons from a data administrator system.

## External Interface Requirements

*EIR1 Appearance of Interface*

The program should be intuitive to regular computer users.

## Performance Requirements

*PR1 Response of program to user input*

Any user input should be appropriately reflected on the screen within one second.

*PR2 Target computer for system*

All software produced should run correctly on computers with a Java interpreter.

## Design Constraints

*DC1* *Use of Java and Eclipse*

It is corporate policy to use Java on all major developments. The game should be built under Eclipse.

*DC2 Reuse of existing software*

Use of existing classes for basic data structures, such as graphs, is encouraged, as is full use of the standard class libraries, but should be documented.

*DC3 Format of provided data*

A set of lessons will be provided as an Excel spreadsheet with two tables.

First table has two columns:

Lesson number: the number of the lesson that this row is part of. Integer between 1 and 50.

Lesson title: the title of this lesson (e.g. “Greetings”)

Second table has four columns:

Lesson number: Integer showing which lesson (above) this is part of.

Phrase in Chinese: e.g. “早 上 好”

Phrase in English: e.g. “Good morning”

Name of sound file for this phrase: e.g. p0101.mp3

This spreadsheet should be used as a basis for the system. Chinese has only been provided for the first lesson - groups will be assigned other lessons to translate into Chinese and add to the file.

## Other Requirements

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

REFERENCES

[1] QA Document SE.QA.01 - Quality Assurance Plan.

DOCUMENT HISTORY

| *Version* | *CCF No.* | *Date* | *Changes made to document* | *Changed by* |
| --- | --- | --- | --- | --- |
| 1.0 | N/A | 30/04/12 | Draft | CJP |
| 1.1 | N/A | 10/05/12 | Changed "web-based" in Scope | CJP |