**Class Simulation**

Detailed design document



**EvaluateFresherTraining**

**Start date of project: 25/3/2020**

**Duration: will be long**

**Topic: WPF mock project**

**Authorized by: ThieuBQ**

**In charged by: ThieuBQ**

**Version: alpha demo**

This is a mock project in order to practice coding C# under the instruction of Mr. Vu Thien An as the mentor. The project is initialized by the order of PM.

**Table of content:**

1. **Introduction**
   1. **Problem definition**
   2. **Purpose**
   3. **Scope**
   4. **Overview**
2. **Design, function, how to implement function**
   1. **Design**
   2. **Function**
   3. **How to implement function**
3. **Introduction**

This Detailed Design Document for Class Simulation project sponsored by FPT Software provides the description of the windows application. This document also mostly follows the requirements of the order of team leader and PM; however, due to the practice purposes, some requirements has changed in order to gain more programming skills. The app has done completely with PWF written in C# and the IDE used to program is Microsoft visual studio.

The final result is a windows application.

* 1. **Problem definition**

The goal in this project is to display information of students in a class with a given number of students.

* 1. **Purpose**

The scope of this DDD is to provide information about the design procedure of the program. The class diagram, user interface design will be elucidated in the scope of this document. The intended audience is the internal fresher team, PM and team leader, thus it is aimed that this document to be a description of how to build up the program, its functions and how they work.

* 1. **Definitions, Acronyms and Abbreviations**

|  |  |
| --- | --- |
| GUI | Graphical User Interface |
| WPF | Windows Presentation Foundation, is graphical subsystem based on XAML to develop user interfaces in Windows-based applications |
| DDD | Detailed design document |
| XAML | Extensible Application Markup Language |
|  |  |
|  |  |

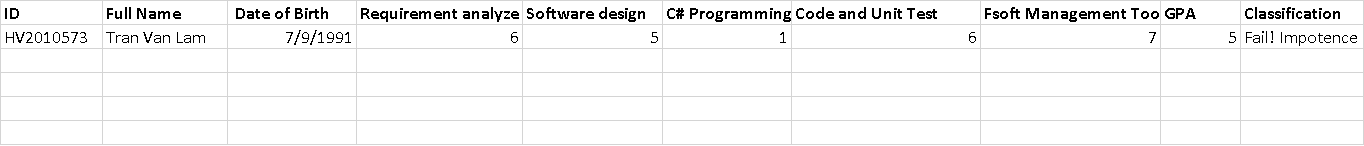
* 1. **Overview**

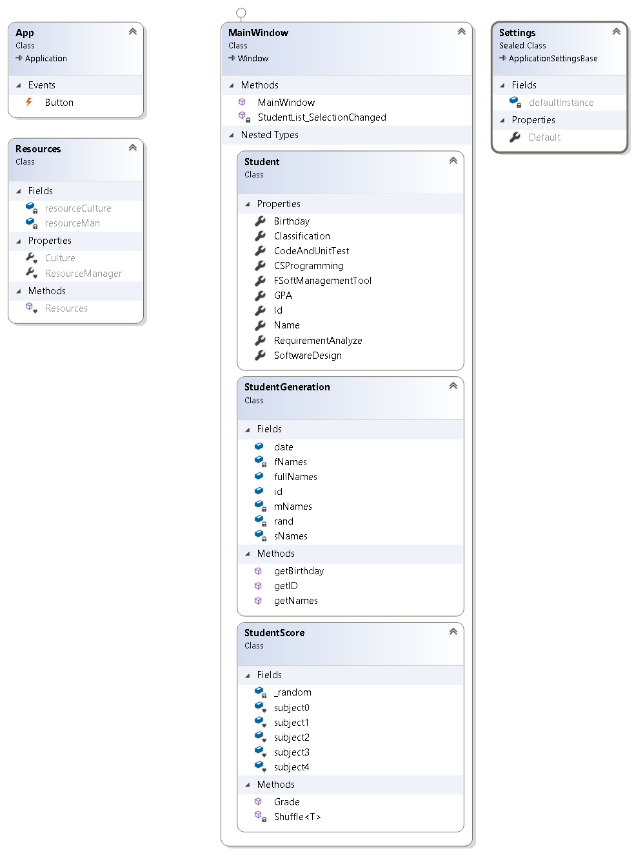
This DDD is divided into 3 sections in order to provide a complete and understandable perception about the window app to the target readers. First section is mostly about the scope and purpose of the document. This section also includes the definition of the problem that is intended to be solved. In the second part, window application, design, functions and the way how to implement functions sophisticatedly.

1. **Design, function, and how to implement function**
   1. **Design**

This is a simulation of a school class for the purposes of practice. In this project, freshmen tried to make up a PWF program to find out more about PWF. Our goal here is to build up a demo classroom, where all the information about the students in the class is displayed on a screen. There is a list of all the students, including their names, ID code, date-of-birth, and grades for 5 subjects: Requirement Analysis; Software Design; C# Programming; Code and Unit Tests; and F-soft Management Tools; with their GPA and their classification.

User interface prototype design:





**Class Diagram design:**

Explain more detail:

There are 4 namespaces:

+ App

+ Resources

+ MainWindow

+ Settings.

The figure on the right shows the class diagram clearly and detailingly.

* 1. **Function**

**-** A student with his/her information

- A list of student

* 1. **How to implement function**

**a) Class Student:**

This class is the main class contains all properties of a student:

Create a class namely Student contain:

* Name
* Id
* RequirementAnalyze
* SoftwareDesign
* FSoftManagementTool
* GPA
* Classification

All the properties of class Student above will be set a approriate data type and encapsulation.

1. **Create the student**

We have to create Name, Id, RequirementAnalyze SoftwareDesign, FSoftManagementTool following the requirements:

* **Create Name:**

**Phase 1:**

**Gernerate random vietnamese name without special charaters:**

From <https://www.fantasynamegenerators.com/vietnamese_names.php>, Generate Vietnamese names on this web tool to create our own names.

Example:

“An Kim Mai

Đàm Hương Trầm

Võ Vân Hương

Trương Thụy Nương

Nguyễn Ngọc Ðào

Nguyễn Quỳnh Trang

Tạ Minh Ngọc

Mạch Ngọc Diệp

Lạc Lan Chi

Vũ Ngọc Huệ”

Standardize all the names by eliminating all the special characters, like this list:

“An Kim Mai

Dam Huong Tram

Vo Van Huong

Truong Thuy Nuong

Nguyen Ngoc Ðao

Nguyen Quynh Trang

Ta Minh Ngoc

Mach Ngoc Diep

Lac Lan Chi

Vu Ngoc Hue”

Extract from the internet, we have 3 sets:

74 surname:

['An', 'Bach', 'Banh', 'Bui', 'Cao', 'Chau', 'Chu', 'Chung', 'Dam', 'Dang', 'Dao', 'Diep', 'Dinh', 'Do', 'Doan', 'Duong', 'Duu', 'Giang', 'Ha', 'Han', 'Ho', 'Hoang', 'Huynh', 'Kieu', 'Kim', 'La', 'Lac', 'Lam', 'Le', 'Lieu', 'Luc', 'Luong', 'Luu', 'Ly', 'Ma', 'Mac', 'Mach', 'Mai', 'Nghiem', 'Ngo', 'Ngu', 'Nguyen', 'Pham', 'Phan', 'Phi', 'Pho', 'Phung', 'Quach', 'Quang', 'Quyen', 'Ta', 'Thach', 'Thai', 'Than', 'Thao', 'Thi', 'Thuy', 'Tieu', 'To', 'Ton', 'Tong', 'Tram', 'Tran', 'Trang', 'Trieu', 'Trinh', 'Truong', 'Uat', 'Uc', 'Van', 'Vinh', 'Vo', 'Vu', 'Vuong']

115 middle name:

['Ai', 'An', 'Anh', 'Bach', 'Bao', 'Bich', 'Binh', 'Boi', 'Cam', 'Cao', 'Cat', 'Chi', 'Chinh', 'Cong', 'Da', 'Diem', 'Dieu', 'Duy', 'Duyen', 'Gia', 'Giang', 'Ha', 'Hai', 'Hao', 'Hiep', 'Hieu', 'Hoa', 'Hoai', 'Hoan', 'Hoang', 'Hong', 'Hue', 'Huong', 'Huu', 'Huy', 'Huyen', 'Kha', 'Khac', 'Khai', 'Khanh', 'Kien', 'Kieu', 'Kim', 'Lam', 'Lan', 'Le', 'Lien', 'Linh', 'Loan', 'Long', 'Luong', 'Mai', 'Manh', 'Minh', 'Mong', 'My', 'Nam', 'Ngan', 'Nghi', 'Ngoc', 'Nguyen', 'Nguyet', 'Nha', 'Nhat', 'Nhu', 'Phi', 'Phuc', 'Phuoc', 'Phuong', 'Quang', 'Quoc', 'Quy', 'Quyet', 'Quynh', 'Son', 'Suong', 'Sy', 'Tan', 'Thach', 'Thai', 'Thanh', 'Thao', 'The', 'Thi', 'Thien', 'Thieu', 'Thinh', 'Thu', 'Thuan', 'Thuc', 'Thuy', 'Tien', 'Tinh', 'Trang', 'Tri', 'Trong', 'Truc', 'Trung', 'Truong', 'Tu', 'Tuan', 'Tung', 'Tuong', 'Tuyet', 'Uy', 'Uyen', 'Van', 'Viet', 'Vinh', 'Vu', 'Vuong', 'Vy', 'Xuan', 'Y', 'Yen']

161 first name:

['An', 'Anh', 'Bach', 'Bang', 'Bao', 'Bich', 'Binh', 'Buu', 'Ca', 'Cam', 'Can', 'Canh', 'Chau', 'Chi', 'Chieu', 'Cuong', 'Dan', 'Danh', 'Di', 'Diem', 'Du', 'Due', 'Dung', 'Duong', 'Duy', 'Gia', 'Giang', 'Ha', 'Hai', 'Han', 'Hang', 'Hanh', 'Hao', 'Hau', 'Hien', 'Hiep', 'Hieu', 'Hoa', 'Hoan', 'Hoang', 'Hong', 'Hop', 'Hung', 'Huong', 'Huu', 'Huy', 'Huyen', 'Khai', 'Khang', 'Khanh', 'Khiem', 'Khoa', 'Khoi', 'Khue', 'Khuong', 'Khuyen', 'Kien', 'Kiet', 'Kieu', 'Lam', 'Lan', 'Lap', 'Le', 'Lien', 'Lieu', 'Linh', 'Loan', 'Loc', 'Long', 'Luan', 'Luong', 'Ly', 'Mai', 'Manh', 'Minh', 'My', 'Nam', 'Nga', 'Ngan', 'Nghi', 'Nghia', 'Ngoc', 'Ngon', 'Nguyen', 'Nguyet', 'Nha', 'Nhan', 'Nhat', 'Nhi', 'Nhien', 'Nhu', 'Nhung', 'Ninh', 'Nuong', 'Oanh', 'Phat', 'Phi', 'Phong', 'Phu', 'Phuc', 'Phung', 'Phuong', 'Quan', 'Quang', 'Quyen', 'Quyet', 'Quynh', 'Sa', 'San', 'Son', 'Suong', 'Tai', 'Tam', 'Tan', 'Thach', 'Thai', 'Tham', 'Thang', 'Thanh', 'Thao', 'Thien', 'Thinh', 'Tho', 'Thoa', 'Thong', 'Thu', 'Thuan', 'Thuy', 'Tien', 'Tin', 'Toan', 'Tra', 'Tram', 'Trang', 'Tri', 'Trieu', 'Trinh', 'Trong', 'Tru', 'Truc', 'Trung', 'Truong', 'Tu', 'Tuan', 'Tung', 'Tuong', 'Tuyen', 'Tuyet', 'Uy', 'Uyen', 'Van', 'Vi', 'Vien', 'Viet', 'Vinh', 'Vo', 'Vu', 'Vy', 'Xuan', 'Y', 'Yen']

Hence the maximum number of distinct names: 161 x 116 x 74 = 1382024 (name).

**Phase 2:**

From the 3 sets above we match the full name: surname + middle name + first name.

Get random element of each then merge each to a string of full name.

* **Generate random ID:**

The format of ID = HV + interval [1000000,9999999], all ID have to be unique, must have no duplicate. Create a random in the interval when HV plus with it, it will turn to be a string.

* **Generate random birthday:**

From 2-1-1991 to 31-12-1991 (the format is ‘mm-DD-yyyy’), we generate a random set of birthday before matching it with each student for all student one by one.

* **Generate score for all 5 subject:**

There requirement for each subject: 30k students got score in interval [8,10], 130k student got from [5,7], 60k [3,4], 20k [1,2], and 10k students drop the examination get zero.

Using Linq to get a loop repeat.

After finishing this phase we shuffle all elements in each array.

* **Count GPA:**

GPA = (Subject 1 + subject 2 + subject 3 + subject 4 + subject 5) /5

* **Get classification:**

First, a student gets a zero, no matter which subject it is, will get an impotent score and will get fail result. Then we set like this:

If GPA [1 – 3) will get “Weak”

If GPA [3,5) will get “Poor”

If GPA [5,6.5) will get “Average”

If GPA [6.5,8) will get “Good”

If GPA >= 8 will get “Excellent”