Chương 1: Tổng Quan

* 1. **Thực trạng trẻ em hiện nay**

Xã hội Việt Nam ngày càng hiện đại, kèm với đó là những thay đổi về cuộc sống của người dân theo nhiều hình thức.

Trẻ em được sống trong môi trường tốt hơn, đầy đủ vật chất hơn, tiếp cận nền giáo dục hiện đại, với những phương pháp nuôi dạy đã được cải tiến. Bên cạnh đó những vấn đề tiêu cực cũng ảnh hưởng không nhỏ tới lối sống của các em như “game online chứa nội dung bạo lực tràn lan, tiếp xúc với cuộc sống người lớn quá sớm qua phim ảnh hay internet… học đòi theo phong cách sống nước ngoài bỏ qua nề nếp dân tộc...”

Trong tình hình hiện nay, những mối quan hệ trong gia đình giữa cha mẹ và con cái cũng có nhiều sự xáo trộn, khi mà xã hội càng phát triển thì các bậc cha mẹ cũng càng có ít thời gian cho con cái, đây là một hệ lụy xấu cho con trẻ.

Với sự phát triển vượt bậc của nền công nghệ thông tin Viêt Nam nói riêng và của thế giới nói chung, trẻ em đã được tiếp xúc với máy tính rất nhiều.

Tuy Việt Nam đã hoàn thành những mục tiêu thiên niên kỷ đối với trẻ em, nhưng ở nông thôn vẫn còn rất nhiều những trẻ em không được không được tiếp cận với nền giáo dục hiện đại.

* 1. **Thực trạng giáo dục tiểu học**

Theo Báo cáo quốc gia về các Mục tiêu Phát triển Thiên niên kỷ (MDG) 2005, Việt Nam đã đạt được những thành tựu đầy ấn tượng trong lĩnh vực giáo dục và đào tạo, đặc biệt khi xét theo mối tương quan với mức thu nhập bình quân đầu người. Tính đến năm 2000, Việt Nam đã hoàn thành mục tiêu phổ cập giáo dục tiểu học.

Tuy nhiên theo thông kê ở Việt Nam, tính đến năm 2008, số người mù chữ trên cả nước là gần 1,7 triệu. Qua con số này ta có thể thấy đôi tượng mù chữ không chỉ có trẻ em mà còn nhiều thanh thiếu niên, người lớn đang ở trong tình trạng đó.

Biết đọc, biết viết là quyền cơ bản của con người, giúp con người không chỉ đọc, viết, làm tính mà còn hiểu, sáng tạo và giao tiếp thông qua các tài liệu, giúp cho cá nhân nâng cao kiến thức.

Có thể thấy rất nhiều lý do như tâm lý ngại ngùng của những người khuyết tật, những người lớn tuổi nhưng chưa biết chữ, hoặc có thể do điều kiện kinh tế, điều kiện thời gian hạn hẹp…

* 1. **Tổng quan khái niệm Mạng Xã Hội**

Mạng xã hội, hay gọi là mạng xã hội ảo, (tiếng Anh: social networking) là dịch vụ nối kết các thành viên cùng sở thích trên Internet lại với nhau với nhiều mục đích khác nhau không phân biệt không gian và thời gian.

Mạng xã hội có những tính năng như chat, e-mail, phim ảnh, voice chat, chia sẻ file, blog và xã luận. Mạng đổi mới hoàn toàn cách cư dân mạng liên kết với nhau và trở thành một phần tất yếu của mỗi ngày cho hàng trăm triệu thành viên khắp thế giới. Các dịch vụ này có nhiều phương cách để các thành viên tìm kiếm bạn bè, đối tác: dựa theo group (ví dụ như tên trường hoặc tên thành phố), dựa trên thông tin cá nhân (như địa chỉ e-mail hoặc screen name), hoặc dựa trên sở thích cá nhân (như thể thao, phim ảnh, sách báo, hoặc ca nhạc), lĩnh vực quan tâm: kinh doanh, mua bán...4

Hiện nay thế giới có hàng trăm mạng mạng xã hội khác nhau, với MySpace và Facebook nổi tiếng nhất trong thị trường Bắc Mỹ và Tây Âu; Orkut và Hi5 tại Nam Mỹ; Friendster tại Châu Á và các đảo quốc Thái Bình Dương. Những mạng xã hội khác gặt hái được thành công đáng kể theo vùng miền như Bebo tại Anh Quốc, CyWorld tại Hàn Quốc, Mixi tại Nhật Bản và Yahoo! 360 tại Việt Nam.

* Lịch sử

Mạng xã hội xuất hiện lần đầu tiên năm 1995 với sự ra đời của trang Classmate với mục đích kết nối bạn học, tiếp theo là sự xuất hiện của SixDegrees vào năm 1997 với mục đích giao lưu kết bạn dựa theo sở thích.

Năm 2002, Friendster trở thành một trào lưu mới tại Hoa Kỳ với hàng triệu thành viên ghi danh. Tuy nhiên sự phát triển quá nhanh này cũng là con dao hai lưỡi: server của Friendster thường bị quá tải mỗi ngày, gây bất bình cho rất nhiều thành viên.

Năm 2004, MySpace ra đời với các tính năng như nhúng phim ảnh (embeded video) và nhanh chóng thu hút hàng chục ngàn thành viên mới mỗi ngày, các thành viên cũ của Friendster cũng lũ lượt chuyển qua MySpace và trong vòng một năm, MySpace trở thành mạng xã hội đầu tiên có nhiều lượt xem hơn cả Google và được tập đoàn News Corporation mua lại với giá 580 triệu USD.

Năm 2006, sự ra đời của Facebook đánh dấu bước ngoặt mới cho hệ thống mạng xã hội trực tuyến với nền tảng lập trình "Facebook Platform" cho phép thành viên tạo ra những công cụ (apps) mới cho cá nhân mình cũng như các thành viên khác dùng. Facebook Platform nhanh chóng gặt hái được thành công vượt bậc, mang lại hàng trăm tính năng mới cho Facebook và đóng góp không nhỏ cho con số trung bình 19 phút mà các thành viên bỏ ra trên trang này mỗi ngày.

* Cấu thành

Nút (node): Là một thực thể trong mạng. Thực thể này có thể là một cá nhân, một doanh nghiệp hoặc một tổ chức bất kỳ nào đó

Liên kết (tie): là mối quan hệ giữa các thực thể đó. Trong mạng có thể có nhiều kiểu liên kết. Ở dạng đơn giản nhất, mạng xã hội là một đơn đồ thị vô hướng các mối liên kết phù hợp giữa các nút. Ta có thể biểu diễn mạng liên kết này bằng một biểu đồ mà các nút được biểu diễn bởi các điểm còn các liên kết được biểu diễn bởi các đoạn thẳng.

* Mục tiêu

Tạo ra một hệ thống trên nền Internet cho phép người dùng giao lưu và chia sẻ thông tin một cách có hiệu quả, vượt ra ngoài những giới hạn về địa lý và thời gian.

Xây dựng lên một mẫu định danh trực tuyến nhằm phục vụ những yêu cầu công cộng chung và những giá trị của cộng đồng.

Nâng cao vai trò của mỗi công dân trong việc tạo lập quan hệ và tự tổ chức xoay quanh những mối quan tâm chung trong những cộng đồng thúc đẩy sự liên kết các tổ chức xã hội.

* 1. **Thực trạng mạng xã hội hiện nay**

Mạng xã hội ở viêt nam đã được hình thành và đang phát triển khá nhanh. Ngoài những mạng xã hội của thế giới ở việt nam hiện nay mạng xã hội đã được các doanh nghiệp công nghệ thông tin trong nước chú ý phát triển như mạng xã hội me.zing.vn, yume.com.. tuy nhiên mức độ trải rộng của các mạng xã hội này chưa được chú ý nhiều.

* 1. **Lý do chọn đề tài**

Ngày nay với sự phát triển mạnh mẽ của khoa học và công nghệ, nhu cầu học tập, giao lưu chia sẽ của con người không còn diễn ra trong một phạm vi nhỏ hẹp mà nó sẽ diễn ra trong phạm vi lớn trong một quốc gia trong một khu vực hoăc trên toàn để giới. Trong những công nghệ đó thì mạng internet có thể coi như là môt phần không thể thiếu trong cuộc sống hiện đại của con người. Chỉ có thể với mạng internet những nhu cầu về học tập giải trí của con người sẽ được đáp ứng một cách tốt nhất, phong phú và đa dạng nhất.­­ Thực tế hiện nay trên mạng có rất nhiều chương trình về học tập, về giải trí, về giao lưu nhưng đó có thể chỉ dành cho những đối tượng là thanh thiếu niên, những người trẻ tuổi và một ít thành phần khác. Với đối tượng là những gia đình trẻ và con cái của họ có nhu cầu chia sẽ thông tin là không nhỏ, họ những người còn đang rất thiếu những kinh nghiệm về cuộc sống gia đình, nuôi dạy con cái… Sự bận rộn với công việc hằng ngày là nguyên nhân chiếm hết thời gian qua tâm con cái của các bậc cha mẹ, nhưng con cái của họ cần một môi trường có thể tiếp thu những kiến thức căn bản cũng như nơi các em có thể giao lưu kết bạn vui chơi giải trí… Trong môi trường đó cha mẹ có thể cùng chơi, hướng dẫn con cái học tập, có thể bù đắp những hạn chế về thời gian.

Qua những nhu cầu đòi hỏi trên chúng tôi mong muốn xây dựng một môi trường tốt, nơi vui chơi, học tập hiệu quả nhằm góp phần vào việc gieo trồng những mầm non cho đất nước, nhưng hạt giống tốt cho xã hội, đó cũng là mục tiêu của đề tài “Hệ Thống Giáo Dục Tiểu Học”.

Chương 2: Đặc tả hệ thống

* 1. Đặc tả chức năng của hệ thống

Các chức năng của hệ thống đều lấy “**mạng gia đình”** làm nền tảng, có những chức năng chính như sau.

* + 1. Giáo dục
* Học tiếng việt, học toán

Hệ thống sẽ xây dựng những bài giảng sinh động của môn **Tiếng việt** và môn **Toán**. Người học có thể dễ dàng tiếp thu những kiến thức mới, đối tượng người dùng chủ yếu của chức năng này sẽ là các em thiếu nhi (những bé đang học mẫu giáo mầm non và học sinh lớp một).

* Kiểm tra kiến thưc

Ứng với mỗi bài học người dùng sẽ được kiểm tra lại kiến thức, nhằm ôn tập lại, trau dồi và nâng cao kiến thức đã học được.

* Công cụ bổ trợ

Nhằm phục vụ tốt hơn nhu cầu học tập của học sinh, hệ thống cung cấp một số công cụ bỗ trợ cho việc học của học sinh như là **bảng tính** (mô tả như những que tính ngoài thế giới thực), **bảng vẽ** là công cụ giúp học sinh có thể vẽ theo sở thích của mình, bảng cửu chương, bảng chử cái …

* + 1. Vui chơi
* Trò chơi

Đi cạnh với nhu cầu học tập là nhu cầu giải trí. Sau những giờ học căng thẳng, học sinh có thể vui chơi. Ở đây hệ thống không tập trung vào những trò chơi giải trí đơn thuần mà là những trò vừa chơi vừa học.

* Đọc sách

Trong cuộc sống hiện đại nhu cầu đọc sách của con người ngày càng lớn. Việc đọc sách của con người không còn đơn thuần là việc cầm cuốn sách và ngồi đọc mà có sẽ chuyển dần sang nhu cầu đọc sách điện tử. Với xu hướng đó hệ thống cũng mong muốn có thêm chức năng đọc sách giúp người dùng có thể đọc những cuốn sách hay ngay trên máy tính của mình mà không nhất thiết phải bỏ tiền ra mua những cuốn sách in.

* + 1. Mạng gia đình
* Kết bạn

Đây là một chức năng hữu ích nhất của “**mạng gia đình**”. Một gia đình có thể kết bạn với những gia đình cùng sở thích nào đó, cùng một khu vực, hoặc có con cái học chung trường. Hệ thống có hỗ trợ việc giới thiệu bạn bè, tìm kiếm bạn bè thông qua thông tin…

* Hộp thư

Đóng vai trò như là một hòm thư trong thế giới thực, bạn chỉ cần có một địa chỉ mọi người có thể gửi thư đến, cũng như là bạn gửi thư cho họ. Thư ở đây giống như ơ thế giới thực nó có thể là thư thông báo, thư mời kết bạn, thư mời gia nhập nhóm, thư liên lạc giữa các gia đình cũng như là thư liên lạc giữa gia đình và thầy cô giáo của con em mình…

* Tạo nhóm

Người dùng có thể tạo cho mình một nhóm theo đặc điểm chung gì đó như là cùng chung sở thích, thể thao, giải trí… Người dùng có thể mời bạn bè của mình gia nhập nhóm hoặc bản thân mình có thể chọn gia nhập một nhóm mà mình thích.

* Chia sẽ hình ảnh

Người dùng có thể đưa ảnh của mình lên hệ thống, bạn bè có thể xem và bình luận. Có thể tạo cho mình những album ảnh, bạn bè cũng có thể bình luận hay cảm nhận về album này.

* Viết Blogs (Nhật ký điện tử)

Người dùng có thể viết nhật ký online từ hệ thống, Có thể cho phép bạn bè của mình nhìn thấy hay bình luận. Cũng có thể xem blog của những người khác trong nhóm hoặc trong danh sách bạn bè của mình.

* Bình luận đánh giá, tag

Khi mà bạn bè đăng lên những thông tin chia sẽ, cảm nhận của họ người dùng có thể bình luận những thông tin đó, Hoặc khi muốn đánh dấu từ khóa cho một thông tin như là blogs, hoặc hình ảnh…

* 1. Đặc tả yêu cầu
     1. Tạo tài khoản và bổ sung thông tin cá nhân

Người thực hiện: Người dùng là gia đình, cá nhân, giáo viên, các em học sinh tiểu học.

Quá trình thực hiện: Người dùng muốn sử dụng hệ thống phải đăng ký một tại khoản bằng cách điền những thông tin mà hệ thống yêu cầu sau đó điền thông tin cá nhân. Và để sử dụng người dùng phải chứng thực bằng mã mà hệ thống gửi vể trong email đã đăng ký.

* + 1. Đăng trạng thái, cảm nhận

Người thực hiện: Người dùng là gia đình, cá nhân, giáo viên, các em học sinh tiểu học

Quá trình thực hiện: Người dùng đăng nhập vào hệ thống, có thể đăng trạng thái hay cảm nhận của mình đối với bạn bè.

* + 1. Xem trạng thái, cảm nhận của mình và bạn bè

Người thực hiện: Người dùng là gia đình, cá nhân, giáo viên, các em học sinh tiểu học

Quá trình thực hiện: Sau khi đăng nhập thành công người dùng chọn vào khu vực mạng gia đình. Thì những thông tin trạng thái mới nhất của mình và bạn bè sẽ được hiện lên.

* + 1. Xem thông tin cá nhân

Người thực hiện: Người dùng là gia đình, cá nhân, giáo viên, các em học sinh tiểu học đã có tài khoản trong hệ thống

Quá trình thực hiện: Khi người dùng đăng nhập thành công vào hệ thống họ có thể xem thông tin cá nhân của mình.

* + 1. Chỉnh sửa thông tin cá nhân

Người thực hiện: Người dùng là gia đình, cá nhân, giáo viên, các em học sinh tiểu học đã có tài khoản trong hệ thống

Quá trình thực hiện: khi người dùng xem thông tin cá nhân của mình nếu có sai sót hoặc muốn thay đổi thì họ có thể chính sửa những thông tin đó.

* + 1. Xem thông tin của bạn bè

Người thực hiện: Người dùng là gia đình, cá nhân, giáo viên, các em học sinh tiểu học đã có tài khoản trong hệ thống

Quá trình thực hiện: Người dùng xem trong danh sách của mình có bao nhiêu bạn và thông tin về những người bạn đó.

* + 1. Tìm kiếm người quen

Người thực hiện: Người dùng là gia đình, cá nhân, giáo viên, các em học sinh tiểu học đã có tài khoản trong hệ thống

Quá trình thực hiện: Người dùng nhập thông tin cần tìm kiếm vào hệ thống dựa vào thông tin nhập vào và tìm kiếm thông tin đó theo yêu cầu.

* + 1. Đăng xuất khỏi hệ thống

Người thực hiện: Người dùng, System

Quá trình thực hiện: Khi quá thời gian lưu session thì hệ thống sẽ tự động đăng xuất, người dùng cũng có thể tự mình thoát khỏi hệ thống nếu mình muốn.

* + 1. Kết bạn

Người thực hiện: Người dùng là gia đình, cá nhân, giáo viên, các em học sinh tiểu học đã có tài khoản trong hệ thống.

Quá trình thực hiên: người dùng phải biết được thông tin của bạn bè sau đó vào hộp thư của mình và gửi thư kết bạn đến người bạn của mình. Và khi người bạn nhận được thư họ sẽ phản hồi là đồng ý thì hai người sẽ trở thành bạn bè trên mạng xã hội này.

* + 1. Xem hộp thư

Người thực hiện: Người dùng là gia đình, cá nhân, giáo viên, các em học sinh tiểu học đã có tài khoản trong hệ thống.

Quá trình thực hiện: Người dùng đăng nhập vào hệ thống sẽ thấy được chức năng hộp thư. Hệ thống sẽ đưa ra danh sách thư mới, thư đã đọc… cho người dùng biết.

* + 1. Gửi thư

Người thực hiện: Người dùng là gia đình, cá nhân, giáo viên, các em học sinh tiểu học đã có tài khoản trong hệ thống.

Quá trình thực hiên: Người dùng xem hộp thư của mình và có thể gửi thư cho người khác.

* + 1. Đọc thư

Người thực hiện: Người dùng là gia đình, cá nhân, giáo viên, các em học sinh tiểu học đã có tài khoản trong hệ thống.

Quá trình thực hiện: Khi có thư mới hệ thống sẽ hiện lên cho người dùng mỗi khi họ đăng nhập vào. Họ có thể đọc những thư này và phản hồi lại.

* + 1. Xóa thư

Người thực hiện: Người dùng là gia đình, cá nhân, giáo viên, các em học sinh tiểu học đã có tài khoản trong hệ thống.

Quá trình thực hiện: Hệ thống sẽ xóa những thư quá lâu hoặc thư rác khi người dùng yêu cầu.

* + 1. Xem danh sách nhóm

Người thực hiện: Người dùng là gia đình, cá nhân, giáo viên, các em học sinh tiểu học đã có tài khoản trong hệ thống.

Quá trình thực hiện: Người dùng khi mới gia nhập vào mạng sẽ không thuộc nhóm nào. Họ có thể xem danh sách nhóm hiện đã có trong mạng. nếu thích họ có thể gia nhập vào nhóm đó.

* + 1. Tạo nhóm mới

Người thực hiện: Người dùng là gia đình, cá nhân, giáo viên, các em học sinh tiểu học đã có tài khoản trong hệ thống.

Quá trình thực hiên: Người dùng có thể tự tạo cho mình một nhóm mới và mời bạn bè mình gia nhập vào nhóm đó.

* + 1. Gia nhập nhóm

Người thực hiện: Người dùng là gia đình, cá nhân, giáo viên, các em học sinh tiểu học đã có tài khoản trong hệ thống.

Quá trình thực hiện: Người dùng có thể gia nhập các nhóm mà mình cảm thấy yêu thích. Khi xem danh sách các nhóm người dùng có thể nhấn nút gia nhập và hệ thống sẽ tự động xử lý gia nhập.

* + 1. Xem danh sách album ảnh

Người thực hiện: Người dùng là gia đình, cá nhân, giáo viên, các em học sinh tiểu học đã có tài khoản trong hệ thống.

Quá trình thực hiên: Hệ thống có chức năng chia sẻ hình ảnh người dùng có thể xem danh sách album do bạn bè của mình đưa lên.

* + 1. Tạo album ảnh mới

Người thực hiện: Người dùng là gia đình, cá nhân, giáo viên, các em học sinh tiểu học đã có tài khoản trong hệ thống.

Quá trình thực hiện: Người dùng có thể tự tạo riêng cho mình album ảnh. đưa hình lên, chỉnh sửa.

* + 1. Chỉnh sửa album ảnh

Người thực hiện: Người dùng là gia đình, cá nhân, giáo viên, các em học sinh tiểu học đã có tài khoản trong hệ thống.

Quá trình thực hiện: Người dùng có thể chỉnh sửa album của mình. Tên ảnh chú thích, giới thiệu ảnh.

* + 1. Xóa album ảnh

Người thực hiện: Người dùng là gia đình, cá nhân, giáo viên, các em học sinh tiểu học đã có tài khoản trong hệ thống.

Quá trình thực hiện: Người dùng có thể xóa những album ảnh do mình đưa lên.

* + 1. Xem danh sách blogs

Người thực hiên: Người dùng là gia đình, cá nhân, giáo viên, các em học sinh tiểu học đã có tài khoản trong hệ thống.

Quá trình thực hiện: Người dùng có thể xem và đọc blog của mình

* + 1. Viết blogs mới

Người thực hiện: Người dùng là gia đình, cá nhân, giáo viên, các em học sinh tiểu học đã có tài khoản trong hệ thống.

Quá trình thực hiện: người dùng có thể viết blogs chia sẽ những tâm sự của mình trên cộng đồng.

* + 1. Chỉnh sửa blogs

Người thực hiện: Người dùng là gia đình, cá nhân, giáo viên, các em học sinh tiểu học đã có tài khoản trong hệ thống.

Quá trình thực hiện: Chức năng chỉnh sửa blog sẽ giúp người dùng chỉnh sửa những blogs của mình.

* + 1. Bình luận ảnh, album ảnh

Người thực hiện: Người dùng là gia đình, cá nhân, giáo viên, các em học sinh tiểu học đã có tài khoản trong hệ thống.

Quá trình thực hiện: Người dùng có thể bình luận, những hình ảnh của người khác hoặc mình đưa lên.

* + 1. Đánh giá album ảnh

Người thực hiện: Người dùng là gia đình, cá nhân, giáo viên, các em học sinh tiểu học đã có tài khoản trong hệ thống.

Quá trình thực hiện: Người dùng có thế đánh giá những hình ảnh người khác đưa lên.

* + 1. Đánh dấu thông tin(từ khóa)

Người thực hiện: Người dùng là gia đình, cá nhân, giáo viên, các em học sinh tiểu học đã có tài khoản trong hệ thống.

Quá trình thực hiện: Có thế đánh giấu thông tin từ khóa cho những những nội dung người khác đưa lên.

**Những chức năng bỗ trợ**

* + 1. Học bài học

Người thực hiện: Tất cả học sinh đã đăng ký học trong hệ thống

Quá trình thực hiện: Cha mẹ học sinh sẽ hướng dẫn cho con em mình cách học tập trên hệ thống, học sinh sẽ tự học những bài học lý thuyết nhằm áp dụng vào làm bài tập kèm theo.

* + 1. Làm bài tập

Người thực hiện: Tất cả học sinh đã đăng ký học trong hệ thống

Quá trình thực hiện: Cha mẹ học sinh sẽ hướng dẫn cho con em mình cách học tập trên hệ thống, học sinh sẽ làm những bài tập kèm theo bài học lý thuyết.

* + 1. Sử dụng công cụ học tập

Người thực hiện: Tất cả học sinh đã đăng ký học trong hệ thống

Quá trình thực hiện: Cha mẹ học sinh sẽ hướng dẫn cho con em mình cách học tập trên hệ thống, học sinh sử dụng những công cụ đã được xây dựng sẵn cho quá trình học của mình.

* + 1. Chơi game vui để học

Người thực hiện: Tất cả học sinh đã đăng ký học trong hệ thống

Quá trình thực hiện: Cha mẹ học sinh sẽ hướng dẫn cho con em mình cách chơi những trò chơi mang tính chất vui để học trên hệ thống, học sinh sẽ chơi những trò chơi mà mình thích, có cập nhật kết quả chơi.

* + 1. Đọc sách

Người thực hiện: Tất cả học sinh, đã đăng ký học trong hệ thống

Quá trình thực hiện: Cha mẹ học sinh sẽ hướng dẫn cho con em mình cách đọc sách trên hệ thống, học sinh sẽ tìm những cuốn sách mà mình thích, có thể đọc trực tiếp trên hệ thống

* 1. Yêu cầu và quy định của hệ thống
     1. Danh sách các yêu cầu

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **STT** | **Mã yêu cầu** | **Tên yêu cầu** | **Loại yêu cầu** | **Mã quy định** |
| 1 | YC01 | Tạo tài khoản - Cập nhật thông tin cá nhân | Lưu trử | QD01, QD02. QD03, QD04 |
| 2 | YC02 | Đăng thông tin, cảm nhận | Lưu trử, Kết xuất | QD07, QD08 |
| 3 | YC03 | Xem thông tin , cảm nhận của bản thân hay bạn bè | Kết xuất | QD07, QD08 |
| 4 | YC04 | Xem thông tin cá nhân | Kết xuất | QD06 |
| 5 | YC05 | Chỉnh sửa thông tin cá nhân | Lưu trử | QD05 |
| 6 | YC06 | Xem thông tin cá nhân của bạn bè | Kết xuất | QD06 |
| 7 | YC07 | Tìm kiếm người quen | Lưu trử, Kết xuất | QD09 |
| 8 | YC08 | Thoát khỏi hệ thống | Tính toán | QD10 |
| 9 | YC09 | Kết bạn | Lưu trử | QD11 |
| 10 | YC10 | Xem hộp thư | Kết xuất |  |
| 11 | YC11 | Gửi thư | Lưu trử | QD12 |
| 12 | YC12 | Đọc thư | Kết xuất |  |
| 13 | YC13 | Xóa thư | Lưu trử | QD13 |
| 14 | YC14 | Xem danh sách nhóm | Kết xuất | QD14 |
| 15 | YC15 | Tạo nhóm mới | Lưu trử | QD15 |
| 16 | YC16 | Gia nhập nhóm | Lưu trử |  |
| 17 | YC17 | Xem danh sách album ảnh | Kết xuất | QD16 |
| 18 | YC18 | Tạo album ảnh mới | Lưu trử | QD16 |
| 19 | YC19 | Chỉnh sửa album ảnh | Lưu trử | QD16 |
| 20 | YC20 | Xóa album ảnh | Lưu trử | QD16 |
| 21 | YC21 | Xem danh sách blogs | Kết xuất | QD17 |
| 22 | YC22 | Viết blogs mới | Lưu trử | QD17 |
| 23 | YC23 | Chỉnh sữa blogs | Lưu trử | QD17 |
| 24 | YC24 | Bình luận ảnh, album ảnh | Lưu trử | QD16 |
| 25 | YC25 | Đánh giá album ảnh | Lưu trử | QD16 |
| 26 | YC26 | Đánh giấu thông tin (từ khóa) | Lưu trử | QD17 |
| 27 | YC27 | Học bài học | Kết xuất, lưu trử | QD18 |
| 28 | YC28 | Làm bài tập | Lưu trử, Kết xuất | QD19 |
| 29 | YC29 | Sử dụng công cụ học tập | Tính toán | QD20 |
| 30 | YC30 | Chơi game vui để học | Lưu trử, kết xuất, tính toán | QD22 |
| 31 | YC31 | Đọc sách | Kết xuất | QD21 |

* + 1. Danh sách các quy định

|  |  |  |  |
| --- | --- | --- | --- |
| **STT** | **Mã Quy định** | **Tên Quy định** | **Mô tả chi tiết** |
| 1 | QD01 | Tên đăng nhập | Tên đăng nhập không quá 50 ký tự, không chứa các ký tự đặc biệt, khoản trắng. |
| 2 | QD02 | Mật khẩu người dùng | Mật khẩu không quá 50 ký tự, có chứa mức độ phức tạp |
| 3 | QD03 | Thông tin người dùng | Những thông tin bắt buộc: email, họ tên, ngày sinh, trường lớp, những thông tin còn lại không bắt buộc |
| 4 | QD04 | Chứng thực người dùng | Chứng thực kích hoặt người dùng thông qua email đăng ký |
| 5 | QD05 | Cập nhật thông tin cá nhân | Chỉ chỉnh sửa được thông tin cá nhân của bản thân mình |
| 6 | QD06 | Xem thông tin cá nhân | Chỉ xem được thông tin của mình, hay bạn bè nếu họ cho phep. |
| 7 | QD07 | Đăng trạng thái | Kiểm tra nội dung đăng, đăng được nhiều thể loại |
| 8 | QD08 | Đăng cảm nhận | Kiểm tra nội dung đăng, đăng được nhiều thể loại |
| 9 | QD09 | Tìm kiếm người quen | Phải có thông tin tìm kiếm, thông tin thuộc nhiều loại khác nhau |
| 10 | QD10 | Đăng xuất | Quá thời gian session, khi người dùng muốn đăng xuất |
| 11 | QD11 | Kết bạn | Bắt buộc phải có thông tin cua người muốn kết bạn như email, tên đăng nhập … |
| 12 | QD12 | Gửi thư | Phải có địa chỉ của người nhận, có thể gửi cho nhiều người, có thể lưu thư nháp. |
| 13 | QD13 | Xóa thư | Chọn thư cần xóa, hỏi trước khi xóa |
| 14 | QD14 | Lựa chọn nhóm tham gia | Nếu thích một nhóm nào đó, người dùng chỉ việc tham gia, nhưng để kết bạn với thành viên trong nhóm đó thì phải được chính người đó đồng ý. |
| 15 | QD15 | Tạo nhóm mới | Tạo nhóm mơi, phải có đặc trưng cụ thể của nhóm đó, mời bạn bè tham gia, người nhận lời mời có thể đồng ý hay từ chối. |
| 16 | QD16 | Tạo chỉnh sửa album ảnh | Người dùng chỉ có quyền tạo hay chỉnh sửa trên trang của chính mình |
| 17 | QD17 | Tạo, viết, chỉnh sửa, xóa, hay phân quyền blogs | Người dùng chỉ có quyền tạo hay chỉnh sửa trên trang của chính mình |
| 18 | QD18 | Học bài mới | Học trong phạm vi trình độ của mình |
| 19 | QD19 | Làm bài tập | Làm bài tập khi học bài xong, có thể làm bài tập nâng cao hay những bài kiểm tra nếu học sinh học xong một chương trình nào đó. |
| 20 | QD20 | Công cụ học tập | Những công cụ có thể vừa học vừa chơi, có thể sử dụng công cụ trong khi học bài hay làm bài tập. |
| 21 | QD21 | Đọc sách | Nếu là học sinh chỉ đọc được sách trong phạm vi của mình. |
| 22 | QD22 | Chơi để học | Những game phải bổ ích mang tính học thuật, vừa giúp các em vui chơi giải trí sau những giờ học căng thẳng. |

* 1. Danh sách người dùng hệ thống

1. Cha mẹ học sinh
2. Học sinh tiểu học
3. Giáo viên tiểu học
4. Người dùng thông thường khác
5. Người quản trị hệ thống.

Chương 3: Phân tích hệ thống

* 1. **Sơ đồ Use case**
     1. **Danh sách use case**

|  |  |
| --- | --- |
| **Ký hiệu** | **Tên Use case** |
| UC01 | Đăng ký |
| UC02 | Đăng nhập |
| UC03 | Quản lý thông tin cá nhân |
| UC04 | Quản lý trạng thái cá nhân |
| UC05 | Quản lý bạn bè |
| UC06 | Quản lý blogs |
| UC07 | Quản lý hộp thư |
| UC08 | Quản lý hình ảnh (album ảnh) |
| UC09 | Bình luận, cảm nhận |
| UC10 | Đánh giá |
| UC11 | Đánh dấu tư khóa |
| UC12 | Vui chơi |
| UC13 | Tìm hiểu trò chơi |
| UC14 | Xem hướng dẫn trò chơi |
| UC15 | Chơi trò chơi |
| UC16 | Lưu kết quả |
| UC17 | Học tập |
| UC18 | Chọn trình độ |
| UC19 | Chọn cấp độ |
| UC20 | Chọn bài học |
| UC21 | Học bài |
| UC22 | Làm bài |
| UC23 | Thoát khỏi hệ thống |

* + 1. **Sơ đồ use case**
* Sơ đồ chi tiết use case

Mạng gia đình:



Học tập:



Vui choi:



* Sơ đồ use case tổng quát



* + 1. **Mô tả chi tiết use case**
  + Để mô tả chi tiết cho từng use case ta sẽ sử dụng sơ đồ hoặt động trong phần 3.2. **Sơ đồ hoặt động**

|  |  |
| --- | --- |
| **Ký hiệu** | **Tên Use Case** |
| **UC01** | **Đăng ký** |
|  | |
| **UC02** | **Đăng nhập** |
|  | |
| **UC03** | **Quản lý thông tin cá nhân** |
|  | |
| **UC04** | **Quản lý trạng thái cá nhân** |
|  | |
| **UC05** | **Quản lý bạn bè** |
|  | |
| **UC06** | **Quản lý blogs** |
|  | |
| **UC07** | **Quản lý hộp thư** |
|  | |
| **UC08** | **Quản lý hình ảnh (album ảnh)** |
|  | |
| **UC09** | **Bình luận, cảm nhận** |
|  | |

* 1. **Sơ đồ lớp**
     1. **Sơ đồ khái niệm**



* + 1. **Danh sách lớp và thuộc tính**

Những lớp thuộc khu vực trò chơi, khu vục học tập. Được kế thừa từ những hệ thống củ trước đó nên không được phân tích chi tiết trong hệ thống này.





* + - 1. **Mô tả thuộc tính**
* **Account: Người dùng**

|  |  |  |
| --- | --- | --- |
| **Tên thuộc tính** | **Kiểu thuộc tính** | **Mô tả** |
| AccountID | Int | Mã người dùng |
| BirthDate | Datetime | Ngày sinh của người dùng |
| CreateDate | Datetime | Ngày đăng ký |
| Email | String | Email của người dùng |
| FirstName | String | Tên người dùng |
| LastName | String | Họ người dùng |
| LastUpdateDate | Date | Lần cập nhật trước đó |
| PassWord | String | Mật khẩu |
| AgreedToTermDate | Date | Ngày chấp nhận điều khoản |
|  | | |

* **Profile: Thông tin người dùng**

|  |  |  |
| --- | --- | --- |
| **Tên thuộc tính** | **Kiểu thuộc tính** | **Mô tả** |
| Address | String | Địa chỉ người dùng |
| Avatar | Binary | Ảnh đại diện |
| CreateDate | Date | Ngày tạo profile |
| Enjoy | String | Sở thích |
| SchoolsName | String | Têm trường học |
| Signature | String | Chử ký cá nhân |
| UserGravatar | Bool | Tình trạng sử dụng hình ảnh dại diện |
|  | | |

* **Friends: Bạn bè**

|  |  |  |
| --- | --- | --- |
| **Tên thuộc tính** | **Kiểu thuộc tính** | **Mô tả** |
| FriendID | Int | Mã bạn bè |
| MyFriendAccountID | Int | Mã người dùng của bạn bè |
| CreateDate | Datetime | Ngày kết bạn |
|  | | |

* **Comments: Bình luận, cảm nhận**

|  |  |  |
| --- | --- | --- |
| **Tên thuộc tính** | **Kiểu thuộc tính** | **Mô tả** |
| CommentID | Int | Mã bình luận, cảm nhận |
| Body | String | Nội dung bình luận |
| CreateDate | Datetime | Ngày bình luận |
| CommentByAccountID | Int | Mã người bình luận |
| CommentByUsername | String | Tên người bình luận |
| SystemObjectID | Int | Đối tượng được bình luận |
|  | | |

* **Tags: Đánh dấu từ khóa**

|  |  |  |
| --- | --- | --- |
| **Tên thuộc tính** | **Kiểu thuộc tính** | **Mô tả** |
| TagID | Int | Mã đánh dấu từ khóa |
| Name | String | Tên được đánh dấu |
| Count | Int | Số lượng đối tượng đã được đánh dấu |
| CreateDate | Datetime | Ngày đánh dấu |
| SystemObject | Int | Mã đối tượng được đánh dấu |
|  | | |

* **Ratings: Đánh giá**

|  |  |  |
| --- | --- | --- |
| **Tên thuộc tính** | **Kiểu thuộc tính** | **Mô tả** |
| RatingID | Int | Mã đánh giá |
| Score | Int | Số điểm đánh giá |
| CreateByUsername | String | Tên người đánh giá |
| CreateByUserAccountID | Int | Mã người đánh giá |
| CreateDate | Datetime | Ngày được đánh giá |
| SystemObjectID | Int | Mã đối tượng được đánh giá |
|  | | |

* **Folder: Thư mục hình ảnh**

|  |  |  |
| --- | --- | --- |
| **Tên thuộc tính** | **Kiểu thuộc tính** | **Mô tả** |
| FolderID | Int | Mã thư mục hình ảnh |
| Name | String | Tên thư mục hình ảnh |
| IsPublicResource | Bool | Có public hay không |
| CreateDate | Datetime | Ngày tạo thư mục hình ảnh |
| Description | String | Mô tả thư mục hình ảnh |
| Location | String | Vị trí, nơi chụp ảnh |
|  | | |

* **Files: Tệp tin**

|  |  |  |
| --- | --- | --- |
| **Tên thuộc tính** | **Kiểu thuộc tính** | **Mô tả** |
| FileID | Int | Mã tệp tin |
| FileSystemName | String | Tên tập tin hệ thống |
| FileName | String | Tên tập tin |
| CreateDate | Datetime | Ngày tạo tập tin |
| AccountID | Int | Mã người dùng |
| IsPublicResource | Bool | Có public hay không |
| Description | String | Mô ta tập tin |
| Size | Double | Kích thước tập tin |
|  | | |

* **Alerts: Trạng thái, thông tin chia sẽ của người dùng**

|  |  |  |
| --- | --- | --- |
| **Tên thuộc tính** | **Kiểu thuộc tính** | **Mô tả** |
| AlertID | Int | Mã Alert |
| CreateDate | Datetime | Ngày tạo Alert |
| Ishidden | Bool | Có ẩn hay không |
| Mesage | String | Nội dung Alert |
|  | | |

* **Blogs: Nhật ký cá nhân**

|  |  |  |
| --- | --- | --- |
| **Tên thuộc tính** | **Kiểu thuộc tính** | **Mô tả** |
| BlogID | Int | Mã nhật ký |
| Title | String | Tiêu đề trang |
| Subject | String | Tên bài nhật ký |
| Port | String | Nội dung bài viết |
| CreateDate | Datetime | Ngày viết nhật ký |
| UpdateDate | Datetime | Ngày cập nhật |
| IsPublished | Bool | Cho phép xuất bản |
|  | | |

* **SystemObject: Hệ thống các đối tượng**

|  |  |  |
| --- | --- | --- |
| **Tên thuộc tính** | **Kiểu thuộc tính** | **Mô tả** |
| SystemObjectID | Int | Mã hệ thống các đối tượng |
| Name | String | Tên hệ thống các đối tượng |
|  | | |

* **Messages: Hộp thư chia sẽ tin tức kết bạn**

|  |  |  |
| --- | --- | --- |
| **Tên thuộc tính** | **Kiểu thuộc tính** | **Mô tả** |
| MessageID | Int | Mã hộp thư |
| Subject | String | Chủ đề bức thư |
| Body | String | Nội dung bức thư |
| CreateDate | Datetime | Ngày viết |
| SendByAccountID | Int | Mã Người gửi |
|  | | |

* **MessageType: Loại thư gửi**

|  |  |  |
| --- | --- | --- |
| **Tên thuộc tính** | **Kiểu thuộc tính** | **Mô tả** |
| MessageTypeID | Int | Mã loại thư |
| Name | String | Tên loại |
|  | | |

* **MessgeRecipients: Thư nhận**

|  |  |  |
| --- | --- | --- |
| **Tên thuộc tính** | **Kiểu thuộc tính** | **Mô tả** |
| MessageRecipients | Int | Mã thư nhận |
| MessageID | Int | Mã thư |
| AccountID | Int | Mã người nhận |
|  | | |

* + - 1. **Mô tả lớp**

**Danh sách lớp:**

|  |  |  |  |
| --- | --- | --- | --- |
| **STT** | **Kiểu trả về** | **Tên lớp** | **Mô tả** |
| 1 | Public class | Account | Khởi tạo đối tượng Người dùng |
| 2 | Public class | Profile | Khởi tạo đối tượng Người dùng |
| 3 | Public class | Friends | Khởi tạo đối tượng Người dùng |
| 4 | Public class | Comments | Khởi tạo đối tượng Người dùng |
| 5 | Public class | Tags | Khởi tạo đối tượng Người dùng |
| 6 | Public class | Ratings | Khởi tạo đối tượng Người dùng |
| 7 | Public class | Folder | Khởi tạo đối tượng Người dùng |
| 8 | Public class | Files | Khởi tạo đối tượng Người dùng |
| 9 | Public class | Alerts | Khởi tạo đối tượng Người dùng |
| 10 | Public class | Blogs | Khởi tạo đối tượng Người dùng |
| 11 | Public class | SystemObject | Khởi tạo đối tượng Người dùng |
| 12 | Public class | Messages | Khởi tạo đối tượng Người dùng |
| 13 | Public class | MessageType | Khởi tạo đối tượng Người dùng |
| 14 | Public class | MessgeRecipients | Khởi tạo đối tượng Người dùng |

**Danh sách các phương thức của lớp:**

* **Account**

|  |  |  |  |
| --- | --- | --- | --- |
| **STT** | **Kiểu trả về** | **Tên phương thức** | **Mô tả** |
| 1 |  | AddPermission(Permission permission) | Cấp quyền cho người dùng |
| 2 | Account | GetAccountByUsername(string username) | Lấy tài khoản người dùng bằng tên đăng nhập |
| 3 | Account | GetAccountByEmail(string email) | Lấy tài khoản người dùng bằng Email |
| 4 | Void | SaveAccount(Account account) | Lưu tài khoản người dùng |
| 5 | Account | GetAccountByID(int AccountID) | Lấy tài khoản người dùng bằng ID |
| 6 | List<Account> | SearchAccounts(string SearchText) | Tìm kiếm danh sách người dùng bằng thông tin nhập vào. |
| 7 | List<Account> | GetApprovedAccountsByGroupID(int GroupID, int PageNumber) | Lấy danh sách người dùng theo nhóm và lấy theo trang |
| 8 | List<Account> | GetAccountsToApproveByGroupID(int GroupID) | Lấy danh sách người dùng theo nhóm. |

* **Profile**

|  |  |  |  |
| --- | --- | --- | --- |
| **STT** | **Kiểu trả về** | **Tên phương thức** | **Mô tả** |
| 1 | Profile | GetProfileByAccountID(int accountID) | Lấy thông tin cá nhân người dùng theo AccountID |
| 2 | int | SaveProfile(Profile profile) | Thêm mới thông tin cá nhân của người dùng |

* **Friend**

|  |  |  |  |
| --- | --- | --- | --- |
| **STT** | **Kiểu trả về** | **Tên phương thức** | **Mô tả** |
| 1 | List<Friend> | GetFriendsByAccountID(Int32 AccountID) | Lấy danh sách bạn bè của người dùng theo AccountID |
| 2 | List<Account> | GetFriendsAccountsByAccountID(int32 AccountID) | Lấy danh sách tài khoản người dùng bạn bè theo AccountID |
| 3 | List<Account> | GetFriendsAccountsByAccountID(Int32 AccountID, int currentPage, int pageSize) | Lấy danh sách tài khoản người dùng bạn bè theo AccountID và có phân trang |
| 4 | void | DeleteFriend(Friend friend) | Xóa bạn bè |
| 5 | void | DeleteFriendByID(Int32 AccountIDToRemoveFriendFrom, Int32 FriendIDToRemove) | Xóa bạn bè theo AccontID |
| 6 | void | SaveFriend(Friend friend) | Thêm bạn mới |

* **Comments**

|  |  |  |  |
| --- | --- | --- | --- |
| **STT** | **Kiểu trả về** | **Tên phương thức** | **Mô tả** |
| 1 | Comment | GetCommentByID(long CommentID) | Lấy bình luận theo CommentID |
| 2 | List<Comment> | GetCommentsBySystemObject(int SystemObjectID, long SystemObjectRecordID) | Lấy danh sách bình luận theo của một đối tượng |
| 3 | List<Comment> | GetMoreCommentsBySystemObject(int SystemObjectID, long SystemObjectRecordID, int More) | Lấy danh sách bình luận của một đối tượng theo phân trang |
| 4 | List<Comment> | GetTopCommentsBySystemObject(int SystemObjectID, long SystemObjectRecordID, int More) | Lấy những bình luận đầu tiên của một đối tượng. |
| 5 | List<Comment> | GetTopCommentsBySystemObjectAjax(int SystemObjectID, long SystemObjectRecordID, out int More) | Lấy những bình luận đầu tiên của một đối tượng |
| 6 | int | CountMore(int SystemObjectID, long SystemObjectRecordID) | Đếm số bình luận của một đối tượng |
| 7 | long | SaveComment(Comment comment) | Thêm một bình luân mới |
| 8 | void | DeleteComment(Comment comment) | Xóa một bình luận |

* **Tags**

|  |  |  |  |
| --- | --- | --- | --- |
| **STT** | **Kiểu trả về** | **Tên phương thức** | **Mô tả** |
| 1 | List<Tag> | GetTagsBySystemObjectAndRecordID(int SystemObjectID, long SystemObjectRecordID) | Lấy danh sách từ khóa của một đối tượng cụ thể |
| 2 | List<Tag> | GetTagsBySystemObject(int SystemObjectID, int TagsToTake) | Lấy danh sách từ khóa của một đối tượng |
| 3 | List<Tag> | GetTagsGlobal(int TagsToTake) | Lấy danh sách từ khóa toàn cục |

* **Rating**

|  |  |  |  |
| --- | --- | --- | --- |
| **STT** | **Kiểu trả về** | **Tên phương thức** | **Mô tả** |
| 1 | bool | HasRatedBefore(int SystemObjectID, long SystemObjectRecordID, int AccountID) | Kiểm trả đã đánh giá chưa |
| 2 | int | GetCurrentRating(int SystemObjectID, long SystemObjectRecordID) | Lấy đánh giá hiện hành |
| 3 | void | SaveRatings(List<Rating> ratings) | Lưu đánh giá mới |

* **Folder**

|  |  |  |  |
| --- | --- | --- | --- |
| **STT** | **Kiểu trả về** | **Tên phương thức** | **Mô tả** |
| 1 | List<Folder> | GetFoldersByAccountID(Int32 AccountID) | Lấy danh sách Album của người dùng |
| 2 | List<Folder> | GetFriendsFolders(List<Friend> Friends) | Lấy danh sách Album ảnh của bạn bè của người dùng |
| 3 | Folder | GetFolderByID(Int64 FolderID) | Lấy một Album theo FolderID |
| 4 | Int64 | SaveFolder(Folder folder) | Thêm Album ảnh mới |
| 5 | Int64 | UpdateFolder(Folder folder) | Cập nhật Album ảnh |
| 6 | void | DeleteFolder(Folder folder) | Xóa Album ảnh |

* **File**

|  |  |  |  |
| --- | --- | --- | --- |
| **STT** | **Kiểu trả về** | **Tên phương thức** | **Mô tả** |
| 1 | bool | IsFlagged(int SystemObjectID, Int64 SystemObjectRecordID) | Ảnh đã được đánh dấu hay chưa |
| 2 | File | GetFileByID(Int64 FileID) | Lấy ảnh theo FileID |
| 3 | File | GetFileByFileSystemName(Guid FileSystemName) | Lấy ảnh theo tên được mã hóa |
| 4 | List<File> | GetFilesByFolderID(Int64 FolderID) | Lấy danh sách ảnh theo Album ảnh |
| 5 | void | UpdateDescriptions(Dictionary<long, string> fileDescriptions) | Cập nhật mô tả cho Ảnh |
| 6 | void | DeleteFilesInFolder(Folder folder) | Xóa tất cả ảnh của một album |
| 7 | void | DeleteFile(File file) | Xóa ảnh |
| 8 | void | DeleteFileFromFileSystem(Folder folder, File file) | Xóa ảnh theo tên mã hóa |

* **Alerts**

|  |  |  |  |
| --- | --- | --- | --- |
| **STT** | **Kiểu trả về** | **Tên phương thức** | **Mô tả** |
| 1 | List<Comment> | GetTopComments() | Lấy danh sách bình luận |
| 2 | Void | SaveAlert(Alert alert) | Tạo mới một thông báo |
| 3 | List<Alert> | GetAlertsByAccountID(int accountID) | Lấy danh sách thông báo theo AccountID |
| 4 | List<Alert> | GetAlertsByAccountID(int accountID, int skip) | Lấy danh sách thông báo theo AccountID và phân trang |
| 5 | Int32 | CountAlertsByAccountID(int accountID) | Đếm số lượng thông báo |
| 6 | List<Alert> | GetAlertsByAccountID(int accountID, int currentIndex, int itemNum) | Lấy danh sách thông báo theo AccountID và phân trang |
| 7 | List<Alert> | GetAlertByAccountIDofMeAndFriend(int accountID) | Lấy danh sách thông báo không phải của người dùng và bạn của họ |
| 8 | bool | CheckFriendByAccount(List<Friend> list, int accID) | Kiểm tra người dùng khác là bạn |
| 9 | bool | CheckAlert(List<Alert> list, Alert i) | Kiểm tra thông báo có trong một danh sách thông báo có sẵn |

* **Blog**

|  |  |  |  |
| --- | --- | --- | --- |
| **STT** | **Kiểu trả về** | **Tên phương thức** | **Mô tả** |
| 1 | List<Blog> | GetBlogsForIndexing(int PageNumber) | Lấy danh sách blog của tất cả theo phân trang |
| 2 | Blog | GetBlogByPageName(string PageName, Int32 AccountID) | Lấy Blog theo tên |
| 3 | List<Blog> | GetLatestBlogs() | Lấy danh sách những blog mới nhất |
| 4 | PagedList<Blog> | GetLatestBlogsPaging(int currentPage, int pageSize) | Lấy danh sách blog của tất cả người dùng theo phân trang |
| 5 | List<Blog> | GetBlogsByAccountID(Int32 AccountID) | Lấy danh sách blog của một người dùng theo AccountID |
| 6 | PagedList<Blog> | GetBlogsByAccountIDPaging(Int32 AccountID, int currentpage, int pagesize) | Lấy danh sách blog của một người dùng theo AccountID và phân trang |
| 7 | Blog | GetBlogByBlogID(Int64 BlogID) | Lấy Blog theo BlogID |
| 8 | bool | CheckPageNameIsUnique(Blog blog) | Kiểm tra tên trang của một Blog đã được dùng chưa |
| 9 | Blog | CleanPageName(Blog blog) | Xóa tên trang của môt Blog |
| 10 | Int64 | SaveBlog(Blog blog) | Thêm một blog mới |
| 11 | void | DeleteBlog(Int64 BlogID) | Xóa blog theo blogID |
| 12 | void | DeleteBlog(Blog blog) | Xóa blog theo Blog |

* **SystemObject**

|  |  |  |  |
| --- | --- | --- | --- |
| **STT** | **Kiểu trả về** | **Tên phương thức** | **Mô tả** |

* **Messages**

|  |  |  |  |
| --- | --- | --- | --- |
| **STT** | **Kiểu trả về** | **Tên phương thức** | **Mô tả** |
| 1 | int | GetPageCount(MessageFolders messageFolder, Int32 RecipientAccountID) | Đếm thư của một Folder thư |
| 2 | List<MessageWithRecipient> | GetMessageByAccountID(Int64 AccountID, Int32 PageNumber, MessageFolders Folder) | Lấy danh sách thư của người dùng theo AccountID và phân trang |
| 3 | MessageWithRecipient | GetMessageByMessageID(Int32 MessageID, Int32 RecipientAccountID) | Lấy thư theo MessageID |
| 4 | void | DeleteMessage(Int32 messageID) | Xóa một thư |

* **MessageType**

|  |  |  |  |
| --- | --- | --- | --- |
| **STT** | **Kiểu trả về** | **Tên phương thức** | **Mô tả** |

* **MessgeRecipients**

|  |  |  |  |
| --- | --- | --- | --- |
| **STT** | **Kiểu trả về** | **Tên phương thức** | **Mô tả** |
| 1 | List<MessageRecipient> | GetMessageRecipientsByMessageID(Int32 MessageID) | Lấy danh sách thư theo MessageID |
| 2 | MessageRecipient | GetMessageRecipientByID(Int32 MessageRecipientID) | Lấy thư theo MessageRecipientID |
| 3 | void | DeleteMessageRecipient(MessageRecipient messageRecipient) | Xóa thư |

* 1. **Sơ đồ tuần tự**
     1. **Ca sử dụng đăng ký UC01**



* + 1. **Ca sử dụng đăng nhập UC02**

1. 
   * 1. **Ca sử dụng Quản lý thông tin cá nhân UC03**

Trong Ca sử dụng quản lý thông tin cá nhân có hai ca sử dụng nhỏ hơn đó là xem thông tin cá nhân và chỉnh sủa thông tin cá nhân.

* Biểu đồ tuần tự của hoạt động xem thông tin cá nhân.



Biểu đồ tuần tự cho hoạt động chỉnh sửa thông tin cá nhân



* + 1. **Ca sử dụng quản lý trạng thái cá nhân UC04**

Trong ca sử dụng này có các hoạt đông con đó là xem trạng thái cá nhân và thêm trạng thái.

Biều đồ tuần tự của hoạt động xem trạng thái cá nhân:

Biểu đồ tuần tự của hoạt động thêm trạng thái cá nhân:



* + 1. **Ca sử dụng Quản lý bạn bè UC05**



* + 1. **Ca sử dụng Quản lý Blogs UC06**

Trong ca sử dụng Quản lý Blogs có các hoạt động con đó là xem danh sách blog, viết blogs, xem danh sách blogs của tôi.

Biểu đồ tuần tự của hoạt động viết blog và xem danh sách blog của tôi:



Biểu đồ tuần tự của hoạt động xem danh sách Blogs:



* + 1. **Ca sử dụng quản lý hộp thư UC07**

Trong ca sử dụng Quản lý hộp thư có các hoạt động con đó là viết thư mới, đọc thư, xem danh sách thư.

Biểu đồ tuần tự của hoạt động viết thư mới:



Biểu đồ tuần tự của hoạt động đọc thư:



Biểu đồ tuần tự của hoạt động xem danh sách thư:



* + 1. **Ca sử dụng Quản lý hình ảnh UC08**

Trong ca sử dụng Quản lý hình ảnh có các hoạt động con đó là Xem danh sách Album ảnh của bạn của, xem danh sách album ảnh của mình, thêm sửa album.

Biểu đồ tuần tự của hoạt động xem danh sách Album ảnh của mình:



Biểu đồ tuần tự của hoạt động xem danh sách Album ảnh của bạn bè:



Biểu đồ tuần tự của hoạt động thêm hoặc sửa Album ảnh:



* + 1. **Ca sử dụng Bình luận UC09**



* + 1. **Ca sử dụng Đánh giá UC10**



* + 1. **Ca sử dụng Đánh dấu từ khóa UC11**



Chương 4: Thiết kế và cài đặt

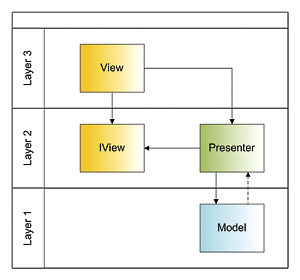
* 1. Nền tảng sử dụng
     1. **Giới thiệu mô hình xây dựng website theo kiến trúc MVP**

Mô hình MVP gồm ba thần phần chính View, Model và Presenter.

Model chứa dữ liệu và các tính toán xử lý logic để giải quyết vấn đề mà Website hướng tới.

View là thành phần đảm nhận việc thể hiện những dữ liệu của Model và là tổng hợp của các form, control được sử dụng.

Presenter là thành phần đảm nhận các xử lý thể hiện cũng như tương tác đến dữ liệu bên dưới và có thể tương tác để thay đổi View trong quá trình xử lý.



Trong mô hình MVP, các lớp View sẽ được sử dụng thông qua một Interface được định nghĩa trong .NET. Các lớp Presenter tương ứng sẽ sử dụng Interface này để đọc và ghi dữ liệu lên trên các View. Trong đa số các cách implement, một View sẽ có một Presenter tương ứng của nó. View sẽ khởi tạo Presenter cho nó và truyền cho Presenter này tham chiếu đến chính nó. Khi một event nào trên view được kích hoạt chẳng hạn như button\_clicked, text\_changed,.. bản thân lớp View sẽ không làm gì cả mà sẽ để cho lớp Presenter xử lý những sự kiện đó. Presenter sẽ đọc dữ liệu từ View (vì nó giữ một instance của View như là một member trong class) thông qua View Interface, thực hiện những xử lý ứng với Event được kích hoạt và set những thay đổi từ dữ liệu Model lên trên View thông qua View Interface.

Trong môi trường .NET, cùng một Presenter có thể được sử dụng cho View trên web như các trang ASP.NET hoặc được sử dụng cho các Form trong Windows Form Application. Các Presenter đọc và ghi dữ liệu thông qua một Interface của .NET nên nó hoàn toàn độc lập với layer View. Chính nhờ cách làm này mà ta có thể áp dụng Unit Test cho các lớp xử lý Presenter rất dễ dàng và nó cũng chính là một trong những lợi ích lớn nhất của MVP cho khả năng tái sử dụng.

**Các nhược điểm của MVP**

Có nhiều lớp, nhiều layer phải thực thi; có nghĩa độ phức tạp của ứng dụng sẽ tăng

Chúng ta phải nghĩ ra một cách để kết hợp giữa các Views và các Presenters. Trong ASP.NET thì các lớp Code-Behind sẽ implement một interface IView nào đó và sử dụng một member Presenter.

Mô hình Model không biết gì về Presenter. Vì thế nếu dữ liệu Model bị thay đổi từ một chương trình khác thay vì bởi Presenter thì phải nghĩ ra một cách để Presenter biết. Các bạn có thể implement chức năng này nếu cần bằng cách sử dụng các events, hoặc tham khảo thiết kế Observer Pattern.

* + 1. **Công cụ StructureMap**

StructureMap là công cụ thường được dùng cho các hệ thống lớn. Nhằm mục đích nâng cao tính linh hoạt, tính mở của hệ thống, khả năng tái sử dụng mã. StructureMap sẽ tham chiếu đến Interface của một lớp thay vì sử dụng chính lớp đó. Do vậy khi thay đổi một đối tượng ta chỉ cần tham chiếu đến interface của nó trong StructureMap. Chúng ta có thể sử dụng StructureMap khi hệ thống cần những yêu cầu như: khả năng mở rộng hệ thống, có một công cụ cấu hình chung, hệ thống sử dụng Test-Driven Development hoặc muốn test ứng dụng tự động…

* 1. **Quá trình thiết kế**
     1. Thiết kế database
        1. Mô hình đồ ER
* Account



* Friends



* Alert s, StatusUpdate, Blogs



* Coments, Tags, Ratings



* Group



* Messages



* Files, Folder



* + - 1. Mô hình quan hệ
* Account



* Friends



* Alert s, StatusUpdate, Blogs



* Coments, Tags, Ratings



* Group



* Messages



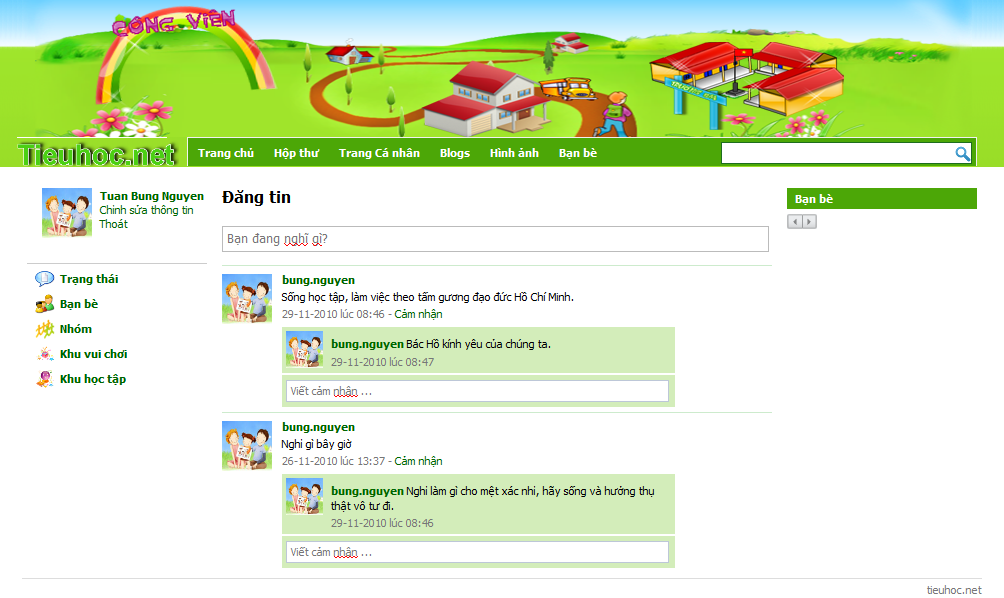
* Files, Folder



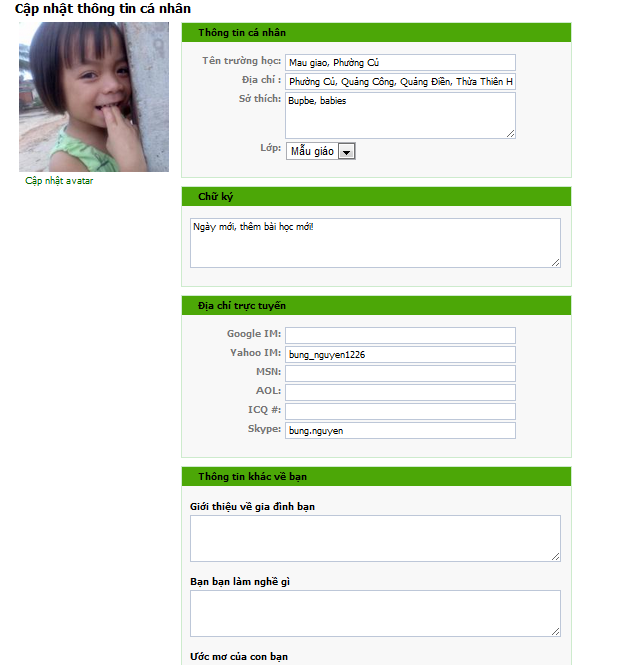
* + 1. **Thiết kế giao diện**
* Giao diện đăng ký, đăng nhập



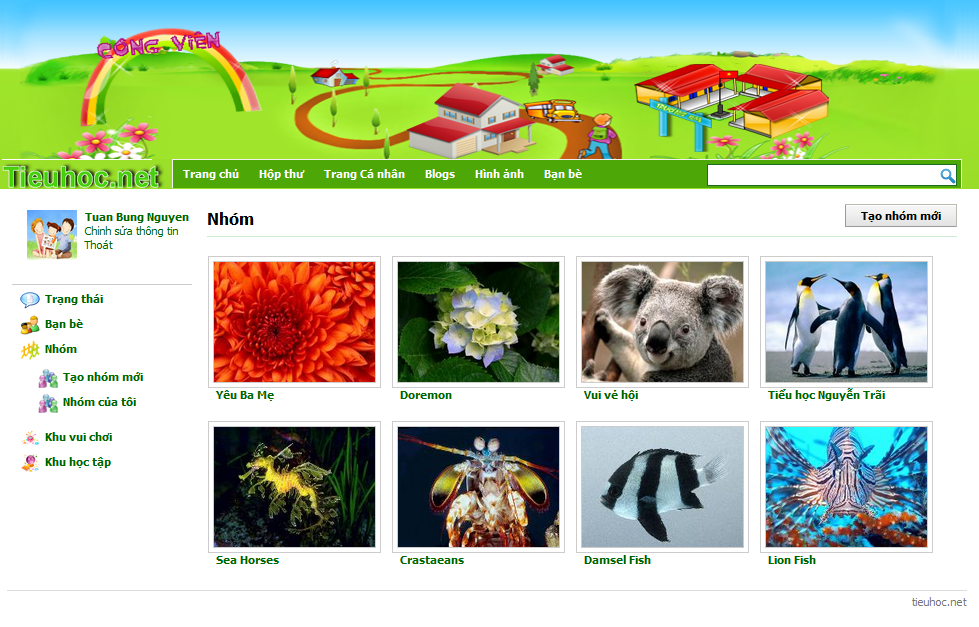
* Giao diện trang chủ



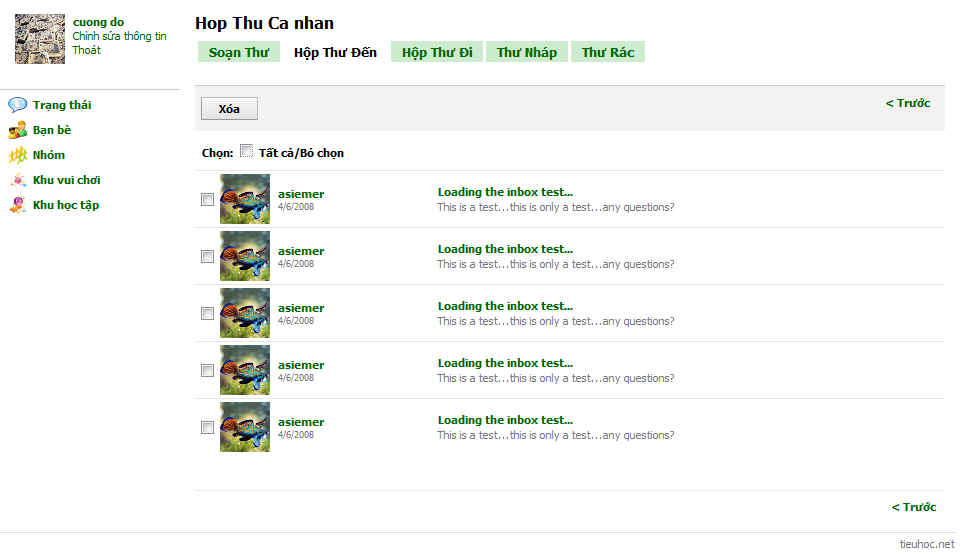
* Chỉnh sữa thông tin cá nhân



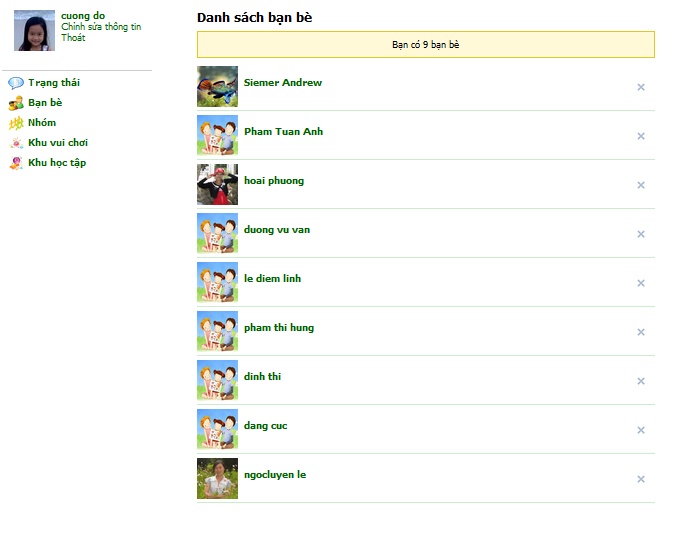
* Danh sách nhóm



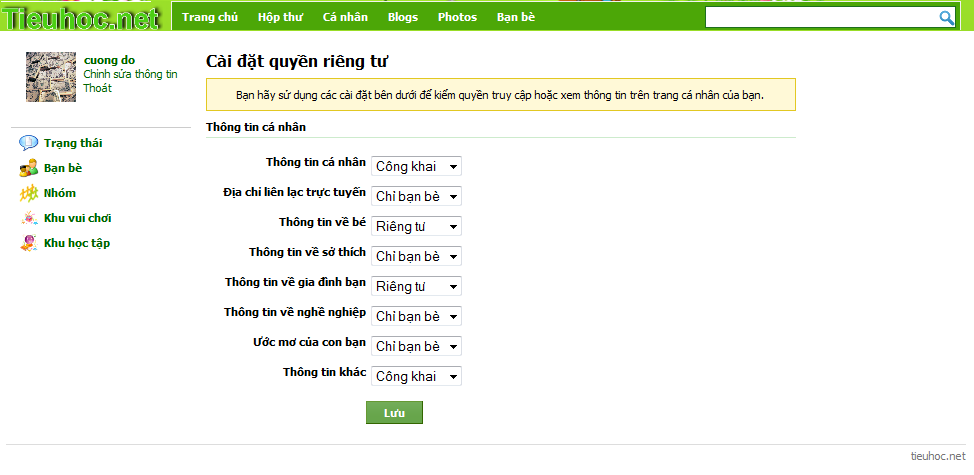
* Hộp thư cá nhân



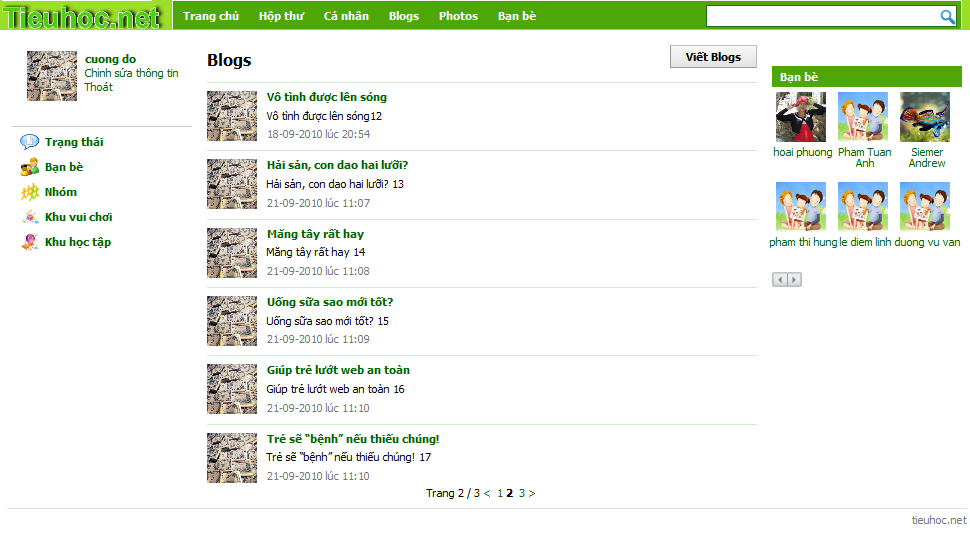
* Danh sách bạn bè



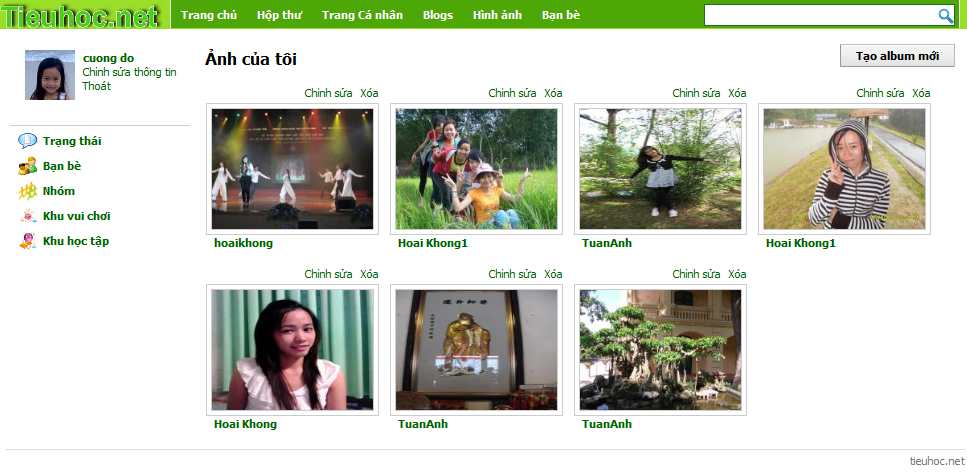
* Quyền riêng tư



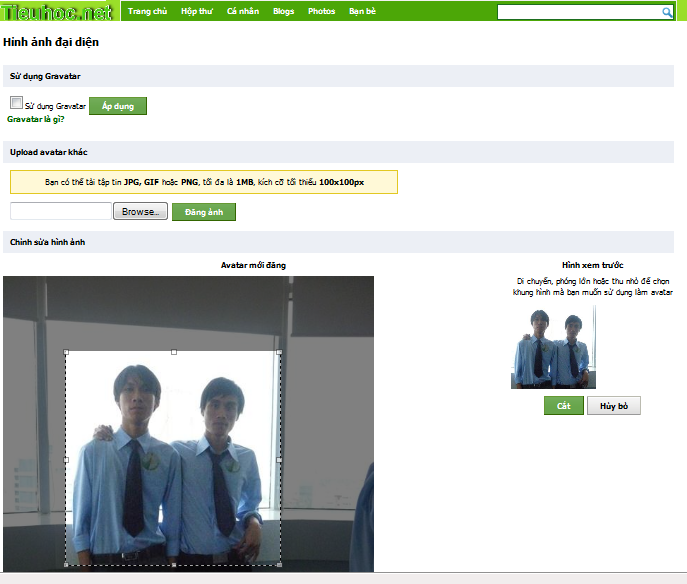
* Blogs cá nhân



* Hình ảnh cá nhân



* Chức năng cập nhật avatar



* + 1. **Thiết kế xử lý**
       1. Kiến trúc hệ thống

Data Access Layer

Business Layer

Presentation Layer

Service Layer

Database

* Tầng Presentation

Tầng này làm nhiệm vụ giao tiếp với người dùng cuối để thu thập dữ liệu và hiển thị kết quả hoặc dữ liệu thông qua các thành phần trong giao diện người dùng. Tầng này sử dụng các dịch vụ do tầng Business hoặc Service cung cấp.

* Tầng service

Tầng này chịu trách nhiệm liên kết các đối tượng các thành phần trong tầng Business với nhau. Tạo ra các dịch vụ để các thành phần trên tầng Presentation có thể sử dụng và hiển thị cho người dùng.

* Tầng Business

Tầng này thực hiện các nghiệp vụ chính của hệ thống, sử dụng các dịch vụ do tầng DataAcess cung cấp, và cung cấp các dịch vụ cho tầng Presentation và service. Tầng này cũng có thể sử dụng các dịch vụ của nhà cung cấp khác để thực hiện công việc của mình. Trong tầng này có các thành phần chính như là Business Entities và Business Component. Có thể xem tầng Service phía trên là một phần của tầng business.

* Tầng Data Access

Tầng này thực hiện các nghiệp vụ liên quan đến lưu trữ và truy xuất dữ liệu của ứng dụng, thường lớp này sẽ sử dụng các dịch vụ của các hệ quản trị cơ sở dữ liệu như SQL Server, Oracle, để thực hiện nhiệm vụ của mình.

* Cơ sở dữ liệu

Mọi dữ liệu mọi thông tin của người dùng sẽ được lưu xuống cơ sở dữ liệu. Trong chương trình sẽ sử dụng hệ quản trị cơ sở dữ liệu SQL Server của Microsoft.

* + - 1. Một số thuật toán trong chương trình

Cắt hình đại diện

Khi upload một hình lên trên server hệ thống cho phép người dùng cắt hình ảnh của mình theo nhưng chi tiết nhỏ hơn. Ở đây chúng ta sẽ sử dụng thư viện javascript để tìm vị trí cắt ảnh:

<script type="text/javascript" src="/js/cropper/lib/prototype.js"

language="javascript"></script>

<script type="text/javascript"

src="/js/cropper/lib/scriptaculous.js?

load=builder,dragdrop" language="javascript"></script>

<script type="text/javascript" src="/js/cropper/cropper.js"

language="javascript"></script>

Đoạn scritp chọn vị trí cắt ảnh ở client:

<script type="text/javascript" language="javascript">

Event.observe( window, 'load', function() {

new Cropper.ImgWithPreview(

'ctl00\_Content\_imgCropImage',

{

previewWrap: 'previewWrap',

minWidth: 100,

minHeight: 100,

ratioDim: {x: 100,y: 100},

displayOnInit: true,

onEndCrop: onEndCrop

}

);

} );

</script>

Chúng ta sẽ tìm dùng các trường ẩn để lưu lại các vị trí cần cắt của ảnh

<asp:HiddenField ID="hidX1" runat="server" />

<asp:HiddenField ID="hidY1" runat="server" />

<asp:HiddenField ID="hidX2" runat="server" />

<asp:HiddenField ID="hidY2" runat="server" />

<asp:HiddenField ID="hidWidth" runat="server" />

<asp:HiddenField ID="hidHeight" runat="server" />

Hàm javascript dưới đây sẽ tự động cập nhật vị trí của các điểm ảnh sẽ bị cắt khi người dùng thay đổi vùng chọn

<script type="text/javascript">

function onEndCrop( coords, dimensions )

{

$( 'ctl00\_Content\_hidX1' ).value = coords.x1;

$( 'ctl00\_Content\_hidY1' ).value = coords.y1;

$( 'ctl00\_Content\_hidX2' ).value = coords.x2;

$( 'ctl00\_Content\_hidY2' ).value = coords.y2;

$( 'ctl00\_Content\_hidWidth' ).value = dimensions.width;

$( 'ctl00\_Content\_hidHeight' ).value = dimensions.height;

}

</script>

Ảnh của người dùng chỉ được cắt khi lên server. Để thực hiện được điều này thì dưới client sẽ gửi lên server vị trí của ảnh cần cắt. Phương thức thực hiện việc này:

public void CropFile(Int32 X, Int32 Y, Int32 Width, Int32 Height)

{

byte[] imageBytes = profile.Avatar.ToArray();

using (MemoryStream ms = new MemoryStream(imageBytes, 0,

imageBytes.Length))

{

ms.Write(imageBytes, 0, imageBytes.Length);

System.Drawing.Image img =

System.Drawing.Image.FromStream(ms, true);

Bitmap bmpCropped = new Bitmap(200, 200);

Graphics g = Graphics.FromImage(bmpCropped);

Rectangle rectDestination = new Rectangle(0, 0,

bmpCropped.Width, bmpCropped.Height);

Rectangle rectCropArea = new Rectangle(X,Y,Width,Height);

g.DrawImage(img, rectDestination, rectCropArea,

GraphicsUnit.Pixel);

g.Dispose();

MemoryStream stream = new MemoryStream();

bmpCropped.Save(stream,

System.Drawing.Imaging.ImageFormat.Jpeg);

Byte[] bytes = stream.ToArray();

profile.Avatar = bytes;

\_profileRepository.SaveProfile(profile);

}

\_view.ShowApprovePanel();

}

Ghi đè liên kết ảo

Nhằm mục đích tạo cho người dùng có môt địa chỉ trên trên website ví dụ như <http://tieuhoc.net/tenguoidung> hoặc muốn tạo ra một liên kết có ý nghĩa với người dùng, hoặc muốn giấu liên kết. Để thực hiện điều này chúng ta phải ghi đè liên kết.

Chúng ta sẽ phải tạo ra một đối tượng UrlRewrite kế thừa đối tượng HttpModule. Module HTTP sẽ thực thi xử lý HTTP. Module này có thay đổi được các yêu cầu được gửi đến từ người dùng.

Đăng ký xử lý ghi đè liên kết vào trong phương thức init của HttpModule

public void Init(HttpApplication application)

{

//let's register our event handler

application.PostResolveRequestCache +=

(new EventHandler(this.Application\_OnAfterProcess));

}

Trong phương thức xử lý ghi đè liên kết AppApplication\_OnAfterProcess

Phương thức này sẽ request liên kết từ client gửi đến nếu nó không phải là đường dẫn tồn tại thì phương thức này sẽ dựa vào thông tin trên liên kết đó để có thể xử lý ghi đè đến liên kết thật trên thực tế.

Nếu liên kết có dạng <http://tieuho.net/blogs/tieu-de-bai-viet.aspx> thì nó sẽ được xử lý như sau từ thông tin trang trên liên kết ta sẽ lấy được nội dung blog theo thông tin “tieu-de-bai-viet”. Từ đó hệ thống thực thi liên kết thực sự của nó là <http://tieuho.net/blogs/ViewPost.aspx?BlogID=1> giả sử như ID của blog bên trên là 1. Cách thức xử lý như vậy sẽ tương tự cho Tags, Group, Profile

if (!System.IO.File.Exists(application.Request.PhysicalPath))

{

#region BLOGS

if (application.Request.PhysicalPath.ToLower().Contains("blogs"))

{

string[] arr = application.Request.PhysicalPath.ToLower().Split('\\');

string blogPageName = arr[arr.Length - 1];

string blogUserName = arr[arr.Length - 2];

blogPageName = blogPageName.Replace(".aspx", "");

if (blogPageName.ToLower() != "profileimage" && blogUserName.ToLower() != "profileavatar")

{

if (blogPageName == "default")return;

Account acc = Account.GetAccountByUsername(blogUserName);

if (acc == null)return;

Blog blog = Blog.GetBlogByPageName(blogPageName, acc.AccountID);

context.RewritePath("~/blogs/ViewPost.aspx?BlogID=" + blog.BlogID.ToString());

}

else {return;}

}

#endregion

#region GROUPS

else if (application.Request.PhysicalPath.ToLower().Contains("groups") && \_WebContext.GroupID == 0)

{

string[] arr = application.Request.PhysicalPath.ToLower().Split('\\');

string groupPageName = arr[arr.Length - 1];

groupPageName = groupPageName.Replace(".aspx", "");

Group group = Group.GetGroupByPageName(groupPageName);

context.RewritePath("~/groups/viewgroup.aspx?GroupID=" + group.GroupID.ToString());

}

#endregion

#region TAGS

else if (application.Request.PhysicalPath.ToLower().Contains("tags"))

{

Tag tag = null;

int tagsPosition = 0;

string tagName;

string[] arr = application.Request.PhysicalPath.ToLowerInvariant().Split('\\');

for (int i = 0; i < arr.Length; i++)

{

if (arr[i].ToLower() == "tags")

{

tagsPosition = i;

}

if (tagsPosition > 0)

{

tagName = arr[i + 1];

tag = Tag.GetTagByName(tagName.Replace("-", " "));

break;

}

}

if (tag != null)

{

context.RewritePath("~/tags/tags.aspx?TagID=" + tag.TagID);

}

}

#endregion

#region PROFILES

else if (CoreSupport.StringCount(application.Request.Path, '/') <= 1)

{

string username = application.Request.Path.Replace("/", "");

Account account = Account.GetAccountByUsername(username);

if (account != null)

{

string UserURL = "~/Profiles/profile.aspx?AccountID=" + account.AccountID.ToString();

context.Response.Redirect(UserURL);

}

else

{

context.Response.Redirect("~/PageNotFound.aspx");

}

}

else

{ }

#endregion

Thông báo thông tin mới nhất của mình cho bạn bè

Khi người dùng đăng ảnh, viết blogs hoặc đăng trang thái lên thì những thông tin này sẽ được tự động thông báo đến bạn bè của họ.

public void AddNewBlogPostAlert(Blog blog)

{

Init();

alert = new Alert();

alert.AccountID = account.AccountID;

alert.AlertTypeID = (int)AlertType.AlertTypes.NewBlogPost;

alertMessage =

GetProfileImage(account.AccountID)

+ "<div class='title'><h2>" + GetProfileUrl(\_userSession.CurrentUser.Username) + "</h2></div>"

+ " đã viết blog: <b><a href='[rootUrl]blogs/" + \_webContext.Username + "/" + blog.PageName + ".aspx'>"

+ blog.Title + "</a></b>";

alert.Message = alertMessage;

SaveAlert(alert);

SendAlertToFriends(alert);

}

Kết bạn

Khi người dùng muốn kết bạn họ sẽ phải biết được thông tin của người đó như Email, hoặc tên đăng nhập và nhấn vào tìm kiếm. nếu người này đã tồn tại trong hệ thống thì hệ thống sẽ hiện thị ra cho người dùng. Người dùng nhấp chuột vào liên kết bạn để gửi yêu cầu kết bạn đến người này.

Phương thức SendInvitations dưới đây sẽ tạo ra một thông điệp từ yêu cầu kết bạn của người dùng thông điệp này có kèm theo một khóa chứng thực. Thông điệp này sẽ được gửi vào hộp thư của người bạn muốn kết bạn và gửi đến địa chỉ email của họ.

public string SendInvitations(Account sender, string ToEmailArray, string Message)

{

string resultMessage = Message;

foreach (string s in ToEmailArray.Split(new char[] { ',', ';' }))

{

FriendInvitation friendInvitation = new FriendInvitation();

friendInvitation.AccountID = sender.AccountID;

friendInvitation.Email = s;

friendInvitation.GUID = Guid.NewGuid();

friendInvitation.BecameAccountID = 0;

FriendInvitation.SaveFriendInvitation(friendInvitation);

Account account = Account.GetAccountByEmail(s);

if (account != null)

{

\_alertService.AddFriendRequestAlert(\_userSession.CurrentUser, account, friendInvitation.GUID,

Message);

}

SendFriendInvitation(s, sender.FirstName, sender.LastName, friendInvitation.GUID.ToString(), Message);

Message = sender.FirstName + " " + sender.LastName +

"Muốn kết bạn với bạn!<HR><a href=\"" + \_configuration.RootURL +

"Friends/ConfirmFriendInSite.aspx?InvitationKey=" + friendInvitation.GUID.ToString() + "\">" + \_configuration.RootURL +

"Friends/ConfirmFriendInSite.aspx?InvitationKey=" + friendInvitation.GUID.ToString() + "</a><HR>" + Message;

Messages m = new Messages();

m.Body = Message;

m.Subject = "Thư mời kết bạn";

m.CreateDate = DateTime.Now;

m.MessageTypeID = (int)MessageTypes.FriendRequest;

m.SentByAccountID = \_userSession.CurrentUser.AccountID;

m.MessageID = 0;

m.Save();

Int64 messageID = m.MessageID;

MessageRecipient sendermr = new MessageRecipient();

sendermr.AccountID = \_userSession.CurrentUser.AccountID;

sendermr.MessageFolderID = (int)MessageFolders.Sent;

sendermr.MessageRecipientTypeID = (int)MessageRecipientTypes.TO;

sendermr.MessageID = messageID;

sendermr.MessageStatusTypeID = (int)MessageStatusTypes.Unread;

sendermr.MessageRecipientID = 0;

sendermr.Save();

Account toAccount = Account.GetAccountByEmail(s);

if (toAccount != null)

{

MessageRecipient mr = new MessageRecipient();

mr.AccountID = toAccount.AccountID;

mr.MessageFolderID = (int)MessageFolders.Inbox;

mr.MessageID = messageID;

mr.MessageRecipientTypeID = (int)MessageRecipientTypes.TO;

mr.MessageRecipientID = 0;

mr.MessageStatusTypeID = 1;

mr.Save();

}

resultMessage += "• " + s + "<BR>";

}

return resultMessage;

}

Khi người bạn của bạn đọc được thông điệp này và nhấn vào liên kết được gửi đến thì người đó sẽ vào danh sách bạn bè của bạn và sẽ được thông báo cho mọi người biết.

* 1. Quá trình cài đặt