BỘ GIÁO DỤC VÀ ĐÀO TẠO
TRƯỜNG ĐẠI HỌC DÂN LẬP HẢI PHÒNG
-------o0o-------

# ĐỒ ÁN TỐT NGHIỆP

*NGÀNH CÔNG NGHỆ THÔNG TIN*

*HẢI PHÒNG 2013*

BỘ GIÁO DỤC VÀ ĐÀO TẠO

TRƯỜNG ĐẠI HỌC DÂN LẬP HẢI PHÒNG

-------o0o-------

# PHÁT TRIỂN ỨNG DỤNG TRÊN NỀN ECLIPSE

ĐỒ ÁN TỐT NGHIỆP ĐẠI HỌC HỆ CHÍNH QUY

Ngành: Công nghệ Thông tin

HẢI PHÒNG - 2013

BỘ GIÁO DỤC VÀ ĐÀO TẠO

TRƯỜNG ĐẠI HỌC DÂN LẬP HẢI PHÒNG

-------o0o-------

# PHÁT TRIỂN ỨNG DỤNG TRÊN NỀN ECLIPSE

ĐỒ ÁN TỐT NGHIỆP ĐẠI HỌC HỆ CHÍNH QUY

Ngành: Công nghệ Thông tin

Sinh viên thực hiện: Phạm Viết Mạnh

Giáo viên hướng dẫn:ThS Nguyễn Trịnh Đông

Mã số sinh viên: 1351010041

HẢI PHÒNG - 2013

BỘ GIÁO DỤC VÀ ĐÀO TẠO
TRƯỜNG ĐẠI HỌC DÂN LẬP HẢI PHÒNG

CỘNG HÒA XÃ HỘI CHỦ NGHĨA VIỆT NAM
*Độc lập - Tự do - Hạnh phúc*
-------*o0o*-------

# NHIỆM VỤ THIẾT KẾ TỐT NGHIỆP

Sinh viên: Phạm Viết Mạnh Mã SV: 1351010041

Lớp: CT1301 Ngành: Công nghệ Thông tin

Tên đề tài: Phát triển ứng dụng trên nền Eclipse

# NHIỆM VỤ ĐỀ TÀI

1. Nội dung và các yêu cầu cần giải quyết trong nhiệm vụ đề tài tốt nghiệp

a. Nội dung

- Tìm hiểu kiến trúc của Eclipse
- Tìm hiểu phát triển phần mềm theo h□ớng thành phần
- Phát triển phần mềm với Plugin
- Xây dựng ứng dụng thực nghiệm

b. Các yêu cầu cần giải quyết

Các phần mềm cần thiết để xây dựng

- Web Server: Xampp
- Hệ quản trị cơ sở dữ liệu MySQL
- Phần mềm Eclipse.

## CÁN BỘ HƯỚNG DẪN ĐỀ TÀI TỐT NGHIỆP

**Người hướng dẫn thứ nhất:**

Họ và tên:Nguyễn Trịnh Đông

Học hàm, học vị: Thạc Sĩ

Cơ quan công tác: Trường Đại Học Dân Lập Hải Phòng

Nội dung hướng dẫn: ……..

……

……

**Người hướng dẫn thứ hai:**

Họ và tên:……

Học hàm, học vị: ……

Cơ quan công tác: ……..

Nội dung hướng dẫn: …….

……

……

Đề tài tốt nghiệp được giao ngày tháng năm 2013

Yêu cầu phải hoàn thành trước ngày tháng năm 2013

| Đã nhận nhiệm vụ: Đ.T.T.N | Đã nhận nhiệm vụ: Đ.T.T.N |
|---|---|
| Sinh viên | Cán bộ hướng dẫn Đ.T.T.N |

ThS Nguyễn Trịnh Đông

**Hải Phòng, ngày .....tháng.....năm 2013**

HIỆU TRƯỞNG

***GS.TS.NGƯT Trần Hữu Nghị***

## PHẦN NHẬN XÉT TÓM TẮT CỦA CÁN BỘ HƯỚNG DẪN

1.Tinh thần thái độ của sinh viên trong quá trình làm đề tài tốt nghiệp:

.....

.....

.....

.....

.....

.....

2.Đánh giá chất lượng của đề tài tốt nghiệp (so với nội dung yêu cầu đã đề ra trong nhiệm vụ đề tài tốt nghiệp)

.....

.....

.....

.....

.....

.....

.....

3. Cho điểm của cán bộ hướng dẫn:

( Điểm ghi bằng số và chữ )

.....

.....

Ngày.....tháng.....năm 2013

Cán bộ hướng dẫn chính

*( Ký, ghi rõ họ tên )*

PHẦN NHẬN XÉT ĐÁNH GIÁ CỦA CÁN BỘ CHẤM PHẢN BIỆN ĐỀ TÀI TỐT NGHIỆP

**1. Đánh giá chất lượng đề tài tốt nghiệp (về các mặt như cơ sở lý luận, thuyết minh chương trình, giá trị thực tế, ...)**

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

**2. Cho điểm của cán bộ phản biện**

( *Điểm ghi bằng số và chữ* )

.....

Ngày.....tháng.....năm 2013

Cán bộ chấm phản biện

*( Ký, ghi rõ họ tên )*

# MỤC LỤC

# DANH MỤC CÁC HÌNH

# DANH SÁCH THUẬT NGỮ

| STT | **Thuật ngữ** | **Mô tả** |
|---|---|---|
| 1 | Eclipse Public License (EPL) | Giấy phép công cộng *Eclipse* |
| 2 | Integrated Development Enviroment (IDE) | Môi trường phát triển Java tích hợp Java |
| 3 | Plugin Development Environment (PDE) | Môi trường phát triển trình cắm thêm |
| 4 | Java Runtime Environment (JRE) | Môi trường thời gian thực |
| 5 | Java Developer Kit (JDK) | Bộ công cụ cho lập trình viên Java |
| 6 | The Platform runtime | Nền tảng thời gian thực thi |
| 7 | Extension point | Điểm mở rộng |
| 8 | Workbench | Bàn làm việc |

# LỜI CẢM ƠN

Trước hết em xin bày tỏ tình cảm và lòng biết ơn đối với Th.S Nguyễn Trịnh Đông – Khoa Công nghệ thông tin – Trường Đại học Dân Lập Hải Phòng, người đã dành cho em rất nhiều thời gian quý báu, trực tiếp hướng dẫn tận tình giúp đỡ, chỉ bảo em trong suốt quá trình làm đồ án tốt nghiệp.

Em xin chân thành cảm ơn tất cả các thầy cô giáo trong Khoa Công nghệ thông tin - Trường ĐHDL Hải Phòng, chân thành cảm ơn các thầy giáo, cô giáo tham gia giảng dạy và truyền đạt những kiến thức quý báu trong suốt thời gian em học tập tại trường, đã đọc và phản biện đồ án của em giúp em hiểu rõ hơn các vấn đề mình nghiên cứu, để em có thể hoàn thành đồ án này.

Em xin cảm ơn GS.TS.NGƯT Trần Hữu Nghị Hiệu trưởng Trường Đại học Dân lập Hải Phòng, Ban giám hiệu nhà trường, Bộ môn tin học, các Phòng ban nhà trường đã tạo điều kiện tốt nhất trong suốt thời gian học tập và làm tốt nghiệp.

Tuy có nhiều cố gắng trong quá trình học tập, trong thời gian thực tập cũng như trong quá trình làm đồ án nhưng không thể tránh khỏi những thiếu sót, em rất mong được sự góp ý quý báu của tất cả các thầy giáo, cô giáo cũng như tất cả các bạn để kết quả của em được hoàn thiện hơn.

Em xin chân thành cảm ơn.

Hải Phòng, tháng 12 năm 2012

Sinh viên

Phạm Viết Mạnh

# GIỚI THIỆU

Ngày nay, các hệ thống thông tin có khuynh hướng ngày càng phức tạp, quản lý hầu hết các lĩnh vực trong đời sống xã hội, được triển khai trên nhiều nền tảng công nghệ khác nhau. Việc phát triển những hệ thống như thế đòi hỏi mất nhiều chi phí, thời gian, nguồn lực tài chính cũng như con người. Một xu hướng xây dựng những hệ thống phức tạp dựa trên các thành phần phần mềm có sẵn đang được các nhà nghiên cứu quan tâm và nhiều công ty lớn trên thế giới đầu tư phát triển. Phát triển các phần mềm dựa trên thành phần có nhiều ưu điểm như: Giảm chi phí, rút ngắn thời gian triển khai, triển khai được trên nhiều nền tảng công nghệ khác nhau. Cụ thể, các thành phần phần mềm được nhiều hãng phần mềm cung cấp. Khi xây dựng hệ thống mới các nhà thiết kế chỉ cần kiểm tra xem thành phần phần mềm nào phù hợp với yêu cầu của hệ thống mới thì sẽ lựa chọn và ghép chúng lại với nhau. Các *Plugin* trong *Eclipse* là một hướng xây dựng phần mềm dựa trên thành phần.*Eclipse* cung cấp nhiều tính năng cho người phát triển bằng các ngôn ngữ lập trình phổ biến hiện nayJava, C, C++, Python, …. *Eclipse* trở thành lõi để phát triển ứng dụng. Việc mở rộng các tính năng của *Eclipse* bằng các *Plugin* đã đem lại cho *Eclipse* sức mạnh và tính thích nghi cao trong lĩnh vực phát triển phần mềm. Vì vậy, trong đợt tốt nghiệp này em chọn đề tài: ***"Phát triển ứng dụng trên nền Eclipse"***để tìm hiểu và nghiên cứu.

*Nội dung khóa luận gồm có các phần sau:*

**Chương 1:***Giới thiệu sự phát triển chung về phát triển phần mềm dựa trên thành phần, đặc biệt sự phát triển các Plugin trong Eclipse.*

**Chương 2:** *Trình bày về kiến trúc của Plugin trong Eclipse*

**Chương 3:** *Trình bày chương trình thử nghiệm sắp xếp thời khóa biểubằng plugin của eclipse.*

**Kết luận**

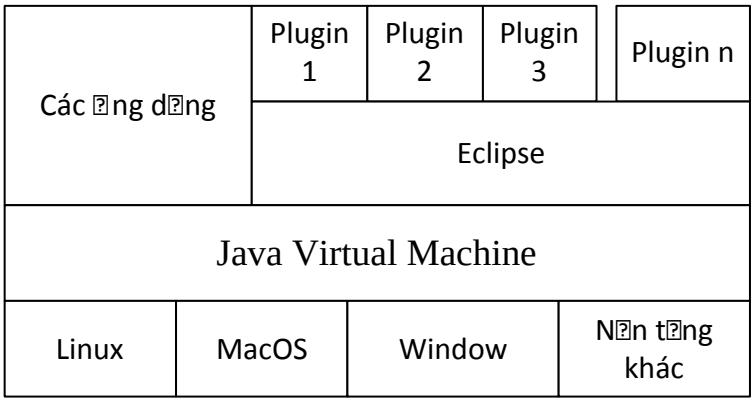
**Tài liệu tham khảo**

# CHƯƠNG 1: MỞ ĐẦU

Trong chương này, khóa luận trình bày về chương trình Eclipse, lịch sử phát triển của Eclipse và kiến trúc nền tảng của Eclipse.

## 1.1Giới thiệu Eclipse

*Eclipse* là một nền tảng có kiến trúc mở dựa trên Java, có thể mở rộng để phát triển các ứng dụng. *Eclipse* bao gồm tập hợp các dịch vụ dùngđể xây dựng các ứng dụng dựa trên các thành phần cắm thêm (*Plugin)*vàcho phép các thành phần cắm thêm tạo thành ứng dụng riêng biệt cho từng mục đích khác nhau, trong đó phải kể đến bộ công cụ phát triển *Java* (JDT - Java Development Tools).




Hình 1: Mô hình Eclipse

Eclipse sử dụng môi trường phát triển thành phần cắm thêm (PDE-*Plugin Development Environment*)bao gồm một không gian làm việc cơ bản và một hệ thống thành phần cắm thêm mở rộng tương thích với mọi môi trường và mọi hệ điều hành. *Eclipse*còn cho phép xây dựng các ứng dụng, công cụ tích hợp vào môi trường *Eclipse*.Điều đó trở thành mối quan tâm chủ yếu của những nhà phát triển muốn mở rộng Eclipse. Nhà phát triển sẽ giảm chi phí đầu tư phát triển ứng dụng, rút ngắn thời gian triển khai ứng dụng khi áp dụng trên nhiều nền tảng công nghệ khác nhau.

Sự bình đẳng và nhất quán luôn có trong Java. *Eclipse* được viết bằng ngôn ngữ lập trình Java và được sử dụng để phát triển các ứng dụnghỗ trợ cho các ngôn ngữ lập trình (Ada,C, C++,COBOL,Fortran,Haskell,JavaScript, Lasso,Perl, PHP, Python, R, Ruby...) cũng như môi trường phát triển như

*Eclipse Java development tools* (JDT) cho *Java*và*Scala*, *EclipseCDT* cho C/C++ và *Eclipse* PDT cho PHP,*Android Development Tools (ADT) Plugin* cho *Android*.

**Eclipse là mã nguồn mở**

Phần mềm mã nguồn mở là phần mềm được phát hành với giấy phép nhằm bảo đảm cấp cho người dùng các quyền tự do nghiên cứu, chỉnh sửa và phân phối lại phần mềm. Nói chung, luật bản quyền (*copyright*) cho phép tác giả cấm người khác tái tạo, sửa đổihoặc phân phối các bản sao tác phẩm của tác giả đó. Ngược lại, một tác giả, nhờ mô hình cấp phép *copyleft*, sẽ trao cho mọi người bản sao tác phẩm với quyền tái tạo, sửa đổi hoặc phân phối tác phẩm miễn là tất cả những bản sao hoặc bản sửa đổi mới đó cũng phải bị ràng buộc bởi mô hình cấp phép *copyleft*. Vì điều này trên thực tế đã đặt mục đích của bản quyền lên đầu, sử dụng quyền tác giả để cấp quyền cho người dùng, thay vì dành riêng chúng cho các nhà phát triển phần mềm. *Copyleft* thường được miêu tả là "giữ bản quyền".

*OSI-Open Software Initiative* là một tổ chức phi lợi nhuận định nghĩa nguồn mở một cách rõ ràng và cấp các chứng nhận cho các giấy phép đáp ứng được các tiêu chuẩn của nó. *Eclipse* được cấp phép theo Giấy phép công cộng *Eclipse* (*EPL- Eclipse Public License*) phiên bản V1.0 được *OSI* phê duyệt. Điều đótạo điều kiện thuận lợi cho việc sử dụng *Eclipse* phục vụ mục đích thương mại trong khi vẫn bảo đảm bản quyền tác giả theo giấy phép mã nguồn mở.

Các nàh phát triển tạo ra các thành phần cắm thêm cho *Eclipse* hoặc những người sử dụng *Eclipse* làm cơ sở cho phát triển phần mềm ứng dụng được phép phát hành bất kỳ mã *Eclipse* nào mà họ sử dụng hoặc sửa đổi theo EPL.

Bản chất mã nguồn mở làm cho *Eclipse* sẵn dùng, miễn phí. Mã nguồn mở khuyến khích sự đổi mới và mang đến một cơ hội cho các nhà phát triển, ngay cả với các nhà phát triển thương mại.Đồng thời, có thể đóng góp trở lại cơ sở mã nguồn mở chung. Có nhiều nguyên nhân, nhưng có lẽ điều quan trọng nhất là càng nhiều nhà phát triển nhiều đóng góp cho dự án, dự án càng

trở nên có giá trị hơn cho mọi người. Khi dự án càng trở nên có ích hơn, càng nhiều nhà phát triển sẽ sử dụng nó và tạo ra một cộng đồng xung quanh nó, giống như những cộng đồng đã được tạo nên xung quanh Apache và Linux.

### 1.2Lịch sử phát triển

Dự án *Eclipse* (*EclipseProject*) ban đầu được *IBM* thành lập vào tháng 11 năm 2001 và được hiệp hội các nhà cung cấp phần mềm hỗ trợ. Quỹ *Eclipse* đã được thành lập vào tháng 1 năm 2004 như là một tổ chức phi lợi nhuận, hoạt động độc lập. Nó được thành lập nhằm cho phép một cộng đồng minh bạch, mở và trung lập đối với các nhà phát triển phần mềmcó sử dụngnền tảng*Eclipse*. Hiện nay, cộng đồng *Eclipse* hội tụ nhiều cá nhân và tổ chức từ nhiều lĩnh vực trong công nghiệp phần mềm.

Nhiều chương trình của *Eclipse* đã được triển khai bởi *IBM* trước khi dự án *Eclipse* được tạo ra. Người tiền nhiệm của *Eclipse* là VisualAge đã được xây dựng có sử dụng *Smalltalk* trong một môi trường phát triển được gọi là *Envy*. Sau khi Java xuất hiện vào những năm 90, *IBM* đã phát triển mộtmáy ảo mà nó đã làm việc với cả Smalltalk và Java. Sự phát triển nhanh chóng của Java và những ưu điểm của nó cùng với sự mở rộng của mạng lưới Internet đã buộc *IBM* phải xem xét tới việc bỏ máy ảo đôi này và xây dựng một nền tảng mới dựa trên Java từ đầu. Sản phẩm cuối cùng là *Eclipse*đã tiêu tốn của *IBM* khoảng 40 triệu USD vào năm 2001.

Cuối năm 2001, *IBM*hợp tácBorland tạo ra quỹ *Eclipse* phi lợi nhuận, mở ra với thế giới phần mềm nguồn mở. Nhóm các doanh nghiệp này đã dần dần liên kết được với các công ty phát triển phần mềm toàn cầu quan trọng như: Oracle, Rational Software, Red Hat, SuSE, HP, Serena, Ericson và Novell. Trong đó hai công ty lớn cạnh tranh với IBM là Microsoft và Sun Microsystem.Microsoft đã bị loại bỏ vì sự độc quyền của hãng đối với thị trường và Sun Microsystem đã có *IDE* của riêng hãngvới sản phẩm cạnh tranh là NetBeans. Trên thực tế, cái tên *Eclipse* đã được chọn vì mục tiêu là để tạo ra một *IDE* có khả năng cạnh tranh với Visual Studio (của Microsoft) và NetBeans (của Sun Microsystem).

Phiên bản ổn định mới nhất của *Eclipse* là sẵn sàng cho các hệ điều hành Windows, Linux, Solaris, AIX, HP-UX và Mac OS X. Tất cả các phiên bản của *Eclipse* cần phải có một máy chủ ảo Java – JVM (Java Virtual Machine) được cài đặt trong hệ thống, môi trường thực thi Java - JRE (Java Runtime Environment) hoặc bộ công cụ cho lập trình viên Java - JDK (Java Developer Kit) của Sun.

### 1.3 Kiến trúc Eclipse

Kiến trúc của Eclipse được xây dựng dựa trên hai thành phần chính: thành phần lõi (*core*) và các thành phần gắn thêm (*plugin*).

- Thành phần lõibao gồm các chức năng, dịch vụ mà các hệ phát triển ứng dụng phải có như chức năng cung cấp giao diện, trình soạn thảo văn bản, gỡ lỗi… cần cho mọi nền tảng lập trình (cần cho các plugin).
- Thành phần gắn thêm bao gồm nhiều thành phần dễ dàng tích hợp vào nhiều ứng dụng chạy trên nền Eclipse.Các chức năng của thành phần lõi tách biệt với các chức năng của phần giao diện.

Kiến trúc *Eclipse* được thiết kế và xây dựng để đáp ứng các yêu cầu sau:

- *Hỗ trợ việc xây dựng các công cụ khác nhau để phát triển ứng dụng.*
- *Hỗ trợ các công cụ để thao tác trên bất kỳ loại nội dung nào.Ví dụ:HTML, Java, C, C++, JSP, PHP, EJB, XML và GIF.*
- *Hỗ trợ tích hợp các công cụ một cách dễ dàng.*
- *Hỗ trợ môi trường phát triển ứng dụng giao diện và không giao diện.*
- *Chạy được trên nhiều hệ điều hành, gồm Window, Linux, Android,…*

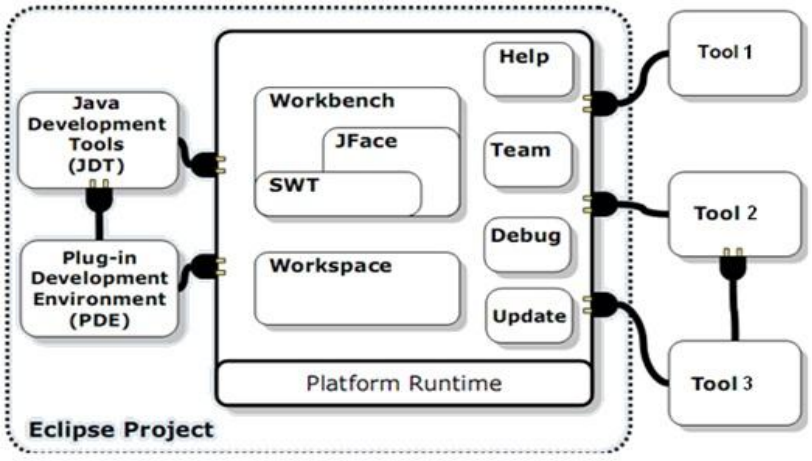
Kiến trúc nền tảng *Eclipse* cung cấp các công cụ cùng cơ chế sử dụng và quy định để có thể tích hợp các công cụ một cách dễ dàng. Những cơ chế này được thể hiện thông qua giao diện *API*, lớp và phương thức. Ngoài ra, nền tảng *Eclipse* còn cung cấp các *framework* để dễ dàng phát triển các công cụ mới.

*EclipseSDK*bao gồm 3 phần chính:

- *Platform*
- *JavaDevelopment Toolkit* (*JDT*)
- *PluginDevelopment Environment* (*PDE*)

Với *JDT*, *Eclipse* là một môi trường hỗ trợ phát triển Java. JDT cũng có thể được coi như là một *Plugin* cho *Eclipse* như là một môi trường lập trình tích hợp (*Java IDE -Integrated Development Enviroment*).PDE hỗ trợ việc mở rộng *Eclipse*, tích hợp các *Plugin* vào *EclipsePlatform*.

*EclipsePlatform* là nền tảng của toàn bộ phần mềm *Eclipse*, cung cấp những dịch vụ cần thiết cho việc tích hợp những bộ công cụ phát triển phần mềm dưới dạng *Plugin*.*EclipsePlatform* được xây dựng dựa trên cơ chếphát hiện, tích hợp và chạy các thành phần gắn thêm. *Plugin* là thành phần gắn thêm có thể hoạt động độc lập hoặc cùng với các thành phần khác. Thông thường, một ứng dụng có thể tổ hợp nhiều thành phần gắn thêm, mỗi thành phần gắn thêm này lại nối với các thành phần gắn thêm khác tùy vào mục đích sử dụng




*Hình2: Kiến trúc tổng quan Eclipse*

**1. Nền tảng thời gian thựcthi (The Platform runtime):**

Công việc chính của *nền tảng thời gian thực thi*là quản lý*Plugin* đang có trong thư mục *Plugin* của *Eclipse*.Mỗi *Plugin* đều có 1 tập tin *Manifest* liệt kê những kết nối mà *Plugin* cần.*Plugin* chỉ được tải vào *Eclipse* khi thực sự cần thiết để giảm lượng tài nguyên yêu cầu và thời gian khởi tạo.

**2. Không gian làm việc (The Workspace) :**

*Không gian làm việc* quản lý tài nguyên người dùng được tổ chức dưới dạng Project. Mỗi dự án là một thư mục con trong thư mục *Không gian làm việc*. Mỗi dự án có chứa các tập tin và thao tác của người dùng. Tất cả các file trong không gian làm việc được trực tiếp tiếp cận với các chương trình chuẩn và công cụ cơ bản của hệ điều hành. Các công cụ tích hợp với nền tảng được cung cấp API để xử lý nguồn tài nguyên trong không gian làm việc (các dự án, các tập tin, và thư mục). Tài nguyên được đại diện bởi các đối tượng thích ứng để các bên khác có thể mở rộng hành vi của họ.

Không gian làm việc bảo quản cấp thấp lịch sử những sự thay đổi tài nguyên, tránh thất thoát tài nguyên người dùng. Tài nguyên được tổ chức dưới dạng cấu trúc cây rất hữu ích và hiệu quả.Từ đó, mỗi nút dữ liệu chỉ ra nơi mà các công cụ cần thêm, loại bỏ, hoặc làm mới.

*Không gian làm việc*thông báo những công cụ cần thiết cho việc thay đổi tài nguyên. Đặc tính này cho phép các nhà xây dựng một số dự án mở rộng được đăng ký trên cùng một dự án và cung cấp nhiều cách để kích hoạt toàn dự án và không gian làm việc với dự án. Một không gian làm việctự độngxây dựng các tính năng, tự động kích hoạt các hoạt động cần thiết trong quá trình xây dựngsau mỗi lần sửa đổi tài nguyên (hoặc hàng loạt hoạt động).

**3. Bàn làm việc (Workbench):**

Bàn làm việc là giao diện đồ họa người dùng của *Eclipse*, gồm có *Standard Widget Toolkit (*SWT) và *JFace*. *Eclipse* không hoàn toàn bắt buộc phải sử dụng *SWT* hay *JFace* để lập trình giao diện, người lập trình vẫn có thể sử dụng*AWT* hay *SWING* của *Java* thông qua việc cài đặt các *Plugin*s.

***SWT (****Standard Widget Toolkit*) là một gói công cụ mã nguồn mở được phát triển bởi *IBM*, cung cấp cho các lập trình viên Java giải pháp để phát triển giao diện đồ họa người dùng. *SWT* cung cấp một hệ điều hành độc lập thông thường *API* cho các vật dụng và đồ họa thực hiện một cách mà cho phép tích hợp chặt chẽ với hệ thống cửa sổ cơ bản. Toàn bộ giao diện người dùng của *EclipsePlatform*, và các thành phần cắm vào nó, đều sử dụng *SWT* để trình bày thông tin cho người sử dụng.

***JFace*** là gói công cụ để xây dựng giao diện người dùng cấp cao, *JFace* là tầng trên cùng của SWT, cung cấp các lớp thuộc mô hình *MVC (*Model-View- Controller) để phát triển các ứng dụng đồ họa dễ dàng hơn.

Cơ chế hoạt động cho phép người sử dụng các lệnh được xác định độc lập từ các *điều khiển* trong giao diện người dùng. Một hành động đại diện cho một lệnh có thể được kích hoạt bởi người dùng thông qua một nút,menu, hoặc các núttrên một thanh công cụ.

**Bàn làm việc**

Không giống như *SWT* và *JFace*, được cả hai mục đích chung giao diện người dùng bộ công cụ, bàn làm việc cung cấp các tính cách giao diện người dùng của nền tảng *Eclipse* và cung cấp các cấu trúc trong đó các công cụ tương tác với người sử dụng. Bởi vì điều này và xác định vai trò trung tâm, bàn làm việclà đồng nghĩa với Nền tảng *Eclipse* giao diện người dùng như một toàn thể và với các cửa sổ chính của người dùng thấy khi các nền tảng đang chạy. Các *APIWorkbench* phụ thuộc vào các *APISWT*ít hơn vào các *APIJFace*. Việc thực hiện *Workbench*được xây dựng bằng cách sử dụng cả hai *SWT* và *JFace*; Java AWT và Swing không được sử dụng

Trong *EclipsePlatform*, giao diện người dùng dựa trên các trình soạn thảo, khung nhìn, và phối cảnh. Từ quan điểm của người dùng, các mẫu và khung nhìn cần được nhìn thấy trên màn hình.

Trình soạn thảocho phép người dùng mở, chỉnh sửa và lưu các đối tượng. Một vòng đời được mở, lưu gần giống như các công cụ hệ thống tập tin, nhưng được tích hợp chặt chẽ hơn vào khung làm việc. Khi hoạt động,

một trình soạn thảocó thể bổ sung các lệnh lên thanh menuvà thanh công cụ. Nền tảng này cung cấp một trình soạn thảo tiêu chuẩn cho các nguồn tài nguyên văn bản, nhiều trình soạn thảocụ thể được cung cấp bởi các *Plugin*s.

Khung nhìn cung cấp thông tin về một số đối tượng mà người dùng đang làm việc tại khung làm việc. Một khung nhìn có thể hỗ trợ một mẫu bằng cách cung cấp thông tin về các tài liệu đang được chỉnh sửa. Một khung nhìn có thể làm tăng thêm các cách nhìn khác nhau bằng cách cung cấp thông tin về các đối tượng đang được chọn.

Một cửa sổ làm việc có thể có một số khung nhìn riêng biệt, chỉ một trong số đó có thể nhìn thấy tại bất kỳ thời điểm nào. Mỗi điểm có khung nhìn riêng và mẫu của nó được sắp xếp để trình bày trên màn hình

**4. Đội hỗ trợ (Team support):**

Nền tảng *Eclipse* cho phép một dự án trong vùng làm việc đặt theo phiên bản và quản lý cấu hình với một nhóm thư mục liên quan. Nền tảng này có các điểm mở rộng và một *API* cung cấp kho lưu trữ cho phép các thành phầnmới được gắn vào.

Các chức năng được cung cấp bởi một sản phẩm cụ thểluôn ảnh hưởng đến công việc của người dùng, ví dụ, bằng cách thêm vào các bước công khai để lấy tập tin từ các kho lưu trữ, trả lại các tập tin được cập nhật vào kho, và để so sánh các phiên bản tập tin khác nhau. Theo đó, các nền tảng *Eclipse* có một cái nhìn tổng quan và cho phép mỗi nhà cung cấp một số thư mục để xác định công việc riêng của mình để người dùng quen thuộc với nhóm sản phẩm lưu trữ có thể nhanh chóng học cách sử dụng nó từ bên trong *Eclipse*.

**5. Trợ giúp (Help):**

Cung cấp hệ thống tài liệu mở rộng, có thể là định dạng HTML hay XML.Những người muốn phát triển *Eclipse* sẽ sử dụng PDE (*PluginDevelopmentEnvironment*)để bổ sung thêm các *Plugin* mới.

## TIỂU KẾT CHƯƠNG 1

Chương 1 đã cung cấp thông tin tổng quát về lịch sử phát triển và ưu nhược điểm của *Eclipse*. Các đặc điểm này được tóm tắt như sau:

***Ưu điểm sau:***

- Khả năng mở rộng, phụ thuộc vào các thành phần gắn thêm
- Các thành phần gắn thêmđược xây dựng thích nghi với việc phát triển trênmọi ứng dụng, từ ứng dụng trong doanh nghiệp, ứng dụng trên máy tính cá nhân cho đến các ứng dụng nhúng cho các thiết bị.
- Những người phát triển có thể tự phát triển các thành phần gắn thêm theo yêu cầu riêng của họ.

***Nhược điểm:***

- Người phát triển cần có hiểu biết về kiến trúc *Eclipse* để biết khi nàocần gắn thêm (hay gỡ ra) hoặc gắn thêm thành phần nào.
- Ngoài ra, các thành phần gắn thêm (các dự án) không ngừng phát triển nên phải biết lúc nào thì nâng cấp lên phiên bản mới hơn.

# CHƯƠNG 2:KIẾN TRÚC CỦA PLUGIN TRONG ECLIPSE

Trong chương này,khóa luận trình bày về kiến trúc *Plugin* nói chung,kiến trúc *Plugin* trong *Eclipse*, phát triển phần mềm dựa trên các thành phần phần mềm.

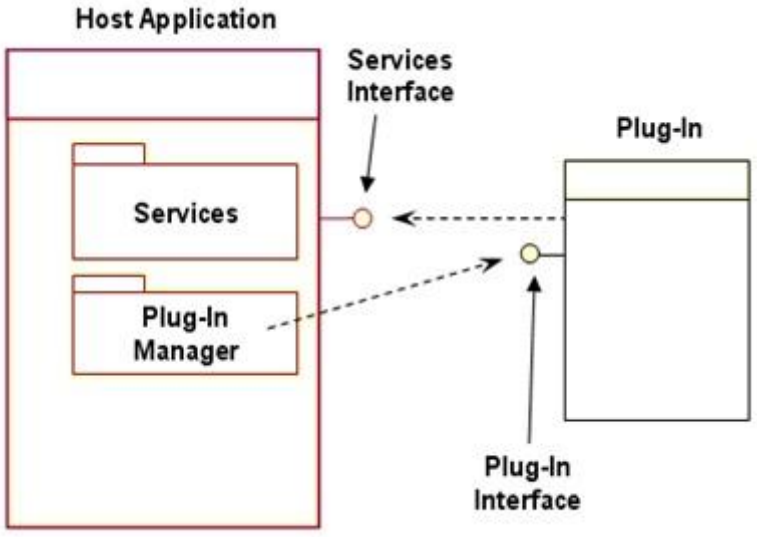
## 2.1 Giới thiệu

*Plugin* còn được gọi là trình cắm, hay phần bổ trợ là một thành phần giúp phần mềm ứng dụng thêm những tính năng cụ thể. Nếu được hỗ trợ, *Plugin* cho phép tùy biến các chức năng của một ứng dụng.

Các phần mềm ứng dụng hỗ trợ*Plugin* vì nhiều lý do. Một số lý do chính bao gồm:

- Cho phép các nhà phát triển thứ ba tạo ra các tính năng để mở rộng phần mềm đó.
- Hỗ trợ một cách dễ dàng trong việc bổ sung thêm các tính năng mới.
- Giảm kích thước của một ứng dụng.
- Tách mã nguồn từ một ứng dụng vì giấy phép phần mềm không tương thích.

**Kiến trúc chung của Plugin**




*Hình3: Kiến trúc Plugin*

Ứngdụng chính cung cấp dịch vụ mà các *Plugin* có thể sử dụng, bao gồm cách để *Plugin* đăng ký với ứng dụng chính và một giao thức cho việc trao đổi dữ liệu. *Plugin* phụ thuộc vào các dịch vụ cung cấp bởi các ứng dụng chính và thường không tự hoạt động. Ngược lại, các ứng dụng chính hoạt động độc lập với *Plugin*.

Giao diện lập trình ứng dụng (*API*) mã nguồn mở cung cấp một giao diện tiêu chuẩn, cho phép các hãng thứ ba để tạo ra các *Plugin* tương tác với các ứng dụng chính. Một *API* ổn định cho phép *Plugin* của hãng thứ ba tiếp tục hoạt động như các thay đổi đối với phiên bản gốc và để mở rộng vòng đời của các ứng dụng đã lỗi thời.

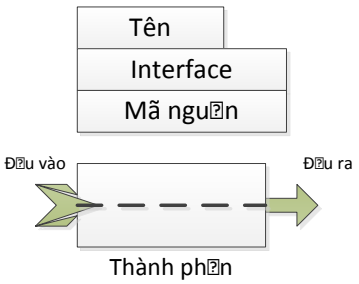
## 2.2 Phát triển phần mềm dựa trên thành phần

Công nghệ phần mềm phát triển qua nhiều giai đoạn và từng giai đoạn luôn có những công nghệ mới xuất hiện giúp phát triển phần mềm thuận lợi và bền vững. Đó là *Component-Based Development* (CBD – Phát triển dựa trên các thành phần).

Ưu điểm của phát triển dựa trên các thành phần là khả năng sử dụng lại và thời gian phát triển ngắn. Tuy nhiên có những trở ngại mà phát triển dựa trên các thành phần phải vượt qua như: Tính tương thích với môi trường, tính kết hợp các thành phần với nhau, tính làm mịn của các thành phần phần mềm.

### 2.2.1 Mô hình trừu tượng

Mô hình trừu tượng được dùng để mô tả các thành phần. Mô hình này đưa ra một vài định nghĩa chung và tổng quát gồm: cú pháp (syntax), ngữ nghĩa (semantics) và tính kết hợp (composition).



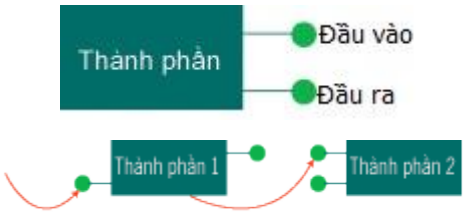

*Hình4: Mô hình trừu tượng của thành phần*

### 2.2.2 Cú pháp

Ngôn ngữ lập trình cấu tạo nên thành phần phần mềm sẽ quyết định cấu trúccủa nó. Trong các thành phần phần mềmhiện có, ngôn ngữ của thành phần là ngôn ngữ lập trình. Ví dụ: EJB tạo bằng ngôn ngữ lập trình Java và là một Java class.

### 2.2.3 Ngữ nghĩa

Thành phần là một đơn vị phần mềm gồm có tên, giao tiếp (interface) và mã nguồn. Mã nguồn cài đặt các dịch vụ của thành phần được giấu kín bên trong thành phần. Interface là phần giao tiếp bên ngoài, cung cấp những thông tin để các thành phần có thể kết nối với nhau bao gồm những dịch vụ đầu vào và các yếu tố cung cấp dịch vụ đầu ra. Nếu có nhiều đầu vào và nhiều đầu ra, thành phần phải xác định quan hệ giữa các đầu vào và đầu ra.




*Hình5:Ví dụ về thành phần với các kiểu đầu vào - ra*

Thành phần là một đơn vị chức năng chưa hoàn chỉnh. Một hệ thống sẽ bao gồm nhiều thành phần nối ghép lại. Đầu vào của thành phần này sẽ ghép với đầu ra của thành phần khác.Tương tự, đầu ra của thành phần cũng có thể là đầu ra của hệ thống hoặc là đầu vào của một thành phần khác (hình 4).

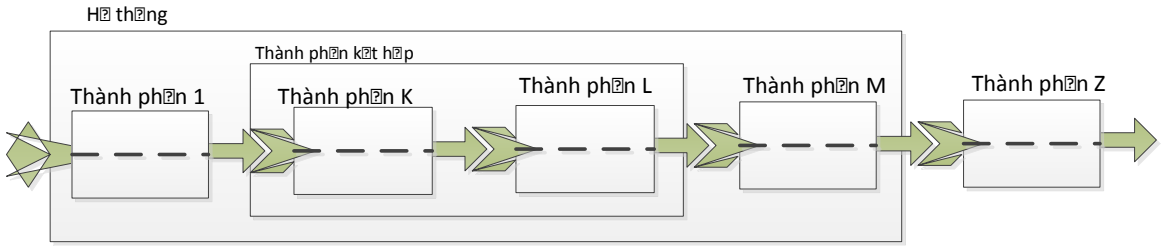
Hiện nay, thành phần có xu hướng chính là các đối tượng trong kiểu lập trình hướng đối tượng. Các dịch vụ đầu ra là các phương thức. Các dịch vụ đầu vào được quản lý và cung cấp bằng các bộ chứa (container) nơi chứa các đối tượng cũng như quản lý sự tiếp cận và tương tác đến chúng.

Ví dụ: JavaBean được quản lý bởi bộ chứa như BeanBox và nó kết nối các JavaBean thông qua sự kiện (event). Trong khi đó, EJB được quản lý bởi chương trình máy chủ J2EE và EJB được tiếp cận thông qua 2 interface là "home" và "remote" hoặc được gọi trực tiếp bởi máy chủ J2EE.

### 2.2.4 Kết hợp (composition)

Trong phát triển phần mềm dựa trên thành phần, sự kết hợp các thành phần với nhau tạo thành hệ thống hoàn chỉnh là vấn đề cốt lõi (các thành phần có thể kết hợp với nhau để tạo thành thành phần phức hợp). Vấn đề nảy sinh là hiện nay chưa có một ngôn ngữ kết hợp đảm bảo tương thích về ngữ nghĩa và cú pháp cho các thành phần EJB, COM, ...

Sự kết hợp thành phần có thể xảy ra ở hai giai đoạn chính trong vòng đời của thành phần là thiết kế (design) và triển khai (deployment).




Hình6: Sự kết hợp các thành phần trong hệ thống

## 2.3Kiến trúc Plugin trong Eclipse

### 2.3.1Giới thiệu về Plugin và extension point

**Plugin**: là một thành phần(component) cung cấp một số chức năng [5]

Ví dụ plugin lớn: HTML editor

Ví dụ plugin nhỏ: Action để tạo file zip

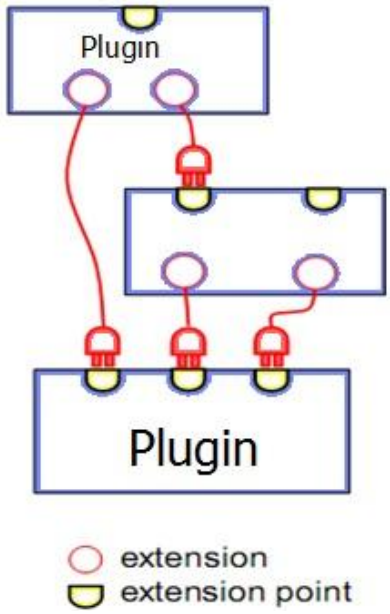
**Extension point**:Điểm mở rộng. [5]

Điểm mở rộng là 1 cơ chế cho phép 1 plugin có thể thêm các chức năng từ 1 plugin khác. [5]

Ví dụ: Điểm mở rộng cho giao diện người dùng workbench. [5]

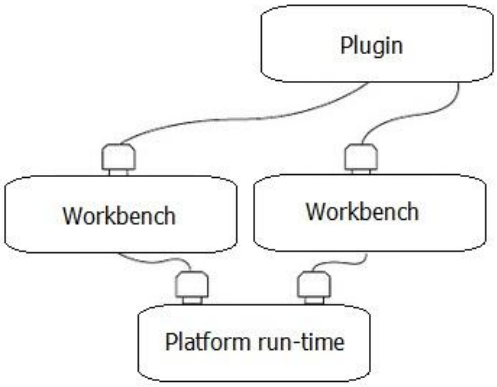
**Extension**: một chức năng

Ví dụ: các chức năng của HTML editor




*Hình7: Plugin và extension*

Mỗi *Plugin* trong *Eclipse* phải sử dụng các nhân *APIEclipse* (core *APIEclipse*) để xác định các lớp mới để chạy như một thành phần của cơ sở *Platform*.




*Hình8: Plugin trong Workbench và Workspace*

Mỗi *Plugin* có một hay nhiều *extension point* và có thể tùy ý khai báo một điểm mở rộng mới. Mỗi *Plugin* phụ thuộc vào một hoặc nhiều *Plugin*

khác. Trong *Eclipse*, *Plugin* phải sử dụng một trong số *Plugin* hệ thống (*Plugin* system)nhưnguồn lưu trữ (projects, folders và các files), thành phần giao diện người dùng để thực hiện công việc (*Workbench user interface components*), các *Plugin* có sẵn, trình gỡ lỗi (*debugging*), công cụ phát triển Java (*JDT*), môi trường phát triển thành phần cắm thêm (*PDE*) [4].

*PDE* cho phép các nhà phát triển dễ cài đặt *Plugin* trong *Eclipse* hơn là thông qua một công cụ bên ngoài. PDE chính là một *Plugin* và dựa vào sự hỗ trợ cùng một cốt lõi để thực hiện nhiệm vụ của mình như mỗi *Plugin* khác

*Eclipse* extensions được chia làm 3 loại:

- Plugin
- Fragment
- Feature

Pluginlà một công cụbổ sung thêm chức năng để *Eclipse* sử dụng các điểm mở rộng. Chu trình sống của *Plugin* được điều khiển bởi *Workbench*. *Plugin* được thêm vào *Eclipse* khi *Eclipse* khởi động mà không được *load* ngay lập tức, *Plugin* chỉ được *load* khi cần thiết. Trong suốt quá trình *load*, *Elipse* sẽ gọi phương thức *startup()* và trước khi *unload* thì gọi *shutdown()*. Hai phương thức *load* và *unload* có thể được coi như là hàm khởi tạo *init()* và hàm hủy *destroy()*.

*Fragment* là tập hợp các bộ mã tạo nên một *Plugin*. Khi một *fragment* được thực thi, *Eclipse* sẽ kết hợp các đoạn mã *fragment* với *Plugin* tạo nên sự rõ ràng trong hành vihay phương thức của *Plugin.*

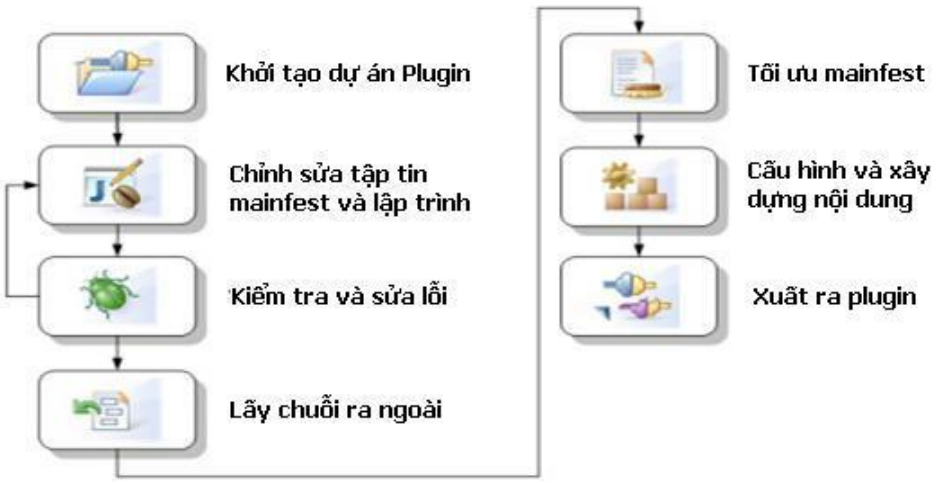
*Feature* là một tập hợp các *Plugin* mà đại diện cho tập các chức năng. Feature được cài đặt bằng tay hay sử dụng quản lý cập nhật (Update Manage). Các *Plugin* và các tập tin hỗ trợ (supporting files) được đóng gói vào một tập tin lưu trữ tính năng (*feature archive*) bao gồm một tính năng tập tin cấu hình, *Feature.xml*, mô tả nội dung của kho lưu trữ và các liên kết *Plugin*. Khi một *Plugin* được đóng gói thành một công cụ hoàn chỉnh, *fragment*gồm mã cập nhật cho một *Plugin* chỉ là một đóng gói của một tập hợp các *Plugin*để dễ dàng cài đặt vào *Eclipse* [4].

### 2.3.2 Điểm mở rộng Plugin (Plugin Extension Points)

Một số lượng lớn các điểm mở rộng*Plugin*được *Eclipse* xác định có sẵn. Các điểm mở rộng*Plugin*được chia thành các lĩnh vực sau:

- Platform runtime: Sử dụng một trong hai điểm mở rộng (extension points), có thể xác định toàn bộ hành vi của Eclipse.
- Không gian làm việc: Điểm mở rộngchịu trách nhiệm quản lý tài nguyên người dùng được tổ chức dưới dạng *Project.*
- *B*àn làm việc:giao diện đồ họa người dùng của *Eclipse*có số lượng lớn nhất các điểm mở rộng. Một mẫu cho các điểm mở rộngbao gồm sự hỗ trợ cho các khung nhìn (view) và trình biên tập, các tổ hợp phím, hỗ trợ kéo thả giữa các khung nhìn và bổ sung bảng điều khiển (panels) cho hộp thoại *Preferences*.
- *Gỡ lỗi*: Các điểm mở rộngđể gỡ lỗi giữa các hành vi và sử dụng điểm giao diện (interface points)
- *Trợ giúp*: Các điểm mở rộngđược tích hợp vào công cụ tìm kiếm trợ giúp của *Eclipse*. Các trợ giúp được hiển thị dưới dạng bảng hoặc sử dụng thêm những *Plugin* mang tính đặc thù để trợ giúp.

### 2.3.3 Tiến trình làm việc của Plugin




*Hình9: Tiến trình làm việc của Plugin*

Về mặt kỹ thuật, một *Plugin* là một Java Archive (JAR) độc lập và tự khởi tạo. *Plugin* độc lập vì *Plugin* chứa mã và tài nguyên các *Plugin*khác cần để chạy. *Plugin* tự khởi tạo bởi vì nó bao gồm các thông tin của *Plugin*, tài nguyên *Plugin*, công dụng của nó. Trong một *Plugin*, hai tập tin mô tả là MANIFEST.MF và *Plugin*.xml.

### 2.3.4Tập tin cấu hình (manifest) của Plugin

Thông tin mô tả chi tiết *Plugin* nằm trong tập tin manifest (*Plugin*.xml). *Eclipse* sử dụng tập tin *cấu hình* để tích hợp *Plugin* vào framework.

Tập tin *manifest* chứa những thông tin chung về *Plugin* bao gồm: tên *Plugin*, phiên bản, tên lớp, tên file JAR. Ngoài ra, tập tin cấu hình còn chứa danh sách những *Plugin* mà *Plugin* hiện tại sử dụng.

Ví dụ về tập tin manifest

```
//Thông tin về Plugin
Manifest-Version: 1.0
Bundle-ManifestVersion: 2
Bundle-Name: Tkb
Bundle-SymbolicName: Tkb;singleton:=true
Bundle-Version: 1.0.0.qualifier
Bundle-Activator: tkb.Activator

/* Các Plugin cần dùng */
Require-Bundle: org.Eclipse.ui,
 org.Eclipse.core.runtime,
Pluginsql
//Môi trường chạy Plugin
Bundle-RequiredExecutionEnvironment: JavaSE-1.7
Bundle-ActivationPolicy: lazy

<?xml version="1.0" encoding="UTF-8"?>
/* Thuộc tính định dạng Plugin */
<?Eclipse version="3.4"?>
```

```
/* Thông tin Plugin */
<Plugin
 id="com.example.TKB"
 name="TKB Plug-in"
 version="1.0.0"
 provIDEr-name="EXAMPLE"
 class="com.example.Tkb.TKBPlugin">

//Khai báo chức năng mà Plugin cung cấp
<extension
 point="org.Eclipse.ui.actionSets">
<actionSet
label="Sample Action Set"
visible="true"
id="Tkb.actionSet">
<menu
 label="&amp;Thờikhóabiểu"
 id="sampleMenu">
<separator
name="sampleGroup">
</separator>
</menu>
<action
 class="view.Help"
 icon="icons/sample.gif"
 id="view.Help"
 label="&amp;Trợgiúp"
 menubarPath="sampleMenu/sampleGroup"
 toolbarPath="sampleGroup"
tooltip="Trợgiúp">
</action>
</actionSet>
</extension>
```

```
</Plugin>
```

Hai *Plugin* quan trọng trong *Eclipse* là *PluginWorkspace* (org.*Eclipse*.core.resources) và *PluginWorkbench* (org.elipse.ui).

Một trong những extension quan trọng là extension có thuộc tính point=org.Eclipse.ui.actionSets. Một *actionSet* là một nhóm các chức năng mà *Plugin* sẽ thêm vào giao diện *Workbench*, ví dụ như là menus, menu items, và toolbars.*ActionSet* sẽ gom nhóm các chức năng để người dùng có thể quản lý chúng một cách dễ dàng. Một *actionSet* có thể chứa 2 thẻ: thẻ menu mô tả nơi *item* xuất hiện và cách hiển thị của nó trên *Workbench*; thẻ action mô tả một *item* sẽ thực hiện chức năng gì và lớp xử lý action để thực hiện chức năng đó.

### 2.3.5 Plugin fragment và feature

***Plugin fragment**:* được hình thành từ một phần của *Plugin*. *Plugin fragment* có ích khi muốn sử dụng một phần của *Plugin* độc lập với phần còn lại của *Plugin*.

*Plugin fragment* được dùng để thêm các đặc điểm nào đó vào một *Plugin* đang có mà không cần *build* lại *Plugin* đó hoặc được dùng để cung cấp chức năng cho một nền tảng nào đó.

*Plugin fragment* gần giống với *Plugin* thông thường, chỉ khác ở một vài đặc điểm sau:

- Thông tin mô tả *Plugin* được lưu trong tập tin fragment.xml thay vì *Plugin*.xml.
- Trong tập tin fragment.xml, nút gốc là <fragment>và nút này có 2 thuộc tính là *Plugin-id* và *Plugin-version* dùng để chỉ ra định danh và phiên bản của *Plugin* cha.
- *Plugin fragment* sẽ tự động kế thừa các nút <requires> của *Plugin* cha và có thể thêm các nút <*requires*> khác nếu cần thiết.

***Plugin feature**:* Trong kiến trúc *Eclipse*, *feature* là việc đóng gói một nhóm các *Plugin* có liên quan lại thành một sản phẩm tích hợp. *Plugin feature*

không có chứa code. Ví dụ: *Java Development Tooling* (JDT) là một feature project được tạo thành bởi các *Plugin* như: *Java editor, debugger*, và *console*.

Tập tin đặc tả *feature project* là *feature.xml*, tập tin này chứa tham chiếu đến các *Plugin* và các tài nguyên khác của *feature project*, đồng thời chứa các thông tin về việc update, copyright và license.

### 2.3.6 Đóng gói Plugin

*Eclipse* quyết định *Plugin* nào được *load* bằng cách kiểm tra thư mục *Plugin*s vào thời điểm khởi động. Để cài đặt một *Plugin* thì cần tạo một thư mục con trong thư mục *Plugin*s, sau đó chép tất cả các tập tin chương trình và tập tin manifest vào. Nên đặt tên thư mục chứa *Plugin* theo chuẩn: tên của *Plugin* được theo sau bởi dấu gạch dưới và chỉ số phiên bản, ví dụ: C:*Eclipse**Plugin*s\com.example.hello. Sau đó, nén thư mục con vừa được tạo thành tập tin JAR.

Tạo tập tin JAR bằng giao diện *Eclipse*: menu File → Export. Sau đó khởi động lại *Eclipse* để *Eclipse* có thể nhận ra *Plugin* mới.

### 2.3.7Perspective,views, editor

Vào một thời điểm bất kỳ, *Workbench* là một cửa sổ ứng dụng đơn chứa một số lượng các kiểu ô cửa sổ (pane) khác nhau hay còn được gọi là các khung nhìn (views), và một view đặc biệt là trình soạn thảo (editor). Trong một số tình huống, một ô cửa sổ đơn có thể chứa một nhóm các ô cửa sổ khác dưới dạng các thẻ (tabs). Tùy vào từng perspective, một ô cửa sổ có thể chứa cửa sổ thực thi lệnh trong khi ô cửa sổ khác chứa một bản phác thảo của dự án đang làm việc. Tuy nhiên, trình soạn thảo là một thành phần căn bản không thể thiếu của tất cả các phối cảnh.

Vì có nhiều loại tài liệu khác nhau, nên có nhiều loại trình soạn thảo khác nhau. Khi một tài liệu được mở ra hoặc được tạo mới, *Eclipse* tìm đến trình soạn thảo thích hợp nhất để người dùng thao tác trên tài liệu này. Nếu một tài liệu là một văn bản thông thường (text document), trình soạn thảo văn bản cài đặt sẵn trong *Eclipse* được lựa chọn. Nếu tài liệu là một tập tin java nguồn, trình soạn thảo Java của JDT được lựa chọn, trình soạn thảo này có

nhiều tính năng đặc biệt như kiểm tra cú pháp của các mã lệnh được soạn, hoàn thành mã lệnh tự động (code completion)… Nếu tài liệu là một văn bản Microsoft Word trên Windows và chương trình Word đã được cài đặt trên máy đang sử dụng, tài liệu được mở bởi Word trong *Eclipse* bằng kỹ thuật kết nối và nhúng đối tượng (*Object Linking and Embedding – OLE*).

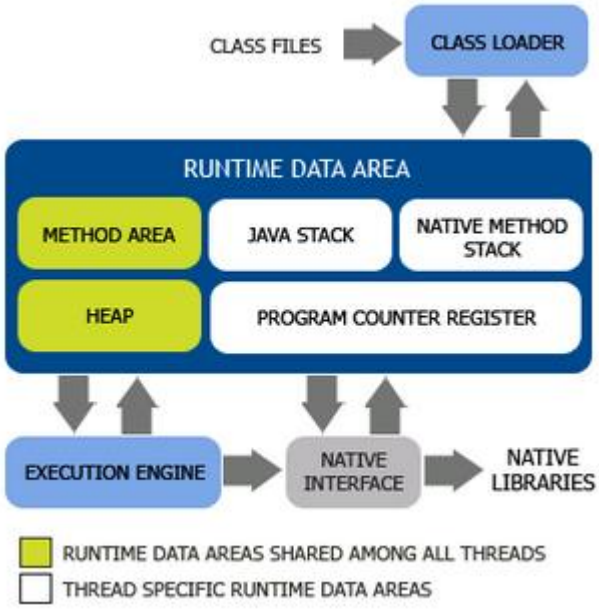
## 2.4Ngôn ngữ lập trình java

### 2.4.1 Lịch sử phát triển của Java

Java được khởi đầu bởi James Gosling và bạn đồng nghiệp ở SunMicrosystemsnăm 1991. Ban đầu ngôn ngữ này được gọi là ***Oak*.** Sauđókhônglâungônngữ mới vớitên gọilà Java ra đời và được giới thiệu năm 1995.

### 2.4.2 Máy ảo Java (Java Virtual Machine)

Tất cả các chương trình muốn thực thi được thì phải được biên dịch ra mã máy. Mã máy của từng kiến trúc CPU của mỗi máy tính là khác nhau (tập lệnh mã máy của CPU Intel, CPU Solarix, CPU Macintosh … là khác nhau), vì vậy trước đây một chương trình sau khi được biên dịch xong chỉ có thể chạy được trên một kiến trúc CPU cụ thể nào đó. CPU Intel có thể chạy các hệ điều hành như Microsoft Windows, Unix, Linux, OS/2, … Chương trình thực thi được trên Windows được biên dịch dưới dạng file có phần mở rộng *.EXE còn trên Linux thì được biên dịch dưới dạng file có đuôi *.ELF, vì vậy trước đây một chương trình chạy được trên Windows muốn chạy được trên hệ điều hành khác như Linux thì phải chỉnh sửa và biên dịch lại. Ngôn ngữ lập trình Java ra đời, nhờ vào máy ảo Java mà khó khăn nêu trên đã được khắc phục. Một chương trình viết bằng ngôn ngữ lập trình Java sẽ được biên dịch ra mã của máy ảo java (mã java bytecode). Sau đó máy ảo Java chịu trách nhiệm chuyển mã java bytecode thành mã máy tương ứng. Sun Microsystem chịu trách nhiệm phát triển các máy ảo Java chạy trên các hệ điều hành trên các kiến trúc CPU khác nhau




Hình 10: Kiến trúc Java Virtual Machine

JVM bao gồm các thành phần chính:

- Class Loader: là một hệ thống con của JVM, làm nhiệm vụ tải các lớp được xác định.
- Class Area: lưu trữ cấu trúc của các lớp, thuộc tính, phương thức của lớp, và code của các phương thức.
- Heap &Java Stack:
  - Heap: là vùng nhớ lưu trử các đối tượng được khởi tạo trong quá trình thực thi.
  - Stack: chứa các frame. Mỗi frame chứa các biến cục bộ và thực thi một hàm gọi và trả kết quả về. Mỗi tiến trình có một Stack riêng, được khởi tạo cùng lúc với tiến trình. Mỗi frame sẽ được tạo khi một hàm được gọi và hủy khi việc gọi hàm kết thúc
- Programming Counter Register: chứa địa chỉ của máy chủ ảo đang thực thi.
- Native Method Stack: chứa các hàm của hệ thống được sử dụng trong chương trình
- Execution Engine: là một hệ thống bao gồm: bộ xử lý ảo, trình thông dịch (đọc Java byte code và thực thi các chỉ thị), JIT compiler biên dịch mã byte code sang mã máy.

### 2.4.3 Một số đặc điểm ngôn ngữ lập trình Java

Java là một ngôn ngữ lập trình hướng đối tượng vừa biên dịch vừa thông dịch. Chương trình nguồn viết bằng ngôn ngữ lập trình Java có đuôi *.java đầu tiên được biên dịch thành tập tin có đuôi *.class và sau đó sẽ được trình thông dịch thông dịch thànhmã máy.

Một chương trình viết bằng ngôn ngữ lậptrình Java sẽ được biên dịch ra mã của máy ảojava (mã java bytecode). Sau đó máy ảoJava chịu trách nhiệm chuyển mã java bytecode thành mã máy tương ứng. Sun Microsystempháttriểncác máyảoJava chạy trên các hệ điều hành trên các kiến trúc CPU khác nhau.

Một chương trình, ứng dụng viết bằng ngôn ngữ Java có thể chạy trên nhiều máy tính có hệ điều hành khác nhau (Windows, MacOS, Linux, …) có cài đặt máy ảojava (Java Virtual Machine). Vì thế chương trình chỉ cần viết một lầnchạy mọi nơi (write once run any where).

Java hỗ trợ lập trình đa nhiệm, đa luồng cho phép nhiều tiến trìnhcóthểchạysongsongcùng mộtthờiđiểmvà tương tác với nhau.

Java hỗ trợ mạnh cho việc phát triển ứng dụng. Công nghệ Java phát triển mạnh mẽ do *Sun Microsystem* cung cấp nhiều công cụ, thư viện lập trình phong phú hỗ trợ cho việc phát triển nhiều loại hình ứng dụng khác nhau, cụ thể như: J2SE (Java 2 Standard Edition) hỗ trợ phát triển những ứng dụng đơn, ứng dụng client-server; J2EE (Java 2 Enterprise Edition) hỗ trợ phát triển các ứng dụng thương mại, J2ME Java 2 Micro Edition) hỗ trợ phát triển các ứng dụng trên các thiết bị di động, không dây, …

*Eclipse*là chương trình chạy trên môi trường Java. Plugin là thành phần cắm thêm trong *Eclipse*. Vì thế chương trình viết dưới dạng Plugin chỉ cần xây dựng 1 lần mà có thể chạy trên nhiều nền tảng khác nhau.

Các *Plugin* trong *Eclipse* có thể tương tác với nhau, hỗ trợ cho nhau và có thể chạy songsong cùng một thời điểm do Java hỗ trợ đa nhiệm, đa luồng.

# CHƯƠNG 3: CHƯƠNG TRÌNH THỬ NGHIỆM

Trong chương này, khóa luận trình bày vềứng dụng sắp xếp thời khóa biểu được xây dựng bằng thành phần cắm thêm (*Plugin*) trên *Eclipse*. *Plugin* này được cài đặt bằng ngôn ngữ lập trình Java. Một số bước trong phân tích thiết kế phân hệ sắp thời khóa biểu được giản lược nhưng không mất tính tổng quát của mục tiêu đề tài là xây dựng thử nghiệm *Plugin* dựa trên nền *Eclipse*.

## 3.1 Mô tả yêu cầu bài toán

Chương trình thử nghiệm hỗ trợ sắp xếp thời khóa biểu với đầu vào là các thông tin như: lớp học (mã môn học, số tín chỉ, buổi học, số lượng sinh viên tối đa), phòng học (địa chỉ, số lượng sinh viên, trang thiết bị giảng dạy đi kèm)... Việc sắp xếp thời khóa biểu cần tuân theo các ràng buộc:

- Về thời gian:
  - Có ít nhất 1 buổi học của 1 môn phải có 2 tiết trở lên
  - Các môn học trong 1 khóa chỉ xếp vào buổi sáng hoặc buổi chiều trong tuần.
- Về phòng học:
  - Số lượng sinh viên trong lớp học phải phù hợp với số lượng sinh viên phòng học có thể chứa.
  - Khi sắp xếp, ưu tiên xếp phòng học có trang thiết bị phù hợp cho các môn có yêu cầu đặc biệt về trang thiết bị.
  - Tại một thời điểm, chỉ được sắp xếp 1 lớp học tại 1 phòng học

**Yêu cầu:**

- Sử dụng mô hình MVC (Model-View-Controller).
- Hệ thống thuần túy là hướng đối tượng, không chấp nhận các cách tiếp cận khác.
- Thiết kế CSDL hợp lý đảm bảo các ràng buộc.
- Chương trình thử nghiệm làthành phần cắm thêm trong *Eclipse*
- Bài toán nhấn mạnh vào việc truy xuất dữ liệu và thuật toán xếp lịch và xây dựng các chức năng:
  - Cập nhật, thêm mới, xóa, lưu … các thông tin nói trên.

- Thống kê, báo cáo, in thời khóa biểu dưới dạng excel.
- Đối với người quản lý: theo dõi lịch học của các lớp trong khoa, trong trường, xếp thời khóa biểu.

## 3.2Xác định mô hình nghiệp vụ

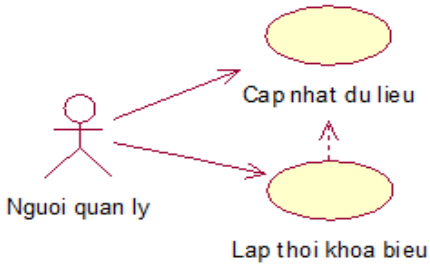
### 3.2.1 Các chức năng nghiệp vụ

Ta có thể xác định các chức năng nghiệp vụ của hệ thống như sau:

| Tham chiếu | **Chức năng** |
|---|---|
| **$R_1$** | **Cập nhật dữ liệu** |
| $R_{11}$ | Cập nhật thông tin phòng học |
| $R_{12}$ | Cập nhật thông tin thiết bị |
| $R_{13}$ | Cập nhật thông tin niên khóa |
| $R_{14}$ | Cập nhật thông tin học phần |
| $R_{15}$ | Cập nhật thông tin khoa |
| **$R_2$** | **Lập thời khóa biểu** |
| $R_{21}$ | Sắp thời khóa biểu |
| $R_{22}$ | In thời khóa biểu |

Tác nhân là một bộ phận bên ngoài hệ thống nhưng có tương tác với hệ thống. Nó chính là đối tượng mà hệ thống phục vụ hoặc cần cung cấp dữ liệu. Hệ thống gồm có tác nhân là người quản lý.

### 3.2.2Biểu đồ use case tổng quan




*Hình11: Biểu đồ Use case tổng quan*

Chương trình thử nghiệm sắp thời khóa biểu xác định được các ca sử dụng và tác nhân như sau:

| Gói ca sử dụng | Các ca sử dụng chi tiết | Tác nhân |
|---|---|---|
| 1. Cập nhật dữ liệu | UC1Cập nhật thông tin phòng học<br>UC2Cập nhật thông tin thiết bị<br>UC3Cập nhật thông tin niên khóa<br>UC4Cập nhật thông tin học phần<br>UC5Cập nhật thông tin khoa | Người quản lý |
| 2. Lập thời khóa biểu | UC6Sắp thời khóa biểu<br>UC7 In thời khóa biểu | Người quản lý |

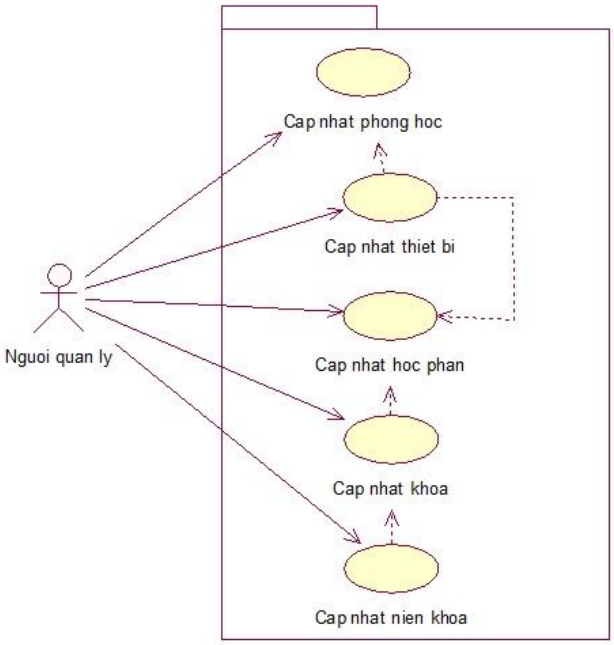
### 3.2.3 Mô tả khái quát các quan hệ con

Hệ thống gồm hai hệ con:

- Cập nhật dữ liệu:có tác nhân duy nhất là người quản lý. Có tác dụng cập nhật thông tin đầu vào như: phòng học, thiết bị, niên khóa, học phần, khoa.
- Lập thời khóa biểu: có tác nhân duy nhất là người quản lý, có tác dụng lập và in thời khóa biểu.

### 3.2.4Các mô hình ca sử dụng chi tiết

#### *a. Gói ca sử dụng "Cập nhật dữ liệu"*




Hình12: Biểu đồ ca sử dụng gói "Cập nhật dữ liệu"

**Mô tả chi tiết ca sử dụng**

- **Ca sử dụng "Cập nhật phòng học"**

| | |
|---|---|
| *Tên ca sử dụng* | Cập nhật phòng học |
| *Tác nhân* | Người quản lý |
| *Mục đích* | Cập nhật thông tin phòng học |
| *Mô tả khái quát* | Người quản lý cập nhật thông tin phòng học |
| *Các tham chiếu* | R11 |

- **Ca sử dụng "Cập nhậtthiết bị"**

| | |
|---|---|
| *Tên ca sử dụng* | Cập nhật thiết bị |
| *Tác nhân* | Người quản lý |
| *Mục đích* | Cập nhật thông tin thiết bị |
| *Mô tả khái quát* | Người quản lý cập nhật thông tin thiết bị trong phòng học |
| *Các tham chiếu* | R12 |

- **Ca sử dụng "Cập nhật niên khóa "**

| | |
|---|---|
| *Tên ca sử dụng* | Cập nhật niên khóa |
| *Tác nhân* | Người quản lý |
| *Mục đích* | Cập nhật thông tin niên khóa |
| *Mô tả khái quát* | Người quản lý cập nhật thông tin niên khóa |
| *Các tham chiếu* | R13 |

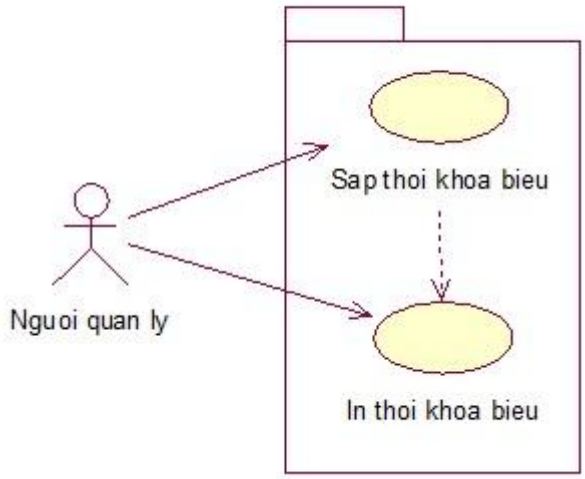
- **Ca sử dụng "Cập nhật học phần"**

| | |
|---|---|
| *Tên ca sử dụng* | Cập nhật học phần |
| *Tác nhân* | Người quản lý |
| *Mục đích* | Cập nhật thông học phần |
| *Mô tả khái quát* | Người quản lý cập nhật thông tin học phần trong kỳ |
| *Các tham chiếu* | R14 |

- **Ca sử dụng "Cập nhật khoa"**

| | |
|---|---|
| *Tên ca sử dụng* | Cập nhật khoa |
| *Tác nhân* | Người quản lý |
| *Mục đích* | Cập nhật thông tin khoa |
| *Mô tả khái quát* | Người quản lý cập nhật thông tin khoa |
| *Các tham chiếu* | R15 |

### *b. Gói ca sử dụng "Lập thời khóa biểu"*




*Hình13:* Biểu đồ ca sử dụng gói "Lập thời khóa biểu"

### Mô tả chi tiết ca sử dụng

- **Ca sử dụng "Sắp thời khóa biểu"**

| | |
|---|---|
| *Tên ca sử dụng* | Sắp thời khóa biểu |
| *Tác nhân* | Người quản lý |
| *Mục đích* | Sắp thời khóa biểu |

| | |
|---|---|
| *Mô tả khái quát* | Người quản lý sắp thời khóa biểu |
| *Các tham chiếu* | R21 |

- **Ca sử dụng "In thời khóa biểu"**

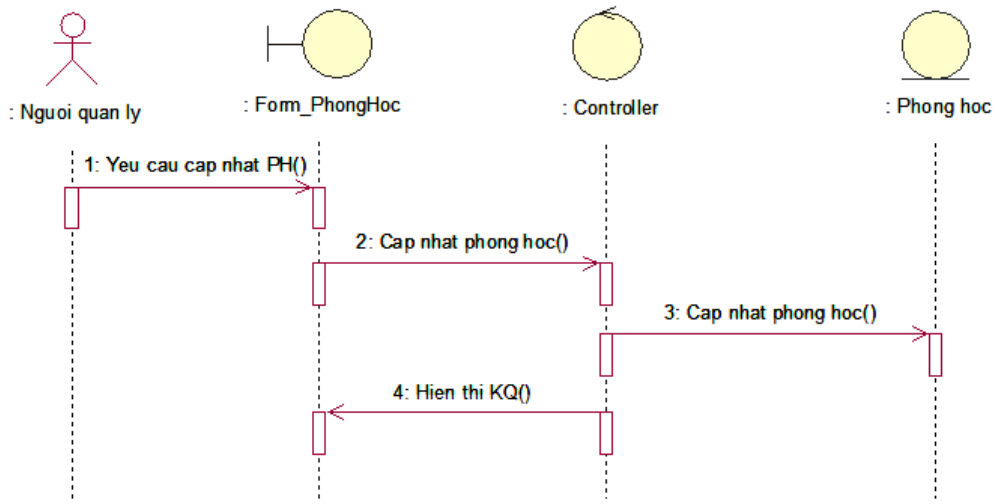
| | |
|---|---|
| *Tên ca sử dụng* | In thời khóa biểu |
| *Tác nhân* | Người quản lý |
| *Mục đích* | In thời khóa biểu |
| *Mô tả khái quát* | Người quản lý muốn in thời khóa biểu dưới dạng excel |
| *Các tham chiếu* | R22 |

## 3.3 Phân tích hệ thống

### 3.3.1 Phân tích gói ca sử dụng "Cập nhật dữ liệu"

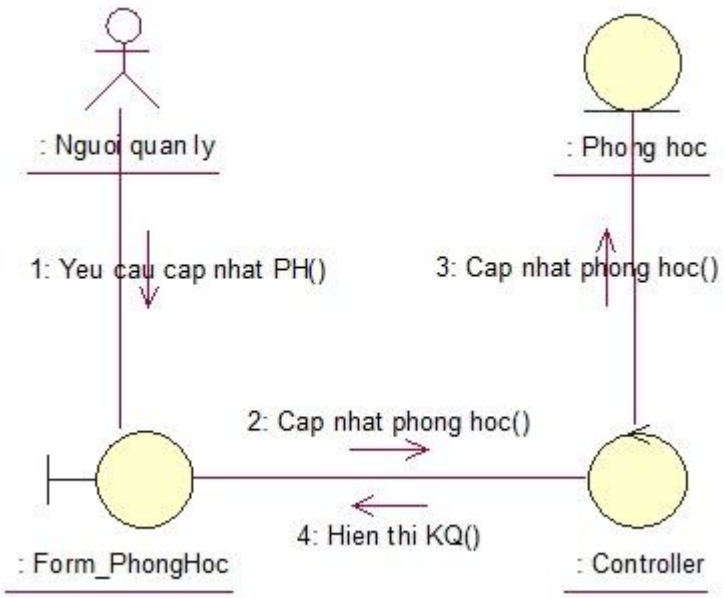
#### *a.Ca sử dụng "Cập nhật phòng học"*

- Biểu đồ tuần tự thực thi ca sử dụng




*Hình 14: Biểu đồ tuần tự thực thi ca sử dụng "Cập nhật phòng học"*

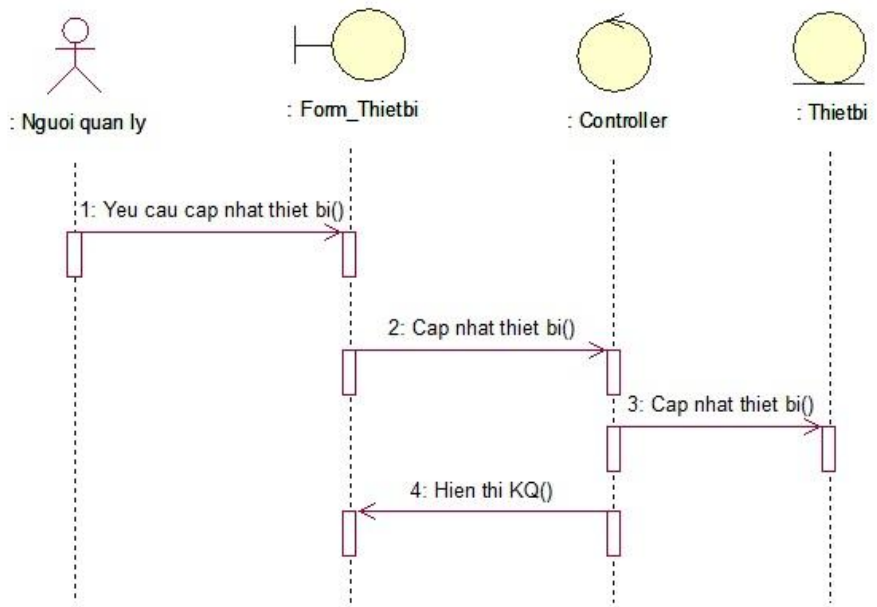
- Biểu đồ cộng tác thực thi ca sử dụng




Hình 15: Biểu đồ cộng tác thực thi ca sử dụng "Cập nhật phòng học"

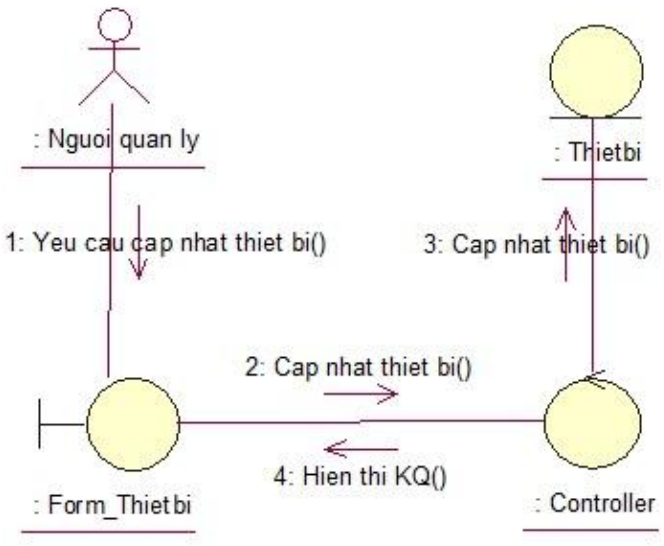
***b.Ca sử dụng "Cập nhật thiết bị"***

- Biểu đồ tuần tự thực thi ca sử dụng




Hình16: Biểu đồ tuần tự thực thi ca sử dụng "Cập nhật thiết bị"

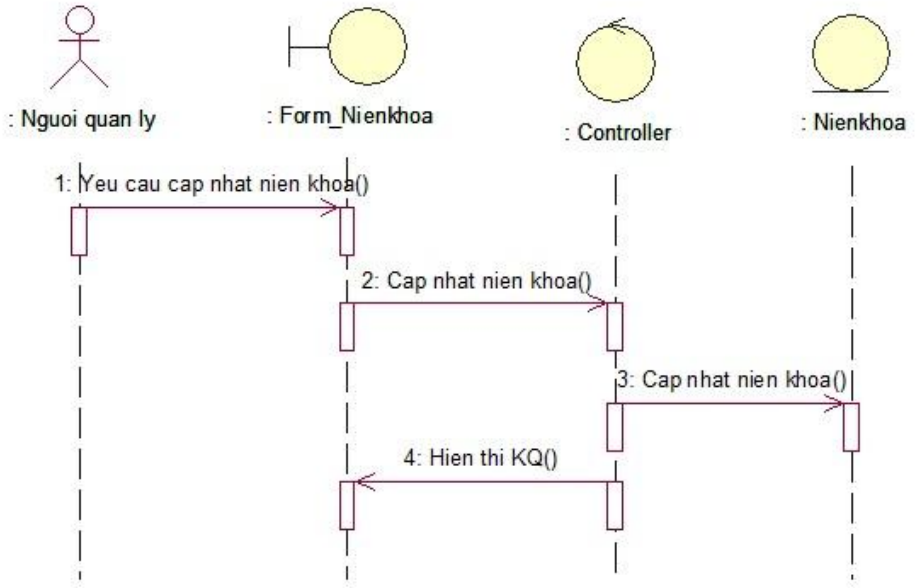
- Biểu đồ cộng tác thực thi ca sử dụng




Hình17: Biểu đồ cộng tác thực thi ca sử dụng "Cập nhật thiết bị"

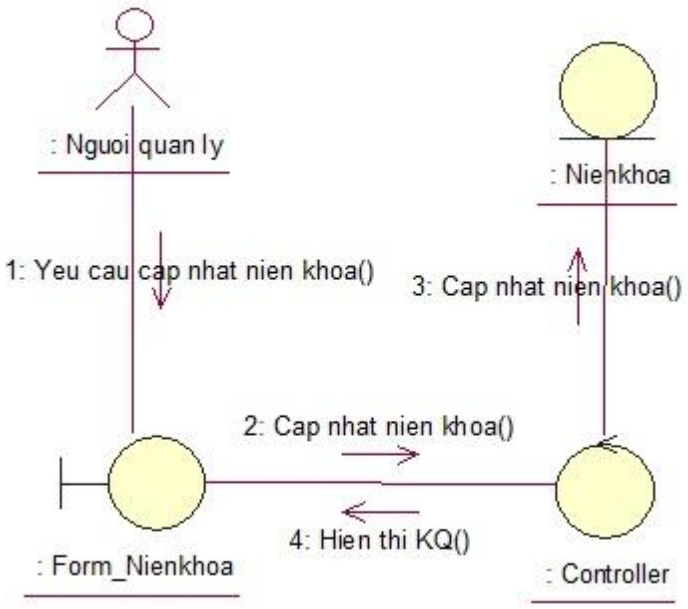
***c.Ca sử dụng "Cập nhật niên khóa"***

- Biểu đồ tuần tự thực thi ca sử dụng




*Hình 18:* Biểu đồ tuần tự thực thi ca sử dụng "Cập nhật niên khóa"

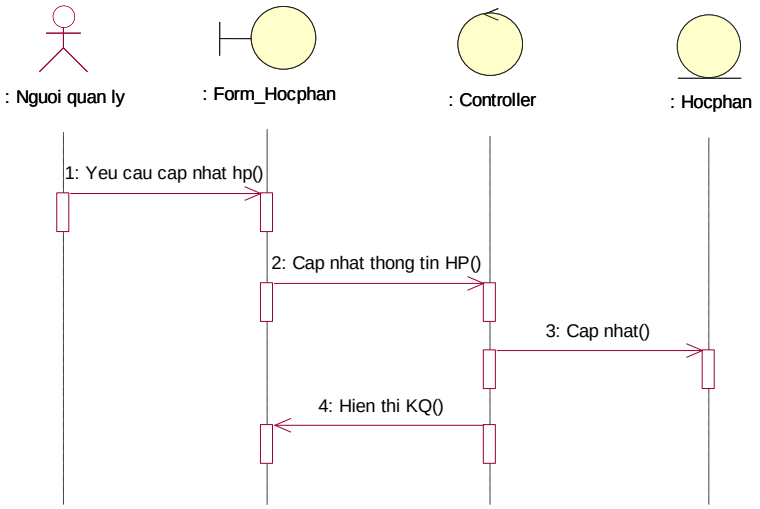
- Biểu đồ cộng tác thực thi ca sử dụng




*Hình19:* Biểu đồ cộng tác thực thi ca sử dụng "Cập nhật niên khóa"

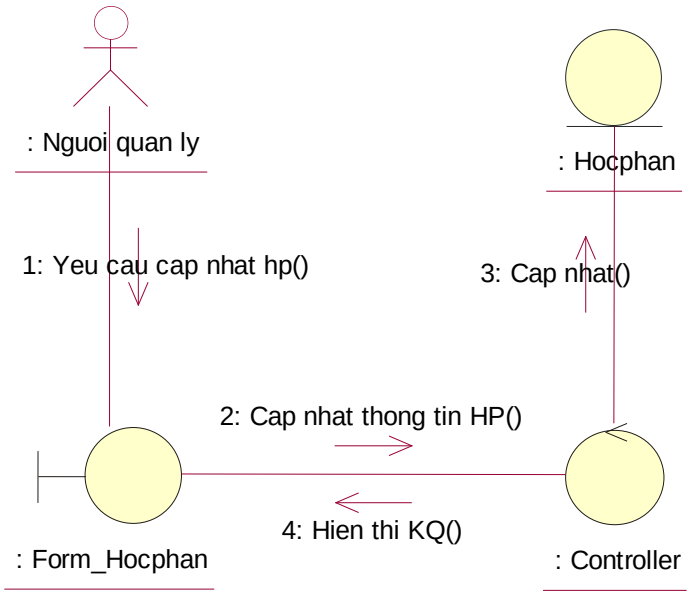
***d.Ca sử dụng "Cập nhật học phần"***

- Biểu đồ tuần tự thực thi ca sử dụng




*Hình20:* Biểu đồ tuần tự thực thi ca sử dụng "Cập nhật học phần"

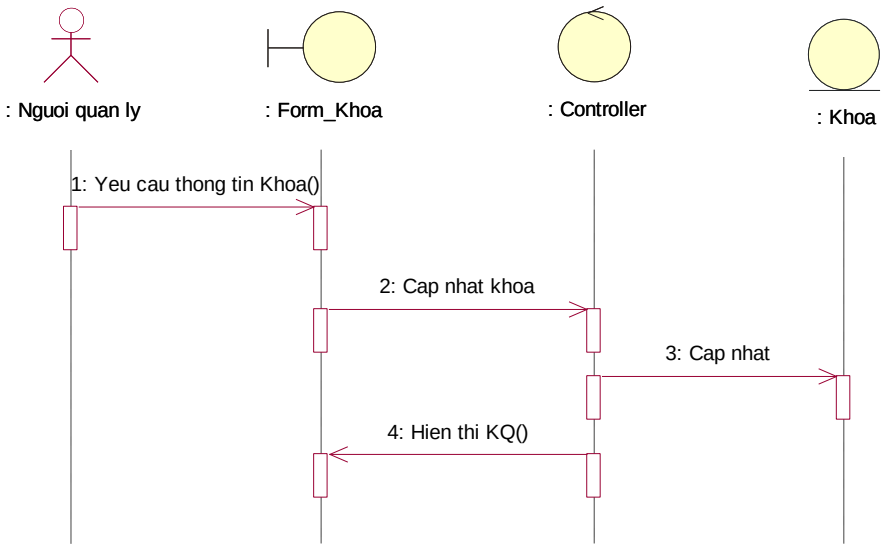
- Biểu đồ cộng tác thực thi ca sử dụng




*Hình21:* Biểu đồ cộng tác thực thi ca sử dụng "Cập nhật học phần"

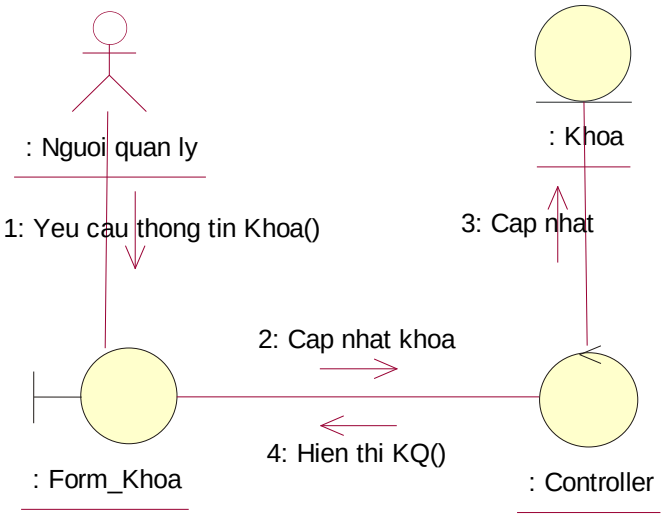
***e.Ca sử dụng "Cập nhật khoa"***

- Biểu đồ tuần tự thực thi ca sử dụng



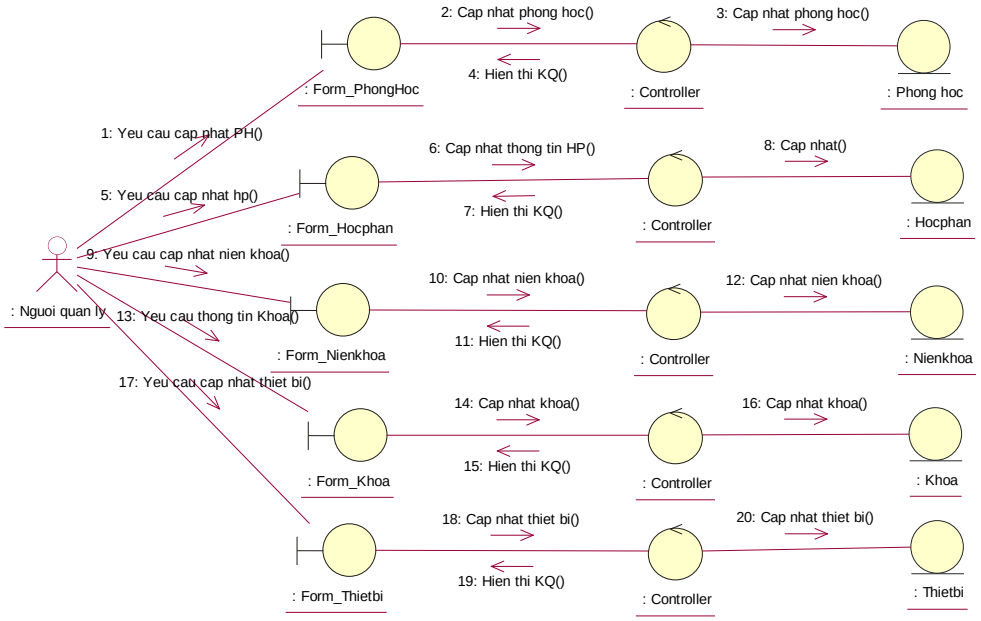

*Hình22:* Biểu đồ tuần tự thực thi ca sử dụng "Cập nhật khoa"

- Biểu đồ cộng tác thực thi ca sử dụng




*Hình23:* Biểu đồ cộng tác thực thi ca sử dụng "Cập nhật khoa"

### *f. Mô hình phân tích gói ca "Cập nhật dữ liệu"*

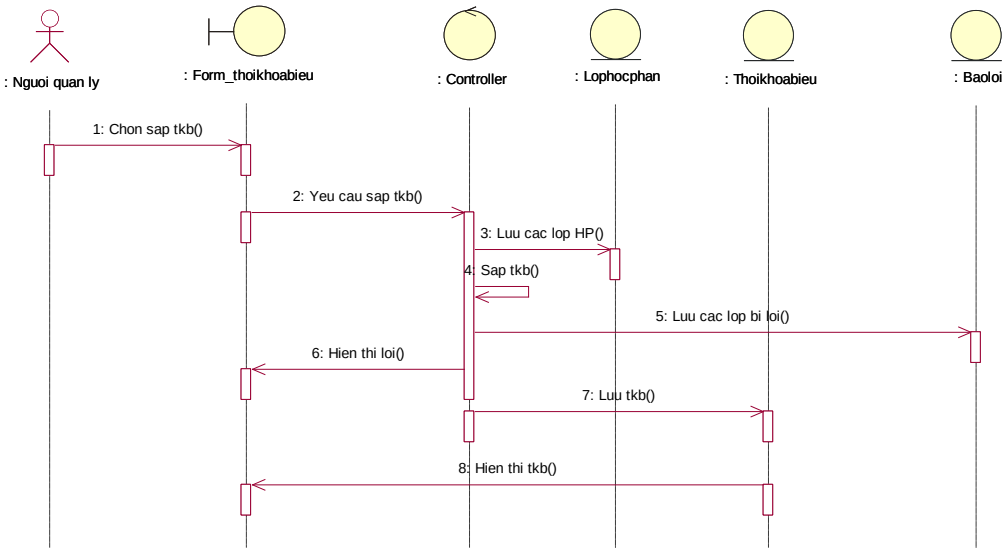



*Hình 24:* Mô hình phân tích gói ca "Cập nhật dữ liệu "

### 3.3.2 Phân tích gói ca sử dụng "Lập thời khóa biểu"

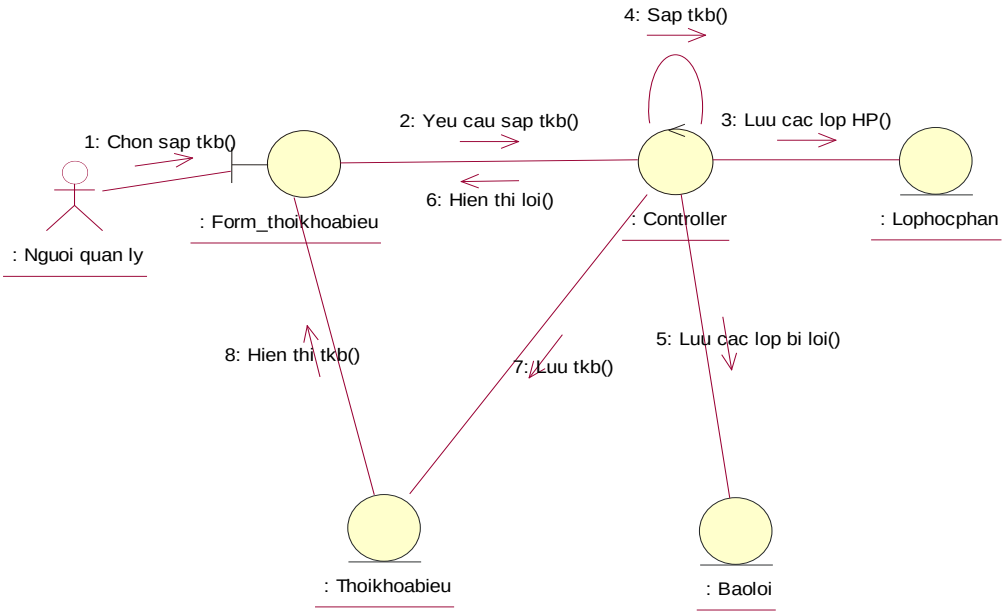
#### *a.Ca sử dụng "Sắp thời khóa biểu"*

- Biểu đồ tuần tự thực thi ca sử dụng




*Hình 25:* Biểu đồ tuần tự thực thi ca sử dụng "Sắp thời khóa biểu"

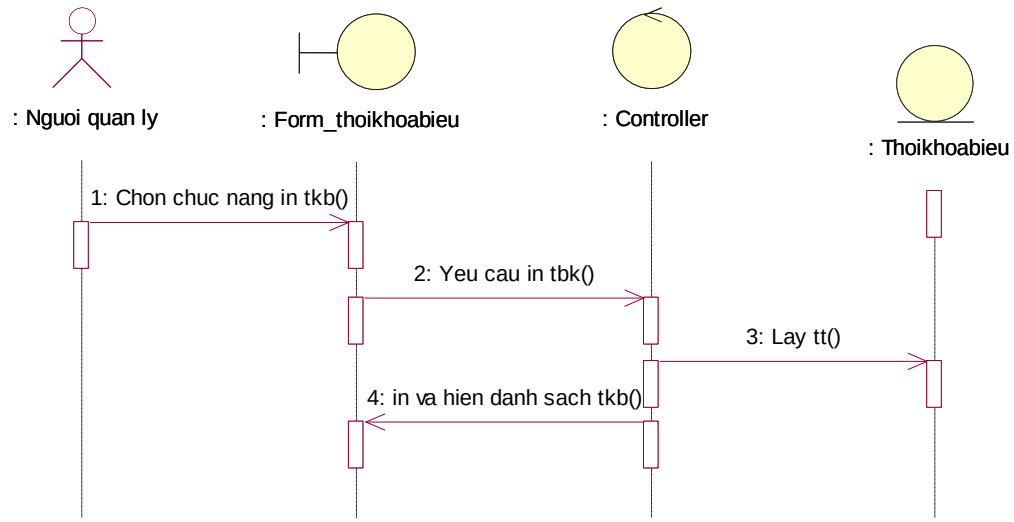
- Biểu đồ cộng tác thực thi ca sử dụng




*Hình 26:* Biểu đồ cộng tác thực thi ca sử dụng "Sắp thời khóa biểu"

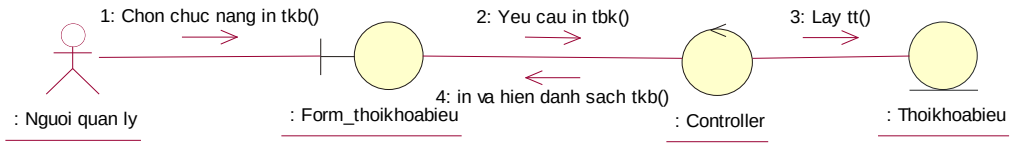
### *b.Ca sử dụng "In thời khóa biểu"*

- Biểu đồ tuần tự thực thi ca sử dụng



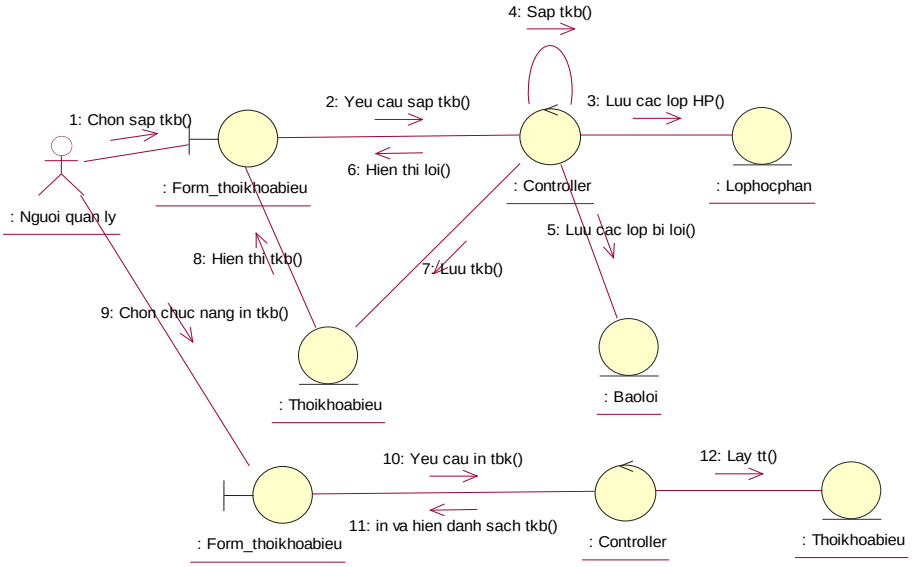

*Hình27:* Biểu đồ tuần tự thực thi ca sử dụng "In thời khóa biểu"

- Biểu đồ cộng tác thực thi ca sử dụng



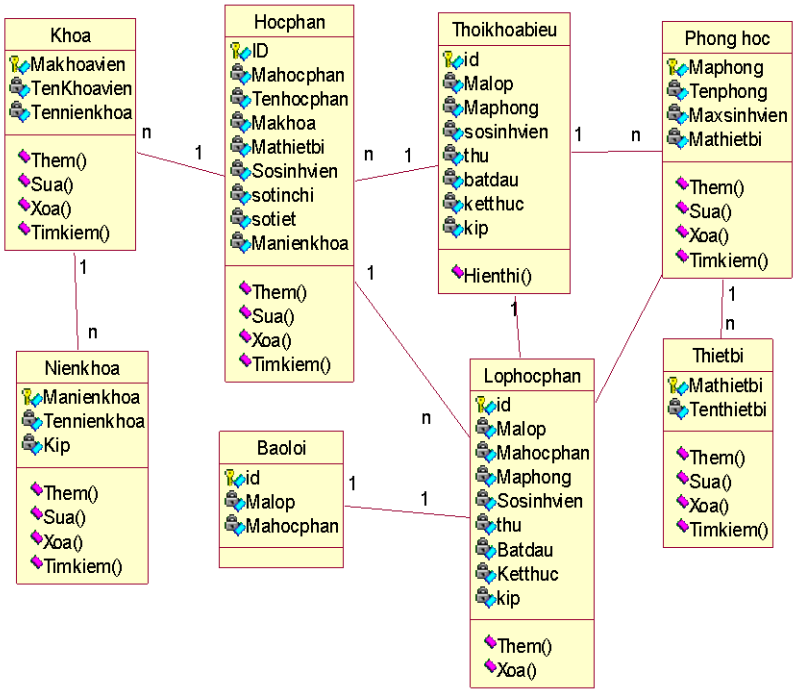

*Hình 28:* Biểu đồ cộng tác thực thi ca sử dụng "In thời khóa biểu"

***c. Mô hình phân tích gói ca "Lập thời khóa biểu"***




Hình29: Mô hình phân tích gói ca "Lập thời khóa biểu"

## 3.4 Thiết kế hệ thống




Hình30: Mô hình cơ sở dữ liệu

### 3.5Thuật toán sử dụng.

Thuật toán sắp xếp dựa trên các điều kiện ràng buộc về số sinh viên, trang thiết bị yêu cầu. Thu nhỏ không gian tìm kiếm bằng việc loại trừ các trường hợp không thỏa mãn.

**Cụ thể:**

Duyệt các lớp học phần sau khi chia học phần thành các lớp học phần.

Kiểm tra điều kiện số tiết học còn lại (chưa được xếp vào phòng học) nếu khác 0 thì tìm phòng xếp cho lớp học phần đó, nếu không thì bỏ qua.

Với các phòng thì chia trạng thái phòng đó thành các thứ từ thứ 2 đến thứ 6; mỗi thứ có 6 tiết một buổi học. Sau khi xếp 1 lớp học phần vào một phòng và một thứ cụ thể thì số tiết trống của phòng đó vào ngày thứ đó giảm đi cho tới khi bằng 0 thì chuyển sang xét tiếp ngày thứ khác. Khi một phòng đầy hết từ thứ 2 đến thứ 6 thì chuyển sang phòng kế tiếp.

Các lớp học phần không tìm được phòng học thỏa mãn sẽ được lưu riêng ra một bảng trong CSDL để thông báo cho người quản trị.

Sử dụngcách truy vẫn trực tiếp vào CSDL để lưu, sửa hoặc kết xuất dữ liệu nhanh.

Các thông tin sau khi sắp xếp sẽ lưu vào bảng thoikhoabieu trong CSDL rồi in ra giao diện đồ họa.

### 3.6Kết quả của chương trình minh họa.

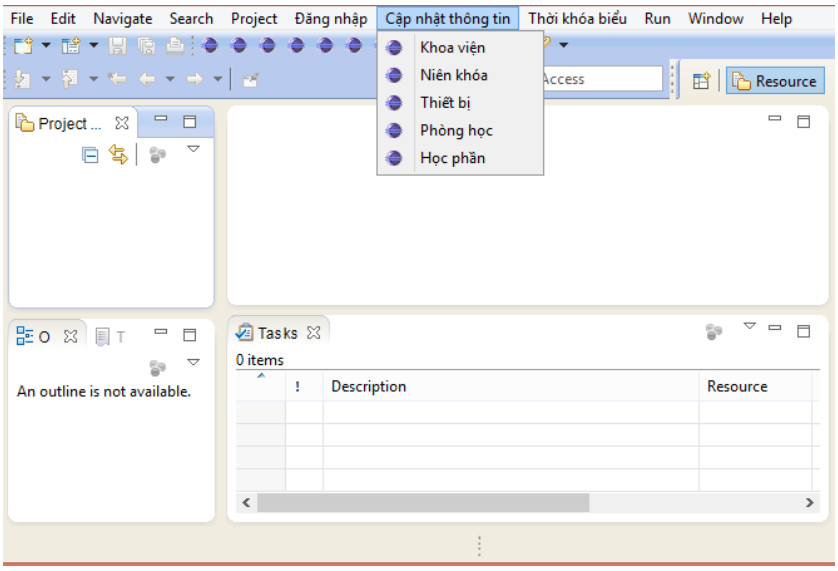
- Chương trình đã giải quyết được vấn đề cơ bản mà bài toán đưa ra là sắp xếp lịch học đơn giản, chương trình chạy ổn định, ít phát sinh lỗi.

- Chương trình đã xây dựng được các chức năng cơ bản mà bài toán đưa ra:

- *Sắp xếp thời khóa biểu thỏa mãn các ràng buộc của bài toán.*
- *Cập nhật, thêm mới, xóa, lưu các thông tin về phòng học, học phần, lớp học…*
- *Cho phép người quản lý sắp xếp thời khóa biểu tự động, xem thời khóa biểu*
- *Có chức năng xuất thời khóa biểu ra file excel.*

## 3.7. Giao diện của chương trình.

### 3.7.1.Giao diện chính của chương trình.



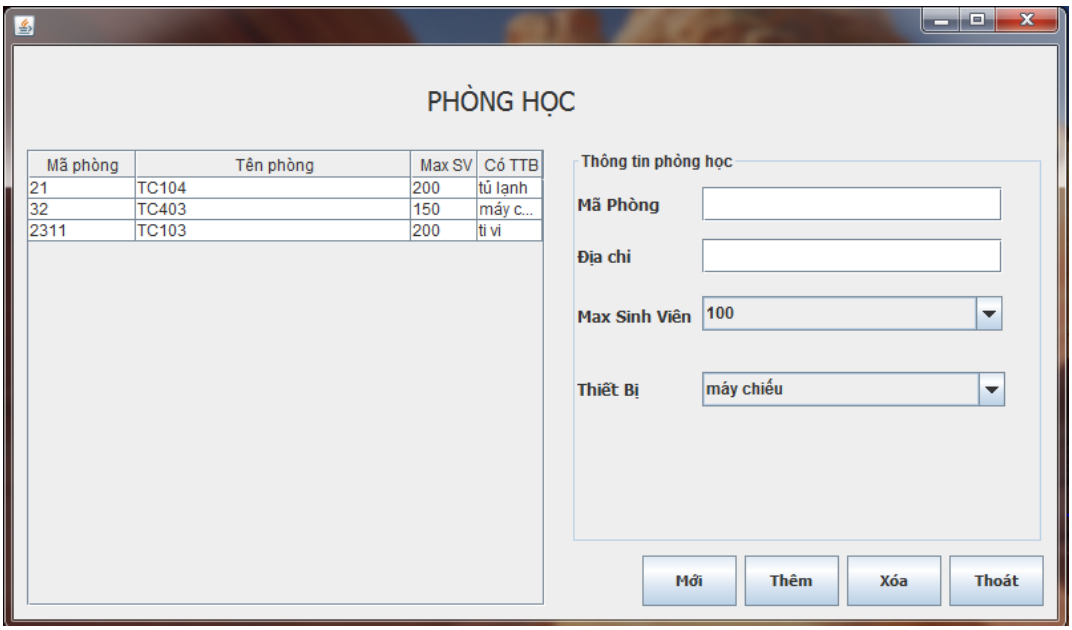

*Hình31: Giao diện chính của Plugin thời khóa biểu*

### 3.7.2 Giao diện đăng nhập

- Người quản lý nhập tên đăng nhập và mật khẩu để kết nối với cơ sở dữ liệu MySQL và chọn đăng nhập để sử dụng chương trình.


*Hình32: Giao diện đăng nhập*

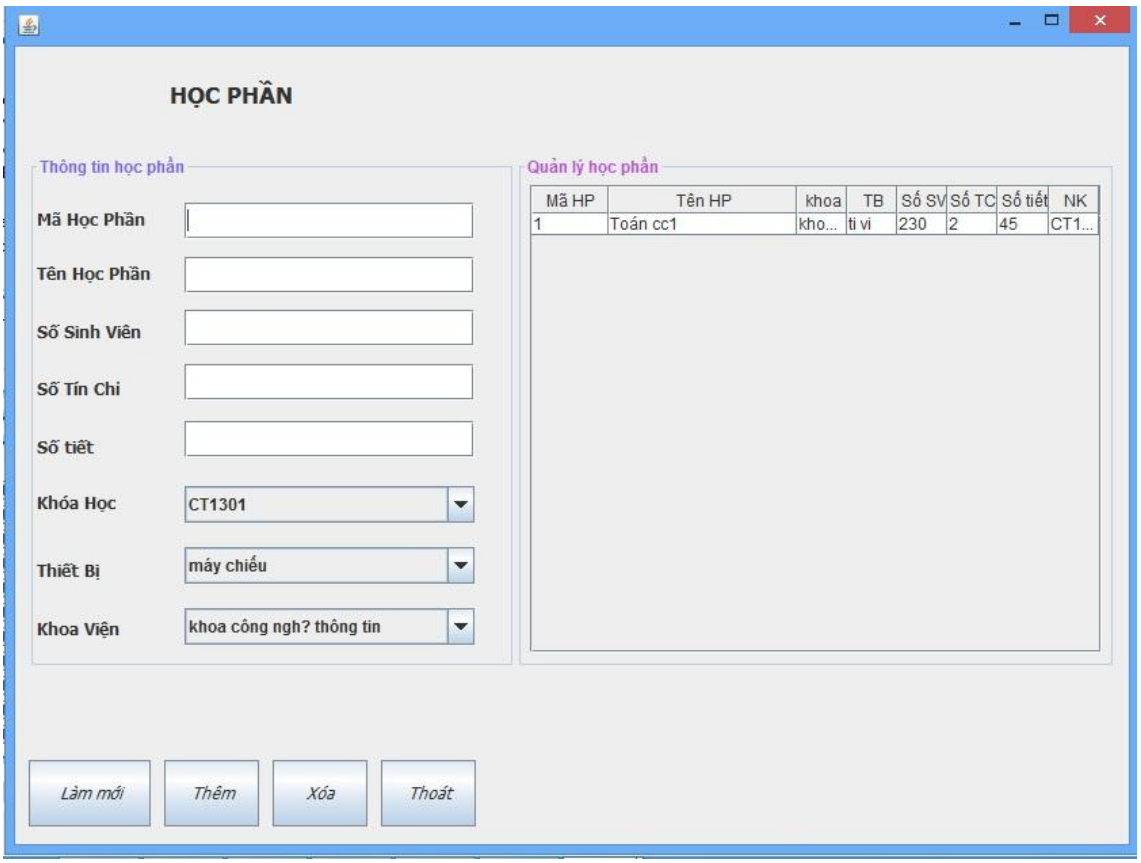
### 3.7.3 Giao diện phòng học




*Hình33: Giao diện phòng học*

- Chức năng **Mới**: sẽ reset mã phòng và địa chỉ phòng trên giao diện.
- Chức năng **Thêm**: Khi người quản lý muốn thêm phòng học vào cơ sở dữ liệu. Người quản lý sẽ nhập vào mã phòng và địa chỉ phòng tương ứng. Click chuột vào chức năng thêm, chương trình sẽ thêm vào cơ sở dữ liệu và hiển thị kết quả ở bảng danh sách các phòng trên giao diện nếu dữ liệu nhập vào không trùng với dữ liệu đã có.
- Chức năng **Xóa**: giúp người quản lý xóa phòng học. Nhập mã phòng của phòng muốn xóa, chọn **Xóa** thì các thông tin của phòng này sẽ bị xóa, nếu đang có trong cơ sở dữ liệu. Kết quả được hiển thị trên bảng.
- Chức năng **Thoát**: Thoát khỏi giao diện phòng học.

### 3.7.4 Giao diện học phần




*Hình34 Giao diện học phần*

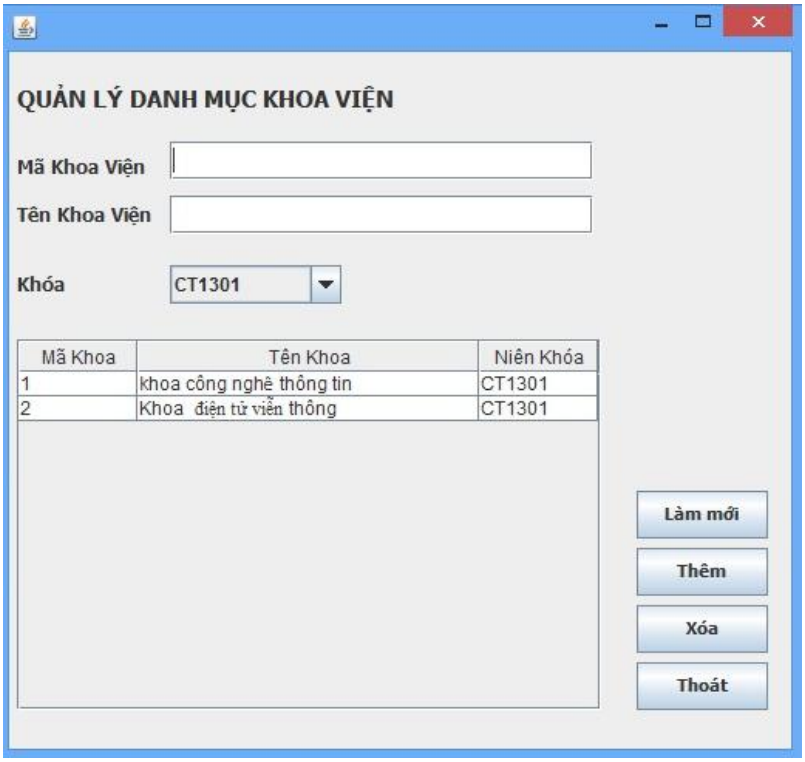
▪ Chức năng **Làm mới**: sẽ reset mã học phần, tên học phần, số sinh viên, số tín chỉ, số tiết trên giao diện.

▪ Chức năng **Thêm**: Khi người quản lý muốn thêm học phần vào cơ sở dữ liệu. Người quản lý sẽ nhập vào mã học phần, tên học phần, số sinh viên, số tín chỉ, số tiết, khóa học, thiết bị, khoa viện tương ứng. Click chuột vào chức năng thêm, chương trình sẽ thêm vào cơ sở dữ liệu và hiển thị kết quả ở bảng danh sách các phòng trên giao diện nếu dữ liệu nhập vào không trùng với dữ liệu đã có.

▪ Chức năng **Xóa**: giúp người quản lý xóa hocp phần. Nhập mã học phần của học phần muốn xóa, chọn **Xóa** thì các thông tin của phòng này sẽ bị xóa trong cơ sở dữ liệu. Kết quả được hiển thị trên bảng.

▪ Chức năng **Thoát**: Thoát khỏi giao diện học phần

### 3.7.5 Giao diện khoa viện




*Hình35: Giao diện khoa viện*

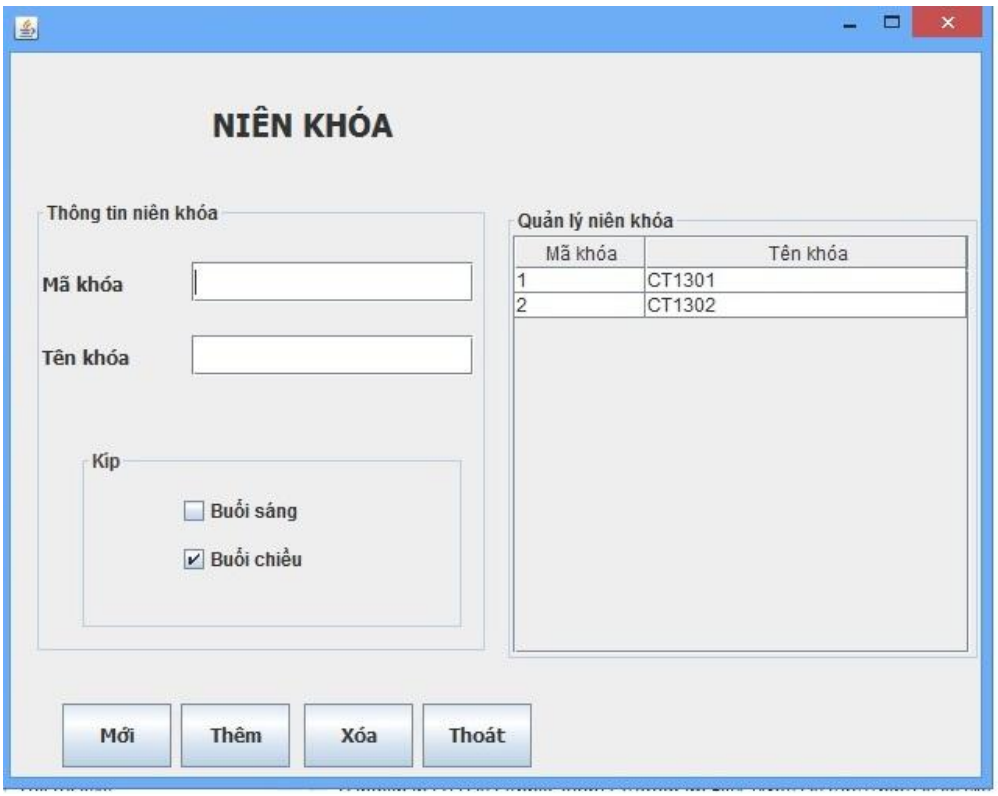
▪ Chức năng **Làm mới**: sẽ reset mã khoa viện, tên khoa viện, niên khóatrên giao diện.

▪ Chức năng **Thêm**: Khi người quản lý muốn thêm thông tin khoa viện vào cơ sở dữ liệu. Người quản lý sẽ nhập vào mã khoa viện, tên khoa viện, niên khóatương ứng. Click chuột vào chức năng thêm, chương trình sẽ thêm vào cơ sở dữ liệu và hiển thị kết quả ở bảng danh sách các khoa trên giao diện nếu dữ liệu nhập vào không trùng với dữ liệu đã có.

▪ Chức năng **Xóa**: giúp người quản lý xóa thông tin khoa viện. Nhập mã khoa viện của học phần muốn xóa, chọn **Xóa** thì các thông tin của khoa viện này sẽ bị xóa trong cơ sở dữ liệu. Kết quả được hiển thị trên bảng.

▪ Chức năng **Thoát**: Thoát khỏi giao diện học phần

### 3.2.6 Giao diện niên khóa




*Hình36: Giao diện niên khóa*

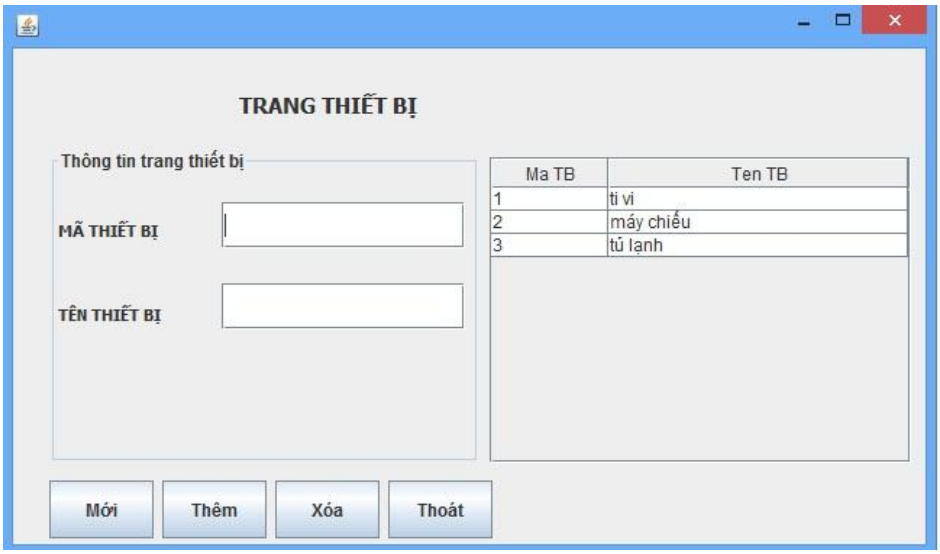
▪ Chức năng M**ới**: sẽ reset mã khóa, tên khóa trên giao diện.

▪ Chức năng **Thêm**: Khi người quản lý muốn thêm thông tin niên khóa vào cơ sở dữ liệu. Người quản lý sẽ nhập vào mã khóa, tên khóa tương ứng và chọn kip học. Click chuột vào chức năng thêm, chương trình sẽ thêm vào cơ sở dữ liệu và hiển thị kết quả ở bảng danh sách các niên khóa trên giao diện nếu dữ liệu nhập vào không trùng với dữ liệu đã có.

▪ Chức năng **Xóa**: giúp người quản lý xóa thông tin niên khóa. Nhập mã niên khóa của học phần muốn xóa, chọn **Xóa** thì các thông tin của niên khóa này sẽ bị xóa trong cơ sở dữ liệu. Kết quả được hiển thị trên bảng.

▪ Chức năng **Thoát**: Thoát khỏi giao diện học phần

### 3.7.7 Giao diện trang thiết bị




*Hình37: Giao diện trang thiết bị*

▪ Chức năng **Làm mới**: sẽ reset mã thiết bị, tên thiết bị trên giao diện.

▪ Chức năng **Thêm**: Khi người quản lý muốn thêm thông tin trang thiết bị vào cơ sở dữ liệu. Người quản lý sẽ nhập vào mã thiết bị, tên thiết bị tương ứng. Click chuột vào chức năng thêm, chương trình sẽ thêm vào cơ sở dữ liệu và hiển thị kết quả ở bảng danh sách các thiết bị trên giao diện nếu dữ liệu nhập vào không trùng với dữ liệu đã có.

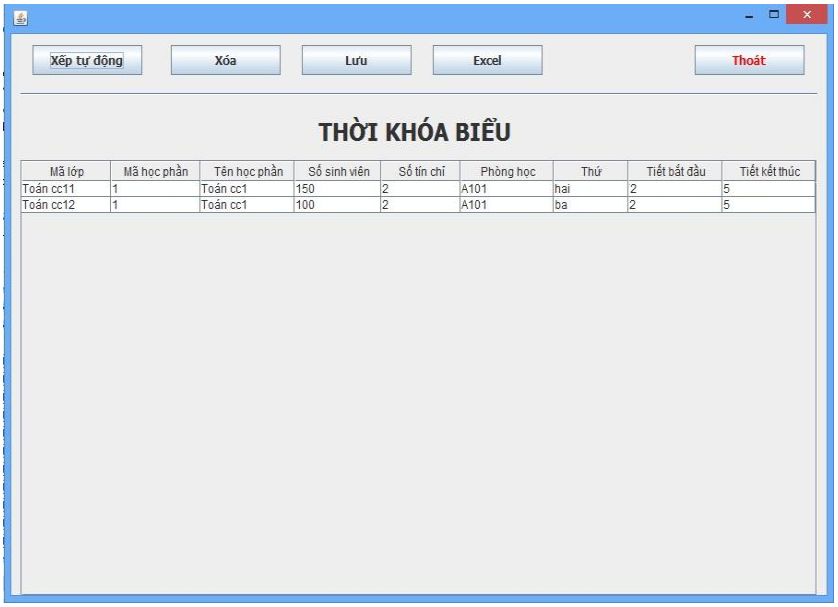
▪ Chức năng **Xóa**: giúp người quản lý xóa thông tin thiết bị. Nhập mãthiết bị của thiết bị muốn xóa, chọn **Xóa** thì các thông tin của thiết bị này sẽ bị xóa trong cơ sở dữ liệu. Kết quả được hiển thị trên bảng.

▪ Chức năng **Thoát**: Thoát khỏi giao diện thiết bị

### 3.7.8 Giao diện thời khóa biểu

- Giao diện thời khóa biểu với các chức năng: xếp tự động, xếp tay, xóa, lưu, excel, thoát.

- **Xếp tự động**: Hiển thị thời khóa biểu được chương trình sắp xếp.
- **Lưu**: Lưu thời khóa biểu đã được sắp xếp.
- **Excel**: Xuất thời khóa biểu ra fiel excel.
- **Thoát**: Thoát khỏi giao diện thời khóa biểu.




*Hình38:Giao diện thông tin thời khóa biểu*

# KẾT LUẬN

Phát triển ứng dụng trên nền *Eclipse*đang được các nhà phát triển quan tâm. Trong quá trình nghiên cứu, tìm hiểu, phân tích đề tài, khóa luận đã đạt được một số kết quả sau:

- Trình bày về sự phát triển các ứng dụng dựa trên *Eclipse;*
- Kiến trúc của Plugin trong *Eclipse;*
- Xây dựng ứng dụng thử nghiệm;
- Xu hướng phát triển phần mềm dựa trên thành phần phần mềm.
- và thời gian thực hiện khóa luận giúp em hiểu biết nhiều kiến thức về lĩnh vực phát triển phần mềm dựa trên thành phần cũng như củng cố kiến thức đã học trong trường;

Bài toán sắp thời khóa biểu được xây dựng bằng *Plugin* đã thể hiện một cách xây dựng ứng dụng phần mềm có thể cài đặt trên nhiều *platform* khác nhau mà không phụ thuộc vào hệ thống cài đặt chúng. Đồng thời, ứng dụng bước đầu đạt được một số chức năng chính, có thể sắp thời khóa biểu cho các lớp vào đầu mỗi học kỳ, hầu hết các chức năng sau khi xử lý xong đều hoạt động tốt, đáp ứng được yêu cầu đặt ra. Nhưng vẫn còn nhiều hạn chế: giải thuật chưa được tối ưu nên đôi khi kết quả đạt được chưa thỏa mãn, chưa tạo được sự linh động trong việc hỗ trợ các mức độ ưu tiên, chưa có chức năng tinh chỉnh thời khóa biểu

Em mong nhận được nhiều ý kiến đóng góp để *Plugin* thời khóa biểu được phát triển hơn trong thời gian tới và đáp ứng đầy đủ mọi yêu cầu

Mặc dù em đã cố gắng rất nhiều trong quá trình tìm hiểu và phân tích đề tài nhưng không tránh khỏi những thiếu sót.Vì vậy em mong quý thầy cô cũng như những ai quan tâm đến đề tài này chỉ dẫn và góp ý kiến cho em để em hoàn thiện ứng dụng này một cách đầy đủ.

Mong muốn của em là phát triển ứng dụng này tối ưu và nhiều tính năng hơn.

# TÀI LIỆU THAM KHẢO

[1] Nguyễn Văn Vỵ (2002), Phân tích thiết kế các hệ thống thông tin hiện đại, hướng cấu trúc vàhướng đối tượng, NXB Thống kê, Hà Nội.

[2] Đoàn Văn Ban (2003), Phân tích thiết kế hướng đối tượng bằng UML, NXB Thống kê.

[3] Đặng Văn Đức (2000), Phân tích hướng đối tượng bằng UML, NXB Giáo dục, Hà Nội.

[4] Carlos Valcarcel (2005), Eclipse KickStart (Ver 3.0)

[5] David Gallardo (developerWorks, 12. 2002), Getting started with the EclipsePlatform

[6] Dr Alex Blewitt, Eclipse 4 Plug-in Development by Example