

BÁO CÁO TIẾN ĐỘ HÀNG TUẦN ĐỒ ÁN CNTT

1. Thông tin chung

- Môn: Đồ án CNTT
- GVHD: TS. Huỳnh Xuân Phụng
- Tên đề tài: Xây dựng chương trình quản lý chương trình đào tạo khoa CNTT
- Danh sách nhóm

STT	Họ và tên	MSSV	Vai trò	Email	SĐT
1	Nguyễn Cước Thiên	18110204	Nhóm trưởng	18110204@student.hcmute.edu.vn	0708033730
2	Tu Ngọc Yến Vy	18110243	Thành viên	18110243@student.hcmute.edu.vn	0334874819

2. Kế hoạch tiến độ

Tuần 5-8 có thể gặp GVHD để báo cáo, trao đổi.

- Tuần 1 (8/10-13/10) : Nhận đồ án, lên lịch làm việc và lập kế hoạch
- Tuần 2 (14/10-20/10): Phân tích, tìm hiểu đề tài và các hệ thống liên quan
- Tuần 3 & 4 (21/10-03/11): Phân tích và thiết kế đồ án, phân công nhiệm vụ (chức năng, giao diện, database....)
- Tuần 5-8 (04/11-1/12): Cài đặt, kết nối, triển khai chương trình
- Tuần 9 (2/12-8/12): Kiểm thử đồ án, viết báo cáo
- Tuần 10 (9/12-): Báo cáo

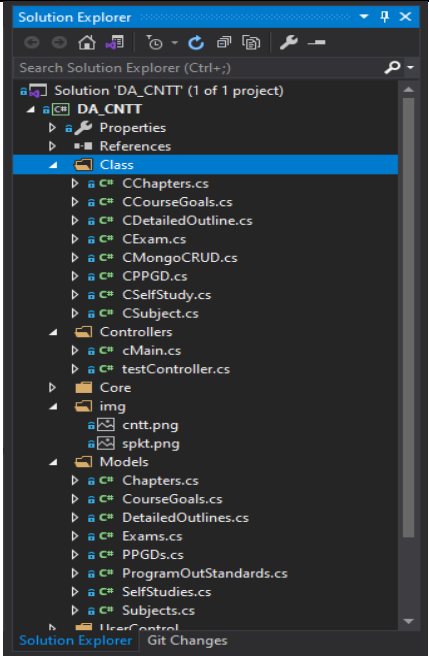
3. Tiến độ của nhóm

a. Nội dung

Tuần	Nội dung	Hoàn thành(%)	GV nhận xét
Tuần 1 (8/10)	Nhận đồ án, lên lịch gặp mặt các thành viên thảo luận, lên kế hoạch	100%	
Tuần 2 (14/10)	<p>Phân tích, tìm hiểu đề tài và các hệ thống liên quan</p> <ul style="list-style-type: none">• Download chương trình đào tạo trên trang fit.hcmute.edu.vn• Chương trình đào tạo gồm đại học và sau đại học• Đại học thì gồm các ngành Công nghệ thông tin, kỹ thuật dữ liệu, sư phạm công nghệ thông tin• Ngành công nghệ thông tin, kỹ thuật dữ liệu chia ra gồm chương trình dành cho khóa trước 2018, khóa 2018 và khóa 2019 trở đi. Riêng ngành sư phạm chỉ áp dụng các khóa trước 2018, hiện nay không còn• Đào tạo sau đại học là chương trình đào tạo thạc sĩ cho ngành khoa học máy tính, gồm 2 hướng ứng dụng và nghiên cứu• Ngoài ra còn có danh sách các môn học quy đổi tương đương cho các khóa dễ dàng đăng ký học lại.• Để hỗ trợ cho việc đào tạo quản lý thêm các bộ môn, để sinh viên tìm hiểu các bộ môn có trong khoa CNTT cũng như các giảng viên phụ trách• Cũng như giới thiệu về khoa CNTT, chức năng, nhiệm vụ	100%	

	vụ, sơ đồ tổ chức...		
Tuần 3(21/10)	Phân tích các chức năng của chương trình, thiết kế databse cơ bản: <ul style="list-style-type: none"> • Tạo sơ đồ chức năng • Tạo phân quyền người dùng • Các thông tin về quản lý chương trình đào tạo (tạo các table) • Mô hình thực thể liên kết ERD 		
Tuần 4 & 5(28/10)	<ul style="list-style-type: none"> • Chỉnh sửa database • Nhập dữ liệu cho một số các bảng trong databse(Bộ Môn, Ngành, Khóa, Hệ, User) • Thiết kế giao diện cơ bản của chương trình 		
Tuần 6 & 7 (18/11)	Làm lại chương trình từ đầu <ul style="list-style-type: none"> • Xây dựng lại databse theo kiểu no sql(sử dụng MongoDB) • Kết nối database với chương trình • Viết class cho model • Viết 1 số chức năng cơ bản 		

b. Hình ảnh

<p>Solution</p>	
<p>CÁC CLASS</p>	
<p>class CChapters</p> <p>các hàm điều khiển của model Chapters sẽ được viết và xử lý ở đây</p>	<pre>namespace DA_CNTT.Class { 2 references public class CChapters { private CMongoCRUD mongo; 1 reference public CChapters() { mongo = new CMongoCRUD("QLCTDT"); } 0 references public List<Chapters> findAll() { var result = this.mongo.Read<Chapters>("Chapters"); return result; } 0 references public Chapters findfromsubject(ObjectId id) { var result = this.mongo.ReadById<Chapters>("Chapters", id); return result; } } } //truyền ob_ID từ controllers 0 references public void Delete() { var id = new ObjectId("5fc509436184428b8096c1d5"); var ID = "Chương 1"; var x = this.mongo.ReadById<Chapters>("Chapters", id); var y = x.Chapter.Where(c => c.ID == ID).SingleOrDefault(); x.Chapter.Remove(y); this.mongo.Update<Chapters>("Chapters", id, x); } //Truyền record từ controllers 1 reference public void Update() { var id = new ObjectId("5fc509436184428b8096c1d5"); var ID = "Chương 2"; string ID2 = "Chương 2.2"; var a = this.mongo.ReadById<Chapters>("Chapters", id); var b = a.Chapter.Where(c => c.ID == ID).SingleOrDefault(); a.Chapter.Remove(b); b.ID = ID2; a.Chapter.Add(b); this.mongo.Update<Chapters>("Chapters", id, a); } //</pre>

<p>class CCourseGoals</p> <p>các hàm điều khiển của model CourseGoals sẽ được viết và xử lý ở đây</p>	<pre> public class CCourseGoals { private CMongoCRUD mongo; 1 reference public CCourseGoals() { mongo = new CMongoCRUD("QLCTDT"); } References public List<CourseGoals> findAll() { var result = this.mongo.Read<CourseGoals>("CourseGoals"); return result; } //truyền ob_ID từ controllers References public void delete() { var id = new ObjectId("5fbbcadb889853cc7e570d30"); this.mongo.DeleteById<CourseGoals>("CourseGoals", id); } //Truyền record từ controllers References public void Update() { var id = new ObjectId("5fbbcadb889853cc7e570d30"); string ID_Goal = "G2"; var a = this.mongo.ReadById<CourseGoals>("CourseGoals", id); //a.ID_Goal = ID_Goal; this.mongo.Update<CourseGoals>("CourseGoals", id, a); } } </pre>
<p>class CDetailedOutline</p> <p>các hàm điều khiển của model DetailedOutlines sẽ được viết và xử lý ở đây</p>	<pre> public class CDetailedOutline { private CMongoCRUD mongo; 1 reference public CDetailedOutline() { mongo = new CMongoCRUD("QLCTDT"); } References public List<DetailedOutlines> findAll() { var result = this.mongo.Read<DetailedOutlines>("DetailedOutline"); return result; } //truyền ob_ID từ controllers References public void delete() { var id = new ObjectId("5fbc04bc87949980dedc76dd"); this.mongo.DeleteById<DetailedOutlines>("DetailedOutline", id); } //Truyền record từ controllers References public void Update() { var id = new ObjectId("5fbc04bc87949980dedc76dd"); string ID = "WEPR440479"; var a = this.mongo.ReadById<DetailedOutlines>("DetailedOutline", id); //a.Course Code = ID; this.mongo.Update<DetailedOutlines>("DetailedOutline", id, a); } } </pre>

class CExam

các hàm điều khiển của
model Exams sẽ được viết
và xử lý ở đây

```
public class CExam
{
    private CMongoCRUD mongo;
    1 reference
    public CExam()
    {
        mongo = new CMongoCRUD("QLCTDT");
    }
    References
    public List<Exams> findAll()
    {
        var result = this.mongo.Read<Exams>("Exam");
        return result;
    }
    //truyền ob_ID từ controllers
    References
    public void delete()
    {
        var id = new ObjectId("5fbbd7fc889853cc7e570d3d");
        this.mongo.DeleteById<Exams>("Exam", id);
    }
    //Truyền record từ controllers
    References
    public void Update()
    {
        var id = new ObjectId("5fbbd7fc889853cc7e570d3d");
        string Date = "Tuần 22";
        var a = this.mongo.ReadById<Exams>("Exam", id);
        // a.Date = Date;
        this.mongo.Update<Exams>("Exam", id, a);
    }
}
```

class CMongoCRUD

các hàm điều khiển của
them xóa sửa tìm kiếm
collection sẽ được viết và
xử lý ở đây

```
public class CMongoCRUD
{
    private IMongoDatabase db;
    7 references
    public CMongoCRUD(string database)
    {
        var client = new MongoClient();
        db = client.GetDatabase(database);
    }
    7 references
    public List<T> Read<T>(string table)
    {
        var collection = this.db.GetCollection<T>(table);
        return collection.Find(new BSONDocument()).ToList();
    }
    9 references
    public T ReadById<T>(string table, ObjectId id)
    {
        var collection = this.db.GetCollection<T>(table);
        return collection.Find(new BSONDocument("_id", id)).SingleOrDefault().Result;
    }
    6 references
    public void DeleteById<T>(string table, ObjectId id)
    {
        var collection = db.GetCollection<T>(table);
        var filter = Builders<T>.Filter.Eq("_id", id);
        collection.DeleteOne(filter);
    }
    8 references
    public void Update<T>(string table, ObjectId id, T record)
    {
        var collection = db.GetCollection<T>(table);
        pragma warning disable CS0618 // Type or member is obsolete
        var result = collection.ReplaceOne(
            new BSONDocument("_id", id),
            record,
            new UpdateOptions
            {
                IsUpsert = true
            }
        );
        pragma warning restore CS0618 // Type or member is obsolete
    }
}
```

class CPPGD

các hàm điều khiển của
model PPGDs sẽ được viết
và xử lý ở đây

```
2 references
public class CPPGD
{
    private CMongoCRUD mongo;
    1 reference
    public CPPGD()
    {
        mongo = new CMongoCRUD("QLCTDT");
    }
    0 references
    public List<PPGDs> findAll()
    {
        var result = this.mongo.Read<PPGDs>("PPGD");
        return result;
    }
    //truyền ob_ID từ controllers
    0 references
    public void delete()
    {
        var id = new ObjectId("5fbc2cdf87949988dedc7d40");
        this.mongo.DeleteById<PPGDs>("PPGD", id);
    }
    //Truyền record từ controllers
    0 references
    public void Update()
    {
        var id = new ObjectId("5fbc2cdf87949988dedc7d40");
        string ID = "PPGD!!";
        var a = this.mongo.ReadById<PPGDs>("PPGD", id);
        // a.ID = ID;
        this.mongo.Update<PPGDs>("PPGD", id, a);
    }
}
```

class CSelfStudy

các hàm điều khiển của
model SelfStudies sẽ được
viết và xử lý ở đây

```
2 references
public class CSelfStudy
{
    private CMongoCRUD mongo;
    1 reference
    public CSelfStudy()
    {
        mongo = new CMongoCRUD("QLCTDT");
    }
    0 references
    public List<SelfStudies> findAll()
    {
        var result = this.mongo.Read<SelfStudies>("SelfStudy");
        return result;
    }
    //truyền ob_ID từ controllers
    0 references
    public void delete()
    {
        var id = new ObjectId("");
        this.mongo.DeleteById<SelfStudies>("SelfStudy", id);
    }
    //Truyền record từ controllers
    0 references
    public void Update()
    {
        var id = new ObjectId("5fbc2cdf87949988dedc7d40");
        string ID = "PPGD!!";
        var a = this.mongo.ReadById<SelfStudies>("SelfStudy", id);
        // a.ID = ID;
        this.mongo.Update<SelfStudies>("SelfStudy", id, a);
    }
}
```

class CSubject

các hàm điều khiển của
model Subjects sẽ được
viết và xử lý ở đây

```
4 references
public class CSubject
{
    private CMongoCRUD mongo;
    1 reference
    public CSubject()
    {
        mongo = new CMongoCRUD("QLCTDT");
    }
    0 references
    public List<Subjects> findAll()
    {
        var result = this.mongo.Read<Subjects>("Subject");
        return result;
    }
    //truyền ob_ID từ controllers
    0 references
    public void delete()
    {
        var id = new ObjectId("5fbddaf687949980dedc7178");
        this.mongo.DeleteById<Subjects>("Subject", id);
    }
    //Truyền record từ controllers
    0 references
    public void Update()
    {
        var id = new ObjectId("5fbde3f287949980dedc724b");
        string Credits = "8";
        var a = this.mongo.ReadById<Subjects>("Subject", id);
        a.Credits = Credits;
        this.mongo.Update<Subjects>("Subject", id, a);
    }
}
```

CÁC LỚP MODEL

Chapters.cs

```
8 references
public class Chapters
{
    0 references
    public ObjectId _id { get; set; }

    [JsonProperty("Chapter")]
    5 references
    public List<Chapter> Chapter { get; set; }
}
1 reference
public class Chapter
{
    [JsonProperty("ID")]
    3 references
    public string ID { get; set; }

    [JsonProperty("Name")]
    0 references
    public string Name { get; set; }

    [JsonProperty("Detail")]
    0 references
    public List<string> Detail { get; set; }
}
```


CourseGoals.cs

```
0 references
public class CourseGoals
{
    0 references
    public ObjectId id { get; set; }

    [JsonProperty("Course_Goal")]
    0 references
    public List<CourseGoal> CourseGoal { get; set; }
}

1 reference
public class CourseGoal
{
    [JsonProperty("ID_Goal")]
    0 references
    public string ID_Goal { get; set; }

    [JsonProperty("Description_Goal")]
    0 references
    public string Description_Goal { get; set; }

    [JsonProperty("ID_CTDI")]
    0 references
    public List<string> ID_CTDI { get; set; }

    [JsonProperty("ID_CDR")]
    0 references
    public List<string> ID_CDR { get; set; }
}
```

DetailedOutlines.cs

```
public class DetailedOutlines
{
    0 references
    public ObjectId _id { get; set; }

    [JsonProperty("Description")]
    0 references
    public List<string> Description { get; set; }

    [JsonProperty("Goal")]
    0 references
    public List<Goal> Goal { get; set; }

    [JsonProperty("Test")]
    0 references
    public List<Test> Test { get; set; }

    [JsonProperty("Detail")]
    0 references
    public List<Detail> Detail { get; set; }

    [JsonProperty("Student_Duties")]
    0 references
    public List<string> Student_Duties { get; set; }

    [JsonProperty("Learning_Materials")]
    0 references
    public List<string> Learning_Materials { get; set; }

    [JsonProperty("Scientific_Ethics")]
    0 references
    public List<string> Scientific_Ethics { get; set; }
}

1 reference
public class Goal
{
    [JsonProperty("ID")]
    0 references
    public string ID { get; set; }

    [JsonProperty("CDR")]
    0 references
    public List<string> CDR { get; set; }
}
```

```
public class Test
{
    [JsonProperty("Formative_Assessment")]
    0 references
    public string Formative_Assessment { get; set; }

    [JsonProperty("Final")]
    0 references
    public string Final { get; set; }

    [JsonProperty("Grading_Scale")]
    0 references
    public string Grading_Scale { get; set; }

    [JsonProperty("BT")]
    0 references
    public List<string> BT { get; set; }

    [JsonProperty("Final_Exam")]
    0 references
    public string Final_Exam { get; set; }
}

1 reference
public class Content
{
    [JsonProperty("AtClass")]
    0 references
    public List<string> AtClass { get; set; }

    [JsonProperty("AtHome")]
    0 references
    public string AtHome { get; set; }
}

2 references
public class Detail
{
    [JsonProperty("Week")]
    0 references
    public string Week { get; set; }

    [JsonProperty("Content")]
    0 references
    public Content Content { get; set; }

    [JsonProperty("CDR")]
    0 references
    public List<string> CDR { get; set; }
}
```

Exams.cs

```
public class Exams
{
    0 references
    public ObjectId _id { get; set; }

    [JsonProperty("Exam")]
    0 references
    public List<Exam> Exam { get; set; }
}
1 reference
public class Exam
{
    [JsonProperty("ID_Test")]
    0 references
    public string ID_Test { get; set; }

    [JsonProperty("Exercise")]
    0 references
    public string Exercise { get; set; }

    [JsonProperty("Date")]
    0 references
    public string Date { get; set; }

    [JsonProperty("Tool")]
    0 references
    public string Tool { get; set; }

    [JsonProperty("ID_CDR")]
    0 references
    public List<string> ID_CDR { get; set; }

    [JsonProperty("Percentage")]
    0 references
    public string Percentage { get; set; }
}
```

PPGDs.cs

```
3 references
public class PPGDs
{
    0 references
    public ObjectId _id { get; set; }

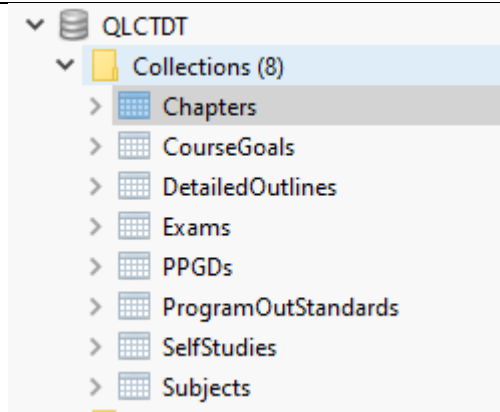
    [JsonProperty("PPGD")]
    0 references
    public List<PPGD> PPGD { get; set; }
}
1 reference
public class PPGD
{
    [JsonProperty("ID")]
    0 references
    public string ID { get; set; }

    [JsonProperty("Detail")]
    0 references
    public List<string> Detail { get; set; }
}
```

ProgramOutStandards.cs		<pre> public class ProgramOutStandards { // 0 references public ObjectId _id { get; set; } [JsonProperty("OutComes")] // 0 references public List<OutCome> OutComes { get; set; } } 1 reference public class OutCome { [JsonProperty("ID_CDR")] // 0 references public string ID_CDR { get; set; } [JsonProperty("Description_CDR")] // 0 references public string Description_CDR { get; set; } [JsonProperty("ID_CDIO")] // 0 references public List<string> ID_CDIO { get; set; } } </pre>
SelfStudies.cs		<pre> public class SelfStudies { public ObjectId _id { get; set; } [JsonProperty("SelfStudy")] // 0 references public List<SelfStudy> SelfStudy { get; set; } } 1 reference public class SelfStudy { [JsonProperty("ID")] // 0 references public string ID { get; set; } [JsonProperty("Details")] // 0 references public List<Detail> Details { get; set; } [JsonProperty("Request")] // 0 references public List<string> Request { get; set; } } 0 references public class Details { [JsonProperty("Read")] // 0 references public string Read { get; set; } [JsonProperty("Chapter")] // 0 references public List<string> Chapter { get; set; } } </pre>

Subjects.cs	<pre> public class Subjects { 0 references public ObjectId _id { get; set; } [JsonProperty("Course_Code")] 0 references public string Course_Code { get; set; } [JsonProperty("Course_Name")] 0 references public string Course_Name { get; set; } [JsonProperty("Credits")] 1 reference public string Credits { get; set; } [JsonProperty("Prerequisite")] 0 references public List<string> Prerequisite { get; set; } [JsonProperty("Chapter_ID")] 0 references public string Chapter_ID { get; set; } [JsonProperty("Goal_ID")] 0 references public string Goal_ID { get; set; } [JsonProperty("DetailOutline_ID")] 0 references public string DetailOutline_ID { get; set; } [JsonProperty("Exam_ID")] 0 references public string ExamID { get; set; } [JsonProperty("PPGD_ID")] 0 references public string PPGD_ID { get; set; } [JsonProperty("ProgramOutStandar_ID")] 0 references public string ProgramOutStandar_ID { get; set; } [JsonProperty("SelfStudy_ID")] 0 references public string SelfStudy_ID { get; set; } } </pre>
Kết nối MongoDB	<pre> using MongoDB.Bson; using Newtonsoft.Json; using System; using System.Collections.Generic; using System.Linq; using System.Text; using System.Threading.Tasks; </pre>

DATABASE



Collections Chapters

```
{
  "_id" : ObjectId("5fc509436184428b8096c1d5"),
  "Chapter" : [
    {
      "ID" : "Chương 1",
      "Name" : "Giới thiệu J2EE",
      "Detail" : [
        "1.1. Mô tả các thử thách trong việc phát triển các ứng dụng kinh doanh trên Web",
        "1.2. Giới thiệu về J2EE",
        "1.3. Giới thiệu các thành phần trong J2EE",
        "1.4. Những thuận lợi khi sử dụng J2EE."
      ]
    },
    {
      "ID" : "Chương 2",
      "Name" : "Tổng quan về Servlets và JSP",
      "Detail" : [
        "2.1. Giới thiệu về Servlet",
        "2.2. Xây dựng một ứng dụng Web động",
        "2.3. Tìm hiểu về mã Servlet",
        "2.4. Tìm hiểu vai trò của JSP",
        "Bài tập áp dụng"
      ]
    },
    {
      "ID" : "Chương 3",
      "Name" : "Cài đặt và cấu hình Tomcat Server",
      "Detail" : [
        "3.1. Tải về và cài đặt Tomcat Server",
        "3.2. Cấu hình môi trường triển khai ứng dụng Web trên Tomcat Server",
        "3.3. Triển khai một ứng dụng Servlet đơn giản",
        "3.4. Triển khai một ứng dụng JSP đơn giản",
        "3.5. Cấu hình một ứng dụng Web trên Tomcat Server",
        "Bài tập áp dụng"
      ]
    }
  ]
}
```

Collections CourseGoals

```
{
  "_id" : ObjectId("5fc509cf6184428b8096c206"),
  "Course_Goal" : [
    {
      "ID_Goal" : "G2",
      "Description_Goal" : "Khả năng phân tích và xây dựng một ứng dụng thông tin quản lý trên Web.",
      "ID_CDT" : [
        "2.1",
        "2.2"
      ],
      "ID_CDR" : [
        "G2.1",
        "G2.2"
      ]
    },
    {
      "ID_Goal" : "G1",
      "Description_Goal" : "Kiến thức về các điều khiển, chức năng và cách sử dụng các điều khiển trong việc xây dựng ứng dụng trên Web.",
      "ID_CDT" : [
        "1.2",
        "1.3"
      ],
      "ID_CDR" : [
        "G1.1",
        "G1.2",
        "G1.3"
      ]
    }
  ],
}
```

Collections
DetailedOutLine

```
"_id" : ObjectId("5fbc04bc87949980dedc76dd"),
"Description" : [
  "Xây dựng các ứng dụng web-based dựa trên 2 công nghệ Servlet và JSP trong J2EE framework.",
  "-Bộ thẻ JSTL, giúp cải tiến code cho các trang JSP.",
  "-Mô hình MVC, kết hợp 2 công nghệ Servlet và JSP "
],
"Goal" : [
  {
    "ID" : "G1",
    "CDR" : [
      "G1.1",
      "G1.2",
      "G1.3"
    ]
  },
  {
    "ID" : "G2",
    "CDR" : [
      "G2.1",
      "G2.2"
    ]
  },
  {
    "ID" : "G3",
    "CDR" : [
      "G3.1",
      "G3.2"
    ]
  }
],
}
```

Collections Exams	<pre>"_id" : ObjectId("5fc50a506184428b8096c234"), "Exam" : [{ "ID_Test" : "BT#2", "Exercise" : "Lập nhóm sinh viên (<=3 sinh viên). Phân tích một hệ thống thông tin quản lý bán hàng đơn giản và tiến hành cài đặt dữ liệu cho nhóm.", "Date" : "Tuần 4", "Tool" : "Bài tập báo cáo theo nhóm", "ID_CDR" : ["G1.2", "G1.3", "G2.1"], "Percentage" : "5" }, { "ID_Test" : "BT#3", "Exercise" : "Thiết kế phân hệ quản trị, quản lý các danh mục cho ứng dụng hệ thống thông tin quản lý bán hàng của nhóm.", "Date" : "Tuần 5", "Tool" : "Bài tập báo cáo theo nhóm", "ID_CDR" : ["G1.2", "G1.3", "G2.1", "G2.2", "G3.2.6"], "Percentage" : "10" }],</pre>	
Collections PPGDs	<pre>"_id" : ObjectId("5fc50aa96184428b8096c260"), "PPGD" : [{ "ID" : "PPGD2", "Detail" : ["Thuyết trình", "Trình chiếu Powerpoint", "Thảo luận nhóm", "Hỏi đáp, đánh giá"] }, { "ID" : "PPGD1", "Detail" : ["Thuyết trình, thực hiện các thí dụ trên máy.", "Trình chiếu Powerpoint", "Tương tác hỏi đáp với sinh viên"] }]</pre>	
Collections ProgramOutStandars	<pre>{ "_id" : ObjectId("5fc50b046184428b8096c27a"), "OutComes" : { ("ID_CDR" : "G1.1", "Description_CDR" : "Biết cách sử dụng được và hiểu được cơ chế làm việc của servlet và JSP trong lập trình ứng dụng trên web.Trình bày và sử dụng được và các đối tượng quản lý trạng thái trong ứng dụng Web: Application, Session, Cookies, Request.Hiểu và sử dụng tốt công nghệ Servlet, JSP, và làm việc với các loại cơ sở dữ liệu qua JDBC.", "ID_CDIO" : { "1.2" }), ("ID_CDR" : "G1.2", "Description_CDR" : "Trình bày được cách thức kết hợp sử dụng của các loại thẻ trong lập trình ứng dụng trên web.", "ID_CDIO" : { "1.2" }), ("ID_CDR" : "G1.3", "Description_CDR" : "Phân biệt và so sánh được ưu điểm và nhược điểm của các phương pháp triển khai ứng dụng theo các mô hình đã học (page centric, three layers, MVC).", "ID_CDIO" : { "1.3" }) } },</pre>	

Collections SelfStudies	<pre> _id" : ObjectId("5fc50b686184428b8096c2a6"), "SelfStudy" : [{ "ID" : "VN1-2", "Details" : [{ "Read" : "Đọc thêm: [2] Marty Hall, Larry Brown...", "Chapter" : ["Chapter 1: An Overview of Servlet and JSP Technology.", "Chapter 2: Server Setup and Configuration"] }] }, { "Request" : ["Thực hiện các bài tập cuối chương", "Thực hiện các bài tập giảng viên giao thêm", "Viết báo cáo tóm tắt chương 1"] }], </pre>
Collections Subjects	<pre> /* 24 */ { "_id" : ObjectId("5fbde8a887949980dedc7305"), "Course_Code" : "WEPR330479", "Course_Name" : "Lập trình Web", "Credits" : "3(2+1)", "Prerequisite" : ["DASA230179", "DBSY230184", "OOPR230279"], "Chapter_ID" : "5fc509436184428b8096c1d5", "Goal_ID" : "5fc509cf6184428b8096c206", "DetailOutline_ID" : "5fbe04bc87949980dedc76dd", "Exam_ID" : "5fc50a506184428b8096c234", "PPGD_ID" : "5fc50aa96184428b8096c260", "ProgramOutStandar_ID" : "5fc50b046184428b8096c27a", "SelfStudy_ID" : "5fc50b686184428b8096c2a6" } </pre>

4. Những khó khăn

Tuần	Nội dung công việc
Tuần 1 (8/10)	Không có
Tuần 2 (14/10)	Không có
Tuần 3(21/10)	-Khó khăn trong việc tạo sơ đồ chức năng và mô hình quan hệ thực thể. Về cơ bản đã vượt qua và hoàn thành, mong nhận được ý kiến đóng góp của thầy để tụi em hoàn thiện một cách chính xác nhất.
Tuần 4&Tuần 5	-Khi nhập liệu xảy ra các xung đột giữa các bảng. Khó khăn trong việc xác định các dữ liệu có trong một bảng và các mối liên kết giữa các bảng => Phải chỉnh sửa databse nhiều lần
Tuần 6 & Tuần 7	-Làm sai yêu cầu đề tài → Phải làm lại từ đầu -Chưa từng sử dụng cơ sở dữ liệu nosql → Phải vừa nghiên cứu vừa làm → Kéo theo tiến độ thực hiện khá chậm và đi lệch với kế hoạch ban đầu xây dựng ra

5. Kế Hoạch tuần tiếp theo

Tuần	Nội dung công việc
Tuần 3 (21/10)	Thiết kế giao diện cơ bản, nghiên cứu database
Tuần 4(27/10)	Chỉnh sửa, hoàn thiện databse, thiết kế giao diện cơ bản
Tuần 6(11/10)	<ul style="list-style-type: none">• Hoàn thành việc nhập liệu cho các bảng trong database• Tiếp tục thiết kế giao diện

	<ul style="list-style-type: none"> • Tìm hiểu cách kết nối database
Tuần 8	<ul style="list-style-type: none"> • Báo cáo tiến độ tuần 6 & 7 ➔ Chờ ý kiến phản hồi từ thầy • Sửa lại những gì đã làm sau khi thầy góp ý(nếu có) • Đi vào xây dựng giao diện(đẩy nhanh tiến độ hoàn thành)