**Student Tracking And PeRformance System**

**(S.T.A.R.S)**

**CS3337 Software Engineering**

**Software Requirements, Software Design and Software Test Plan Document**

Prepared by

Arroyo-Martinez, Refugio

Borunda, Jaime

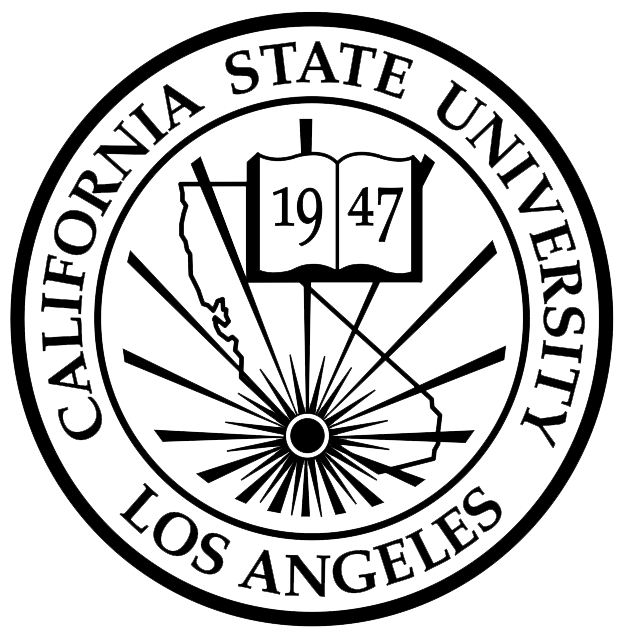
Obay, Leonardo

Orejel, Edmundo

March 25, 2017

CALIFORNIA STATE UNIVERSITY

LOS ANGELES



Los Angeles, California

This page intentionally left blank.

**Student Tracking And PeRformance System**

**(STARS)**

**Prepared By:**

Arroyo-Martinez, Refugio

Barunda, Jaime

Obay, Leonardo

Orejel, Edmundo

**Approved by:**

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_Month day, 2017**

Jose Macias Date

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_Month day, 2017**

Richard Cross Date

**Document Change Log**

|  |  |  |
| --- | --- | --- |
| **Update** | **Date Released** | **Changes** |
| Draft #1 | 3/25/2017 | Delivery of the Software Requirements document |
| Draft #2 | 04/12/2017 | Update to Sections …. |
| Final Document |  | Last update and upload to CSNS |

# **Table of Contents**

**1.0 INTRODUCTION**

1.1 Purpose 1-1

1.2 Scope 1-1

1.2.1 Document Organization 1-1

1.2.2 Relationship to Other Documents 1-3

1.3 **STARS** System Architecture 1-4

1.3.1 Context Diagram (Level 0 DFD)

1.3.2 Description of **STARS** Major Units

1.4 Requirement Development Process 1-9

1.5 References 1-11

1.5.1 Controlling Documents 1-11

1.5.2 Applicable Documents 1-11

1.5.3 Standards

**2.0 DETAILED FUNCTIONAL DESCRIPTION OF THE *STARS***

2.1 Detailed Functional Description 2-1

2.1.1 Higher Level DFDs

2.1.2 Detailed Functional Description of Major Sub-Units

**3.0 STARS REQUIREMENTS**

3.1 STARS FUNCTIONAL REQUIREMENTS 3-1

3.1.1

3.2 STARS NON-FUNCTIONAL REQUIREMENTS 3-

3.2.1

**4.0 STARS DETAILED DESIGN**

4.1 Introduction 4-1

4.2 STARS External Interface Descriptions

**5.0 STARS ELEMENTS OF IMPLEMENTATION**

5.1 Introduction 5-1

5.1.1 Assumptions and Constraints 5-

5.2 Implementation Goals 5-

**6.0 STARS TEST PLAN**

6.1 External Peer Reviews 6-1

**6.2.** Functional and Performance Requirements Validation Matrix

**A. ACRONYMS** A-1

**B. DATA DICTIONARY** B-1

**List of Figures**

Figure 1.1 Context Diagram (Level 0 Data Flow Diagram)

Figure 2.1 Level 1 Data Flow Diagram

**1.0 INTRODUCTION**

**1.1 Purpose**

The purpose of this document is **four-fold**:

a) Completely define a full set of requirements for the **Project Name – Section 3.0**.

(These sections correspond to a Software Requirements Document, SRD).

b) Completely define the design for the **Project Name – Section 4.0**.

(These sections correspond to a Software Design Document, SDD).

c) Define and partially implement feasible modules for the **Project Name – Section 5.0**.

(These sections correspond to the Software Implementation Document, SID).

d) Completely define the Test Plan for the **Project Name – Section 6.0**.

(These sections correspond to a Software Test Plan, STP).

The complete definition of all **Project Name** requirements provides the source requirement inputs for the development of the subsequent supporting software subsystems documents.

**1.2 Scope**

The documentation developed as part of this CS-3337 class, starts with the SRD including elements of Software Design and parts of a Test Plan.

The scope of this document includes the following:

* All functional and nonfunctional requirements on the **STARS** are captured. This includes Verification & Validation (V&V) requirements, as well as inter-software subsystems requirements.
* A complete set of **STARS** Requirements. These requirements are organized by key **STARS** functional units shown on the Level 1 DFD. The Level 1 DFD is shown on page [8].
* A trace matrix, relating all **STARS** functional requirements to functional subunits as expanded in lower level DFDs. Level 2 and higher DFDs are provided on pages [10].
* The functional requirements defined in the **STARS** Requirements section have been expanded to include more specific hardware requirements.

**1.2.1 Document Organization**

The organization of this document provides a natural flow or allocation of requirements to each succeeding section.

Details regarding the overall document are given in sub-section 1.5 below.

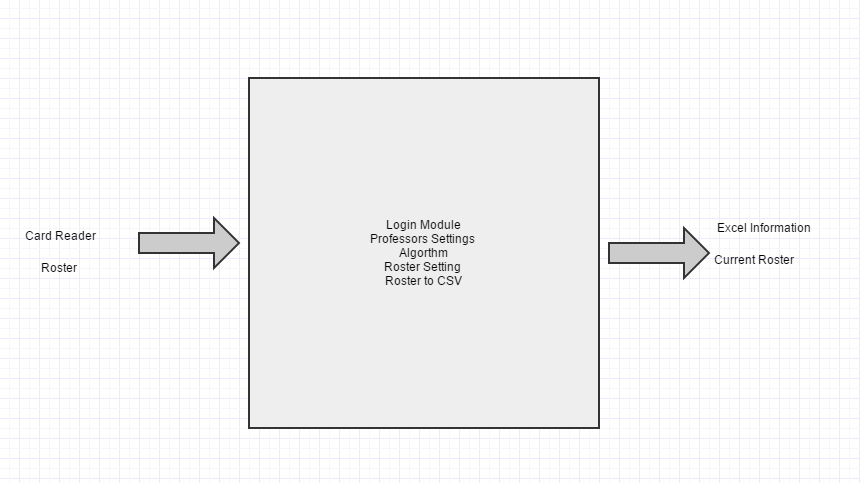
**1.2.2 Relationship to Other Documents**

The **STARS** SRD/SDD/STP/SID is a complete self contained document. Some relationships to other documents in the literature are indicated below in subsection 1.5.

**1.3 STARS Architecture**

1.3.1 Detailed Context Diagram (DFD Level 0)

The **STARS** architecture is summarized in the Context Diagram (DFD Level 0) given below. A more complete Functional Description is given in Section 2 of this document. The Context Diagram provides the overall structure of the software modules and all its inputs and outputs. The notation used corresponds to that defined for any Data Flow Diagram (DFD).



**Figure 1.1 DFD Level 0**

1.3.2 Description and major functions of the STARS

Student Tracking and PeRformance System (STARS) is the next generation of attendance and monitoring system. It provides the average professor/classroom the ability to monitor and track

attendance and gather useful information anytime without ever losing class time. Through a process called STARS software, the client will be able to see and monitor his/her class environment through the stars software without losing valuable time.

With the STARS technology classrooms do not need to waste valuable time and resources on attendance the software will track users based on a simple and effective system. The students in this case, sign in with school ID’s and check in to a predefined roster. The professor can view an attendance record from his laptop online. STARS manages attendance records and collects time logs for a professor/company to benefit from. With STARS the client can benefit from real time viewer and set given time intervals to their needs.

The client will be prompted a login page where he is pre-registered on. The client will then set settings to accommodate his needs and the roster is live online ready to view. Students will login and the roster will become populated setting time of login and marked late if late and absent if absent.

**1.4 Documentation Development Process**

The **STARS** detailed functional description is documented in section 2.0. Basically, Section 2 is a succinct software description document. The overall detailed functional description is based on higher level DFDs (above level 1). All major functional units are described in detail in this part of the document.

In general, all requirements affecting **STARS** are captured in Section 3.0 of this document. These requirements are a refinement and completion of requirements first collected as part of this Software Engineering project. The document is cited in Section 1.2.2. This section is the one worked in most detail to become a reasonably complete Software Requirements Document (SRD). It includes both functional and non-functional software requirements together with several detailed “rational” paragraphs whenever necessary to complete the understanding of each requirement.

Section 4 is the detailed **STARS** Software Design Description Document (SDD). This part of the document includes all higher level DFDs as described in section 2 plus all interface units. The document is highly technical and it is based on section 2 descriptions. An important component is the addition of a SIS (software interface specification) document in sub-section 4.2.

Section 5 includes elements of a partial implementation of **STARS**. This section includes the various constraints that effectively limit the implementation as well as the sub-units that will be coded. The implementation goals are defined and the code and pseudo code are included as an attachment to this section.

Section 6 is the last major section in this document and includes the overall Test Plan (TP) of the **STARS**. The test plan details the various techniques used to test the requirements and it also includes a Validation Matrix where each requirement specified in section 3 is listed with its corresponding validation method. The validation methods may include Testing, Analysis and Demonstration, and possible other V&V methods. In addition, the TP specifies the mandated peer reviews needed to validate the stakeholders part of the requirements.

**1.5 References**

All references used in the creation of this document are listed below.

**1.5.1 Controlling Documents**

1) There is no document controlling this document.

**1.5.2 Applicable Documents**

1) No additional applicable document has been used in the production of this document.

**1.5.3 Standards**

No Standard has been used in the creation of this document. However, some Standards described in textbooks have been examined as a reference. In particular, the IEEE standard has been briefly discussed in class.

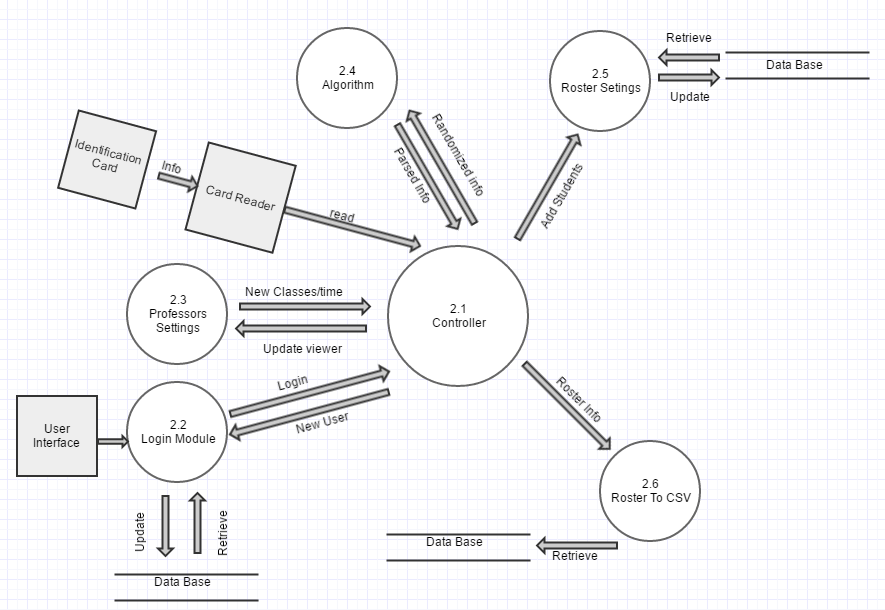
**2.0 DETAILED FUNCTIONAL DESCRIPTION OF THE *STARS***

2.1 Detailed **STARS** Functional Description.

The major tool used to design **STARS** is the Data Flow Diagram, DFD. The rational behind the selection of DFDs as the preferred design tool, was their simplicity and versatility. In the future more sophisticated tools may be used particularly if a correlation from Design to Requirement to Implementation and Testing is found to be a necessary addition.

2.1.1 Higher Level Data Flow Diagrams.

The **STARS** major functional design components are shown in the DFDs below.



**Figure 2.1: Level 1 DFD**

2.1.2 Detailed Description of **STARS** Major Sub-Units

The **STARS** major functional subunits shown in the DFDs in the previous sub-section, are described in detail below.

INTERNAL Descriptions

Card Reader- the card reader is the initial process a piece of hardware it gets randomized information from a user's identification card. Through the card reader we can take advantage of an already existing useful tool we all have as students an Identification card and the card reader can give us quick access to the information.

Algorithm- the algorithm parses randomized information into valid information. The card reader itself will not give us sorted information so the algorithm sub-unit will provide us a way to use the information in a organized matter. For example the information from the identification card parsing strings and integers the information form to compare users firstname, lastname, or CIN from a database to validate the user.

Validation- once the Algorithm is computed the Validation will search the student and check a MySQL database for reference checking if student exists in the database.

Database- The database is used to store the user’s username and CIN information. The client must first create an account where he/she will add the students. After the client has created their information it will then be stored in the database system.If a student does not exist he/she can be added later or a student can be discarded.

Viewer- The user interface is a web page with a username and password field. Prior to using the system, the user must create an account. If the user already created an account, they can login into the

system with their login credential. If the user is a new user they will need to register with the system.

There is a link for the new user and when they click on the link it will take them to another page where

they will fill in all the required information and submitted to the system. The viewer will display a table of students with a clear status of absent, late, or present. If the students are late the time record will be displayed for the user to see and a logout for the user.

**3.0 *STARS* REQUIREMENTS**

3.1 Project Name Functional Requirements

This Section collects all **STARS** Functional Requirements. The Section includes the complete set of functional requirements with explanation and rational where the statement of the requirement was deemed insufficient or needing additional background/justification. All requirements relate to the design modules described in Section 2. An effort has been made to standardize the correlation between the design modules and the requirements to make their access and organization more consistent. For example, module 2.1 requirements are labeled 3.1, sub-module 2.1.1 requirements are labeled 3.1.1 and so on. The list of requirements follows.

|  |  |
| --- | --- |
| *Requirements* | *Related to Design Module 3.1 : Main Controller and Sub-modules 3.1.1,3.1.2,etc...* |
| **Requirement No.** | **Requirement Description** |
| 3.1-1 | The Main Control Module shall receive data from Student ID |
| 3.1-2 | The Main Control Module shall pass any Student info received to a database |
| 3.1-3 | The Main Control Module shall display any error caused by the user |
| 3.1-4 | The Main Control Module shall send incoming info to designated destinations. |

|  |  |
| --- | --- |
| *Requirements* | *Related to Design Module 3.2 :Login Controller and Sub-modules 3.2.1,3.2.2,etc...* |
| **Requirement No.** | **Requirement Description** |
| 3.2-1 | The Login Control Module shall receive data from professor/instructor |
| 3.2-2 | The Login Control Module shall validate credentials from the professor/instructor |
| 3.2-3 | The Login Control Module shall display any error caused by professor/instructor |
| 3.2-4 | The Login Control Module shall login the user |

|  |  |
| --- | --- |
| *Requirements* | *Related to Design Module 3.3 :Database and Sub-modules 3.3.1,3.3.2,etc...* |
| **Requirement No.** | **Requirement Description** |
| 3.3-1 | The Database Module shall receive and store students data |
| 3.3-2 | The Database Module shall verify that the student is enrolled in the class |
| 3.3-3 | The Database Module shall hold multiple rosters |
| 3.3-4 | The Database Module shall send incoming info to designated destinations. |

|  |  |
| --- | --- |
| *Requirements* | *Related to Design Module 3.4 : Settings and Sub-modules 3.4.1,3.4.2,etc...* |
| **Requirement No.** | **Requirement Description** |
| 3.4-1 | The Settings Module shall allow the professor to modify class settings |
| 3.4-2 | The Settings Module shall display what course |
| 3.4-3 | The Settings Module shall choose what the roster of the class |
| 3.4-4 | The Settings Module shall enable the professor to add/delete students from roster |

|  |  |
| --- | --- |
| *Requirements* | *Related to Design Module 3.5 : Courses and Sub-modules 3.5.1,3.5.2,etc...* |
| **Requirement No.** | **Requirement Description** |
| 3.5-1 | The Courses Module shall allow the professor to add/delete a new section. |
| 3.5-2 | The Courses Module shall allow the professor to set a late time. |
| 3.5-3 | The Courses Module shall allow the professor to set the time when class starts. |
| 3.5-4 | The Courses Module shall allow the professor to select the section of there choosing. |

**3.2 *STARS* Non-Functional Requirements**

This Section collects all the **STARS** Non-Functional Requirements. All non-functional requirements are numbered “NF – n” where “n” indicates the nth requirement.

NF - 1 **STARS** requires sufficient data storage to maintain records of professors

preferences, roster of students in the class.

NF - 2 **STARS** must respond quickly enough to avoid frustration by users. Upon

checking into class , there should be a maximum 2 second processing

time before a response is represented to the user indicating whether or not they were checked in.

NF - 3 **STARS** must be operable by an average, everyday person, and should

be intuitive enough to fit this requirement.

**3.3 *STARS* Hardware Requirements**

This Section collects all the Project-Acronym Hardware Requirements. All hardware requirements are numbered “H – n” where “n” indicates the nth requirement.

H - 1 ***STARS***  will run on standard microcomputer hardware that is able to access the internet.

H - 2 ***STARS***  will utilize a standard card reader machine that will be connected to the microcomputer through usb port.

H - 3 ***STARS***  will be a system for taking attendance in a very large class.

**4.0 *STARS* DETAILED DESIGN**

In this section the **STARS** described in Section 2 with requirements listed in Section 3 will be designed in detail possibly including higher level DFDs. Each major module detailed design is included in correspondence with the design sections defined in Section 2 and responding to the requirements listed in its correlated sub-section in chapter 3.

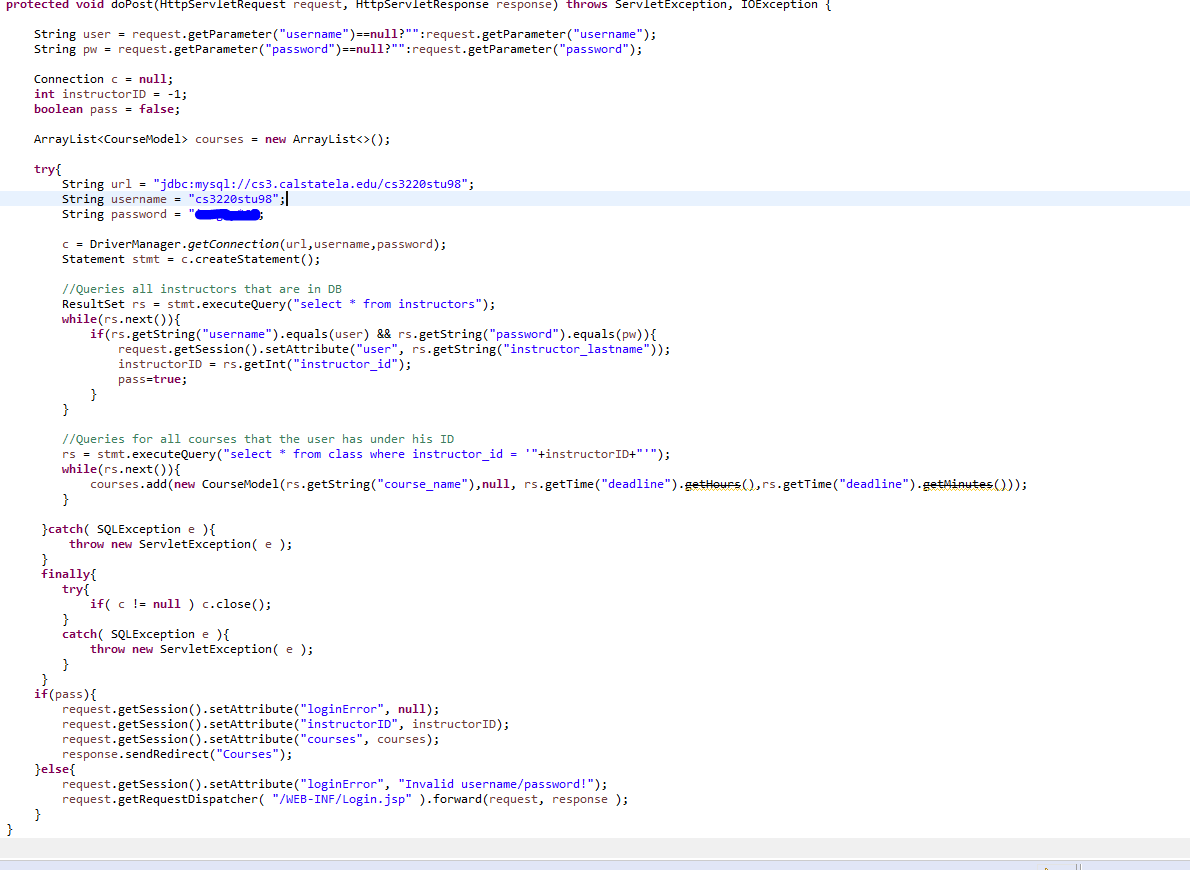
**5.0 *STARS* ELEMENTS OF IMPLEMENTATION**

In this section (some of) the modules designed in Section 4 with requirements listed in Section 3 will be implemented initially at least at the level of pseudo code. Where possible, actual code will be provided. Each module is implemented in correspondence with the design sections defined in chapter 2 and responding to the requirements listed in its correlated sub-section in chapter 3.

5.1 STARS Module 3.1 (This module is the main controller that initiates the program)



5.2 STARS Module 3.2 (This module is the Login module)



5.3 STARS Module 3.3 (This module takes student info and submit it to DB)



5.4 STARS Module 3.4 (This module manages the classes settings)



5.5 STARS Module 3.5 (This module keeps to track of all courses and allows for adds and deletion)



**6.1 INTRODUCTION**

In this section the testing methodology to be used to V&V each of the requirements listed in section 3.0 has been identified. At points some additional testing may be required and they shall be documented as an attachment to this document.

The methodologies and testing strategies identified at this point include four major approaches: TESTING, DEMONSTRATION, INSPECTION, and ANALYSIS with various variations to adapt to the ***STARS*** characteristics:

* **Testing** using additional ad-hoc created software including a correlation testing unit.
* **Demonstration** of the specified capability
* **Inspection** of the software code possibly using additional inspection techniques
* **Analysis** of the specific code operation/algorithm to prove functionality.

**6.2 FUNCTIONAL REQUIREMENTS VALIDATION MATRIX**

The *STARS*  Functional and Performance Requirements Validation Matrix is given below.

|  |  |  |
| --- | --- | --- |
|  | ***3.1 : Main Controller*** |  |
| **Requirement No.** | **Requirement Description** | **Testing Methodology** |
| 3.1-1 | The Main Control Module shall receive data from Student ID | Demonstration |
| 3.1-2 | The Main Control Module shall pass any Student info received to a database | Testing |
| 3.1-3 | The Main Control Module shall display any error caused by the user | Testing |
| 3.1-4 | The Main Control Module shall send incoming info to designated destinations. | Testing |
|  |  |  |

|  |  |  |
| --- | --- | --- |
|  | ***3.2 : Login Module*** |  |
| **Requirement No.** | **Requirement Description** | **Testing Methodology** |
| 3.2-1 | The Login Control Module shall receive data from professor/instructor | Testing |
| 3.2-2 | The Login Control Module shall validate credentials from the professor/instructor | Testing |
| 3.2-3 | The Login Control Module shall display any error caused by professor/instructor | Testing |
| 3.2-4 | The Login Control Module shall login the user | Testing |
|  |  |  |

|  |  |  |
| --- | --- | --- |
|  | ***3.3: Database Module*** |  |
| **Requirement No.** | **Requirement Description** | **Testing Methodology** |
| 3.3-1 | The Database Module shall receive and store students data | Testing |
| 3.3-2 | The Database Module shall verify that the student is enrolled in the class | Testing |
| 3.3-3 | The Database Module shall hold multiple rosters | Testing |
| 3.3-4 | The Database Module shall send incoming info to designated destinations. | Testing |
|  |  |  |

|  |  |  |
| --- | --- | --- |
|  | ***3.4 : Settings Module*** |  |
| **Requirement No.** | **Requirement Description** | **Testing Methodology** |
| 3.4-1 | The Settings Module shall allow the professor to modify class settings | Testing |
| 3.4-2 | The Settings Module shall display what course | Testing |
| 3.4-3 | The Settings Module shall choose what the roster of the class | Testing |
| 3.4-4 | The Settings Module shall enable the professor to add/delete students from roster | Testing |
|  |  |  |

|  |  |  |
| --- | --- | --- |
|  | ***3.5 : Course Module*** |  |
| **Requirement No.** | **Requirement Description** | **Testing Methodology** |
| 3.5-1 | The Courses Module shall allow the professor to add/delete a new section. | Demonstration |
| 3.5-2 | The Courses Module shall allow the professor to set a late time. | Demonstration |
| 3.5-3 | The Courses Module shall allow the professor to set the time when class starts. | Demonstration |
| 3.5-4 | The Courses Module shall allow the professor to select the section of their choosing. | Demonstration |
|  |  |  |

**A. ACRONYMS**

**See examples below**

**CT** Computerized Tomography

**PDA** Personal Digital Assistant

**V&V** Verification and Validation

**GB** Gigabyte

**STARS** Student Tracking And PeRformance System

**AA** Automated Attendance

**DB** database

**CR** Card Reader