**NHẬN XÉT CỦA GIÁO VIÊN HƯỚNG DẪN**

Hà Nội, ngày…..tháng……2014

*Giáo viên hướng dẫn*

**ThS. Lưu Xuân Duẩn**

MỤC LỤC

[LỜI NÓI ĐẦU 6](#_Toc385172526)

[LỜI CẢM ƠN 9](#_Toc385172527)

[CHƯƠNG I: KHẢO SÁT THỰC TẾ 10](#_Toc385172528)

[1.1. Sơ qua về hệ điều hành Android 10](#_Toc385172529)

[1.2. Thị trường ứng dụng trên thiết bị di động ở Việt Nam 11](#_Toc385172530)

[1.3. Ứng dụng giải trí đọc truyện trên Android 11](#_Toc385172531)

[CHƯƠNG II: CÔNG NGHỆ SỬ DỤNG 14](#_Toc385172532)

[2.1. Ngôn ngữ lập trình Java 14](#_Toc385172533)

[2.1.1. Giới thiệu ngôn ngữ lập trình Java 14](#_Toc385172534)

[2.1.2. Những đặc điểm của ngôn ngữ Java 14](#_Toc385172535)

[2.2. Hệ quản trị cơ sở dữ liệu SQLite 16](#_Toc385172536)

[2.2.1. Giới thiệu SQLite 16](#_Toc385172537)

[2.2.2. Một số đặc điểm chính của SQLite 17](#_Toc385172538)

[2.2.3. Một số hạn chế của SQLite 18](#_Toc385172539)

[2.2.4. Ứng dụng cơ sở dữ liệu của SQLite 18](#_Toc385172540)

[2.3. Nền tảng phát triển ứng dụng Android 19](#_Toc385172541)

[2.3.1. Lịch sử phát triển 19](#_Toc385172542)

[2.3.2. Kiến trúc Android 21](#_Toc385172543)

[2.3.3. Các thành phần của Android 25](#_Toc385172545)

[2.3.4. Bộ công cụ phát triển Eclipse for Android 30](#_Toc385172546)

[CHƯƠNG III: PHÂN TÍCH BÀI TOÁN 31](#_Toc385172547)

[3.1. Mục đích của đề tài 31](#_Toc385172548)

[3.2. Mô hình tương tác của phần mềm 31](#_Toc385172549)

[3.3. Phân tích quy trình tương tác 32](#_Toc385172550)

[3.3.1. Quy trình đọc truyện online 32](#_Toc385172551)

[3.3.2. Quy trình đọc truyện từ lịch sử 34](#_Toc385172552)

[3.3.3. Quy trình đọc truyện từ yêu thích 34](#_Toc385172553)

[3.3.4. Quy trình đọc truyện từ tải về 35](#_Toc385172554)

[3.4. Sơ đồ phân rã chức năng 36](#_Toc385172555)

[3.5. Xây dựng biểu đồ hoạt động 37](#_Toc385172556)

[3.5.1. Biểu đồ hoạt động tổng quát 37](#_Toc385172557)

[3.5.2. Biểu đồ hoạt động quy trình đọc truyện online 39](#_Toc385172559)

[3.5.3. Biểu đồ hoạt động quy trình đọc truyện từ lịch sử 40](#_Toc385172560)

[*3.5.4.* Biểu đồ hoạt động đọc truyện từ yêu thích 41](#_Toc385172561)

[3.5.5. Biểu đồ hoạt động đọc truyện từ tải về 42](#_Toc385172562)

[3.6. Xây dựng biểu đồ ca sử dụng 43](#_Toc385172563)

[3.6.1. Biểu đồ ca sử dụng tổng quát 43](#_Toc385172564)

[3.6.2. Biểu đồ chi tiết ca sử dụng đọc truyện online 43](#_Toc385172565)

[3.6.3. Biểu đồ chi tiết ca sử dụng quản lý lịch sử 45](#_Toc385172566)

[3.6.4. Biểu đồ chi tiết ca sử dụng quản lý yêu thích 47](#_Toc385172567)

[3.6.5. Biểu đồ chi tiết ca sử dụng tải về 48](#_Toc385172568)

[3.6.6. Biểu đồ chi tiết ca sử dụng tìm kiếm 50](#_Toc385172569)

[CHƯƠNG IV: THIẾT KẾ HỆ THỐNG 52](#_Toc385172570)

[4.1. Phân tích và thiết kế cơ sở dữ liệu 52](#_Toc385172572)

[4.1.1. Xác định thực thể và các thuộc tính 52](#_Toc385172573)

[4.1.2. Mô hình quan hệ 54](#_Toc385172574)

[4.2. Các giao diện cần có của chương trình 57](#_Toc385172577)

[4.3. Thiết kế thành phần hệ thống 58](#_Toc385172578)

[4.3.1. Module server 58](#_Toc385172579)

[4.3.2. Module client 59](#_Toc385172580)

[4.4. Triển khai ứng dụng vào thực tế 59](#_Toc385172581)

[4.4.1. Hướng dẫn cài đặt 59](#_Toc385172582)

[4.4.2. Giao diện ứng dụng 60](#_Toc385172585)

[KẾT LUẬN 62](#_Toc385172586)

[TÀI LIỆU THAM KHẢO 63](#_Toc385172588)

**MỤC LỤC HÌNH ẢNH**

[Hình 1: *Ứng dụng Mobiletruyen* 11](#_Toc385172048)

[*Hình 2: Ứng dụng Vuoncuoi* 12](#_Toc385172049)

[*Hình 3: Ứng dụng Babyteller* 12](#_Toc385172050)

[*Hình 4: Mô hình tổng quát các thành phần của hệ điều hành Android* 20](#_Toc385172051)

[*Hình 5: Vòng đời 1 Activity* 26](#_Toc385172052)

[*Hình 6: Truyền dữ liệu giữa 2 Activity* 27](#_Toc385172053)

[*Hình 7: Mô hình tương tác của phần mềm* 31](#_Toc385172054)

[*Hình 8: Sơ đồ phân rã chức năng phần mềm đọc truyện* 36](#_Toc385172055)

[*Hình 9: Biểu đồ hoạt động tổng quát* 37](#_Toc385172056)

[*Hình 10: Biểu đồ hoạt động đọc truyện online* 38](#_Toc385172057)

[*Hình 11: Biểu đồ đọc truyện từ lịch sử* 39](#_Toc385172058)

[*Hình 12: Biểu đồ hoạt động đọc truyện từ yêu thích* 40](#_Toc385172059)

[*Hình 13: Biểu đồ hoạt động đọc truyện từ tải về* 41](#_Toc385172060)

[*Hình 14: Ca sử dụng tổng quát* 42](#_Toc385172061)

[*Hình 15: Biểu đồ ca sử dụng đọc truyện online* 42](#_Toc385172062)

[*Hình 16: Biểu đồ ca sử dụng quản lý lịch sử* 44](#_Toc385172063)

[*Hình 17: Biểu đồ ca sử dụng quản lý yêu thích* 46](#_Toc385172064)

[*Hình 18: Biểu đồ chi tiết ca sử dụng tải về* 47](#_Toc385172065)

[*Hình 19: Biểu đồ chi tiết ca sử dụng tìm kiếm* 49](#_Toc385172066)

[*Hình 20: Mô hình thực thể liên kết ở client* 52](#_Toc385172067)

[*Hình 21: Mô hình thực thể liên kết ở server* 53](#_Toc385172068)

[*Hình 22: Mô hình quan hệ ở client* 55](#_Toc385172069)

[*Hình 23: Mô hình quan hệ ở server* 56](#_Toc385172070)

[*Hình 24: Biểu đồ di chuyển giữa các Activity* 57](#_Toc385172071)

[*Hình 25: Mô hình các gói của module server* 57](#_Toc385172072)

[*Hình 26: Mô hình các gói module client* 58](#_Toc385172073)

[*Hình 27: Giao diện danh sách truyện* 60](#_Toc385172074)

[*Hình 28: Giao diện chi tiết truyện* 60](#_Toc385172075)

[*Hình 29: Giao diện đọc truyện* 60](#_Toc385172076)

[*Hình 30: Giao diện menu* 60](#_Toc385172077)

# LỜI NÓI ĐẦU

Mạng điện thoại di động xuất hiện tại Việt Nam từ đầu những năm 1990 và theo thời gian số lượng các thuê bao cũng như các nhà cung cấp dịch vụ di động tại Việt Nam càng ngày càng tăng. Do nhu cầu trao dổi thông tin ngày càng tăng và nhu cầu sử dụng sản phẩm công nghệ cao nhiều tính năng, cấu hình cao, chất lượng tốt, kiểu dáng mẫu mã đẹp, phong phú nên nhà cung cấp phải luôn luôn cải thiện, nâng cao nhưng sản phẩm của mình. Với những ưu điểm về tính đa dụng và mức giá thành không còn khó tiếp cận, các thiết bị điện thoại thông minh (smartphone) đang dần dần chiếm lĩnh thị trường và có xu thế thay thế sự phổ biến của điện thoại phổ thông.

Cùng với sự phát triển của thị trường điện thoại di động là sự phát triển mạnh mẽ của xu hướng lập trình phẩn mềm ứng dụng cho các thiết bị di động. Người sử dụng cần ở chiếc điện thoại nhiều chức năng hơn là một thiết bị nghe gọi, nhắn tin đơn thuần. Với sức mạnh xử lý của mình, một chiếc điện thoại thông minh hay một chiếc máy tính bảng ngày nay hoàn toàn có thể trở thành một trợ lý cho các doanh nhân, thiết bị dẫn đường cho người du lịch, hay một cỗ máy giải trí cầm tay cao cấp… Phần mềm, ứng dụng cho điện thoại di động hiện nay rất đa dạng và phong phú trên các hệ điều hành di động cũng phát triển mạnh mẽ, đang thay đổi từng ngày. Bên cạnh các hệ điều hành Android, IOS, BlackBerry đã có từ lâu là sự gia nhập thêm hệ điều hành Windown phone của Microsoft, Tizen của Samsung… Nokia khai tử Symbian già cỗi tạo sự cạnh tranh giành thị phần hệ điều hành trên thiết bị di động. Tùy vào sở thích của mỗi người dùng, các tính năng, trải nghiệm của từng hệ điều hành sẽ có thể hấp dẫn hoặc không. Hơn nữa, mỗi OS có điểm mạnh, điểm hạn chế riêng. Đương nhiên, các hệ điều hành mới gia nhập làng OS smartphone sẽ có những lợi thế nhất định, nhưng kinh nghiệm và nền tảng vững chắc, sự ưu ái lâu nay của người dùng đối với một số tên tuổi đã khẳng định được thương hiệu cũng là thế mạnh đáng kể.

Trong vài năm trở lại đây, hệ điều hành Android ra đời với sự kế thừa ưu việt của các hệ điều hành ra đời trước và sự kết hợp của nhiều công nghệ tiên tiến nhất hiện nay, đã được nhà phát triển công nghệ rất nổi tiếng hiện nay là Google. Android đã nhanh chóng là đối thủ cạnh tranh mạnh mẽ với các hệ điều hành trước đó và đang là hiệu điều hành di động của tương lai và được nhiều người ưu chuộng nhất.

Những tháng gần đây, sự thành công của game **Plappy Bird** của Nguyễn Hà Đông đã ghi dấu ấn Việt Nam trên sân chơi ứng dụng quốc tế. Dẫu Plappy Bird đã bị gỡ bỏ nhưng việc một chàng trai Việt tạo nên một cơn sốt trên thế giới chắc hẳn là một cánh cửa mở với thị trường ứng dụng giải trí trên thiết bị di động của Việt Nam. Nó truyền cho em cảm hứng, sự kích thích, tò mò với việc viết ứng dụng trên thiết bị di động.

Chính vì những lý do trên, sinh viên đã chọn đề tài “**Xây dựng phần mềm đọc truyện trên thiết bị di động**” để làm đồ án tốt nghiệp của mình, với mong muốn tìm hiểu và khám phá một lĩnh vực đầy mới mẻ và thú vị: lập trình ứng dụng cho thiết bị thông minh cá nhân.

Với vốn kinh nghiệm ít ỏi và khả năng hạn chế của mình, việc thực hiện đồ án chắc chắn còn có nhiều thiếu sót. Sinh viên rất mong nhận được sự góp ý của các thầy cô!

# LỜI CẢM ƠN

Trước hết, em xin gửi lời tri ân tới toàn thể các thầy cô giáo trong **khoa** **Công nghệ thông tin** – **Trường Đại học Giao thông vận tải**, những người đã truyền hết tâm huyết và vốn kiến thức của mình cho chúng em trong suốt bốn năm học vừa qua tại mái trường này. Những kiến thức mà thầy cô đã truyền thụ cho chúng em là vô giá và sẽ là hành trang cho chúng em trong suốt đường đời!

Em xin gửi lời cảm ơn sâu sắc tới **Thầy giáo – Thạc sĩ Lưu Xuân Duẩn**, người đã tận tình giúp đỡ, dẫn dắt em trong suốt quá trình thực hiện đồ án. Những hướng dẫn đáng trân trọng của thầy đã giúp em hoàn thiện đồ án một cách tốt nhất có thể!

Em xin gửi lời cảm ơn tớihọc viện ITPlush nơi em theo học khóa lập trình Android đã đưa em tiếp cận với hệ điều hành Android, cảm ơn anh Phạm Quang Long đã cho em những lời khuyên quý báu, cảm ơn tất cả những người đã tạo điều kiện và sát cánh bên em từ lúc đề tài chỉ là một ý tưởng cho đến khi hoàn thành. Mỗi sự giúp đỡ, góp ý của mọi người là vô cùng quý giá đối với em. Không có những sự hỗ trợ quý báu đó, em đã không thể hoàn thành được đồ án này!

Cuối cùng, con xin giành những tình cảm yêu thương nhất tới bố mẹ và toàn thể gia đình, những người đã quan tâm, chăm sóc, nuôi nấng con nên người như ngày hôm nay. Gia đình là chỗ dựa vững chắc cho con trong suốt những năm tháng đã qua và cả cuộc đời trước mắt!

|  |  |
| --- | --- |
|  | *Hà Nội, ngày 07 tháng 04 năm 2014*  Sinh viên  **Nguyễn Hữu Chiều** |

# CHƯƠNG I: KHẢO SÁT THỰC TẾ

* 1. **Sơ qua về hệ điều hành Android**

Thị trường điện thoại thông minh hiện nay đang chứng kiến sự cạnh tranh của hai nền tảng thiết bị lớn là Iphonevới hệ điều hành ***iOS*** của hãng điện tử Mỹ *Apple* và điện thoại sử dụng hệ điều hành ***Android*** của Công ty *Google*, bên cạnh những nền tảng khác như ***Window phone*** của *Microsoft,* ***BlackBerry OS*** của *Research In Motion (RIM)…* cũng có chỗ đứng của riêng mình. **Iphone** đã sớm tạo được vị thế của mình trên thị trường nhờ những ưu điểm về thiết kế tinh tế, giao diện đơn giản mà bóng bẩy, xây dựng được trải nghiệm người dùng tốt. Tuy nhiên, những chiếc điện thoại chạy hệ điều hành **Android** với sức mạnh cấu hình đa dạng, chức năng phong phú và giá cả hợp lý, phù hợp nhu cầu của người sử dụng mới là những thiết bị có lượng người dùng phong phú, đa dạng nhất.

Android là hệ điều hành trên điện thoại di động (và hiện nay là cả trên một số đầu phát HD, HD Player, TV) phát triển bởi Google và dựa trên nền tảng Linux. Trước đây, Android được phát triển bởi công ty liên hợp Android ( sau đó được Google mua lại vào năm 2005).Các nhà phát triển viết ứng dụng cho Android dựa trên ngôn ngữ Java. Sự ra mắt của Android vào ngày 5 tháng 11 năm 2007 gắn với sự thành lập của liên minh thiết bị cầm tay mã nguồn mở, bao gồm 78 công ty phần cứng, phần mềm và viễn thông nhằm mục đính tạo nên một chuẩn mở cho điện thoại di động trong tương lai.

Mỗi hệ điều hành đứng trên góc độ người sử dụng cũng như người lập trình đều có những ưu điểm và khuyết điểm riêng, và việc lựa chọn một nền tảng để phát triển là tùy thuộc vào quyết định của người lập trình. Và để bắt đầu bước vào con đường của một lập trình viên, cá nhân em đã lựa chọn ***Android***, bởi nhiều lý do: ***Android*** là một hệ điều hành mã nguồn mở, nên khả năng tùy biến cao và có cộng đồng hỗ trợ đông đảo; chính sách hỗ trợ ứng dụng của Google đơn giản hơn sự cứng nhắc thái quá của Appe. Và một lý do nữa như đã nói ở trên, sự đa dạng các thiết bị chạy hệ điều hành ***Android*** cùng nhiều mức giá thành khác nhau sẽ giúp ***Android*** tiếp cận người dùng dễ dàng hơn, nhất là ở Việt Nam, nơi mức sống của người dân chưa cao.

* 1. **Thị trường ứng dụng trên thiết bị di động ở Việt Nam**

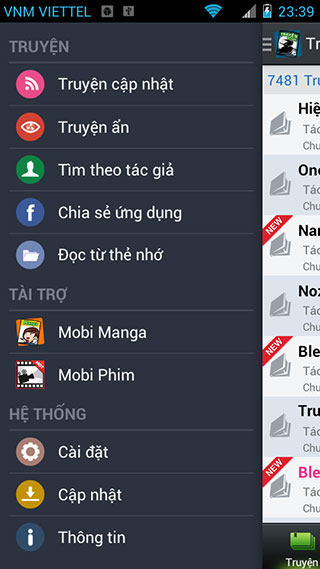
Vài năm trở lại đây, với sự phổ biến và độ phủ sóng rộng rãi của các mạng 3G và wifi, các thiết bị công nghệ di động ngày càng được ưa chuộng và sử dụng nhiều hơn. Tính đến cuối năm 2012, có khoảng 36.5 triệu thuê bao di động đang sở hữu điện thoại thông minh. Do tính cơ động cao mà các thiết bị di động này ngày càng tham gia mật thiết vào cuộc sống người dùng. Chỉ riêng trong năm 2013, đã có gần 80 tỷ lượt tải về các ứng dụng hỗ trợ trên Google Play và tại Việt Nam, kho ứng dụng của Samsung cán mức 27 triệu lượt tải. Con số này đã phần nào nói lên nhu cầu tìm kiếm lợi ích thực tế từ ứng dụng di động của người dùng đồng thời khẳng định vai trò “người bạn đồng hành” của các thiết bị di động trong đời sống hằng ngày của họ.

Trong thực tế đã có rất nhiều sản phẩm “made in Vietnam” có mặt trên các kho ứng dụng lớn, nhưng chưa có nhiều người dùng trong nước biết tới và khai thác. Chỉ đến khi thế giới bùng lên cơn sốt Flappy Bird, một ứng dụng trò chơi đóng mác nội địa mang về hàng chục ngàn đô la Mỹ mỗi ngày nhờ doanh thu quảng cáo, thì việc phát triển một ngành công nghiệp ứng dụng di động (mobile apps industry) mới được quan tâm nhiều hơn.

Đây chính là một thị trường rất tiềm năng cho những nhà phát triển phần mềm. Tuy vậy, các sản phẩm phần mềm trên điện thoại thông minh dành cho người Việt vẫn còn rất khiêm tốn cả về số lượng và chất lượng. Thêm nữa, thói quen sử dụng phần mềm bẻ khóa của người dùng Việt Nam cũng là một vấn đề gây nhiều khó khăn cho nhà phát triển. Hầu hết những nhà phát triển phần mềm mới chỉ đưa ra những phần mềm thăm dò mà chưa đưa mảng phát triển này vào chiến lược kinh doanh.

* 1. **Ứng dụng giải trí đọc truyện trên Android**

Hiện nay nhu cầu đọc truyện của giới trẻ cũng như các lứa tuổi khác đặc biệt là đối với các e nhỏ ham đọc truyện trên internet đã trở nên phổ biến. Tuy nhiên thì không thể lúc nào cũng ngồi cạnh máy tính để đọc, bởi vậy nó không chỉ phát triển trên website mà còn được phố biến trên phần mềm ứng dụng smartphone. Có rất nhiều chương trình đã được xây dựng với nhiều tính năng tiện ích thân thiện với người dùng. Hiện tại ứng dụng Mobitruyen đang là ứng dụng tốt nhất cho việc đọc truyện trên di động. Ứng dụng sử dụng giao diện ActionBar, menu Sliding, hỗ trợ các hiệu ứng lật trang rất sống động, chia sẻ hình ảnh qua facebook, chạy rất mượt. Tuy nhiên một điểm yếu nhỏ là chưa hỗ trợ đọc truyện audio. Dưới đây là một số hình ảnh của ứng dụng.

Hình 1: *Ứng dụng Mobiletruyen*

Ngoài ra còn một số ứng dụng khác như Vuoncuoi, BabyTeller… tuy nhiên mỗi phần mềm hướng tới mục đích, nhóm đối tượng khác nhau, nội dung truyện lấy từ các nguồn không rõ ràng, nội dung chưa phong phú…

|  |  |
| --- | --- |
| C:\Users\chieu_000\Desktop\Image 2.png  *Hình 2: Ứng dụng Vuoncuoi* | *Hình 3: Ứng dụng Babyteller* |

Từ những phân tích trên, trong thời gian thực tập tốt nghiệp, em đã sử dụng vốn kiến thức ít ỏi của mình tìm hiểu và phân tích bài toán “**Xây dựng ứng dụng giải trí đọc truyện trên thiết bị di động**”. Với đề tài này, trước hết em có thể tìm hiểu sâu về nên tảng Android, phục vụ cho công việc của bản thân sau này, và hơn thế, là hi vọng cung cấp cho thị trường ứng dụng di động một phần mềm nhỏ hữu ích.

# CHƯƠNG II: CÔNG NGHỆ SỬ DỤNG

* 1. **Ngôn ngữ lập trình Java**
     1. **Giới thiệu ngôn ngữ lập trình Java**

Java là một ngôn ngữ lập trình dạng lập trình hướng đối tượng (OOP). Khác với phần lớn ngôn ngữ lập trình thông thường, thay vì biên dịch mã nguồn thành mã máy hoặc thông dịch mã nguồn khi chạy, Java được thiết kế để biên dịch mã nguồn thành bytecode, bytecode sau đó sẽ được môi trường thực thi (runtime environment) chạy. Bằng cách này, một ứng dụng Java có thể chạy trên nhiều nền tảng khác nhau.

Cú pháp Java được vay mượn nhiều từ C & C++ nhưng có cú pháp hướng đối tượng đơn giản hơn và ít tính năng xử lý cấp thấp hơn. Do đó việc viết một chương trình bằng Java dễ hơn, đơn giản hơn, đỡ tốn công sửa lỗi hơn.

* + 1. **Những đặc điểm của ngôn ngữ Java**
* ***Máy ảo Java (JVM - Java Virtual Machine)***: Tất cả các chương trình muốn thực thi được thì phải được biên dịch ra mã máy. Mã máy của từng kiến trúc CPU của mỗi máy tính là khác nhau (tập lệnh mã máy của CPU Intel, CPU Solarix, CPU Macintosh … là khác nhau), vì vậy trước đây một chương trình sau khi được biên dịch xong chỉ có thể chạy được trên một kiến trúc CPU cụ thể nào đó. Đối với CPU Intel chúng ta có thể chạy các hệ điều hành như Microsoft Windows, Unix, Linux, OS/2, … Chương trình thực thi được trên Windows được biên dịch dưới dạng file có đuôi .EXE còn trên Linux thì được biên dịch dưới dạng file có đuôi .ELF, vì vậy trước đây một chương trình chạy được trên Windows muốn chạy được trên hệ điều hành khác như Linux chẳng hạn thì phải chỉnh sửa và biên dịch lại. Ngôn ngữ lập trình Java ra đời, nhờ vào máy ảo Java mà khó khăn nêu trên đã được khắc phục. Một chương trình viết bằng ngôn ngữ lập trình Java sẽ được biên dịch ra mã của máy ảo java (mã java bytecode). Sau đó máy ảo Java chịu trách nhiệm chuyển mã java bytecode thành mã máy tương ứng. Sun Microsystem chịu trách nhiệm phát triển các máy ảo Java chạy trên các hệ điều hành trên các kiến trúc CPU khác nhau.
* ***Đơn giản***: Java được phát triển trên nền tảng C++ nên khá quen thuộc với lập trình viên C++. Tuy nhiên Java lại đơn giản hơn nhờ loại bỏ các đặc điểm phức tạp, dễ gây nhầm lẫn của C++ như nạp chồng toán tử, thao tác con trỏ, đa kế thừa…
* ***Hướng đối tượng***: Hướng đối tượng trong Java tương tự như C++ nhưng Java là một ngôn ngữ lập trình hướng đối tượng hoàn toàn. Tất cả mọi thứ đề cập đến trong Java đều liên quan đến các đối tượng được định nghĩa trước, thậm chí hàm chính của một chương trình viết bằng Java (đó là hàm main) cũng phải đặt bên trong một lớp. Hướng đối tượng trong Java không có tính đa kế thừa (multi inheritance) như trong C++ mà thay vào đó Java đưa ra khái niệm interface để hỗ trợ tính đa kế thừa.
* ***Phân tán***: Java là ngôn ngữ thông dụng trong việc xây dựng các ứng dụng phân tán (ứng dụng mạng, web…). Java có thư viện mở rộng dùng cho lập trình phân tán như net,rmi,idl, CORBA… Các ứng dụng Java có thể truy xuất các đối tượng liên mạng dùng địa chỉ URL tương tự như truy xuất file cục bộ.
* ***Mạnh mẽ***: Java va có một cơ chế định kiểu mạnh, tường minh, kiểm tra lúc biên dịch và kiểm tra khi thông dịch trước khi thực thi nên giới hạn được lỗi; kiểm tra truy xuất phần tử của mảng, chuỗi lúc thực thi, kiểm tra ép kiểu run-time. Java cũng có mô hình quản lý bộ nhớ hiệu quả, tự động thu hồi bộ nhớ bằng trình gom rác – garbage collection.
* ***An toàn***: Java cung cấp môi trường thực thi có kiểm tra chặt chẽ. Không bao giờ giả định mã thực thi là an toàn. Có cơ chế kiểm tra an ninh hệ thống đa tầng.
* ***Khả chuyển:*** Chương trình ứng dụng viết bằng ngôn ngữ Java chỉ cần chạy được trên máy ảo Java là có thể chạy được trên bất kỳ máy tính, hệ điều hành nào có máy ảo Java, đúng như thông điệp của Java :“Viết một lần, chạy mọi nơi” (Write Once, Run Anywhere).
* ***Biên dịch và thông dịch***: Java là một ngôn ngữ lập trình vừa biên dịch vừa thông dịch. Chương trình nguồn viết bằng ngôn ngữ lập trình Java có đuôi \*.java đầu tiên được biên dịch thành tập tin bytecode có đuôi \*.class và sau đó sẽ được trình thông dịch thông dịch thành mã máy.
* ***Độc lập nền***: Một chương trình viết bằng ngôn ngữ Java có thể chạy trên nhiều máy tính có hệ điều hành khác nhau (Windows, Unix, Linux, …) miễn sao ở đó có cài đặt máy ảo java (Java Virtual Machine). Viết một lần chạy mọi nơi (write once run anywhere).
* ***Đa nhiệm - đa luồng:*** Java hỗ trợ lập trình đa nhiệm, đa luồng cho phép nhiều tiến trình, tiểu trình có thể chạy song song cùng một thời điểm và tương tác với nhau.
* ***Hỗ trợ mạnh cho việc phát triển ứng dụng:*** Công nghệ Java phát triển mạnh mẽ nhờ vào Sun Microsystem và nay là Oracle đã cung cấp nhiều công cụ, thư viện lập trình phong phú hỗ trợ cho việc phát triển nhiều loại hình ứng dụng khác nhau cụ thể như: J2SE (Java 2 Standard Edition) hỗ trợ phát triển những ứng dụng đơn, ứng dụng client-server; J2EE (Java 2 Enterprise Edition) hỗ trợ phát triển các ứng dụng thương mại, J2ME (Java 2 Micro Edition) hỗ trợ phát triển các ứng dụng trên các thiết bị di động, không dây, …
  1. **Hệ quản trị cơ sở dữ liệu SQLite**
     1. **Giới thiệu SQLite**

Mỗi ứng dụng đều sử dụng dữ liệu, dữ liệu có thể đơn giản hay đôi khi là cả 1 cấu trúc. Trong Android thì hệ cơ sở dữ liệu được sử dụng là SQLite Database, đây là hệ thống mã nguồn mở được sử dụng rộng rãi trong các ứng dụng ( Mozilla Firefox sử dụng SQLite để lưu trữ các dữ liệu về cấu hình, iPhone cũng sử dụng cơ sở dữ liệu là SQLite).

Được giới thiệu từ năm 2000 để giúp các ứng dụng quản lý dữ liệu của mình thuận tiện hơn, SQLite là một bộ thư viện dùng trong lập trình để hiện thực một SQL Database Engine có khả năng tự tổ chức quản lý dữ liệu, không cần server, không cần cấu hình mà vẫn hỗ trợ đầy đủ các tính năng quản lý giao tác. SQLite hiện đang là SQL Database Engine mã nguồn mở theo mô hình dữ liệu quan hệ đang được sử dụng nhiều nhất trên thế giới do tính cơ động cao, dễ sử dụng, gọn nhẹ, hiệu quả và tin cậy.

### Một số đặc điểm chính của SQLite

SQLite có những ưu điểm sau:

* **Đảm bảo đầy đủ 4 đặc tính ACID của các giao tác:** tuy cực kỳ nhỏ gọn nhưng các giao tác làm việc với SQLite được đảm bảo đầy đủ 4 tính chất cơ bản của giao tác là tính nguyên tổ (Atomic), tính nhất quán (Consistent), tính cô lập (Isolated), và tính bền vững (Durable) thậm chí sau khi hệ thống bị crash hoặc gặp các sự cố về nguồn điện.
* **Không cần cấu hình:** với SQLite chúng ta sẽ không cần phải cài đặt hay quản trị nó.
* **Đặc tính:** SQLite có gần như toàn bộ các đặc tính phổ biến của SQL theo chuẩn SQL92. Tuy nhiên còn một số tính năng như RIGHT OUTER JOIN, FULL OUTER JOIN, FOR EACH STATEMENT TRIGGER,... chưa được hỗ trợ.
* **Lưu trữ:** Toàn bộ Database được lưu trữ trong 1 tập tin trên đĩa duy nhất. Hỗ trợ CSDL lên tới hằng TetraByte, trong đó các kiểu dữ liệu chuỗi và blobs lên tới hằng GigaByte.
* **Bộ thư viện quản lý rất nhỏ, gọn:** dưới 500 KB cho bản đầy đủ tính năng, và có thể ít hơn nếu loại bớt một số đặc tính. Cụ thể, kích thước phiên bản SQLite 3.7.17 (Phiên bản hiện tại, mới nhất của SQLite) ít hơn 350 KB trên hệ điều hành 32 bit và ít hơn 409 KB trên các hệ thống 64 bit.
* **Truy xuất:** Hầu hết các thao tác trên dữ liệu thông thường đều chạy nhanh hơn các Database Engine theo kiểu Client/Server phổ biến khác.
* **Đơn giản và dễ sử dụng** với bộ API tương ứng
* **Mã nguồn** mở được viết bằng ANSI-C, được comment rất tốt và có thể tái sử dụng với bất cứ mục đích gì.
* **SQLite** có thể được tải về và nhúng vào các dự án khác nhau dưới hình thức một Single ANSI-C source-code file.
* **Tự tổ chức lưu trữ (**self-contained) mà không phải phụ thuộc vào các thư viện bên ngoài. Đây là một đặc điểm khá quan trọng khiến SQLite trở thành CSDL phù hợp để nhúng vào các thiết bị di động hoặc tích hợp vào các ứng dụng muốn chạy mà không cần phải điều chỉnh cấu hình hệ thống.
* **Trong bộ thư viện SQLite** có sẵn 1 client đơn giản giao tiếp theo chế độ dòng lệnh (Command-Line Interface – CLI) tên là sqlite3 (trên môi trường Windows là tiện ích sqlite3.exe) để có thể được dùng để quản trị CSDL SQLite.
  + 1. **Một số hạn chế của SQLite**

Ngoài các ưu điểm được ca tụng nhiều trên website của SQLite, nếu đem so với các CSDL mạnh khác như Oracle Dabase, SQL Server,... thì SQLite sẽ bọc lộ một số yếu điểm sau:

* **Tính đồng thời**: SQLite sử dụng cơ chế khóa coarse-gained locking có thể hỗ trợ nhiều người đọc dữ liệu, nhưng chỉ có 1 người có thể ghi dữ liệu một lúc mà thôi.
* **Nối kết mạng:**mặc dù SQLite có thể được chia sẻ thông qua các network file systems, tuy nhiên độ trễ giữa các hệ thống tập tin này sẽ ảnh hưởng nghiêm trọng tới hiệu suất thực hiện của SQLite. Tệ hại hơn nữa là, các lỗ hỏng trong việc triển khai các hệ thống tập tin trong môi trường mạng có thể gây ra nhiều lỗi do các tập tin có thể bị mở và điểu chỉnh từ xa.
* **Phù hợp với các ứng dụng có qui mô dữ liệu nhỏ:**trong thời đại bùng nổ thông tin như hiện nay, SQLite không phải là lựa chọn lý tưởng để đáp ứng các nhu cầu xử lý trên 1 khối lượng dữ liệu lớn, phát sinh liên tục.
* **Hỗ trợ tiếng việt:** Một số phiên bản cũ còn bị lỗi font tiếng việt.
  + 1. **Ứng dụng cơ sở dữ liệu của SQLite**

Do đặc điểm được thiết kế của SQLite ngay từ đầu, nên CSDL nhúng này phù hợp với một số tình huống sử dụng như sau:

* **SQLite có thể sử dụng như định dạng tập tin thích hợp cho các ứng dụng:** trước đây các tập tin cấu hình có thể được lưu dưới dạng file .ini, rồi sau đó là .xml,... tuy nhiên chúng ta sẽ cần phải viết hoặc phụ thuộc vào việc sử dụng các parser phù hợp. Sử dụng SQLite không những giúp chúng ta khắc phục được yếu điểm này mà chương trình của chúng ta lại còn có thể chạy trên nhiều nền tảng khác nhau.
* **SQLite có thể sử dụng làm CSDL cho các thiết bị điện tử.** Quả thật SQLite đang là sự lựa chọn phổ biến về Database Engine cho các thiết bị điện tử như PDA, điện thoại di động, MP3 Player, các hộp set-top box và nhiều loại thiết bị điện tử khác. SQLite nhỏ, sử dụng bộ nhớ, không gian lưu trữ và băng thông đĩa hiệu quả, tin cậy mà không cần phải quản trị bởi các quản trị viên chuyên nghiệp. Nếu bạn đang viết ứng dụng trên các thiết bị chạy iOS như iPhone, iPad,... hay các thiết bị chạy Android hoặc Windows Mobile,... thì SQLite chính là CSDL phù hợp cho các thiết bị di động này.
* **SQLite có thể sử dụng như CSDL cho các website.** Do SQLite không cần phải cấu hình và dữ liệu được lưu trữ thành các tập tin trên đĩa thật sự nên nó đang trở thành lựa chọn phổ biến cho các website vừa và nhỏ.
* **SQLite có thể thay thế cho các CSDL** **quan hệ tầm doanh nghiệp** trong một số tình huống như kiểm thử, demo minh họa,... do SQLite chạy khá nhanh và không cần cài đặt
  1. **Nền tảng phát triển ứng dụng Android**
     1. **Lịch sử phát triển**

Android là 1 nền tảng phần mềm dựa trên mã nguồn mở Linux OS ( Kernel 2.6) cho máy di động, máy tính bảng và những phần mềm trung gian (middleware). Nó không đơn thuần là một hệ điều hành, một công cụ lập trình hay một phần mềm trung gian mà nó bao gồm tất cả.

Ban đầu, Android là hệ điều hành cho các thiết bị cầm tay dựa trên lõi Linux do công ty Android Inc. (California, Mỹ thiết kế. Công ty này sau đó được google mua lại vào năm 2005 và bắt đầu xây dựng Android Flatform.

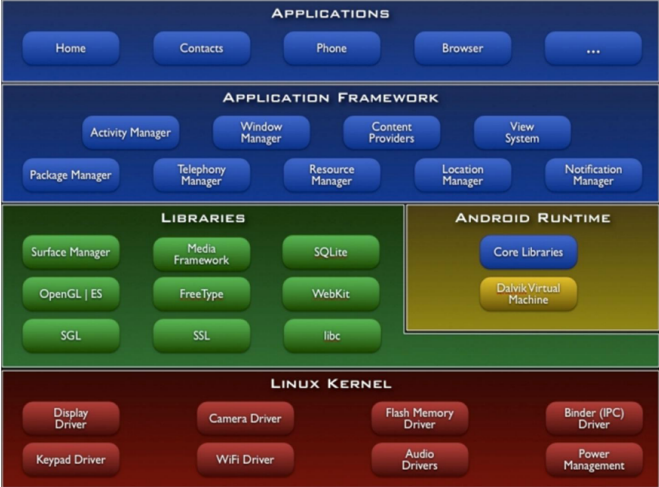
Và sau tiếp, vào cuối năm 2007, thuộc về Liên minh thiết bị cầm tay mã nguồn mở (Open Handset Alliance). Mục tiêu của liên minh này là nhanh chóng đổi mới để đáp ứng tốt hơn cho nhu cầu người tiêu dùng và kết quả đầu tiên của nó chính là nền tảng Android. Android dược thiết kể để phục vụ nhu cầu của các nhà sản xuất, các nhà khai thác và các lập trình viên thiết bị cầm tay.

Phiên bản SDK lần đầu tiên phát hành vào tháng 11 năm 2007, hãng T-Mobile cũng công bố chiếc điện thoại Android đầu tiên đó là chiếc T-Mobile G1, chiếc smartphone đầu tiên dựa trên nền tảngAndroid. Một vài ngày sau đó google lại tiếp tục công bố sự ra mắt phiên bản Android SDK release Candidate 1.0. Trong tháng 10 năm 2008, google được cấp giấy phép mã nguồn mở cho Android Platform.

Khi Android được phát hành thì một trong số các mục tiêu trong kiến trúc của nó là cho phép các ứng dụng có thể tương tác được với nhau và có thể sử dụng lại các thành phần từ những ứng dụng khác. Việc tái sử dụng không chỉ được ứng dụng cho các dịch vụ mà nó còn được ứng dụng cho cả các thành phần dữ liệu và giao diện người dùng.

Vào cuối năm 2008, Google cho phát hành một thiết bị cầm tay được gọi là Android Dev Phone 1 có thể chạy được tất cả các ứng dụng Android mà không bị ràng buộc vào các nhà cung cấp mạng điện thoại di động. Mục tiêu của thiết bị này là cho phép các nhà phát triển thực hiện các cuộc thí nghiệm trên một thiết bị thực có thể chạy hệ điều hành Android mà không phải ký một bản hợp đồng nào. Vào khoảng cùng thời gian đó thì Google cũng cho phát hành một phiên bản vá lỗi 1.1 của hệ điều hành này. Ở cả hai phiên bản 1.0 và 1.1 Android chưa hỗ trợ soft-keyboard mà đòi hỏi các thiết bị phải sự dụng bàn phím vật lý. Android cố định vấn đề này bằng cách phát hành SDK 1.5 vào tháng tư năm 2009, cùng với một số tính năng khác. Chẳng hạn như nâng coa khả năng ghi âm truyền thông, vật dụng, và các live folder.

* + 1. **Kiến trúc Android**



*Hình 4: Mô hình tổng quát các thành phần của hệ điều hành Android*

* **Linux Kernel**

Hệ điều hành Android được phát triển dựa trên hạt nhân linux, cụ thể là hạt nhân linux phiên bản 2.6, điều đó được thể hiện ở lớp dưới cùng này. Tất cả mọi hoạt động của điện thoại muốn thi hành được thì đều được thực hiện ở mức cấp thấp ở lớp này bao gồm quản lý bộ nhớ (memory management), giao tiếp với phần cứng (driver model), thực hiện bảo mật ( security), quản lý tiến trình ( process).

Tuy được phát triển dựa vào nhân linux nhưng thực ra nhân linux đã được nâng cấp và sửa đổi rất nhiều để phù hợp với tính chất của những thiết bị cầm tay như hạn chế về bộ vi xử lý, kích thước màn hình , nhu cầu kết nối mạng không dây..

Các thành phần của linux

* **Display Driver:** Điều khiển việc hiển thị lên màn hình cũng như thu nhận những điều khiển của người dùng lên màn hình ( di chuyển, cảm ứng..)
* **Camera Driver:** Điều khiển hoạt động của camera , nhận luông dữ liệu từ camera trả về.
* **Bluetooth Driver:** Điều khiển thiết bị phát và thu sóng Bluetooth.
* **USB driver:** Quản ;ý hoạt động của cổng giao tiếp USB
* **Keypad driver:** chịu trách nhiệm về việc thu phát sóng wifi.
* **Audio driver:** điều khiển các bộ thi phát âm thanh, giải mã các tín hiệu dạnh audio thành tín hiệu số và ngược lại.
* **Power Madagement:** giám sát việc tiêu thụ điện năng.
* **M-System driver:** quản lý việc đọc,ghi… lên các tiheets bị nhớ như thẻ SD =, flash.
* **Binder IPC driver:** chịu trác nhiệm về việc kết nối và liên lạc với mạng vô tuyến như CDMA, GSM, 3G, 4G, E để đảm bảo những chức năng truyền thông được thực hiện.
* **Library và Android Runtime**

Phần này có 2 thành phần là phần Library và Android Runtime

* **Library:**

Thành phần này cón nhiều thư viện được viết bằng C/C ++ để các phần mền có thể sử dụng các thư việ đó được tập hợp thành một số nhóm như :

* Thư viện hệ thống : thư viện dựa trên chuẩn C, được sử dụng chỉ bởi hệ điều hành
* Thư viện Media ( Media Libraries): có nhiều codec để hỗ trợ việc phát và ghi các loại định dạnh âm thanhm hình ảnh, video thông dụng.
* Thư viện web ( LibWebCore): đây là thành phần để xem nội dung trên web được sử dụng để xây dựng phần mền duyệt wev(Android Browse) cũng như để các ứng dụng khác có thể nhúng vào. Nó cực kỳ mạnh, hỗ trợ được nhiều công nghệ mạnh mẽ như HTML5, JavaScript, CSS,DOM, AJAX..
* Surface Manage: quản lý hiển thị 2D và 3D
* SGL công nghệ 2D
* SQLite : quản lý database của ứng dụng.
* Free Type: dùng cho hiển thị ảnh, font.
* SSL( Secure Soket Layer): tầng thiết bị giao dịch an toàn trên 2 ứng dụng trên 1 cổng(socket).
* Open GL/EL: hỗ trợ thư viện tùy chọn 3D tĩnh và động.
* **Android runtime:**

Phần này chứa các thư viện mà một chương trình viết bằng ngôn ngữ Java có thể hoạt động. phần này có 2 bộ phận tương tự như mô hình chạy Java trên máy tính thường. thứ nhất là các thư việ lõi( Cỏe Library), chứa cá lớp như JAVA IO, Collectinons, File Access. Thứ hai một máy ảo java( Dalvik Virtual Machine).

Mặc dù cũng được viết từ ngôn ngữ Java nhưng một ứng dụng Java của hệ điều hành Android không chạy bằng JRE của Sun(này là Oracle ) (JVM) mà chạy bằng máy aoe Dalvik( máy ảo dalvik được đặt tên sau khi 1 nhà phát triển phần mền của google tạo ta nó sau khi đi thăm một ngôi làng ở nơi tổ tiên ông sinh sống) do Google phát triển. Máy aoe davik thực thi cái tập tin davik (dex), khác với Java là chuyển sang dạng bytecode. Định dạnh này được tối ưu hóa cho bộ nhớ tối thiểu.

* **Famework**

Bằng cách cung cấp một nền tảng phát triển mở,Android cung cấp cho các nhà phát triển khả năng xây dựng các ứng dụng cực kỳ phong phú và sáng tạo. nhà phát triển được tự do tần dụng các thiết bị phần cứng , thông tin địa điểm truy cập, các dịch vụ chạy nền, thiết lập hệ thống báo động, thêm các thông báo để các thah trạng thái, và nhiều nhiều hơn nữa.

Nhà phát triển có thể truy cập vào các API cùng một khuôn khổ được sử dụng bởi các ứng dụng lõi. Các kiến trúc ứng dụng được thiết kế đơn giản hóa việc sử dụng lại các thành phần. cơ bản tất cả các ứng dụng là một bộ phận các dịch vụ và các hệ thống bao gồm :

* View Manage: một tập hợp rất nhiều các View có khả năng kế thừa lẫn nhau dùng để thiết kế phần giao diện ứng dụng như : gridview, tableview, linearlayout..
* Activity Manager: quản lý các chu kỳ sống của một ứng dụng cũng như cung cấp công cụ điều khiển các Activity.
* Telephony manager: cung cấp công cụ để thực hiện việc liên lạc như gọi điện thoại..
* Location manager: quản lý việc xây dựng và hiển thị các giao diện người dùng cũng như tổ chức quản lý các giao diện giữa các ứng dụng.
* Notication Manager: quản lý việc hiện thị các thông báo ( như báo có tin nhắn, có e-mail mới)
* Resource manager: quản lý tài nguyên tĩnh của các ứng dụng bao foomf các file hình ảnh, âm thanh, layout, string.(những thành phần không được viết bởi ngôn ngữ lập trình).
* Content Provider: cho phép các ứng dụng có thể truy xuất dữ liệu từ các ứng dụng khác chẳng hạn như Contacts hoặc là chia sẻ dữ liệu giữa các ứng đó.
* **Application**

Đây là lớp ứng dụng giao tiếp với người dùng, bao gồm các ứng dụng như: các ứng dụng cơ bản được cài đặt đi liền với hệ điều hành gọi là điện thoại ( phone), quản lý danh bạ( Contacts), duyệt web( brower), nhắn tin (sms), lịch làm việc( calendar), đọc e-mail (email- client), bản đồ ( map), quay phim chụp ảnh( camera)..các ứng dụng được cài thêm như các phần mền chứng khoán( stock), các trò chơi, từ điển các widget các ứng dụng của bên thứ 3..

Các ứng dụng viết sẽ có những đặc tính sau:

* Viết bằng Java, phần mở rộng là .apk.
* Khi mỗi ứng dụng được chạy, nó có một phiên bản Virtual Machine được dựng lên để phục vụ cho nó. Nó có thể là một Active Program : Chương trình có giao diện với người sử dụng hoặc là một background: chương trình chạy nền hay là dịch vụ.
* Android là hệ điều hành đa nhiệm, điều đó có nghĩa là trong cùng một thời điểm, có thể có nhiều chương trình cùng chạy một lúc, tuy nhiên , với mỗi ứng dụng thì có duy nhất một thực thể( instance) được phép chạy mà thôi. Điều đó có tác dụng hạn chế sự lạm dụng tài nguyên, giúp hệ thống hoạt động tốt hơn. Các ứng dụng được phân quyền và tự hệ thống sẽ kill nếu mức tiên thấp.
* Các ứng dụng được gán sỗ ID của người sử dụng nhằm phân định quyền hạn khi sử dụng tài nguyên, cấu hình phần cứng và hệ thống.
* Android là một hệ điều hành có tính mở, khác với nhiều hệ điều hành di động khác, Android cho phép một ứng dụng của bên thứ ba được phép chạy nền. các ứng dụng đó chỉ một hạn chế nhở đó là nó không được phép sử dụng quá 5-10% công suất CPU, điều đó nhằm để tránh độc quyền việc sử dụng CPU.
  + 1. **Các thành phần của Android**
* **Activity:**

Một activity thể hiện một giao diện đồ họa người dùng. Ví dụ một activity có thể biểu diễn một danh sách các menu item để người dùng có thể chọn và có thể hiện thị ảnh cùng với tiêu đề. Một ứng dụng gửi tin nhắn văn bản có thể có một hoạt động là hiển thị một danh sách các liên hệ để gửi tin nhắn tới, hoạt động thứ hai là viết tin nhắn tới liên hệ được chọn, các hoạt động khác nữa là xem lại tin nhắn cũ hay thay đổi cài đặt. Mặc dù chúng làm việc cùng nhau để tạo thành một giao diện người dùng, mỗi activity độc lập với những cái khác. Mỗi activity là một lớp con của lớp cơ sở Activity.

Mỗi ứng dụng có thể gồm chỉ một activity hay nhiều activity. Thông thường một trong số các activity được đánh dấu như là activity đầu tiên phải được trình diễn tới người dùng khi ứng dụng được khởi động. chuyển từ một activity sang activity khác được hoàn thành bằng cách cho activity hiện thời khởi động activity kế tiếp.

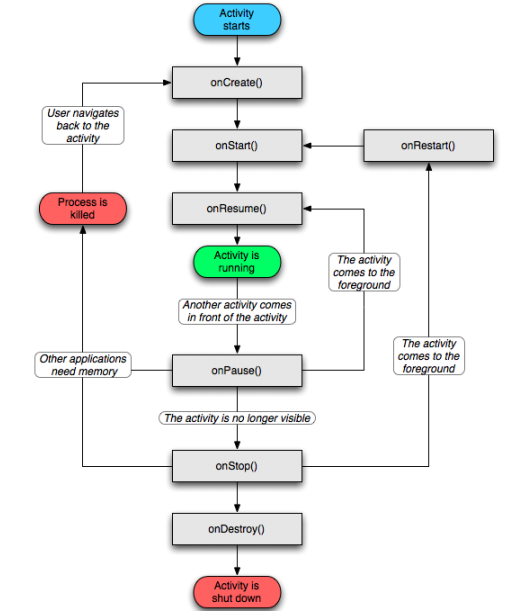
Mỗi activity được đưa ra một cửa sổ mặc định để vẽ vào. Thông thường, cửa sổ sẽ lấp đầy màn hình, nhưng nó có thể nhỏ hơn màn hình và nằm trên các cửa sổ khác. Một activity có thể sử dụng cửa sổ bổ sung- ví dụ như một pop-up dialog gọi cho một đáp ứng người dùng ở giữa của activity, hay một cửa sổ biểu diễn những cho người dùng với thông tin quan trọng khi họ chọn một item cụ thể trên màn hình.

Nội dung trực quan của cửa sổ được cung cấp bởi một cây phân cấp các đối tượng view dẫn xuất từ lớp view. Mỗi view điều khiển một khoảng hình chữ nhật cụ thể bên cửa sổ. View cha chứa và tổ chức bố cục các view con. Các view lá vẽ trong hình chữ nhật mà chúng điều khiển và đáp ứng lại các hành động người dùng trực tiếp ở khoảng trống này. Do đó, các view là nơi mà các tương tác của activity với người dùng diễn ra. Ví dụ một view có thể hiện một hình ảnh nhỏ và khơi tạo một hoạt động khi người dùng nhấn vào hình ảnh đó. Android có một số view đã xây dựng sẵn mà bạn có thể sử dụng – gồm các buttons, text fields, scroll bars, menu item, check boxes…

Một cây phân cấp view được đặt trong một cửa sổ của actitity bằng phương thức Activity.setcontenview. content view là đối thượng View ở gốc của cây phân cấp.

Class cơ sở Activity định nghĩa một loạt các sự kiện mà điều chỉnh vòng đời của một hoạt động. Class Activity định nghĩa các sự kiện sau đây:

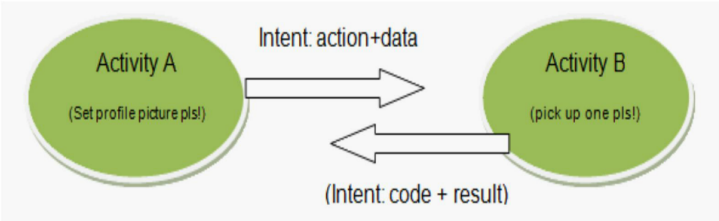
* onCreate(): Được gọi khi hoạt động tạo ra lần đầu tiên.
* onStart(): Được gọi khi hoạt động trở nên hữu hình so với người dùng.
* onResume(): Được gọi khi hoạt động bắt đầu tương tác với người sử dụng.
* onPause(): Được gọi để dừng các hoạt động hiện tại và nối lại các hoạt động trước đó.
* onStop(): Được gọi khi hoạt động không còn hiển thị với người dùng.
* onDestroy():Được gọi trước khi hoạt động bị phá hủy bởi hệ thống ( bằng tay hoặc bằng hệ thống để bảo tồn bộ nhớ).
* onRwRestart():Được gọi khi hệ thống đã được dừng lại và khởi động lại một lần nữa.



*Hình 5: Vòng đời 1 Activity*

* **Intent:**

Là một cấu trúc dữ liệu mô tả thách thức , đối tượng thực hiện một Activity. Là cầu nối giữa các Activity: ứng dụng androi thường bao gồm nhiều Activity, mỗi Activity hoạt động độc lập với nhau và thực hiện những công việc khác nhau. Intent chính là người đưa thư, giúp các Activity có thể triệu gọi cũng như truyền các dữ liệu cần thiết tới một Activity khác. Điều này cũng giống như việc di chuyển qua lại giữa các Forms trong lập trình Windows Form.



*Hình 6: Truyền dữ liệu giữa 2 Activity*

* **Sevice**

Một sevice không có giao diện trực quan, nó chạy trong một khoảng thời gian không gian xác định. Ví dụ một serice có thể chơi nhạc nền, hay nó nạp dữ liệu trên mạng hay tính toán cái gì đó và cung cấp kết quả cho Activity cần đến nó. Mỗi service mở rộng từ lớp cơ sở Service.

Trong khi kết nối , bạn có thể giao tiếp với dịch vụ thông qua giao diện mà service tung ra. Giống như các Activity và cách thành phần các nhau, service chạy trong thread chính của tiến trình ứng dụng, chúng thường được tạo ra các thaed khác cho nhiệm vụ hao tốn thời gian

* **Broadcast Reveie**

Một Broadcast Reveier là một thành phần không làm gì ngoài việc nhận và đáp lại các thông báo broadcast. Nhiều Broadcast khởi đầu trong mã hệ thống- ví dụ như thông báo mũi giờ thay đổi, pin yếu,ảnh đã được chụp, hay người dùng đã thay đổi ngôn ngữ… các ứng dụng có thể tạo ra các Broadcast, chẳng hạn để ứng dụng khác biết được một số dữ liệu đã được tải về thiết bị và sẵn sàng cho việc sử dụng chúng.

Một ứng dụng có thể có một số Broadcast Receiver để đáp lại bất cứ thông báo nào mà nó cho là quan trọng. tất cả các receiver mở rộng từ lớp cơ sở Broadcast Receiver.

Broadcast Receiver không hiển thị một giao diện người dùng. Tuy nhiên chúng có thể bắt đầu một activity để đáp lại thông tin mà chúng nhận, hay chúng có thể sử dụng NotificationManager để cảnh báo người dùng. Notification có thể lấy sự chú ý của người dùng bằng nhiều cách,lóe sáng đèn sau, rung,tạo ra âm thanh,… Chúng thường lấy một biểu thượng bền vững trong thanh trạng thái, cái mà người dùng có thể mở lấy thông điệp.

* **Content Provider**

Một content provider tạo ra một cụ thể các dữ liệu của ứng dụng khả dụng cho các ứng dụng khách. Dữ liệu có thể được lưu trữ trong hệ thống file, trong một cơ sở dữ liệu SQLite, hay trong một cách khách nào đó. Conten provider mở rộng lớp cơ sở content provider để cài đặt một tập các chuẩn các phương thức cho phép các ứng dụng khác đạt được và lưu trữ dữ liệu của kiểu mà nó điều khiển. tuy nhiên, các ứng dụng không gọi trực tiếp các phương thức này, chúng sử dụng một đối tượng content Resoler và gọi các phương thức của nó. Một Content resoler có thể nói chuyện với bất cứ content provider nào,chúng cộng tác với provider để quản lý giao tiếp liên tiến trình.

* **Manifest flie**

Trước khi Android có thể khởi động một thành phần ứng dụng nó phải biết rằng thành phần đó đã tồn tại. Vì vậy ứng dụng khai báo những thành phần của mình trong một manifest file được gắn vào Android package, file .apk này cũng chứa mã của ứng dụng tài nguyên.

Manifest file là một file XML có cấu trúc thường được đặt tên là AndroidManifest.xml cho mọi ứng dụng. Nó thực hiện một số bổ sung để khai báo các thành phần của ứng dụng, như là nêu tên các thư viện ứng dụng cần đến, và xác định các quyền hạn được phân cấp như quyền truy cập internet, SDCard hay vấn đề gửi tin nhắn, cuộc gọi…

* + 1. **Bộ công cụ phát triển Eclipse for Android**

Đối với các nhà phát triển, Android SDK (Bộ công cụ phát triển phần mềm của Android) cung cấp một tập hợp các công cụ phong phú, bao gồm trình gỡ rối, các thư viện, trình mô phỏng thiết bị cầm tay, tài liệu, mã mẫu và các hướng dẫn. Các ứng dụng Android có thể dễ dàng được phát triển khi sử dụng Eclipse (nền tảng phát triển chính thức của Android) với sự trợ giúp của một trình cắm thêm được gọi là ADT (Android Development Tools – Các công cụ phát triển Android). Điều này giúp tận dụng các tính năng phong phú của Eclipse, như là hỗ trợ nội dung, tìm kiếm Java, các tài nguyên mở, tích hợp JUnit và các khung nhìn và các ảnh phối cảnh khác nhau để phát triển một ứng dụng Android. Một mảng rộng rãi các widget (tiện ích), tương tự như các widget swing của Java, tạo điều kiện thuận lợi trong việc tạo một giao diện người dùng phong phú cho các ứng dụng. Một công cụ Javadoc chi tiết giúp cho quá trình phát triển này khá dễ dàng.

Đối với lập trình viên, đây là phần mềm được sử dụng nhiều nhất khi phát triển android, nó cung cấp giao diện thân thiện, có thể thêm các thư viện ngoài vào, cho phép giả lập máy ảo Android (ADV) ngay trên máy tính để test code. Một khi AVD này được khởi chạy, bạn có thể thấy nó mang lại dáng vẻ và cảm nhận đầy đủ của một điện thoại di động dựa trên Android thực, có đủ bàn phím và hỗ trợ đa chạm. Nó cũng có thể được sử dụng trong một loạt các cấu hình để thử nghiệm ứng dụng của bạn, như là chế độ hướng trang nằm ngang/thẳng đứng, cường độ mạng và mạng chuyển vùng , v.v. Tất cả những tùy chọn này có thể được cấu hình khi sử dụng trình quản lý AVD. AVD hoàn toàn độc lập để mô phỏng các thiết bị khác nhau sẵn có. Bạn có thể tạo các AVD khác nhau theo các cấu hình khác nhau và thử nghiệm ứng dụng của bạn trên từng AVD để chắc chắn rằng nó tương thích với mọi loại thiết bị.

Bộ công cụ em sử dụng trong đồ án là phiên bản 22.0.1 mới nhất. Mọi người có thể tải bộ phát triển này ở địa chỉ: <http://code.google.com/android/download.html> .

# CHƯƠNG III: PHÂN TÍCH BÀI TOÁN

* 1. **Mục đích của đề tài**

Việc tìm hiểu và phát triển đề tài “Xây dựng ứng giải trí đọc truyện trên thiết bị di động” mang ba mục đích chính:

+ Xây dựng thành công một máy chủ cung cấp các truyện đọc online, cho phép tải về.

+ Xây dựng thành công một ứng dụng đọc truyện trên nền tảng Android.

Bên cạnh đó là những mục tiêu mà cá nhân em muốn nhận được sau khi hoàn thành dự án:

+ Tìm hiểu được kỹ thuật lập trình máy chủ trên ngôn ngữ php.

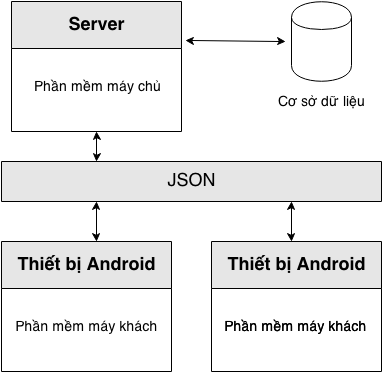
+ Tìm hiểu được kỹ thuật lập trình game trên nền tảng Android.

* 1. **Mô hình tương tác của phần mềm**

Dự án được chia làm 2 modun: modun server và modun trên thiết bị android, 2 modun này giao tiếp với nhau qua JSON. Server trả dữ liệu về qua Json, modunle trên thiết bị android sẽ đọc dữ liệu trả về, lấy ra các dữ liệu cần thiết.

Server được viết bằng ngôn ngữ PHP, sẽ chạy trên một máy chủ Xampp đảm nhận việc vận hành máy chủ trong phần đọc truyện onlie, đồng thời kết nối tới một cơ sở dữ liệu để lưu trữ và cập nhật danh sách các truyện.

Module máy khách chạy trên thiết bị Android sẽ đảm nhiệm việc tương tác trực tiếp với người dùng và kết nối tới server, được chia thành hai phần cho phù hợp với việc đọc truyện online và đọc truyện offline. Trong phần đọc truyện onlie, phần mềm sẽ phải kết nối tới máy chủ để lấy về danh sách các truyện. Trong phần đọc truyện offline, người dùng sẽ đọc truyện trực tiếp trên thiết bị mà không cần phải kết nối tới server, truyện này đã được tải về trước.

****

*Hình 7: Mô hình tương tác của phần mềm*

* 1. **Phân tích quy trình tương tác**
     1. **Quy trình đọc truyện online**

Đây là quy trình tổng quát thể hiện quá trình thực hiện việc đọc truyện online. Điều kiện cần để có thể thực hiện thành công quy trình này là điện thoại phải bật kết nối internet để kết nối tới sever lấy dữ liệu. Quy ước **Host** là phần mềm trên điện thoại của của người dùng và **Server** là phần mềm máy chủ

| **Mã HĐ** | **Thực hiện** | **Mô tả chi tiết** |
| --- | --- | --- |
| DT01 | Người dùng | Chạy ứng dụng |
| DT02 | Host | Kiểm tra kết nối internet  + Nếu có kết nối, chuyển sang DT03  + Nếu không có kết nối, chuyển sang DT06 |
| DT03 | Host | Gửi yêu cầu kết nối tới server, tải dữ liệu tên truyện, tác giả, số chương, lượt đọc về máy. |
| DT04 | Server | Server trả về danh sách bản ghi các truyện gồm: tên truyện, số chương, lượt đọc. |
| DT05 | Host | Đọc dữ liệu server trả về, hiển thị lên giao diện danh sách truyện đồng thời cập nhật cơ sở dữ liệu trong máy. Chuyển tới DT07. |
| DT06 | Host | Truy vấn cơ sở dữ liệu, lấy ra bản ghi các truyện gồm: tên truyện, tác giả, số chương, lượt đọc hiển thị lên giao diện danh sách truyện. Chuyển tới DT07. |
| DT07 | Người dùng | Bấm chọn 1 truyện cần đọc |
| DT08 | Host | Kiểm tra kết nối internet  + Nếu có kết nối, chuyển sang DT09  + Nếu không có kết nối, đưa ra thông báo cần kết nối mạng, quay lại DT02 |
| DT09 | Host | Gửi 1 truy vấn tới sever yêu cầu lấy ra chi tiết truyện người dùng đã chọn ở bước DT07. |
| DT10 | Server | Trả về bản ghi chi tiết truyện gồm: tên truyện, tác giả, số chương, danh sách các chương… |
| DT11 | Host | Đọc dữ liệu server trả về, hiển thị lên giao diện chi tiết truyện đồng thời cập nhật cơ sở dữ liệu trong máy. |
| DT12 | Người dùng | Bấm chọn 1 chương để đọc. |
| DT13 | Host | Gửi 1 truy vấn tới sever yêu cầu lấy ra hình ảnh các trang truyện của chương mà người dùng đã chọn. |
| DT14 | Server | Trả về hình ảnh các trang được yêu cầu. |
| DT15 | Host | Nhận dữ liệu, đẩy lên giao diện đọc truyện. |
| DT16 | Người dùng | Bấm thoát |
| DT17 | Host | Lưu lại trạng thái hệ thống vào cơ sở dữ liệu. Kết thúc quy trình. |

* + 1. **Quy trình đọc truyện từ lịch sử**

| **Mã HĐ** | **Thực hiện** | **Mô tả chi tiết** |
| --- | --- | --- |
| LS01 | Người dùng | Bấm chọn truyện cần đọc từ giao diện lịch sử. |
| LS02 | Host | Kiểm tra kết nối internet  + Nếu có kết nối, chuyển sang LS03  + Nếu không có kết nối, đưa ra thông báo cần kết nối mạng, kết thúc quy trình. |
| LS03 | Host | Gửi 1 truy vấn tới sever yêu cầu lấy ra hình ảnh các trang truyện của chương mà người dùng đã chọn. |
| LS04 | Server | Trả về hình ảnh các trang của chương đang đọc được lưu trong lịch sử. |
| LS05 | Host | Nhận dữ liệu, đẩy lên giao diện đọc truyện lịch sử. |
| LS05 | Người dùng | Bấm thoát |
| LS06 | Host | Lưu lại trạng thái hệ thống vào cơ sở dữ liệu. Kết thúc quy trình. |

* + 1. **Quy trình đọc truyện từ yêu thích**

| **Mã HĐ** | **Thực hiện** | **Mô tả chi tiết** |
| --- | --- | --- |
| YT01 | Người dùng | Bấm chọn truyện cần đọc từ giao diện yêu thích. |
| YT02 | Host | Kiểm tra kết nối internet  + Nếu có kết nối, chuyển sang YT03  + Nếu không có kết nối, đưa ra thông báo cần kết nối mạng, kết thúc quy trình. |
| YT03 | Host | Gửi 1 truy vấn tới sever yêu cầu lấy ra chi tiết truyện người dùng đã chọn. |
| YT04 | Server | Trả về bản ghi chi tiết truyện gồm: tên truyện, tác giả, số chương, danh sách các chương… |
| YT05 | Host | Đọc dữ liệu server trả về, hiển thị lên giao diện chi tiết truyện đồng thời cập nhật cơ sở dữ liệu trong máy. |
| YT05 | Người dùng | Bấm chọn 1 chương để đọc. |
| YT06 | Host | Gửi 1 truy vấn tới sever yêu cầu lấy ra hình ảnh các trang truyện của chương mà người dùng đã chọn. |
| YT07 | Server | Trả về hình ảnh các trang được yêu cầu. |
| YT08 | Host | Nhận dữ liệu, đẩy lên giao diện đọc truyện. |
| YT09 | Người dùng | Bấm thoát |
| YT10 | Host | Lưu lại trạng thái hệ thống vào cơ sở dữ liệu. Kết thúc quy trình. |

* + 1. **Quy trình đọc truyện từ tải về**

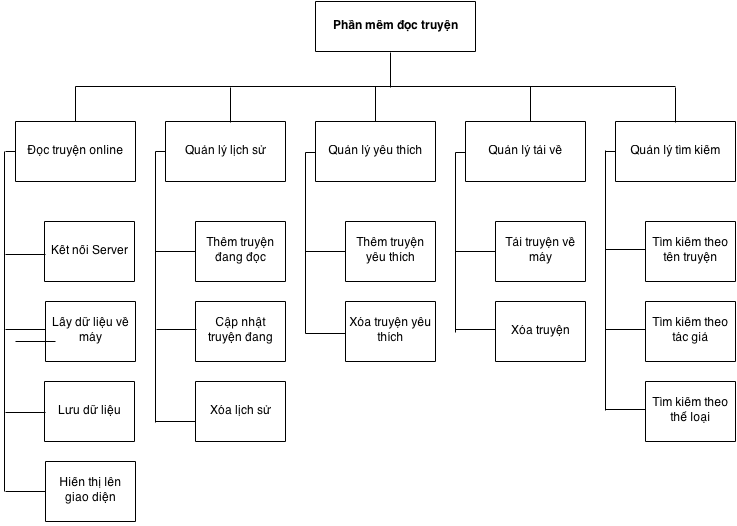
| **Mã HĐ** | **Thực hiện** | **Mô tả chi tiết** |
| --- | --- | --- |
| TV01 | Người dùng | Bấm chọn chương hoặc toàn bộ truyện cần tải từ giao diện chi tiết truyện. |
| TV02 | Host | Kiểm tra kết nối internet  + Nếu có kết nối, chuyển sang TV03  + Nếu không có kết nối, đưa ra thông báo cần kết nối mạng, kết thúc quy trình. |
| TV03 | Host | Kiểm tra tài khoản đăng nhập của người dùng  +Đăng nhập thành công chuyển sang TV06  +Nếu chưa có tài khoản chuyển sang TV04. |
| TV04 | Host | Hiển thị giao diện đăng ký, sau khi người dùng nhập đầy đủ và đúng thông tin, gửi thông tin tài khoản lên server. |
| TV05 | Server | Xác nhận thông tin tài khoản. |
| TV06 | Host | Tải các chương truyện hoặc toàn bộ truyện mà người dùng đã chọn, lưu vào thẻ nhớ. Khi tải thành công thì cập nhật cơ sở dữ liệu. |
| TV07 | Người dùng | Chọn truyện cần đọc đã tải về từ giao diện tải về |
| TV08 | Host | Truy cập cơ sở dữ liệu, lấy ra hình ảnh các trang mà người dùng yêu cầu, đẩy lên giao diện đọc truyện. Kết thúc quy trình. |

* 1. **Sơ đồ phân rã chức năng**

Sau khi phân tích các quy trình nghiệp vụ bài toán, ta rút ra một số chức năng chính của chương trình:

* Đọc truyện online: ứng dụng kết nối tới server để lấy dữ liệu, hình ảnh của truyện cần đọc, sau đó cập nhật một số thông tin vào cơ sở dữ liệu trong máy.
* Lịch sử: Quản lý danh sách các truyện vừa đọc, tạo sự thuận tiện cho người sử dụng có thể tiếp tục đọc lại ngay truyện mình đang đọc dở.
* Yêu thích: Quản lý danh sách các truyện yêu thích, phù hợp với sở thích của mỗi người.
* Tải về: Quản lý các truyện đã được tải về, lưu trong máy, có thể đọc truyện từ đây ngay cả khi không kết nối mạng.
* Tìm kiếm: giúp người dùng nhanh chóng tìm kiếm được truyện cần tìm theo các bộ lọc khác nhau.

Ngoài ra còn một số chức năng phụ như: chia sẻ ứng dụng lên mạng xã hội, tải thêm ứng dụng, cập nhật số lượt đọc của truyện, đăng ký thành viên, đăng nhập …

****

*Hình 8: Sơ đồ phân rã chức năng phần mềm đọc truyện*

* 1. **Xây dựng biểu đồ hoạt động**
     1. **Biểu đồ hoạt động tổng quát**

# C:\Users\chieu_000\Downloads\Biểu đồ hoạt động tổn quát (2).png

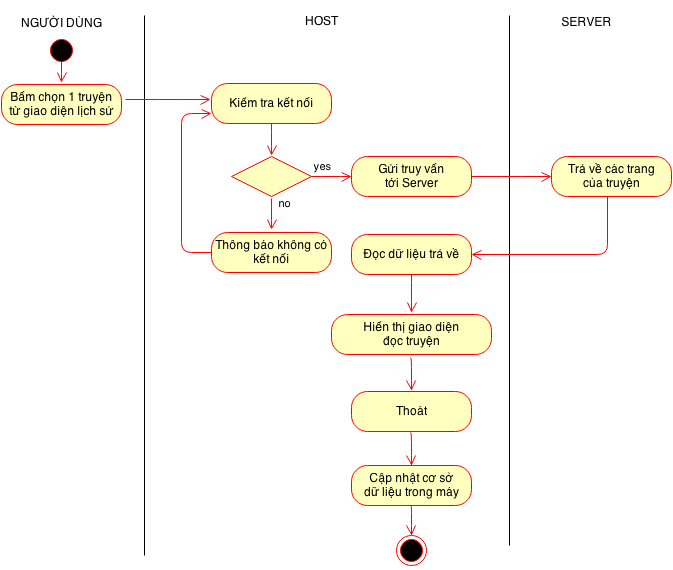
# *Hình 9: Biểu đồ hoạt động tổng quát*

* + 1. **Biểu đồ hoạt động quy trình đọc truyện online**



*Hình 10:* *Biểu đồ hoạt động đọc truyện online*

* + 1. **Biểu đồ hoạt động quy trình đọc truyện từ lịch sử**



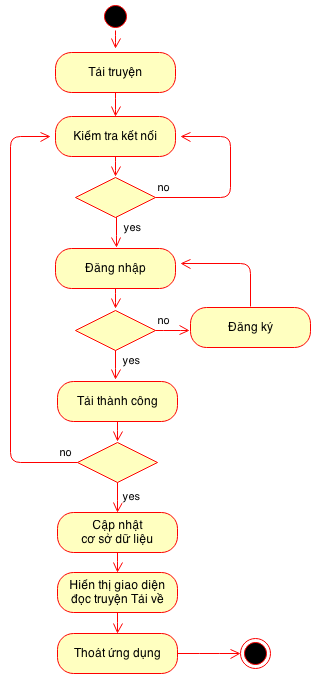
*Hình 11:* *Biểu đồ đọc truyện từ lịch sử*

* + 1. **Biểu đồ hoạt động đọc truyện từ yêu thích**



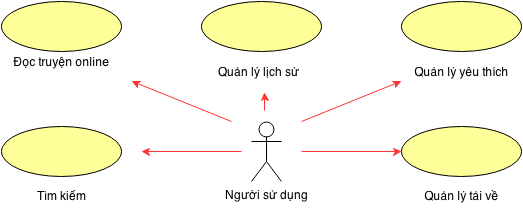
*Hình 12:* *Biểu đồ hoạt động đọc truyện từ yêu thích*

* + 1. **Biểu đồ hoạt động đọc truyện từ tải về**



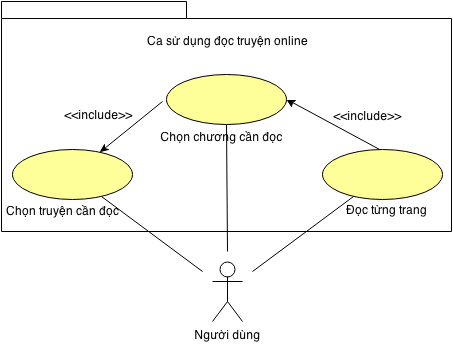
*Hình 13:* *Biểu đồ hoạt động đọc truyện từ tải về*

* 1. **Xây dựng biểu đồ ca sử dụng**
     1. **Biểu đồ ca sử dụng tổng quát**



*Hình 14:* *Ca sử dụng tổng quát*

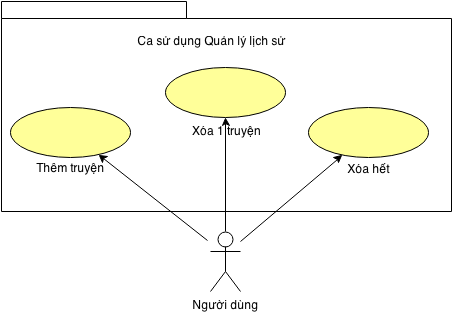
* + 1. **Biểu đồ chi tiết ca sử dụng đọc truyện online**

****

*Hình 15:* *Biểu đồ ca sử dụng đọc truyện online*

|  |  |  |  |
| --- | --- | --- | --- |
| **User case – Đọc truyện online** | | | |
| Mô tả | | | User case này cho phép người dùng có thể xem danh sách truyện , thông tin chi tiết truyện và đọc từng truyện trong danh sách. |
| Tác nhân | Chính | | Người dùng |
| Phụ | |  |
| Tiền điều kiện | | | + Phải có kết nối internet trong lần cài đặt đầu tiên, phải kết nối thành công tới server để tải dữ liệu về máy. Server phải trả về danh sách truyện, chi tiết truyện, trang truyện.  + Lần chạy thứ 2 không cần kết nối tới server vì dữ liệu đã được lưu trong máy |
| Hậu điều kiện | | Success | Khi use case này được thực hiện thì người dùng có thể biết được thông tin , hình ảnh , nội dung truyện mà phần mềm cung cấp. |
| Fail | Thông báo không có kết nối, hoặc không tìm thấy dữ liệu |
| *Đặc tả chức năng* | | | |
| *Luồng sự kiện chính*   |  |  |  |  | | --- | --- | --- | --- | | **STT** | **Hành động của**  **tác nhân** | **Hệ thống hoạt động** | **Dữ liệu liên quan** | | 1 | Chạy ứng dụng | Kết nối tới server lấy dữ liệu về các truyện có trong cơ sở dữ liệu, hiển thị lên màn hình. | Danh sách truyện | | 2 | Chọn truyện cần đọc | Kết nối tới server lấy dữ liệu chi tiết truyện mà người dùng chọn, hiển thị lên màn hình. | + Chi tiết truyện  + Danh sách chương | | 3 | Chọn chương cần đọc | Kết nối tới server lấy hình ảnh của chương truyện mà người dùng chọn, hiển thị ảnh lên màn hình cho người dùng đọc | Danh sách trang truyện | | 4 | Bấm quay lại | Cập nhật thông tin vào cơ sở dữ liệu trong máy. Hiển thị chi tiết truyện và danh sách chương cho người dùng chọn. | Bảng lịch sử | | | | |
| *Luồng sự kiện phụ và sự kiện đặc biệt*   |  |  | | --- | --- | | Luồng sự kiện phụ | Luôn kiểm tra kết nối internet khi tải ảnh của trang về máy, nếu không có mạng yêu cầu người dùng kiểm tra kết nối. | | Sự kiện đặc biệt | Danh sách các truyện, chi tiết truyện, hình ảnh truyện đã có trong cơ sở dữ liệu ở server. | | | | |

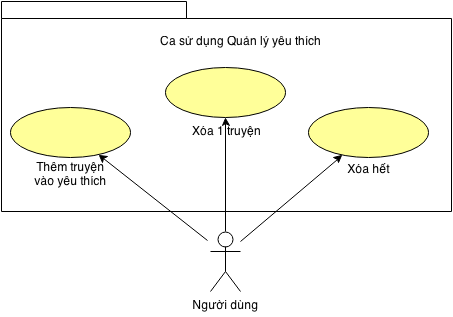
* + 1. **Biểu đồ chi tiết ca sử dụng quản lý lịch sử**

****

*Hình 16:* *Biểu đồ ca sử dụng quản lý lịch sử*

|  |  |  |  |
| --- | --- | --- | --- |
| **User case – Quản lý lịch sử** | | | |
| Mô tả | | | User case này cho phép người dùng có thể xem những thông tin truyện đã đọc như : tên truyện , thời gian đọc gần nhất , khi click chọn vào truyện ở đây sẽ dẫn tới trực tiếp trang đang đọc dở. |
| Tác nhân | Chính | | Người dùng |
| Phụ | |  |
| Tiền điều kiện | | | Bảng lịch sử đã có dữ liệu. |
| Hậu điều kiện | | Success | Khi thao tác thành công các thông tin về truyện, thời gian đọc, trang đang đọc sẽ được cập nhật vào cơ sở dữu liệu |
| Fail | Thông báo không có kết nối, hoặc không tìm thấy dữ liệu |
| *Đặc tả chức năng*  *Luồng sự kiện chính*   |  |  |  |  | | --- | --- | --- | --- | | **STT** | **Hành động của**  **tác nhân** | **Hệ thống hoạt động** | **Dữ liệu liên quan** | | 1 | Đang đọc truyện online | Cập nhật vào cơ sở dữ liệu trong máy tên truyện, trang đang đọc, thời gian đọc | Danh sách truyện đang đọc | | 2 | Xóa 1 truyện | Xóa truyện trong bảng lịch sử | Danh sách truyện đang đọc | | 3 | Xóa toàn bộ | Xóa toàn bộ truyện trong bảng lịch sử | Danh sách trang truyện |   *Luồng sự kiện phụ và sự kiện đặc biệt*   |  |  | | --- | --- | | Luồng sự kiện phụ | Khí thoát khỏi đọc truyện online, trả về tên truyện, mã chương, thời gian, mã trang | | Sự kiện đặc biệt | Tên truyện, mã chương được lấy từ quá trình đọc truyện online. Mã trang đang đọc, thời gian là giờ của hệ thống được tính khi thoát khỏi việc đọc online | | | | |

* + 1. **Biểu đồ chi tiết ca sử dụng quản lý yêu thích**

****

*Hình 17:* *Biểu đồ ca sử dụng quản lý yêu thích*

|  |  |  |  |
| --- | --- | --- | --- |
| **User case – Quản lý yêu thích** | | | |
| Mô tả | | | User case này cho phép người dùng có thể xem những thông tin truyện đã theo sở thích của mình giúp tạo tính cá nhân trong ứng dụng, có thể đọc nhanh truyện từ đây, không cần tìm kiếm trong danh sách truyện. |
| Tác nhân | Chính | | Người dùng |
| Phụ | |  |
| Tiền điều kiện | | | Không có |
| Hậu điều kiện | | Success | Khi thao tác thành công thông tin của truyện sẽ được cập nhật vào cơ sở dữu liệu |
| Fail | Thông báo không có kết nối, hoặc không tìm thấy dữ liệu |
| *Đặc tả chức năng* | | | |
| *Luồng sự kiện chính*   |  |  |  |  | | --- | --- | --- | --- | | **STT** | **Hành động của**  **tác nhân** | **Hệ thống hoạt động** | **Dữ liệu liên quan** | | 1 | Bấm thêm vào yêu thích | Cập nhật vào cơ sở dữ liệu trong máy các thông tin của truyện được chọn. | + Danh sách truyện  + Danh sách truyện yêu thích | | 2 | Xóa 1 truyện | Xóa truyện trong bảng yêu thích | Danh sách truyện yêu thích | | 3 | Xóa toàn bộ | Xóa toàn bộ truyện trong bảng yêu thích | Danh sách truyện yêu thích | | | | |
| *Luồng sự kiện phụ và sự kiện đặc biệt*   |  |  | | --- | --- | | Luồng sự kiện phụ | Kiểm tra xem truyện đã có trong danh sách yêu thích chưa, nếu đã có rồi thì hiện biểu tượng yêu thích trong chi tiết truyện, nếu chưa có thì ẩn biểu tượng yêu thích trong chi tiết truyện. | | Sự kiện đặc biệt | Nếu chưa có truyện yêu thích thì trong giao diện yêu thích phải đưa ra thông báo | | | | |

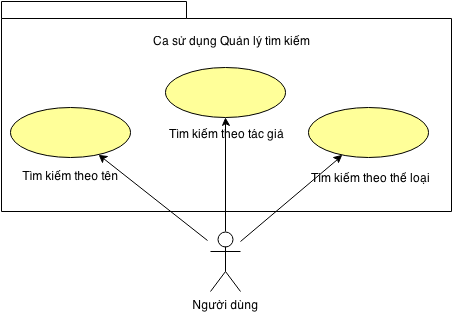
* + 1. **Biểu đồ chi tiết ca sử dụng tải về**

# C:\Users\chieu_000\Downloads\Ca sử dụng Lịch sử (2).png

*Hình 18: Biểu đồ chi tiết ca sử dụng tải về*

|  |  |  |  |
| --- | --- | --- | --- |
| **User case – Quản lý tải về** | | | |
| Mô tả | | | Kết nối tới server tải dữ liệu về truyện được chọn, lưu vào trong thẻ nhớ, hiển thị tiến trình tải lên màn hình. Khi tải xong thì cập nhật danh sách truyện đã tải về lên màn hình để người dùng có thể đọc khi không có kết nối. |
| Tác nhân | Chính | | Người dùng |
| Phụ | |  |
| Tiền điều kiện | | | Người dùng phải đăng nhập. Có kết nối tới server, có thẻ nhớ trong điện thoại. |
| Hậu điều kiện | | Success | Khi thao tác thành công các thông tin về truyện, thời gian đọc, trang đang đọc sẽ được cập nhật vào cơ sở dữu liệu |
| Fail | Thông báo không có kết nối, hoặc không tìm thấy dữ liệu |
| *Đặc tả chức năng* | | | |
| *Luồng sự kiện chính*   |  |  |  |  | | --- | --- | --- | --- | | **STT** | **Hành động của**  **tác nhân** | **Hệ thống hoạt động** | **Dữ liệu liên quan** | | 1 | Chọn truyện tải về | Kết nối tới server tải dữ liệu về truyện được chọn, lưu vào trong thẻ nhớ, hiển thị tiến trình tải lên màn hình. Khi tải xong thì cập nhật danh sách truyện đã tải về lên màn hình. | Danh sách truyện tải về | | 2 | Xóa 1 truyện | Xóa truyện trong bảng tải về, xóa dữ liệu của truyện đã ghi trong thẻ nhớ. | Danh sách truyện tải về | | 3 | Xóa toàn bộ | Xóa toàn bộ truyện trong bảng tải về và dữ liệu của truyện đã ghi trong thẻ nhớ. | Danh sách truyện tải về | Danh sách truyện tải về | | | | |
| *Luồng sự kiện phụ và sự kiện đặc biệt*   |  |  | | --- | --- | | Luồng sự kiện phụ | Khí thoát khỏi đọc truyện online, trả về tên truyện, mã chương, thời gian, mã trang | | Sự kiện đặc biệt | Tên truyện, mã chương được lấy từ quá trình đọc truyện online. Mã trang đang đọc, thời gian là giờ của hệ thống được tính khi thoát khỏi việc đọc online | | | | |

* + 1. **Biểu đồ chi tiết ca sử dụng tìm kiếm**

**

*Hình 19: Biểu đồ chi tiết ca sử dụng tìm kiếm*

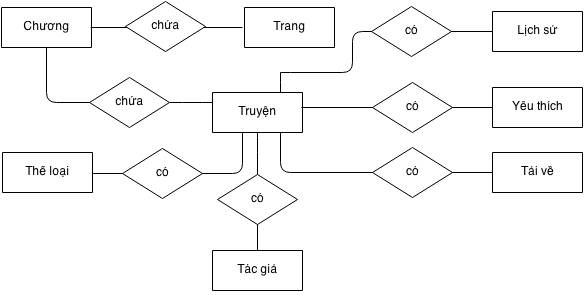
|  |  |  |  |
| --- | --- | --- | --- |
| **User case – Quản lý tìm kiếm** | | | |
| Mô tả | | | User case này cho phép người dùng tìm kiếm thông tin của truyện , tác giả , thể loại qua cách nhập tên truyện tên tác giả, thể loại vào ô tìm kiếm . |
| Tác nhân | Chính | | Người dùng |
| Phụ | |  |
| Tiền điều kiện | | | Người dùng chọn bộ lọc tương ứng |
| Hậu điều kiện | | Success | Sau khi nhập tên truyện, tên tác giả thì hiển thị ra thông tin về truyện , tác giả cần tìm. |
| Fail | Thông báo không có kết nối, hoặc không tìm thấy dữ liệu |
| *Đặc tả chức năng* | | | |
| *Luồng sự kiện chính*   |  |  |  |  | | --- | --- | --- | --- | | **STT** | **Hành động của**  **tác nhân** | **Hệ thống hoạt động** | **Dữ liệu liên quan** | | 1 | Chọn bộ lọc tương ứng | Hiển thị form tìm kiếm |  | | 2 | Nhập điều kiện tìm kiếm | Hiển thị form các truyện tìm được | + Danh sách truyện  + Tác giả  + Thể loại | | | | |
| *Luồng sự kiện phụ và sự kiện đặc biệt*   |  |  | | --- | --- | | Luồng sự kiện phụ | Nếu không tìm thấy truyện thỏa mãn điều kiện thì yêu cầu nhập lại điều kiện tìm hoặc dừng. | | Sự kiện đặc biệt |  | | | | |

# CHƯƠNG IV: THIẾT KẾ HỆ THỐNG

1. 1. **Phân tích và thiết kế cơ sở dữ liệu**
      1. **Xác định thực thể và các thuộc tính**

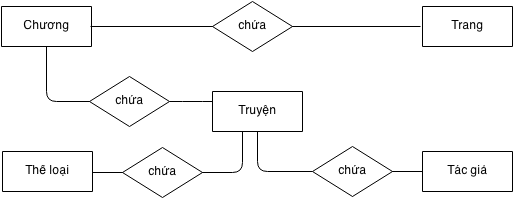
Xuất phát từ yêu cầu lưu trữ thông tin của ứng dụng, chúng ta có thể nêu ra những thực thể như sau:

* **Ở phía client**
* Thực thể **Thể Loại** :là thực thể lưu thông tin thể loại truyện bao gồm các thuộc tính :(Mã Thể Loại , Tên Thể Loại)
* Thực thể **Tác giả:** là thực thể lưu thông tin tác giả bao gồm các thuộc tính: (Mã tác giả, Tên tác giả).
* Thực thể **Truyện**: là thể thực thể lưu chi tiết các thông tin của truyện gồm các thuộc tính :(Mã Truyện , Tên Truyện , Ảnh Bìa , Tác Giả , Mã Thể Loại , Mô Tả , Số Trang).
* Thực thể **Chương**: là thực thể lưu chi tiết thông tin các chương của truyện gồm các thuộc tính: (Mã truyện, Mã chương, Tên chương);
* Thực thể **Trang** : là thực thể lưu lại nội dung của bộ truyện gồm các thuốc tính:( Mã Truyện ,Mã chương, Mã Trang , Hình Ảnh , Âm Thanh)
* Thực thể **Yêu Thích** : là thực thể lưu lại danh sách những truyện mà người dùng yêu thích gồm các thuộc tính : (Mã Yêu Thích , Mã Truyện)
* Thực thể **Lịch Sử** : là thực thể lưu lại danh sách những truyện đã được đọc gồm các thuộc tính: (Mã Lịch Sử , Mã Truyện , Thời Gian , Lượt Xem).
* Thực thể **Tải về** : là thực thể lưu lại danh sách cách truyện đã tải về gồm các thuộc tính: (Mã truyện, Mã chương, Mã trang, Hình ảnh….)



*Hình 20: Mô hình thực thể liên kết ở client*

* **Ở phía server**
* Thực thể **Thể Loại** :là thực thể lưu thông tin thể loại truyện bao gồm các thuộc tính :(Mã Thể Loại , Tên Thể Loại)
* Thực thể **Tác giả:** là thực thể lưu thông tin tác giả bao gồm các thuộc tính: (Mã tác giả, Tên tác giả).
* Thực thể **Truyện**: là thể thực thể lưu chi tiết các thông tin của truyện gồm các thuộc tính :(Mã Truyện , Tên Truyện , Ảnh Bìa , Tác Giả , Mã Thể Loại , Mô Tả , Số Trang).
* Thực thể **Chương**: là thực thể lưu chi tiết thông tin các chương của truyện gồm các thuộc tính: (Mã truyện, Mã chương, Tên chương);
* Thực thể **Trang** : là thực thể lưu lại nội dung của bộ truyện gồm các thuốc tính:( Mã Truyện , Mã Trang , Hình Ảnh , Âm Thanh)
* Thực thể **Góp ý**: là thực thể lưu thông tin của truyện để chia sẻ lên mạng xã hội
* Thực thể **Người dùng**: là thực thể lưu lại tên thông tin người dùng bao gồm các thuộc tính: (Mã người dùng, Tên đăng nhập, Mật khẩu)

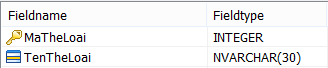


*Hình 21: Mô hình thực thể liên kết ở server*

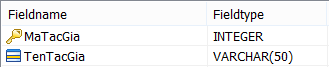
* + 1. **Mô hình quan hệ**
* **Ở phía Client**

Các bảng quan hệ:

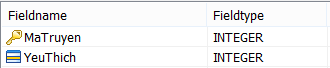
* Thể Loại (Mã Thể Loại , Tên Thể Loại)



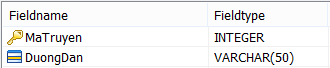
* Tác Giả(Mã Tác Giả, Tên Tác Giả)



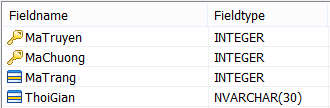
* Yêu Thích(Mã Truyện, Yêu Thích)



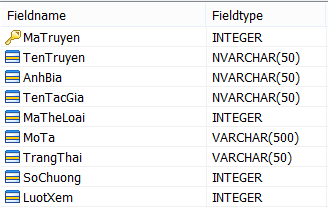
* Tải Về(Mã Truyện, Đường Dẫn)



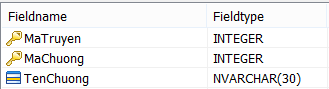
* Lịch Sử(Mã Truyện, Mã Chương, Mã Trang, Thời Gian)



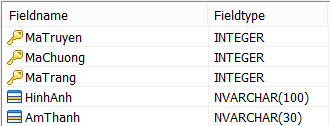
* Truyện (Mã Truyện , Tên Truyện , Ảnh Bìa , Mã Tác Giả , Mã Thể Loại , Mô Tả , Trạng thái, Số chương, Lượt xem).



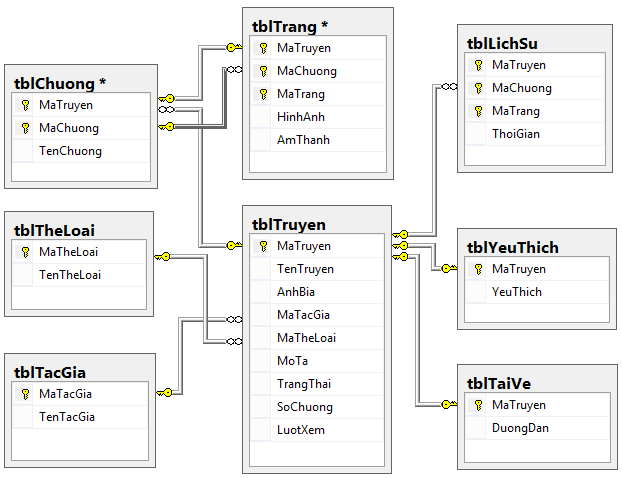
* Chương(Mã truyện, Mã Chương, Tên Chương)



* Trang ( Mã Truyện , Mã Chương, Mã Trang , Hình Ảnh , Âm Thanh)



Sau khi phân tích các bảng ta có mô hình quan hệ giữa các thực thể ở client như sau:



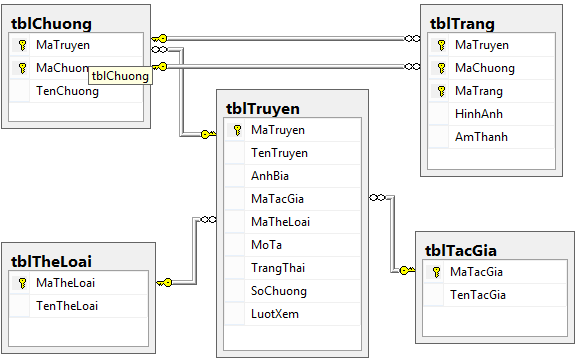
*Hình 22: Mô hình quan hệ ở client*

* **Ở phía Server**

Các bảng quan hệ ở phía server có thuộc tính và kiểu dữ liệu như ở client:

* Thể Loại (Mã Thể Loại , Tên Thể Loại)
* Tác Giả(Mã Tác Giả, Tên Tác Giả)
* Truyện (Mã Truyện , Tên Truyện , Ảnh Bìa , Mã Tác Giả , Mã Thể Loại , Mô Tả , Trạng thái, Số chương, Lượt xem)
* Chương(Mã truyện, Mã Chương, Tên Chương)
* Trang ( Mã Truyện , Mã Chương, Mã Trang , Hình Ảnh , Âm Thanh)

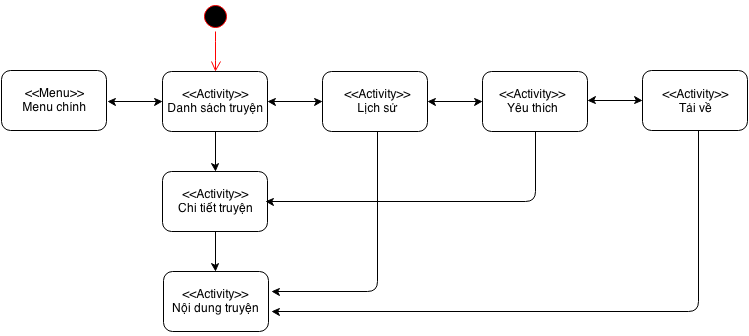
Sau khi phân tích các bảng ta có mô hình quan hệ giữa các thực thể ở server như sau:



*Hình 23: Mô hình quan hệ ở server*

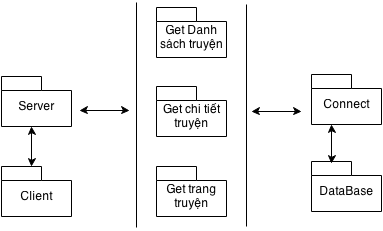
* 1. **Các giao diện cần có của chương trình**

|  |  |
| --- | --- |
| **Các hoạt động** | **Phần tử giao diện cần thiết** |
| * Chọn truyện | Màn hình danh sách truyện |
| * Xem chi tiết truyện | Màn hình chi tiết truyện |
| * Xem nội dung truyện | Màn hình xem nội dung truyện |
| * Xem lịch sử đã đọc | Màn hình danh sách lịch sử |
| * Xem danh sách yêu thích | Màn hình danh sách lịch sử |
| * Xem danh sách truyện đã tải về * Menu | Màn hình danh sách truyện đã tải về  Màn hình hiện giao diện menu |

****

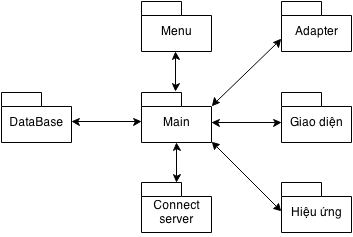
*Hình 24: Biểu đồ di chuyển giữa các Activity*

* 1. **Thiết kế thành phần hệ thống**
     1. **Module server**



*Hình 25: Mô hình các gói của module server*

* + 1. **Module client**

****

*Hình 26: Mô hình các gói module client*

* 1. **Triển khai ứng dụng vào thực tế**
     1. **Hướng dẫn cài đặt**



* **Ở client**
* Yêu cầu phần cứng:
* CPU : 512MHz
* RAM : 256MB
* Yêu cầu phần mềm:
* Hệ điều hành: Android 3.0 trở lên
* Hướng dẫn cài đặt:
* Sử dụng filemybook.apk để cài đặt ứng dụng lên thiết bị.
* **Ở server**
* Yêu cầu phần cứng
* CPU : 1 GHz
* RAM : 512MB
* Yêu cầu phần mềm
* Hệ điều hành windown 98 trở lên
* Có cài đặt phần mềm xampp để tạo server
* Hướng dẫn cài đặt
* Import file doctruyen.sql vào cơ sở dữ liệu ở server
* Copy thư mục Doctruyen vào htdoc của xampp, chỉnh sửa thông tin scác trường DB\_CONNECTION\_STRING, DB\_USER và DB\_PASSWORD tương ứng là đường dẫn cơ sở dữ liệu, tên tài khoản và mật khẩu đăng nhập.
* Khởi chạy server
  + 1. **Giao diện ứng dụng**

Sau đây là một số hình ảnh giao diện của chương trình:

|  |  |
| --- | --- |
| danhsach.png  *Hình 27: Giao diện danh sách truyện* | chitiet.png  *Hình 28: Giao diện chi tiết truyện* |
| Image 17.png  *Hình 29: Giao diện đọc truyện* | menu.png  *Hình 30: Giao diện menu* |

# KẾT LUẬN

Trong quá trình thực hiện đề tài, do phải tiếp cận với nhiều lĩnh vực mới nên sinh viên đã gặp phải không ít khó khăn, tuy nhiên với sự giúp đỡ của những người có kinh nghiệm cùng với nỗ lực của bản thân, đồ án đã được hoàn thành đúng hạn và thu được kết quả khả quan. Bên cạnh đó, do thời gian thực hiện có hạn nên một vài phần của đồ án có kết quả chưa được hoàn toàn như mong đợi. Trong thời gian tới, đề tài sẽ tiếp tục được chỉnh sửa, mở rộng để hoàn thiện hơn. Sau quá trình triển khai thực hiện, sinh viên đã hoàn thành khá tốt các yêu cầu đề ra, cụ thể :

* Xây dựng thành công module phần mềm máy chủ tương đối hoàn thiện, có đầy đủ các chức năng đề ra: cho phép người dùng kết nối để đọc truyện trực tiếp, cho phép tải truyện về…
* Xây dựng thành công module máy khách với đầy đủ các chức năng: kết nối tới server, đọc truyện online, tải truyện đọc offline, xây dựng thành công hiệu ứng lật trang.

Tự đánh giá kết quả đạt được:

**Ưu điểm:**

* Chương trình có giao diện thân thiện, hoạt động ổn định, ít xảy ra lỗi trong quá trình chạy.
* Chương trình dễ sử dụng, có dung lượng nhỏ, khả năng cài đặt cho hầu hết các máy chạy hệ điều hành Android.

**Nhược điểm:**

* Hiệu ứng lật trang còn chưa được mượt, ít hiệu ứng hình ảnh.
* Chương trình chạy không được trơn tru trên một số thiết bị di động có cấu hình thấp.

**Hướng phát triển**

* Chương trình sẽ được hoàn thiện, mở rộng thêm chức năng đọc truyện tổng hợp cả truyện tranh và truyện chữ với nhiều hiệu ứng bắt mắt và thân thiện hơn .

# TÀI LIỆU THAM KHẢO

* ***Android Developers Website***

<http://developer.android.com/sdk/index.html>

* ***The JavaTM Tutorials***

<http://docs.oracle.com/javase/tutorial/index.html>

* ***Một số Diễn đàn Android - Góc lập trình***

<http://vietandroid.com/forum.php>

http://android.vn/articles/lap-trinh-mobile/

* ***Các hỏi đáp tại website***

<http://stackoverflow.com/>