ĐẠI HỌC QUỐC GIA HÀ NỘI
TRƯỜNG ĐẠI HỌC CÔNG NGHỆ

Nguyễn Trung Kiên

XÂY DỰNG HỆ THỐNG ĐỌC TIN TRÊN MOBILE

KHOÁ LUẬN TỐT NGHIỆP ĐẠI HỌC HỆ CHÍNH QUY

Ngành: Công nghệ thông tin

Cán bộ hướng dẫn: TS. Phạm Bảo Sơn

HÀ NỘI – 2010

Lời cảm ơn

Trước tiên, em xin gửi lời cảm ơn sâu sắc nhất đến thầy Phạm Bảo Sơn, người đã không quản vất vả hướng dẫn em trong suốt thời gian làm khóa luận tốt nghiệp vừa qua.

Em xin bày tỏ lời cảm ơn sâu sắc đến các thầy cô giáo trong Trường Đại Học Công Nghệ đã tận tình dạy dỗ em suốt bốn năm học qua.

Con xin cảm ơn bố, mẹ và gia đình đã luôn bên con, cho con động lực để làm việc tốt hơn.

Tôi xin cám ơn tất cả các bạn đồng nghiệp tại Công ty cổ phần công nghệ SEE đã giúp tôi rất nhiều khi nghiên cứu đề tài này.

Cảm ơn tất cả bạn bè K51CA đã luôn sát cánh cùng tôi.

Tóm tắt nội dung

Với việc bùng nổ các thông tin, tin tức trên web hiện nay nhiều vô kể và bạn không thể nào có đủ thời gian để đọc hết. Lấy một ví dụ đơn giản, hàng ngày có rất nhiều tin tức được đăng tải ở các website báo điện tử như vnexpress, dantri, vietnamenet,… Nếu phải vào từng trang để đọc thì rất mất thời gian, do đó nếu dùng trình tổng hợp tin tức để chỉ định các trang, mục nào của các báo cần được gom lại trong một giao diện duy nhất để đọc thì sẽ tiện lợi hơn rất nhiều. Hơn nữa với xu thế hiện nay ở Việt Nam, 3G bắt đầu phát triển, nhu cầu đọc tin của người dùng bằng điện thoại là rất lớn. Chính vì thế việc ra đời một hệ thống đọc tin tự động từ các nguồn báo khác nhau trên điện thoại là cần thiết

Trong khóa luận này, chúng tôi trình bày mô hình để giải quyết bài toán tổng hợp tin từ các nguồn khác nhau thông việc đọc các kênh RSS, cùng với đó là quá trình xây dựng phần mềm bằng ngôn ngữ Java (J2ME) cho các dòng điện thoại để hiển thị các tin tức này. Dựa trên framework KUIX – một framework mã nguồn mở để xây dựng các ứng dụng J2ME, chúng tôi đã mở rộng và phát triển để viết một ứng dụng có thể chạy trên hầu hết các dòng máy di động hỗ trợ Java hiện nay.

Mục lục

Danh sách các bảng

Danh sách các hình vẽ

Thuật ngữ viết tắt

CLDC	Connected Limit Device Configuration
CDC	Connected Device Configuration
GPRS	General Packet Radio Service
J2EE	Java 2 Platform, Enterprise Edition
J2ME	Java 2 Platform, Micro Edition
J2SE	Java 2 Platform, Standard Edition
JAD	Java Application Descriptor
JAR	Java Application Archive
JNI	Java Native Interface Support
JSR	Java Specification Request
KVM	Kilo Virtual Machine
m-Commerce	Mobile Commerce
MIDlet	MIDP applet
MIDP	Mobile Information Device Profile
MVC	Model-View-Controller
OTA	Over The Air
PDA	Personal Digital Assistant
RMS	Record Management System
SDK	Software Developer’s Kit
RSS	Really Simple Syndication
XML	eXensible Markup Language

Chương 1

Mở đầu

1.1.Tại sao cần các trình tổng hợp tin tự động cho các dòng máy di động

1.1.1 Nguyên nhân ra đời các hệ thống tổng hợp tin tự động

Cập nhật thông tin luôn là nhu cầu thiết yếu của con người, cầm tớ báo mới cặm cụi đọc trên vỉa hè, trong công viên, hay nhâm nhi cốc cà phê vào buổi sáng đã là thói quen của nhiều người. Sự bùng nổ của internet đã cho ra đời báo điện tử. Với việc liên tục cập nhật và đưa ra các thông tin mới và nóng nhất, đồng thời cho phép người đọc tiếp cận các thông tin đó ở bất cứ thời gian và địa điểm nào, báo điện tử đã dần trở thành kênh thông tin quan trọng đối với người dùng internet. Có nhiều đánh giá cho rằng báo điện tử là điểm sáng của cách mạng công nghệ thông tin. Ngày càng xuất hiện nhiều tờ báo điện tử truyền tải thông tin dưới mọi hình thức mà các loại báo truyền thống cung cấp. Có thể kể tên một số trang báo điện tử lớn ở Việt Nam như: vnexpress.vn, dantri.com.vn, vietnamnet.vn, 24h.com.vn, tuoitre.com.vn, thanhnien.com.vn,…

Tuy nhiên, khi mà các trang báo điện tử ra đời quá nhanh, sẽ xuất hiện tình trạng "loạn" thông tin. Quá nhiều trang web tin tức, quá nhiều thông tin trùng lặp sẽ làm cho người đọc không biết phải chọn nguồn tin nào để xem. Lấy một ví dụ đơn giản, hàng ngày có rất nhiều tin tức được đăng tải ở các website báo điện tử như vnexpress, tuoitre, thanhnien, dantri, hanoimoi,… Nếu phải vào từng trang để đọc thì rất mất thời gian, thêm vào đó nếu chỉ đọc 1, 2 mục tin trên mạng có lẽ là không đủ, chính vì nguyên nhân này, các trình đọc tin tự động, hay các trang tổng hợp tin tức (tiếng Anh gọi là News aggregator) đã ra đời. Các trang này sẽ tổng hợp nội dung các trang, các mục từ các báo điện tử khác nhau, và đưa ra một giao diện duy nhất để tiện lợi cho người đọc. Như vậy thay vì phải đi kiếm thông tin, bằng cách dùng các trang tin tổng hợp, thông tin sẽ tự động đưa xuống cho người đọc. Đối với trang tổng hợp tin tức cho tiếng Việt, có thể nói baomoi.com đi tiên phong. Với hơn 100 nguồn tin và được cập nhật liên tục, các tin trên baomoi.com khá phong phú và cập nhật. Bên cạnh đó có thể

kể đến một số site khác như vietica.com, xalo.vn, gocnhin.com, socbay.com, vsearch.vn,….

1.1.2 Các ứng dụng thương mại di động

Thương mại di động (m-Commerce) là một bước phát triển và kế thừa của thương mại điện tử (e-Commerce). với những đặc thù và thử thách riêng cho thị trường thiết bị di động. Các ứng dụng m-Commerce được chia thành nhiều loại. Một trong những loại đó là dịch vụ thông tin (information service), nhằm mục đích cung cấp thông tin cần thiết cho người dùng thiết bị di động, với thiết bị di động là một phương tiện truy xuất cực kỳ tiện lợi và hiệu quả.

Lĩnh vực lập trình ứng dụng không dây là một lĩnh vực khó tiếp cận với những ràng buộc chặt chẽ, các nhà sản xuất và nhà phát triển đã cố gắng đưa ra các tiêu chuẩn và công nghệ để có thể hỗ trợ tốt nhất cho lĩnh vực này. Ứng dụng không dây, ngoài bản thân ứng dụng, còn phải được hỗ trợ rất nhiều từ phía server và nhà cung cấp dịch vụ.

Trong tình hình hiện nay của Viêt Nam, mạng 3G đang được phát triển mạnh mẽ và rầm rộ, trong khi đó nguồn ứng dụng di động cho thị trường tiềm năng này vẫn còn đang để ngỏ, việc các ứng dung di động được phát triển không ngừng là điều không có gì để bàn cãi. Có ý kiến chuyên gia cho rằng: “Năm 2010 sẽ là năm của các ứng dụng trên di động”[9].

Việc kết hợp hai ý tưởng “phần mềm trên di động” và “hệ thống tổng hợp tin tức tự động” chính là nguyên nhân chúng tôi lựa chọn và nghiên cứu đề tài “Xây dựng hệ thống đọc tin trên mobile”

1.2. Mục đích của để tài khóa luận

Mục tiêu của đề tại là xây dựng một hệ thống hỗ trợ việc đọc báo tiếng Việt trên các mobile. Các nguồn báo được tổng hợp từ trên server, người dùng sử dụng mobile có kết nối internet (GPRS hoặc 3G) như một thiết bị client gửi yêu cầu tới server và lấy về các nguồn báo họ muốn xem.

Người dùng nếu có điện thoại hỗ trợ Java thì có thể sử dụng chương trình. Nếu điện thoại của người dùng và nhà cung cấp dịch vụ cho phép tải ứng dụng trên Internet xuống điện thoại di động thì người dùng có thể tải trực tiếp ứng dụng từ địa chỉ URL do Web server cung cấp, nếu không thì phải cài đặt chương trình bằng cách giao tiếp với máy tính bằng hồng ngoại, cáp,...

1.3. Các thách thức đối với đề tài

1.3.1. Thách thức đối với phần tổng hợp tin tức

Đối với các trình đọc tin, có hai bước để xử lý. Bước thứ nhất, hệ thống đơn giản sẽ chỉ load và hiện thị các tin theo thứ tự từ nguồn tin mà người dùng muốn đọc về dựa vào danh sách các rss của nguồn tin đó. Bước thứ hai, phức tạp hơn, đó là sau khi đã lấy được nội dung các nguồn tin về, cần phân loại các nguồn tin vào các nhóm khác nhau, xử lý loại bỏ các tin trùng lặp nội dung từ các nguồn khác nhau, đồng thời sắp xếp hiện thị các tin phù hợp với sở thích người dùng.

Ở bước thứ nhất, hệ thống sẽ phải truy cập vào các trang tin rss từ các báo điện tử, từ đó lấy ra các đường dẫn tới bài báo gốc. Sau đó truy cập vào các bài báo gốc này để lấy ra nội dung của tin. Tuy nhiên, do mỗi một báo lại có một cách tổ chức hiện thị tin tức khác nhau, với mỗi một trang lại có các mã html khác nhau, nên hệ thống cần phải có cách xử lý cho từng trang báo một.

Sau khi đã lấy hết nội dung các trang tin, hệ thống cần đưa ra cách để sắp xếp các tin tức này vào các chuyên mục khác nhau. Việc sắp xếp này là không thể phụ thuộc vào cách phần chia chuyên mục ở từng báo riêng biệt, bởi vì mỗi một tờ báo lại có một cách phân chia khác nhau.

Trên Bảng 1 là danh sách các chuyên mục từ hai tờ báo có thể coi là có số lượng độc giả lớn nhất Việt Nam (theo thống kê từ alexa.com, báo vnexpress.net đứng thứ 4, và báo dantri.com.vn đứng thứ 6 [16] trong danh sách các site có lượng truy cập nhiều nhất tại Việt Nam). Hai báo này tuy có một số chuyên mục là giống nhau, nhưng số chuyên mục còn lại lại rất khác nhau.

Một điều cần chú ý bóc tách nội dung cho các trang báo điện tử đó là, nội dung một số bài báo chứa các ảnh liên quan, hệ thống tin tức cần phải giữ lại các ảnh. Hơn nữa mục đích của việc bóc tách nội dung là để cho các máy điện thoại hiển thị nên các ảnh trong từng bài báo phải được lưu giữ để phù hợp với kích thước của tất cả các loại điện thoại khác nhau. Để giới hạn phạm vi bài toán, trong đề tài chỉ xét tới hai loại kích thước màn hình điện thoại là 240 x 320 và 172 x 220

Bảng 1. Danh sách chuyên mục từ báo vnexpress và dantri.com.vn

vnexpress.net	dantri.com.vn
+ ***thế giới*** + ***thể thao*** + ***kinh doanh*** + ***ô tô - xe máy*** + văn hóa + xã hội + vi tính + pháp luật + đời sống	+ ***thế giới*** + ***thể thao*** + ***kinh doanh*** + ***ô tô - xe máy*** + giáo dục - khuyến học + giải trí + nhịp sống trẻ + tình yêu - giới tính + sức khỏe + công nghệ

Một vấn đề cần quan tâm nữa khi tổng hợp các tin đó là làm sao phân biệt được tin nào là tin gốc, tin nào là tin đăng lại. Việc phân biệt này có các tác dụng:

- Giúp cho người đọc không cần phải đọc lại một tin nhiều lần, người đọc chỉ cần quan tâm đến tin được đưa lên đầu tiên mà thôi
- Giúp cho hệ thống không cần phải lưu lại các tin đã có rồi
- Giúp hệ thống xác định được các nguồn tin gốc, và các nguồn tin sao lưu lại. Từ đó sẽ có cách ứng xử riêng với từng nguồn tin một. Ví dụ: sẽ tập trung lấy từ các nguồn tin gốc, các nguồn tin lặp thì chỉ lấy các chuyên mục ít bị lặp hơn.

1.3.2. Thách thức đối với ứng dụng xây dựng trên mobile

Sau khi các tin tức đã được xử lý xong, các tin này được một phần mềm trên di động trình bày và hiện thị. Các tin được phân loại theo các chuyên mục khác nhau, và sắp xếp theo thời gian. Các tin có nội dung trùng lặp sẽ được nhóm lại với nhau, và chỉ hiện thị ra tin gốc.

Vấn đề đầu tiên cần quan tâm đối với một ứng dụng trên di động, đó là giao diện của tương tác người sử dụng. Màn hình của các điện thoại di động thường là nhỏ, do đó việc hiện thị các tin tức trên ứng dụng cần đảm bảo rõ ràng, dễ đọc và dễ thao tác cho người dùng. Trong đề tài của mình, chúng tôi sử dụng giao diện giống như giao diện trong phần mềm iMedia (do Công ty Naiscorp và VTC hợp tác xây dựng)[15].

Thêm vào đó, điện thoại di động cũng được chia làm hai loại: hỗ trợ màn hình cảm ứng và không hỗ trợ cảm ứng. Các loại điện thoại không hỗ trợ màn hình cảm ứng thì giá rẻ và phổ biến hơn. Đặc điểm của các loại điện thoại này là ngoài 4 phím điều hướng, thì còn có bàn phím để tương tác trong khi phần lớn các loại điện thoại cảm ứng thì thường không có bàn phím. Một ứng dụng muốn sử dụng được trên nhiều dòng điện thoại khác nhau, thì cần phải hỗ trợ cả các máy có cảm ứng và không có cảm ứng

Khó khăn cuối cùng, là làm sao để hiện thị được tiếng Việt trên các dòng điện thoại khác nhau. Một điểm cần chú ý đó là không phải điện thoại nào cũng hỗ trợ hiện thị tiếng Việt. Ví dụ là: hầu hết các điện thoại Nokia thì đều có sẵn font tiếng Việt, nhưng các điện thoại dòng BlackBerry thì phần lớn không hỗ trợ.

1.4. Các kết quả thu được:

Với những mục tiêu và khó khăn thách thức đã được đưa ra ở trên, nội dung khóa luận sẽ tập trung giải quyết các vấn đề chính sau

- Xây dựng hệ thống crawl tự động cập nhật tin tức liên tục từ các nguồn báo tiếng Việt khác nhau
- Các tin tức sau khi được thu thập về sẽ trải qua hai bước làm mịn đó là phân loại vào các chuyên mục và so sánh, phát hiện ra các tin có cùng nội dung với nó để từ đó xác định xem tin nào là tin gốc, tin nào là tin đưa lại
- Xây dựng phần mềm trên điện thoại di động, hỗ trợ cả các dòng máy có màn hình cảm ứng và không cảm ứng với giao diện đơn giản, phù hợp giúp đọc các tin đã được thu thập về
- Chúng tôi cũng đưa ra giải pháp để giải quyết trọn vẹn bài toán hiện thị tiếng Việt trên điện thoại di động với hầu hết các dòng máy phổ biến trên thị trường Việt Nam như Nokia, Motorola, SamSung,…

1.5. Tóm lược nội dung các chương còn lại

Phần còn lại của khóa luận chia làm 5 chương:

- Chương 2: Chúng tôi giới thiệu một cách tổng quan về J2ME – công nghệ của Sun để xây dựng các ứng dụng trên điện thoại di động, đồng thời giới thiệu về framework KUIX dùng để xây dựng giao diện cho các ứng dụng J2ME

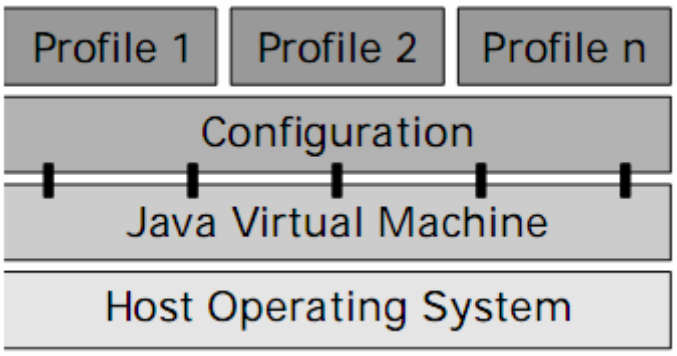
- Chương 3: Chúng tôi giới thiệu mô hình đề xuất cho hệ thống đọc tin tự động và các công nghệ, các ngôn ngữ lập trình liên quan tới đề tài
- Chương 4: Chúng tôi giới thiệu về module Crawl tin tức và phát hiện tin tức trùng lặp được xây dựng trong hệ thống.
- Chương 5: Chúng tôi giới thiệu chi tiết về ứng dụng mNews, cũng như giải pháp để giải quyết bài toán hiển thị tiếng Việt trên các dòng điện thoại đời thấp
- Chương 6: Chúng tôi tổng kết và đánh giá lại những mặt được và chưa được của hệ thống đọc tin trên di động, và đưa ra những hướng phát triển tiếp theo cho sản phẩm.

Chương 2

Giới thiệu về J2ME và framework KUIX

2.1. Khái quát về công nghệ J2ME

Mục tiêu của J2ME là cho phép người lập trình viết các ứng dụng độc lập với thiết bị di động, không cần quan tâm đến phần cứng thật sự. Môi trường phát triển của J2ME bao gồm một máy ảo (Java Virtual Machine), một cấu hình (Configuration) và một hay nhiều hiện trạng (Profile). Máy ảo định nghĩa các giao dịch giữa cấu hình và hoạt động của hệ điều hành. Các hiện trạng định nghĩa giao diện giữa một ứng dụng và môi trường J2ME. Hình 1 chỉ ra cách các tầng được tổ chức với nhau.



Hình 1. Các tầng của J2ME[7]

Từ dưới lên trên:

Tầng máy ảo Java (Java Virtual Machine)

Tầng máy ảo Java bao gồm KVM (Kilo Virtual Machine) là bộ biên dịch mã bytecode. KVM có nhiệm vụ chuyển mã của chương trình Java sau khi đã được biên dịch thành mã bytecode, thành ngôn ngữ máy để chạy trên thiết bị di động. Các chương trình Java khi cài đặt trên thiết bị di động chính là các mã bytecode. Nhờ có tầng máy áo cung cấp một sự chuẩn hóa cho các thiết bị di động mà ứng dụng J2ME có thể hoạt động trên bất kỳ thiết bị di động nào có J2ME.

Tầng cấu hình (Configuration Layer)

Tầng cấu hình của CLDC bao gồm một tập các API bậc thấp định nghĩa các thuộc tính chạy của một môi trường J2ME xác định. Cụ thể hơn, tầng cấu hình chịu trách nhiệm định nghĩa: các lớp Java cơ bản, các đặc trưng của ngôn ngữ Java, các đặc

trưng của máy ảo. Tầng cấu hình làm tăng khả năng khả chuyển của các ứng dụng J2ME trên các thiết bị di động.

Lập trình viên có thể sử dụng các lớp và phương thức của các API trên tầng cấu hình này tuy nhiên tập các API hữu dụng hơn được chứa trong tầng hiện trạng (profile layer).

Tầng hiện trạng (Profile Layer)

Tầng hiện trạng hay MIDP (Hiện trạng thiết bị thông tin di động-Mobile Information Device Profile) cung cấp tập các API hữu dụng hơn cho lập trình viên. Tầng cấu hình và tầng hiện trạng được phân tách trong kiến trúc của J2ME để phục vụ cho mục đích khả chuyển và hỗ trợ một lượng lớn các thiết bị với các khả năng khác nhau.

Mục đích của hiện trạng là xây dựng trên lớp cấu hình và cung cấp nhiều thư viện ứng dụng hơn. MIDP định nghĩa các API riêng biệt cho thiết bị di động.Ví dụ: tầng cấu hình bao gồm các đặc trưng cốt lõi của Java như: String, System, Thread và Object cũng như các luồng I/O, các kết nối mạng. Trong khi đó tầng hiện trạng quan tâm tới các thuộc tính của thiết bị như giao diện người dùng, cơ chế xử lý sự kiện, cơ chế lưu giữ dữ liệu.

2.1.1.Chi tiết về tầng cấu hình

Các cấu hình được định nghĩa bên trong kiến trúc J2ME bởi một tổ chức các chuyên gia gọi là Java Community Process (JCP). Chi tiết các cấu hình được tạo ra bởi sự hợp tác giữa JCP và rất nhiều các đối tác công nghiệp khác.

Hiện tại J2ME định nghĩa hai cấu hình:

- Cấu hình cho các thiết bị giới hạn (Connected Limited Device Configuration – CLDC) dùng cho các dòng máy điện thoại cấu hình thấp
- Cấu hình cho các thiết bị kết nối (Connected Device Configuration – CDC) dùng cho các dòng máy thông minh, đời cao giống như các smartphone, các PDAs,….

Các cấu hình định nghĩa sự "hợp đồng" giữa một hiện trạng (profile) và tấng máy ảo Java. Cả CDC và CLDC đều có máy ảo riêng của chúng. CDC sử dụng C-Virtual Machine (CVM) trong khi CLDC sử dụng Kilo Virtual Machine (KVM). CDC là một

mức cao hơn của CLDC. Phần lớn các dòng điện thoại hỗ trợ Java hiện nay đều sử dụng CLDC.

2.1.1.1. CLDC – Connected Limited Device Configuration

Phạm vi: Định nghĩa các thư viện tối thiểu và các API.

Định nghĩa:

- Tương thích ngôn ngữ JVM
- Các thư viện lỗi
- I/O
- Mạng
- Bảo mật
- Quốc tế hóa

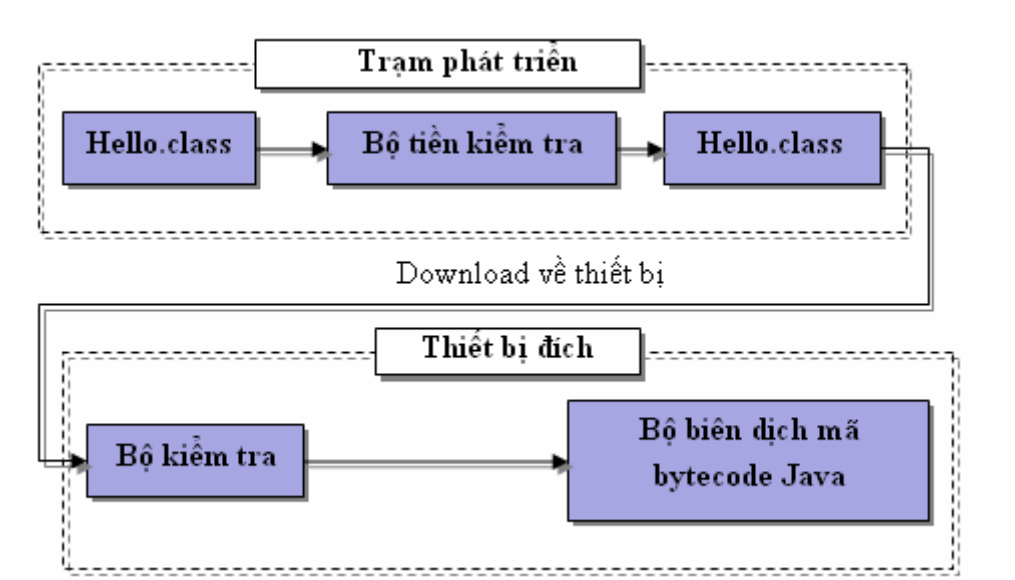
Không định nghĩa:

- Chu kỳ sống ứng dụng
- Giao diện người dùng
- Quản lý sự kiện
- Giao diện ứng dụng và người dùng

Các lớp lõi Java cơ bản, input/output, mạng, và bảo mật được định nghĩa trong CLDC. Các API hữu dụng hơn như giao diện người dùng và quản lý sự kiện được dành cho hiện trạng MIDP.

CLDC định nghĩa một mô hình an toàn, bảo mật được thiết kế để bảo vệ thiết bị di động, KVM, và các ứng dụng khác khỏi các mã phá hoại. Hai bộ phận được định nghĩa bởi CLDC này là bộ tiền kiểm tra và mô hình sandbox.

Hình 2 biểu diễn cách mà bộ tiền kiểm tra và bộ kiểm tra làm việc với nhau để kiểm tra mã chương trình Java trước khi chuyển nó cho KVM.

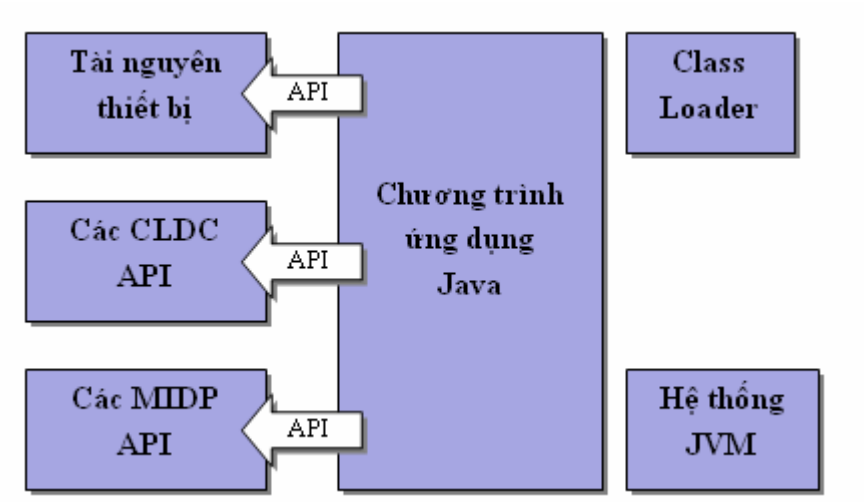


Hình 2. Bộ tiền kiểm tra

Như đã đề cập trước đây, các tập tin lớp được gán nhãn bằng một thuộc tính trên máy trạm của nhà phát triển. Thuộc tính này sau đó được kiểm tra bởi bộ tiền kiểm tra trước khi mã chương trình được giao cho KVM hay bộ biên dịch mã bytecode.

Một bộ phận khác của bảo mật trong CLDC là mô hình sandbox.

Hình biểu diễn khái niệm mô hình sandbox:



Hình 3. Mô hình Sandbox

Hình 3 cho thấy ứng dụng J2ME đặt trong một sandbox có nghĩa là nó bị giới hạn truy xuất đến tài nguyên của thiết bị và không được truy xuất đến Máy ảo Java hay bộ nạp chương trình. Ứng dụng được truy xuất đến các API của CLDC và MIDP. Ứng dụng được truy xuất tài nguyên của thiết bị di động (các cổng, âm thanh, bộ rung, các

báo hiệu,…) chỉ khi nhà sản xuất điện thoại di động cung cấp các API tương ứng. Tuy nhiên, các API này không phải là một phần của J2ME[7].

2.1.2. MIDP (Mobile Information Device Profile)

Tầng J2ME cao nhất là tầng hiện trạng và mục đích của nó là định nghĩa các API cho các thiết bị di động. Một thiết bị di động có thể hỗ trợ nhiều hiện trạng. Một hiện trạng có thể áp đặt thêm các giới hạn trên các loại thiết bị di động (như nhiều bộ nhớ hơn hay độ phân giải màn hình cao hơn). Hiện trạng là tập các API hữu dụng hơn cho các ứng dụng cụ thể. Lập trình viên có thể viết một ứng dụng cho một hiện trạng cụ thể và không cần quan tâm đến nó chạy trên thiết bị nào.

Hiện tại hiện trạng được công bố là MIDP (Mobile Information Profile) với đặc tả JSR - 37. Có 22 công ty là thành viên của nhóm chuyên gia tạo ra chuẩn MIDP.

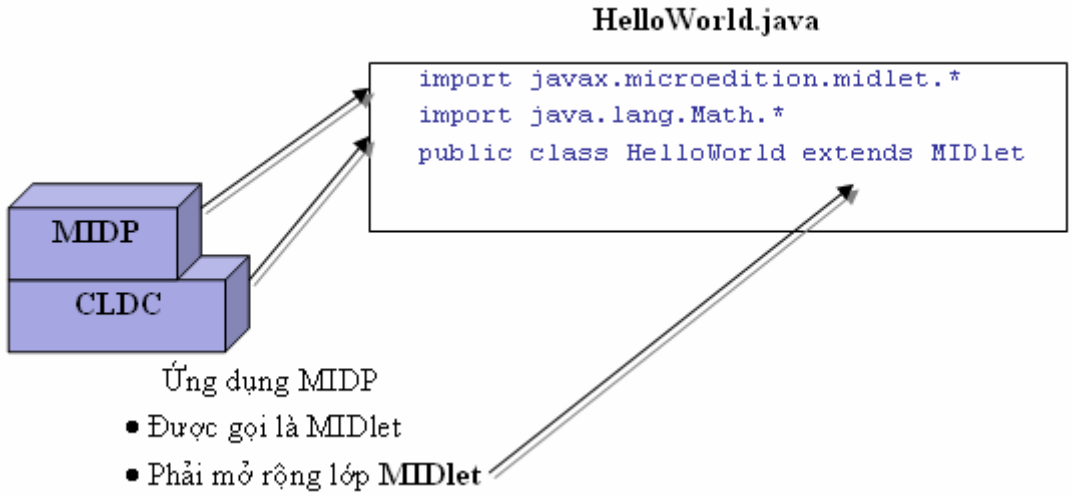
MIDP cung cấp các API cho phép thay đổi trạng thái chu kỳ sống ứng dụng, đồ họa (mức cao và mức thấp), tuyến đoạn, timer, lưu trữ bền vững (persistent storage), và mạng.

Nó không định nghĩa cách mà ứng dụng được nạp trong thiết bị di động. Đó là trách nhiệm của nhà sản xuất. Nó cũng không định nghĩa bất kỳ loại mô hình bảo mật end-to-end nào, vốn cần thiết cho ứng dụng kinh doanh nhận số thẻ tín dụng của người dùng. Nó cũng không bắt buộc nhà sản xuất cách mà lớp MIDP được thực hiện.

2.2.MIDlet

Các ứng dụng J2ME được gọi là MIDlet (Mobile Information Device applet).

Hình 4 đưa ra các thông tin cơ bản nhất để có thể tạo ra được một Midlet



Hình 4. Tổng quan về Midlet

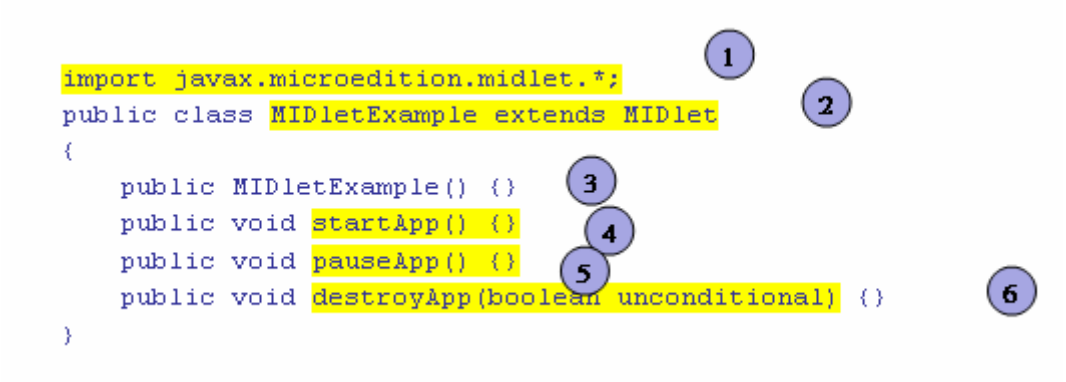
Thông báo import dùng để truy xuất các lớp của CLDC và MIDP.

Lớp chính của ứng dụng được định nghĩa là lớp kế thừa lớp MIDlet của MIDP. Có thể chỉ có một lớp trong ứng dụng kế thừa lớp này. Lớp MIDlet được trình quản lý ứng dụng trên điện thoại di động dùng để khởi động, dừng, và tạm dừng MIDlet (ví dụ, trong trường hợp có cuộc gọi đến).

2.2.1. Bộ khung MIDlet (MIDlet Skeleton)

Một MIDlet là một lớp Java kế thừa (extend) của lớp trừu tượng java.microedition.midlet.MIDlet và thực thi (implement) các phương thức startApp(), pauseApp(), và destroyApp().

Hình 5 biểu diễn bộ khung yêu cầu tối thiểu cho một ứng dụng MIDlet



Hình 5. Bộ khung MIDlet

- **Phát biểu import:** Các phát biểu import được dùng để include các lớp cần thiết từ các thư viện CLDC và MIDP.
- **Phần chính của MIDlet:** MIDlet được định nghĩa như một lớp kế thừa lớp MIDlet. Trong ví dụ này MIDletExample là bắt đầu của ứng dụng.
- **Hàm tạo (Constructor):** Hàm tạo chỉ được thực thi một lần khi MIDlet được khởi tạo lần đầu tiên. Hàm tạo sẽ không được gọi lại trừ phi MIDlet thoát và sau đó khởi động lại.
- **startApp():** Phương thức startApp() được gọi bởi bộ quản lý ứng dụng khi MIDlet được khởi tạo, và mỗi khi MIDlet trở về từ trạng thái tạm dừng. Nói chung, các biến toàn cục sẽ được khởi tạo lại trừ hàm tạo bởi vì

các biến đã được giải phóng trong hàm pauseApp(). Nếu không thì chúng sẽ không được khởi tạo lại bởi ứng dụng.

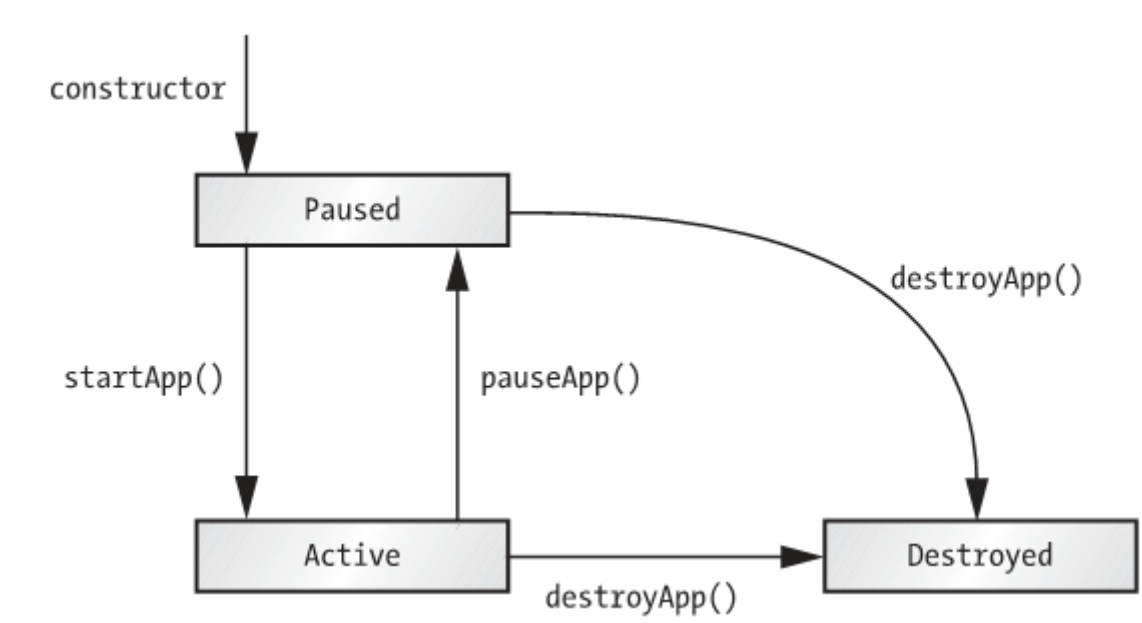
- **pauseApp()**: Phương thức pauseApp() được gọi bởi bộ quản lý ứng dụng mỗi khi ứng dụng cần được tạm dừng (ví dụ, trong trường hợp có cuộc gọi hoặc tin nhắn đến). Cách thích hợp để sử dụng pauseApp() là giải phóng tài nguyên và các biến để dành cho các chức năng khác trong điện thoại trong khi MIDlet được tạm dừng. Cần chú ý rằng khi nhận cuộc gọi đến hệ điều hành trên điện thoại di động có thể dừng KVM thay vì dừng MIDlet. Việc này không được đề cập trong MIDP mà đó là do nhà sản xuất quyết định sẽ chọn cách nào.
- **destroyApp():** Phương thức destroyApp() được gọi khi thoát MIDlet. (ví dụ khi nhấn nút exit trong ứng dụng). Nó chỉ đơn thuần là thoát MIDlet. Nó không thật sự xóa ứng dụng khỏi điện thoại di động. Phương thức destroyApp() chỉ nhận một tham số Boolean. Nếu tham số này là true, MIDlet được tắt vô điều kiện. Nếu tham số là false, MIDlet có thêm tùy chọn từ chối thoát bằng cách ném ra một ngoại lệ MIDletStateChangeException.

Tóm tắt các trạng thái khác nhau của MIDlet:

- **Tạo (Created)**: Hàm tạo MIDletExample() được gọi một một lần
- **Hoạt động (Active)**: Phương thức startApp() được gọi khi chương trình bắt đầu hay sau khi tạm dừng
- **Tạm dừng (Paused)**: Phương thức pauseApp() được gọi. Có thể nhận các sự kiện timer.
- **Hủy (Destroyed)**: Phương thức destroy() được gọi.

2.2.2. Chu kỳ sống của MIDlet

Hình 6 mô tả các chu kỳ sống của một MIDlet



Hình 6. Chu kỳ sống của MIDlet[3]

Khi người dùng yêu cầu khởi động ứng dụng MIDlet, bộ quản lý ứng dụng sẽ thực thi MIDlet (thông qua lớp MIDlet). Khi ứng dụng thực thi, nó sẽ được xem là đang ở trạng thái tạm dừng. Bộ quản lý ứng dụng gọi hàm tạo và hàm startApp(). Hàm startApp() có thể được gọi nhiều lần trong suốt chu kỳ sống của ứng dụng. Hàm destroyApp() chỉ có thể gọi từ trạng thái hoạt động hay tạm dừng. Lập trình viên cũng có thể điều khiển trạng thái của MIDlet.

Các phương thức dùng để điều khiển các trạng thái của MIDlet:

- resumeRequest(): Yêu cầu vào chế độ hoạt động. Ví dụ: Khi MIDlet tạm dừng, và một sự kiện timer xuất hiện.
- notifyPaused(): Cho biết MIDlet tự nguyện chuyển sang trạng thái tạm dừng. Ví dụ: Khi đợi một sự kiện timer.
- notifyDestroyed(): Sẵn sàng để hủy. Ví dụ: Xử lý nút nhấn Exit

Lập trình viên có thể yêu cầu tạm dừng MIDlet trong khi đợi một sự kiện timer hết hạn. Trong trường hợp này, phương thức notifyPaused() sẽ được dùng để yêu cầu bộ quản lý ứng dụng chuyển ứng dụng sang trạng thái tạm dừng.

2.2.3. Tập tin JAR

Các lớp đã biên dịch của ứng dụng MIDlet được đóng gói trong một tập tin JAR (Java Archive File). Đây chính là tập tin JAR được download xuống điện thoại di động.

Tập tin JAR chứa tất cả các tập tin class từ một hay nhiều MIDlet, cũng như các tài nguyên cần thiết. Hiện tại, MIDP chỉ hỗ trợ định dạng hình .png (Portable Network Graphics). Tập tin JAR cũng chứa tập tin kê khai (manifest file) mô tả nội dung của MIDlet cho bộ quản lý ứng dụng. Nó cũng phải chứa các tập tin dữ liệu mà MIDlet cần. Tập tin JAR là toàn bộ ứng dụng MIDlet. MIDlet có thể load và triệu gọi các phương thức từ bất kỳ lớp nào trong tập tin JAR, trong MIDP, hay CLDC. Nó không thể truy xuất các lớp không phải là bộ phận của tập tin JAR hay vùng dùng chung của thiết bị di động.

2.3. Đồ họa (Graphic)

2.3.1. Đồ họa mức thấp (low level) và mức cao (high level)

Các lớp MIDP cung cấp hai mức đồ họa: đồ họa mức thấp và đồ họa mức cao. Đồ họa mức cao dùng cho văn bản hay form. Đồ họa mức thấp dùng cho các ứng dụng trò chơi yêu phải vẽ lên màn hình.

Cả hai lớp đồ họa mức thấp và mức cao đều là lớp con của lớp Displayble. Trong MIDP, chỉ có thể có một lớp displayable trên màn hình tại một thời điểm. Có thể định nghĩa nhiều màn hình nhưng một lần chỉ hiển thị được một màn hình.

2.3.1.1. Đồ họa mức cao (High Level Graphics) (Lớp Screen)

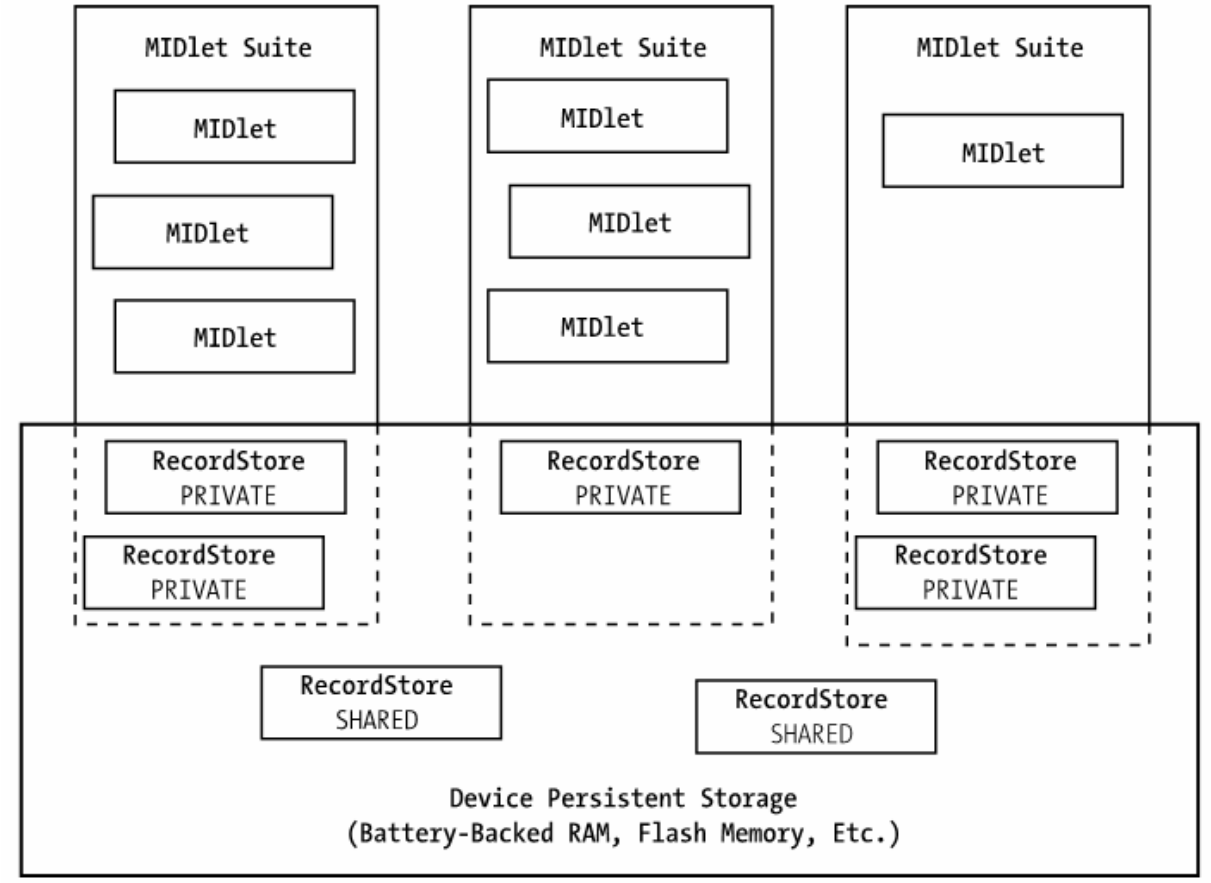
Đồ họa mức cao là lớp con của lớp Screen. Nó cung cấp các thành phần như text box, form, list, và alert. Ta ít điều khiển sắp xếp các thành phần trên màn hình. Việc sắp xếp thật sự phụ thuộc vào nhà sản xuất.

2.3.1.2. Đồ họa mức thấp (Lớp Canvas)

Đồ họa mức thấp là lớp con của lớp Canvas. Lớp này cung cấp các phương thức đồ họa cho phép vẽ lên màn hình hay vào một bộ đệm hình cùng với các phương thức xử lý sự kiện bàn phím. Lớp này dùng cho các ứng dụng trò chơi cần điều khiển nhiều về màn hình.

2.4. Lưu trữ bản ghi (Record Store)

Lưu trữ bản ghi cho phép lưu dữ liệu khi ứng dụng thoát, khởi động lại và khi thiết bị di động tắt hay thay pin. Dữ liệu lưu trữ bản ghi sẽ tồn tại trên thiết bị di động cho đến khi ứng dụng thật sự được xóa khỏi thiết bị di động. Khi một MIDlet bị xóa, tất cả các lưu trữ bản ghi của nó cũng bị xóa.



Hình 7. Lưu trữ bản ghi

Như trong Hình 7, các MIDlet có thể có nhiều hơn một tập lưu trữ bản ghi, chúng chỉ có thể truy xuất dữ liệu lưu trữ bản ghi chứa trong bộ MIDlet của chúng. Do đó, MIDlet 1 và MIDlet 2 có thể truy xuất dữ liệu trong Record Store 1 và Record Store 2 nhưng chúng không thể truy xuất dữ liệu trong Record Store3. Ngược lại, MIDlet 3 chỉ có thể truy xuất dữ liệu trong Record Store 3 và không thể truy xuất dữ liệu dữ liệu trong Record Store 1 và Record Store 2. Tên của các lưu trữ bản ghi phải là duy nhất trong một bộ MIDlet nhưng các bộ khác nhau có thể dùng trùng tên.

Các bản ghi trong một lưu trữ bản ghi được sắp xếp thành các mảng byte. Các mảng byte không có cùng chiều dài và mỗi mảng byte được gán một số ID bản ghi.

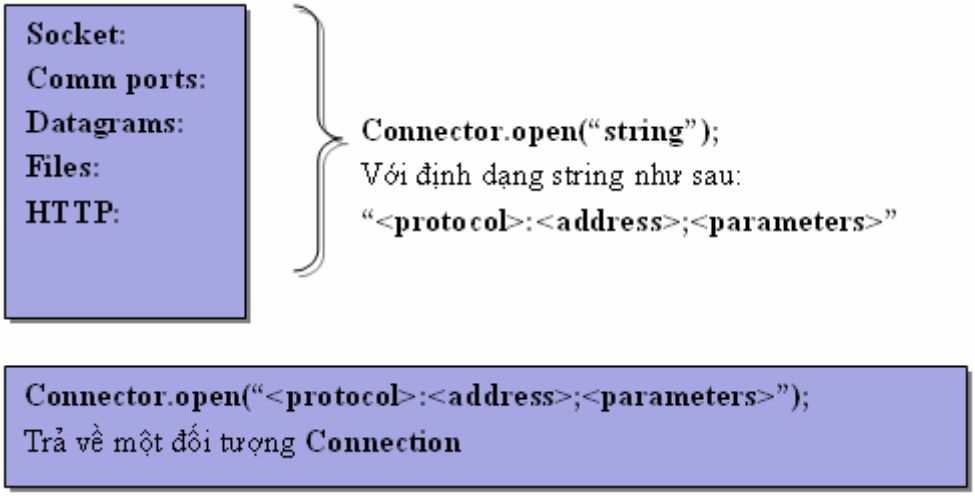
Các bản ghi được định danh bằng một số ID bản ghi (record ID) duy nhất. Các số ID bản ghi được gán theo thứ tự bắt đầu từ 1. Các số sẽ không được dùng lại khi một bản ghi bị xóa do đó sẽ tồn tại các khoảng trống trong các ID bản ghi. Đặc tả MIDP không định nghĩa chuyện gì xảy ra khi đạt đến số ID bản ghi tối đa, điều này phụ thuộc vào ứng dụng.

2.5. Lập trình mạng

2.5.1.Khung mạng CLDC tổng quát

Mạng cho phép client di động gởi và nhận dữ liệu đến server. Nó cho phép thiết bị di động sử dụng các ứng dụnsg như tìm kiếm cơ sở dữ liệu, trò chơi trực tuyến… Trong J2ME, mạng được chia làm hai phần. Phần đầu tiên là khung được cung cấp bởi CLDC và phần hai là các giao thức thật sự được định nghĩa trong các hiện trạng.

CLDC cung cấp một khung tổng quát để thiết lập kết nối mạng. Ý tưởng là nó là đưa ra một khung mà các hiện trạng khác nhau sẽ sử dụng. Khung CLDC không định nghĩa giao thức thật sự. Các giao thức sẽ được định nghĩa trong các hiện trạng.



Hình 8. Khung mạng CLDC tổng quát

Kết nối mạng được xây dựng bằng phương thức open() của lớp Connector trong CLDC. Phương thức open() nhận một tham số đầu vào là chuỗi. Chuỗi này dùng để xác định giao thức. Định dạng của chuỗi là:

protocol:address;parameters

CLDC chỉ xác định tham số là một chuỗi nhưng nó không định nghĩa bất kỳ giao thức thật sự nào. Các hiện trạng có thể định nghĩa các giao thức kết nối như HTTP,

socket, cổng truyền thông, datagram,… Phương thức open() trả về một đối tượng Connector. Đối tượng này sau đó có thể đóng vai trò là một giao thức xác định được định nghĩa trong hiện trạng.

MIDP hỗ trợ giao thức HTTP:

HTTP: Connector.open("http://www.sonyericsson.com");

Trả về một đối tượng Connection

Tất cả các kết nối mạng đều có cùng định dạng, không quan tâm đến giao thức thật sự. Nó chỉ khác nhau ở chuỗi chuyển cho phương thức open(). Phương thức open() sẽ trả về một đối tượng Connection đóng vai trò là lớp giao thức (ví dụ. HttpConnection) để có thể sử dụng các phương thức cho giao thức đó. J2ME chỉ định nghĩa một kết nối là kết nối HTTP trong MIDP.

2.5.3. Kết nối HTTP

Hiện trạng MIDP hỗ trợ kết nối HTTP phiên bản 1.1 thông qua giao diện HttpConnection. Hỗ trợ GET, POST, HEAD của HTTP. Yêu cầu GET (GET request) được dùng để lấy dữ liệu từ server và đây là phương thức mặc định. Yêu cầu POST dùng để gởi dữ liệu đến server. Yêu cầu HEAD tương tự như GET nhưng không có dữ liệu trả về từ server. Nó có thể dùng để kiểm tra tính hợp lệ của một địa chỉ URL.

Phương thức open() của lớp Connector dùng để mở kết nối. Phương thức open() trả về một đối tượng Connection sau đó có thể đóng vai trò là một HttpConnection cho phép dùng tất cả các phương thức của HttpConnection.

Một kết nối HTTP có thể ở một trong ba trạng thái khác nhau: Thiết lập (Setup), Kết nối (Connectd), hay Đóng (Close). Trong trạng thái Thiết lập, kết nối chưa được tạo. Phương thức setRequestMethod() và setRequestProperty() chỉ có thể được dùng trong trạng thái thiết lập. Chúng được dùng để thiết lập phương thức yêu cầu (GET, POST, HEAD) và thiết lập thuộc tính HTTP (ví dụ. User-Agent). Khi sử dụng một phương thức yêu cầu gởi dữ liệu đến hay nhận dữ liệu về từ server sẽ làm cho kết nối chuyển sang trạng thái Kết nối. Gọi phương thức close() sẽ làm cho kết nối chuyển sang trạng thái Đóng.

Lưu ý rằng gọi bất kì phương thức nào liệt kê ở trên (ví dụ. openInputStream(), getLenght()) cũng sẽ làm cho kết nối chuyển sang trạng thái Kết nối.

2.6. Giới thiệu về Framework KUIX

J2ME là một cách thích hợp để phát triển các ứng dụng trên điện thoại di động. Tuy nhiên nền tảng giao diện đồ họa trên J2ME rất yếu. Mặc dù J2ME cung cấp cho chúng ta một danh sách các lớp đồ họa mức cao. Nhưng các lớp đồ họa này lại được cài đặt phụ thuộc vào từng nhà sản xuất, thêm vào đó là các lớp này cũng chưa đáp ứng được các yêu cầu khi thiết kế các giao diện phức tạp. Chính vì thế, khi muốn xây dựng các ứng dụng với một giao diện phù hợp, chúng ta thường phải dựa vào các framework xây dựng giao diện có sẵn. KUIX là một trong số các framework như vậy.

2.6.1. KUIX là gì?

KUIX[12] là cụm từ được viết tắt cho Kalmeo User Interface eXtensions (Giao diện người sử dụng mở rộng Kalmeo). KUIX là một khung làm việc phát triển ứng dụng cho phép tạo ra các ứng dụng J2ME cấp cao. Nó cung cấp phần lớn các thành phần đồ họa (button, text fields, list, menu, …) cần thiết để tạo ra các giao diện ứng dụng ở mức cao.

KUIX là một ứng dụng mã nguồn mở. KUIX được cung cấp dưới giấy phép GPL, do đó chung ta có thể tải và sử dụng nó để tạo nên các ứng dụng một cách hoàn toàn miễn phí.

Hình 9. Một vài ứng dụng sử dụng KUIX

2.6.2. Điểm mạnh của KUIX

KUIX là một framework mạnh. Các ưu điểm của nó bao gồm:

- Tương thích với rất nhiều dòng máy. Ngay từ ban đầu, mục tiêu thiết kế KUIX là hướng tới việc hỗ trợ được các dòng máy khác nhau. Kết quả là, tới phiên bản 1.0.1, KUIX đã hỗ trở một danh sách rộng lớn các loại thiết bị khác nhau. Về cơ bản, KUIX kết hợp giữa CLDC 1.0 và MIDP 2.0
- Cung cấp mội trường phát triển ứng dụng cấp cao. KUIX chứa phần lớn các thành phần cần thiết để thiết kết các ứng dụng cấp cao. Nó sử dụng mẫu các widget (các ứng dụng nhỏ) và mô hình thừa kế để tạo nên các ứng dụng một cách đơn giản, và dễ tùy chỉnh.
- Việc phát triển các ứng dụng dùng KUIX sẽ rất nhanh và dễ dàng. Các form và các widget được tổ chức thông qua cách tiếp cận sử dụng XML, kết hợp với các file CSS, cho phép các lập trình viên xây dựng các ứng dụng rất nhanh chóng.
- Thiết kế ứng dụng rất nhẹ

2.6.2. Cơ bản về thiết kế giao diện trong KUIX

Giao diện người sử dụng trong KUIX, được phát triển dựa vào 3 tính chất chính:

- Hướng bố cục (layout oriented)
- Sử dụng các widget như các phần tử đồ họa
- Có thể được mô tả với ngôn ngữ Java hoặc cách tiếp cận sử dụng XML/stylesheet

Hướng bố cục nghĩa là các phần tử đồ họa được đặt tại những địa điểm được định nghĩa trước thông qua bố cục, điều này giúp cho các ứng dụng sẽ tự động phù với các kích thước màn hình khác nhau. Cách tiếp cận này cũng cho phép việc thiết kế giao diện người sử dụng được mô tả bằng các yêu cầu giữa các phần tử và giúp cho hệ thống đồ họa sắp xếp vị trị của chúng tại thời điểm ứng dụng chạy phụ thuộc vào khả năng của các thiết bị.

Các widget là các thành phần giao diện sử dụng có thể được sử dụng lại để xây dựng các màn hình phức tạp hơn. Widgets có thể tùy chỉnh trong mẫu giao diện sử

dụng. Một số widget đặc biệt luôn được định nghĩa và dễ dàng cho vào các ứng dụng bằng phương pháp kế thừa.

Việc mô tả giao diện có thể thực hiện bằng hai cách: XML/CSS hoặc Java. Lợi ích của việc sử dụng cách tiếp cận thứ nhất đó là:

Phân tách giữa việc phát triển ứng dụng và kỹ năng đồ họa

Quá trình xử lý nghiệp vụ, logic sẽ được tách biệt với giao diện đồ họa

Cách thiết kế sử dụng Java sẽ đạt được hiệu quả cao hơn, bởi vì nó không đòi hỏi phải quá trình dịch các file XML và các file CSS trong khi chạy ứng dụng, nhưng điều này không cho phép phân tách các kỹ năng trong một đội phát triển.

Với cách tiếp cận bằng XML/CSS, một giao diện có thể được mô tả bởi file XML và được "trang điểm" với các file CSS. Các file XML và CSS này sẽ được đưa vào chương trình thông qua các đoạn mã nguồn Java. Bất cứ điều gì chúng ta có thể làm với XML và CSS, đều có thể làm trực tiếp với mã nguồn Java, nhưng điều này làm cho mã nguồn trở nên kém linh động và khó đọc hơn

2.6.3. Worker trong KUIX

Worker là một thread chạy liên tục trong KUIX. Đây chính là thành phần quan trọng của KUIX trong việc sử lý các sự kiện. Worker chứa trong nó nhiều WorkerTask – tương ứng với các nhiệm vụ cần chạy. Khi chạy, Worker sẽ chạy lần lượt từng WorkerTask một cho tới khi WorkerTask đó trả về giá trị, sau đó sẽ chạy tiếp tới WorkerTask tiếp theo. Sau khi WorkerTask chạy xong, hoặc nếu trong lúc chạy, WorkerTask sinh ra lỗi, nó sẽ bị loại khỏi danh sách các WorkerTask của Worker.

2.6.4. KUIX Widget:

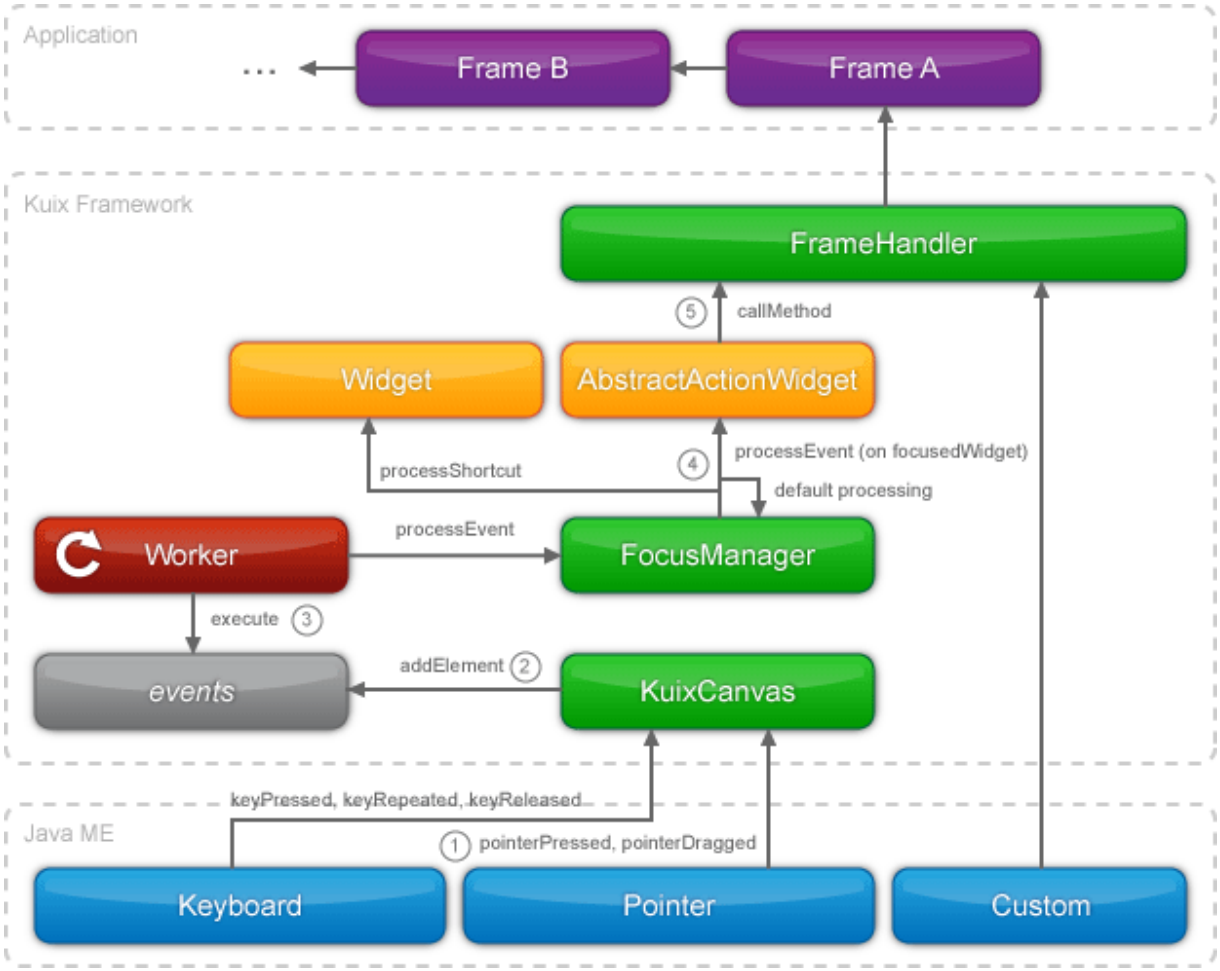
Widget là thành phần đồ họa cơ bản trong KUIX. Widget đại diện cho một vùng diện tích hiện thị trên màn hình điện thoại. Widget được tạo ra bởi việc sử dụng lớp Canvas (thành phần đồ họa mức thấp trong J2ME) để vẽ lên màn hình chi tiết giao diện của đối tượng.

Trong KUIX đã cài đặt sẵn nhiều Widget: như button, checkbox, choice, list, menu, … Các loại widget này được phân biệt với nhau bởi các thuộc tính "tag". Ví dụ: các thuộc tính tag của các Widget kể trên lần lượt là: button, checkbox, choice, list, menu,….

2.6.5. Cơ chế xử lý sự kiện trong KUIX

KUIX cung cấp một cơ chế thống nhất xử lý tất cả các sự kiện được sinh ra trong ứng dụng từ các sự kiện do người dùng sinh ra như ấn phím, chạm màn hình (đối với các máy hỗ trợ màn hình cảm ứng), tới các sự kiện như việc bật ra các popup, việc làm tươi màn hình,…

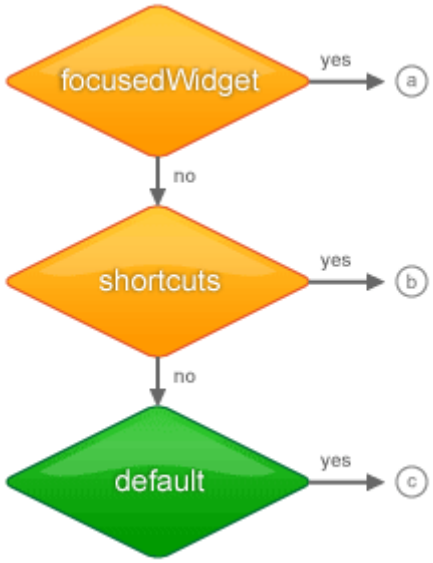
Hình 10 chỉ ra cơ chế xử lý sự kiện của KUIX



Hình 10. Cơ chế xử lý sự kiện của KUIX[12]

Với mỗi một sự kiện từ người dùng (1), J2ME sẽ gửi một thông điệp (message) tới canvas hiện tại. Trong KUIX, canvas này tự nó không xử lý sự kiện này mà đẩy sự kiện vào một ngăn xếp sự kiện (2). Ngăn xếp này được lấy ra thường xuyên bởi một tiểu trình chạy liên tục (trong KUIX gọi là worker)(3). Tiểu trình sẽ gọi đến lớp FocusManager để đưa ra các điều khiển thích hợp đối với sự kiện

Tới đây, lớp FocusManager sẽ có những xử lý tùy thuộc vào loại sự kiện là sự kiện nào. Thuật toán được mô tả trong hình 11 như sau:



Hình 11. Thuật toán xử lý của FocusManager[12]

FocusManager cố gắng nhận diện widget đang được tập trung hiện tại. Nếu nó tồn tại và nó là chủ cung cấp message, thì hàm xử lý sẽ trả về. Sự kiện được đẩy đến FrameHandler thích hợp chứa điều khiển của widget hiện tại.

Nếu không có widget được tập trung hiện tại, FocusManager sẽ so sánh mã khóa của sự kiện với danh sách các mã shortcut. Nếu mã sự kiện có trong danh sách, FocusManager sẽ nhận diện và phân phối sự kiện cho widget thích hợp.

Trong trường hợp còn lại, FocusManager sẽ áp dụng hành động mặc định là thực hiện di chuyển, tìm tới widget có thể tập trung tiếp theo.

2.7. Tổng kết chương

Trong chương này, chúng tôi đã giới thiệu một cách tổng quan về J2ME, kiến trúc, các cấu hình cho từng loại thiết bị, vòng đời của một MIDlet – đơn vị cơ bản để tạo nên một ứng dụng J2ME cũng như các API để lập trình mạng và giao tiếp với các bản ghi trong J2ME

Đối với cấu hình MIDP 2.0, mặc dù còn nhiều còn nhiều hạn chế khi lập trình các giao diện đồ họa cho ứng dụng, nhưng bằng cách sử dụng các framework hỗ trợ tạo giao diện như KUIX, chúng tôi đã làm giải quyết được điểm yếu này. Với KUIX, việc xây dựng các giao diện ứng dụng đã trở nên đơn giản hơn rất nhiều bằng cách tạo ra các file .xml và file .css tương ứng.

J2ME chính là nền tảng để chúng tôi xây dựng và phát triển ứng dụng đọc báo trên các thiết bị di động. Chi tiết về toàn bộ kiến trúc của hệ thống từ việc thu thập và

xử lý dữ liệu tới việc cung cấp dữ liệu cho thiết bị sẽ được chúng tôi trình bày ở các chương tiếp theo.

Chương 3

Kiến trúc đề xuất cho hệ thống

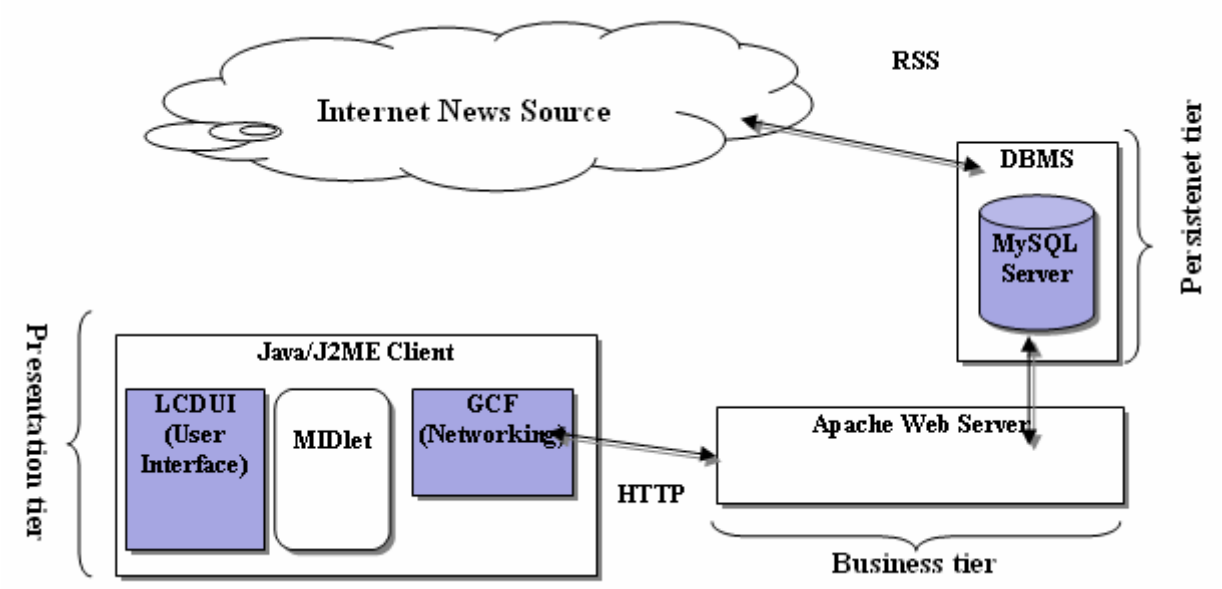
3.1.Tổng quan về hệ thống

Toàn bộ hệ thống bao gồm một ứng dụng trên mobile, có thể coi là một máy trạm (client) và một máy chủ (server) phục vụ các yêu cầu từ phía máy trạm và trả về dữ liệu cho máy trạm.

Hệ thống được phân làm 3 tầng riêng biệt. Ưu điểm của việc phân tầng đó là:

- Các tầng sẽ được tách biệt, việc thay đổi một tầng sẽ không ít ảnh hưởng đến tầng khác.
- Ngoài ra mỗi tầng có thể nằm trong một hệ thống khác với các tầng khác. Máy chủ ở tầng xử lý có thể nằm ngoài máy chủ quản trị cơ sở dữ liệu ở tầng lưu giữ. Việc này sẽ giúp triển khai từng hệ thống chuyên biệt với chức năng của nó. Đồng thời nó còn giúp cho việc tăng hiệu năng hoạt động và tính chịu tải của hệ thống sau này.

Hình 12 mô tả kiến trúc tổng quan của toàn bộ hệ thống với 3 tầng khác nhau là tầng lưu trữ (Persistant tier), tầng xử lý (Bussiness tier), tầng trình diễn (Presentation tier)



Hình 12. Kiến trúc tổng quan của hệ thống đọc tin trên mobile

3.1.1. Tầng lưu giữ (Persistant tier):

Tầng lưu giữ là một hệ quản trị cơ sở dữ liệu, để lưu giữ nội dung các tin tức đã lấy được, đồng thời cũng lưu giữ thông tin về từng tin tức (như tin tức đó thuộc báo nào, được cập nhật lên khi nào, có bao nhiêu tin đã đăng lại, …).

Hệ quản trị cơ sở dữ liệu được chọn là MySQL. MySQL là hệ quản trị cơ sở dữ liệu mã nguồn mở phổ biến nhất thế giới và được các nhà phát triển rất ưa chuộng trong quá trình phát triển ứng dụng. Vì MySQL là cơ sở dữ liệu tốc độ cao, ổn định và dễ sử dụng, có tính khả chuyển, hoạt động trên nhiều hệ điều hành cung cấp một hệ thống lớn các hàm tiện ích rất mạnh.Với tốc độ và tính bảo mật cao, MySQL rất thích hợp cho các ứng dụng có truy cập CSDL trên internet.

Cơ sở dữ liệu của hệ thống được tổng hợp từ các nguồn báo trên internet. Trên server cho chạy liên tục các bọ thu thập dữ liệu (crawler). Các bọ này có nhiệm vụ đọc các RSS lấy từ các nguồn tin tức khác nhau và lấy nội dung của từng tin tức này đưa vào cơ sở dữ liệu.

Các bọ tìm kiếm được viết bằng ngôn ngữ Python. Python là ngôn ngữ khá mạnh trong xử lý xâu, văn bản. Chẳng hạn tương tác với khối lượng lớn dữ liệu trong các file, hoặc muốn thay đổi tên, hay sắp xếp lại các file hình ảnh theo một tiêu chuẩn phức tạp. Đặc biệt Python là ngôn ngữ rất được ưa chuông khi viết các bọ tìm kiếm, bản thân Google cũng sử dụng Python để viết các bọ tìm kiếm của họ

Sau khi các bọ tìm kiếm tổng hợp tin tức từ các báo khác nhau thông qua các kênh thông tin RSS, một thuật toán sẽ được áp dụng để tìm ra các tin tức có nội dung trùng lặp nhau, thông qua đó xác định xem tin tức nào là tin gốc, tin nào là tin đăng lại. Cả tin gốc và tin đăng lại sẽ được lưu lại trong cơ sở dữ liệu, nhưng khi hiển thị ra kết quả trả về cho ứng dụng trên mobile, thì các tin tức gốc sẽ được ưu tiên hiện thị trước nhất. Các tin tức trùng nội dung thì được gộp lại thành một nhóm

3.1.2. Tầng xử lý nghiệp vụ (Business tier):

Tầng xử lý nghiệp vụ (Business tier): Là máy chủ phục vụ các yêu cầu từ máy trạm. Máy chủ này phải đồng thời tiếp nhận nhiều yêu cầu từ các máy trạm khác nhau. Có thể nói tầng xử lý là cầu nối giữa máy trạm và hệ quản trị cơ sở dữ liệu. Bất cứ khi

nào máy trạm gửi yêu cầu lên máy chủ, máy chủ sẽ tương tác với tầng lưu giữ, và trả về cho máy trạm các nội dung tương ứng.

Máy chủ được sử dụng là máy chủ web Apache – máy chủ web miễn phí và thông dụng nhất hiện nay. Tính đến năm 2009, Apache là máy chủ web đầu tiên đạt ngưỡng 100 triệu website sử dụng nó[11]. Apache chạy trên các hệ điều hành tựa Unix, Microsoft Windows, Novell Netware và các hệ điều hành khác. Apache đóng một vai trò quan trọng trong quá trình phát triển của mạng web thế giới. Mặc dù mục đích thiết kế chính của Apache không phải là để trở thành máy chủ "nhanh nhất", nhưng hiệu năng của Apache có thể so sánh với các máy chủ có "hiệu năng cao" khác.

Việc sử dụng Apache kết hợp với PHP là một xu hướng đang rất được ưa chuộng trên thế giới. PHP với vai trò là một ngôn ngữ kịch bản (script) chạy phía server sẽ giúp việc tao ra các web động hết sức đơn giản. Điều này đặc biệt đúng khi sử dụng các framework cho lập trình PHP. Cụ thể trong khóa luận này, là sử dụng framework CakePHP.

3.1.3. Tầng trình diễn (Presentation tier):

Tầng trình diễn là một ứng dụng chạy trên một điện thoại di động. Nó sẽ cung cấp giao diện cho phép người dùng lựa chọn đọc các tin theo từng chuyên mục khác nhau, đọc các tin mới nhất, đồng thời có thể tìm kiếm các trong các tin tức của các báo khác nhau. Thông qua tương tác với người dùng, ứng dụng sẽ giao tiếp với máy chủ để lấy về các dữ liệu với một định dạng xác định.

Ứng dụng trên mobile được viết bằng ngôn ngữ java, sử dụng công nghệ J2ME của SUN. J2ME là công nghệ được SUN đưa ra J2ME ra như một chuẩn đơn mà thông qua đó các nhà phát triển có thể tạo nên các phần mềm có tính khả chuyển (portable) cho các thiết bị đơn giản. Ngôn ngữ Java là sự lựa chọn đương nhiên cho lĩnh vực này, bởi vì về cơ bản nó đã hướng nhiều về tính khả chuyển. Bằng cách này, Sun đã đảm nhận bài toán lớn về tính đa dạng của thiết bị ở một mức tổng quát, do đó các nhà phát triển không phải quan tâm đến vấn đề này nữa. Với phần lớn các dụng điện thoại trên thị trường hiện nay, một ứng dụng di động J2ME sẽ chạy được trên hầu hết các dòng máy, bất kể nó sử dụng hệ điều hành nào.

Giao thức được sử dụng giữa ứng dụng trên mobile và máy chủ là giao thức HTTP. Đây là giao thức đơn giản, phổ biến và đặc biệt là được J2ME hỗ trợ trên tất cả các dòng máy.

3.2. Các ngôn ngữ lập trình sử dụng

Với kiến trúc 3 tầng như đã trình bày ở trên, việc cài đặt của các tầng là tách biệt với nhau. Chính vì thế với mỗi tầng chúng ta có thể lựa chọn các ngôn ngữ lập trình và các công nghệ phù hợp sao cho quá trình cài đặt và phát triển là đơn giản và đỡ tốn kém nhất. Cụ thể, đối với tầng lưu giữ, phụ trách việc thu thập và xử lý dữ liệu từ các nguồn báo trên internet, ngôn ngữ lập trình được sử dụng là ngôn ngữ Python. Trong khi đó tầng trình diễn là một phần mềm chạy trên các thiết bị di động của người dùng, được viết bằng J2ME. Cuối cùng tầng xử lý nghiệp vụ, là cầu nối điều khiển việc trả về dữ liệu giữa phần mềm trên di động (tầng trình diễn) và dữ liệu thu thập được tầng lưu giữ, được viết bằng ngôn ngữ PHP dựa trên framework CakePHP – một framework MVC nổi tiếng về tính đơn giản trong cách sử dụng, cũng như hiệu quả khi thực hiện.

3.2.1. Python

Python là ngôn ngữ khá mạnh trong xử lý xâu, văn bản. Chẳng hạn tương tác với khối lượng lớn dữ liệu trong các file, hoặc muốn thay đổi tên, hay sắp xếp lại các file hình ảnh theo một tiêu chuẩn phức tạp[17].

Bạn có thể viết mã để chạy trên Unix, hay Windows. Bạn có thể viết một chương trình C/C++/Java, nhưng rất mất thời gian. Python thì rất đơn giản, chạy trên mọi hệ điều hành, Windows, MacOS X, Unix, đồng thời giúp bạn nhanh chóng có kết quả trong công việc.

Rất đơn giản để sử dụng. Python mạnh hơn C trong việc kiểm tra lỗi, là một ngôn ngữ bậc cao, hỗ trợ nhiều kiểu dữ liệu, các mảng linh động và từ điển.

Python cũng cho phép chia nhỏ chương trình để thành các module để sử dụng lại ở các chương trình khác nhau. Nó cũng có nhiều module có sẵn, như xử lý file, tương tác socket, hay ngay cả bộ giao diện người dùng.

Python là một ngôn ngữ thông dịch, nghĩa là không cần biên dịch hay liên kết nào cả, chỉ cần file mã nguồn là có thể chạy chương trình. Python giúp bạn viết chương trình ngắn gọn hơn các ngôn ngữ như C/C++/Java vì các lý do sau đây:

- Kiểu dữ liệu bậc cao cho phép tối ưu các thao tác phức tạp chỉ trong một câu lệnh
- Nhóm câu lệnh được kết thúc bởi dấu lùi đầu dòng thay vì dấu mở ngoặc và đóng ngoặc.

- Không cần thiết khai báo biến.

Vì những lý dó trên, nên việc xử lý lấy dữ liệu từ web sử dụng python rất hiệu quả

3.2.2. J2ME

Thế giới của các thiết bị di động và các thiết bị "sub-PC" không có các đặc tính giống như trong lĩnh vực PC và server.

Ngoài ra, không phải mọi thiết bị trong lĩnh vực này đều cùng làm một việc. Sự khác nhau về thiết kế và mục đích giữa PDA, điện thoại, và máy nhắn tin là rất đáng kể.

Bất kể nó mang lại sự đổi mới gì cho thị trường, thì tính đa dạng của các thiết bị này là một ác mộng đối với các lập trình viên. Nếu lập trình viên muốn xây dựng một ứng dụng cho điện thoại di động, lập trình viên có phải viết mã lại, xây dựng lại, và kiểm tra lại cho mọi thiết bị hay không? Nếu lập trình viên muốn xây dựng một client có kết nối mạng, lập trình viên phải xét đến các công nghệ kết nối nào? v.v...

J2ME ra đời nhằm mục đích chính là thiết lập một chuẩn đơn mà thông qua đó các nhà phát triển có thể tạo nên các phần mềm có tính khả chuyển cho các thiết bị micro. Ngôn ngữ Java là sự lựa chọn đương nhiên cho lĩnh vực này, bởi vì về cơ bản nó đã hướng nhiều về tính khả chuyển. Bằng cách này, Sun đã đảm nhận bài toán lớn về tính đa dạng của thiết bị ở một mức tổng quát, do đó các nhà phát triển không phải quan tâm đến vấn đề này nữa. Nếu mọi nhà cung cấp PDA, điện thoại và máy nhắn tin đều thực hiện J2ME cho thiết bị của họ, thì chúng ta có khả năng viết chương trình "viết một lần, chạy mọi nơi" (write once, run anywhere) trong lĩnh vực micro, cũng giống như ta đã quen với khái niệm này ở các hệ thống máy lớn.

Chi tiết về J2ME đã được trình bày chi tiết ở chương 2

3.2.3. Cake PHP

3.2.3.1. Giới thiệu

PHP là một ngôn ngữ khá phổ biến trên thế giới. Nhưng nó lại không có một cấu trúc cụ thể trong lập trình, tùy thuộc rất nhiều vào từng người lập trình. Họ có thể tùy biến chương trình của mình theo nhiều cách khác nhau, và đôi khi là theo những cách có thể gây nguy hiểm cho chương trình của họ. Chính vì thế mà khi lập trình với PHP

nhiều người sẽ thấy khó khăn và đôi khi là phức tạp. Nhiều đoạn mã lặp lại ở nhiều nơi, hay quên kết nối tới cơ sở dữ liệu … Chính vì thế cần có một bộ khung cho PHP để giúp việc lập trình đơn giản hơn, nhanh chóng hơn và hiệu quả, an toàn hơn.

PHP hiện tại đã cho phép lập trình OOP (Object Oriented Programming) – lập trình hướng đối tượng – giống như các ngôn ngữ Java, C++. Từ đây, các nhà phát triển PHP đã dần dần tạo ra những bộ khung giúp cho PHP phát triển nhanh hơn. Cake PHP là một trong số các framework ra đời và được cộng đồng sử dụng nhiều nhất. Cake PHP đã áp dụng triệt để mô hình lập trình MVC để xây dựng nên các ứng dụng phức tạp với thời gian và chi phí thấp nhất[10].

3.2.3.2. Mô hình MVC

MVC là tên viết tắt của Model-View-Controller. Tại sao lại có mô hình này? Bình thường khi lập trình thì mọi xử lý dữ liệu, xử lý logic đều trong một file. Chẳng hạn khi kết nối tới cơ sở dữ liệu. Trong nhiều file chúng ta đều phải sử dụng tới nó, như thế mã lặp đi lặp lại rất nhiều. Nếu có thay đổi trong kết nối thì lại phải sửa ở từng file, rất mất thời gian, không hiệu quả. Trong một file vừa cập nhật dữ liệu vào cơ sở dữ liệu, vừa xử lý logic, vừa hiển thị tới người dùng. Như vậy rất khó kiểm soát mã nguồn, người đọc mã nguồn cũng rất khó hiểu.

Còn một vấn đề nữa. Một ứng dụng có nhiều người cùng phát triển. Làm thế nào để phân chia công việc cho từng người một cách cụ thể khi mà mỗi một file đều tồn tại nhiều xử lý logic, liên quan tới cơ sở dữ liệu. Chẳng hạn có người chỉ làm về giao diện, có người chỉ làm về cơ sở dữ liệu. Rõ ràng, với cách truyền thống thì việc phân chia công việc sẽ không hiệu quả.

Chính vì thế mô hình MVC ra đời, giải quyết được các vấn đề trên, đem lại một phong cách lập trình khá hiệu quả. Không chỉ ngôn ngữ PHP mà rất nhiều ngôn ngữ khác, như Java, ASP.Net … đều hỗ trợ.

Ứng dụng sử dụng MVC được chia thành ba phần riêng biệt:

- Bộ điều khiển (Controller): Chứa đựng các xử lý logic. Mỗi một controller chứa nhiều phương thức xử lý riêng biệt các yêu cầu. Nó nhận và xử lý dữ liệu từ model, đồng thời tạo ra các đối tượng sẽ được sử dụng ở view.
- Mô hình (Model): Là thể hiện dữ liệu. Nó kết nối tới cơ sở dữ liệu, xử lý mọi vấn đề về dữ liệu, như truy vấn lấy dữ liệu, hay cập nhật, hay xóa…

Không có một tương tác nào giữa model và view, tất cả tương tác với view được xử lý thông qua controller.

- Khung nhìn (View): Là một mẫu file dùng để trình bày dữ liệu tới người dùng. Các biến, mảng, hay đối tượng sử dụng trong view được khởi tạo ở trong controller. View không chứa các xử lý logic phức tạp.

Khi mới làm quen với MVC thì mất một chút thời gian, nhưng khi đã tạo được ứng dụng rồi thì chắc chắn bạn sẽ không muốn viết ứng dụng theo cách truyền thống nữa.

3.3. Tổng kết chương

Trong chương này, chúng tôi đã trình bày về kiến trúc 3 tầng của hệ thống tổng hợp và đọc tin cho điện thoại di động. Việc phân chia thành các tầng như vậy không chỉ có tác dụng giúp phân tách các chức năng hệ thống thành từng module riêng biệt mà còn giúp cho việc phát triển từng tầng không bị phụ thuộc vào nhau. Với mỗi tầng, tùy vào nhiệm vụ và đặc trưng kỹ thuật của nó, mà chúng ta sử dụng các ngôn ngữ lập trình cho phù hợp. Cụ thể là tầng lưu giữ được viết bằng ngôn ngữ python để thu thập và lưu giữ các tin bài từ các nguồn báo tiếng Việt trên internet, tầng xử lý nghiệp vụ sử dụng ngôn ngữ PHP trên nền framework KUIX chạy trên máy chủ Apache. Cả hai tầng này đều được chạy ở phía server. Riêng tầng cuối cùng, tầng trình diễn là phần mềm được viết bằng ngôn ngữ J2ME, chạy trên các máy điện thoại cầm tay của người sử dụng.

Chi tiết về hoạt động và cách cài đặt cuả tầng lưu giữ của nó sẽ được trình bày trong chương tiếp theo.

Chương 4

Module thu thập tin tức và phát hiện các tin trùng lặp

4.1. Nhiệm vụ của module thu thập tin tức và phát hiện các tin trùng lặp

Module thu thập tin tức và phát hiện các tin trùng lặp nằm ở tầng thứ nhất – tầng lưu giữ (Persistant tier) trong kiến trúc 3 tầng đã được trình bày ở chương hai.

Nhiệm vụ của module này thu thập và phát hiện các tin trùng lặp đó là liên tục đọc dữ liệu mới từ các nguồn báo tiếng Việt trên internet thông qua các kênh RSS feed. Sau đó từ các kênh RSS này, trích xuất ra đường link dẫn tới bài báo gốc rồi từ đó lấy ra nội dung chi tiết của bài báo. Sau đó nội dung của bài báo cùng các thông tin liên quan đến nó sẽ được lưu trữ vào trong cơ sở dữ liệu được quản lý bằng hệ quản trị cơ sở dữ liệu MySQL

Tất cả các quá trình này được chạy tự động và được đặt lịch để chạy 30 phút một lần. Toàn bộ module được cài đặt bằng ngôn ngữ python.

4.2. Giới thiệu về các kênh tin tức RSS

4.2.1. RSS là gì?

RSS được viết tắt cho cụm từ Really Simple Syndication - dịch vụ cung cấp thông tin cực kì đơn giản. Dành cho việc phân tán và khai thác nội dung thông tin Web từ xa (ví dụ như các tiêu đề, tin tức). Sử dụng RSS, các nhà cung cấp nội dung Web có thể dễ dàng tạo và phổ biến các nguồn dữ liệu ví dụ như các link tin tức, tiêu đề, và tóm tắt.

RSS được dùng phổ biến bởi cộng đồng weblog để chia sẻ những tiêu đề tin tức mới nhất hay toàn bộ nội dung của nó, và ngay cả các tập tin đa phương tiện đính kèm Vào giữa năm 2000, việc sử dụng RSS trở nên phổ dụng đối với hãng tin tức lớn, bao gồm Reuters, CNN, và BBC. Những nhà cung cấp tin này cho phép các website khác tổng hợp những tiêu đề tin tức "được chia sẻ" hay cung cấp các tóm tắt ngắn gọn của các bản tin chính dưới nhiều hình thức thỏa hiệp khác nhau. RSS ngày nay được dùng

cho nhiều mục đích, bao gồm tiếp thị, báo cáo lỗi (bug-reports), hay các hoạt động khác bao gồm cập nhật hay xuất bản định kì.

Ở Việt Nam hiện nay, RSS được hầu hết các trang báo điện tử ở Việt Nam sử dụng như một cách đơn giản nhất để cung cấp các thông tin mới cập nhật.

RSS có các ưu điểm:

- Cập nhật rất nhanh chóng
- Cú pháp đơn giản
- Là định dạng chuẩn chung cho tất cả các trang web

Chính vì thế để thu thập nội dung từ các trang tin tức, sử dụng RSS từ được cung cấp từ các trang tin đó là một cách làm rất hiệu quả.

4.2.1. Cấu trúc của các văn bản RSS

Các văn bản RSS có định dạng chung như sau[9]:

```
<?xml version="1.0" encoding="ISO-8859-1" ?>
<rss version="2.0">
<channel>
 <title>W3Schools Home Page</title>
 <link>http://www.w3schools.com</link>
 <description>Free web building tutorials</description>
 <item>
  <title>RSS Tutorial</title>
  <link>http://www.w3schools.com/rss</link>
  <description>New RSS tutorial on W3Schools</description>
 </item>
 </channel>
</rss>
```

Dòng đầu tiên trong văn bản – khởi tạo XML – định nghĩa phiên bản XML và kiểu mã hóa ký tự được sử dụng trong văn bản. Trong trường hợp này văn bản sử dụng chuẩn XML 1.0 và kiểu mã hóa ISO-8859 (Latin/West European)

Dòng tiếp theo là khai báo RSS để xác định, đây là một văn bản RSS (cụ thể ở đây là RSS phiên bản 2.0).

Dòng tiếp theo chứa phần tử <channel>. Phần tử này được sử dụng để miêu tả kênh thông tin RSS. Phần tử <channel> có 3 thành phần con:

- <title> - Định nghĩa tiêu đề của kênh
- <link> - Định nghĩa siêu liên kết trở tới kênh này
- <description> - Mô tả kênh

Mỗi phần tử <channel> có thể có một hoặc nhiều phần tử <item>

Mỗi phần tử <item> định nghĩa một tin tức trong bản tin RSS

Phần tử <item> cần có 3 thành phần con:

- <title> - Định nghĩa tiêu đề cho thành phần này
- <link> - Định nghĩa siêu liên kết của thành phần
- <description> - Mô tả nội dung của tin tức được đai diện bởi thành phần <item>

Hai dòng cuối cùng là các thẻ đóng <channel> và <rss>

4.2. Chi tiết hoạt động

Module crawler là các script được viết bằng ngôn ngữ python. Các script được đặt lịch chạy liên tục 30 phút một lần. Việc đặt lịch được thực hiện bằng các crontab đối với các hệ thống UNIX hoặc các schedules đối với hệ thống WINDOWS. Chi tiết hoạt động của module được miêu tả như sau:

+ Với mỗi nguồn báo khác nhau, hệ thống lấy các link rss khác nhau tương ứng với các chuyên mục của nguồn báo đó. Do việc phân chia chuyên mục của các nguồn báo khác nhau là khác nhau, nên cần có một cách phân chia thống nhất giữa các nguồn báo trong hệ thống. Để đơn giản, trong khóa luận, sử dụng một danh sách các chuyên mục chung như sau: 1. Xã hội, 2. Thế giới, 3. Kinh doanh, 4. Văn hóa, 5. Thể thao, 6.Pháp luật, 7. Đời sống, 8.Khoa học, 9. Vi tính, 10.Ô tô – xe máy, 11. Bạn đọc viết, 12. Tâm sự, 13. Cười, 14. Khác. Các chuyên mục trên các báo sẽ được ánh xạ với một trong các chuyên mục trên. Ví dụ về ánh xạ chuyên mục trên báo vnexpress với bảng chuyên mục chung

Bảng 2. Bảng ánh xạ chuyên mục của báo vnexpress

Báo vnexpress.net	**Hệ thống**
Văn hóa	Văn hóa
Thế giới	Thế giới
Xã hội	Xã hội
Cười	Cười
Kinh doanh	Kinh doanh
Vi tính	Vi tính
Thể thao	Thể thao
Pháp luật	Pháp luật
Đời sống	Đời sống
Ô tô – xe máy	Ô tô xe máy
Bạn đọc viết tâm sự	Tâm sự
Bạn đọc viết	Bạn đọc viết

+ Module đọc các link rss từ các nguồn báo, và trích xuất ra thông tin về một tin tức nhất định. Cụ thể, module sẽ lấy ra 3 thông tin chính là:

- <link>: link của tin
- <pubdate>: thời điểm tin được đưa lên mạng
- <title>: tiêu đề tin

+ Do thông tin <pubdate> được đưa các nguồn tin đưa lên với nhiều định dạng khác nhau nên cần phải chuẩn hóa lại thời gian tin được đưa lên. Ví dụ: các <pubdate> của vnexpress.vn đưa lên với định dạng: “a, d b Y H:M:S GMT” (trong đó a là tên viết tắt của ngày trong tuần, d là ngày trong tháng, b là tên viết tắt của tháng, y là năm, H là giờ, M là phút, S là giây – Ví dụ như: “Sat, 15 May 2010 14:30:28 GMT”), nên khi chuẩn hóa, cần +7 giờ nữa để thành “2010-05-15 21:30:28”. Thông tin <pubdate> này là rất quan trọng bởi vì nó sẽ quyết định tới việc tin là tin gốc hay là tin đăng lại sau này nếu có nhiều tin có cùng nội dung. Cụ thể ở đây tin gốc là tin được đăng lên đầu tiên, tức là có <pubdate> nhỏ nhất.

+ Từ các link lấy được của các tin từ các nguồn báo, module crawl sẽ trích xuất ra id tương ứng của tin đó, id này là id của tin trong nguồn báo đó chứ không phải là id trong hệ thống crawl. Ví dụ: một link từ trang vnexpress.net có dạng: http://vnexpress.net/GL/Van-hoa/San-khau-Dien-anh/2010/05/3BA1BDF4/, thì id được trích xuất ra sẽ là ***3BA1BDF4***. Việc trích xuất id của từng nguồn báo khác nhau là khác nhau. Việc trích xuất id này và lưu lại trong hệ thống nhằm mục đích để tránh phải crawl lại các tin đã crawl rồi từ nguồn báo đó. Ví dụ: 10h30 sáng ngày 10/5/2010, crawl tin từ báo vnexpress có chứa link http://vnexpress.net/GL/Van-hoa/San-khau-Dien-anh/2010/05/3BA1BDF4/, đến 11h30 cùng ngày, ta lại đọc file rss của báo vnexpress, lúc này một số tin mới đã được đưa lên, nhưng tin ở link http://vnexpress.net/GL/Van-hoa/San-khau-Dien-anh/2010/05/3BA1BDF4/ vẫn còn. Khi đó do ta đã lưu lại id ***3BA1BDF4*** nên lúc này ta không cần phải đọc lại link trên để lấy nội dung nữa mà bỏ qua luôn. Điều này sẽ giúp tiết kiệm thời gian lấy tin và tiết kiệm bộ nhớ để lưu các tin trùng lặp

+ Sau khi trích xuất ra được id và thời gian <pubdate> mà các tin được đưa lên, module crawl sẽ đọc trực tiếp vào các link của tin để lấy nội dung tin về. Đối với một số trang báo, như vnexpress ngoài trang chính của tin, còn có một trang chứa bản in của tin. Trong trang chứa bản in này, chỉ chứa nội dung của tin mà không chứa các thành phần liên quan khác của trang web ví dụ như : menu, hay các quảng cáo flash. Do vậy module crawl sẽ đọc các trang chứa bản in này để lấy nội dung tin về. Ví dụ link từ vnexpress: http://vnexpress.net/GL/Van-hoa/San-khau-Dien-anh/2010/05/3BA1BDF4/, sẽ có trang bản in là http://vnexpress.net/GL/Van-hoa/San-khau-Dien-anh/2010/05/3BA1BDF4/?q=1.

+ Do mục đích của việc lấy nội dung tin là lấy để hiển thị trên các thiết bị di động, nên các tin được lấy về đều phải loại bỏ đi các thẻ html và các ký tự đặc biệt. Thêm vào đó, các tin cần đảm bảo lưu trữ lại cả ảnh và các ảnh này phải hiện thị đúng trên các thiết bị di động với các kích thước khác nhau. Để giải quyết vấn đề ảnh đối với các loại điên thoại di động khác nhau, khóa luận này sử dung phương pháp cache ảnh (lưu giữ ảnh trên chính server của mình). Tức là đối với một ảnh trong tin, module crawl sẽ phải download ảnh về server, sau đó covert ảnh sang định dạng .jpg với 2 chuẩn kích thước có chiều rộng là 172 pixel và 240 pixel. Việc chọn lựa 2 kích thước này là bởi vì trên thị trường phần lớn các loại điện thoại (không kể smart phone cao cấp như Iphone, Android) thì đều có kích thước là 240x320 hoặc 172x220. Sau khi tải và sinh ảnh mới ra trên server, thì nội dung của tin lấy về cũng phải sửa lại đường dẫn các ảnh để các ảnh trong tin trở tới các ảnh trên server

+ Sau khi đã lấy được nội dung và các ảnh từ các báo, các tin sẽ được đưa vào cơ sở dữ liệu của hệ thống. Nhưng trước khi đưa vào cơ sở dữ liệu, các tin cần trải qua bước kiểm tra tính trùng lặp của các tin. Quá trình kiểm tra trùng lặp này sẽ dựa vào nội dung của các tin và so sánh nó với các tin cùng được đưa lên trong 2 ngày gần đây để kiểm tra xem có tin nào giống với nó hay không. Thuật toán kiểm tra trùng lặp sẽ được trình bày chi tiết ở phần tiếp theo.

4.3. Thuật toán kiểm tra sự trùng lặp các tin

4.3.1. Độ giống nhau của hai xâu

Cho hai xâu s1 và s2. Độ giống nhau của hai xâu được tính như sau:

Set1 = { các từ trong xâu s1}

Set2 = { các từ trong xâu s2}

Set3 = Set1 ∩ Set2

Khi đó

$$\text{SimilarityRate} = \text{Min}\ \{\frac{|Set3|}{|Set1|}, \frac{|Set3|}{|Set1|}\}$$

Trong đó |Set| = số phần tử trong tập Set

4.3.2. Thuật toán

Thuật toán kiểm tra sự trùng lặp giữa các tin trong hệ thống được tiến hành bao gồm hai bước với hai tham số là TITLE_SIMILARITY (độ giống nhau của title) và CONTENT_SIMILARITY (độ giống nhau của nội dung)

+ Kiểm tra tiêu đề của hai tin, nếu như độ giống nhau của hai tin là > TITLE_SIMILARITY thì tiến hành sang bước 2

+ Kiểm tra độ giống nhau của nội dung hai tin. Nếu như nội dung hai tin có độ giống nhau > CONTENT_SIMILARITY, thì đánh dấu hai tin này là trùng lặp nhau. Đồng thời trong hai tin xác định tin có thời gian đưa ra trước là tin gốc, còn tin đưa ra sau thì coi là tin đưa lại(việc kiểm tra xem tin nào đưa ra trước, tin nào đưa ra sau dựa vào tham số <pubdate> khi lấy tin từ RSS)

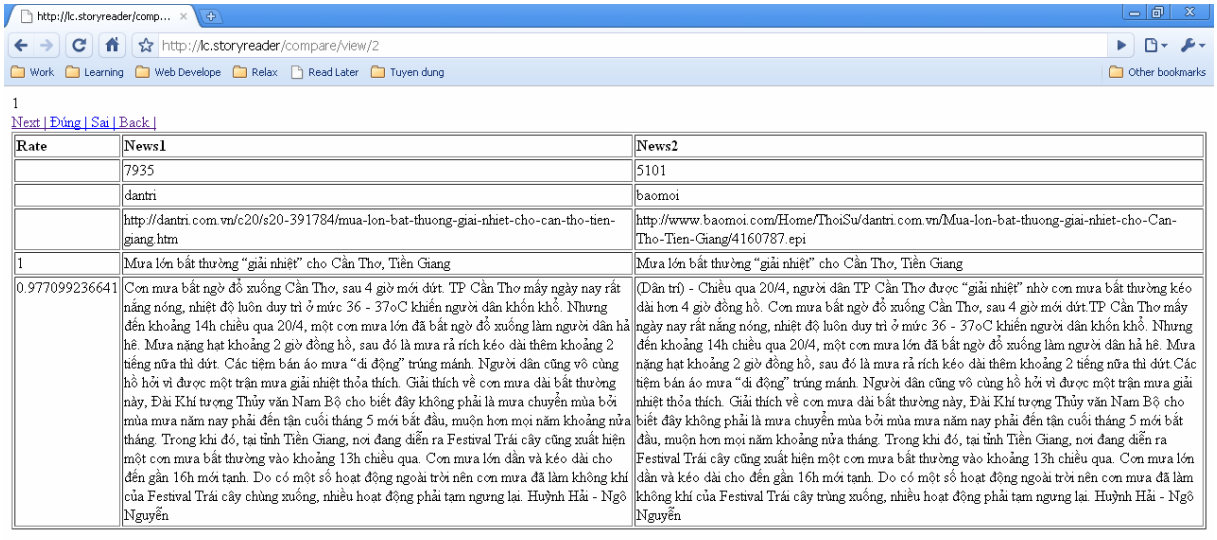
4.3.3. Thực nghiệm và kiểm tra độ chính xác của thuật toán

Bộ test để kiểm tra độ chính xác của thuật toán bao gồm 302 tin được chọn từ 4 nguồn báo trong khoảng thời gian từ ngày 20/04/2010 đến ngày 13/05/2010: vnexpress.net (51 tin), dantri.com.vn (50 tin), vietnamnet.vn (64 tin) và baomoi.vn (136 tin). Bộ test bao gồm 68 cặp tin trùng lặp, đều là các tin từ trang baomoi.vn đăng lại của các nguồn báo kia. Cụ thể các tin đăng lại như sau: 16 tin đăng lại từ vnexpress.vn, 24 tin đăng lại từ dantri.com.vn, 28 tin đăng lại từ vietnamnet.vn

Tất cả các tin được lưu giữ trong cơ sở dữ liệu MySQL server.

Mỗi lần test, chúng tôi thay đổi hai tham số kiểm tra độ tương đồng của các tin tức đó là TITLE_SIMILARITY (mức độ tương đồng của tiêu đề hai bài báo) và CONTENT_SIMILARITY (mức độ tương đồng của nội dung hai bài báo). Ý nghĩ của hai tham số này giống như ở phần 3.3.2 đã trình bày. Trong quá trình kiểm tra, nếu như hai bài báo bất kỳ mà có tỉ lệ giống nhau ở tiêu đề > TITLE_SIMILARITY và ở nội dung > CONTENT_SIMILARITY thì hai bài báo đó được coi là lặp lại nhau.

Sau khi test xong, tất cả các cặp bài báo giống nhau sẽ được lưu vào trong bảng duplicate_news_test của cơ sở dữ liệu. Việc kiểm tra lại từng cặp báo giống nhau mà chương trình đưa ra, được chúng tôi thực hiện lại hoàn toàn bằng tay. Chúng tôi viết một script PHP để xem chi tiết hai bài báo của từng cặp một. Hình 13 là màn hình khi chúng tôi kiểm tra nội dung của từng cặp dữ liệu được đưa ra bởi chương trình. Hai bài báo được so sánh với nhau dựa trên nội dung mà chúng được crawler lấy về.


http://lc.storyreader/comp...

http://lc.storyreader/compare/view/2

Work | Learning | Web Develope | Relax | Read Later | Tuyen dung | Other bookmarks

1

Next | Đúng | Sai | Back |

Rate	News1	News2
	7935	5101
	dantri	baomoi
	http://dantri.com.vn/c20/s20-391784/mua-lon-bat-thuong-giai-nhiet-cho-can-tho-tien-giang.htm	http://www.baomoi.com/Home/ThoiSu/dantri.com.vn/Mua-lon-bat-thuong-giai-nhiet-cho-Can-Tho-Tien-Giang/4160787.epi
1	Mưa lớn bất thường "giải nhiệt" cho Cần Thơ, Tiền Giang	Mưa lớn bất thường "giải nhiệt" cho Cần Thơ, Tiền Giang
0.977099236641	Con mưa bất ngờ đổ xuống Cần Thơ, sau 4 giờ mới dứt. TP Cần Thơ mấy ngày nay rất nắng nóng, nhiệt độ luôn duy trì ở mức 36 - 37oC khiến người dân khốn khổ. Nhưng đến khoảng 14h chiều qua 20/4, một con mưa lớn đã bất ngờ đổ xuống làm người dân hả hê. Mưa năng hạt khoảng 2 giờ đồng hồ, sau đó là mưa rả rích kéo dài thêm khoảng 2 tiếng nữa thì dứt. Các tiệm bán áo mưa "di động" trúng mánh. Người dân cũng vô cùng hồ hởi vì được một trận mưa giải nhiệt thỏa thích. Giải thích về con mưa dài bất thường này, Đài Khí tượng Thủy văn Nam Bộ cho biết đây không phải là mưa chuyển mùa bởi mùa mưa năm nay phải đến tận cuối tháng 5 mới bắt đầu, muộn hơn mọi năm khoảng nửa tháng. Trong khi đó, tại tỉnh Tiền Giang, nơi đang diễn ra Festival Trái cây cũng xuất hiện một con mưa bất thường vào khoảng 13h chiều qua. Con mưa lớn dần và kéo dài cho đến gần 16h mới tạnh. Do có một số hoạt động ngoài trời nên con mưa đã làm không khí của Festival Trái cây chùng xuống, nhiều hoạt động phải tạm ngưng lại. Huỳnh Hải - Ngô Nguyễn	(Dân trí) - Chiều qua 20/4, người dân TP Cần Thơ được "giải nhiệt" nhờ con mưa bất thường kéo dài hơn 4 giờ đồng hồ. Con mưa bất ngờ đổ xuống Cần Thơ, sau 4 giờ mới dứt.TP Cần Thơ mấy ngày nay rất nắng nóng, nhiệt độ luôn duy trì ở mức 36 - 37oC khiến người dân khốn khổ. Nhưng đến khoảng 14h chiều qua 20/4, một con mưa lớn đã bất ngờ đổ xuống làm người dân hả hê. Mưa năng hạt khoảng 2 giờ đồng hồ, sau đó là mưa rả rích kéo dài thêm khoảng 2 tiếng nữa thì dứt.Các tiệm bán áo mưa "di động" trúng mánh. Người dân cũng vô cùng hồ hởi vì được một trận mưa giải nhiệt thỏa thích. Giải thích về con mưa dài bất thường này, Đài Khí tượng Thủy văn Nam Bộ cho biết đây không phải là mưa chuyển mùa bởi mùa mưa năm nay phải đến tận cuối tháng 5 mới bắt đầu, muộn hơn mọi năm khoảng nửa tháng. Trong khi đó, tại tỉnh Tiền Giang, nơi đang diễn ra Festival Trái cây cũng xuất hiện một con mưa bất thường vào khoảng 13h chiều qua. Con mưa lớn dần và kéo dài cho đến gần 16h mới tạnh. Do có một số hoạt động ngoài trời nên con mưa đã làm không khí của Festival Trái cây trùng xuống, nhiều hoạt động phải tạm ngưng lại. Huỳnh Hải - Ngô Nguyễn

Hình 13. Màn hình để kiểm tra nội dung hai bản tin.

Cụ thể các lần chạy test như sau:

+ Lần 1: TITLE_SIMILARITY = CONTENT_SIMILARITY = 90%. Kết quả phát hiện ra 46 tin trùng lặp. Thời gian chạy : 1.5150001049 s

+ Lần 2: TITLE_SIMILARITY = CONTENT_SIMILARITY = 80%. Kết quả phát hiện 57 tin trùng lặp. Thời gian chạy 1.65600013733 s

+ Lần 3: TITLE_SIMILARITY = CONTENT_SIMILARITY = 70%. Kết quả phát hiện: 63 tin trùng lặp. Thời gian chạy: 1.82899999619s

+ Lần 4: TITLE_SIMILARITY = CONTENT_SIMILARITY = 60%. Kết quả phát hiện 64 tin trùng lặp, trong đó có một tin phát hiện không chính xác. Thời gian chạy: 1.78099989891s

+ Lần 5: TITLE_SIMILARITY = 50%, CONTENT_SIMILARITY = 0 (coi như chỉ chạy với TITLE). Kết quả phát hiện 71 tin trùng lặp, trong đó có 3 tin sai. Thời gian chạy: 1.90600013733s

4.3.4. Phân tích lỗi

Qua các lần chạy thực nghiệm, ta rút ra kết luận nếu để TITLE_SIMILARITY và CONTENT_SIMILARITY càng thấp thì càng phát hiện ra nhiều tin trùng lặp. Tuy nhiên trong đó lại có nhiều nguy cơ phát hiện ra các tin không chính xác. Ví dụ với lần chạy thứ 4, phát hiện ra 2 tin có id 5660 và 5400 là trùng lặp nhau. Hai tin này tương ứng với hai link: http://vietnamnet.vn/xahoi/201004/2-oto-cua-Bi-thu-Dang-uy-bi-gai-min-lien-tiep-905669/, và http://vnexpress.net/GL/Phap-luat/2010/04/3BA1B0F8/. Cụ thể nội dung là tiêu đề của hai tin như sau:

5660	5400
http://vietnamnet.vn/xahoi/201004/2-oto-cua-Bi-thu-Dang-uy-bi-gai-min-lien-tiep-905669/	http://vnexpress.net/GL/Phap-luat/2010/04/3BA1B0F8/
2 ôtô của Bí thư Đảng ủy bị gài mìn liên tiếp	**Ôtô của bí thư đảng ủy bị cài mìn**
Theo những người dân quanh khu vực cho biết, tiếng nổ phát ra vào rạng sáng ngày	Ôtô của bí thư đảng ủy bị cài mìn Hai quả mìn tự tạo được cài trong hai ôtô tại

19/4 tại nhà riêng của ông Đỗ Văn Công (Thị trấn Uyên Hưng, huyện Uyên Hưng), Bí thư Đảng ủy khối Dân chính tỉnh Bình Dương. Thông tin ghi nhận ban đầu cho thấy nhà ông Công có 2 chiếc xe ôtô là chiếc Toyota Land Cruiser cùng một chiếc xe bán tải. Tiếng nổ kia được xác định phát ra trên chính chiếc xe Toyota. Tuy nhiên rất may không có người nào bị thương. Sau vụ nổ, một bánh của chiếc xe Toyota bị nát toàn bộ. Thấy vậy ông Công đã chuyển sang lái chiếc xe bán tải để đến chỗ làm. Do vẫn chưa thật sự yên tâm về độ an toàn nên ngay lập tức ông xuống xe tiến hành kiểm tra và ngỡ ngàng khi nhìn thấy một vật lạ gần giống quả mìn được cài đặt dưới nắp capo. Nhận được tin báo, các cơ quan chức năng đã đến ngay hiện trường để xem xét, điều tra vụ việc. Kết quả ban đầu cho thấy, quả mìn được đặt trên xe bán tải là một loại mìn tự tạo cỡ nhỏ được kích nổ tự động thông qua điện thoại di động. Hiện vụ việc đang được cơ quan chức năng khẩn trương điều tra, làm rõ. Vũ Đạt	nhà Bí thư Đảng ủy khối Dân chính đảng tỉnh Bình Dương Đỗ Văn Công. Một quả đã phát nổ. Rạng sáng 19/4, tại khu để xe trong nhà riêng của ông Đỗ Văn Công tại thị trấn Uyên Hưng, huyện Uyên Hưng, chiếc Toyota Land Cruiser bỗng phát nổ tại vùng bánh xe bởi một quả mìn tự tạo mà ai đó đã cài sẵn. Tuy nhiên, vụ nổ này không gây thiệt hại cho người và phương tiện. Sau đó đến giờ đi làm, vị bí thư định lái chiếc xe khác (xe bán tải) đến cơ quan thì tiếp tục phát hiện một vật lạ nằm dưới nắp ca-po chiếc xe này. Nhận được tin báo, cơ quan chức năng đã có mặt phong tỏa hiện trường, phục vụ cho công tác tháo gỡ vật lạ kia. Qua kiểm tra, cơ quan chức năng xác định đây là quả mìn tự tạo giống như quả phát nổ trước đó. Nó có hình trụ bằng giấy nặng 500 g, trong đó gồm 200 g thuốc nổ dạng công nghiệp màu đỏ, bộ phận kích nổ gắn với chiếc điện thoại di động. Kiểm tra chiếc điện thoại này, lực lượng chức năng thấy có 4 cuộc gọi nhỡ. Cơ quan điều tra nhận định, kẻ xấu đã kích nổ nhiều lần nhưng không thành. Đây có thể là hành động trả thù ông Đỗ Văn Công. Vụ việc đang được cơ quan chức năng khẩn trương làm rõ. Nguyệt Triều

Mặc dù hai tin này cùng đưa về một nội dung, nhưng đều chứa các tình tiết khác nhau. Tuy nhiên do thuật toán chỉ kiểm tra các từ trùng lặp giữa hai tin nên vẫn cho rằng đây là hai tin trùng nhau.

Một trường hợp khác. Khi chạy với độ chính xác là 60 % vẫn không phát hiện ra hai tin có id là 7966 (link http://vietnamnet.vn/xahoi/201004/Chum-anh-Kham-pha-nhung-dia-dao-tai-pho-co-Ha-Noi-905651/) và 5299 (link http://www.baomoi.com/Info/Chum-anh-Kham-pha-nhung-dia-dao-tai-pho-co-Ha-Noi/137/4162367.epi). Mặc dù bài báo trên trang baomoi.vn là đăng lại từ bài báo trên trang vietnamnet, nhưng do ở trang baomoi.vn, các nội dung có nhiều ảnh thì các ảnh sẽ bị cắt đi và đẩy xuống cuối bài, đồng thời các tiêu đề liên quan đến ảnh cũng bị loại bỏ nên độ chính xác khi so sánh nội dung là rất thấp. Chính vì thế thuật toán không phát hiện ra được trường hợp này.

Ngoài ra, từ thời gian chạy của các test, ta cũng thấy thời gian để thuật toán kiểm tra độ trùng lặp của tin là rất nhanh. Thời gian kiểm tra 302 tin tức là $\frac{302x301}{2}$ = 45451 cặp tin là < 2s. Do vậy nếu với số lượng tin một ngày < 2000 tin thì thời gian kiểm tra sẽ rất nhanh.

4.4. Tổng kết chương

Trong chương này, chúng tôi đã trình bày chi tiết về hoạt động của module thu thập và phát hiện tin tức trùng lặp. Chúng tôi cũng đưa ra thuật toán để phát hiện tin tức trùng lặp. Thuật toán tuy đơn giản, nhưng thực nghiệm chỉ ra độ thời gian chạy thuật toán rất nhanh (qua 5 test, thời gian để so sánh 45451 cặp tin đều < 2s) và độ chính xác cũng chấp nhận được (điều này phụ thuộc vào việc lựa chọn hai tham số quyết định độ trùng lặp nhỏ nhất của tiêu đề và nội dung bài báo là TITLE_SIMILARITY và CONTENT_SIMILARITY).

Nằm trong tầng lưu giữ (Persistant tier), có thể nói hoạt động của module thu thập và phát hiện tin tức trùng lặp là hoàn toàn bị che giấu với người dùng thực sự. Tuy nhiên vai trò của nó lại vô cùng quan trọng. Toàn bộ dữ liệu của hệ thống đều được tổng hợp nhờ module này.

Chương tiếp theo, sẽ trình bày chi tiết về ứng dụng mNews - ứng dụng đọc báo trên mobile được chúng tôi xây dựng trên công nghệ J2ME của SUN và framework KUIX.

Chương 5

Xây dựng ứng dụng đọc báo mNews trên di động

5.1. Ứng dụng đọc báo trên di động:

Ứng dụng mNews là một ứng dụng viết bằng ngôn ngữ J2ME dựa trên framework KUIX được chạy trên các điện thoại di động. Ứng dụng chính là tầng trình diễn (Presentation tier) trong mô hình ba tầng của kiến trúc hệ thống đã được trình bày chi tiết ở chương 2.

Ứng dụng mNews là một client, mỗi khi chạy, ứng dụng sẽ kết nối vào web server của hệ thống và lấy về các tin bài được hệ thống thu thập thông qua tầng lưu giữ (Persistant tier).

5.2. Phân tích yêu cầu

5.2.1. Yêu cầu người sử dụng

- Người dùng có thể chọn lựa đọc tin theo hai hình thức: đọc tin theo từng chuyên mục, hoặc là đọc theo thứ tự các tin mới nhất
- Khi đọc một tin yêu cầu cần có ảnh minh họa đối với các tin đó. Các tin tức nếu bị trùng lặp thì chỉ hiển thị tin gốc
- Có thể duyệt các trang tin theo thứ tự được
- Cần có chức năng tìm kiếm để giúp người dùng tìm các tin liên quan dễ dàng

5.2.2. Yêu cầu đối với hệ thống

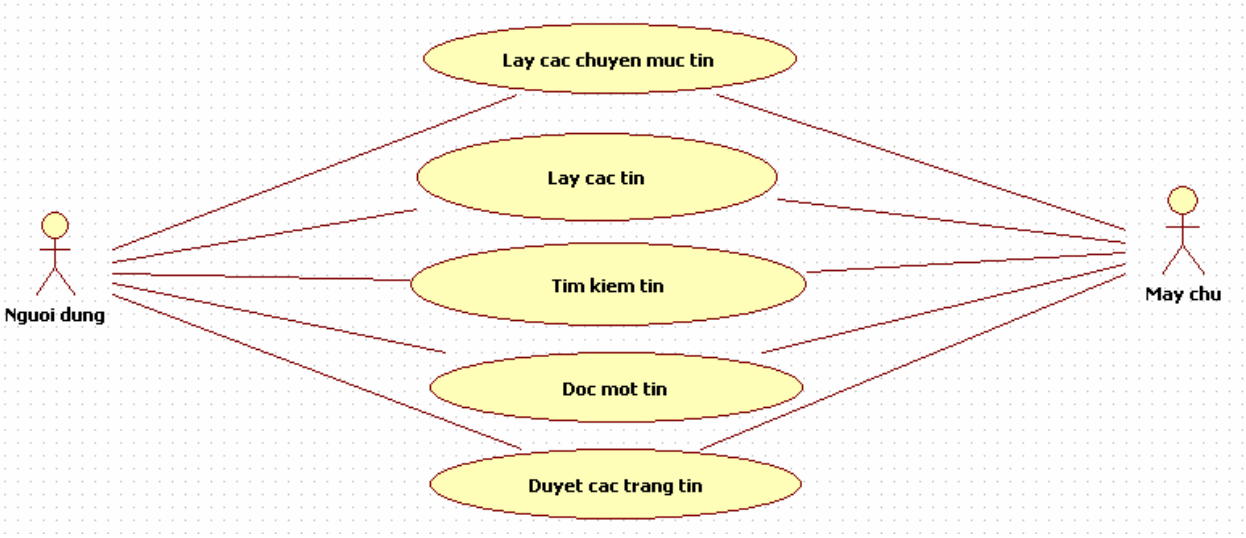
Từ các yêu cầu của người dùng, hệ thống cần có các chức năng sau:

- Cung cấp các tin theo từng chuyên mục riêng biệt, sắp xếp các tin theo thứ tự giảm dần của thời gian cập nhật
- Đọc tin: Tin tức được lấy từ các nguồn báo trong nước. Khi có những tin dài quá, cần tự động cắt tin để tin hiện thị phù hợp trên điện thoại. Nếu

một tin bị cắt thành > 1 trang, thì cần có chức năng cho người dùng chọn lựa giữa các trang tin cần đọc. Cụ thể, khi người dùng ấn phím Left thì chuyển về trang trước đó, ấn phím Right thì chuyển sang trang kế tiếp. Ngoài ra còn cần có chức năng cho người dụng lựa chọn tùy ý trang muốn nhảy tới

- Tìm kiếm tin: Hệ thống tìm trong cơ sở dữ liệu tin tức chứa từ khóa cần tìm và trả về một danh sách các tin cho người dùng

5.3. Biểu đồ Usecase



Hình 14. Biểu đồ Usecase phần mềm mNews

Biểu đồ Usecase của hệ thống có hai tác nhân đó là Người dùng và Server. Có năm chức năng chính đó là: Lấy các chuyên mục tin, Lấy các tin mới nhất, Lấy các tin trong chuyên mục, Đọc một tin, Duyệt các trang tin

5.3. Luồng sự kiện

5.3.1. Lấy các chuyên mục tin

Bảng 3 . Usecase Lấy các chuyên mục tin

Tên Use Case	**Lấy các chuyên mục tin**
Tác nhân	Người dùng, Server
Mức	2
Sự kiện kích hoạt	Người dùng lựa chọn chức năng đọc theo chuyên mục
Luồng sự kiện chính: 1. Hiện ra thanh load dữ liệu ở dưới màn hình 2. Phần mềm gửi yêu cầu tới máy chủ 3. Máy chủ lấy ra các chuyên mục tin từ cơ sở dữ liệu và trả về cho phần mềm 4. Phần mềm render dữ liệu trả về thành giao diện danh sách các chuyên mục cho người dụng lựa chọn	
Luồng sự kiện phụ: 2.1 Không thể kết nối tới máy chủ, yêu cầu kết nối lại	

5.3.2. Lấy các tin

Bảng 4. Usecase Lấy các tin

Tên Use Case	**Lấy các tin**
Tác nhân	Người dùng, Máy chủ
Mức	2
Sự kiện kích hoạt	Người dùng lựa chọn chức năng đọc tin mới nhất, hoặc lựa chọn đọc tin theo một chuyên mục
Luồng sự kiện chính: 1. Hiện ra thanh load dữ liệu ở dưới màn hình 2. Phần mềm gửi yêu cầu tới máy chủ 3. Máy chủ lấy ra các tin trong từng chuyên mục trả về cho người dùng.	

4. Phần mềm render dữ liệu trả về thành giao diện danh sách các tin cho người dụng lựa chọn đọc
Luồng sự kiện phụ: 2.1. Không thể kết nối tới máy chủ, yêu cầu kết nối lại

5.3.3. Tìm kiếm tin

Bảng 5. Usecase Tìm kiếm tin

Tên Use Case	**Tìm kiếm tin**
Tác nhân	Người dùng, Máy chủ
Mức	2
Sự kiện kích hoạt	Người dùng gõ vào từ để tìm kiếm
Luồng sự kiện chính: 1. Hiện ra thanh load dữ liệu ở dưới màn hình 2. Phần mềm gửi một POST request lên máy chủ có chứa từ để tìm 3. Máy chủ tìm trong cơ sở dữ liệu và trả về các tin có chứa từ cần tìm 4. Phần mềm render dữ liệu trả về thành giao diện danh sách các tin cho người dụng lựa chọn đọc	
Luồng sự kiện phụ: 2.1. Không thể kết nối tới máy chủ, yêu cầu kết nối lại	

5.3.4. Đọc một tin

Bảng 6. Usecase Đọc một tin

Tên Use Case	**Đọc một tin**
Tác nhân	Người dùng, Máy chủ
Mức	2
Sự kiện kích hoạt	Người dùng lựa chọn một tin trong danh sách
Luồng sự kiện chính: 1. Hiện ra thanh load dữ liệu ở dưới màn hình	

2. Phần mềm gửi yêu cầu tới máy chủ 3. Máy chủ lấy ra các tin trong từng chuyên mục trả về cho người dùng. 4. Phần mềm render dữ liệu trả về thành giao diện của tin cho người dùng. 5. Nếu dữ liệu trả về có chứa các link ảnh. Phần mềm gửi request tới link các ảnh đó 6. Máy chủ trả về nội dung các ảnh 7. Phần mềm tạo ra ảnh và đặt vào đúng vị trí trong phần tin tức vừa mới lấy được
Luồng sự kiện phụ: 2.1. Không thể kết nối tới máy chủ, yêu cầu kết nối lại

5.3.5. Duyệt các tin

Bảng 7. Usecase Duyệt các tin

Tên Use Case	**Duyệt các tin**
Tác nhân	Người dùng, Máy chủ
Mức	2
Sự kiện kích hoạt	Người dùng ấn vào phím sang trái, sang phải, hoặc gõ vào số trang cần nhảy tới
Luồng sự kiện chính: 1. Hiện ra thanh load dữ liệu ở dưới màn hình 2. Phần mềm sinh ra link tương ứng với số trang mà người dùng muốn tới, và gửi request tới máy chủ 3. Máy chủ tìm trong cơ sở dữ liệu và trả về các tin có chứa từ cần tìm 4. Phần mềm render dữ liệu trả về thành giao diện danh sách các tin cho người dụng lựa chọn đọc	
Luồng sự kiện phụ: 2.1. Không thể kết nối tới máy chủ, yêu cầu kết nối lại	

5.4. Giao diện của ứng dụng:

Hình 15. Giao diện khi chạy ứng dụng

Hình 16. Giao diện danh sách các chuyên mục tin

Hình 17. Giao diện các tin trong một chuyên mục

Hình 18. Giao diện chi tiết một tin

5.5. Giao thức giữa ứng dụng và máy chủ

5.5.1. So sánh kết nối bằng socket và kết nối bằng HTTP

Giao thức kết nối giữa một máy khách trên điện thoại di động bằng J2ME và một máy chủ có thể là một trong hai kiểu sau: Kết nối thông qua socket, hoặc kết nối thông qua HTTP

Bảng 8. So sánh giữa kết nối bằng socket và kết nối bằng HTTP

	Kết nối socket	**Kết nối HTTP**
Ưu điểm	- Thời gian tạo kết nối nhanh - Chỉ cần duy trì duy nhất một kết nối trong quá trình sử dụng ứng dụng - Không mất thời gian tạo kết nối, khi thực hiện yêu cầu tiếp theo tới server	- Cài đặt trên điện thoại và trên server đơn giản (do J2ME đã hỗ trợ cách thức này) - Tất cả các dòng máy đều hỗ trợ
Nhược điểm	- Phía server cài đặt phức tạp - Một số dòng điện thoại không hỗ trợ kết nối socket, ví dụ như: Motorola ROKR E6	- Phải tạo nhiều kết nối tới server - Thời gian chạy sẽ chậm hơn do mất thời gian khởi tạo kết nối

Nhìn vào bảng 8 ta có thể thấy, kết nối tạo bằng socket có được ưu điểm lớn là thời gian tạo kết nối rất nhanh, hơn nữa chỉ mất duy nhất một lần tạo kết nối. Điều này rất quan trọng trong các ứng dụng J2ME bởi vì khi chạy trên một thiết bị thật, vì những yêu cầu bảo mật, các ứng dụng khi muốn truy cập tới các tài nguyên như: tương tác với internet, tương tác qua mạng (nhắn tin sms, gọi điện), tương tác đọc/ghi với bộ nhớ của thiết bị, … đều bị hỏi quyền truy cập. Chính vì thế, bằng cách chỉ tạo ra một kết nối socket và giữ cho tới khi ứng dụng bị đóng, sẽ tạo ra tiện lợi rất lớn cho người dùng. Tuy nhiên, do việc cài đặt trên phía server đối với kết nối socket lại rất phức tạp. Server sẽ phải xử lý việc đa kết nối, và đồng thời phải lưu và giữ cho tất cả kết nối hoạt động. Như thế server sẽ phải chịu tải rất lớn. Trong khi đó, kết nối bằng HTTP, tuy sẽ mất thời gian hơn trong việc khởi tạo kết nối, bởi mỗi lần ứng dụng yêu cầu lên server, ứng dụng phải sinh ra một kết nối mới. Tuy nhiên, việc cài đặt lại đơn giản hơn

rất nhiều, phía server, ta sẽ dùng chính web server để xử lý, còn phía client, ta sử dụng Collection Framework đã được hỗ trợ sẵn trong J2ME.

Chính vì thế, trong khóa luận này, chúng tôi sử dụng kết nối dạng HTTP để việc cài đặt được đơn giản hơn.

5.5.2. Chi tiết giao thức

Khi ứng dụng mNews muốn gửi một yêu cầu tới máy chủ, ứng dụng sẽ gọi tới các PHP script đã được cài đặt trên server. Việc gọi tới các script này được thực hiện thông qua các HTTP GET/POST request.

Khi nhận được yêu cầu từ phía client, máy chủ trả về các message với định dạng xác định. Mỗi định dạng máy chủ trả về, ứng dụng mNews sẽ render ra giao diện phù hợp. Cụ thể ở đây là 3 dạng giao diện

Giao thức liệt kê các chuyên mục:

```
$prev_link|$next_link|$title|$status|$search_link|
$item1_title;$item1_link|
$item2_title;$item2_link|…
```

Trong đó:

+ $prev_link là link trang liền trước của trang hiện thị, nếu số trang > 1

+ $next_link là trang liền sau của trang hiển thị.

+ $title là tiêu đề của trang

+ $status là dòng chữ hiện thị ở góc dưới của trang (nó có dạng số trang hiện tại/tổng số trang. Ví dụ: < 3/10 >)

+ $search_link là link sẽ được request tới khi người dùng gõ vào ô tìm kiếm. Nếu link này là “”, thì phần mềm sẽ không hiển thị ô tìm kiếm.

+ $item_title là tiêu đề của một chuyên mục

+ $item_link là đường dẫn tới chuyên mục đó

Giao thức liệt kê tin trong một chuyên mục

```
$prev_link|$next_link|$title|$status|$search_link|
$item1_title;$item1_link;$item1_description;|
$item2_title;$item2_link;$item2_description|…
```

Giống với giao thức khi liệt kê các chuyên mục, nhưng mỗi item có thêm một tham số là $item_description là mô tả cho tin tức đó.

Giao thức này cũng dùng để liệt kê các tin mới nhất, và các tin tìm được tương ứng

Giao thức chi tiết một tin

```
$prev_link|$next_link|$title|$status|
$news_title|$news_content|
```

+ $news_title: tiêu đề của tin

+ $news_content: nội dung tin

Nội dung của tin có thể có chứa các thẻ dạng <img>$image_link</img> - là link tới các ảnh trong tin. Trong quá trình parse, nếu gặp đoạn mã này, ứng dụng sẽ tạo các kết nối để lấy các về nội dung ảnh từ $image_link.

5.6. Parser dữ liệu từ server gửi về

Sau khi nhận được dữ liệu từ phía server gửi về, phần mềm sẽ parse dữ liệu để sinh ra các giao diện cho người dùng.

Giao diện của người dùng được sinh từ các file XML. Tương ứng với 3 kiểu dữ liệu trả về là 3 file XML

File XML ứng với giao thức liệt kê các chuyên mục tin

```
<scrollPane class="listtext">
  <list id="list">
    <_renderer>
    <![CDATA[
    <listItem class="listtext">
      <_onAction>link(0, @{link})</_onAction>
      <container class="listtext">
        <text class="slide listtext bold">@{title}</text>
      </container>
    </listItem>
    ]]>
    </_renderer>
    <_items>@{entry}</_items>
  </list>
</scrollPane>
```

File XML ứng với giao thức liệt kê các tin

```
<scrollPane class="listtext">
  <list>
    <_renderer>
    <![CDATA[
    <listItem class="listtext">
      <_onAction>link(0, @{link}, @{number}, {message})</_onAction>
      <container class="listtext">
        <text class="slide listtext bold">@{title}</text>
        <text style="font-size:small;">@{description}</text>
      </container>
      <picture src="icon/arrow.png" class="listtext">
        <_visible>@{picVisible}</_visible>
      </picture>
    </listItem>
    ]]>
    </_renderer>
    <_items>@{entry}</_items>
  </list>
</scrollPane>
```

File XML ứng với giao thức đọc một tin

```
<scrollPane class="detail">
  <textarea class="bold title">@{title}</textarea>
  <picture class="news"><_src>@{picLink}</_src></picture>
  <list>
    <_renderer>
    <![CDATA[
    <textarea class="detail">@{content}</textarea>
    ]]>
    </_renderer>
    <_items>@{entry}</_items>
  </list>
</scrollPane>
```

5.7. Bài toán xử lý tiếng Việt trên điện thoại

Đối với điện thoại di động, việc hiện thị tiếng Việt, có thể coi như việc hiện thị một font mới trên điện thoại. Đối với bài toán này có một cách tiếp cận rất hay được sử dụng đó là dùng một file ảnh chứa các ảnh của từng ký tự, mỗi ký tự này được chứa trong một cửa sổ với kích thước xác định. Khi ứng dụng chạy, sẽ đọc file ảnh đó và

tách từng ký tự ra một. Cách làm này đảm bảo sẽ hiện thị đúng tiếng Việt trên tất cả các dòng máy.

Tuy nhiên có một vấn đề đó là nếu sử dụng các ảnh thông thường, mỗi khi muốn thay đổi màu chữ, hoặc thay đổi kiểu chữ (như chữ viết thường, chữ in nghiêng, chữ in đậm) ta lại phải tạo ra một ảnh mới. Điều này rất lãng phí.

Để giải quyết vấn đề này, Sergey Tkachev[13] đã đưa ra giải pháp như sau: tất cả các ảnh được tạo bởi các pixel với màu đen trên nền trong suốt trong hệ màu PNG. Khi chúng ta muốn vẽ một ký tự lên màn hình, phần hình chữ nhật tương ứng với ký tự đó sẽ được vẽ lên canvas của J2ME.

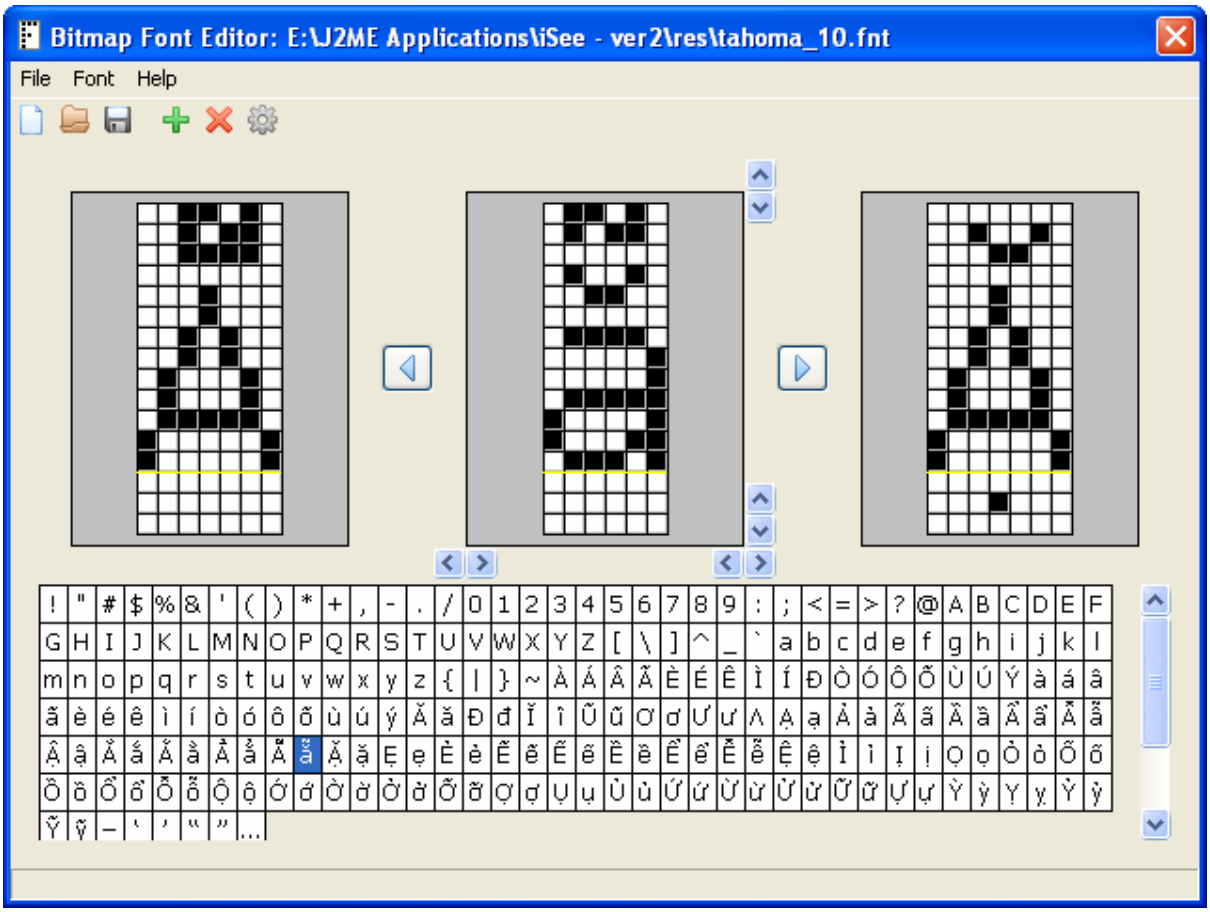
Các kiểu chữ khác nhau có thể đạt được từ kiểu chữ thông thường bằng cách sau:

- Kiểu chữ bôi đậm: Một ký tự bôi đậm được vẽ từ hai ký tự bình thường liên tiếp nhau, cách nhau đúng một 1 pixel theo chiều ngang.
- Kiểu chữ in nghiêng: Mỗi ký tự in nghiêng được tạo thành từ ký tự bình thường bằng cách dịch chuyển các bit ở nửa trên của ký tự đó sang phải 1 pixel
- Kiểu chữ nghiêng đậm: thì sử dụng 2 cách ở trên

Ảnh tạo ra chỉ bao gồm các ký tự màu đen, vậy các màu khác thì làm sao để có thể tạo ra. Khi muốn thay đổi màu của một ký tự, ta đơn giản chỉ cần thay đổi màu vẽ của đối tượng graphics là được. Nếu màu không phải là màu đen (0x000000), font chữ sẽ tạo ra ảnh mới cho ký tự bằng cách load ký tự và cập nhật bộ byte hiện thị màu của chúng. Quá trình này tốn một khoảng thời gian, nên cách tốt nhất là lưu các ảnh màu vào một bộ nhớ tạm. Kích thước của bộ nhớ tạm này là bị giới hạn và màu cuối cùng trong mảng bộ nhớ tạm sẽ bị xóa khi nó tới ngưỡng giới hạn

Việc tạo nên các file ảnh cho ứng dụng, được tạo bởi bộ thư viện mã nguồn mở Bitmap Font Editor, bộ thư viện này có thể được tải về tại địa chỉ http://sourceforge.net/projects/mobilefonts/.

Hình 19 là giao diện khi sử dụng phần mềm Bitmap Font Editor để tạo nên file ảnh cho bộ font Tahoma cỡ chữ 10pt. Đây là bộ font được chúng tôi sử dụng trong chính ứng dụng mNews



Hình 19. Tạo font bằng phần mềm Bitmap Font Editor

5.8. Tổng kết chương

Trong chương này, chúng tôi đã giới thiệu chi tiết về cách thức hoạt động và cài đặt của phần mềm mNews trên điện thoại di động. Phần mềm mNews được viết bằng ngôn ngữ Java trên nền tảng J2ME với sự hỗ trợ của framework KUIX.

Trong chương này chúng tôi cũng trình bày giải pháp để giải quyết bài toán hiển thị tiếng Việt trên phần lớp các loại điện thoại đời thấp không hỗ trợ các font chữ unicode thông qua việc sử dụng các file ảnh thay thế các font chữ. Bằng cách này, để hiện thị các dòng chữ tiếng Việt, phần mềm sẽ vẽ lại tất cả các ảnh của các ký tự tạo nên dòng chữ đó.

Dựa trên cơ chế xử lý sự kiện dựa theo các message của KUIX (như đã trình bày trong chương 4), phần mềm mNews hỗ trợ thao tác trên cả các dòng điện thoại có màn hình cảm ứng và không có màn hình cảm ứng. Tất cả các tin bài trên phần mềm đều có chứa các hình ảnh với kích thước phù hợp với màn hình hiện thị. Điều này giúp cho việc đọc tin trên điện thoại di động bảo đảm giống như đọc tin trên web thông thường

Chương 6

Tổng kết

Thông qua khóa luận, chúng tôi đã xây dựng được một hệ thống thu thập thông tin từ các nguồn báo tiếng Việt trên mạng thông qua các kênh RSS feed chạy ổn định và nhanh chóng cập nhật. Chúng tôi cũng đưa ra thuật toán đơn giản để phát hiện ra các tin tức trùng lặp từ các nguồn báo khác nhau với thời gian chạy nhanh (trong tất cả các test, để kiểm tra 45451 cặp tin, thời gian chạy < 2s) và kết quả cũng khá tốt (trong hệ thống khi để hai tham số TITLE_SIMILARITY và CONTENT_SIMILARITY là 0.7, độ chính xác đạt được là 90%)

Cùng với hệ thống tự động thu thập và xử lý tin tức chạy trên máy chủ, chúng tôi cũng phát triển một phần mềm mNews chạy trên các điện thoại hỗ trợ Java để đọc các tin tức mà hệ thống cập nhật được. Phần mềm mNews đưa ra giao diện thao tác đơn giản hỗ trợ các dòng máy điện thoại có màn hình cảm ứng lẫn không có màn hình cảm ứng. Việc hiện thị tiếng Việt trên phần mềm được thực hiện tốt trên hầu hết các loại điện thoại hỗ trợ Java nhờ giải pháp sử dụng các ảnh PNG để thay cho font chữ.

Việc phân loại các chuyên mục tin tức hiện nay của hệ thống đang được thực hiện bằng cách tạo nên các bảng ánh xạ chuyên mục từ nguồn báo gốc, tới các chuyên mục đã có sẵn trên hệ thống. Việc ánh xạ này đôi khi chưa thực sự chính xác. Trong tương lai gần, chúng tôi sẽ áp dụng các thuật toán phân lớp để thực hiện quá trình này một cách tự động hoàn toàn

Một hướng phát triển cho phần mềm mNews đó là để tăng tốc độ load dữ liệu từ máy chủ về phần mềm, đó là sử dụng duy nhất một kết nối socket trong suốt quá trình chạy phần mềm. Việc này đòi hỏi cả sự thay đổi ở phía server. Server sẽ phải lưu giữ hàng trăm ngàn kết nối socket một lúc. Một giải pháp đã được đưa ra cho vấn đề này đó là sử dụng các tiếp cận Non Blocking IO.

Tài liệu tham khảo

[1] El-Sayed Atlam, M. Fuketa, K. Morita, Jun-ichi Aoe , *Documents similarity measurement using field association terms*, pp 804-829, Information Processing and Management 39, 2003.

[2] Vikram Goyal, *Pro J2ME MMAPI: Mobile Media API for J2ME,* Apress, 2006

[3] J. Knudsen, S. Li, *Beginning J2ME From Novice to Professional*, Apress, Chapter 3, 2005.

[4] Lê Ngọc Quốc Khánh, *Xây dựng hệ thống M-Commerce: Hỗ trợ thông tin tuyển sinh trên điện thoại di động áp dụng công nghệ Java*, Luận văn tốt nghiệp, 2004

[5] Ralf Steinberger, Bruno Pouliquen, Johan Hagman, *Cross-lingual Document Similarity Calculation Using the Multilingual Thesaurus*, In CICLing, pp 415, 2002.

[6] Cong Thanh Truong, The Duy Bui, Bao Son Pham, *Near-Duplicates Detection for Vietnamese Documents in Large Database*, International Conference on Advanced Language Processing and Web Information Technology, pp.70-75, 2008.

[7] J. White, D. Hemphill, *Java 2 Micro Editon, Java in Smallthing*, Manning Publications, 2002.

[8] Michael Juntao Yuan, *Enterprise J2ME: Developing Mobile Java Applications*, Prentice Hall PTR, 2003.

[9] *2010: The year of mobile*, http://www.beingpeterkim.com/2010/01/2010-mobile.html, Being Peter Kim, 2010.

[10] *CakePHP Cookbook*, http://book.cakephp.org, Cake Software Foundation, 2010.

[11] *February 2009 Web Server Survey*, http://news.netcraft.com/archives/2009/02/18/february_2009_web_server_survey.html, Netcraft, 2009.

[12] *KUIX Project*, http://www.kalmeo.org/projects/kuix, Kalmeo, 2008

[13] *Mobile Bitmap Fonts*, http://mobilefonts.sourceforge.net/, Sergey Tkachev

[14] *RSS Tutorial*, http://w3schools.com/RSS/, W3Schools.

[15] *Socbay iMedia*, http://mobile.socbay.com/, Naiscorp, 2010.

[16] *Top Sites in Vietnam*, http://www.alexa.com/topsites/countries/VN, Alexa, 2010.

[17] *What is Python Good For?*, http://www.python.org/doc/faq/general/, General Python FAQ, Python Foundation. 2008.