**Software Requirements Specification**

**for**

**WISD Registry Entry Application**

**Version 1.4 approved**

**Prepared by Bill Conn**

**August 2, 2018  
Updated: July 26, 2020**

* **Introduction**
* **Purpose**

An order entry application to allow students of WISD to create students and place schedule orders into a database. The scope of this SRS is the desktop application and its backend support system.

* **Document Conventions**

This document follows the following conventions:

* **Bold** text signifies user actions.
* *Italic* text signifies existing systems outside the scope of this document.
* Requirement statements are prioritized individually.
* **Intended Audience and Reading Suggestions**

This document serves as a reference for all Capestra Furniture employees who have a stake in its implementation, testing, deployment, and ongoing support, including system architects, developers, test engineers, support engineers, operations, and legal. This document covers multiple facets of the software application. It is recommended that it is first read end-to-end, and then serve as a reference source.

* **Product Scope**

The application will run on Windows desktops but can be ported to mobile devices. It will provide a frontend to WISD backend systems for the purpose of handling student class entry. It will allow students to create schedules, select classes, view students details, view class details, view order details, and view product details.

* **Overall Description**
* **Product Perspective**

WISD Register entry application is a new product that will integrate with WISD third party web site. Students will be provided support to be migrated to the desktop application database. In future releases, data from this database will be synced to the third-party website or directly linked to the hosting database.

* **Product Functions**

The Student Registry order entry application will consist of the following primary features:

* Allow the students to log into the application and connect to the database.
* Allow the students to add students.
* Allow the students to register for a class.
* Allow the student to run a report displaying a list current classes as well as those he or she registered for.
* Allow the student to run a report displaying the details about a specific class.
* Allow the teacher to run a report displaying the details of all students registered for a class.
* Allow the teacher to run a report displaying the his or her class schedule.
* **User Types and Characteristics**
* All teachers will be able to access all part of the application.
* All students will have access to Login, Add Student, Select Class, Student Progress, Reports of Class and Student Schedule
* **Operating Environment**

The application will operate on any device using JRE 8 or higher.

The application will be designed to support future versions as well through automatic updates.

* **Design and Implementation Constraints**

The expected implementation constraints are:

* Use Java and JavaFX for development.
* Use NetBeans 8.2. for the IDE.
* Use MySQL 5.7 for the database.
* **User Documentation**

The user documentation will include:

* Commented code.
* Presentation of the final implementation.
* Contextual help within the application.
* **Assumptions and Dependencies**

The employee list and content will be provided from the third party in a .csv file. Capestra Furniture will also provide a template application to expand on with documentation to explain the application framework.

* **External Interface Requirements**
* **User Interfaces**

These user interfaces will be provided by the designer/developer of the application:

* Student login screen
* Add student screen.
* Selecting Career Path Screen screen.
* Select Classes Screen
* Progress/Audit Screen
* Report screen for class.
* Report screen for student details.
* Report screen for class details.
* Report screen for teacher schedule details.
* **Hardware Interfaces**

The software will work with the any OS that supports Java JRE 8. There will be no hardware interfaces that are out of the ordinary. IOS and Android

* **Software Interfaces**
* The application will be written in Java and JavaFX and will interface directly with the MySQL database.
* The application will not rely on any third-party libraries beyond the Java SDK.
* Data will be communicated to and from Jacket Registar backend system using the JDBC protocol.
* **Communication Interfaces**
* The application must communicate with MySQL 5.7 (**not 8.0**) using JDBC drivers.
* **System Features**

The system provides a menu bar containing two drop-down menus: **Action** and **Report**. The 4 Actions are: **Login**, **Add Student**, and **Select Class, Student Progress**. The 4 Reports are: **Class Info**, **Student Schedule**, **Teacher Schedule**, and **Teacher Roster**.

* **Action: Login**

4.1.1 Description and Priority

All students from WISD have an account in the web site data. All of the student data will be provided through a .csv file for import into the database. This feature is of high priority.All Teachers will from WISD will have an account in the web site data. They will have access to all reports whereas the students will not except for class info and student schedule.

4.1.2 Stimulus/Response Sequences

The student/teacher opens the application for the first time.

The student/teacher is prompted for username and password to log into the system.

The student/teacher enters his or her existing account credentials.

The employee presses the “Login” button.

The system responds telling if the credentials are valid or not.

4.1.3 Functional Requirements

REQ-1: Render login screen.

REQ-2: Validate student/teacher username / password against the database.

REQ-3: Check with the database to determine if an account with the entered username already exists.

REQ-4: Log the student/teacher into the system. Display a message telling the first name, last name, and ID of the employee who just logged in.

REQ-5: An appropriate error message must be displayed if the database is inaccessible.

* **Action: ADD Student**

4.2.1 Description and Priority

The student should be able to add a career pathway into the application to be inserted into the database. This feature is of a high priority.

4.2.2 Stimulus/Response Sequences

The student opens the application.

The student logs into the application.

The student selects “Add Student” from the Action menu.

The student enters in the customer’s **first name**, **middle name, last name**, **email address, graduating year**, **career pathway**.

The student presses the “Add student” button to add the student.

The system responds telling if the student was added successfully or if there was an error.

4.2.3 Functional Requirements

REQ-1: Verify the student has logged in.

REQ-2: Verify that the student was successfully added.

REQ-3: Validate that all of the input fields have been entered.

REQ-4: The successful insertion confirmation message must include the name of the student that was added.

REQ-5: Display any errors (like missing fields) in an appropriate fashion; e.g., beside each input field or above all of the input fields.

REQ-6: All input fields should be cleared if and only if the insertion was successful.

REQ-7: A student must not be added if there already is a student with the same first name, last name, and email address in the database. In this case an appropriate message must be displayed.

REQ-8: A note stating that if you have already been added you can skip this step.

* **Action: Select Class**

4.3.1 Description and Priority

The student should be able to select up to 4 classes. The student should be able to audit to track progress towards goal

4.3.2 Stimulus/Response Sequences

The student opens the application.

The student logs into the application.

The employee selects “select class” from the Action menu.

The system displays a list of classes in a drop down.

The system displays a list of career pathways in a drop down.

The student selects one career pathway

The student selects a class (up to 4).

The student presses the “select” button to place the order.

The system responds telling if the classes was added successfully or if there was an error.

4.3.3 Functional Requirements

REQ-1: Verify the student has logged in.

REQ-2: Check that there is enough seats to take the course. If not, raise an exception, otherwise, continue.

REQ-3: Check that the student has taken a prerequist to take the course. If not, raise an exception, otherwise continue

REQ-4: Insert a record in the **student** table to include in the customer ID, employee ID who is logged in, date of the class being put in, and the class status (approved, not approved)

REQ-5: Insert a record in the **class\_detail** table to indicate the selection ID, class ID, .

REQ-6: Update the **class** table to deduct the number of items ordered.

REQ-7: Display any errors either beside each input field or above all of the input fields.

REQ-8: Validate that all of the input fields have been entered.

REQ-9: All of the database statements must execute in a database transaction.

REQ-10: All input fields should be cleared if the transaction was successful.

* **Report: Student Schedule Info**

4.4.1 Description and Priority

The system should display a list of current current classes. This system should display future classes and be able to print out new schedule

4.4.2 Stimulus/Response Sequences

The employee opens the application.

The employee logs into the application.

The employee selects “Student Schedule” from the Report menu.

The system responds by displaying the list of current classed. The information displayed must include employee ID, first and last names, and email address. current classes and classes that have been registered

4.4.3 Functional Requirements

REQ-1: Verify the employee has logged in.

REQ-2: Display an appropriate message if there are no classes for the upcoming semester.

REQ-3: All student information must be displayed in tabular form.

* **Report: Teacher Schedule Info**

4.5.1 Description and Priority

The system should display full details about the specified teacher. This feature is of a high priority.

4.5.2 Stimulus/Response Sequences

The teacher opens the application.

The teacher logs into the application.

The teacher selects “Teacher Schedule Info” from the Report menu.

As soon as the employee selects a specific teacher, the system displays all details about the specified teacher including id, first name, last name, class name, time of class, Number of students in class

4.5.3 Functional Requirements

REQ-1: Verify the teacher has logged in.

REQ-2: Display an appropriate message if there are no teacher or if no teacher has been selected.

REQ-3: Teachers are selected using a drop-down list.

REQ-4: Display full details about the specified Teacher.

REQ-5: The teacher selection drop-down should remain available during the display of the teacher information, being shown just above the teacher details.

* **Report: Teacher Roster Info**

4.6.1 Description and Priority

The system should display a three-part screen including: (a) List of all students in class; (b) List showing details of students; This feature is of a high priority.

4.6.2 Stimulus/Response Sequences

The teacher opens the application.

The teacher logs into the application.

The teacher selects “Teacher Roster Info” from the Report menu.

In the top table, the system displays all student details including first name, last name and all contact information for the student, ordered by student's last name.

.

4.6.3 Functional Requirements

REQ-1: Verify the employee has logged in.

REQ-2: Display an appropriate message if there are no students..

* **Report: Class Info**

4.7.1 Description and Priority

The system should display a list containing full details about all classes in inventory. This feature is of a high priority.

4.7.2 Stimulus/Response Sequences

The student/teacher opens the application.

The student/teacher logs into the application.

The student/teacher selects “Class Information” from the Report menu.

The system displays a list showing full details about all classes in inventory including their categories, names, descriptions, pre-requists, and quantities of seats available.

4.7.3 Functional Requirements

REQ-1: Verify the teacher/student has logged in.

REQ-2: Display an appropriate message if there are no students/teacher.

* **Other Nonfunctional Requirements**
* **Performance Requirements**

The order entry system needs to be fast. This will require the SQL statements to be optimized and the database design to be set up in third normal form.

* **Safety Requirements**

The application has no special safety requirements.

* **Security Requirements**

All communications need to be properly authenticated.

Because the application interfaces with other systems, it need to be able to handle errors and failures gracefully, with sensible re-try logic and clear messaging for the user.

WISD is intending to expand this application in the future. Therefore, the application needs to be designed with componentization and reusability in mind.

* **Other Requirements**

The application needs to have locally-managed storage.

The backend system will interface with Capestra Furniture’s MySQL 5.7 **(not 8.0**) database server.

The backend system will communicate with the MySQL database using a JDBC driver.

The application needs to be in English, but designed with internationalization in mind for future versions.

The minimum tables that are required for the database include:

* Student (consists of student related information including **first name**, **last name**, **email address**, ).
* Teacher (consist of teacher related information including **first name, last name, unique id**
* Class\_table (consists of high-level order information such as the order date, the student who ordered it, ).
* Class Description (product information including the class name, description, pre-reques, and quantity currently available).
* Career Pathways (list of the categories for the pathways.).

**Appendix A: Glossary**

* API: Application Programming Interface. A definition and implementation of a layer that exposes and provisions certain services to external systems (such as client applications).
* LAN: Local Area Network. A network of a group of related computers and other devices (usually within a home or an office).