**TRƯỜNG ĐẠI HỌC SƯ PHẠM KỸ THUẬT TP. HỒ CHÍ MINH**

**KHOA ĐÀO TẠO CHẤT LƯỢNG CAO**

**NGÀNH CÔNG NGHỆ THÔNG TIN**

**---o0o---**



**BÁO CÁO ĐỒ ÁN 1**

**Phần mềm nhập điểm tổng kết và phân cụm sinh viên**

**SVTH: ĐOÀN QUỐC HÙNG** **17110154**

**NGUYỄN HUỲNH ANH TRỰC** **17110247**

**VÕ TẤN NGUYÊN** **17110194**

**GVPT: TỪ TUYẾT HỒNG**

**Tp. Hồ Chí Minh, tháng 11 năm 2019**

MỤC LỤC

[LỜI CẢM ƠN 5](#_Toc24387892)

[MÔ TẢ PROJECT: 1](#_Toc24387893)

[1. Lý do xây dựng phần mềm: 1](#_Toc24387894)

[2. Dữ liệu thông tin đầu vào: 1](#_Toc24387895)

[3. Các tình huống sử dụng: 1](#_Toc24387896)

[4. Phương pháp và ngôn ngữ: 2](#_Toc24387897)

[5. Giao diện dự kiến. 2](#_Toc24387898)

[II. Phân công công việc: 5](#_Toc24387899)

[III. Thiết kế: 7](#_Toc24387900)

[1. Thuật toán: 7](#_Toc24387901)

[2. Thiết kế lớp: 10](#_Toc24387902)

[3. Thiết kế cơ sở dữ liệu 38](#_Toc24387903)

[4. Mô tả các Field trong 1 table 39](#_Toc24387904)

[**a.** **Bảng Classes.** 39](#_Toc24387905)

[**b.** **Bảng Students.** 39](#_Toc24387906)

[**c.** **Bảng Grades.** 40](#_Toc24387907)

[**d.** **Bảng Subjects.** 40](#_Toc24387908)

[**e.** **Bảng Professors.** 41](#_Toc24387909)

[**f.** **Bảng Departments.** 41](#_Toc24387910)

[5. Thiết kế giao diện 42](#_Toc24387911)

[IV. Cài đặt và kiểm thử : 47](#_Toc24387912)

[V. KẾT LUẬN: 49](#_Toc24387913)

[1. Đánh giá chung: 49](#_Toc24387914)

[2. Các khó khăn gặp phải : 49](#_Toc24387915)

[3. Ưu điểm: 49](#_Toc24387916)

[4. Nhược điểm: 50](#_Toc24387917)

[5. Hướng phát triển: 50](#_Toc24387918)

[VI. TÀI LIỆU KHAM KHẢO: 51](#_Toc24387919)

**Danh mục hình**

[Hình 1: Giao diện Login 3](#_Toc24387646)

[Hình 2: Giao Diện DashBoard chứa thông tin về giảng viên 3](#_Toc24387647)

[Hình 3: Giao diện quản lý thông tin sinh viên 4](#_Toc24387648)

[Hình 4: Giao diện quản lý điểm 4](#_Toc24387649)

[Hình 5: Giao diện cho việc thực hiện thuật toán phân cụm K-mean 5](#_Toc24387650)

[Hình 6: Ví dụ thuật toán K-mean 8](#_Toc24387651)

[Hình 7: Sơ đồ ERD 38](#_Toc24387652)

**Danh mục bảng**

[Bảng 1: Bảng phân công công việc 5](#_Toc24387657)

[Bảng 2: Bảng thiết kế lớp 10](#_Toc24387658)

[Bảng 3: Bảng thiết kế phương thức 12](#_Toc24387659)

[Bảng 4: Thiết kế bảng cơ sở dữ liệu 38](#_Toc24387660)

[Bảng 5: Mô tả các field trong bảng Classes 39](#_Toc24387661)

[Bảng 6: Mô tả các field trong bảng Students 40](#_Toc24387662)

[Bảng 7: Mô tả các field trong bảng Grades 40](#_Toc24387663)

[Bảng 8: Mô tả các field trong bảng Subjects 40](#_Toc24387664)

[Bảng 9: Mô tả các field trong bảng Professors 41](#_Toc24387665)

[Bảng 10: Mô tả các field trong bảng Departments 41](#_Toc24387666)

[Bảng 11: Bảng thiết kế giao diện 42](#_Toc24387667)

[Bảng 12: Bảng cài đặt và kiểm thử 47](#_Toc24387668)

# LỜI CẢM ƠN

Trên thực tế không có sự thành công nào mà không gắn liền với những sự hỗ trợ, giúp đỡ dù ít hay nhiều, dù trực tiếp hay gián tiếp của người khác. Nay em xin được phép gửi lời cảm ơn chân thành này đến cô Từ Tuyết Hồng, người đã trực tiếp hỗ trợ nhóm em trong quá trình định hướng chọn đề tài, hướng dẫn, nhận xét và gợi ý đường hướng thực hiện đồ án. Nếu không có những lời hướng dẫn của cô thì em nghĩ rằng đồ án này sẽ khó có thể hoàn thiện và hoàn thành đúng thời hạn được. Một lần nữa, em xin cảm ơn cô.

Em cũng xin chân thành cảm ơn các quý thầy cô trong khoa Đào Tạo Chất Lượng Cao đã giúp đỡ hỗ trợ kiến thức cũng như giải đáp thắc mắc của em. Cùng với đó, em xin được gửi cảm ơn đến các bạn cùng khóa đã cung cấp nhiều thông tin và kiến thức hữu ích giúp cho em hoàn thiện đề tài hơn.

Bài thu hoạch được thực hiện trong khoảng thời gian gần 8 tuần. Khoảng thời gian có hạn, cùng với kiến thức còn hạn chế và còn nhiều bỡ ngỡ khác do đó thiếu sót là điều không thể tránh khỏi nên em rất mong nhận được những ý kiến đóng góp quý báo của các quý Thầy Cô để kiến thức của em được hoàn thiện hơn sau này. Em xin chân thành cảm ơn.

|  |  |
| --- | --- |
|  |  |

# 

# MÔ TẢ PROJECT:

## *Lý do xây dựng phần mềm:*

Trong thời đại ngày nay ứng dụng công nghệ thông tin trở thành 1 phần hiển nhiên do đó đồ án lần này chúng em đã có ý định xây dưng một phần mềm có thể ứng dụng vào trong lĩnh vực giáo dục, giúp giáo viên trong quy mô một ngôi trường, có thể nhập điểm cuối kỳ cho các lớp mà giáo viên quản lý, cũng như là khả năng phân tích và phân loại sinh viên dựa trên điểm số một cách thông minh hơn qua thuật toán phân cụm K-mean.

## *Dữ liệu thông tin đầu vào:*

* + Xây dựng giả định một cơ sở dữ liệu gồm danh sách 600 sinh viên và 20 môn học, gần 20 giảng viên và 5 khoa.
  + Sơ lược điểm tổng kết bao gồm 12000 điểm ( có thể được đọc từ 2 file text đính kèm trong project )

## *Các tình huống sử dụng:*

Tạo ra được một phần mềm Quản lí sinh viên gần gũi đơn giản giúp cho giảng viên đơn giản hóa trong việc nhập liệu điểm cuối kỳ và phân tích kết quả điểm để phân cụm sinh viên dễ dàng.

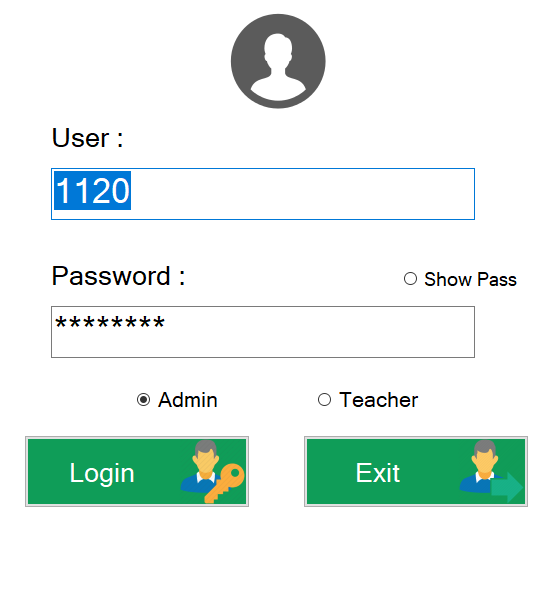
Có 2 đối tượng chính của ứng dụng:

-Người giảng viên dùng ứng dụng để nhập điểm cuối kỳ cho sinh viên, Giảng viên chỉ có thể truy cập đến sinh viên thuộc những lớp mình đứng giảng. Giảng viên có thể chạy thuật toán k-mean để phân tích tình hình học tập của các sinh viên trong môn ( trường hợp cụ thể trong bài giả định là 1 đến 2 môn) mà mình giảng và xuất kết quả chi tiết các sinh việc thuộc cụm nào ra file excel hoặc txt, nhưng không có quyền chạy ở cấp độ admin, người có thể phân theo Khoa, phạm vi toàn bộ môn học có trong trường.( trường hợp cụ thể trong bài đang là 20 môn) và giảng viên cũng không thể nhập điểm từ file ( chức năng chỉ có ở admin )

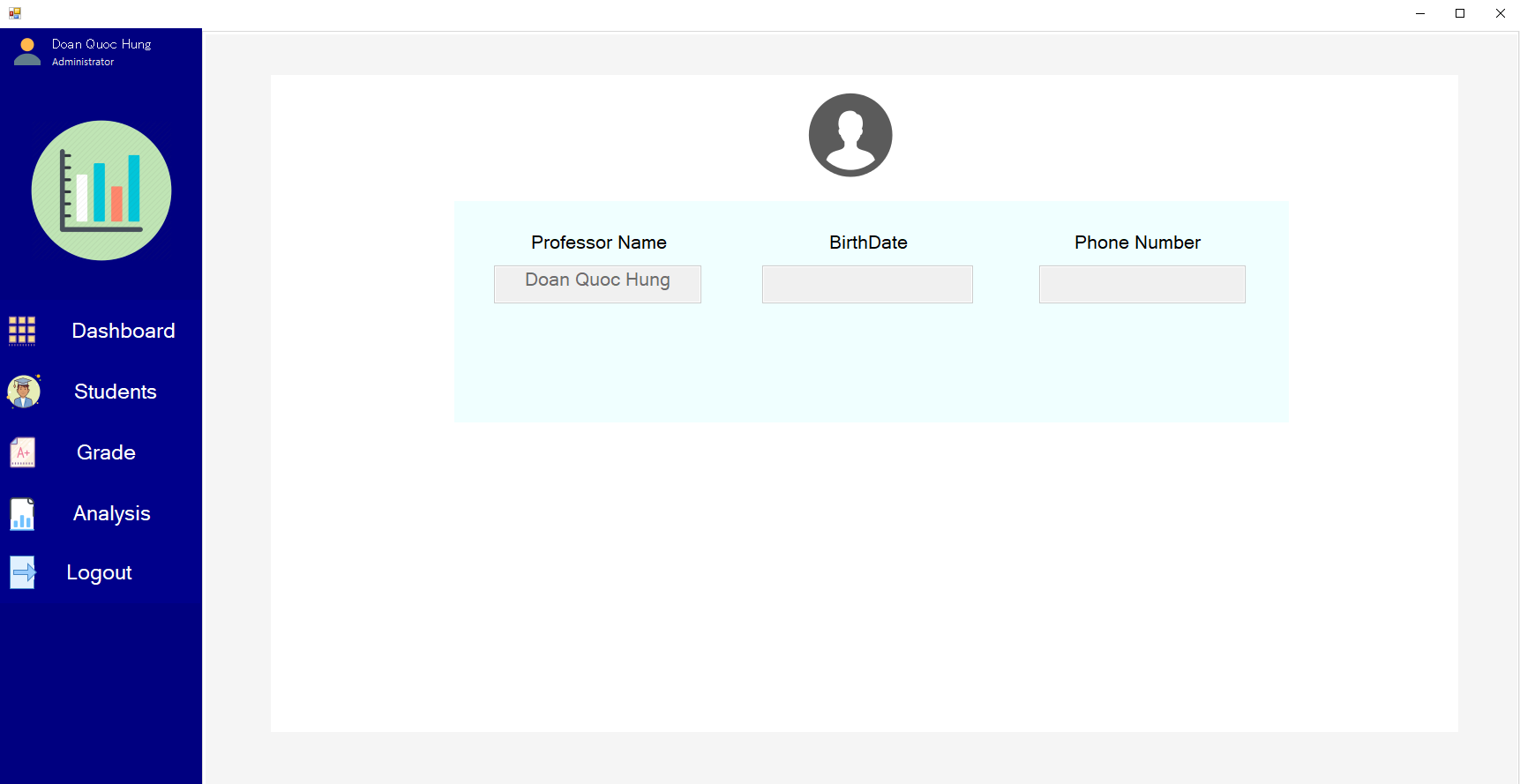
-Người thứ 2 là admin, người có toàn quyền và có thể truy cập, thay đổi ở tất cả các môn học, các lớp học, có thể phân sinh viên theo từng khoa và thi hành việc chạy thuật toán phân cụm các sinh viên dự trên điểm số các môn học ( trường hợp cụ thể trong bài đang là 20 môn) của những sinh viên đó, xuất ra kết quả ra file bên ngoài, và có thể nhập điểm từ 1 file TXT bên ngoài theo cách tổ chức dữ liệu được quy định ( mã sinh viên mã lớp điểm)

## *Phương pháp và ngôn ngữ:*

* Về phương pháp: trong quá trình tìm hiểu đề tài thì em cảm thấy K-mean là một thuật toán có tính hiệu quả và thực tiễn cao trong việc giải các bàn toán phân cụm, nên em đã áp dụng thuật toán này để giải quyết việc phân cụm sinh viên qua những tương đồng về điểm cuối kỳ.
* Về ngôn ngữ: ngôn ngữ C# trên nền tảng Microsoft Framework với công nghệ Microsoft Window Form và Entity Framework.
  1. ***Giao diện dự kiến.***



Hình 1: Giao diện Login



Hình 2: Giao Diện DashBoard chứa thông tin về giảng viên



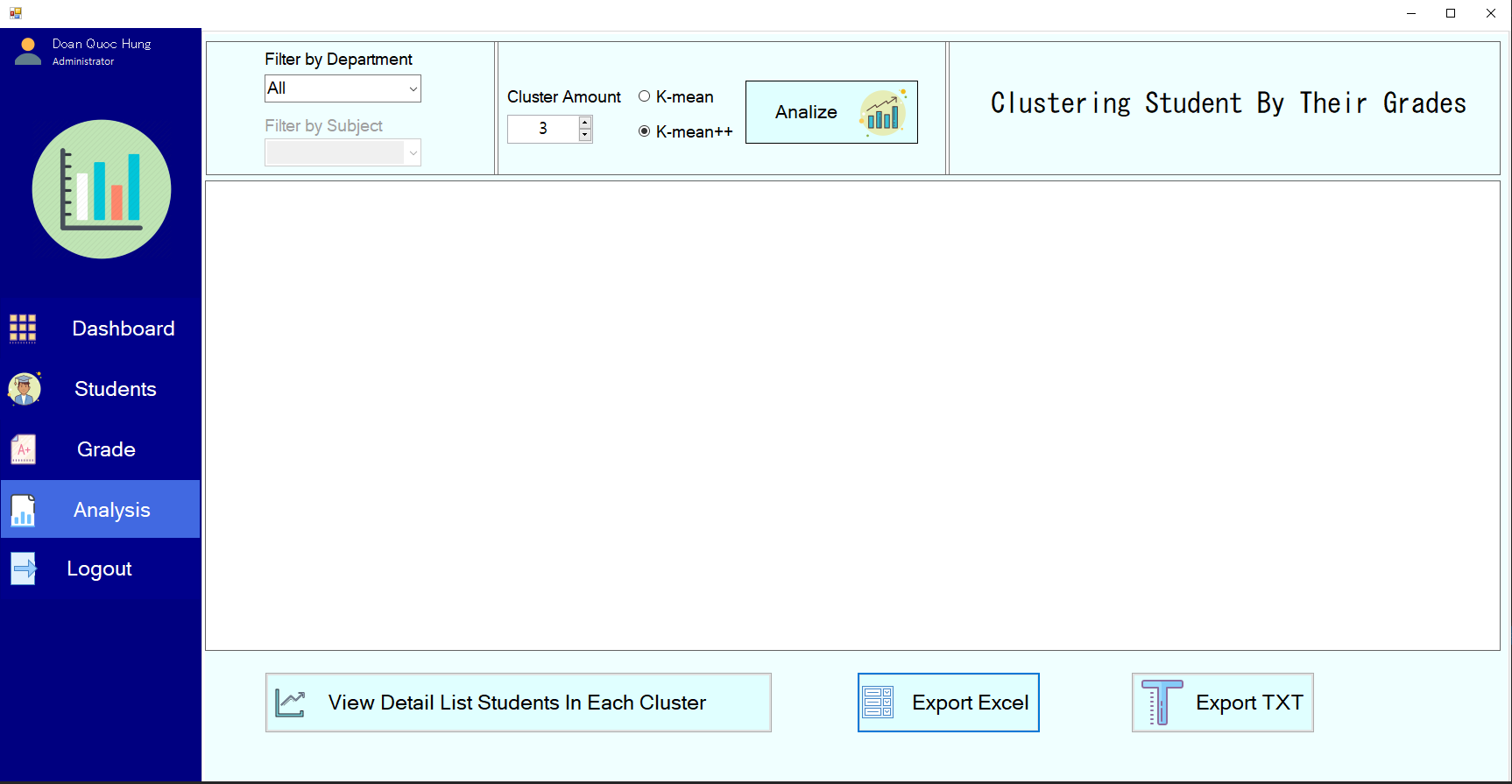
Hình 3: Giao diện quản lý thông tin sinh viên

Giao diện quản lý thông tin sinh viên, chứa thông tin các sinh viên học ở các lớp thuộc các môn mà giảng viên đang phụ trách, với quyền admin thì có thể coi được tất cả sinh viên trong trường.



Hình 4: Giao diện quản lý điểm

Giao diện quản lý điểm ở các lớp mà giảng viên đang dạy, có các chức năng sửa điểm, nhập điểm từ file txt và search 1 thông tin về sinh viên và điểm môn học cụ thể theo tên sinh viên hoặc mã sinh viên. Giảng viên chỉ được truy cập đến điểm những lớp mà mình phụ trách, còn admin thì có thể xem được tất cả các môn và các lớp.



Hình 5: Giao diện cho việc thực hiện thuật toán phân cụm K-mean

Phần giao diện cho việc thực hiện thuật toán phân cụm K-mean, được chia làm 3 phần lớn là những mảng ngang. Mảng ở trên cùng dùng để thiết lập bộ lọc cho dữ liệu đầu vào của Kmean( có thể lọc theo Khoa với tài khoản quyền Admin và lọc theo môn với tài khoản quyền Professor ), cũng như số lượng cluster mà người dùng muốn phân cụm. Mảng ngang ở giữa chứa không gian hiển thi của những biểu đồ thể hiện kết quả lần chạy K-mean gần nhất. Mảng cuối cùng cho phép xem kết quả chi tiết của thuật toán và cũng như khả năng xuất kết quả ra những file bên ngoài như excel và text.

1. **Phân công công việc:**

Bảng 1: Bảng phân công công việc

|  |  |  |  |
| --- | --- | --- | --- |
| TT | Tên Sinh viên | Công việc | Phần trăm đóng góp |
| 1 | Đoàn Quốc Hùng | Thiết kế cơ sở dữ liệu | 40% |
| Xây dựng thuật toán K-mean |
| Phương pháp khởi tạo K-mean++ |
| Xây dựng biểu đồ |
| Hàm hiển thị thông tin chi tiết của lần chạy thuật toán gần nhất lên form Detail Form |
| 2 | Nguyễn Huỳnh Anh Trực | Xây dựng các chức năng hiển thị trong TabPage Quản lý thông tin điểm | 30% |
| Xuất ra file Excel kết quả của thuật toán |
| Viết hàm tự sinh để tạo cơ sở dữ liệu ban đầu |
| Hàm đọc điểm từ 1 file TXT |
| 3 | Võ Tấn Nguyên | Xây dựng chức năng đăng nhập | 30% |
| Xây dựng giao diện cho ứng dụng |
| Xuất ra file TXT kết quả của thuật toán |
| Xây dựng các chức năng hiển thị trong TabPage Quản lý thông tin sinh viên |

# Thiết kế:

## *Thuật toán:*

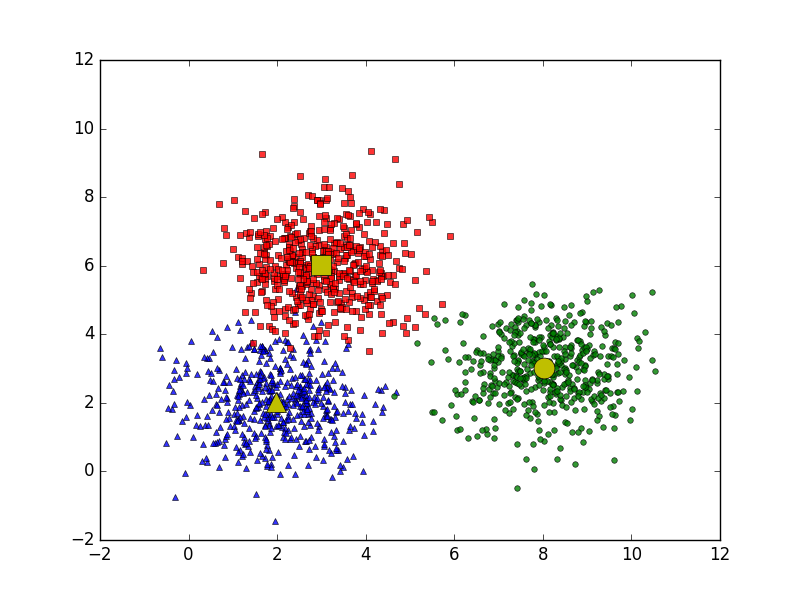
Thuật toán K-mean là một thuật toán un[supervised learning](https://machinelearningcoban.com/2016/12/27/categories/#unsupervised-learning-hoc-khong-giam-sat) có nghĩa là trong thuật toán K-means Clustering, chúng ta không biết nhãn (label) của từng điểm dữ liệu. Mục đích của giải thuật là làm thể nào để phân dữ liệu thành các cụm (cluster) khác nhau sao cho dữ liệu trong cùng một cụm có tính chất giống nhau.

Một cách đơn giản để mô phỏng bài toán này là biểu diễn qua cái nhìn hình học. Các dữ liệu có thể coi là các điểm trong không gian và khoảng cách giữa các điểm có thể được coi là thông số mức độ giống nhau của chúng. 2 điểm càng gần nhau thì chúng càng giống nhau.

Hình dưới đây là mô phỏng 3 cụm điểm (mỗi màu là mỗi cụm điểm ) trong một không gian 2 chiều, ở đây số chiều của không gian tương ứng với số thuộc tính của các đối tượng mà ta xét.

Với cách nhìn hình học như vậy, ta có thể viết lại bài toán dưới dạng hình thức như sau:

* Dữ liệu: tập dữ liệu gồm n điểm dữ liệu có d chiều
* Nhiệm vụ: phân tập dữ liệu ra làm k cụm các dữ liệu tương tự nhau. Đương nhiên là k ≤ n.



Hình 6: Ví dụ thuật toán K-mean

Tóm tắt cách thuật toán hoạt động:

**Bước 1:** Chọn K điểm bất kỳ làm các center ban đầu.

**Bước 2:** Phân mỗi điểm dữ liệu vào cluster có center gần nó nhất.

**Bước 3:** Nếu việc gán dữ liệu vào từng cluster ở bước 2 không thay đổi so với vòng lặp trước nó thì ta dừng thuật toán.

**Bước 4:** Cập nhật center cho từng cluster bằng cách lấy trung bình cộng của tất các điểm dữ liệu đã được gán vào cluster đó sau bước 2.

**Bước 5:** Quay lại bước 2.

Hạn chế của K-mean:

* + Cần biết trước số nhóm, có nhóm sai có thể dẫn đến kết quả sai. Vấn đề này có thể [cải thiện](https://en.wikipedia.org/wiki/Determining_the_number_of_clusters_in_a_data_set) dựa vào một số phương pháp sau: phương pháp [Elbow](https://en.wikipedia.org/wiki/Determining_the_number_of_clusters_in_a_data_set#The_elbow_method) hoặc phương pháp [X-means](https://en.wikipedia.org/wiki/Determining_the_number_of_clusters_in_a_data_set#X-means_clustering).
  + Khởi tạo sẽ ảnh hưởng tới chất lượng kết quả, một khởi tạo không tốt có thể dẫn đến kết quả sai. Cách khắc phục là sử dụng phương pháp khởi tạo K-mean++.

Như đã nhắc ở trên về Kmean++ vậy thì K-mean là gì ?

K-mean++ tức là thay vì sử dụng cách random ngẫu nhiên để tạo ra 1 tập các Central Point ban đầu cho các cluster thì K-mean++ là phương pháp khởi tạo có chủ đích khi chỉ khởi tạo central point đầu tiên bằng cách random sau đó sẽ chọn ra điểm có khả năng cao là điểm thuộc 1 cluster khác biệt là central point tiếp theo, đến khi nào đủ số lượng central point cần thiết thì dừng lại và tiếp tục chạy K-mean

Tóm tắt cách k-mean ++ hoạt động:

**Bước 1:** Khởi tạo Central Point đầu tiên bằng cách random.

**Bước 2:** Kiểm tra nếu đã đủ số lượng thì thoát khỏi vòng lặp và tiếp tục quá trình chạy K-mean.

**Bước 3:**Tính khoảng cách của từng điểm đến central point gần nó nhất.

**Bước 4:** Điểm có khoảng cách (đến central point gần nhất) xa nhất trong các điểm sẽ được chọn làm điểm central point tiếp theo.

**Bước 5:** Quay lại bước 2.

Ở bài toán phân cụm sinh viên mà tụi em làm thì mỗi một sinh viên sẽ được xem xét như là 1 điểm trong hệ không gian có n chiều ( n là số môn học, cụ thể hơn trong ứng dụng là từ 1 tới 20 môn ), từ đó áp dụng Kmean hoặc Kmean++ để phân cụm các sinh viên có tính chất về các điểm số gần giống nhau ( tức là khoảng cách dựa trên công thức tính khoảng cách Euclid trong không gian) tức những điểm chung trông 1 cluster sẽ gần nhau sẽ có khoảng cách gần tâm của cụm hơn những điểm khác.

## *Thiết kế lớp:*

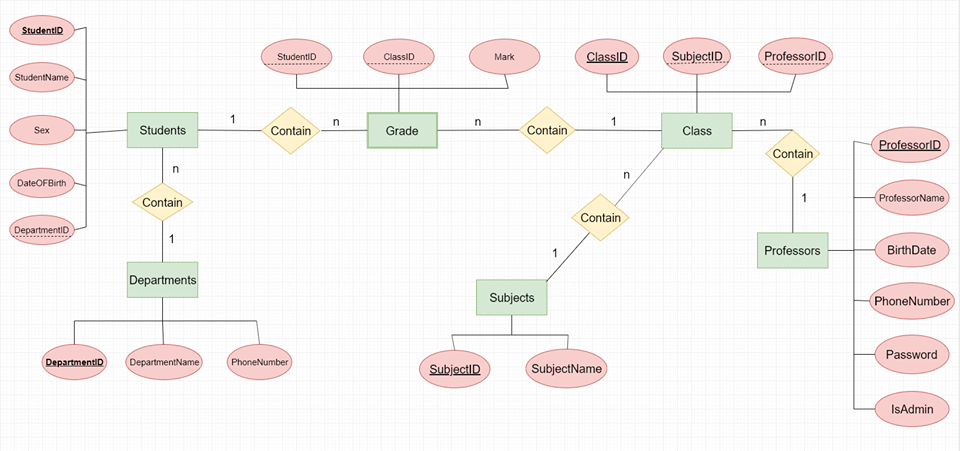
Bảng 2: Bảng thiết kế lớp

|  |  |  |  |
| --- | --- | --- | --- |
| STT | Tên lớp | Mục đích | Tên sinh viên phụ trách |
| 1 | UniversityContext | Cấu hình các thuộc tính và các ràng buộc khóa ngoại của các model khi ánh xạ xuống cơ sở dữ liệu class thông qua Entity Framework. | Đoàn Quốc Hùng |
| Võ Tấn Nguyên |
| Nguyễn Huỳnh Anh Trực |
| 2 | UniversityDBInitializer | Cấu hình việc khởi tạo Database, các dữ liệu đầu vào mặc định | Đoàn Quốc Hùng |
| Võ Tấn Nguyên |
| Nguyễn Huỳnh Anh Trực |
| 5 | MyGrouping<TKey, TElement> | Thiết lập một loại dữ liệu mới dựa trên 2 interface List<TElement> và IGrouping<TKey, TElement> trong framework để có thể lấy được những thuộc tính của 2 interface này là phân theo Group và khả năng khởi tạo 1 đối tượng mới cho 1 Group để sử dụng về sau trong bài. | Đoàn Quốc Hùng |
| 6 | Class | Thể hiện cho đối tượng lớp học trong cơ sở dữ liệu | Võ Tấn Nguyên |
| 7 | Department | Thể hiện cho đối tượng khoa trong cơ sở dữ liệu | Võ Tấn Nguyên |
| 8 | Grade | Thể hiện cho đối tượng điểm số trong cơ sở dữ liệu | Đoàn Quốc Hùng |
| 9 | Professor | Thể hiện cho đối tượng giảng viên trong cơ sở dữ liệu | Đoàn Quốc Hùng |
| 10 | Student | Thể hiện cho đối tượng sinh viên trong cơ sở dữ liệu | Đoàn Quốc Hùng |
| 11 | Subject | Thể hiện cho đối tượng môn học trong cơ sở dữ liệu | Võ Tấn Nguyên |
| 12 | BLStudent | Tầng nghiệp vụ chứa các hàm xử lý dữ liệu bên dưới tabpage Student | Nguyễn Huỳnh Anh Trực |
| 13 | BLGrade | Tầng nghiệp vụ chứa các hàm xử lý dữ liệu bên dưới tabpage Grade | Nguyễn Huỳnh Anh Trực |
| 14 | BLAnalysis | Tầng nghiệp vụ chứa các hàm xử lý dữ liệu bên dưới tabpage Analysis | Đoàn Quốc Hùng |
| 15 | Main\_Form | Là tầng giao diện chứa tất cả các chức năng của chương tình | Đoàn Quốc Hùng, Nguyễn Huỳnh Anh Trực, Võ Tấn Nguyên |
| 16 | Login\_Form | Là tầng giao diện của chức năng đăng nhập | Võ Tấn Nguyên, Nguyễn Huỳnh Anh Trực |
| 17 | Detail\_Form | Là tầng giao diện cho form hiển thị thông tin đầy đủ của lần phân cụm thành công gần nhất | Đoàn Quốc Hùng |

Bảng 3: Bảng thiết kế phương thức

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| *STT* | *Các phương thức chính* | Mục đích | Tên file, số thứ tự dòng chứa khai báo | Tên sinh viên phụ trách |
| *1* | DetailStudentInfo\_cbb\_Subjects  Input: int ID, string ProID  Output: List<dynamic>  Pseudo Code:  **Using new UniversityContext** tên là ctx.  **If** quyền truy cập không là Admin  **Then** tạo ra một biến var gồm các bảng join với nhau **where** mã môn học = ID và mã giảng viên gần giống với ProID  **Select new** chọn ra các thuộc tính của biến var để in lên DatagridView gồm Mã sinh viên, tên sinh viên, ngày sinh,…  **Return** biến var dưới dạng List<dynamic>  **Else** khi quyền truy cập là Admin  **Then** tạo ra một biến var gồm các bảng join với nhau **where** mã môn học = ID  **Select new** chọn ra các thuộc tính của biến var để in lên DatagridView gồm Mã sinh viên, tên sinh viên, ngày sinh,…  **Return** biến var dưới dạng List<dynamic>  **End if.** | Dùng đế xuất ra DatagridView danh sách các sinh viên theo mã môn học trong ComboBox. | BLStudent.cs, dòng thứ 11 | Nguyễn Huỳnh Anh Trực |
| *2* | DetailStudentInfo\_cbb\_Classes  Input: string ID, string ProID  Output: List<dynamic>  Pseudo Code:  **Using new UniversityContext** tên là ctx.  **If** quyền truy cập không là Admin  **Then** tạo ra một biến var gồm các bảng join với nhau **where** mã lớp = ID và mã giảng viên gần giống với ProID  **Select new** chọn ra các thuộc tính của biến var để in lên DatagridView gồm Mã sinh viên, tên sinh viên, ngày sinh,…  **Return** biến var dưới dạng List<dynamic>  **Else** khi quyền truy cập là Admin  **Then** tạo ra một biến var gồm các bảng join với nhau **where** mã lớp = ID  **Select new** chọn ra các thuộc tính của biến var để in lên DatagridView gồm Mã sinh viên, tên sinh viên, ngày sinh,…  **Return** biến var dưới dạng List<dynamic>  **End if.** | Dùng đế xuất ra DatagridView danh sách các sinh viên theo mã lớp trong ComboBox. | BLStudent.cs, dòng thứ 62 | Nguyễn Huỳnh Anh Trực |
| *3* | Search  Input: string studentIdentity,string classID, string subjectID  Output: List<dynamic>  Pseudo Code:  **Using new UniversityContext** tên là ctx.  Tạo biến var gồm các bảng join với nhau where studentIdentity gần giống với mã hoặc tên sinh viên, classID gần giống với mã lớp, subjectID gần giống với mã môn học  **Select new** chọn ra các thuộc tính của biến var để in lên DatagridView gồm Mã sinh viên, tên sinh viên, ngày sinh,…  **Return** biến var dưới dạng List<dynamic> | Dùng để xuất ra DatagridView danh sách các sinh viên được tìm kiếm bằng mã sinh viên hoặc tên sinh viên trong TextBox ứng với mã lớp và mã môn học trong các ComboBox. | BLStudent.cs, dòng thứ 112 | Nguyễn Huỳnh Anh Trực |
| *4* | DetailGradeInfo\_cbb\_Classes  Input: string ID, string ProID  Output: List<dynamic>  Pseudo Code:  **Using new UniversityContext** tên là ctx.  **If** quyền truy cập không là Admin  **Then** tạo ra một biến var gồm các bảng join với nhau **where** mã lớp = ID và mã giảng viên gần giống với ProID  **Select new** chọn ra các thuộc tính của biến var để in lên DatagridView gồm Mã lớp, mã môn học, tên môn học, tên sinh viên,...  **Return** biến var dưới dạng List<dynamic>  **Else** khi quyền truy cập là Admin  **Then** tạo ra một biến var gồm các bảng join với nhau **where** mã lớp = ID  **Select new** chọn ra các thuộc tính của biến var để in lên DatagridView gồm Mã lớp, mã môn học, tên môn học, tên sinh viên,...  **Return** biến var dưới dạng List<dynamic>  **End if.** | Dùng đế xuất ra DatagridView danh sách điểm sinh viên theo mã lớp trong ComboBox. | BLGrade.cs, dòng thứ 11 | Nguyễn Huỳnh Anh Trực |
| *5* | DetailGradeInfo\_cbb\_Subjects Input: string ID, string ProID  Output: List<dynamic>  Pseudo Code:  **Using new UniversityContext** tên là ctx.  **If** quyền truy cập không là Admin  **Then** tạo ra một biến var gồm các bảng join với nhau **where** mã môn học = ID và mã giảng viên gần giống với ProID  **Select new** chọn ra các thuộc tính của biến var để in lên DatagridView gồm Mã lớp, mã môn học, tên môn học, tên sinh viên,...  **Return** biến var dưới dạng List<dynamic>  **Else** khi quyền truy cập là Admin  **Then** tạo ra một biến var gồm các bảng join với nhau **where** mã môn học = ID  **Select new** chọn ra các thuộc tính của biến var để in lên DatagridView gồm Mã lớp, mã môn học, tên môn học, tên sinh viên,...  **Return** biến var dưới dạng List<dynamic>  **End if.** | Dùng đế xuất ra DatagridView danh sách điểm sinh viên theo mã môn học trong ComboBox. | BLGrade.cs, dòng thứ 53 | Nguyễn Huỳnh Anh Trực |
| *6* | Search  Input: string studentIdentity, string classID, string subjectID  Output: List<dynamic>  Pseudo Code:  **Using new UniversityContext** tên là ctx.  Tạo biến var gồm các bảng join với nhau where studentIdentity gần giống với mã hoặc tên sinh viên, classID gần giống với mã lớp, subjectID gần giống với mã môn học  **Select new** chọn ra các thuộc tính của biến var để in lên DatagridView gồm Mã lớp, mã môn học, tên môn học, tên sinh viên,...  **Return** biến var dưới dạng List<dynamic> | Dùng để xuất ra DatagridView danh sách điểm số sinh viên được tìm kiếm bằng mã sinh viên hoặc tên sinh viên trong TextBox ứng với mã lớp và mã môn học trong các Combobox. | BLGrade.cs, dòng thứ 95 | Nguyễn Huỳnh Anh Trực |
| *7* | UpdateMark  Input: int studentID, string classID, double? mark  Output: Boolean  Pseudo Code:  **Using new UniversityContext** tên là ctx.  **Try**  Trong bảng Grades của ctx, khi sinh viên = studentID, mã lớp = classID thì điểm của sinh viên sẽ = mark được nhập vào.  **Return true** khi điều kiện trên đúng.  **Catch** khi điều kiện trên sai và **return false**.  **End try catch.** | Dùng để cập nhật điểm cho sinh viên qua việc nhập điểm số mới trên TextBox. | BLGrade.cs, dòng thứ 117 | Nguyễn Huỳnh Anh Trực |
| *8* | Readfile  Input: string path  Output: int  Pseudo Code:  **Using new UniversityContext** tên là ctx.  **If** tồn tại file path trong hệ thống  **Then** với mỗi dòng trong file path  Chia nội dung mỗi hàng thành 3 nội dung nhỏ được ngăn cách bởi ký tự khoảng trắng.   * Cụm đầu là mã sinh viên. * Cụm thứ hai là mã lớp. * Cụm thứ ba là điểm số.   **Try**  Trong bảng Grades của ctx, khi mã sinh viên = nội dung cụm đầu tiên , mã lớp = nội dung cụm thứ hai thì điểm của sinh viên sẽ = nội dung cụm thứ ba có trong file path.  **Catch** khi bắt được lỗi trong câu lệnh trên và sẽ **return 1**.  **Return 2** khi đọc được file và Savechanges cho ctx để lưu dữ liệu điểm vào cơ sở dữ liệu.  **Else** khi không tồn tại file path **then return 0.**  **End try catch.** | Dùng để cập nhật điểm số sinh viên thông qua việc đọc một file .txt có sẵn trong hệ thống. | BLGrade.cs, dòng thứ 133 | Đoàn Quốc Hùng |
| *9* | Kmean  Input: bool KmeanPlusPlus, string DepartmentName, string SubjectName, int ProfessorID  Output: int    Pseudo Code:  **Using** new UniversityContext tên là ctx.  **If** DepartmentName được truyền vào là tất cả khoa  **Then**  **Foreach** phần tử trong bảng Grades của ctx  **If** điểm của sinh viên không tồn tại  **Return 0** và không chạy thuật toán  **End if.**  **End foreach.**  Lưu lại tất cả các phần tử trong bảng Grade group by mã sinh viên, mỗi phần tử là tổ hợp các điểm theo các môn học của 1 sinh viên , với key là mã sinh viên đó. Danh sách này là pointList  **Else**  DepartmentName được truyền vào không phải là tất cả khoa  **Then**  **If**( người dùng đăng nhập thuộc quyền professor )  **Then**  Lưu danh sách những sinh viên học ở những lớp thuộc môn mà professor đó đang quảng lý vào PointList.  Điều chỉnh số lượng môn cần xem xét trong thuật toán xuống bằng 1  **Else** ( người đùng đăng nhập là Admin và có lựa chọn bộ lõ để lọc ra những sinh viên thuộc 1 khoa nào đó )  **Then**  Lưu lại danh sách mã khoa khi tên khoa = DepartmentName được truyền ,lưu danh sách sinh viên đang học khoa đó và điểm tương ứng là tempPointList.  **Foreach** mỗi phần tử trong danh sách vừa mới tạo trên  **If** điểm của sinh viên không tồn tại  **Return 0** và không chạy thuật toán  **End foreach**  Lưu lại tất cả tất cả các phần tử trong tempPointList trên group by mã sinh viên, mỗi phần tử là tổ hợp các điểm theo các môn học của 1 sinh viên , với key là mã sinh viên đó. Danh sách này là pointList  **End if**  //Dùng để lưu danh sách sinh viên và điểm tương ứng khi nhập vào một khoa.    **For(**tăng từ 0 đến nhỏ hơn Số lượng cluster cần chia**)**  Cluster = danh sách các cluster mà trong đó mỗi cluster là 1 List chứa các IGrouping tượng đương 1 sinh viên và các điểm của sinh viên đó  Số lượng phần tử trong Cluster = Số cluster cần chia.  **End loop for** //Tạo một một số lượng cluster cần chia, mỗi cluster sẽ chứa một bảng bao gồm mã sinh viên và điểm của sinh viên đó.    **If** thuật toán KmeanPlusPlus không được thực thi  **Then**  CentralPoint là List các điểm trung tâm trong mỗi cluster  **For(**tăng từ 0 đến nhỏ hơn Số lượng cluster cần chia**)**  Khởi tạo list CentralPoint qua việc lấy số lượng điểm ngẫu nhiên = số lượng cluster, những điểm này không trùng nhau và việc lấy ngẫu nhiên sẽ từ mã sinh viên đầu tiên đến cuối danh sách pointlist được tạo ở phần trên.  **End loop for.**  **Else** khi thuật toán KmeanPlusPlus được thực thi  **Then**  Khởi tạo lại một list CentralPoint mới  Tạo số lượng phần tử null của CentralPoint = số lượng cluster cần chia  Thông qua thuật toán KmeanPlusPlus để khởi tạo những điểm này.  **End if.**    **While** (true) là một vòng lặp vô hạn  Khởi tạo lại danh sách các cluster, khởi tạo mới từng phần tử trong cluster = Số lượng cluster cần chia.  Khởi tạo lại danh sách các CompareCentralPoint với số lượng phần tử = Số lượng cluster cần chia. Giá trị các phần tử trong CompareCentralPoint là các CentralPoint được tính trước đó.  **Foreach** mỗi phần tử point trong pointList  **For(**ităng từ 0 đến nhỏ hơn số môn học (20)**)**  **For(**jtăng từ 0 đến nhỏ hơn số cluster cần chia**)**  Kết quả thứ j được lưu lại += (point. điểm môn thứ i – CentralPoint thứ j . điểm môn thứ i ) ^ 2  **End loop for**  **End loop for**    **For(**tăng từ 0 đến nhỏ hơn số cluster cần chia**)**  Lần lượt căn bậc 2 các kết quả lưu ở trên ta được số lượng khoảng cách tới điểm trung tâm tính được = số lượng cluster.  **End loop for**    **For(** ităng từ 0 đến số lượng phần tử kết quả khoảng cách tính được**)**  Cho kết quả min = kết quả thứ 0  **If** kết quả thứ i nhỏ hơn kết quả min  **Then** Kết quả min là kết quả thứ i  GroupMin = i  **End if**  **End loop for**  Add point đó vào Cluster thứ i  *//Với mỗi điểm point, ta tính khoảng cách của điểm đó đến số điểm trung tâm, khoảng cách nhỏ nhất sẽ được lưu vị trí và thêm add point đó vào vị trí cluster tương ứng.*    **Foreach** phần tử eachCluster trong Clusters  **Foreach** phần tử point trong eachCluster  **Foreach** phần tử element trong point  Tính tổng các giá trị điểm ứng với từng môn trong mỗi Cluster(group) là elementList  **End foreach**  **End foreach**  **For (**i tăng từ 0 đến nhỏ hơn số lượng phần tử trong elementList**)**  Tính lại điểm trung tâm = elementList / số lượng phần tử trong mỗi Cluster.  **End foreach**  **End foreach**  Đặt biến cờ flag = true  **For**(i tăng từ 0 đến số lượng cluster )  **For**(j tăng từ 0 đến số môn học)  **If** mỗi điểm CompareCentralPoint khác với điểm trung tâm vừa tính trên  **Then**  flag = false  **End loop for**  **End if**  **If** flag = true  **Then** thoát khỏi vòng lặp while kết thúc thuật toán  **End while.**  **Return 1.** | Phân cụm danh sách sinh viên theo độ tương đồng về điểm số. | BLAnalysis.cs, dòng thứ 143 | Đoàn Quốc Hùng  Nguyễn Huỳnh Anh Trực  Võ Tấn Nguyên |
| *10* | KmeanPlusPlus  Input: ref List<MyGrouping<int, Grade>> CentralPoint, System.Collections.Generic.List<IGrouping<int, Grade>> pointList  Output:    Lấy 2 khóa chính đầu và cuối của pointList để xác định khoảng random.  Khởi tạo CentralPoint đầu tiên bằng random trong khoảng trên là TheFirstPoint  Khởi tạo số lượng point đã tạo là Count  **While(true)** là một vòng lặp vô hạn  **If** Count = số cluster cần chia  **Then** Thoát khỏi vòng while  **End if**  Khởi tạo danh sách chứa khoảng cách đến centralpoint gần nhất của tất cả các điểm là MinDistanceList.  **Foreach** phần tử trong pointList  **For**(i tăng dần từ 0 đến nhỏ hơn Count)  Tạo list Result chứa khoảng cách đến từng centralpoint của 1 điểm  **End loop for**  **For**(i tăng dần từ 0 đến nhỏ hơn số môn(20))  **For**(j tăng dần từ 0 đến Count)  Kết quả thứ j được lưu lại += (point.điểm môn thứ i – CentralPoint thứ j . điểm môn thứ i ) ^ 2  **End loop for**  **End loop for**  Cho kết quả min là minResult = kết quả thứ 0  **For**(i tăng dần từ 0 đến nhỏ hơn Count)  **If** kết quả thứ i < minResult  **Then** MinResult = kết quả thứ i  **End if**  **End loop for**  *//Tìm ra khoảng cách bé nhất trong các khoảng cách từ 1 điểm đến các central point*  Khởi tạo 1 đối tượng đại diện cho 1 point với key là khoảng cách ngắn nhất tới central point gần nhất của point đó là temporaryCentralPoint  **For**(j tăng dần từ 0 đến nhỏ hơn số môn)  Thêm thuộc tính điểm của 20 môn học cho temporaryCentralPoint  **End loop for**  **Foreach** mỗi phần tử trong tempororyCentralPoint  Thêm dữ liệu điểm 20 môn học cho mỗi phần tử đó với key là minResult  **End foreach**  Thêm temporary này vào MinDistanceList  **End foreach**  *//Sau khi có danh sách các khoảng cách của mỗi điểm đến CentralPoint gần nhất, ta so sánh khoảng cách mỗi điểm này với nhau, khoảng cách lớn nhất sẽ được chọn làm centralpoint kế tiếp cho đến khi số lượng centralpoint này = số lượng cluster muốn chia*  Gán MaxPoint = phần tử đầu tiên của MinDistanceList  **Foreach** phần tử point trong MinDistanceList  **If** key của phần tử đó > key của MaxPoint  **Then** MaxPoint = phần tử đó  **End if**  **End foreach**  Thêm một phần tử cho list CentralPoint  **For**(j tăng dần từ 0 đến nhỏ hơn số môn học)  Thêm thuộc tính điểm của 20 môn học cho CentralPoint mới này  **End loop for**  **Foreach** phần tử trong list CentroPoint[Count]  Gán giá trị điểm của các centralpoint mới = giá trị điểm của MaxPoint  **End foreach**  *//Mỗi lần vòng lặp while chạy xong thì sẽ có thêm 1 central point mới.*  Count tăng lên 1  **End while.**  **Return.** | Dùng để khởi tạo các điểm trọng tâm tương ứng với các Cluster cần chia | BLAnalysis.cs, dòng thứ 56 | Đoàn Quốc Hùng |
| *11* | Authentication  Input: string Account, string Password  Output: Boolean  Pseudo Code:  **Using new UniversityContext** tên là ctx.  Trong bảng Professors của ctx, khi có mã giảng viên = Account và mật khẩu = Password được truyền vào thì lưu thông tin của tài khoản giảng viên đó lại.  **If** thông tin giảng viên được lưu lại có tồn tại trong cơ sở dữ liệu **then return true**.  **Else** khi thông tin giảng viên được lưu lại không tồn tại **then return false**.  **End if.** | Dùng để đăng nhập vào ứng dụng. | Login\_Form.cs, dòng thứ 45 | Nguyễn Huỳnh Anh Trực |
| *12* | Btn\_ExportExcel\_Click  Input: none  Output: void  Pseudo Code:  Khởi tạo ứng dụng excel mới  Khởi tạo một Workbook mới trong ứng dụng excel vừa tạo  Khởi tạo một Worksheet  Điều chỉnh thuộc tính Visible cho ứng dụng excel hiển thị lên màn hình  **For**( j chạy từ 0 đến số lượng của ListClusterDataGridView)  Khởi tạo Worksheet mới và chọn làm active worksheet với tên là “Cluster” + j  **Begin try**  **For** ( i chạy từ 0 đến số lượng cột của phần thứ j trong ListClusterDataGridView )  Ô ở vị trí [1,i+1] trong worksheet đang được active bằng giá trị text của cột thứ I trong phần tử thứ j của ListClusterDataGridView  **End lopp for**  **For** ( i chạy từ 0 đến số lượng hàng của phần tử thứ j trong ListClusterDataGridView )  **For** ( f chạy từ 0 đến số lượng cột của phần thứ j trong ListClusterDataGridView )  Ô ở vị trí [i,f+1] trong worksheet đang được active bằng giá trị ở hàng i cột f của phần tử thứ j trong ListClusterDataGridView  **End loop for**  **End lopp for**  **End try**  **Begin catch**  Hiện lên màn hình thông tin lỗi  **End catch**  **End loop for** | Dùng để xuất file excel | Main\_Form.cs, dòng thứ 696 | Đoàn Quốc Hùng |
| *13* | Btn\_ExportTXT\_Click  Input: none  Output: void  Pseudo Code:  Mở một hợp thoại của Microsoft thiết kế đê lưu đường dẫn  Khởi tạo 1 biến kiểu xâu ký tự  Thêm vào xâu thuật toán chạy trong bao nhiêu vòng lặp  **For**( i =0 chạy đến số lượng cluster )  Thêm vào xâu “Central point thứ I là:”  **For**( j =0 chạy đến số lượng thuộc tính trong mỗi thành phần của 1 central point)  Xâu thêm vào giá trị điểm đó cộng với 1 khoảng trống  **End loop for**  Xâu thêm vào “Sau đây là danh sách chi tiết các sinh viên thuộc các cluster ”  **For**( f = 0 chạy đến số lượng phần tử (số cluster ) trong ListClusterDataGridView )  Xâu thêm vào lần lượt các hàng bằng giá trị các thành phần của phần tử thứ f trong cluster thứ i theo theo thứ tự Mã sinh viên, tên sinh viên và lần lượt số điểm tương ứng với số môn ( nguồn lấy các hàng trong phần tử ListDataGridView thứ f)  **End loop for**  **End loop for** | Dùng để xuất file text | Main\_Form.cs, dòng thứ 755 | Võ Tấn Nguyên |
| *14* | DetailForm\_Load | Xuất ra màn hình 1 form hiển thị thông tin lần chạy Kmean gần nhất | DetailForm.cs, dòng thứ 24 | Võ Tấn Nguyên |
| *15* | CreateDataGridviewList()  Input: List<DataGridView>  Output: void  Pseudo Code:  **For**( i =0 chạy đến bé hơn số lượng cluster)  Tạo ra một đối tượng Datagridview  Thêm 2 cột đầu của Datagridview đó lần lượt là Mã số sinh viên và tên sinh viên  **If** (người đăng nhập là admin)  **Then**  **Foreach** (point trong Cluster thứ i)  DataGriview hiện thời thêm lần lượt các hàng bằng giá trị các thành phần của mỗi phần tử trong cluster thứ i theo theo thứ tự Mã sinh viên, tên sinh viên và lần lượt số điểm tương ứng với số môn  **Else** ( người đăng nhập là professor )  **Then**  **Foreach** (point trong Cluster thứ i)  DataGriview hiện thời thêm lần lượt các hàng bằng giá trị các thành phần của mỗi phần tử trong cluster thứ i theo theo thứ tự Mã sinh viên, tên sinh viên và số điểm tương ứng với 1 môn mà Professor đang lọc  **End loop for** | Khởi tạo 1 List các DataGridView dùng cho sự kiện DetailFormLoad cũng như Btn\_ExportExcel\_Click, Btn\_ExportTXT\_Click | Main\_Form.cs, dòng thứ 608 | Đoàn Quốc Hùng |
| *16* | CopyPoint  Input: MyGrouping<int, Grade> a, IGrouping<int, Grade> b  Output: Void | Dùng để copy những giá trị thuộc tính trong 1 biến kiểu reference type sang 1 biến reference type khác, ở đây là biến dạng MyGrouping<key,element> do nhóm tự thêm từ các interface trong thư viện của microsoft và biến dạng Igrouping<key,element> có sẵn trong thư viện | Analysis.cs,  Dòng thứ 32 | Đoàn Quốc Hùng |
| *17* | ExceptDuplicate  Input: int startRan, List<int> listRan, int startingStudentID, int endingStudentID, Random ran  Output: void | Dùng để tránh trùng Central Point khi khởi tạo bằng K-mean truyền thống | Analysis.cs,  Dòng thứ 45 | Võ Tấn Nguyên |
| *18* | btn\_DetailList\_Click  Input:  Output: void | Dùng để hiện 1 đối tượng Detail\_Form trong đó chứa các Datagridview có thông tin chi tiết của lần chạy thuật toán phân cụm gần nhất. | Main\_Form.cs,  Dòng thứ 690 | Đoàn Quốc Hùng |
| *19* | btn\_Kmean\_Click  Input: none  Output: void  Pseudo Code:  Khởi tạo 1 đối tượng BLAnalysis tên Ana truyền vào tham số là số lượng cluster.  **If** ( Ana gọi hàm Kmean bằng 0)  **Then** Hiện ra màn hình thông báo có giá trị trong dữ liệu điểm bằng null nên không chạy được và return  **If** ( bộ lọc Department là tất cả các Khoa)  **Then** studentAmount = tổng số lượng student trong cơ sở dữ liệu.  **Else**  **Then**  **If**(người dùng đăng nhập dưới quyền admin và chọn lọc ra sinh viên thuộc 1 khoa)  **Then** studentAmount bằng tổng số sinh viên thuộc khoa đó  **Else**(Người dùng đăng nhập dưới quyền Professor)  **Then** studentAmount = tổng số lượng sinh viên trong các lớp thuộc môn mà professor đó đang quản lý.  **Foreach**( từng cluster trong danh sách cluster )  Biểu đồ cột nhận giá trị bằng số lượng sinh viên trong mỗi cluster  Biểu đồ tròn nhận giá trị bằng phần trăm của số lượng sinh viên trong 1 cluster so với tổng số sinh viên của dữ liệu đầu vào  Biểu đồ cột biểu hiện điểm nhận giá trị điểm trung bình từ giá trị trung bình của của các thành phần trong 1 CentralPoint  **End loop foreach** | Dùng để chạy K-mean hoặc K-mean++ và xuất ra màn hình biểu đồ trực quan | Main\_Form.cs,  dòng thứ 543 | Đoàn Quốc Hùng |

## *Thiết kế cơ sở dữ liệu*



Hình 7: Sơ đồ ERD

Bảng 4: Thiết kế bảng cơ sở dữ liệu

|  |  |  |  |
| --- | --- | --- | --- |
| **STT** | **Tên bảng** | **Mục đích chính của bảng trong chương trình** | **Tên các SV phụ trách viết** |
| 1 | Students | Dùng để lưu trữ danh sách sinh viên và thông tin gồm mã sinh viên, tên, ngày sinh, giới tính,... | Đoàn Quốc Hùng  Võ Tấn Nguyên |
| 2 | Professors | Dùng để lưu trữ danh sách giảng viên và thông tin gồm mã giảng viên, tên, ngày sinh, số điện thoại,… và một thuộc tính isAdmin để biết đối tượng đó có phải là admin hay không. | Đoàn Quốc Hùng  Nguyễn Huỳnh Anh Trực . |
| 3 | Grades | Dùng để lưu trữ danh sách gồm mã sinh viên, mã lớp và điểm số ứng với sinh viên thuộc lớp học đó. | Đoàn Quốc Hùng  Nguyễn Huỳnh Anh Trực |
| 4 | Departments | Dùng để lưu trữ danh sách khoa và thông tin bao gồm mã khoa, tên khoa và số điện thoại. | Võ Tấn Nguyên  Nguyễn Huỳnh Anh Trực |
| 5 | Subjects | Dùng để lưu trữ danh sách các môn học của sinh viên và thông tin gồm mã môn học, tên môn học. | Đoàn Quốc Hùng  Võ Tấn Nguyên |
| 6 | Classes | Dùng để lưu trữ danh sách các lớp học của sinh viên và thông tin bao gồm mã lớp, mã môn học, mã giảng viên. | Đoàn Quốc Hùng  Võ Tấn Nguyên |

## *Mô tả các Field trong 1 table*

1. **Bảng Classes.**

Bảng 5: Mô tả các field trong bảng Classes

|  |  |  |  |
| --- | --- | --- | --- |
| STT | Tên trường | Kiểu dữ liệu | Mục đích |
| 1 | ClassID | nvarchar(128) | Khóa chính cho bảng lớp, với mỗi lớp sẽ có một khóa chính khác nhau. |
| 2 | SubjectID | nchar(6) | Khóa ngoại liên kết với khóa chính của bảng Subjects. |
| 3 | ProfessorID | int | Khóa ngoại liên kết với khóa chính của bảng Professors. |

1. **Bảng Students.**

Bảng 6: Mô tả các field trong bảng Students

|  |  |  |  |
| --- | --- | --- | --- |
| STT | Tên trường | Kiểu dữ liệu | Mục đích |
| 1 | StudentID | int | Khóa chính cho bảng sinh viên, với mỗi sinh viên sẽ có một khóa chính khác nhau. |
| 2 | StudentName | nvarchar(MAX) | Lưu trữ thông tin về tên sinh viên. |
| 3 | Sex | nvarchar(MAX) | Lưu trữ thông tin về giới tính sinh viên. |
| 4 | DateOFBirth | datetime | Lưu trữ thông tin về ngày sinh của sinh viên. |
| 5 | DepartmentID | int | Khóa ngoại liên kết với khóa chính của bảng Department. |

1. **Bảng Grades.**

Bảng 7: Mô tả các field trong bảng Grades

|  |  |  |  |
| --- | --- | --- | --- |
| STT | Tên trường | Kiểu dữ liệu | Mục đích |
| 1 | StudentID | int | Khóa chính được liên kết với khóa chính của bảng Students. |
| 2 | ClassID | nvarchar(128) | Khóa chính được liên kết với khóa chính của bảng Classes. |
| 3 | Mark | float | Lưu trữ thông tin về điểm số của sinh viên. |

1. **Bảng Subjects.**

Bảng 8: Mô tả các field trong bảng Subjects

|  |  |  |  |
| --- | --- | --- | --- |
| STT | Tên trường | Kiểu dữ liệu | Mục đích |
| 1 | SubjectID | nchar(6) | Khóa chính cho bảng môn học, với mỗi môn học sẽ có một khóa chính khác nhau. |
| 2 | SubjectName | nvarchar(MAX) | Lưu trữ thông tin về tên môn học. |

1. **Bảng Professors.**

Bảng 9: Mô tả các field trong bảng Professors

|  |  |  |  |
| --- | --- | --- | --- |
| STT | Tên trường | Kiểu dữ liệu | Mục đích |
| 1 | ProfessorID | int | Khóa chính cho bảng giảng viên, với mỗi đối tượng giảng viên sẽ có một mã giảng viên khác nhau. |
| 2 | ProfessorName | nvarchar(MAX) | Lưu trữ thông tin về tên của giảng viên. |
| 3 | BirthDate | datetime | Lưu trữ thông tin về ngày sinh của giảng viên. |
| 4 | PhoneNumber | nvarchar(MAX) | Lưu trữ thông tin về số điện thoại của giảng viên. |
| 5 | Password | nvarchar(MAX) | Lưu trữ thông tin về mật khẩu của giảng viên để đăng nhập vào ứng dụng. |
| 6 | isAdmin | bit | Dùng để hiển thị giảng viên đó có phải là Admin hay không. |

1. **Bảng Departments.**

Bảng 10: Mô tả các field trong bảng Departments

|  |  |  |  |
| --- | --- | --- | --- |
| STT | Tên trường | Kiểu dữ liệu | Mục đích |
| 1 | DepartmentID | int | Khóa chính cho bảng khoa, với mỗi đối tượng khoa sẽ có một mã giảng khoa khác nhau. |
| 2 | DepartmentName | nvarchar(MAX) | Lưu trữ thông tin về tên khoa. |
| 3 | PhoneNumber | nchar(10) | Lưu trữ thông tin về số điện thoại của khoa. |

* 1. ***Thiết kế giao diện***

Bảng 11: Bảng thiết kế giao diện

|  |  |  |  |
| --- | --- | --- | --- |
| TT | Màn hình/Cửa sổ/Dialog (Tên màn hình trên 1 dòng; dưới dòng này là hình chụp màn hình) | Mục đích | Giải thích ngắn gọn các quyết định thiết kế khi thiết kế màn hình (Người thiết kế ở trên 1 dòng. Sau dòng đó là phần giải thích. |
| 1 | Login\_Form | Dùng để tạo đăng nhập vào phần mềm và phân quyền hạn trong phần mềm. | * Võ Tấn Nguyên   Dữ liệu vào sẽ có tên tài khoản và mật khẩu. Ở đây có 5 chức năng:  1 là Show Pass dùng để show Password đã nhập.  2 là chọn quyền truy cập là admin để vào ứng dụng có quyền của admin.  3 là quyền teacher để truy cập vào ứng dụng có quyền của teacher.  4 Button Login dùng để đăng nhập vào ứng dụng.  5 Button Exit dùng để thoát.  Khi ta đăng nhập đúng thì sẽ vào được ứng dụng còn sai thì sẽ có bảng báo lỗi hiện lên. |
| 2 |  | Dùng để hiển thị thông tin của giáo viên. | Võ Tấn Nguyên  Dùng 3 cái textbox để show thông tin của teacher |
| 3 |  | Dùng để hiển thị thông tin của Student. Lựa chọn hiển thị theo lớp và theo môn. Tìm kiếm một student nào đó. | * Võ Tấn Nguyên   Dùng DatagridView để hiển thị các thông tin của học sinh. Có 2 ComboBox một cái để cho mình chọn hiển thị thông tin sinh viên theo lớp học, 1 cái để chọn hiển thị thôn tin sinh viên theo môn học. 1 cái Textbox dùng để tìm kiếm sinh viên theo tên, ID |
| 4 |  | Đây là menu chính của phần mềm dùn để khi click vào sẽ có sự kiện hiện các tab\_control ứng với mỗi button lên | * Võ Tấn Nguyên   Đây là Menu chính để chọn các tác vụ. Menu này t dùng 1 Panel trong Panel có 2 label để chứa dòng trên là tên người đăng nhập vào, dòng thứ 2 là quyền của người này trong ứng dụng.  PictureBox để hiển thị ảnh đại diện cho ứng dụng. Có 5 cái Button , button Dashboard để vào xem thông tin của người đăng nhập, button Student để vào xem thông tin student, buttongrade để xem điểm, button Analysis đùng để chạy thuật toán K-Means, button logout để đăng xuất người dùng. |
| 5 |  | Dùng để hiện thị điểm của các sinh viên, hiển thị theo lớp, hiển thị theo môn học, chúng ta sẽ được sửa điểm cho sinh viên tìm kiếm và đặc biệt là nhập file text để đọc điểm đưa vào database. | * Võ Tấn Nguyên   Dùng DatagidView để hiển thị thông tin điểm của các sinh viên. Có 2 ComboBox cái trên dùng để chọn xem điểm theo môn, cái ở dưới chọn xem điểm theo lớp. có 3 Textbox t, textbox đầu dùng để chỉnh sửa điểm cho học sinh. Khi ta click vào một học sinh trong DatagridView textbox này mới hiện lên, có 1 textbox của chọn file text nhập điểm vào, 1 textbox của tìm kiếm sinh viên theo tên và mã số. |
| 6 |  | Dùng để chay thuật toán K-Means và hiển thị theo dạng biểu đồ, dạng bảng xuất file text và xuất file excel | * Võ Tấn Nguyên   Có 1 ComboBox dùng để chọn danh sách sinh viên đầu vào cho thuật toán K-mean theo khoa và chỉ có sử dụng bởi tài khoản quyền admin, có 1 ComboBox dùng để chọn danh sách sinh viên đầu vào cho thuật toán K-mean theo những lớp thuộc những môn mà giảng viên đang dạy và chỉ có sử dụng bởi tài khoản quyền Professor. Numericupdown dùng để chọn số lượng vùng muốn phân chia trong thuật toán K-mean, 2 radioButton, cái trên dùng để chạy chọn chạy thuật toán theo kiểu K-mean, cái dưới dùng để chọn chạy thuật toán theo kiểu K-mean++, 3 Button, Button Analize dùng để chạy thuật toán K-mean ra biểu đồ, Button “View Detail List Students In Each Cluster” đẻ hiển thị ra các bảng chứa điểm các môn học và tên học sinh của từng phân vùng mà ta đã chạy chia trong K-mean. Button Export Excel để thực hiện việc xuất dữ liệu đã chạy ra file excel, button Export TXT thực hiện xuất dữ liệu đã chạy K-mean ra file text |

1. **Cài đặt và kiểm thử :**

Bảng 12: Bảng cài đặt và kiểm thử

|  |  |  |  |
| --- | --- | --- | --- |
| 1 | Thử chạy K-mean++ với 12000 điểm cho 20 môn học, mỗi sinh viên có 20 điểm, với tỉ lệ 2 sinh viên khá thì có 1 sinh viên giỏi và 1 sinh viên trung bình. ( file test.txt đính kèm trong project)  Số lượng cluster yêu cầu cho thuật toán là 3.  Kết quả dự kiến: 3 cụm điểm rõ ràng | Kết quả có 3 cụm rõ ràng, phân điểm 8 phân điểm 6 và phân điểm 3 ( hiển thị thông qua thông số của các central point )  Kết quả đúng như dự đoán  Thuật toán chạy trong 1 vòng lặp | Dữ liệu đầu vào đẹp được tạo theo hướng sẽ có 3 cụm, nên thử chạy thuật toán với số cụm yêu cầu là 3 để xem kết quả có đẹp không. |
| 2 | Thử chạy K-mean++ với 12000 điểm cho 20 môn học, mỗi sinh viên có 20 điểm, với tỉ lệ 1 sinh viên khá thì có 1 sinh viên giỏi và 1 sinh viên trung bình,1 sinh viên yếu, với số lượng điểm. ( file test2.txt đính kèm trong project)  Số lượng cluster yêu cầu cho thuật toán là 4.  Kết quả dự kiến: 4 cụm điểm rõ ràng | Kết quả có 4 cụm điểm rõ ràng, phân điểm gần 9, phân điểm 7, phân điểm 5, phân điểm 2,5. ( hiển thị thông qua thông số của các central point )  Kết quả đúng như dự đoán  Thuật toán chạy trong 1 vòng lặp | Dữ liệu đầu vào đẹp được tạo theo hướng sẽ có 4 cụm, nên thử chạy thuật toán với số cụm yêu cầu là 4 để xem kết quả có đẹp không. |
| 3 | Thử chạy K-mean truyền thống và dữ liệu giống với trường hợp 2  Số lượng cluster yêu cầu cho thuật toán là 4.  Kết quả dự kiến: 4 cụm điểm có thể bị sai vì K-mean truyền thống khởi tạo có thể dẫn đến kết quả không tốt | Kết quả có 4 cụm điểm, phân điểm gần 8, phân điểm 5, phân điểm 4.9 gần 5 và phân điểm 2,5 ( hiển thị thông qua thông số của các central point )  Kết quả đúng như dự đoán  Thuật toán chạy trong 9 lần | Thử chạy K-mean truyền thống để xem khả năng khởi tạo của K-mean có gây ra kết quả xấu không |
| 4 | Thử chạy K-mean++ và dữ liệu giống với trường hợp 2  Số lượng cluster yêu cầu cho thuật toán là 5  Kết quả dự đoán: dự đoán là nếu chia số cụm không phù hợp thì thuật toán sẽ ra 2 cụm tương đối gần nhau | Kết quả có 5 cụm điểm, phân điểm 2,2, phân điểm 2,7 , phân điểm 4,9, phân điểm 6.9, phân điểm 8.9  Kết quả đúng như dự đoán  Thuật toán chạy trong 6 lần | Thử chạy K-mean++ để xem số lượng cụm điểm không phù hợp có thể dẫn đến kết quả xấu không |

# KẾT LUẬN:

## *Đánh giá chung:*

* Đồ án đã được hoàn thành được 80% mục tiêu đề ra là tạo ra được 1 ứng dụng có thể mang tính thực tiễn.

1. ***Các khó khăn gặp phải :***
   * Các thành viên ít có cơ hội gặp nhau nên quá trình làm bài đôi khi không nhất quán, việc viết 1 ứng dụng ORM để quản lý 1 mô hình nào đó đã được học qua môn lập trình trên Window ở học kỳ trước nên tương đối quen thuộc không có trở ngại nhưng khó khăn lớn nhất của nhóm là cách biểu diễn quá trình chạy thuật toán K-mean cho người dùng dễ hiểu cách K-mean chạy thì nhóm vẫn chưa giải quyết được.

## *Ưu điểm:*

* Giao diện dễ hiểu, trực quan dễ nắm bắt
* Có khả năng xuất kết quả phân cụm ra các định dạng file khác (Excel và TXT) để dễ dàng hơn trong việc xem lại, có thể lấy điểm 1 file txt điểm thay cho việc nhập điểm thủ công
* Có sử dụng phương pháp khởi tạo K-mean ++ hiệu quả và chính xác hơn so với giải thuật K-mean truyền thống.
* Có việc tổ chức các class, tổ chức cơ sỡ dữ liệu và sử dụng giải thuật phù hợp với yêu cầu bài toán, cũng như có sử dụng mô hình ba tầng (giao diện, nghiệp vụ, dữ liệu) để xây dựng ứng dụng.
* Xuất kết quả của việc phân cụm ra màn hình theo dạng biểu đồ trực quan với người dùng
* Phân tách rõ ràng khả năng của 2 nhóm người dùng: giảng viên và admin, trong đó giảng viên chỉ được truy cập và quản lý những lớp mà mình đăng dạy và cũng chỉ được chạy phân tích phân cụm ở những môn mình đang dạy

## *Nhược điểm:*

* Chưa tối ưu hóa hiệu năng phần mềm, các câu truy vấn qua entity framework còn tốn nhiều thời gian ( đặc biệt là lúc khởi tạo form sau khi đăng nhập thành công )
* Mô hình thiết kế cơ sở dữ liệu dữ theo mô hình hướng đối tượng còn đơn giản, tính đóng gói chưa sử dụng hiệu quả, chưa sử dụng các tính chất kế thừa, đa hình, chỉ tập trung xây dựng với mục tiêu là việc dễ dàng nhập điểm và thực hiện thuật toán phân cụm, nên được mở rộng phức tạp hơn thành một ứng dụng quản lý sinh viên cho phép thực hiện các chức năng CRUD trên nhiều đối tượng trong quy mô trường học, cũng như phát triển các class chặt chẽ để có thể sử dụng phát triển rộng hơn nữa với các chức năng quản lý khác
* Ứng dụng bị giới hạn ở mức độ máy tính nội bộ, khi chạy ứng dụng lần đầu thì dữ liệu sẽ được đẩy vào trong Hệ quản trị cơ sở dữ liệu thông qua phương pháp Entity Framework code first, điều này là điểm hạn chế không thực tiễn, ứng dụng nên được xây dựng là 1 ứng dụng online có thể lấy dữ liệu từ 1 cơ sở dữ liệu cố định trên 1 máy chủ
* Ứng dụng chỉ có sử sung 2 giải thuật theo hướng machine learning đơn giản, chưa có nhiều tùy chọn hiệu quả hơn cho người dùng
* Chưa thể hiển cách K-mean chạy bằng 1 đồ thị các điểm thay đổi( mỗi màu là mỗi cụm) mỗi vòng lặp của thuật toán, để người dùng dễ hiểu cách thuật toán chạy.
* Ứng dụng có ít chức năng.

1. ***Hướng phát triển:***
   * Xây dựng một ứng dụng online có thể sử dụng bởi nhiều người dùng với cơ sở dữ liệu ở trên 1 máy server.
   * Cải thiện hiểu quả của việc phân nhóm bằng cách tìm hiểu thêm nhiều giải thuật phân nhóm khác để có thể thêm hiệu quả
   * Sử dụng kỹ thuật lazy loading hoặc cập nhật bất đồng bộ trong entity framework để tăng hiệu xuất truy vấn đến cơ sở dữ liệu
   * Xây dựng cho ứng dụng có tính năng thể hiện rõ quá trình chạy k-mean để mô tả tốt cho người dùng hiểu rõ nguyên lý thuật toán.
   * Xây dựng một mô hình hướng đối tượng phức tạp hơn để có thể mở rộng ứng dụng ra các chức năng khác.
2. **TÀI LIỆU KHAM KHẢO:**

Đồ án của nhóm có tham khảo một số bài báo về việc cơ chế thực thi giải thuật K-mean và K-mean ++ trên Google.com để hiểu giải thuật, chứ không sử dụng lại mã của người khác.

Các trang đã tham khảo:

<https://machinelearningcoban.com/2017/01/01/kmeans/>

<https://kipalog.com/posts/Thuat-toan-Kmean-va-ung-dung>

<https://dominhhai.github.io/vi/2018/02/ml-kmeans/#1-%C4%91%E1%BA%B7t-v%E1%BA%A5n-%C4%91%E1%BB%81>