

UTB Accreditation Software

Software Requirements Specification

Version 1.0

October 27, 2014

Igor Gnatchenko
Andrey Gnatchenko

Prepared for
Software Engineering
Instructor: MK Quweider, Ph.D.
Fall 2014

Table of contents

1 Introduction	3
1.1 Purpose	3
1.2 Scope	3
1.3 Definitions, acronyms and abbreviations	4
1.4 References	4
1.5 Overview	4
2 General Description	5
2.1 Product perspective	5
2.1.1 System Interfaces	5
2.1.2 User Interfaces	5
2.1.3 Hardware Interfaces	6
2.1.4 Software Interfaces	6
2.1.5 Communication Interfaces	6
2.1.6 Memory Constraints	7
2.2 Product Functions	7
2.3 User Characteristics	8
2.4 Constraints	8
2.5 Assumptions and Dependencies	8
2.6 Apportioning of Requirements	8
3 Specific Requirements	9
3.1 External Interface Requirements	9
3.2 Functional Requirements	9
3.3 Performance Requirements	11
3.4 Logical Database Requirements	11
3.5 Design Constraints	11
3.6 Software System Attributes	11
3.6.1 Reliability	11
3.6.2 Availability	11
3.6.3 Security	11
3.6.4 Maintainability	11
3.6.5 Portability	12
3.7 Organizing the Specific Requirements	12
3.7.1 System Mode	12
3.7.2 User Class	12
3.7.3 Objects	12
3.7.4 Response	12

1 Introduction

1.1 Purpose

The main purpose of this document is to provide a reader with a high level description of the software that will be designed and developed. The main parts of the software, restrictions that the software can have and another necessary information that can be useful during development process will be detailed in this document.

This document can successfully be utilized by the development team and the customer as well. The development team can use this document as a guide based on which they will be able to plan their work, design application's architecture. When it comes to the customer, it will help them to firmly establish their requirements. It is useful to use this document to tackle any arguments that can arise between the development team and the customer.

1.2 Scope

The software that is being developed is named as "UTB Accreditation Software". The application will be able to automate the university's annual accreditation process. Also, it will allow a user to retrieve information related to Colleges belonging to Departments. In addition, it is available for the user to get necessary information about Colleges in the University.

The user can get different types of information:

- The mission and the vision of each College
 - The mission and the vision of each Department
 - The Department faculty information
 - The number of available programs for each department
 - Detailed information about specifically selected students
 - Learning outcomes of each selected course
- etc...

What makes this application useful is meaningful and intuitive Graphical User Interface, which will be designed according to the main GUI trends of other applications. This GUI will be fully customizable, which will make user experience better.

The application will be able to produce specifically formatted report containing information related to the selected entity (such as University, College or Department). Also, it is useful to notify a user about newly generated reports. This application can do it and will do so via email.

The groups of people by whom this application will be mainly used are:

- teachers and tutors
- faculty members
- students
- university executives

This application will have certain advantages compared to other applications with the same purpose. First of all, it will be implemented in Java programming language, which means that the application will be portable by nature. The second advantage is the modular structure of the application. It allows an easy upgrade in case of necessity.

1.3 Definitions, acronyms and abbreviations

Term	Definition
GUI	Graphical User Interface
UTB	University of Texas at Brownsville
SRS	Software Requirements Specification
IDE	Integrated Development Environment
Database	A special structure to store information
CVS	Concurrent Version System
UML	Unified Modeling Language
Email	Electronic Mail
Java	Object-oriented programming language
PDF	Portable Document Format
POS	Program of study
CV	Curriculum vitae
JVM	Java Virtual Machine
RAM	Random Access Memory
CPU	Central Processing Unit
OS	Operating System

1.4 References

At this moment there are no reference used in this document.

1.5 Overview

This document can serve as a source of general information about the application that is being developed. This document is divided into two logical parts.

The first part is high level requirements. In this part a reader can find information related to the main structure of the application, a brief description of an application's architecture. It is important to emphasize that detailed explanations are omitted in order to make understanding easier.

The second part is intended to give details of the application. It describes detailed architecture, additional functionality, modules' description and the links and relations between modules. This part can be utilized by the development team so as to create the software that meets requirements.

2 General Description

2.1 Product perspective

The "UTB Accreditation Software" is a standalone system that provides functionality briefly in the previous section. This application includes all subsystems that are needed to meet all software requirements. In addition, the application has interfaces to external systems, such Debug System, Version Control System, File System, Operating System. Names interfaces will be implemented in accordance with available industry documents and guides. The application can work in offline mode.

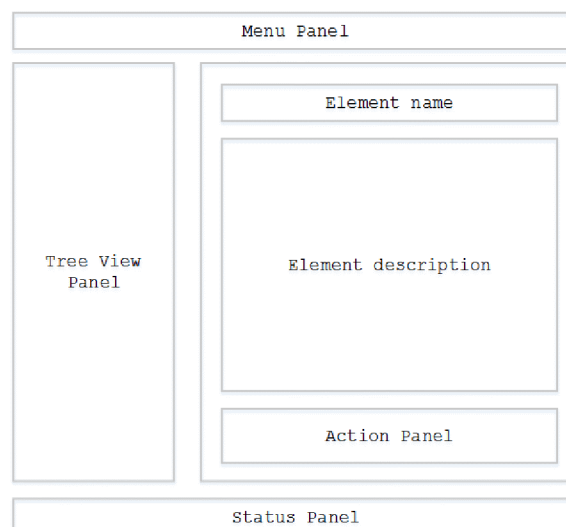
The application has to distinguish a Data Storage System from other external systems as without this system the application will not be able to function normally.

2.1.1 System Interfaces

There will be no system interfaces due to the fact that all functionality can be provided by JVM.

2.1.2 User Interfaces

The application will have GUI based interface. The interface will consist of three main panels. Each panel has its own function. Panel can be restyled in case the user wants to customize it. Also, a few functions will be doubled in menu and in working area as some users prefer to execute commands differently.



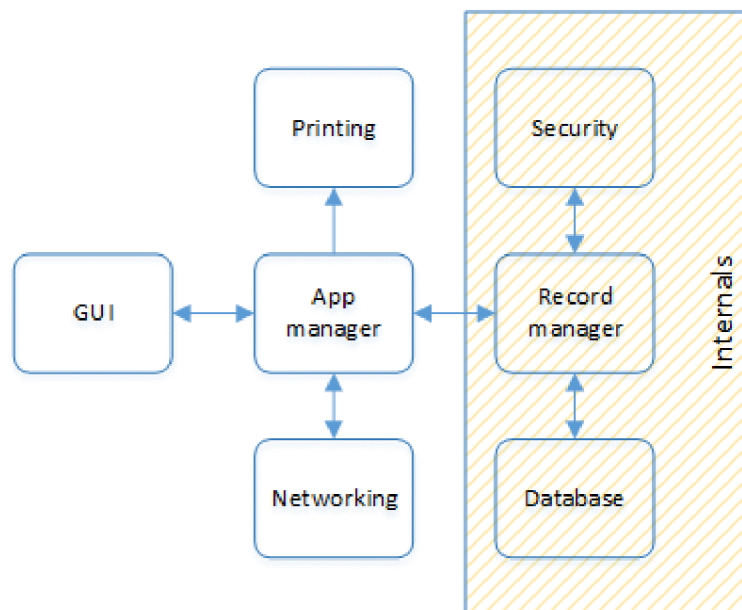
General interface overview

Essentially, here is how GUI will be organized:

Navigation panel. This panel is tree-based, which helps the users to select the object they are looking for. A tree-based panel is easy to navigate, intuitively understandable and many users like it.

Information panel. This panel shows the detailed view of the selected element. The content of this panel changes dynamically - when a user clicks another element in the navigation panel changes take place here.

Action panel. This panel accumulates the main functionality that can be used by users interacting with the application. The actions that can be taken by the user are editing existing element, adding new element, deleting existing element, printing selected element.



Application architecture

2.1.3 Hardware Interfaces

Since the language of choice in this case is Java and the application will run under control of JVM, which is hardware independent virtual machine, there are no specific hardware restriction. Consequently, it is advisable to use common configurations of hardware interfaces.

2.1.4 Software Interfaces

The development team will use Java as the main language to develop the application within the project. Java itself is a platform independent language and coupled with JVM provides developers with all necessary means. So no software interfaces are required except JVM.

2.1.5 Communication Interfaces

The application needs to have an Internet access so as to send notification emails and reports via email. Also, in the future it is possible to

expand functionality of the application so it can store information in a remote database.

2.1.6 Memory Constraints

Considering the fact that the application can be used on 32 bits processor architecture, it is necessary for the database that is a part of the application to be less than 4 GB in size.

2.2 Product Functions

The application can provide a user with a wide range of functions. These functions let the user to do accreditation work related to universities, colleges and departments with ease.

These functions can be written as follows:

Provide a user with the installation wizard

Show a splash screen when a user starts the application

Give access in two modes - the admin mode and the user mode

Input university information

- The name and the logo of the university
- The mission and the vision of the university
- The number of colleges belonging to the university

Input college information

- The name and the logo of the college
- The mission and the vision of the college
- The number of departments belonging to the college

Input department information

- The name and the logo of the department
- The mission and the vision of the department
- The number of learning programs belonging to the department

Input student information

- The first and the last name of the student
- The score of the student
- The gender of the student
- The age of the student

Input course information

- The name of the course
- The description of the course
- The number of practical lessons
- The number of lectures

Output selected element information

Create a new element

Edit an existing element

- Delete a selected element
- Print information about selected elements
- Notify users about changes via email
- Change display settings
 - Font settings
 - Color settings
 - Theme settings
- Show hits if needed
 - Popup hits
 - Information labels
 - Embedded help

2.3 User Characteristics

The main demand that is imposed on the potential users of this application is that the users should have basic level of computer literacy, they also should know how to work with input devices (such as a mouse and a keyboard) and general English knowledge.

2.4 Constraints

The development team should organize their work according to specific restrictions or constraints that will be imposed on this project and are listed below.

The application should be designed as a desktop application. It means that users will not be able to run this product on tablets, web browsers or cell phones.

The application should be developed in Java. It means that this application and all the modules should be developed so that they can be successfully executed on JVM. It is necessary for the application to have platform independent nature.

The application is designed to have a monitor as the main output device. User can click icons, choose different menu items and click buttons. So a general user cannot run this application on a machine equipped with a non-graphic monitor only.

2.5 Assumptions and Dependencies

It is necessary to make the assumption that the operating system under which the application is supposed to be running will have sufficient amount of RAM, CPU will have 32 or 64 bits architecture and at least a 256 color monitor will be available.

2.6 Apportioning of Requirements

It is possible that the application will be upgraded and will be able to connect not to the local database, but to the remote one. Therefore, Internet interfaces should be designed to take this factor into consideration.

3 Specific Requirements

3.1 External Interface Requirements

There is not any specific requirements that can restrict or limit development of this application.

3.2 Functional Requirements

Installation wizard	
Introduction	Provide a user with the installation wizard
Inputs	The basic directory path in which the application will be installed
Processing	Performs the installation process by unpacking the archive with the application
Outputs	Installed program in the path directory
Error handling	If there is no directory that is specified, notify a user

Splash screen	
Introduction	Show a splash screen when a user starts the application
Inputs	None
Processing	Loading all necessary resources and modules while showing the splash screen
Outputs	Loaded application
Error handling	If some of the necessary resources are not found, notify a user and terminate

Accounting	
Introduction	Give access in two modes - the admin mode and the user mode
Inputs	Usually a name and a password
Processing	Connect to the database, check for match and give privileges
Outputs	Given access such as "user" or "administrator"
Error handling	If there is no user in the database, notify a user with warning message

University input	
Introduction	Input university information
Inputs	Name, logo, mission, vision of the university, number of colleges
Processing	Insert information into the database
Outputs	The row is inserted into the database and a user is notified
Error handling	If a user forgets to input information, remind them to do so

College input	
Introduction	Input college information
Inputs	Name, logo, mission, vision of the college, number of departments
Processing	Insert information into the database
Outputs	The row is inserted into the database and a user is notified
Error handling	If a user forgets to input information, remind them to do so

Department input	
Introduction	Input department information
Inputs	Name, logo, mission, vision of the department, number of teaching courses
Processing	Insert information into the database
Outputs	The row is inserted into the database and a user is notified
Error handling	If a user forgets to input information, remind them to do so

Student input	
Introduction	Input student information
Inputs	First name, last name, score, gender and age of the student
Processing	Insert information into the database

Outputs	The row is inserted into the database and a user is notified
Error handling	If a user forgets to input information, remind them to do so

Course input	
Introduction	Input course information
Inputs	Course label, description, number of practical lessons, number of lectures
Processing	Insert information into the database
Outputs	The row is inserted into the database and a user is notified
Error handling	If a user forgets to input information, remind them to do so

Display element	
Introduction	Output selected element information
Inputs	Selected element that is going to be printed
Processing	Displaying the information in the appropriate form on the user's screen
Outputs	Displayed information related to a selected element
Error handling	If a user forgets to choose an element, remind them to do so

Create element	
Introduction	Create a new element
Inputs	Type of the element (university, college, etc.) and its position
Processing	Create a new element that will be stored in the database
Outputs	Specific element is created and stored
Error handling	Depends on the type of element. Generally, it is useful to notify a user

Edit element	
Introduction	Edit an existing element
Inputs	Selected element that is going to be edited
Processing	Edit existing element
Outputs	Specific element is edited
Error handling	Depends on the type of element. Generally, it is useful to notify a user

Delete element	
Introduction	Delete an existing element
Inputs	Selected element that is going to be deleted
Processing	Delete existing element from the database
Outputs	Specific element is deleted
Error handling	Depends on the type of element. Generally, it is useful to notify a user

Print element	
Introduction	Print information about selected element
Inputs	Selected element that is going to be printed
Processing	Create a report containing specific information
Outputs	Formatted report
Error handling	If a user does not specify the element to be printed, remind them to do so

Notification	
Introduction	Notify users about changes via email
Inputs	An input event by which a notification should be launched
Processing	Sending information letter via email
Outputs	Created and sent letter
Error handling	If Internet is not available or the email address is not valid, notify a user

Display settings	
Introduction	Change display settings
Inputs	Font, color, theme data related to the application
Processing	Change appearance of the application
Outputs	New look of the application

Error handling	If there is no such font or a theme was deleted, notify a user and set default theme
-----------------------	--

Display hits	
Introduction	Show hits if needed
Inputs	Cursor position
Processing	Show popup hit when the mouse cursor is hovering over an element
Outputs	Shown popup hint
Error handling	Do not display empty hints

3.3 Performance Requirements

Considering the fact that this application does not require instant responsiveness, performance is not considered as a major issue. Nonetheless, every operation described above should not take more than 3 seconds to execute.

3.4 Logical Database Requirements

No specific logical requirements will be imposed on the project. The only fact that matters is that the database must be relational (table) based and it should be able to store information and react to the users' queries.

3.5 Design Constraints

There is not any design constraints the development team should be guided by.

3.6 Software System Attributes

3.6.1 Reliability

The application should experience no more than two faults per week. In case of a fault, the application should provide the opportunity to save user data.

3.6.2 Availability

The application shall be unavailable for 5 minutes of the day. The main point here is that the user, who works with this application, should not experience the feeling of computer hang.

3.6.3 Security

Security should be provided by storing confidential data in the form of hash.

3.6.4 Maintainability

The average time for an administrator to restart the system in case of emergence should be no more than 4 person-hours.

3.6.5 Portability

The application shall be designed so that it can run in both the Windows and Linux operating systems. Generally, this software will be available in every OS that has JVM installed.

3.7 Organizing the Specific Requirements

3.7.1 System Mode

The application has two modes - user and administrator mode. In user mode the user is granted with read only access, whereas the user with administrator privileges is allowed to change information stored in the database.

3.7.2 User Class

User access will be provided by two classes: the first one reflects a normal user and the second one reflects an administrator.

3.7.3 Objects

Objects	
University	Class contains all necessary information related to a university entity
College	Class contains all necessary information related to a college entity
Department	Class contains all necessary information related to a department entity
Student	Class contains all necessary information related to a student entity
Report	Class contains all necessary information related to a report entity
Preferences	Class contains user preferences
Course	Class contains course related information
Database	Class provides all necessary functions to work with database
Hint	Class provides functions and methods to show hints

3.7.4 Response

Each action performed by a user should result in appropriate message on the screen to avoid ambiguity.