**Object Oriented Programming (OOP)**

CCIT4023, 2015-2016 Semester 1

Project Progress Submission Form (**Final** Version)

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Class: | 01 | Group no.: | 01G1 | Date of submission: | 20/11/2015 |

|  |  |  |
| --- | --- | --- |
| Program Title: | Sports System E-Management (SSE) – Community Swimming Team | |
| Member 1 (SOUL): Eric Fung | | Member 3: Donny Mak |
| Member 2: Peter Cheung | | Member 4: Fred Fung |

1. **Items of submission and general evaluation:**

**Basic Evaluation (*Functional Requirements)***

*\* Description should be given (in remarks and/or later sections) in case of not 100% completed.*

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | **Self-Evaluation** | | | | | | | |
| **Items of Requirements** |  | **100%**  **(Completed)** | **75%** | **50%** | **25%** | **0%**  **(Not done)** |  | **Remarks** |
| **General (2.1.1)** |  |  |  |  |  |  |  |  |
| Login/Logout |  | O |  |  |  |  |  |  |
| File Load when login system |  | O |  |  |  |  |  |  |
| File Backup when logout system |  | O |  |  |  |  |  |  |
| Graphical User Interface |  |  | O |  |  |  |  |  |
| **Trainee (2.1.2)** |  |  |  |  |  |  |  |  |
| Self-Register a new account |  | O |  |  |  |  |  | Login page |
| Browse current training courses |  | O |  |  |  |  |  | Option 1 |
| Join training courses |  | O |  |  |  |  |  | Option 2 |
| Display training courses |  | O |  |  |  |  |  | Option 3 |
| Analyze/Display progress |  | O |  |  |  |  |  | Option 3 (Current/Completed) + 4 (display course target) |
| **Trainer (2.1.3)** |  |  |  |  |  |  |  |  |
| Search/List training courses |  | O |  |  |  |  |  |  |
| Track/Display records |  |  | O |  |  |  |  | Option 2 \*Cant search trainees |
| Evaluate/Grade performance |  | O |  |  |  |  |  | Option 3 |
| **Administrator (2.1.4)** |  |  |  |  |  |  |  |  |
| Register/Deregister an account |  | O |  |  |  |  |  |  |
| Search/List all users |  | O |  |  |  |  |  |  |
| Search/List all courses |  | O |  |  |  |  |  |  |
| Update/Edit all records |  |  |  | O |  |  |  | Edit course |
| Create/Remove training courses |  |  |  | O |  |  |  | Create course |
| **Default Settings / testing (2.1.5)** |  |  |  |  |  |  |  |  |
| Accounts (6) |  | O |  |  |  |  |  |  |
| Testing Record Files (4) |  |  |  |  |  | O |  |  |

1. **Details of Design and Implementation**

|  |  |
| --- | --- |
| **A) Program Title:** | **Sum Sum Eye Record System (Sports System E-Management)** |
| (login Page) (Trainee Menu)  (Admin Menu) | |

|  |  |  |
| --- | --- | --- |
| **B) Objectives, and Other General Ideas / Assumptions** |  |  |
| * The major purpose of this program is to create a specialized database system that would allow a simple and straightforward method to manage trainees and courses record for any sports training team. In our case we designed the software to suit the need for managing training courses from a community swimming team. * The objectives include:   1. Functional requirement – The system should be able to perform the basic requirements such as storing/updating trainees and courses record.   2. User-friendliness – The user interface should be easily accessible for any users, such as the trainees, trainers, and the system administrator regardless of their technological knowledge.   3. Dependability – The system should be able to account for as many records as possible without misinformation as it would be in use by a lot of users.   4. Database integrity – Password and sensitive information are encrypted with the MD5 algorithm, therefore ensuring security for the personal information of trainees. * Comparison with other similar software in the market (e.g. Sportslyzer, TrainingPeaks) has been made so that we could clearly identify the key user requirements. While there are a lot of software that provide similar function, we focus on providing a sleek and direct workspace with the least amount of bloat unlike other software. Our goal is to ensure that the program is most optimized the greatest amount of system while accommodating to all possible users. | | |

|  |  |  |
| --- | --- | --- |
| **C) Design Details** (Include ***UML class diagrams of major classes & special algorithms if any***) |  |  |
|  | | |

|  |  |  |
| --- | --- | --- |
| **D) Implementation / Production** (Major production issues; NO detailed code explanation) |  |  |
| * User input validation – One of the major issues during development is that the system cannot process invalid input in the menu selection and the registration section. However it was resolved by implementing methods that would validate the user input before putting it into the system. * Encryption – During production, we have debated on how we are going to implement encryption on sensitive information in the database. In the end, we have agreed on using the MD5 encryption to protect the password in the database, it is validated by encrypting the user input and comparing it with the encrypted password in the database. | | |

1. **Development Environment, Evaluation and Testing**

*Development Platforms and Tools:*

Java SE

Notepad ++

Draw.io

Photoshop

*Errors and Warnings:*

|  |
| --- |
| **Major Known Errors** (not solved yet), with Descriptions and Remarks |
| Compile-time errors (and warnings), if any:  n/a |
| Run-time / Semantic errors during execution, if any:  n/a |

1. **List of Additional Features or Techniques Applied, if any**

\*MD5 Encryption – Use of MD5 encryption to protect password in database

\*Record Search – A for-loop is put into place so search can be made by comparing the queried record and the records in the database in a loop

**~** END ~

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Functional Req. | | | Programming Req. | | | Development & Doc | | | Others | | Overall | |
|  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |