***Mục lục***

[PHẦN 1 – MỞ ĐẦU 3](#_Toc319278543)

[1. Lý do chọn đề tài: 3](#_Toc319278544)

[2. Mục tiêu 3](#_Toc319278545)

[3. Phạm vi nghiên cứu 4](#_Toc319278546)

[PHẦN 2 – TỔNG QUAN 4](#_Toc319278547)

[PHẦN 3 – CƠ SỞ LÝ THUYẾT 5](#_Toc319278548)

[CHƯƠNG 1: CÔNG NGHỆ J2ME 5](#_Toc319278549)

[1.2 Các thành phần của J2ME 6](#_Toc319278550)

[1.2.1 Profile 6](#_Toc319278551)

[1.2.2 Configuration 7](#_Toc319278552)

[**1.2.3 Máy ảo java (java virtual machine)** 8](#_Toc319278553)

[1.3 MIDlet 8](#_Toc319278554)

[1.4 Lưu trữ dữ liệu trên thiết bị di động (RMS): 9](#_Toc319278555)

[CHƯƠNG 2: MÔ HÌNH CLIENT – SERVER 11](#_Toc319278556)

[2.1 Định nghĩa 11](#_Toc319278557)

[2.2 Mô hình 12](#_Toc319278558)

[*2.2.1 Giao thức TCP* 13](#_Toc319278559)

[2.2.2 Giao thứcUDP 13](#_Toc319278560)

[CHƯƠNG 3: KIẾN THỨC HỆ CHUYÊN GIA 15](#_Toc319278561)

[3.1 Định nghĩa: 15](#_Toc319278562)

[3.2 Các kỹ thuật suy diễn 16](#_Toc319278563)

[3.2.1 Suy diễn tiến 16](#_Toc319278564)

[3.2.2 Suy diễn lùi 17](#_Toc319278565)

[3.3 Cấu trúc hệ chuyên gia 18](#_Toc319278566)

[3.3.4 Bộ tiếp nhận tri thức 19](#_Toc319278567)

[3.3.5 Cơ sở tri thức 19](#_Toc319278568)

[3.3.6 Vùng nhớ làm việc 19](#_Toc319278569)

[3.4 Ứng dụng hệ chuyên gia: 19](#_Toc319278570)

[*3.5 Biểu diễn tri thức* 20](#_Toc319278571)

[3.5.1 Dạng Object – Attribute – Value 20](#_Toc319278572)

[3.5.2 Dạng Frame 21](#_Toc319278573)

[3.5.3 Dạng các luật dẫn 23](#_Toc319278574)

[3.5.4 Dạng mạng ngữ nghĩa 25](#_Toc319278575)

[3.5.5 Dạng logics 25](#_Toc319278576)

[**3.5.5.1 Logic mệnh đề** 25](#_Toc319278577)

[**3.5.5.2 Logic vị từ** 26](#_Toc319278578)

[PHẦN 4 – NGHIÊN CỨU THỰC NGHIỆM ÁP DỤNG CHƯƠNG TRÌNH CỐ VẤN HỌC TẬP ĐIỆN TỬ 26](#_Toc319278579)

[CHƯƠNG 1 – CÁC YÊU CẦU 26](#_Toc319278580)

[1.2 Yêu cầu phần cứng: 26](#_Toc319278581)

[1.3 Yêu cầu phần mềm: 27](#_Toc319278582)

[1.4 Môi trường thực thi (thiết bị) 27](#_Toc319278583)

[CHƯƠNG 2 – THỰC THI 27](#_Toc319278584)

[2.1 Phân tích yêu cầu và thiết kế các chức 27](#_Toc319278585)

[2.2 Thiết kế CSDL: các bảng dữ liệu 28](#_Toc319278586)

[2.3 Phân tích các thuật toán áp dụng: hệ chuyên gia suy diễn tiến 29](#_Toc319278587)

[2.4 Thiết kế các mô hình giao tiếp giữa Client – Server 34](#_Toc319278588)

[Xây dựng từ đầu Clientnetworking và Servernetworking 34](#_Toc319278589)

[CHƯƠNG 3 – DEMO ỨNG DỤNG 34](#_Toc319278590)

[PHẦN 5 – KẾT QUẢ THỰC NGHIỆM ĐỀ TÀI 36](#_Toc319278591)

[PHẦN 6 – HƯỚNG PHÁT TRIỂN 36](#_Toc319278592)

[PHẦN 7 – TÀI LIỆU THAM KHẢO 36](#_Toc319278593)

[PHẦN 8 – PHỤ LỤC 36](#_Toc319278594)

# PHẦN 1 – MỞ ĐẦU

## 1. Lý do chọn đề tài:

Cùng với việc hệ thống giáo dục theo tín chỉ ra đời thay thế cho hệ niên chế giúp cho sinh viên chủ động hơn trong việc học tập, rút ngắn thời gian học tập tại trường đại học cũng như tập trung cho chuyên môn của mình làm nền tảng kiến thức sau khi ra trường. Việc này giúp sinh viên có nhiều sự chọn lựa hơn về môn học của mình. Và việc chọn lựa như thế nào cho phù hợp với khả năng cũng như nghành học của mình thì cần những cố vấn học tập thường là các thầy giáo bộ môn do khoa phân.

Một Giảng viên bộ môn thường chỉ chuyên về bộ môn đó, trong khi một tập thể Sinh viên lại chọn lựa theo nhiều chuyên môn khác nhau, hơn nữa Giảng viên không thể hướng dẫn hết cho từng Sinh viên trong một đợt đăng kí tín chỉ diễn ra nhanh trong vài ngày.

Yêu cầu đặt ra là làm thế nào để có thể tư vấn cho sinh viên trong mỗi đợt đăng kí một cách nhanh nhất và hiệu quả nhất. Theo đó, việc tạo ra một hệ tri thức hoàn chỉnh để tư vấn cho sinh viên qua mỗi đợt đăng kí tín chỉ và được sử dụng trên thiết bị cầm tay (điện thoại cấu hình thấp) mà hầu như sinh viên nào cũng có.

## 2. Mục tiêu

Tạo ra một hệ chuyên gia đơn giản giúp cho sinh viên trong việc chọn lựa môn học đăng kí phù hợp với khả năng và nghành học của mình trong mỗi đợt đăng kí tín chỉ.

Chạy được trên nhiều môi trường.

## 3. Phạm vi nghiên cứu

Trong khuông khổ đề tài được xây dựng trên mô hình Client – Server. Client là thiết bị cấu hình thấp chạy trên nền tảng J2ME và Server là máy PC chạy trên nền tảng J2SE.

Ứng dụng được cài đặt thử lên thiết bị ảo (emulator)

# PHẦN 2 – TỔNG QUAN

Với một sinh viên, việc xem thời khóa biểu, đăng kí môn học theo mỗi đợt hoặc xem điểm sau khi kết thúc môn học hiện tại đang diễn ra dưới dạng web application, tức là thông qua server web. Đề tài này đề cập theo một hướng khác là xây dựng riêng một hệ thống Client – Server giao tiếp qua socket chạy trên môi trường thiết bị di động cấu hình thấp, trong đó ứng dụng chính là cố vấn học tập cho sinh viên. Có 2 điểm khác biệt so với phương pháp đang sử dụng là:

* Với mục đích nghiên cứu công nghệ mới thay vì sử dụng web server
* Thiết bị cầm tay được sử dụng thay vì máy tính cá nhân có kết nối internet
* Xây dựng thêm hệ tư vấn học tập cho sinh viên

Đề tài này là kết hợp của 2 nền tảng công nghệ cùng với việc xây dựng một hệ chuyên gia tư vấn đơn giản

Các nội dung chính được nêu trong luận văn này bao gồm:

**Phần mở đầu:** Dẫn dắt đề tài, về nội dung cần nghiên cứu trong toàn bộ luận văn.

**Phần cơ sở lý thuyết:** bao gồm những lý thuyết được áp dụng để phát triển ứng dụng của đề tài. Công nghệ J2ME, J2SE để phát triển hệ thống client – server và Hệ chuyên gia để giải quyết vấn đề tư vấn học tập

* Chương 1: J2ME: Giới thiệu về môi trường chạy ứng dụng trên Client
* Chương 2: Giới thiệu mô hình Client – Server
* Chương 3: Hệ chuyên gia: Những kiến thức cơ bản nhất về hệ chuyên gia

Các vấn đề về hệ chuyên gia bao gồm:

+ Quản trị tri thức

+ Mô tơ suy diễn

+ Giao diện

+ Hỏi đáp

+ Thu nạp tri thức

+ Hệ chuyên gia phân tán

**Phần nghiên cứu thực nghiệm:**

* Phần ứng dụng là chương trình cố vấn học tập điện tử dựa trên mô hình Client – Server sử dụng hệ chuyên gia để tư vấn cho Sinh viên những môn học nên đăng kí trong một đợt đăng kí học phần.

**Phần kết quả thực nghiệm:** Nêu ra những kết quả đạt được sau khi kết thúc nghiên cứu đề tài

**Phần Hướng phát triển:** Nêu ra những tồn tại, những vấn đề còn chưa giải quyết được của yêu cầu đề tài đặt ra và hướng phát triển, xây dựng đề tài hoàn thiện hơn

**Phần tài liệu tham khảo:** Những tài liệu dùng đề nghiên cứu đề tài. Bao gồm phần lý thuyết và thực hành áp dụng

**Phần phụ lục:** Những từ viết tắt dùng trong đề tài

# PHẦN 3 – CƠ SỞ LÝ THUYẾT

## CHƯƠNG 1: CÔNG NGHỆ J2ME

**1.1 Tổng quan về công nghệ J2ME**

Công nghệ J2ME là công nghệ java được thực thi dành riêng cho thiết bị có cấu hình thấp như điện thoại di động …

Các vấn đề được đề cập trong phần J2ME này bao gồm chủ yếu là kiến thức cơ bản về J2ME

Trong đề tài này chỉ để cập đến công nghệ J2ME với tư cách là môi trường thực thi.

Công nghệ J2ME bao gồm MIDP

CLDC là các thiết bị cấu hình



*H1.1 Cấu trúc công nghệ J2ME*

### 1.2 Các thành phần của J2ME

#### 1.2.1 Profile

Profile là một khái niệm mở rộng của Configuration. Profile định nghĩa các thư viện giúp lập trình viên phát triển các ưng dụng cho một dạng thiết bị nào đó. Ta có các profile như : MIDP, PDA…

MIDP định nghĩa các hàm API cho các thành phần giao diện, nhập liệu và xử lý sự kiện, lưu trữ, kết nối mạng, quản lý thời gian… phù hợp với màn hình hiển thị và khả năng xử lý của các thiết bị di động. Profile được định nghĩa trên nền tảng CLDC.

MIDP cung cấp các thư viện giao diện người dùng có các đặc trưng :

* Đủ bộ nhớ để chạy các ứng dụng của MIDP
* Độ phân giải màn hình tối thiểu 96 x 56 pixels, màu hoặc trắng đen
* Bộ phím : Keypad, keyboard, hoặc touch screen
* Kết nối mạng không dây 2 chiều ( bluetoot)

Với bộ MIDP 3.0 có các tính năng :

* Hỗ trợ tốt cho thiết bị có màn hình rộng
* Cho phép MIDlets tương tác với Display thứ 2 (khác current display)
* Nâng cao hiệu suất cho các thiết bị
* Hỗ trợ bảo mật cho lưu trữ RMS
* Removable/Remote RMS
* IPv6
* Hỗ trợ nhiều giao thức mạng cho cùng một thiết bị

#### 1.2.2 Configuration

Không cung cấp giao diện người dùng bao gồm CLDC và CDC :Được xem như là một dạng máy ảo đặc trưng của java có các đặc trưng về quản lý bộ nhớ, độ phân giải màn hình, giao thức kết nối. Có 2 bộ Configuration trong J2ME

* **CDLC (Connected Limited Device Configuration - Cấu hình thiết bị kết nối giới hạn):** được thiết kế để nhắm vào thị trường các thiết bị cấp thấp (low-end), các thiết bị này thông thường là máy điện thọai di động và PDA với khoảng 512 KB bộ nhớ. Vì tài nguyên bộ nhớ hạn chế nên CLDC được gắn với Java không dây (Java Wireless ), dạng như cho phép người sử dụng mua và tải về các ứng dụng Java, ví dụ như là Midlet.
* **CDC (Connected Device Configuration - Cấu hình thiết bị kết nối):** CDC được đưa ra nhắm đến các thiết bị có tính năng mạnh hơn dòng thiết bị thuộc CLDC nhưng vẫn yếu hơn các hệ thống máy để bàn sử dụng J2SE. Những thiết bị này có nhiều bộ nhớ hơn (thông thường là trên 2Mb) và có bộ xử lý mạnh hơn. Các sản phẩm này có thể kể đến như các máy PDA cấp cao, điện thoại web, các thiết bị gia dụng trong gia đình …

##### **1.2.3 Máy ảo java (java virtual machine)**

Với tính năng đã được đề cập trước đó là môi trường trung gian giữa ứng dụng và máy thực (môi trường thực thi chính của ứng dụng) và mang tính đặc trưng của ngôn ngữ java

Với CDC, máy ảo java có cùng đặc tính với J2SE. Tuy nhiên với CLDC Sun (giờ đã được sát nhập với Oracle) đã phát triển riêng một dạng máy ảo chuyên biệt dành cho các thiết bị có cấu hình thấp là K Virtual machine, gọi tắt là KVM

KVM có đặc tính:

* Chỉ cần 40 – 80 Kb bộ nhớ
* Chỉ đòi hỏi tối thiểu 20 – 40 Kb bộ nhớ động
* Có thể chạy với bộ vi xử lý của thiết bị 16 bit và xung nhịp 25 MHz
* Nếu chương trình được biên dịch với CDC, chương trình sẽ chạy máy ảo truyền thống JVM (Java Virtual Machine) và mang các đặc tính như J2SE
* Nếu chương trình được biên dịch với CLDC, chương trình sẽ chạy máy ảo KVM (K Virtual Machine) và mang các đặc tính của J2ME.
* Vì tính đặc thủ nhỏ gọn của trình biên dịch CLDC nên một số chương trình viết bằng MIDP sẽ không chạy được trong môi trường J2SE và ngược lại

Có một bộ máy ảo được thiết kế riêng cho thiết bị cấu hình thấp là KVM

### 1.3 MIDlet

MIDlet được tạo ra trong một ứng dụng J2ME để thực thi chương trình. MIDP không bắt đầu từ phương thức static main (giống như bình thường), thay vào đó nó sử dụng một MIDlet



*H1.2 Vòng đời 1 MIDlet*

### 1.4 Lưu trữ dữ liệu trên thiết bị di động (RMS):

RMS (Recort Management System) là hệ thống quản lý bản ghi dưới dạng record. Mỗi Record có thể chứa bất kỳ loại dữ liệu nào, chúng có thể là kiểu số nguyên, chuỗi ký tự hay có thể là một ảnh và kết quả là một Record là một chuỗi (mảng) các byte.

Một tập các RecordStore là tập hợp các Record được sắp xếp có thứ tự. Mỗi Record không thể đứng độc lập mà nó phải thuộc vào một RecordStore nào đó, các thao tác trên Record phải thông qua RecordStore chứa nó. Khi tạo ra một Record trong RecordStore, Record được gán một số định danh kiểu số nguyên gọi là Record ID. Record đầu tiên được tạo ra sẽ được gán Record ID là 1 và sẽ tăng thêm 1 cho các Record tiếp theo.



*H1.3 Cơ chế hoạt động của RMS*

Đường liền thể hiện việc truy xuất Record store do MIDlet đó tạo ra, đường nét đứt là Record store do MIDlet khác tạo ra.  
Trong MIDLET Suite One, MIDlet #1 và MIDlet #2 cùng có thể truy xuất 4 Record store. MIDLET Suite One không thể truy xuất Record store trong Suite Two. Trong MIDlet Suite One tên của các Record store là duy nhấy, tuy nhiên Record store trong các MIDlet Suite khác nhau có thể dùng chung một tên.

Với ứng dụng Cố vấn học tập trong phần này thì

* Dữ liệu sau khi nhận được từ server response lại Cliet sẽ được lưu lại để người dùng cóc thể xem lại nó.
* Sau mỗi lần có kết quả mới từ một request mới thì dữ liệu cũ sẽ bị ghi chồng lên. Tức là vùng nhớ chứa dữ liệu cũ sẽ bị thay thế để đảm bảo việc chiếm dụng vùng nhớ không nhiều đối với thiết bị có cấu hình thấp.

Trong ứng dụng này hệ thống lưu trữ dữ liệu dưới dạng bảng ghi các byte data. Dữ liệu truyền lên và trả về từ Client – Server đều dưới dạng byte

## CHƯƠNG 2: MÔ HÌNH CLIENT – SERVER

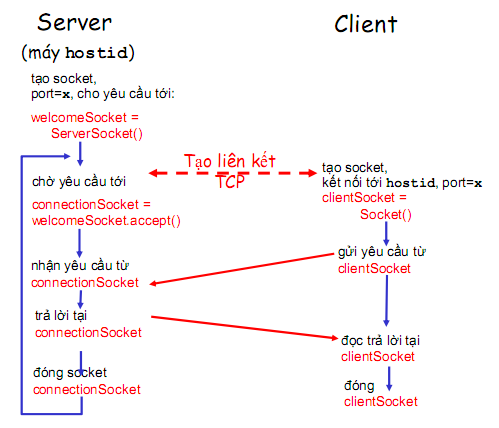
### 2.1 Định nghĩa

Mô hình Client – Server là mô hình tương tác giữa một bên là máy khách và một bên là máy chủ. Trong đó Client là máy khách thực hiện gửi một yêu cầu lên server thông qua socket. Server thực nhận dạng socket này và thực hiện tao tác xử lý sau đó trả về thông qua socket trong cùng cổng (port).

Socket là giao diện do ứng dụng tạo ra trên máy trạm, quản lý bởi hệ điều hành qua đó các ứng dụng có thể gửi và nhận các thông điệp đến/từ các ứng dụng khác. Client và Server giao tiếp thông quan socket



### 2.2 Mô hình



*H03. Mô hình client – server*

**2.2 Phương thức kết nối**

### *2.2.1 Giao thức TCP*



Có 2 phương thức kết nối là TCP truyền theo byte dữ liệu và UDP truyền theo các gói packge. Khi Server nhận yêu cầu kết nối nó sẽ chấp nhận yêu cầu và khởi tạo socket mới để giao tiếp với Client. Có thể chấp nhận nhiều Client tại cùng một thời điểm.

* **Client:** phải gửi yêu cầu tới Server bằng cách tạo một socket trên máy khách. Chỉ rõ port number trên máy Server. Khi Client tạo socket client liên kết với server
  + Khởi tạo TCP socket
  + Xác định IP address, port number của Server
  + Thiết lập kết nối đến Server
* **Server:** Tiến trình máy chủ phải đang thực thi. Máy chủ phải đang mở socket để nhận yêu cầu từ Client

- Server process phải chạy tước

- Server phải tạo một socket để lắng nghe và chấp nhận các kết nối từ Client

### 2.2.2 Giao thứcUDP



Với giao thức UDP ta có các đặc tính

* Không cần thiết lập kết nối giữa Client và Server
* Sender phải gửi kèm địa chỉ IP và port
* Server khi nhận được dữ liệu sẽ phân tích địa chỉ của sender để gửi lại
* Có thể Server chấp nhận nhiều Client tại cùng một thời điểm

**Client:**

**-** Xác định địa chỉ Server

- Tạo socket

- Gửi/ nhận dữ liệu theo giao thức lớp ứng dụng đã thiết kế

- Đóng socket

**- Server:**

## CHƯƠNG 3: KIẾN THỨC HỆ CHUYÊN GIA

### 3.1 Định nghĩa:

Hệ chuyên gia là một loại cơ sở tri thức được thiết kế cho một lĩnh vực ứng dụng cụ thể

Hệ Cơ sở tri thức là một chương trình máy tính được thiết kế để mô hình hóa khả năng giải quyết vấn đề của chuyên gia con người

Hệ cơ sở tri thức là hệ thống dựa trên tri thức cho phép mô hình hóa các tri thức của chuyên gia, dùng tri thức này để giải quyết vấn đề phức tạp thuộc cùng lĩnh vực

Hai vấn đề quan trọng của hệ tri thức là: tri thức chuyên gia và lập luận tương ứng với hệ thống có 2 khối chính là Cơ sở tri thức và động cơ suy diễn

Cơ sở tri thức chứa các tri thức chuyên sâu về lĩnh vực như chuyên gia. Cơ sở tri thức bao gồm: các sự kiện, các luật, các khái niệm và các quan hệ.

Xử lý tri thức với heuristic: Người chuyên gia có rất nhiều kinh nghiệm để giải quyết vấn đề chuyên môn của họ. Với kinh nghiệm này giúp họ giải quyết vấn đề rất nhanh. Giống như cách giải quyết vấn đề của các chuyên gia, các hệ chuyên gia hầu hết đều sử dụng thông tin heuristic thu thập được từ kinh nghiệm của người chuyên gia giúp hệ giải quyết vấn đề nhanh nhất và hiệu quả nhất.

Xử lý tri thức không chắc chắn: Hơn 80% ứng dụng trong thực tế không thể giải quyết được bằng các phương pháp lập luận chắc chắn. Hệ chuyên gia có thể giải quyết được những ứng dụng này nhờ vào các phương pháp xử lý tri thức không chắc chắn.

Giải Thích

Suy Diễn

Thu thập tri thức từ Chuyên gia

CSTT

Hệ Thống Giao Tiếp

User

Expert

### 3.2 Các kỹ thuật suy diễn

#### 3.2.1 Suy diễn tiến

Là sử dụng tập luật tương tự nhau nhằm cho việc sử dụng suy diễn lùi. Mặc dù vậy, tiến trình suy diễn có khác đi, hệ thống giữ vết cho tình trạng hiện tại của giải pháp và tìm kiếm luật, điều này sẽ dẫn đến việc đi dẫn đến giải pháp cuối cùng.

Các bước để xây dựng hệ chuyên gia suy diễn tiến

* Bước 1: định nghĩa vấn đề

Bước này gồm phân tích vấn đề nắm hướng giải quyết vấn đề chi tiết và cụ thể đó là dữ liệu vào ra của hệ thống và phương thức xử lý số liệu vào ra của hệ thống.

* Bước 2: Định nghĩa dữ liệu vào ra của hệ thống

Định nghĩa dữ liệu vào của hệ thống sao cho dữ liệu ban đầu của hệ thống phù hợp với điều kiện của luật suy diễn thứ nhất để dữ liệu đích của nó là dữ liệu vào của luật tiếp theo

* Bước 3: Định nghĩa cấu trúc điều khiển của hệ thống

Cấu trúc điều khiển dữ liệu suy diễn tiến của hệ thống đó là cơ sở luật suy diễn tiến bao gồm tất cả các luật mô tả tổng quát cách giải bài toán được thể hiện dưới dạng luật If Then với vế điều kiện của luật đầu tiên hợp với dữ liệu ban đầu của bài toán để vế phải của luật phát sinh ra đích thứ nhất, về điều kiện của luật thứ 2 và cứ thế cho đến luật phát sinh thứ n mà về kết luận của nó đạt đến lời giải cuối cùng.

* Bước 4: Mã hóa cơ sở tri thức

Cơ sở tri thức bao gồm cơ sở luật và sơ sở dữ liệu. Các thành phần này được mã hóa nhờ phương pháp biểu diễn tri thức như logic vị từ, khung ..

* Bước 5: Thử nghiệm hệ thống

Cho số liệu vào, quá trình xử lý của hệ thống trả số liệu ra với nhiều tình huống khác nhau bao trùm cả không gian vào

* Bước 6: Thiết kế hệ thống gia diện người xử dụng hệ chuyên gia
* Bước 7 : Mở rộng hệ thống

Mở rộng cơ sở tri thức của hệ sao cho giải quyết vấn đề được linh hoạt, càng mềm dẻo càng tốt. Đó là quá trình cải tiến hoặc thêm bớt suy luật suy diễn và cơ sở dữ liệu

* Bước 8: Đánh giá hệ thống

Đưa hệ thống vào thực nghiệm trong các trường hợp thực tế để rút ra kết luận đánh giá chất lượng vận hành của hệ thống đáng tin cậy hoặc chưa.

Ví dụ:

#### 3.2.2 Suy diễn lùi

Là tính hiệu quả để giải quyết vấn đề mà có thể mô hình hóa lại là cấu trúc lựa chọn các vấn đề. Trang bị cho hệ thống khả năng chọn cái tốt nhất từ những cái có thể.

Kiến thức được cấu trúc lại và đưa vào các luật, chúng mô tả làm như thế nào một khả năng nào đó được chọn. Luật chia nhỏ vấn đề thành nhiều vấn khác nhỏ hơn. Có nhiều vấn đề được đặt ra, khó mà có thể thống kê đựơc số lượng các câu trả lời và chọn được một câu đúng nhất trong tất cả những câu đó.

### 3.3 Cấu trúc hệ chuyên gia

****

*h.01 – Cấu trúc hệ chuyên gia*

***3.3.1 Giao diện người máy***

Thực hiện giao tiếp giữa hệ chuyên gia và người dùng. Nhận các thông tin, câu hỏi từ người dùng và đưa ra các lời khuyên, giải thích. Giao diện này bao gồm: Menu, bộ xử lý các ngôn ngữ tự nhiên và các hệ thống tương tác.

***3.3.2 Bộ giải thích***

Giải thích các hoạt động khi có yêu cầu của người dùng

***3.3.3 Động cơ suy diễn***

Quá trình trong hệ chuyên gia cho phép khớp các sự kiện trong vùng nhớ làm việc với các tri thức về lĩnh vực trong cơ sở tri thức, để rút ra các kết luận về vấn đề đang giải quyết.

#### 3.3.4 Bộ tiếp nhận tri thức

Tiếp nhận tri thức từ các chuyên gia, kỹ sư tri thức, những người có kiến thức chuyên sâu về một lĩnh vực nào đó. Tri thức được tiếp nhận này sẽ được lưu trữ lại trong bộ nhớ.

#### 3.3.5 Cơ sở tri thức

Lưu trữ, biểu diễn các tri thức mà hệ đảm nhận, làm cơ sở cho các hoạt động tư vấn của hệ chuyên gia này. Cơ sở tri thức bao gồm các sự kiện và các luật

Ví dụ về các luật: Môn toán giỏi 🡪 môn hóa giỏi

Ví dụ về sự kiện: Lựa chọn môn học, đăng kí tín chỉ…

#### 3.3.6 Vùng nhớ làm việc

Phần này đảm trách chứa các sự kiện của vấn đề đang xét

### 3.4 Ứng dụng hệ chuyên gia:

Hệ chuyên gia có chức năng khá quan trọng trong việc tư vấn

Hệ chuyên gia được áp dụng trong nhiều lĩnh vực khác nhau: Công nghiệp, nông nghiệp, khoa học máy tính, thương mại, khí tượng, y học…Ở đâu cần tư vấn ở đó cần xây dựng một hệ chuyên gia.

Các dạng bài toán (sự tư vấn)

* Diễn giải (Interpretation): Đưa ra mô tả tình huống các dữ liệu thu thập được
* Dự báo (Hediction): Đưa ra hậu quả của một tình huống nào đó
* Chuẩn đoán (Dianosis): Xác định các lỗi, các hỏng hóc của hệ thống dựa trên các dữ liệu quan sát được
* Gỡ rối (Debugging): Mô tả các phương pháp khắc phục khi hệ thống gặp sự cố
* Thiết kế (Design): Lựa chọn cấu hình các đối tượng nhằm thỏa mãn một số ràng buộc nào đó
* Giảng dạy (Instruction): Phần mềm dạy học có thể chuẩn đoán và sửa lỗi của học sinh trong quá trình học tập.

Mục đích chính của việc nghiên cứu đề tài này là việc áp dụng trong tư vấn học tập.

### *3.5 Biểu diễn tri thức*

#### 3.5.1 Dạng Object – Attribute – Value

Cơ chế tổ chức nhận thức của con người thường được xây dựng dựa trên các sự kiện (fact), xem như các đơn vị cơ bản nhất. Một sự kiện là một dạng tri thức khai báo. Nó cung cấp một số hiểu biết về một biến cố hay một vấn đề nào đó.

Một sự kiện có thể được dùng để xác nhận giá trị của một thuộc tính xác định của một vài đối tượng. Ví dụ, mệnh đề "quả bóng màu đỏ" xác nhận "đỏ" là giá trị thuộc tính "màu" của đối tượng "quả bóng". Kiểu sự kiện này được gọi là bộ ba Đối tượng-Thuộc tính-Giá trị (O-A-V – Object-Attribute-Value).

Nhưng khi biểu diễn cho một vấn đề có tính tương đối thì chúng ta phải thêm tác nhân chắc chắn vào bên trong luật (CF).

rule(Name, LHS, RHS).

rhs(Goal, CF)

lhs(GoalList)

av(Attribute, Value)

rule(Name, lhs([av(A1, V1), av(A2, V2), ....] ), rhs( av(Attr, Val), CF)

).

rule(5,

lhs([

av(turns\_over, yes),

av(gas\_gauge, empty)

]),

rhs(

av(problem, flooded),

80

)

).

left hand side (LHS) ==> right hand side (RHS).

#### 3.5.2 Dạng Frame

Là một cấu trúc dữ liệu chứa đựng tất cả những tri thức liên quan đến một đối tượng cụ thể nào đó. Frames có liên hệ chặt chẽ đến khái niệm hướng đối tượng (thực ra frame là nguồn gốc của lập trình hướng đối tượng). Ngược lại với các phương pháp biểu diễn tri thức đã được đề cập đến, frame "đóng gói" toàn bộ một đối tượng, tình huống hoặc cả một vấn đề phức tạp thành một thực thể duy nhất có cấu trúc. Một frame bao hàm trong nó một khối lượng tương đối lớn tri thức về một đối tượng, sự kiện, vị trí, tình huống hoặc những yếu tố khác. Do đó, frame có thể giúp ta mô tả khá chi tiết một đối tượng.

rule <rule id>:

[<N>: <condition>, .......]

==>

[<action>, ....].

Trong đó:

rule id — mã nhận dạng luật.

N — tùy chọn cho điều kiện.

condition — mẫu điều kiện.

action — hành động.

frame(name, [

slotname1 - [

facet1 val11,

facet2 val12,

...

],

slotname2 – [

facet1 val21,

facet2 val 22,

...

],

...

]

).

**Ví dụ:**

frame(man,

[

ako- [val [person]],

hair- [def short, del bald],

weight-[calc male\_weight]

]

).

frame(woman,

[

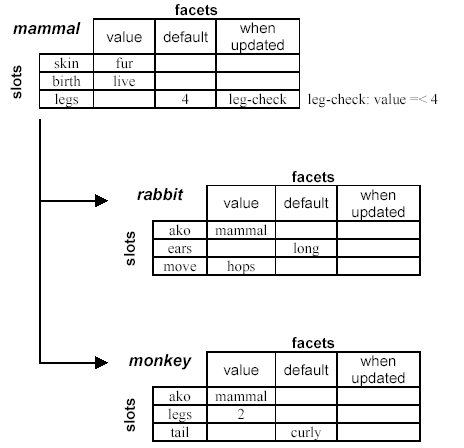
ako-[val [person]],

hair-[def long],

weight-[calc female\_weight]

]

).



class - Name with [Attr - Val, ...]

rule f11:

[table\_lamp - TL with [position-none],

end\_table - ET with [position-wall/W]]

==>

[update( table\_lamp - TL with [position-end\_table/ET] )].

match([], []).

match([Prem|Rest], [Prem/Time|InstRest]) :-

mat(Prem, Time),

match(Rest, InstRest).

mat(N:Prem, Time) :-

!,

fact(Prem, Time).

mat(Prem, Time) :-

fact(Prem, Time).

mat(Test, 0) :- test(Test).

fact(Prem, Time) :-

conv(Prem, Class, Name, ReqList),

getf(Class, Name, ReqList, Time).

conv(Class-Name with List, Class, Name,

conv(Class-Name, Class, Name, []).

#### 3.5.3 Dạng các luật dẫn

Luật là cấu trúc tri thức dùng để liên kết thông tin đã biết với các thông tin khác giúp đưa ra suy luận, kết luận từ những thông tin đã biết

Trong hệ thống dựa trên các luật, người ta thu thập các tri thức lĩnh vực trong một tập và lưu chúng trong cơ sở tri thức của hệ thống. Hệ thống dùng các luật này cùng với các thông tin trong bộ nhớ để giải bài toán. Việc xử lý các luật trong hệ thống dựa trên các luật được quản lý bằng một module gọi là *Bộ suy diễn*

Có 7 dạng luật dẫn cơ bản:

- Quan hệ:

Ví dụ: IF Bình điện hỏng THEN xe sẽ không khởi động được

- Lời khuyên

Ví dụ: IF xe không khởi động được THEN không đi bằng xe

- Hướng dẫn

Ví dụ: IF xe không khởi động được AND hệ thống nhiên liệu tốt THEN kiểm tra hệ thống điện

- Chiến lược

Ví dụ: IF xe không khởi động được THEN đầu tiên hãy kiểm tra xem hệ thống nhiên liệu sau đó kiểm tra hệ thống điện

- Diễn giải

Ví dụ: IF xe nổ AND tiếng giòn THEN động cơ hoạt động tốt

- Chuẩn đoán

Ví dụ: IF sốt cao AND hay ho AND đau họng THEN viêm họng

- Thiết kế

Ví dụ: IF là nam, body tốt THEN nên chọn quần jean áo ôm

* Mở rộng cho các luật:

Khi mệnh đề phát biểu về sự kiện, hay bản thân sự kiện có thể không chắc chắn, người ta dùng hệ số chắc chắn CF (Certain fact). Luật thiết lập quan hệ không chính xác giữa các sự kiện giả thiết và kết luận được gọi là luật không chắc chắn.

Ví dụ: IF lạm phát cao THEN hầu như chắc chắn lãi suất sẽ cao

Luật này được biểu diễn lại với giá trị CF có thể có như sau:

IF lạm phát cao THEN lãi xuất cao, CF = 0.8

Dạng siêu luật:

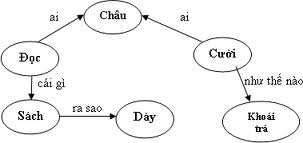
Một luật với chức năng mô tả các thức dùng các luật khác. Siêu luật sẽ đưa ra chiến lược sử dụng các luật theo lĩnh vực chuyên sâu thay vì đưa thông tin mới.

Ví dụ: IF xe không khởi dộng AND Hệ thống làm việc bình thường THEN Có thể sử dụng các luật liên quan đến hệ thống điện

#### 3.5.4 Dạng mạng ngữ nghĩa

Mạng ngữ nghĩa là một phương pháp biểu diễn tri thức dùng đồ thị trong đó đối tượng được biểu diễn bằng nút và quan hệ giữa các đối tượng được biểu diễn bằng cung

Ví dụ:



#### 3.5.5 Dạng logics

##### **3.5.5.1 Logic mệnh đề**

Là thể hiện một mệnh đề IF THEN dưới dạng điều kiện đầu ra kết quả cuối. Kết quả của logic mệnh đề là đúng hoặc sai

Các toán tử trong logic bao gồm:

* ^ : toán tử logic liên từ và
* v: toán tử logic giới từ hoặc
* ̚̚ : toán tử phủ định
* 🡪 : toán tử logic kéo theo
* ⬄ : toán tử logic tương đương nếu và chỉ nếu

Ví dụ: IF Xe không khởi động được (A) AND khoảng cách từ nhà đến chỗ làm xa (B) THEN sẽ trễ giờ làm (C)

Ta biểu diễn lại luật như sau: A ^ B 🡪 C

##### **3.5.5.2 Logic vị từ**

Logic vị từ cũng giống như logic mệnh đề, dùng các kí hiệu để biểu diễn tri thức. Những kí tự này bao gồm hằng số, vị từ, biến và hàm. Cách biểu diễn các đề xuất dùng logic vị từ cho phép ta có thể truy cập các thành phần cá thể trong một mệnh đề

**Sử dụng phương pháp hợp giải trong logic vị từ để giải các bài toán khó**

Ta có các biểu thức tương đương sau:

¬(¬P) = P.

P→Q = ¬P∨Q.

P↔Q = P→Q∧Q→P.

¬(P∧Q) = ¬P∨¬Q

¬(P∨Q) = ¬P∧¬Q.

¬(∃X) P(X) = (∀X)¬P(X).

¬(∀X) P(X) = (∃X)¬P(X).

Ví dụ:

Thực hiện ở đây

# PHẦN 4 – NGHIÊN CỨU THỰC NGHIỆM ÁP DỤNG CHƯƠNG TRÌNH CỐ VẤN HỌC TẬP ĐIỆN TỬ

## CHƯƠNG 1 – CÁC YÊU CẦU

Đưa ra các yêu cầu bài toán và phân tích, tìm hướng giải quyết

**1.1 Yêu cầu bài toán**

### 1.2 Yêu cầu phần cứng:

Phần cứng cho device

Trên máy PC: cấu hình tối thiểu Duo Core, RAM 1Gb

Trên Device: Thiế bị có cấu hình thấp hỗ trợ : JSR

### 1.3 Yêu cầu phần mềm:

Trên máy PC server:

JRE: *jre-7u1-windows-x64*

SDK : bản cài đặt *sun\_java\_me\_sdk-3\_0-win* bản mới nhất

*sun\_java\_wireless\_toolkit-2\_5\_2-windows*: dùng để chạy ứng dụng trên emulator

Trên Client device: Hỗ trợ cho J2ME

### 1.4 Môi trường thực thi (thiết bị)

J2ME trên device client và J2SE trên PC server

Yêu cầu máy client và server phải kết nối thông qua internet wifi

## CHƯƠNG 2 – THỰC THI

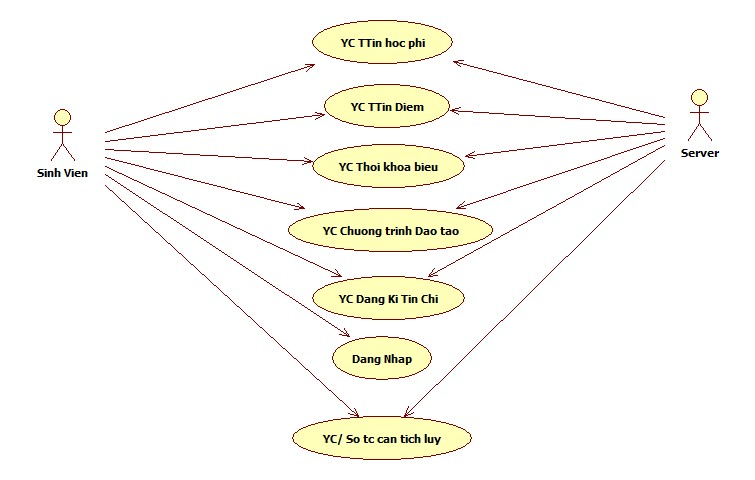
### 2.1 Phân tích yêu cầu và thiết kế các chức

Ứng dụng được thiết kế với các chức năng cơ bản sau:

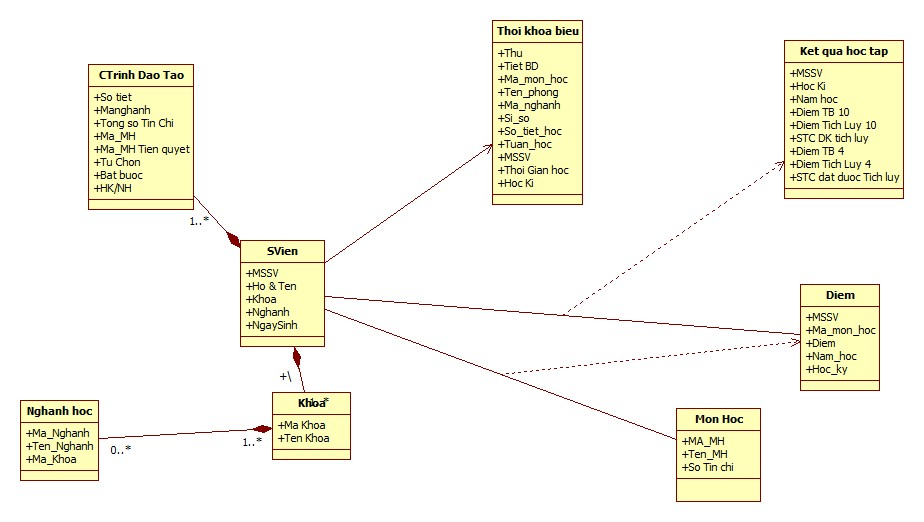
Xem điểm qua các học kì

Xem điểm tổng kết

Lựa chọn môn học



### 2.2 Thiết kế CSDL: các bảng dữ liệu



### 2.3 Phân tích các thuật toán áp dụng: hệ chuyên gia suy diễn tiến

Các tập luật được áp dụng dựa trên kết quả học tập của từng môn theo chuyên nghành

Bài toán được áp dụng: Thiết kế hệ chuyên gia suy diễn tiến Cố vấn sinh viên học tập giải quyết các vấn đề sau:

* Giải quyết các môn học mà sinh viên đã thi đậu cho qua
* Xử lý các môn học mà sinh viên đã được đặt cách cho qua
* Xử lý các môn học có các môn học tiên quyết
* Xử lý các môn học mà sinh viên được phép đăng kí trong mỗi kỳ: bao gồm đúng chuyên nghành, chưa học hoặc đã học nhưng chưa qua, môn tiên quyết (nếu có) đã qua.
* **Định nghĩa dữ liệu vào**
  + Các môn học bắt buộc
  + Các môn học tự chọn
  + Các môn học có các môn tiên quyết
  + Cac môn học mà sinh viên đã học xong
  + Các môn học mà sinh viên được phép đăng kí trong mỗi kỳ
* **Cấu trúc điều khiển dữ liệu suy diễn tiến của hệ thống:** Để xử lý số liệu vào ra của hệ thống, cơ sở luật của hệ thống được thiết lập gồm các luật là:
* Luật 1: Nếu X là môn học mà Sinh viên đã học và thi đậu thì sinh viên đã học xong môn X
* Luật 2: Nếu X là môn học mà sinh viên đã được đặt cách cho qua thì sinh viên đã học xong môn X
* Luật 3: Nếu sinh viên đã học xong môn X và Q là danh sách các môn học mà sinh viên đã học xong thì Q chứa X
* Luật 4: Nếu X có môn học tiên quyết là Y thì môn học tiên quyết của X là Y
* Luật 5: Nếu X có môn học tiên quyết là Y và Y có môn học tiêng quyết là Z thì X có môn học tiên quyết là Z
* Luật 6: Nếu môn học tiên quyết của X là Y và P là danh sách chứa các môn học tiên quyết của X thì P phải chứa Y
* Luật 7: Nếu Q là danh sách chứa các môn học mà sinh viên đã học xong với X, P là danh sách chứa các môn tiên quyết của X và P là tập con của Q thì sinh viên đã học xong tất cả với các môn học tiên quyết của X
* Luật 8: Nếu X là môn học bắt buộc, sinh viên chưa học xong với X, sinh viên đã học xong tất cả với các môn học tiên quyết của X và X là môn học cho phép sinh viên đăng kí học trong học kỳ thì cho phép sinh viên đăng kí môn học với X
* Luật 9: Nếu X là môn học tự chọn. Sinh viên chưa học xong với X, sinh viên đã học xong tất cả các môn học tiên quyết của X và X là môn học cho phép sinh viên đăng kí học trong học kì này thì cho phép sinh viên đăng kí môn X
* **Mã hóa cơ sở tri thức:**
* Các môn học bắt buộc được mã hóa dạng:
  + - Req(“tin học căn bản CNTT”)
    - Req(“Cau truc du lieu va giai thuat”)
    - Req(“Anh van chuyen nghanh CNTT 1”)
    - Req(“Giai tich 1”)
    - Req(“Giai tich 2”)
    - Req(“Anh van chuyen nghanh CNTT 2”)
    - Req(“Vat ly 1”)
    - Req(“Vat ly 2”)
    - Req(“xac suat thong ke”)
    - Req(“Toan roi rac”)
    - ….
* Các môn học tự chọn được mã hóa dưới dạng:
  + - Elec(“An toan mang”)
    - Elect(“He thong web 2”)
    - Elect(“He thong so”)
    - …
* Các môn học tiên quyết được mã hóa dưới dạng:
  + Impreq(“Anh van chuyen nghanh CNTT 2”,”Anh van chuyen nghanh CNTT1”)
  + Impreq(“Cau truc du lieu va giai thuat”,”ky thuat lap trinh”)
  + Impreq(“He thong web 2”,”He thong web 1”)
  + …
  + Faf
  + Fafa
  + Âf
  + À
  + À
* Các môn học sinh viên được phép đăng kí trong học kì này (tương ứng với thời khóa biểu được trích lọc ra rồi)
* Given\_now(“mạng máy tính”)
* Given\_now(“Internet/Intranet”)

**Mã hóa các luật**

* + Luật 1 được mã hóa bằng kí hiệu điển hình là:

IF passed(X) THEN done\_with(X)

* + Luật 2 được mã hóa bằng kí hiệu

IF waived(X) THEN done\_with(X)

* + Luật 3 được mã hóa bằng kí hiệu

IF findall(Y, done\_with(Y),X) THEN all\_done\_with(X)

* + Luật 4 được mã hóa bằng kí hiệu

IF impreq(X,Y) THEN preq(X,Y)

* + Luật 5 được mã hóa bằng kí hiệu

IF impreq(X,Y) AND preq (Y,Z) THEN preq (X,Z)

* + Luật 6 được mã hóa bằng kí hiệu

IF findall(Y, preq(X,Y), Z) then all\_preq\_for(X,Z)

* + Luật 7 được mã hóa bằng kí hiệu

IF all\_preq\_for(X,Z) AND all\_done\_with(Q) and subset(Z,Q) THEN have\_preq\_for(X)

* + Luật 8 được mã hoas bằng kí hiệu

IF req(X) AND NOT(done\_with(X)) AND given\_now(X) AND have\_preq\_for(X) THEN pos\_req\_course(X)

* + Luật 9 được mã hóa bằng kí hiệu

IF elec(X) AND NOT(done\_with(X)) AND given\_now(X) AND have\_preq\_for(X) THEN pos\_elec\_course(X)

Lập luận logic mệnh đề

Ta có các tập luật

Quy mô: Chỉ xác định cho các môn học của nghành trong khoa CNTT

Nghành CNTT có 3 chuyên nghành là CNPM, HTTT và MMT &TT. Ứng với mỗi chuyên nghành có một chương trình đào tạo riêng. Mỗi chuyên nghành đều học chung một hệ thống môn học đại cương và cơ sở nghành bắt buộc cho tất cả các sinh viên trong khoa.

Các môn cơ sở nghành thường là các môn tiên quyết để chọn nghành sau này. Ta sẽ dựa vào kết quả của các môn cơ sở nghành này trên cơ sở đó tiếp hướng dẫn sinh viên trên các môn tùy chọn (không thuộc chuyên nghành của mình)

---------//----------------

Liệt kê danh sách các môn tùy chọn của SV tương ứng với các nghành (có kèm theo độ khó của môn đó)

Nếu thỏa hết các môn bắt buộc:

Giải tích 1(good) 🡪

Giải tích 1(average) 🡪

Giải tích 1(bad) 🡪

Tin học đại cương (good) 🡪

Tin học đại cương (average) 🡪

Tin học đại cương (bad) 🡪 học cải thiện

Kiến trúc máy tính (good), tin học đại cương (good) 🡪 chọn chuyên nghành CNPM

Kiến trúc máy tính (average) 🡪 nên chọn các chuyên nghành HTTT hoặc MMT&TT

Kiến trúc máy tính (bad) 🡪 học cải thiện

Kỹ thuật lập trình (good) 🡪 nên chọn các môn nghành CNPM

Kỹ thuật lập trình (average) 🡪 nên chọn các môn nghành HTTT hoặc MMT&TT

Kiến trúc máy tính (average), Nếu muốn chọn các môn thuộc chuyên nghành CNPM 🡪 học cải thiện môn Kiến trúc máy tính (option)

Kỹ thuật lập trình (average), nếu muốn chọn các môn thuộc chuyên nghành CNPM 🡪 học cải thiệt môn kỹ thuật lập trình (option)

Toán rời rạc (good) 🡪

Toán rời rạc (average) 🡪

Cấu trúc dữ liệu và giải thuật (good) 🡪

Cấu trúc dữ liệu và giải thuật (average) 🡪

Lập trình hướng đối tượng (good) 🡪

Lập trình hướng đối tượng (average) 🡪

Hệ điều hành (good) 🡪 nên chọn các môn thuộc chuyên nghành MMT&TT

Hệ điều hành (average) 🡪 nên chọn các môn thuộc chuyên nghành HTTT hoặc CNPM

Cơ sở dữ liệu (good) 🡪 nên chọn các môn thuộc chuyên nghành HTTT

Cơ sở dữ liệu (average) 🡪 nên chọn các môn thuộc chuyên nghành MMT&TT hoặc CNPM

Lý thuyết đồ thị (good) 🡪 nên chọn các môn thuộc chuyên nghành CNPM

Lý thuyết đồ thị (average) 🡪 nên chọn các môn thuộc chuyên nghành MMT&TT hoặc HTTT

Công nghệ phần mềm (good) 🡪

Công nghệ phần mềm (average) 🡪

Mạng máy tính (good) 🡪 nên chọn các môn thuộc chuyên nghành MMT&TT

Mạng máy tính (average) 🡪 nên chọn các môn thuộc chuyên nghành CNPM hoặc HTTT

Trí tuệ nhân tạo (good) 🡪 nên chọn các môn thuộc chuyên nghành CNPM

Trí tuệ nhân tạo (average) 🡪 nên chọn các môn thuộc chuyên nghành MMT&TT hoặc HTTT

Hệ thống web 1 (good) 🡪 nên chọn các môn thuộc chuyên nghành HTTT

Hệ thống web 1 (average) 🡪 nên chọn các chuyên nghành CNPM hoặc MMT&TT

Hệ thống web 1 (averate) và muốn học các môn thuộc chuyên nghành HTTT 🡪 học cải thiện môn Hệ thống web 1

Chú thích: Good ⬄ diem\_thi.diem >= 5

Averate ⬄ diem\_thi.diem < 5

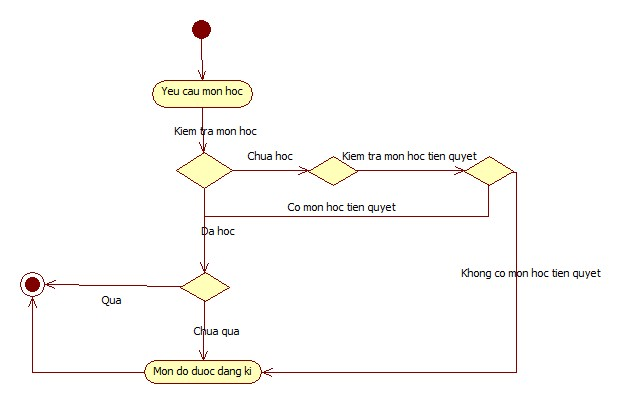
### 2.4 Thiết kế các mô hình giao tiếp giữa Client – Server

### Xây dựng từ đầu Clientnetworking và Servernetworking



*H3.1 Sự tương tác giữa Client – Server*

## Ta có một phiên cố vấn



*H3.2 Một phiên cố vấn môn học*

## CHƯƠNG 3 – DEMO ỨNG DỤNG

Phần này được thể hiện trong file cài đặt

Giao diện chính của chương trình

Giao diện đăng nhập:



Ta có 2 file: setup.jar và setup.jad

# PHẦN 5 – KẾT QUẢ THỰC NGHIỆM ĐỀ TÀI

Các kết quả thu được từ việc nghiên cứu thực nghiệm đề tài:

* Nghiên cứu và xây dựng được mô hình Client – Server
* Giúp cho Sinh viên có thể chọn lựa đúng môn học theo chuyên nghành của mình và nhanh chóng hơn việc thực hiện trên web
* abc

# PHẦN 6 – HƯỚNG PHÁT TRIỂN

Chưa chạy được trên thực tế với nhiều client yêu cầu (tương ứng nhiều thiết bị di động cùng request lên server)

Theo đó, chưa xét xây dựng được chức năng đăng kí môn học

Hệ chuyên gia còn khá đơn giản

# PHẦN 7 – TÀI LIỆU THAM KHẢO

# PHẦN 8 – PHỤ LỤC

Các từ viết tắt được dùng trong tài liệu

KDD: Hệ thống thu nạp tri thức

MTSD: Mô tơ suy diễn

CSTT: Cơ sở tri thức

NSD: Người sử dụng