VIETNAMESE – GERMAN UNIVERSITY

DEPARTMENT OF COMPUTER SCIENCE

Frankfurt University of Applied Science

Faculty 2: Computer Science and Engineering

BACHELOR THESIS

Thesis topic:

Dashboard website to support recruitment consultancy for the International Training program at Ho Chi Minh University of Technology and Education in 2018

Author:

Tran Minh Thang

Submitted in partial fulfillment of the requirements for the degree of Bachelor Engineering in study program Computer Science, Vietnamese - German University, 2017

Supervisors:

Dr. Tran Thi Thu Huong

M.Sc. Nguyen Minh Dao

(This page is intentionally left blank)

Declaration of Authorship

I, TRAN MINH THANG, declare that this thesis titled, "Dashboard website to support recruitment consultancy for the International Training program at Ho Chi Minh University of Technology and Education in 2018" and the work presented in it are my own. I confirm that:

- This work was done wholly or mainly while in candidature for a Bachelor degree at this University.
- No part of this thesis has previously been submitted for a degree or any other qualification at this University or any other institution.
- Where I have consulted the published work of others, this is always clearly attributed.
- Where I have quoted from the work of others, the source is always given. With the exception of such quotations, this thesis is entirely my own work.
- I have acknowledged all main sources of help.

Signature: _____

Name: _____

Date: _____

Abstract

In this day and age, web application is used in many purpose – marketing, selling products, social network… It is becoming more and more popular. This thesis will mainly focus on building a dashboard to support recruitment consultancy for the International Training program at Ho Chi Minh University of Technology and Education in 2018.

Due to the purpose of the web application, I decided to build it as a dashboard. Dashboard is a single-page web application so that every data will be presented in the same page. Information can be found faster and easier in comparison needs.

The dashboard used technologies that are popular in building web application. For the front-end development, the dashboard use three core technologies – HTML5, CSS3 and JavaScript. For the back-end development, ASP.NET MVC 5 is used. There is also require a database to store data, the database is built as a data warehouse and data is stored SQL Server. Each technologies will be briefly explained in the thesis.

After nine weeks of this bachelor thesis, the dashboard is fully developed and ready to deploy. Everything works properly including the web page, main function, data retrieved from database, etc.

Acknowledgements

I would like to express my gratitude to my two advisors Msc. Minh Dao Nguyen for helping me choose my topic and advise on the content of the thesis, and Dr. Thi Thu Huong Tran for helping me modify my thesis.

I also want to give my thanks to my colleagues at HCMUTE – STC, where I spent my 15 weeks internship. The knowledge they provided to me during my internship helps me a lot in this thesis.

Last but not least, I want to send my sincere thanks all of the staff, my friends at VGU, my girlfriend's family, my family and anyone who gave their support throughout my journey at VGU and especially this final venture.

Table of Contents

Table of Figures

List of Acronyms

Ho Chi Minh University of Technology and Education – HCMUTE

International Education Exchange Center – IEEC

Hypertext Transfer Protocol – HTTP

Hyper Text Markup Language – HTML

Cascading Style Sheets – CSS

On-line Transaction Processing – OLTP

On-line Analytical Processing – OLAP

Model View Controller – MVC

SQL Server Integration Services – SSIS

Business Intelligent – BI

Extracting Transform Load – ETL

Chapter 1 Introduction

1.1 Preface

In the former days of client-server model, each application has its own client program. It serves as the user interface and must be installed separately on each user's computer. The code on the client-side which pre-installed in users' workstation needs to be upgraded whenever there is an upgrade to the code on the server-side. As the result, increase the cost of support and decrease productivity.

In contrast, web application [1] is written in standard web programming language such as HTML. HTML is then styled with CSS and added effect, become dynamic with the help of JavaScript. Web application can be considered as another form of client-server software where the client-side is downloaded whenever users access the web page using protocol such as HTTP. The client-side software updates each time the web page is visited. But web development is not just like nowadays easily, it takes a very long time to formation and evolve.

In the earlier days, web pages was provided to client as a static document but it still can be interactive by using number of pages. Nevertheless, every changes must be sent back to the server and reload the whole page.

In 1995, JavaScript was created by Netscape. Instead of sending back data to server to refresh the entire page, JavaScript allows programmer to add dynamic elements to the user interface. The embedded script can do tasks such as showing/ hiding parts of the page or validating user input.

In 1999, the term "web application" was first introduced but there were only JavaScript and XML at that time, people only knew about Ajax since 2005. One of the most famous web application using Ajax is Gmail, web pages are more and more dynamic and they don't need to download the whole page to store or retrieve data anymore.

Dashboard is basically a single-page web application. It only displays the most important information. Dashboard has uniform structure and every data will only be displayed in one page only so that users can find what they need in a blink of an eye.

1.2 About the project

About International Education Exchange Center

International Education Exchange Center is part of Ho Chi Minh University of Technology and Education. It was established in September 2006. The Center has following objectives:

- Manage UTE's joint-training programs.

- Seek opportunities for joint-training programs with high rating regional and international educational institutions.

- Create opportunities for Vietnamese students to get access to international programs and obtain internationally recognized degrees at low costs.

- Provide opportunities for UTE's staff to improve their foreign language skills, professional expertise, teaching methodology and evaluation skills following international standards.

- Provide benchmark standards for UTE's offices and departments in improving their organization and management of the educational process, within the common trend of integration and globalization.

The main partner of UTE is University of Sunderland (UK), which provides three joint-training programs: Bachelor's program in Business Management, Bachelor's program in Accounting and Financial Management and Bachelor's program in Electrical & Electronic Engineering.

Recruitment in 2018

In the school year of 2018, HCMUTE recruit 200 new students for the International Training programs by multiple ways:

- Based on National high school exam result.
- Based on student's score on grade 12.
- Student from other University with equivalent entry requirement.
- Internal exam: Mathematics, English and Literature.

Purpose of Dashboard

In order to support recruitment process, a dashboard is built. Its aim is to support recruitment department makes good decision by provide multi-dimensional information.

Requirements

- Sorting recruitment statistics by following criteria: year, campus, area, major and/or recruitment method.
- Sorting statistics about how students know about UTE by following criteria: year and/or area.

1.3 About the thesis

The introduction has given a brief description about web application and dashboard. It also gives an introduction about IEEC, the current recruitment status there, therefore, the reason why this dashboard is created and also its mission.

During the next chapter, the thesis will concentrate on every web development technology that have been used throughout the project.

Chapter 3 will explain about the data flow from database to web page. The View and Controller of the dashboard are also mentioned along with function, how functions work and how it affects users.

In chapter 4 - the conclusion describes the advantages and disadvantages of the dashboard, the difficulty and moreover future development of the dashboard.

Chapter 2 Development Technology

This chapter would give a fundamental knowledge of technologies that had been used in the dashboard. This dashboard is built using Model – View – Controller structure with data warehouse in the database side to store data that can be retrieved by request of the dashboard. This chapter will cover web programming language in front-end side follow up with back-end side and finally, the database.

2.1 Front-end development

In front-end side, several JavaScript frameworks like JQuery along with HTML5 and CSS3 are used to combine the power of all front-end tools. In addition, frameworks like Bootstrap are also used to standardize the CSS3 and create the responsive design. As the result, a rich and dynamic UI will be built and it will be fully responsive to work on any devices.

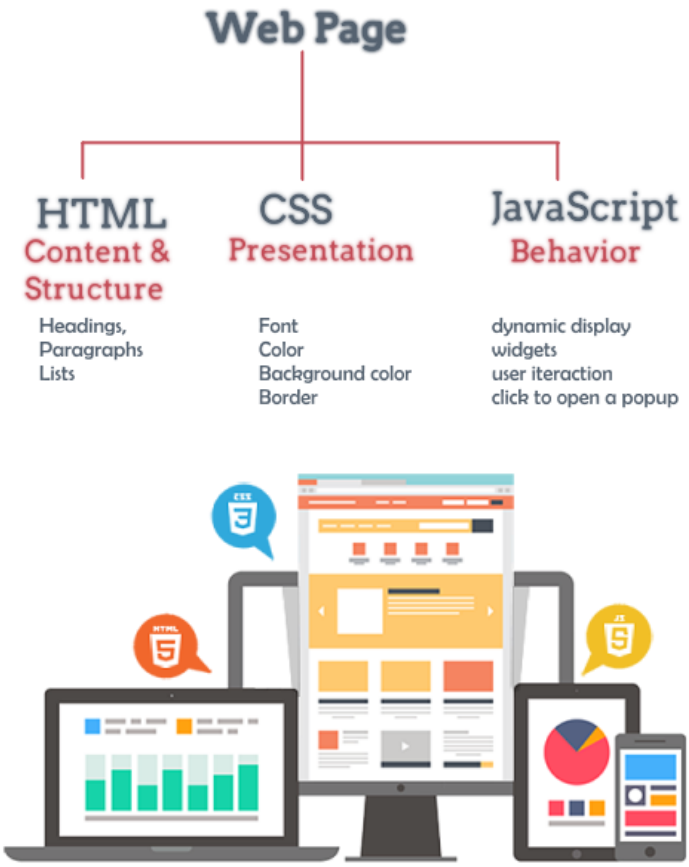


Figure 1 Websites are basically created in HTML, CSS and JavaScript

2.1.1 HTML

HTML [2] stands for Hyper Text Markup Language. It is the standard markup language for creating Web pages. It defines the structure of Web pages using markup. HTML uses elements to build blocks of HTML pages and they are represented by

tags. Tags label group of content for instance “heading”, “paragraph”, “table”, etc. HTML tags are not displayed by browsers, it is only used to extract the content of the page.

The page structure of an HTML page can be visualized:

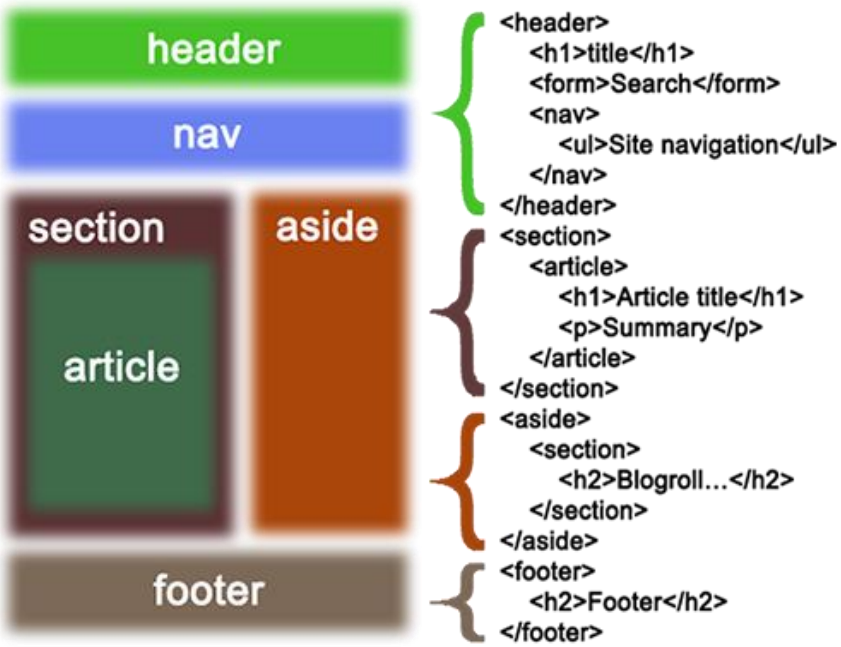


Figure 2 HTML structure of a basic website

2.1.2 CSS

CSS [3] stands for Cascading Style Sheets. It describes how HTML elements are to be displayed on screen, browser, or in other media. CSS can control the layout of multiple web pages at the same time so that it saves a lot of time. External stylesheets are stored in CSS files.

CSS is used to define styles for your web pages, including the design, layout and variations in display for different devices and screen sizes.

A CSS rule-set consists of a selector and a declaration block:

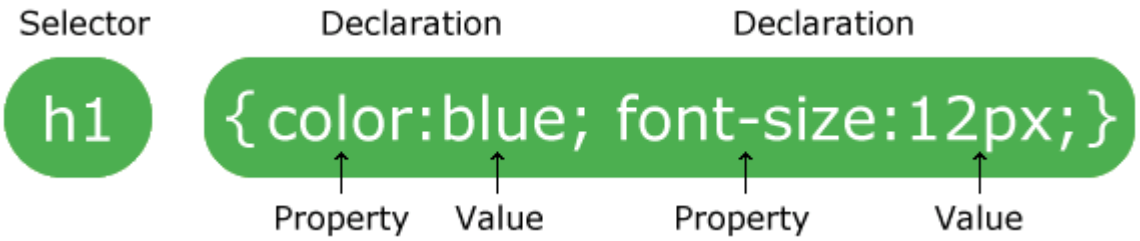


Figure 3 CSS syntax

The selector points to the HTML element that will be styled.

The declaration block contains one or more declarations separated by semicolons.

Each declaration includes a CSS property name and a value, separated by a colon.

A CSS declaration always ends with a semicolon, and declaration blocks are surrounded by curly braces.

2.1.3 JavaScript

JavaScript [4] is a lightweight programming language. It is an open and cross-platform which is designed for creating applications that using network. JavaScript is free and integrated in Java and HTML so that it is very easy to implement.

For the client-side, JavaScript is the most popular form of the language. In order the code could be understood by the browser, it should be included in or referenced by an HTML document. It leads to the fact that a web page does not need static HTML, it can have programs that interact with the user, control the browser, and dynamically create HTML content. There are many benefits of using JavaScript client-side instead of the server-side scripts. For instance, JavaScript can be used to check if the user has entered a valid e-mail address in a form field. The JavaScript code is executed whenever user submit a form, and it would be submitted to the Web Server only if all entries are validated. JavaScript can be used to detect users' activities for example click buttons, link navigation or other actions.

JavaScript takes many advantages:

- Less server interaction – user input can be validated before sending the page to the server. This will save server traffic so server is required to load less.
- Fast response to the visitors – users don't have to wait for a page reload to see if they have forgotten to enter something.
- Increase interaction – JavaScript can create interfaces that act in response to users' actions life hover mouse over it or trigger it by keyboard.
- Richer interfaces – JavaScript can be used to include such items as drag-and-drop components or sliders to give a rich interface to visitors.

2.2 Back-end development

ASP.NET [5] is very popular framework for developing Web Based Application. ASP.NET MVC is based on Model – View – Controller pattern that makes it more secure, robust and fast. ASP.NET MVC5 is the next generation technology of ASP.NET Webforms. There are many benefits of MVC over traditional asp.net web forms.

In ASP.NET MVC 5, Web pages, MVC and Web API were implemented separately, leading to duplication and inconsistencies. The programming model was as depicted in the figure below.

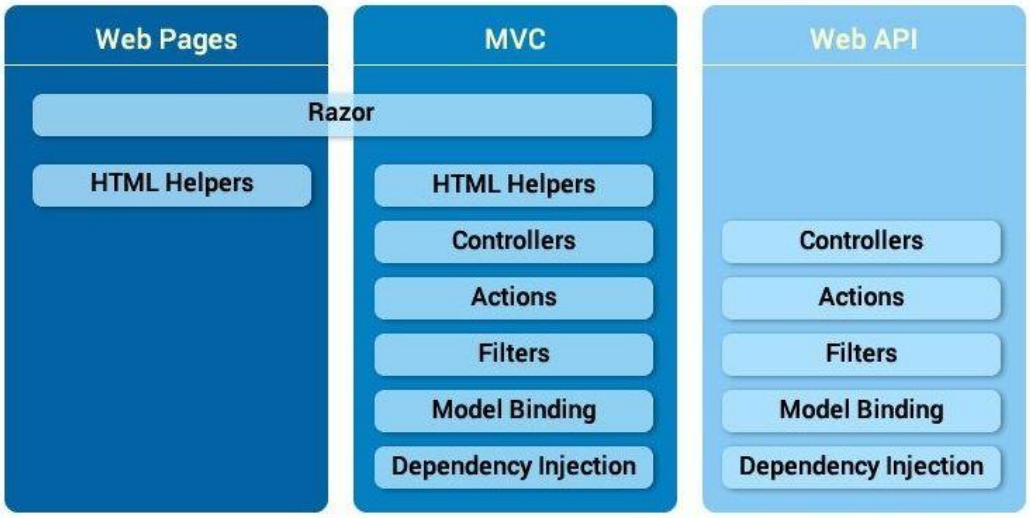


Figure 4 ASP.NET MVC5 structure

MVC is short form of Model, View and Controller.

- Model is responsible for keeping database operational logic like connecting and retrieving information from database.
- View is responsible for serving web page user interface to client.
- Controller keeps all the programming logics.

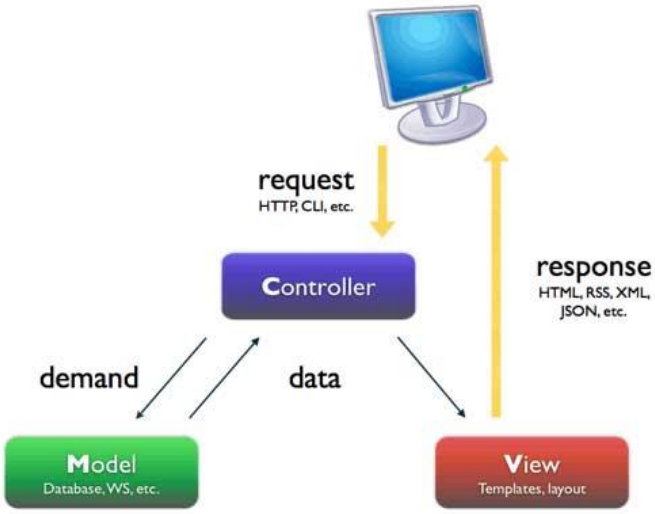


Figure 5 MVC model

MVC works in four steps.

Step 1: User hits the button on page or make a request through View page.

Step 2: The request comes to Controller.

Step 3: Depending on the user request, the Controller creates object of the Model. Then Model retrieve correspond information from database.

Step 4: After that, this data passed to View page through Model to serve the user.

The whole idea behind using the Model View Controller design pattern is that you maintain a separation of concerns. Your controller is no longer encumbered with a lot of ties to the ASP.NET runtime or ties to the ASPX page, which is very hard to test. You now just have a class with regular methods on it that you can invoke in unit tests to find out if that controller is going to behave correctly.

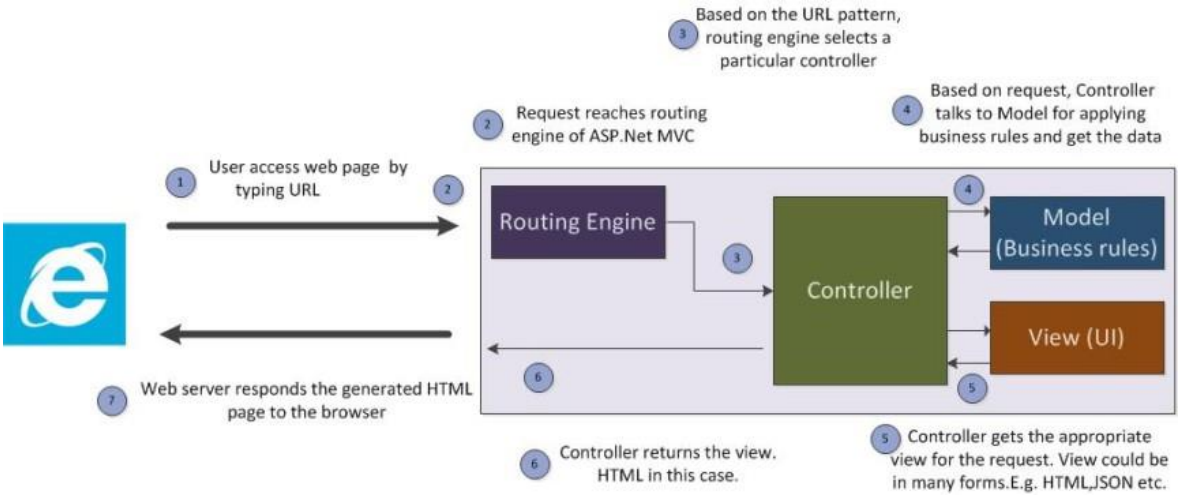


Figure 6 How MVC works

2.3 Database

2.3.1 OLTP vs. OLAP

Every IT systems can be divided into transactional (OLTP) and analytical (OLAP). [6] Each has different responsibility, OLTP systems transfer source data to data warehouses while OLAP systems support to analyze that.

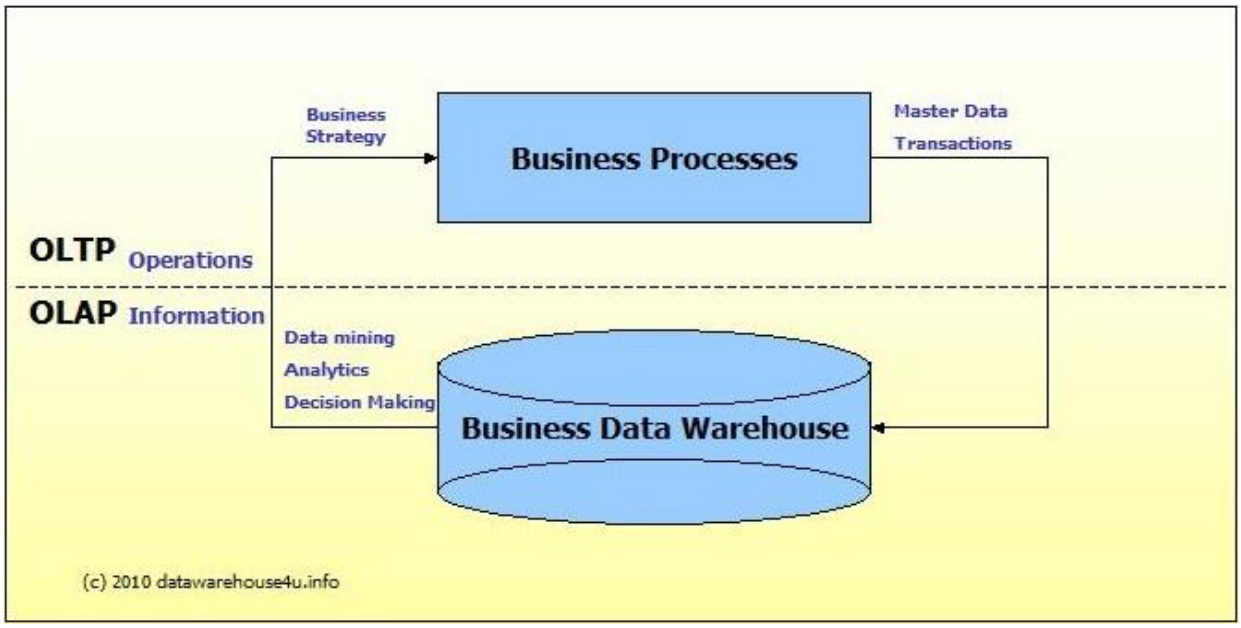


Figure 7 OLTP vs. OLAP

- OLTP (On-line Transaction Processing) is made of a large number of queries (INSERT, UPDATE, DELETE)
- OLAP (On-line Analytical Processing) is characterized by relatively low capacity of transactions. Queries are often very complex and involve aggregations. For OLAP systems a response time is an effectiveness measure. OLAP applications are widely used by Data Mining techniques. In OLAP database there is aggregated, historical data, stored in multi-dimensional schemas (usually star schema).

	OLTP System Online Transaction Processing (Operational System)	OLAP System Online Analytical Processing (Data Warehouse)
Source of data	Operational data; OLTPs are the original source of the data.	Consolidation data; OLAP data comes from the various OLTP Databases
Purpose of data	To control and run fundamental business tasks	To help with planning, problem solving, and decision support
What the data	Reveals a snapshot of ongoing business processes	Multi-dimensional views of various kinds of business activities

Inserts and Updates	Short and fast inserts and updates initiated by end users	Periodic long-running batch jobs refresh the data
Queries	Relatively standardized and simple queries Returning relatively few records	Often complex queries involving aggregations
Processing Speed	Typically very fast	Depends on the amount of data involved; batch data refreshes and complex queries may take many hours; query speed can be improved by creating indexes
Space Requirements	Can be relatively small if historical data is archived	Larger due to the existence of aggregation structures and history data; requires more indexes than OLTP
Database Design	Highly normalized with many tables	Typically de-normalized with fewer tables; use of star and/or snowflake schemas
Backup and Recovery	Backup religiously; operational data is critical to run the business, data loss is likely to entail significant monetary loss and legal liability	Instead of regular backups, some environments may consider simply reloading the OLTP data as a recovery method

2.3.2 Business Intelligence

Business Intelligence [7] is a technology infrastructure. Its aim is acquiring information from existing data in order to improving business processes. Business Intelligence has some typical infrastructure components: software solution for gathering, cleansing, integrating, analyzing and sharing data. Business Intelligence produces analysis and provides trustworthy information to help making effective and high quality business decisions.

The most common kinds of Business Intelligence systems are:

- **EIS** - Executive Information Systems
- **DSS** - Decision Support Systems
- **MIS** - Management Information Systems
- **GIS** - Geographic Information Systems
- **OLAP** - Online Analytical Processing and multidimensional analysis
- **CRM** - Customer Relationship Management

Business Intelligence systems based on Data Warehouse technology. Data Warehouse gathers information from a wide range of company's operational systems. Data loaded to Data Warehouse need to be sufficient and clean so it can produce reliable information.

2.3.3 SSIS (SQL Server Integration Services)

SSIS [8] is a platform for building enterprise-level data integration and workflow solutions. SSIS packages allow to move data from source to destination and then alter it if needed. SSIS is basically an ETL (will be described in session 2.3.5) tool. The responsibilities of ETL are already in its name: extraction, transformation and loading. Beside that, it can be used to maintain SQL Server databases automatically, update multidimensional cube data, etc.

2.3.4 ETL Operations (Extracting Transform Load)

E - Extracting → Getting data

T - Transform → Performing intermediate operations

L - Load → Load to destination

DB ↔ DB

DB ↔ File

File ↔ File

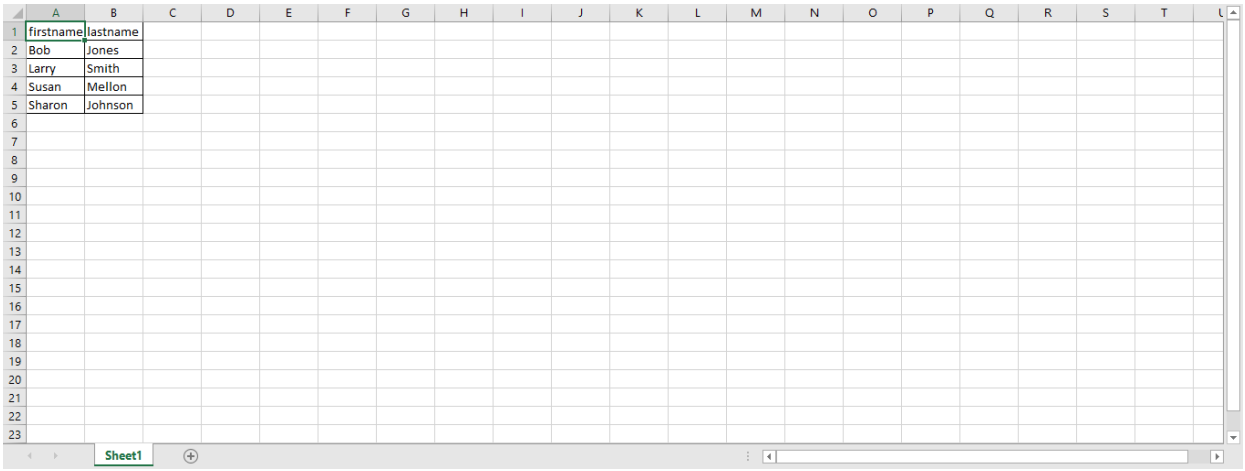
2.3.5 ETL process

ETL [9] stands for Extract, Transform and Load. In data warehousing, doing ETL means taking data out of the source systems and placing it into a data warehouse. ETL process has three main missions:

- **E**xtracting the data from source systems like SAP, ERP, etc. The data is then converted into one of the data warehouse format in order to ready the transforming process.
- **T**ransforming the data may include the following tasks:
    - applying business rules (so-called derivations, e.g., calculating new measures and dimensions),
    - cleaning (e.g., mapping NULL to 0 or "Male" to "M" and "Female" to "F" etc.),
    - filtering (e.g., selecting only certain columns to load),
    - splitting a column into multiple columns and vice versa,
    - joining together data from multiple sources (e.g., lookup, merge),
    - transposing rows and columns,
    - applying any kind of simple or complex data validation (e.g., if the first 3 columns in a row are empty then reject the row from processing)
- **L**oading the data into a data warehouse or data repository other reporting applications.

2.3.6 Import data to data warehouse

From the beginning, data given from customers are raw e.g. an excel file



	A	B
1	firstname	lastname
2	Bob	Jones
3	Larry	Smith
4	Susan	Mellon
5	Sharon	Johnson

Sheet1

Figure 8 An excel data source

It must be imported to database (SQL Server) through SQL Server Data Tools (SSDT) in order to use. It can be seen in the example that there are columns firstname and lastname so a table with those columns must be created first.

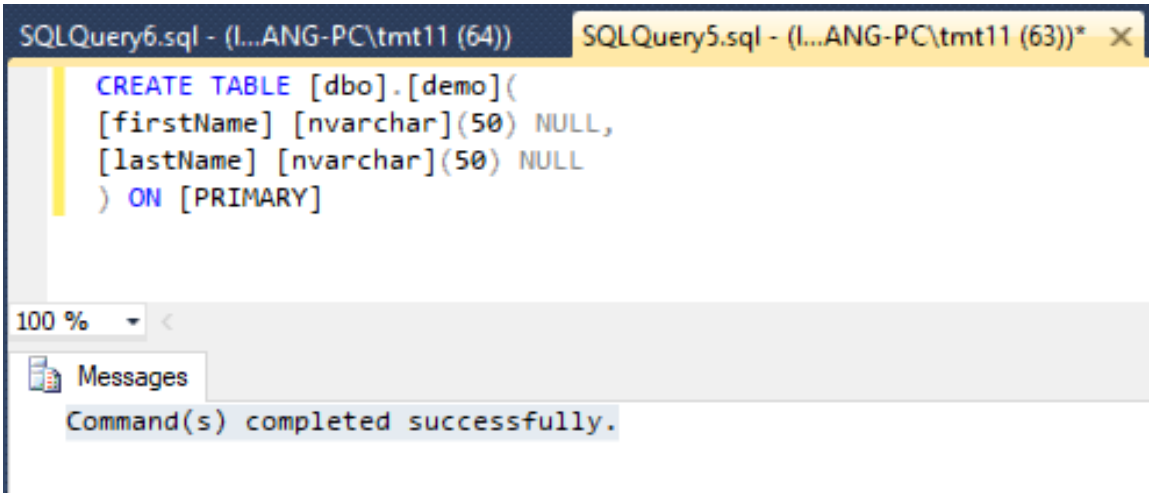


Figure 9 Create table with columns' name corresponding to data source file

In SSDT, create a Data Flow task.

Figure 10 Create data flow task

Inside the task, create an Excel Source and an OLE DB Destination. Because the aim is to import data from Excel to Database.

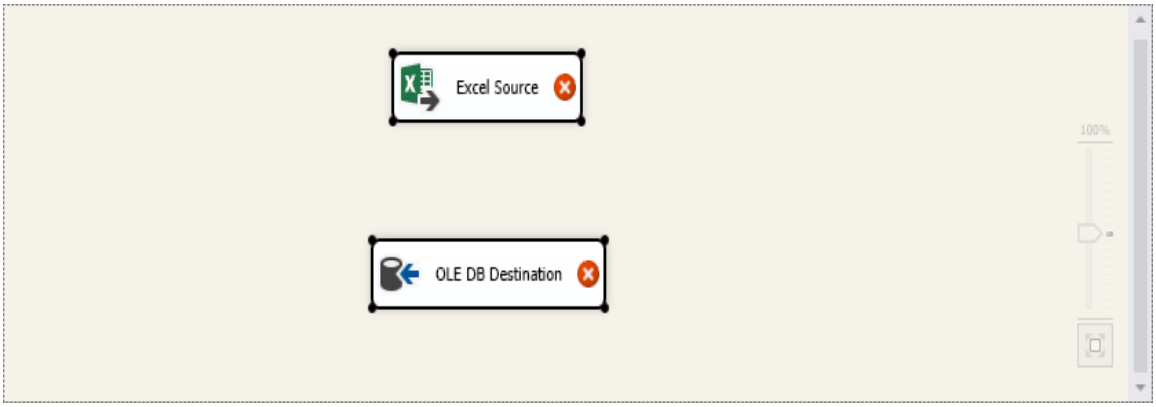


Figure 11 Inside data flow task

Choose the excel file above for Excel Source.

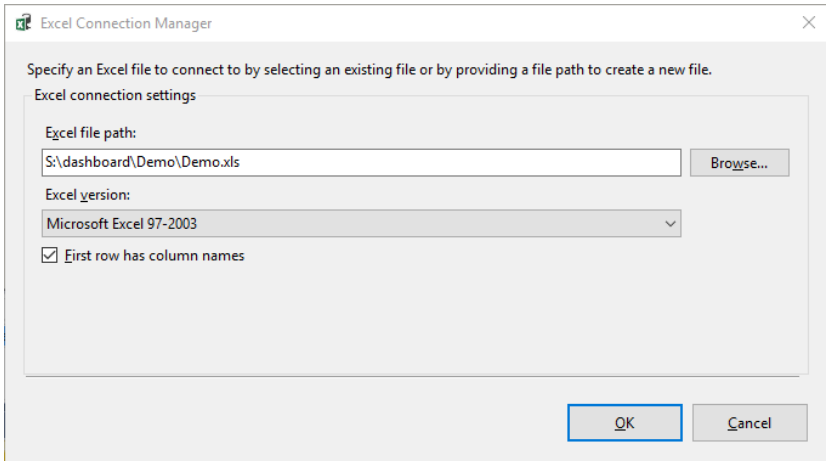


Figure 12 Setup for Excel data source

Choose the database and table created above for the OLE DB destination.

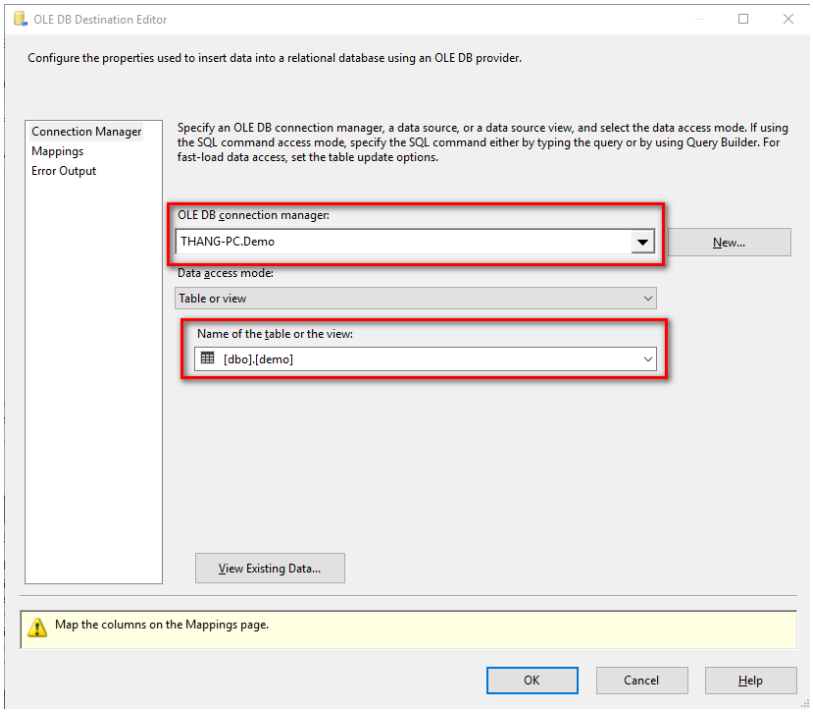


Figure 13 Setup for OLE DB destination

Finally, execute the Data Flow task. As can be seen, four rows have been imported to the database.

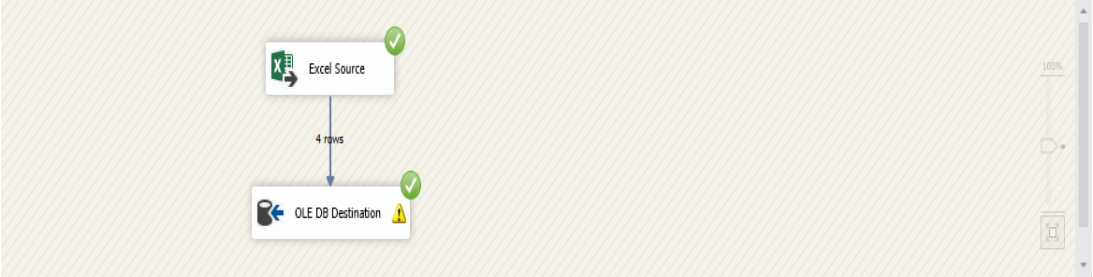


Figure 14 Data are transferred from source to destination

The result from database.

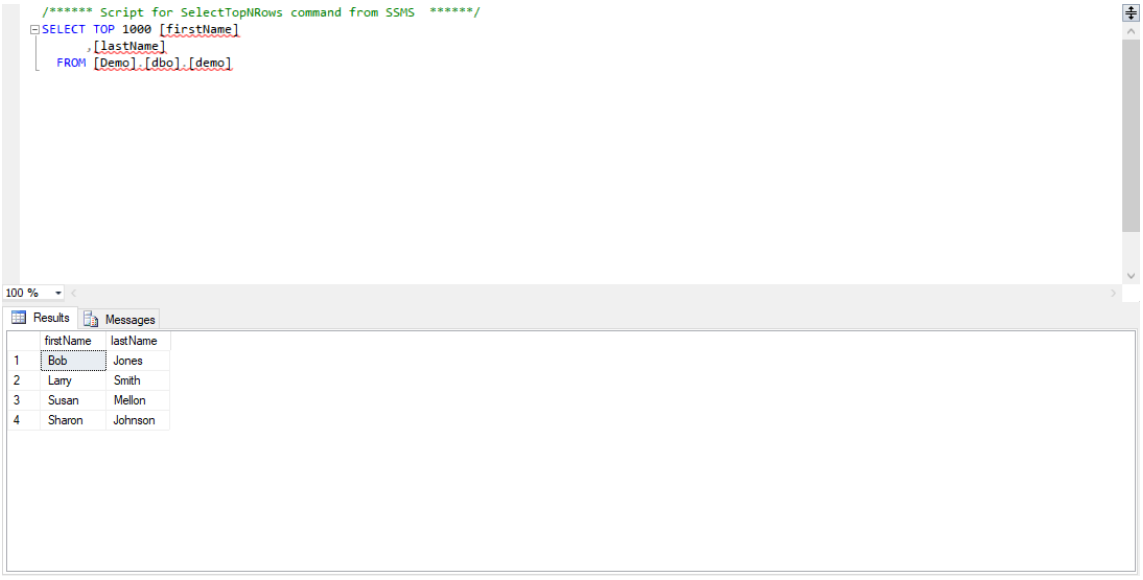


Figure 15 Data successfully transferred to data warehouse

2.3.7 Data warehouse

Data Warehouse [10] commonly converts the relational data model into special architectures. There are many schema models created for data warehousing but the most frequently used are:

- Star schema
- Snowflake schema

Each schema model takes advantage in different case. It should be decided based upon the analysis of project requirements, accessible tools and project team preferences.

Star Schema

The star schema architecture is the most simple data warehouse schema. It is called star schema because the diagram looks like a star, with points radiating from a center. The center of the star consists of fact table and the points of the star are the dimension tables. The fact tables in a star schema are typically in third normal form (3NF) while dimensional tables are de-normalized. Although the star schema is the most simple architecture, it is most commonly used nowadays and is recommended by Oracle.

Fact Tables

A fact table often has two types of columns:

- Foreign keys to dimension tables.

- Numeric facts that measures those foreign keys' data.

A fact table can contain fact's data on detail or aggregate level.

Dimension Tables

A dimension is a structure usually composed of one or more hierarchies that categorizes data. If a dimension does not have a hierarchies and levels it is called flat dimension or list. The primary keys of each of the dimension tables are part of the composite primary key of the fact table. Dimensional attributes help to describe the dimensional value. They are normally descriptive, textual values. Dimension tables are generally small in size then fact table.

Typical fact tables store data about sales while dimension tables data about geographic region (markets, cities), clients, products, times, channels.

The main characteristics of star schema:

- Simple structure → easy to understand.
- Great query effective → few tables to join.
- Relatively long time of loading data into dimension tables -> de-normalization, redundancy data caused that size of the table could be large.
- The most commonly used in the data warehouse implementations -> widely supported by a large number of business intelligence tools.

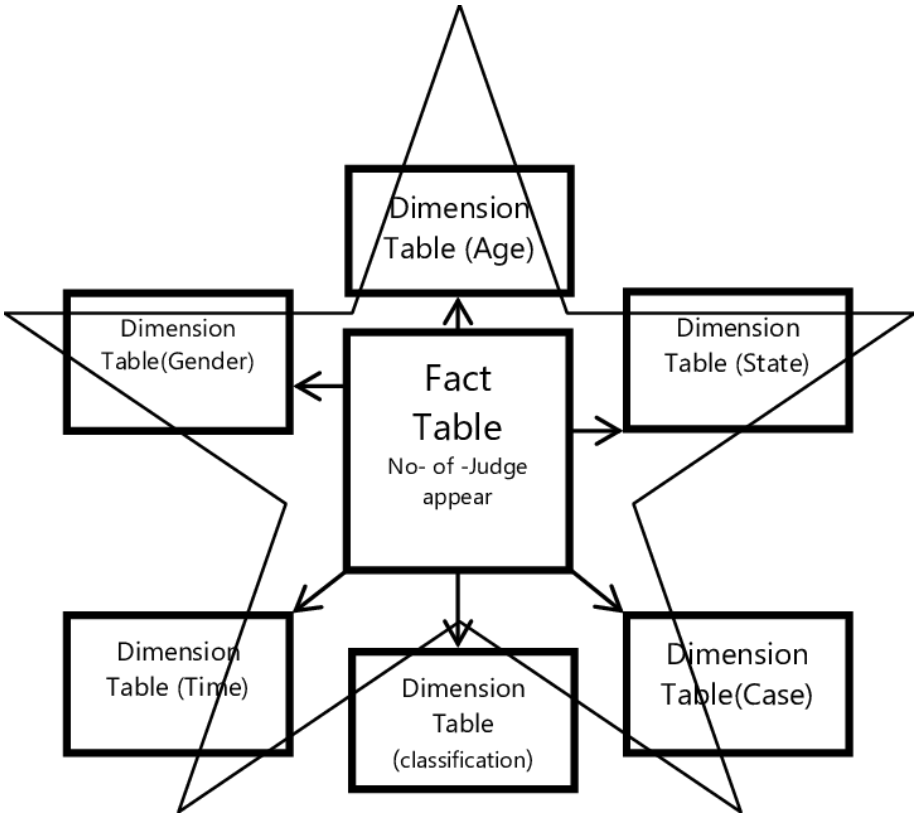


Figure 16 Star schema structure

2.3.8 Design database

The dashboard use SQL Server to manage data. Detail of each table in database:

Table "DimCoSo"

Column Name	Data type	Note
CoSoKey	bigint	Key of campus (PK)
TenCoSo	nvarchar	Name of campus
MaCoSo	nvarchar	Code of campus

Table "DimHinhThucTuyenSinh"

Column Name	Data type	Note
HinhThucKey	bigint	Key of recruitment method (PK)
TenHinhThuc	nvarchar	Name of recruitment method
MaHinhThuc	int	Code of recruitment method

Table "DimKhuVuc"

Column Name	Data type	Note
KhuVucKey	bigint	Key of area (PK)
TenKhuVuc	nvarchar	Name of area
MaKhuVuc	nvarchar	Code of area

*Area is which area students come from.

Table "DimNamTuyenSinh"

Column Name	Data type	Note
NamKey	bigint	Key of year (PK)
Nam	int	Year
TenNam	nvarchar	Name of year

Table "DimNganh"

Column Name	Data type	Note
NganhKey	bigint	Key of major (PK)
MaNganh	nvarchar	Name of major
TenNganh	nvarchar	Code of major

Table “DimNguon”

Column Name	Data type	Note
ID	bigint	ID of source (PK)
TenNguon	nvarchar	Name of source
MaNguon	nvarchar	Code of source

*Source is where students know about UTE from.

Table “FactDangKy”

Column Name	Data type	Note
ID	bigint	ID of register (PK)
SoLuong	int	Number of register
NamKey	bigint	Key of year (FK)
CoSoKey	bigint	Key of campus (FK)
NganhKey	bigint	Key of major (FK)
HinhThucKey	bigint	Key of recruitment method (FK)
KhuVucKey	bigint	Key of area (FK)

*Register is students that apply to UTE

Table “FactNhapHoc”

Column Name	Data type	Note
ID	bigint	ID of enroll (PK)
SoLuong	int	Number of enroll
NamKey	bigint	Key of year (FK)
CoSoKey	bigint	Key of campus (FK)
NganhKey	bigint	Key of major (FK)
HinhThucKey	bigint	Key of recruitment method (FK)
KhuVucKey	bigint	Key of area (FK)

*Enroll is students that choose UTE after their application for accepted

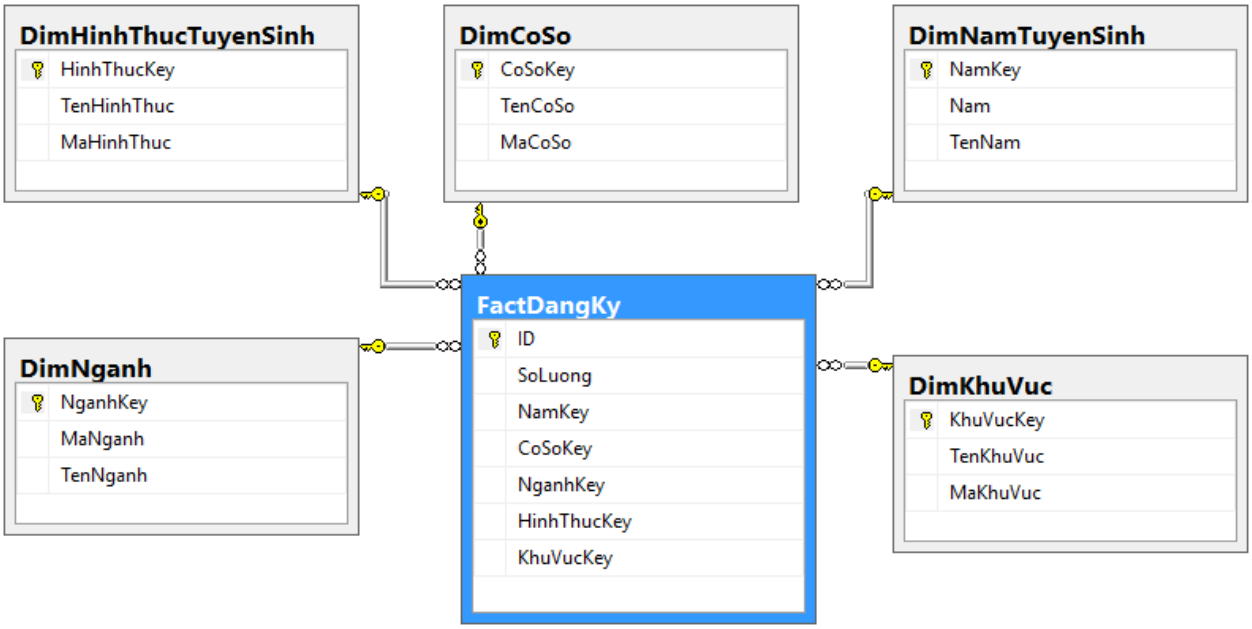


Figure 17 Star schema 1 – Register

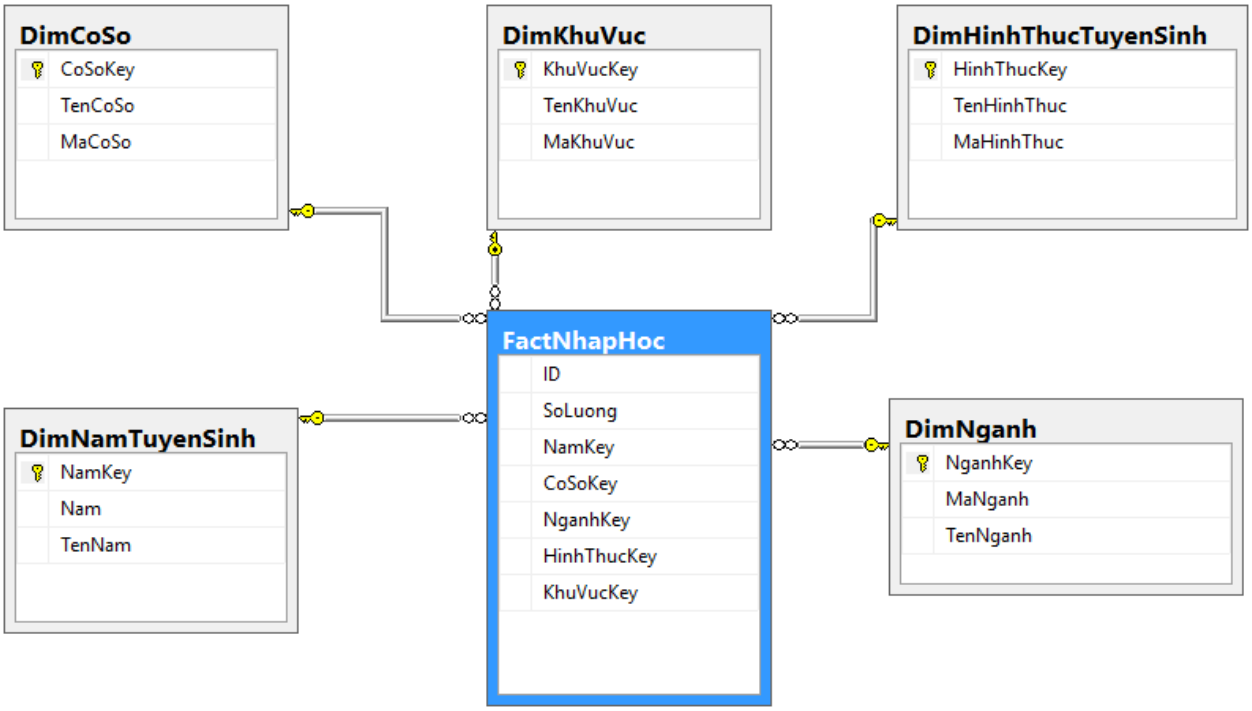
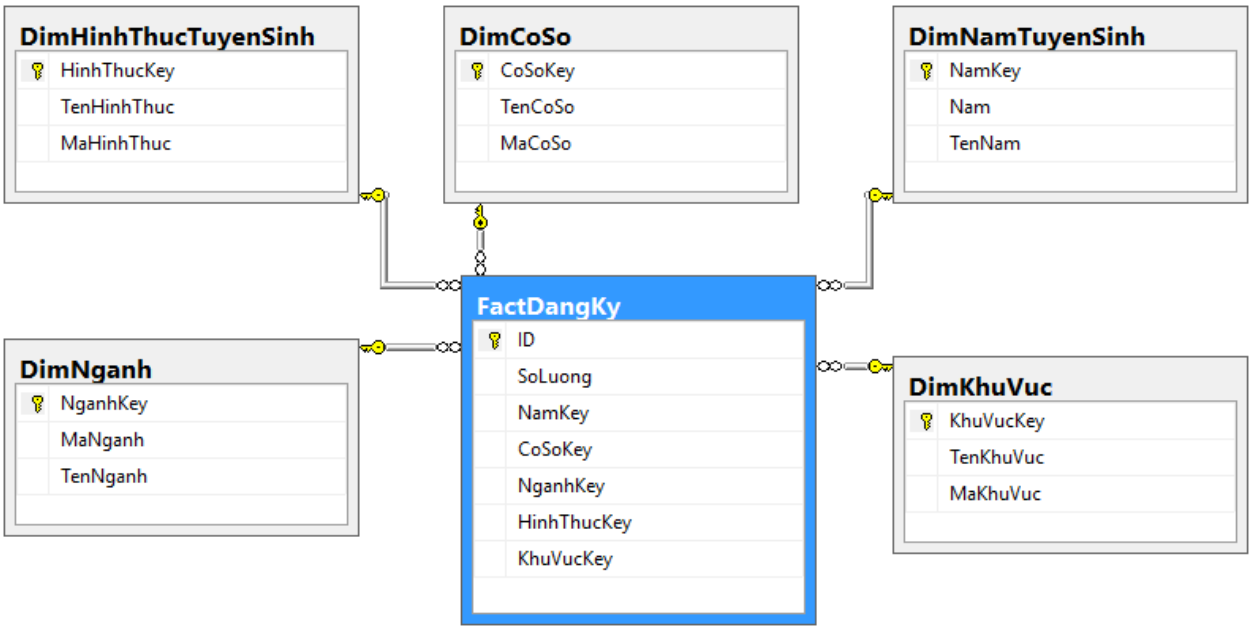


Figure 18 Star schema 2 – Enroll

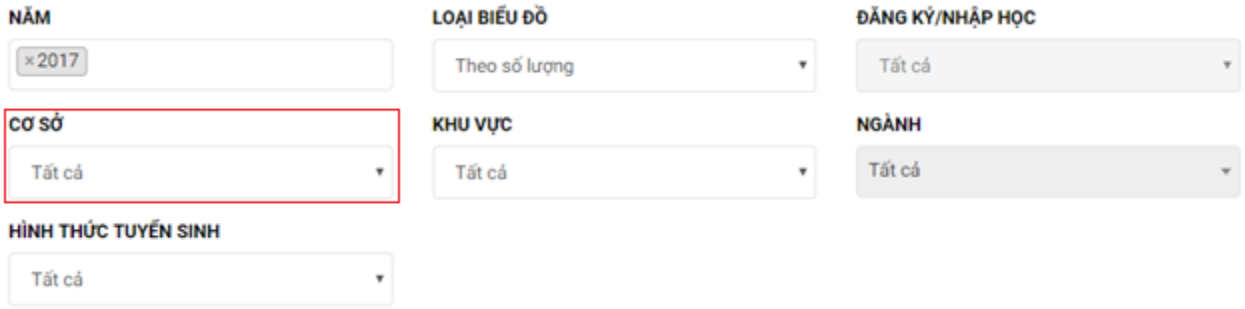
Chapter 3 Dashboard Visualization

3.1 Design data and convert to data model

In session 2.3.8, figure 17 and 18 are two star schemas that was introduced in session 2.3.7.



In figure 17, data tables have been created in MSSQL Server (all tables are listed in session 2.3.8). Let's dive deeper into each of them. In the dashboard, whenever a sorting criteria is chosen, "cơ sở" for instance.



In database, table DimCoSo in created.

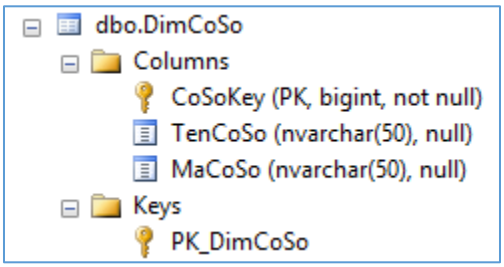


Figure 19 Table DimCoSo in database MSSQL Server

Whenever a "cơ sở" is chosen, it will query by dimension CoSo including CoSoKey, TenCoSo, MaCoSo fields. For each table in MSSQL Server, it will create a correspond class in model element of ASP.NET MVC web application.

```
namespace DashboardQHQT.Models
{
    2 references
    public class CoSoDTO
    {
        2 references | 0 exceptions
        public string MaCoSo { get; set; }
        2 references | 0 exceptions
        public string TenCoSo { get; set; }
        2 references | 0 exceptions
        public bool isSelected { get; set; }
    }
}
```

Figure 20 Creates CoSoDTO entity class corresponding to the DimCoSo data table

DTO stands for Data Transfer Object is a data entity object in Entity Framework 6. To be simple, data (from database) has to be converted into object (DTO) to be usable by the server.

From CoSoDTO class, continue to create a service method to manipulate DimCoSo table in MSSQL Server, which is the stored procedure in database, for high security and fast operation in purpose.

```
USE [TuyenSinhDashboardQHQT]
GO
ALTER PROCEDURE [dbo].[sp_dim_coso_select_all]
AS
BEGIN
    SELECT * from DimCoSo
END
```

Figure 21 Stored procedure

After that, an action method is created in controller element of ASP.NET MVC structure.

```
//2. lấy cơ sở
[HttpGet]
1 reference | 0 requests | 0 exceptions
public JsonResult GetCoSo()
{
    var lstCoSo = db.sp_dim_coso_select_all().Select(x => new CoSoDTO()
    {
        MaCoSo = x.MaCoSo,
        TenCoSo = x.TenCoSo,
        isSelected = false
    }).ToList();
    lstCoSo.Add(new CoSoDTO { MaCoSo = "all", TenCoSo = "Tất cả", isSelected = true });
    ViewBag.ListCoSo = lstCoSo;
    return Json(lstCoSo, JsonRequestBehavior.AllowGet);
}
```

Figure 22 Action method GetCoSo() in controller

Method GetCoSo() will return list of "cơ sở" with JSON data type as the result which is then retrieved by view element to display to the web page.


```
[{"MaCoSo":"TD","TenCoSo":"Thủ Đức","isSelected":false},{"MaCoSo":"PN","TenCoSo":"Phú Nhuận","isSelected":false},
{"MaCoSo":"all","TenCoSo":"Tất cả","isSelected":true}]
```


Figure 23 GetCoSo() Method

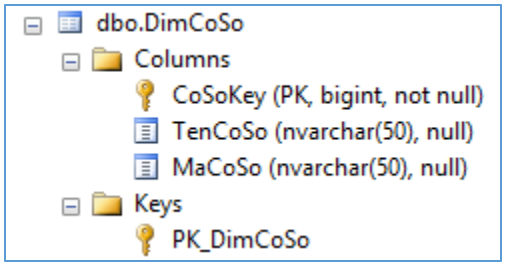
Next page is the visualization of what have been explained from the beginning of this session. The next few pages of this session is the visualization of another dimension: DimHinhThucTuyenSinh, DimKhuVuc, DimNamTuyenSinh, DimNganh, DimNguonTiepCan.

// sp_dim_coso_select_all ➔ List DimCoSo ←➔ Model

```
USE [TuyenSinhDashboardQHQT]
GO
ALTER PROCEDURE [dbo].[sp_dim_coso_select_all]
AS
BEGIN
    SELECT * from DimCoSo
END
```

TABLES ←➔ ENTITY

DimCoSo

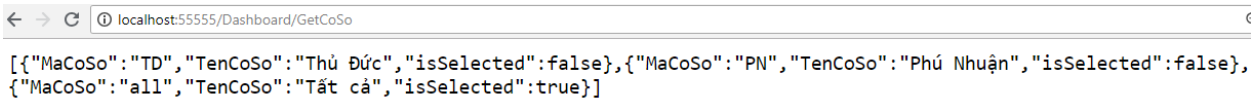


CoSoDTO

```
namespace DashboardQHQT.Models
{
    2 references
    public class CoSoDTO
    {
        2 references | 0 exceptions
        public string MaCoSo { get; set; }
        2 references | 0 exceptions
        public string TenCoSo { get; set; }
        2 references | 0 exceptions
        public bool isSelected { get; set; }
    }
}
```

```
//2. lấy cơ sở
[HttpGet]
1 reference | 0 requests | 0 exceptions
public JsonResult GetCoSo()
{
    var lstCoSo = db.sp_dim_coso_select_all().Select(x => new CoSoDTO()
    {
        MaCoSo = x.MaCoSo,
        TenCoSo = x.TenCoSo,
        isSelected = false
    }).ToList();
    lstCoSo.Add(new CoSoDTO { MaCoSo = "all", TenCoSo = "Tất cả", isSelected = true });
    ViewBag.ListCoSo = lstCoSo;
    return Json(lstCoSo, JsonRequestBehavior.AllowGet);
}
```

http://localhost:55555/Dashboard/GetCoSo

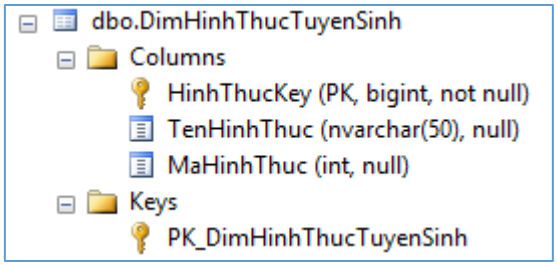


// sp_dim_hinhthuctuyensinh_select_all ➔ List DimHinhThucTuyenSinh ←➔ Model

```
USE [TuyenSinhDashboardQHQT]
GO
ALTER PROCEDURE [dbo].[sp_dim_hinhthuctuyensinh_select_all]
AS
BEGIN
    SELECT * from DimHinhThucTuyenSinh
END
```

TABLES ←➔ ENTITY

DimHinhThucTuyenSinh — HinhThucTuyenSinhDTO




```
namespace DashboardQHQT.Models
{
    2 references
    public class HinhThucTuyenSinhDTO
    {
        2 references | 0 exceptions
        public string MaHinhThucTuyenSinh { get; set; }
        2 references | 0 exceptions
        public string TenHinhThucTuyenSinh { get; set; }
        2 references | 0 exceptions
        public bool isSelected { get; set; }
    }
}
```

```
//5. lấy hình thức tuyển sinh
[HttpGet]
1 reference | 0 requests | 0 exceptions
public JsonResult GetHinhThucTuyenSinh()
{
    var lstHinhThucTuyenSinh = db.sp_dim_hinhthuctuyensinh_select_all().Select(x => new HinhThucTuyenSinhDTO()
    {
        MaHinhThucTuyenSinh = x.MaHinhThuc.ToString(),
        TenHinhThucTuyenSinh = x.TenHinhThuc,
        isSelected = false
    }).ToList();
    lstHinhThucTuyenSinh.Add(new HinhThucTuyenSinhDTO { MaHinhThucTuyenSinh = "all",
                                                        TenHinhThucTuyenSinh = "Tất cả", isSelected = true });
    ViewBag.ListHinhThucXetTuyen = lstHinhThucTuyenSinh;
    return Json(lstHinhThucTuyenSinh, JsonRequestBehavior.AllowGet);
}
```

http://localhost:55555/Dashboard/GetHinhThucTuyenSinh

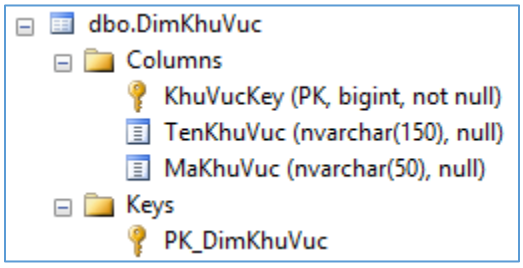

```
[{"MaHinhThucTuyenSinh":"1","TenHinhThucTuyenSinh":"Thi THPT quốc gia","isSelected":false},
{"MaHinhThucTuyenSinh":"2","TenHinhThucTuyenSinh":"Xét tuyển học bạ","isSelected":false},
{"MaHinhThucTuyenSinh":"3","TenHinhThucTuyenSinh":"Xét tuyển sinh viên các trường đại học","isSelected":false},
{"MaHinhThucTuyenSinh":"4","TenHinhThucTuyenSinh":"Thi tuyển 3 môn","isSelected":false},
{"MaHinhThucTuyenSinh":"all","TenHinhThucTuyenSinh":"Tất cả","isSelected":true}]
```

// sp_dim_khuvuc_select_all → List DimKhuVuc ←→ Model

```
USE [TuyenSinhDashboardQHQT]
GO
ALTER PROCEDURE [dbo].[sp_dim_khuvuc_select_all]
AS
BEGIN
    SELECT * from DimKhuVuc
END
```

TABLES ←→ ENTITY

DimKhuVuc KhuVucDTO




```
namespace DashboardQHQT.Models
{
    4 references
    public class KhuVucDTO
    {
        2 references | 0 exceptions
        public string MaKhuVuc { get; set; }
        2 references | 0 exceptions
        public string TenKhuVuc { get; set; }
        2 references | 0 exceptions
        public bool isSelected { get; set; }
    }
}
```

```
//3. khu vực
[HttpGet]
1 reference | 0 requests | 0 exceptions
public JsonResult GetKhuVuc()
{
    var lstKhuVuc = db.sp_dim_khuvuc_select_all().Select(x => new KhuVucDTO()
    {
        MaKhuVuc = x.MaKhuVuc,
        TenKhuVuc = x.TenKhuVuc,
        isSelected = false
    }).ToList();
    lstKhuVuc.Add(new KhuVucDTO { MaKhuVuc = "all", TenKhuVuc = "Tất cả", isSelected = true });
    ViewBag.ListKhuVuc = lstKhuVuc;
    return Json(lstKhuVuc, JsonRequestBehavior.AllowGet);
}
```

http://localhost:55555/Dashboard/GetKhuVuc

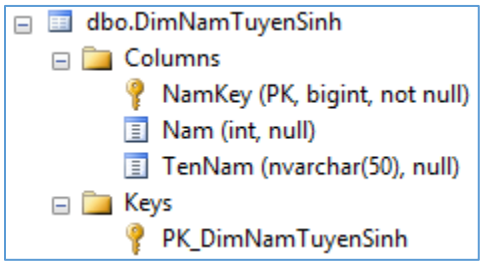

```
[{"MaKhuVuc":"26","TenKhuVuc":"Bến Tre","isSelected":false},{"MaKhuVuc":"27","TenKhuVuc":"Bình Định","isSelected":false},
{"MaKhuVuc":"28","TenKhuVuc":"Bình Dương","isSelected":false},{"MaKhuVuc":"29","TenKhuVuc":"Bình Thuận","isSelected":false},
{"MaKhuVuc":"30","TenKhuVuc":"BR - VT","isSelected":false},{"MaKhuVuc":"31","TenKhuVuc":"Cần Thơ","isSelected":false},
{"MaKhuVuc":"32","TenKhuVuc":"Đắk Lắk","isSelected":false},{"MaKhuVuc":"33","TenKhuVuc":"Đắk Nông","isSelected":false},
{"MaKhuVuc":"34","TenKhuVuc":"Đồng Nai","isSelected":false},{"MaKhuVuc":"35","TenKhuVuc":"Đồng Tháp","isSelected":false},
{"MaKhuVuc":"36","TenKhuVuc":"Gia Lai","isSelected":false},{"MaKhuVuc":"37","TenKhuVuc":"Lâm Đồng","isSelected":false},
{"MaKhuVuc":"38","TenKhuVuc":"Nghệ An","isSelected":false},{"MaKhuVuc":"39","TenKhuVuc":"Ninh Thuận","isSelected":false},
{"MaKhuVuc":"40","TenKhuVuc":"Phú Yên","isSelected":false},{"MaKhuVuc":"41","TenKhuVuc":"Quảng Nam","isSelected":false},
{"MaKhuVuc":"42","TenKhuVuc":"Quảng Ngãi","isSelected":false},{"MaKhuVuc":"43","TenKhuVuc":"Tây Ninh","isSelected":false},
{"MaKhuVuc":"44","TenKhuVuc":"Thừa Thiên Huế","isSelected":false},{"MaKhuVuc":"45","TenKhuVuc":"Tiền Giang","isSelected":false},
{"MaKhuVuc":"46","TenKhuVuc":"TPHCM","isSelected":false},{"MaKhuVuc":"47","TenKhuVuc":"Vĩnh Long","isSelected":false},
{"MaKhuVuc":"48","TenKhuVuc":"Vũng Tàu","isSelected":false},{"MaKhuVuc":"all","TenKhuVuc":"Tất cả","isSelected":true}]
```

// sp_dim_namtuyensinh_select_all ➔ List DimNamTuyenSinh ←➔ Model

```
USE [TuyenSinhDashboardQHQT]
GO
ALTER PROCEDURE [dbo].[sp_dim_namtuyensinh_select_all]
AS
BEGIN
    SELECT * from DimNamTuyenSinh
END
```

TABLES ←➔ ENTITY

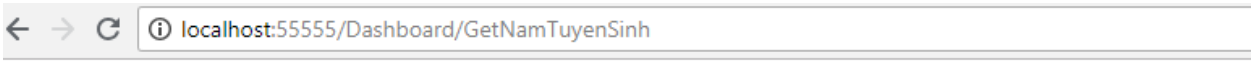
DimNamTuyenSinh YearDTO




```
namespace DashboardQHQT.Models
{
    4 references
    public class YearDTO
    {
        2 references | 0 exceptions
        public string Year { get; set; }
        2 references | 0 exceptions
        public bool isSelected { get; set; }
    }
}
```

```
//1. lấy năm tuyển sinh
[HttpGet]
0 references | 0 requests | 0 exceptions
public JsonResult GetNamTuyenSinh()
{
    var lstNamTuyenSinh = db.sp_dim_namtuyensinh_select_all().OrderByDescending(x => x.Nam).Select(x => new YearDTO()
    {
        Year = x.TenNam,
        isSelected = int.Parse(x.TenNam) == (DateTime.Now.Year - 1) ? true : false
    }).Take(3).ToList();
    ViewBag.ListNamTuyenSinh = lstNamTuyenSinh;
    return Json(lstNamTuyenSinh, JsonRequestBehavior.AllowGet);
}
```

http://localhost:55555/Dashboard/GetNamTuyenSinh



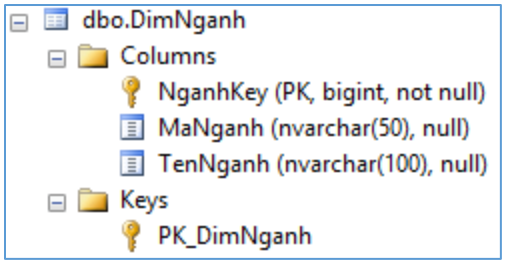

```
[{"Year":"2017","isSelected":true},
{"Year":"2016","isSelected":false},
{"Year":"2015","isSelected":false}]
```

// sp_dim_nganh_select_all ➔ List DimNganh ←➔ Model

```
USE [TuyenSinhDashboardQHQT]
GO
ALTER PROCEDURE [dbo].[sp_dim_nganh_select_all]
AS
BEGIN
    SELECT * from DimNganh
END
```

TABLES ←➔ ENTITY

DimNganh NganhDTO




```
namespace DashboardQHQT.Models
{
    7 references
    public class NganhDTO
    {
        10 references | 0 exceptions
        public string MaNganh { get; set; }
        9 references | 0 exceptions
        public string TenNganh { get; set; }
        2 references | 0 exceptions
        public bool isSelected { get; set; }
    }
}
```

```
//4. lấy ngành
[HttpGet]
1 reference | 0 requests | 0 exceptions
public JsonResult GetNganh()
{
    var lstNganh = db.sp_dim_nganh_select_all().Select(x => new NganhDTO()
    {
        MaNganh = x.MaNganh,
        TenNganh = x.TenNganh,
        isSelected = false
    }).ToList();
    lstNganh.Add(new NganhDTO { MaNganh = "all", TenNganh = "Tất cả", isSelected = true });
    ViewBag.ListNganh = lstNganh;
    return Json(lstNganh, JsonRequestBehavior.AllowGet);
}
```

http://localhost:55555/Dashboard/GetNganh

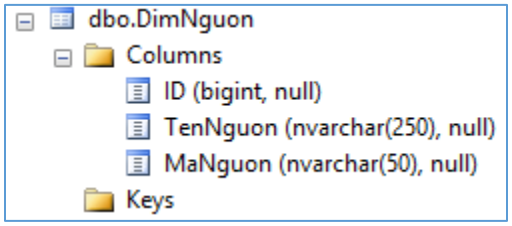

```
[{"MaNganh":"ĐĐT","TenNganh":"Kỹ thuật Điện - Điện tử","isSelected":false},
{"MaNganh":"KT\u0026QTTC","TenNganh":"Kế toán \u0026 Quản trị Tài chính","isSelected":false},
{"MaNganh":"QTKD","TenNganh":"Quản trị Kinh doanh","isSelected":false},
{"MaNganh":"all","TenNganh":"Tất cả","isSelected":true}]
```

// sp_dim_nguontiepcan_select_all ➔ List DimNguon ←➔ Model

```
USE [TuyenSinhDashboardQHQT]
GO
ALTER PROCEDURE [dbo].[sp_dim_nguontiepcan_select_all]
AS
BEGIN
    SELECT * from DimNguon
END
```

TABLES ←➔ ENTITY

DimNguon NguonTiepCanDTO




```
namespace DashboardQHQT.Models
{
    1 reference
    public class NguonTiepCanDTO
    {
        3 references | 0 exceptions
        public string MaNguonTiepCan { get; set; }
        5 references | 0 exceptions
        public string TenNguonTiepCan { get; set; }
    }
}
```

Finally, queries in the next page will join every dimension that users have chosen as the fact table (see figure 17 again, session 2.3.7 again for definition).

```sql
USE [TuyenSinhDashboardQHQT]
GO
ALTER PROCEDURE [dbo].[sp_fact_dangky_filter]
/*Fact needs Dimensions*/
 @Nam int,
 @MaHinhThuc nvarchar(50),
 @MaCoSo nvarchar(50),
 @MaNganh nvarchar(50),
 @MaKhuVuc nvarchar(50)
AS
BEGIN
    SELECT Sum (SoLuong) as TongSo
    FROM DimNamTuyenSinh as nam,
         DimCoSo as coso,
         DimHinhThucTuyenSinh as hinhthuc,
         DimKhuVuc as kv,
         DimNganh as nganh,
         FactDangKy as fact

    WHERE /* filter params*/
             ((@Nam is null) or (nam.Nam = @Nam))
         and ((@MaHinhThuc is null) or (hinhthuc.MaHinhThuc = @MaHinhThuc))
         and ((@MaCoSo is null) or (coso.MaCoSo = @MaCoSo))
         and ((@MaKhuVuc is null) or (kv.MaKhuVuc = @MaKhuVuc))
         and ((@MaNganh is null) or (nganh.MaNganh = @MaNganh))

         /* join conditions table */
         and fact.NamKey = nam.NamKey
         and fact.HinhThucKey = hinhthuc.HinhThucKey
         and fact.NganhKey = nganh.NganhKey
         and fact.KhuVucKey = kv.KhuVucKey
         and fact.CoSoKey = coso.CoSoKey

    GROUP BY
           CASE WHEN @Nam is not null then nam.Nam ELSE '' END,
           CASE WHEN @MaHinhThuc is not null then hinhthuc.MaHinhThuc ELSE '' END,
           CASE WHEN @MaCoSo is not null then coso.MaCoSo ELSE '' END,
           CASE WHEN @MaKhuVuc is not null then kv.MaKhuVuc  ELSE '' END,
           CASE WHEN @MaNganh is not null then nganh.MaNganh ELSE '' END

    ORDER BY  Sum (SoLuong)
END
```

```sql
USE [TuyenSinhDashboardQHQT]
GO
ALTER PROCEDURE [dbo].[sp_fact_dangky_tongtheonam]
@nam int
AS
BEGIN
    if @nam is null
        select sum (SoLuong) as TongSo from FactDangKy
    else
        select Nam, sum(SoLuong) as TongSo
        from DimNamTuyenSinh inner join FactDangKy on DimNamTuyenSinh.NamKey = FactDangKy.NamKey
        where Nam = @nam
        group by Nam
END
```

3.2 Main function of the dashboard (view and correspond controller)

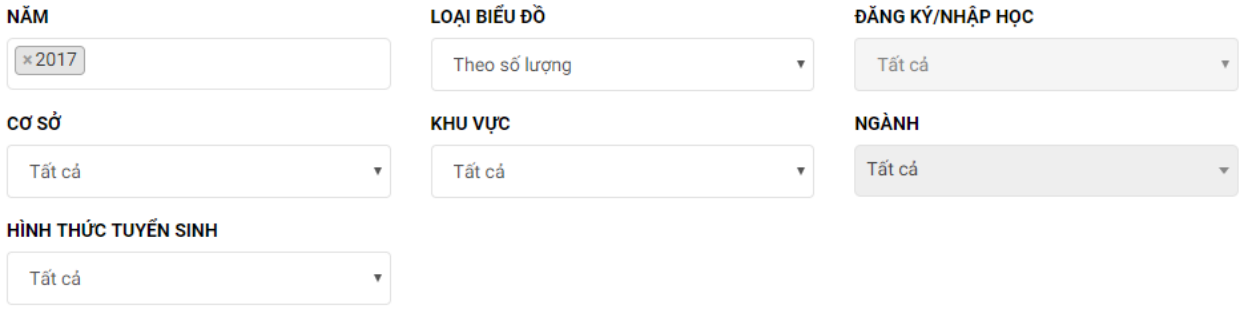
In this this session, every parts of the dashboard will be shown following by its view and controller elements in the ASP.NET MVC structure. The using definitions with brief explanations will be followed also.

View

```
<!--Đăng ký + nhập học 2015 - 2017-->
<div class="container-fluid">
    <div class="row" style="margin-top: 15px;">
        @foreach (var soLuong in ViewBag.SoLuongDKNH)
        {
            <div class="number-candidate">
                <div class="col-lg-4 col-md-6 col-sm-6">
                    <div class="card card-stats">
                        <div class="card-header" data-background-color="@soLuong.Color">
                            @soLuong.Nam
                        </div>
                        <div class="card-content">
                        </div>
                        <div class="card-footer">
                            <div class="stats text-center">
                                <div class="numberStudent">
                                    <p>Số lượng Thí sinh Đăng ký  </p>
                                    <p style="font-weight: bold;font-size: 14pt; color: #e53935">@soLuong.SoLuongDangKy @soLuong.GiaoDongDangKy</p>
                                </div>
                                <div class="numberStudent">
                                    <p>Số lượng Thí sinh Nhập học  </p>
                                    <p style="font-weight: bold;font-size: 14pt; color: #e53935">@soLuong.SoLuongNhapHoc @soLuong.GiaoDongNhapHoc</p>
                                </div>
                            </div>
                        </div>
                    </div>
                </div>
            </div>
        }
    </div>
```

Controller

```
0 references | 1 request | 0 exceptions
public ActionResult Index()
{
    //lấy danh sách các số liệu đăng ký, nhập học
    int indexColor = 0;
    List<SoLuongDangKyNhapHocDTO> lstSoLuongDKNH = new List<SoLuongDangKyNhapHocDTO>();
    List<YearDTO> lstYear = new List<YearDTO>();
    List<KhuVucDTO> lstKhuVuc = new List<KhuVucDTO>();

    //lấy danh sách các năm tuyển sinh
    var lstYearTemp = db.sp_dim_namtuyensinh_select_all().ToList();
    string[] ArrYear = lstYearTemp.OrderByDescending(x => x.Nam).Select(x => x.TenNam).Take(3).ToArray();

    lstYear = lstYearTemp.Select(x => new YearDTO()
    {
        Year = x.TenNam,
        isSelected = int.Parse(x.TenNam) == (DateTime.Now.Year - 1) ? true : false
    }).ToList();

    foreach (var year in ArrYear)
    {
        SoLuongDangKyNhapHocDTO dto = new SoLuongDangKyNhapHocDTO();
        dto.Color = ArrColor[indexColor++];
        dto.Nam = year;

        //lấy số lượng đăng ký
        var SoLuongDangKy = db.sp_fact_dangky_tongtheonam(int.Parse(year)).ToList();
        dto.SoLuongDangKy = int.Parse(SoLuongDangKy.Select(x => x.TongSo).FirstOrDefault().ToString());
        if ((indexColor - 1) != ArrYear.Count() - 1)
        {
            dto.GiaoDongDangKy = GetGiaoDongDangKyNhapHoc(int.Parse(year), "dk");
            dto.GiaoDongNhapHoc = GetGiaoDongDangKyNhapHoc(int.Parse(year), "nh");
        }
        else
        {
            dto.GiaoDongDangKy = "";
            dto.GiaoDongNhapHoc = "";
        }

        //lấy số lượng nhập học
        var SoLuongNhapHoc = db.sp_fact_nhaphoc_tongtheonam(int.Parse(year)).ToList();
        dto.SoLuongNhapHoc = int.Parse(SoLuongNhapHoc.Select(x => x.TongSo).FirstOrDefault().ToString());
        lstSoLuongDKNH.Add(dto);
    }
    ViewBag.ListNamTuyenSinh = lstYear;
    GetCoSo();
    GetKhuVuc();
    GetNganh();
    GetHinhThucTuyenSinh();
    ViewBag.SoLuongDKNH = lstSoLuongDKNH;

    return View();
}

2 references | 0 requests | 0 exceptions
public string GetGiaoDongDangKyNhapHoc(int Year, string Loai)
{
    string GiaoDong = "";
    double SoLuongNam = 0;
    double SoLuongNamSau = 1;

    if (Loai == "dk")
    {
        SoLuongNam = double.Parse(db.sp_fact_dangky_tongtheonam(Year).FirstOrDefault().TongSo.ToString());
        SoLuongNamSau = double.Parse(db.sp_fact_dangky_tongtheonam(Year - 1).FirstOrDefault().TongSo.ToString());
    }
    else
    {
        SoLuongNam = double.Parse(db.sp_fact_nhaphoc_tongtheonam(Year).FirstOrDefault().TongSo.ToString());
        SoLuongNamSau = double.Parse(db.sp_fact_nhaphoc_tongtheonam(Year - 1).FirstOrDefault().TongSo.ToString());
    }

    double GiaoDongSoLuong = SoLuongNam - SoLuongNamSau;
    float Percent = float.Parse(Math.Round(100 * Math.Abs(GiaoDongSoLuong) / SoLuongNamSau, 2).ToString());

    if (GiaoDongSoLuong < 0)
        GiaoDong = "(Giảm " + Percent + " % so với năm " + (Year - 1) + ")";
    else
        GiaoDong = "(Tăng " + Percent + " % so với năm " + (Year - 1) + ")";
    return GiaoDong;
}
```



View "Năm"

```
<div class="col-md-4">
    <div class="form-group">
        <!--Filter option-->
        <label>Năm</label>
        <select class="form-control" id="select2YearTK" onchange="Select2YearTKOnChange()" multiple>
            @foreach (var year in ViewBag.ListNamTuyenSinh)
            {
                <option value="@year.Year" selected="@(year.isSelected == true ? "selected" : null)">@year.Year</option>
            }
        </select>
    </div>
</div>
```

Controller "Năm"

```
//1. lấy năm tuyển sinh
[HttpGet]
0 references | 0 requests | 0 exceptions
public JsonResult GetNamTuyenSinh()
{
    var lstNamTuyenSinh = db.sp_dim_namtuyensinh_select_all().OrderByDescending(x => x.Nam).Select(x => new YearDTO()
    {
        Year = x.TenNam,
        isSelected = int.Parse(x.TenNam) == (DateTime.Now.Year - 1) ? true : false
    }).Take(3).ToList();
    ViewBag.ListNamTuyenSinh = lstNamTuyenSinh;
    return Json(lstNamTuyenSinh, JsonRequestBehavior.AllowGet);
}
```

Definition

```
function Select2YearTKOnChange() {
    var lstYear = $('#select2YearTK').val();
    //1. trường hợp chọn 1 năm
    if (lstYear.length == 1) {
        $('#select2LoaiBieuDo').prop('disabled', false);
        $('#select2DKNH').prop('disabled', 'disabled');
        $('#select2DKNH').find('option[value="all"]').attr("disabled", false);
        $('#select2DKNH').val("all");
        GetDataForChartAndGrid();
    }
    //2. trường hợp chọn nhiều năm
    else if (lstYear.length > 1) {
        $('#select2LoaiBieuDo').prop('disabled', 'disabled');
        $('#select2LoaiBieuDo').val("soluong");
        $('#select2DKNH').prop('disabled', false);
        $('#select2DKNH').val("DK");
        $('#select2DKNH').find('option[value="all"]').attr("disabled", "disabled");
        GetDataForChartAndGrid();
    }
}
```

Function Select2YearTKOnChange() will check if one or more years are chosen. If one year is chosen, it will disable "Đăng ký/Nhập học" and set it to "all" else it will disable "Loại biểu đồ" and set it to "theo số lượng"

View "Loại biểu đồ"

```
<div class="col-md-4">
    <div class="form-group">
        <label>Loại biểu đồ</label>
        <select class="form-control" id="select2LoaiBieuDo" onchange="Select2LoaiBieuDoOnChange()">
            <option value="soluong" selected>Theo số lượng</option>
            <option value="phantram">Theo phần trăm</option>
        </select>
    </div>
</div>
```

Definition

```
function Select2LoaiBieuDoOnChange() {
    var loaiBieuDo = $("#select2LoaiBieuDo").val();

    if (loaiBieuDo == "phantram") {
        $('#typeChart').hide();
        $('#select2DKNH').prop('disabled', false);
        $('#select2DKNH').val("DK");
        $('#select2DKNH').find('option[value="all"]').attr("disabled", "disabled");
    }
    else {
        $('#typeChart').show();
        $('#select2DKNH').val("all");
        $('#select2DKNH').prop('disabled', 'disabled');
        $('#select2DKNH').find('option[value="all"]').attr("disabled", false);
    }
    GetDataForChartAndGrid();
}
```

Function Select2LoaiBieuDoOnChange() will check if it is by number or percentage. If it is by number, it will disable "Đăng ký/Nhập học" and set it to "all".

View "Đăng ký/ nhập học"

```
<div class="col-md-4">
    <div class="form-group">
        <label>Đăng ký/Nhập học</label>
        <select class="form-control" id="select2DKNH" onchange="Select2DKNHOnChange()" disabled>
            <option value="all" selected>Tất cả</option>
            <option value="DK">Đăng ký</option>
            <option value="NH">Nhập học</option>
        </select>
    </div>
</div>
```

Definition

```
function Select2DKNHOnChange() {
    GetDataForChartAndGrid();
}
```

View “Cơ sở”

```
<div class="col-md-4">
    <div class="form-group">
        <label>Cơ sở</label>
        <select class="form-control" id="select2CoSoTK" onchange="Select2CoSoTKOnChange()">
            @foreach (var coSo in ViewBag.ListCoSo)
            {
                <option value="@coSo.MaCoSo"
                        selected="@(coSo.isSelected == true ? "selected" : null)">
                @coSo.TenCoSo</option>
            }
        </select>
    </div>
</div>
```

Controller “Cơ sở”

```
//2. lấy cơ sở
[HttpGet]
1 reference | 0 requests | 0 exceptions
public JsonResult GetCoSo()
{
    var lstCoSo = db.sp_dim_coso_select_all().Select(x => new CoSoDTO()
    {
        MaCoSo = x.MaCoSo,
        TenCoSo = x.TenCoSo,
        isSelected = false
    }).ToList();
    lstCoSo.Add(new CoSoDTO { MaCoSo = "all", TenCoSo = "Tất cả", isSelected = true });
    ViewBag.ListCoSo = lstCoSo;
    return Json(lstCoSo, JsonRequestBehavior.AllowGet);
}
```

Definition

```
function Select2CoSoTKOnChange() {
    GetDataForChartAndGrid();
}
```

View “Khu Vực”

```
<div class="col-md-4">
    <div class="form-group">
        <label>Khu vực</label>
        <select class="form-control" id="select2KhuVucTK" onchange="Select2KhuVucTKOnChange()">
            @foreach (var khuVuc in ViewBag.ListKhuVuc)
            {
                <option value="@khuVuc.MaKhuVuc" selected="@(khuVuc.isSelected == true ? "selected" : null)">@khuVuc.TenKhuVuc</option>
            }
        </select>
    </div>
</div>
```

Controller “Khu Vực”

```
//3. khu vực
[HttpGet]
1 reference | 0 requests | 0 exceptions
public JsonResult GetKhuVuc()
{
    var lstKhuVuc = db.sp_dim_khuvuc_select_all().Select(x => new KhuVucDTO()
    {
        MaKhuVuc = x.MaKhuVuc,
        TenKhuVuc = x.TenKhuVuc,
        isSelected = false
    }).ToList();
    lstKhuVuc.Add(new KhuVucDTO { MaKhuVuc = "all", TenKhuVuc = "Tất cả", isSelected = true });
    ViewBag.ListKhuVuc = lstKhuVuc;
    return Json(lstKhuVuc, JsonRequestBehavior.AllowGet);
}
```

Definition

```
function Select2KhuVucTKOnChange() {
    GetDataForChartAndGrid();
}
```

View “Ngành”

```
<div class="col-md-4">
    <div class="form-group">
        <label>Ngành</label>
        <select class="form-control" id="select2Nganh" onchange="SelectNganhOnChange()" disabled style="width: 100%">
            @foreach (var nganh in ViewBag.ListNganh)
            {
                <option value="@nganh.MaNganh" selected="@(nganh.isSelected == true ? "selected" : null)">@nganh.TenNganh</option>
            }
        </select>
    </div>
</div>
```

Controller “Ngành”

```
//4. lấy ngành
[HttpGet]
1 reference | 0 requests | 0 exceptions
public JsonResult GetNganh()
{
    var lstNganh = db.sp_dim_nganh_select_all().Select(x => new NganhDTO()
    {
        MaNganh = x.MaNganh,
        TenNganh = x.TenNganh,
        isSelected = false
    }).ToList();
    lstNganh.Add(new NganhDTO { MaNganh = "all", TenNganh = "Tất cả", isSelected = true });
    ViewBag.ListNganh = lstNganh;
    return Json(lstNganh, JsonRequestBehavior.AllowGet);
}
```

Definition

```
function SelectNganhOnChange() {
    GetDataForChartAndGrid();
}
```

View “Hình thức tuyển sinh”

```
<div class="col-md-10">
    <div class="row" style="margin-top: -10px">
        <div class="filter-dashboard">
            <div class="col-md-4">
                <div class="form-group">
                    <label>Hình thức tuyển sinh</label>
                    <select class="form-control" id="select2HTTS" onchange="Select2HTTSOnChange()">
                        @foreach (var htts in ViewBag.ListHinhThucXetTuyen)
                        {
                            <option value="@htts.MaHinhThucTuyenSinh" selected="@(htts.isSelected == true ? "selected" : null)">@htts.TenHinhThucTuyenSinh</option>
                        }
                    </select>
                </div>
            </div>
        </div>
    </div>
</div>
```

Controller “Hình thức tuyển sinh”

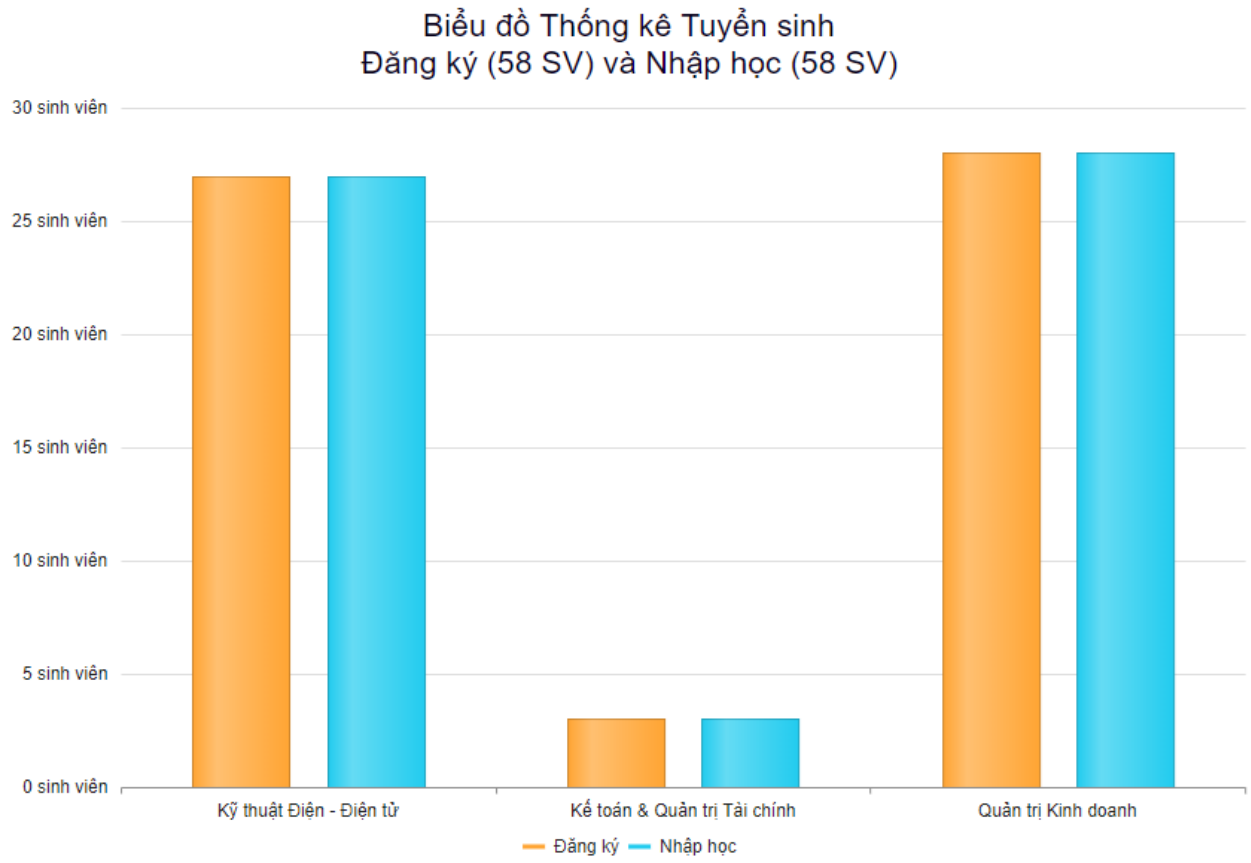
```
//5. lấy hình thức tuyển sinh
[HttpGet]
1 reference | 0 requests | 0 exceptions
public JsonResult GetHinhThucTuyenSinh()
{
    var lstHinhThucTuyenSinh = db.sp_dim_hinhthuctuyensinh_select_all().Select(x => new HinhThucTuyenSinhDTO()
    {
        MaHinhThucTuyenSinh = x.MaHinhThuc.ToString(),
        TenHinhThucTuyenSinh = x.TenHinhThuc,
        isSelected = false
    }).ToList();
    lstHinhThucTuyenSinh.Add(new HinhThucTuyenSinhDTO { MaHinhThucTuyenSinh = "all", TenHinhThucTuyenSinh = "Tất cả", isSelected = true });
    ViewBag.ListHinhThucXetTuyen = lstHinhThucTuyenSinh;
    return Json(lstHinhThucTuyenSinh, JsonRequestBehavior.AllowGet);
}
```

Definition

```
function Select2HTTSOnChange() {
    GetDataForChartAndGrid();
}
```

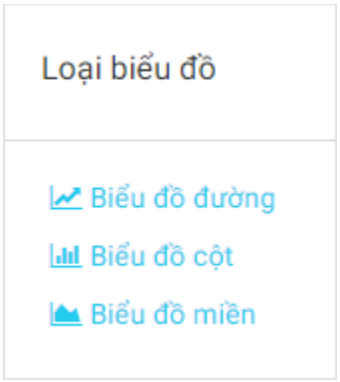
```
function GetDataForChartAndGrid() {
    kendo.ui.progress($("#container"), true);

    var Loai = $('#select2DKNH').val();
    var lstYear = $('#select2YearTK').val();
    var TypeChart = $('#select2LoaiBieuDo').val();
    var MaHinhThuc = $('#select2HTTS').val();
    var MaCoSo = $('#select2CoSoTK').val();
    var lstMaNganh = $('#select2Nganh').val();
    var MaKhuVuc = $('#select2KhuVucTK').val();

    $.ajax({
        url: RootUrl + 'Dashboard/GetDataForChartAndGrid',
        data: {
            Loai: Loai,
            lstYear: lstYear,
            TypeChart: TypeChart,
            MaHinhThuc: MaHinhThuc,
            MaCoSo: MaCoSo,
            lstMaNganh: lstMaNganh,
            MaKhuVuc: MaKhuVuc
        },
        type: 'POST',
        success: function (data) {
            if (TypeChart == "soluong") {
                var series = [];
                $.each(data.result, function (index, value) {
                    series = series.concat(value);
                });
                CreateChart(data, data.ArrNganh, series);
                CreateGrid(data.dataForGrid);
            }
            else {
                var dataPieChart = [];
                $.each(data.resultPieChart, function (index, value) {
                    dataPieChart = dataPieChart.concat(value);
                });
                CreatePieChartThongKe(dataPieChart);
                CreateGrid(data.dataForGrid);
            }
        },
        error: function (err) {
            console.log("Error in GetDataForChartAndGrid");
            console.log(err);
        }
    });
}
```

Function GetDataForChartAndGrid() is called whenever users change any sorting criteria. It will get all data and grid the chart again at once so that every time users change a criteria the chart will up to date immediately.



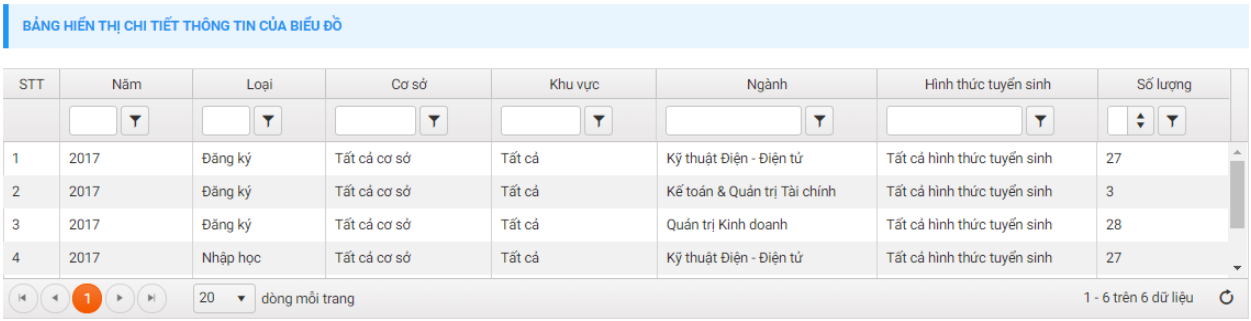
View Chart

```
<div class="col-md-10 col-sm-10 col-xs-10">
    <div class="demo-section k-content wide chart-wrapper">
        <div id="chart" style="width:100%"></div>
    </div>
</div>
```



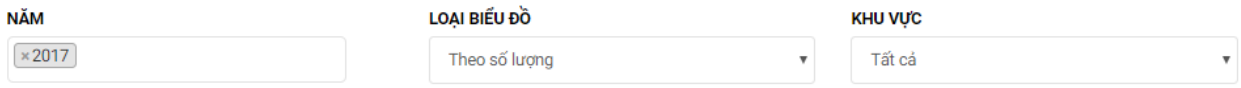
View Chart type

```
<div class="col-md-2 col-sm-2 col-xs-2">
    <div class="card card-category-chart" id="typeChart">
        <div class="header">
            <h5>Loại biểu đồ</h5>
        </div>
        <div class="content">
            <div class="choose-chart">
                <ul class="choose-chart-element">
                    <li>
                        <a onclick="LineChart()">
                            <i class="fa fa-line-chart"></i> Biểu đồ đường
                        </a>
                    </li>
                    <li>
                        <a onclick="ColumnChart()">
                            <i class="fa fa-bar-chart"></i> Biểu đồ cột
                        </a>
                    </li>
                    <li>
                        <a onclick="AreaChart()">
                            <i class="fa fa-area-chart"></i> Biểu đồ miền
                        </a>
                    </li>
                </ul>
            </div>
        </div>
    </div>
</div>
```



View Data table

```
<div class="col-md-12 col-sm-12 col-xs-12">
    <h6 style="margin-top: -25px;margin-bottom: 20px">Bảng hiển thị chi tiết thông tin của biểu đồ</h6>
    <div id="grid"></div>
</div>
```



View "Năm"

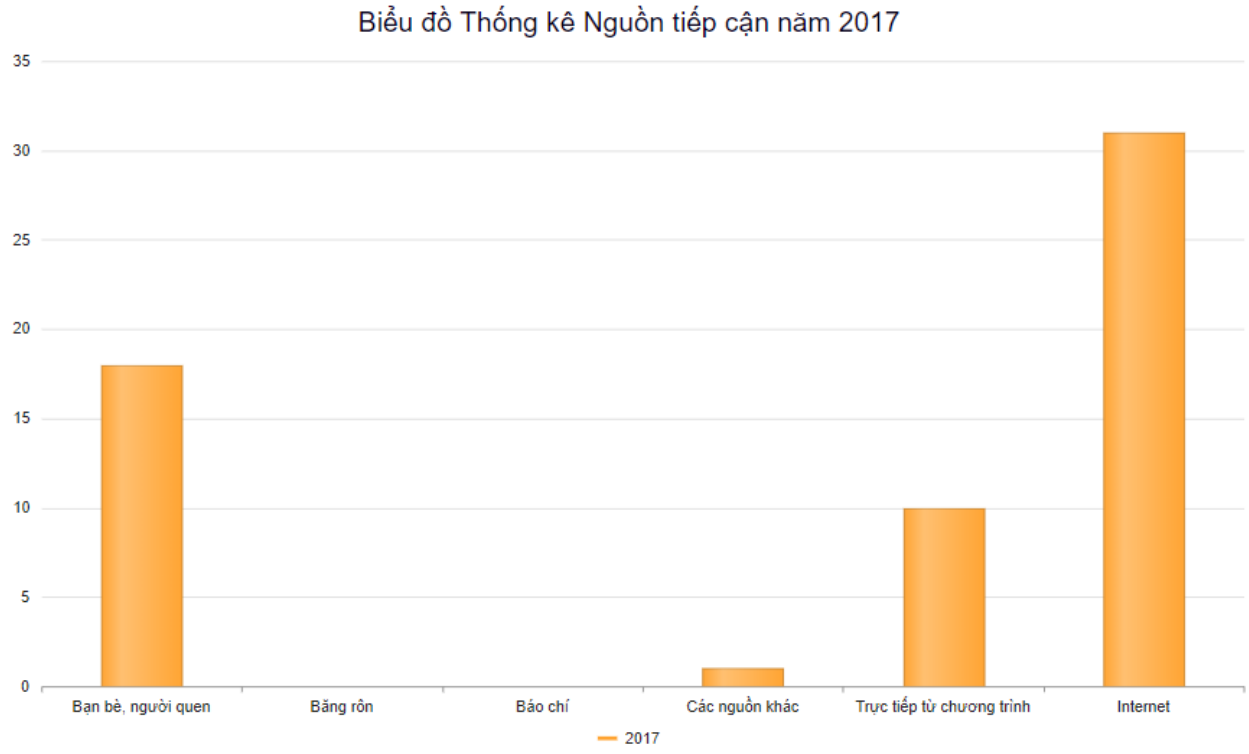
```
<div class="col-md-4">
    <div class="form-group">
        <label>Năm</label><br />
        <select class="form-control" id="select2Year" onchange="Select2YearOnChange()" multiple>
            @foreach (var year in ViewBag.ListNamTuyenSinh)
            {
                <option value="@year.Year" selected="@year.isSelected">@year.Year</option>
            }
        </select>
    </div>
</div>
```

View "Loại Biểu đồ"

```
<div class="col-md-4">
    <div class="form-group">
        <label>Loại biểu đồ</label>
        <select class="form-control" id="select2LoaiBieuDoNTC" onchange="Select2LoaiBieuDoNTCOnChange()">
            <option value="soluong" selected>Theo số lượng</option>
            <option value="phantram">Theo phần trăm</option>
        </select>
    </div>
</div>
```

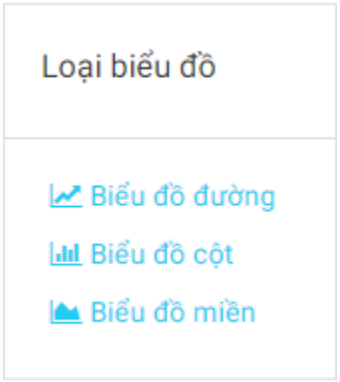
View “Khu vực”

```
<div class="col-md-4">
    <div class="form-group">
        <label>Khu vực</label>
        <select class="form-control" id="select2KhuVuc" onchange="Select2KhuVucOnChange()">
            @foreach (var kv in ViewBag.ListKhuVuc)
            {
                <option value="@kv.MaKhuVuc" selected="@kv.isSelected">@kv.TenKhuVuc</option>
            }
        </select>
    </div>
</div>
```



View chart

```
<div class="col-md-10 col-sm-10 col-xs-10">
    <div class="demo-section k-content wide chart-wrapper">
        <div id="chartNTC" style="width:100%"></div>
    </div>
</div>
```



View chart type

```
<div class="col-md-2 col-sm-2 col-xs-2">
    <div class="card card-category-chart" id="typeChartNTC">
        <div class="header">
            <h5>Loại biểu đồ</h5>
        </div>
        <div class="content">
            <div class="choose-chart">
                <ul class="choose-chart-element">
                    <li>
                        <a onclick="LineNTCChart()">
                            <i class="fa fa-line-chart"></i> Biểu đồ đường
                        </a>
                    </li>
                    <li>
                        <a onclick="ColumnNTCChart()">
                            <i class="fa fa-bar-chart"></i> Biểu đồ cột
                        </a>
                    </li>
                    <li>
                        <a onclick="AreaNTCChart()">
                            <i class="fa fa-area-chart"></i> Biểu đồ miền
                        </a>
                    </li>
                </ul>
            </div>
        </div>
    </div>
</div>
```

BẢNG HIỂN THỊ CHI TIẾT THÔNG TIN CỦA BIỂU ĐỒ

STT	Năm	Khu vực	Nguồn tiếp cận	Số lượng
1	2017	Tất cả khu vực	Bạn bè, người quen	18
2	2017	Tất cả khu vực	Băng rôn	0
3	2017	Tất cả khu vực	Báo chí	0
4	2017	Tất cả khu vực	Các nguồn khác	1

1 20 dòng mỗi trang 1 - 6 trên 6 dữ liệu

View data table

```
<div class="col-md-12 col-sm-12 col-xs-12">
    <h6 style="margin-top: -25px;margin-bottom: 20px">Bảng hiển thị chi tiết thông tin của biểu đồ</h6>
    <div id="gridNTC"></div>
</div>
```

Chapter 4 Conclusion

Through this thesis, I have learnt how to build a web page from interface to the backside, import data to SQL server and connect it to the web page, create charts based on retrieved data.

By building this web page as a dashboard, it receive all the benefits of dashboard. It is easy to understand and use. Only most important data appears so that users waste no time in finding information. The dashboard has fast and live changing charts with many comparison options and sorting criteria for users to have a good landscape of recruitment status of IEEC in the past few years.

On the other hand, the dashboard still need some features to make it better. The difficulty is time period for this thesis is only nine weeks and it takes too much time collecting data.

Besides that, there are some future development that can be possible for the dashboard. For instance, a tool that can analyze outcome data and give advice to the recruitment department so that they can have good recruit strategy in the future. Alternatively, a chatbot that can support in consulting.

Reference

[1] Wikipedia, Web application. [Online] Available at https://en.wikipedia.org/wiki/Web_application/ [Accessed 14 September 2018]

[2] w3schools, HTML5 Tutorial. [Online] Available at https://www.w3schools.com/html/ [Accessed 25 July 2018].

[3] w3schools, CSS3 Tutorial. [Online] Available at https://www.w3schools.com/css/ [Accessed 25 July 2018].

[4] w3schools, JavaScript, [Online] Available at https://www.w3schools.com/js/ [Accessed 25 July 2018].

[5] Microsoft, Getting Started with ASP.NET MVC 5, [Online] Available at https://docs.microsoft.com/en-us/aspnet/mvc/overview/getting-started/introduction/ [Accessed 29 July 2018].

[6] datawarehouse4u, OLTP vs. OLAP, [Online] Available at http://datawarehouse4u.info/OLTP-vs-OLAP.html [Accessed 5 August 2018].

[7] datawarehouse4u, What is business intelligence?, [Online] Available at http://datawarehouse4u.info/What-is-Business-Intelligence.html [Accessed 5 August 2018].

[8] Microsoft, Introduction to SQL Server Integration Services, [Online] Available at https://docs.microsoft.com/en-us/sql/integration-services/sql-server-integration-services?view=sql-server-2017 [Accessed 6 August 2018].

[9] datawarehouse4u, ETL process, [Online] Available at http://datawarehouse4u.info/ETL-process.html [Accessed 7 August 2018].

[10] datawarehouse4u, Data Warehouse Schema Architecture, [Online] Available at http://datawarehouse4u.info/Data-Warehouse-Schema-Architecture.html [Accessed 9 August 2018].