Dance Registration System

Evaluation Manual

Group Number: 2

CPSC 488 Section 01

**TABLE OF CONTENTS**

**1 INTRODUCTION………………………..……………………………………………………3**

* 1. DESCRIPTION……………………………………………………………………………….3

**2 CURRENT STATE………….………………………………………………………………....3**

2.1 PROBLEMS………………………………...………………………………………………...3

*2.1.1 Database Backup*…………………..…………………………………………......*.*...3

*2.1.2 Deactivating Children/Activities*………..……………………………………….......3

*2.1.3 Foreign Keys….*……………..……………………………………………………....3

*2.1.4 Deleting Activities with Students Enrolled*………………………………………….4

2.2 CLEANUP……………………..…...…………………………………………………………4

*2.2.1 Database Backup*…………………..…………………………………………......*.*...4

*2.2.2 Deactivating Children/Activities*………..………………………………………......4

*2.2.3 Foreign Keys….*……………..……………………………………………………....4

*2.2.4 Deleting Activities with Students Enrolled*………………………………………….4

**3 FUTURE STATE…………....………………………………………………………………....5**

3.1 IMPROVEMENT…………………………...………………………………………………...5

*3.1.1 Calculating Class Charges….*……..……………………………………………......*.*5

*3.1.2 Parent Payment*…………………………………………………………………...…*5*

*3.1.3 Child Attendance*……………………………………………………………………*.5*

*3.1.4 Security*………………………………………………………………………..….....5

*3.1.5 Edit Activity*……..……………………………………………………………..….....5

1 INTRODUCTION

* 1. **Description**

The purpose of this report is to describe where current problems lie in the program. The problems listed are a high priority and need to be implemented first to ensure maximum efficiency. While discussing the problems, the Clean Up section describes possible solutions and the logic behind why these problems are depicted as higher priority. The Future State section lays out lower priority issues to be implemented as the program is passed along. Additionally, that section discusses features of the program that were unable to be implemented in the given time frame.

2 CURRENT STATE

**2.1 Problems**

2.1.1 Database Backup

The admin should have the option to backup the database at any point to retrieve the information being stored. Additionally, there should also be a method that is continuously backing up the database. Methods were written for the backup from a button push and backup most of the database but fail to retrieve the Children data.

2.1.2 Deactivating Children/Activities

The manager or administrative user are allowed the privileges to remove children from the classes they are enrolled in. In the current state of the program, when a child is removed from a class that entry is removed off of the HTML page and also removed from the database. The children removed from classes should show as deactivated from the given class. With this implementation, this should still log and show that the child was once enrolled in the class.

The same logic should be implemented for deactivating activities on through the schedule HTML page. Activities that were once offered at the dance studio should still be shown in the database, along with the rosters for said activities.

2.1.3 Foreign Keys

Foreign keys are the simplest way to connect database tables with corresponding data to further display and understand the data in the database. Unfortunately, these were not implemented correctly, therefore, variables that should be foreign keys have become composite keys or are simply hard-coded to load into the database and, for the most part, queried when needed.

2.1.4 Deleting Activities with Students Enrolled

Currently, managers can initiate a deletion of an activity. The manager can delete an activity, whether there are students enrolled or not. However, in order to prevent issues with class enrollment, the manager should only be able to delete classes that do not have any students enrolled. Students should either have to be deleted or deactivated from the activity set to be deleted. There should also be a method to notify the parents that a class was deleted from the schedule.

**2.2 Cleanup**

2.2.1 Database Backup

In order to fully backup the database, the method needs to be updated to include logic for backing up the child data. After this logic is implemented, the entire database at its most updated state would be backed up to an Excel file. Methods to add continuous database backup need to be implemented as well.

2.2.2 Deactivating Children/Activities

With the implementation of deactivating activities and children, this data should continually be stored in the database. Managers should have the capabilities to deactivate a student or activity, but not completely remove them from the database. There may be a case where a child is bouncing between being active and inactive and it is more convenient to activate/deactivate versus having to enroll the child altogether. On the HTML side, the managers should be able to see the children that were once enrolled in each class, and each class that was once offered at said studio.

2.2.3 Foreign Keys

The foreign keys are beneficial when displaying, associating, or corresponding data together. The logic needs to be updated to populate the foreign key columns when the data is being read in from the Excel sheets. After the logic and associations are made, the foreign keys will make it extremely easy to reference specific data, such as navigating between parent and child.

2.2.4 Deleting Activities with Students Enrolled

In order to prevent activities being deleted while students are still enrolled, there should be a check implemented to make sure there are currently no students in the class. The check would need to verify there are no students currently enrolled to a specific activityID, activityLevel, and location before allowing a delete.

3 FUTURE STATE

**3.1 Improvement**

3.1.1 Calculating Class Charges

For future work, the parent should receive an outstanding balance based off calculations of how many activities their children attended. The manager should have the ability to track how many of a specific activity occurred in a month. Based off this number, the system would then calculate how much the parent owed, taking into account the specific discounts if applicable.

3.1.2 Parent Payment

In the current state of the program, only the manager has the capabilities to make payments on the parent’s account. This payment is their fee for enrolling children in classes, tying in with the issue of calculating child charges. Parents can currently see the transaction history and account balance but cannot make payments on their own account. Along with making payments on their own account, parents should be able to use a credit card, PayPal, or other types of electronic transaction to complete their payment. Managers should also be able to complete an electronic transaction on the parent’s behalf.

3.1.3 Child Attendance

The instructors should be capable of taking attendance of each class they participate in teaching. Keeping track of attendance will also allow for the managers to properly bill each family for how many activities their children participate in in a given month. This is beneficial and needed for many reasons as a functioning business. This attendance report should also be saved and backed up in the database.

3.1.4 Security

More security needs to be implemented as far as multi-factor authentication and passwords are concerned. Right now, there is a password check to make sure the password has the bare minimum strength, however, MFA could further increase the security in the system. End-to-end encryption is another security feature that should be implemented, especially when dealing with the payment feature. If credit card and online transactions are implemented, compliance with Payment Card Industry Data Security Standard (PCI DSS) needs to be addressed. By adding these few security features, the system will overall be less prone to attacks and user’s data will be more secure.

3.1.5 Edit Activity

Currently, activity level, start/end time, and activity day is able to be edited. There should also be a feature to edit the studio location and activity name. The children composite key is comprised of childID, activityLevel, and activityID, which is currently prohibiting the change of the activityID, as it needs to remain a unique value. A check needs to be employed before allowing the user to edit the activity name.